

MACHINE LEARNING APPLICATIONS IN PREDICTIVE
MAINTENANCE: A FOCUS ON CLUTCH FAILURES

by

Niraj Dev Pandey, M.Sc., B.Tech

Presented to the Swiss School of Business and Management Geneva
In Partial Fulfillment
Of the Requirements
For the Degree

DOCTOR OF BUSINESS ADMINISTRATION

SWISS SCHOOL OF BUSINESS AND MANAGEMENT GENEVA
SEPTEMBER 2024

MACHINE LEARNING APPLICATIONS IN PREDICTIVE
MAINTENANCE: A FOCUS ON CLUTCH FAILURES

by

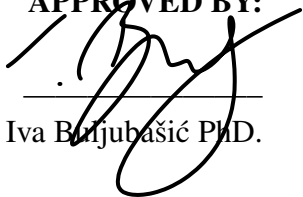
Niraj Dev Pandey, M.Sc., B.Tech

Supervised by

Sagar Bansal, DBA

Swiss School of Business and Management Geneva
SEPTEMBER 2024

APPROVED BY:



Iva Buljubašić Ph.D.

RECEIVED BY:

Admission Director

Acknowledgments

I express my profound gratitude to my advisor, Dr. Sagar Bansal, for his steadfast support, invaluable guidance, and scholarly insights that have profoundly shaped my entire doctoral journey. His mentorship has been instrumental in directing the trajectory of my research and fostering my intellectual development. I extend my appreciation to the esteemed faculty of the Swiss School of Business and Management, whose collective expertise and encouragement have significantly enriched my academic experience.

I wish to convey my sincere thanks to the members of my research committee for their insightful feedback and constructive critique, which has notably elevated the quality of this work. I extend special appreciation to ZF Group Friedrichshafen, Germany, for their collaborative spirit and the intellectually stimulating environment they facilitated for this research.

My heartfelt thanks go to my friends and family for their unwavering encouragement, understanding, and patience during the demanding phases of this arduous journey. Their unwavering emotional support provided the fortitude required to navigate the challenges encountered. I am deeply indebted to my dear friends Malin Kleefeld, Ankit Singh, Yazdan Asadi, Tjark Strich, and Manish Mishra, whose companionship offered moments of joy and distraction when needed. Additionally, my family members, notably Shivam, Prince, and Ashutosh, have been steadfast pillars of support.

This thesis is dedicated to all those individuals who, in varying capacities, contributed to this intellectual endeavor—acknowledging their collective influence, both substantial and subtle.

Abstract

MACHINE LEARNING APPLICATIONS IN PREDICTIVE MAINTENANCE: A
FOCUS ON CLUTCH FAILURES

NIRAJ DEV PANDEY
OCTOBER 2024

Dissertation Chair: Iva Buljubašić PhD.

This paper aims to present an overview of predictive maintenance in the automotive industry, focusing on machine learning (ML) techniques predicting failure of the Clutches. The paper also aims to address the pros and cons of such an approach for business. How it has and will impact the automotive industry in the coming future. Predictive maintenance is an important aspect of the automotive industry as it enables the proactive identification of potential failures in equipment and systems, reducing the risk of downtime and improving overall efficiency.

In recent years, machine learning techniques have emerged as powerful tools for predictive maintenance (PdM), enabling the development of more accurate and efficient predictive models. The paper will provide an overview of the various machine-learning techniques used in predictive maintenance for vehicle Clutch damage prediction. This research includes regression models, decision trees, and neural networks. Additionally, it will explore the challenges and opportunities associated with the implementation of predictive maintenance using machine learning in the automotive industry, including data quality, class imbalance, model interpretation, and organizational buy-in.

Additionally, the paper will present some case studies of predictive maintenance in the automotive industry that have successfully utilized machine learning techniques, highlighting the business benefits and potential of this approach. Moreover, our research highlights various instances of predictive maintenance implementation within the industry, providing insightful and pertinent content for senior executives at manufacturing and transportation companies. These decision-makers can gain valuable knowledge about

the advantages of predictive maintenance solutions and gain insight into the advancements made by their counterparts in this field. This paper contributes to the field of PdM by identifying and discussing significant research gaps in the field. Our analysis of the current literature highlights the need for further research in this area, and we propose several avenues for future investigation.

Keywords— Machine Learning; Classification; Predictive Maintenance; Automotive; Vehicle Clutch; Reliability; Lifetime prediction; Condition monitoring

Contents

List of Figures	vii
List of Tables	ix
List of Algorithms	x
Acronyms	xi
1 Introduction	1
1.1 Introduction	1
1.1.1 Predictive Maintenance — terminology and taxonomy	3
1.1.2 Statistical Predictive Maintenance (PdM)	5
1.1.3 Condition-Based Predictive Maintenance (PdM)	7
2 Literature Review	10
2.1 Literature Review	10
2.1.1 Current Challenges for Industry	10
2.1.2 Time Spent on Maintenance	11
2.1.3 PdM Satisfaction Among Industries	16
2.1.4 PdM Team Size in Industries	17
2.1.5 Which Components are Focus of PdM	18
2.1.6 Obstacle Faced in PdM	18
2.1.7 Data Collection Methods in PdM	19
2.1.8 PdM Adaptation to Monitoring Tools	20
2.2 Research Gap	22
3 Proposed Methodology	24
3.1 Proposed Methodology	24
3.1.1 Machine Learning	25
3.1.2 Types of Machine Learning Predictive Modelling	30
3.1.3 Tackling Class Imbalance	34

3.1.4	Validating the PdM Results	40
4	Experiments	42
4.1	Data Collection	42
4.2	Types of ECU's	43
4.3	Data Overview	44
4.4	Class Distribution	47
4.5	Analysis of Independent Variable	48
4.5.1	Distribution of Mileage	49
4.5.2	Mileage and Clutch Failure Relation	50
4.5.3	Converter Clutch Shifting	51
4.5.4	Counter of Retard	52
4.5.5	Counter Shifting of Clutch	53
4.5.6	Time in Clutch Shifting	54
4.5.7	Feature Correlation	55
4.6	Experiment Setup	57
4.6.1	Data Cleaning	58
4.7	Model Selection	62
4.7.1	Logistic Regression	63
4.7.2	Decision Tree	64
4.7.3	Support Vector Machine(SVM)	67
4.7.4	Random Forest	70
4.7.5	Artificial Neural Networks(ANN)	73
4.7.6	Models Weakness and Strength	75
5	Results	78
5.1	Evaluation Metric	78
5.2	Results: Imbalanced data-set	79
5.2.1	Result Imb Data: Logistic Regression	80
5.2.2	Result Imb Data: Decision Tree	83
5.2.3	Result Imb Data: Support Vector Machine	86
5.2.4	Result Imb Data: Random Forest	89
5.2.5	Result Imb Data: Artificial Neural Network	91
5.3	Sampling the Data-Set	95
5.4	Result: Balanced Data-set	98
5.4.1	Result Balnc Data: Logistic Regression	99
5.4.2	Result Balnc Data: Decision Tree	101
5.4.3	Result Balnc Data: Support Vector Machine (SVM)	103
5.4.4	Result Balnc Data: Random Forest	106
5.4.5	Result Balnc Data: Artificial Neural Network	109

6	Discussions & Conclusion	113
6.1	Discussions	113
6.1.1	Result Interpretation: Imbalance Data	113
6.1.2	Result Interpretation: Balanced Data	115
6.2	Limitations	116
6.2.1	Sensor Errors and Data Entry Mistakes	116
6.2.2	Inconsistencies in Measurement Methods	117
6.2.3	Data Fragmentation and Access Restrictions	117
6.2.4	Data Privacy Regulations	117
6.2.5	Ethical Considerations	118
6.2.6	Bias and Representativeness	118
6.3	Research Implications	118
6.3.1	Research Implications for Industries	118
6.3.2	Research Implications for Academia	119
6.4	Future Work	121
6.4.1	Expansion of Data Sources and Quality	121
6.4.2	Advanced Feature Engineering and Selection	121
6.4.3	Utilization of Advanced Machine Learning Algorithms	122
6.4.4	Predictive Maintenance Optimization	122
6.4.5	Integration of Real-Time Feedback Loops	122
6.4.6	Explainable AI and Model Interpretability	123
6.4.7	Scalability and Deployment in Real-World Applications	123
6.4.8	Collaborative Data Sharing and Model Standardization	123
6.4.9	Ethical and Privacy Considerations	123
6.5	Conclusion	125
	Bibliography	129

List of Figures

1	Types of predictive maintenance (MathWork 2021)	3
2	Various RUL approaches (MathWork 2021)	4
3	CXP Report: Current Challenges for Industries	11
4	CXP Report: Status of your predictive maintenance initiatives	12
5	2021 CFE Media Report on time spent on maintenance	13
6	2021 CFE Media Report on maintenance strategies in use	14
7	2021 CFE Media Report on solution to downtime	14
8	Plant Services Media Report on how satisfied industry is with PdM	16
9	Plant Services Report on how big PdM team is in Industry	17
10	Plant Services Report on which PdM technologies deployed	18
11	Plant Services Report on obstacle faced by industries	19
12	Plant Services Report on Data Collection Methods	20
13	Plant Services Report on PdM adaptation to monitoring tools	20
14	Basic types of Machine Learning (Shiksha 2022)	26
15	Machine Learning Algorithms (Akshay et al. 2021)	29
16	PdM: types of approaches (MathWork 2021)	30
17	Similarity Based PdM Approach (Aburakhia et al. 2022)	31
18	Over and Under Sampling	37
19	Hybrid Sampling Approach	39
20	Top 10 rows of the data set	44
21	Bottom 10 rows of the data set	45
22	Class Distribution of the Target Variable	47
23	Distribution of Mileage	49
24	Clutch Failure and Mileage Correlation	50
25	Histogram of Converter Clutch Shifting per km	51
26	Histogram of Counter of Retard per km	52
27	Histogram of Counter Shifting of Clutch	53
28	Histogram of Time in Shifting of Clutch	54

29	Correlation in Features	56
30	Confusion Matrix for Imb Logistic Regression Prediction	80
31	ROC and AUC for Imb Logistic Regression Prediction	82
32	Confusion Matrix for Imb Decision Tree Prediction	83
33	ROC Curve for Imb Decision Tree Prediction	85
34	Confusion Matrix for Imb SVM Prediction	86
35	ROC Curve for Imb SVM Prediction	88
36	Confusion Matrix for Imb Random Forest Prediction	89
37	Neural Network Training and Validation Accuracy for Imb Data-set . .	91
38	Confusion Matrix for Imb Neural Network Prediction	92
39	ROC Curve for Imb Neural Network Prediction	94
40	Class Distribution after Over Sampling	96
41	Confusion Matrix for Balanced Logistic Regression Prediction	99
42	ROC Curve for Balanced data Logistic Regression	100
43	Confusion Matrix for Balanced Decision Tree Prediction	101
44	ROC Curve for Balanced data Decision Tree	102
45	Confusion Matrix for Balanced Data SVM Prediction	103
46	ROC Curve for Balanced data SVM	105
47	Confusion Matrix for Balanced Data Random Forest Prediction	106
48	ROC Curve for Balanced data Random Forest	107
49	Neural Network Training and Validation Accuracy for Balnc Data-set .	109
50	Confusion Matrix of Balanced Data for Neural Network Prediction . . .	110
51	ROC Curve for Balanced data Neural Network	111

List of Tables

1	Dependent and Independent Variables	45
2	Data Set Information	46
3	Clean Data Set Information	60
4	Evaluation Metrics for Logistic Regression with Imbalanced Data	81
5	Evaluation Metrics for Decision Tree with Imbalanced Data	84
6	Evaluation Metrics for SVM with Imbalanced Data	87
7	Evaluation Metrics for Random Forest with Imbalanced Data	90
8	Evaluation Metrics for Neural Network with Imbalanced Data	93
9	Evaluation Metrics for Logistic Regression with Balanced Data	100
10	Evaluation Metrics for Decision Tree with Balanced Data	103
11	Evaluation Metrics for SVM with Balanced Data	104
12	Evaluation Metrics for Random Forest with Balanced Data	107
13	Evaluation Metrics for all models with imbalanced Data	114
14	Evaluation Metrics for all models with Balanced Data	115

List of Algorithms

1	SMOTE for Under-sampling	36
2	SMOTE for Over-sampling	38
3	Data Splitting: 70% Training, 30% Test	61
4	Logistic Regression Algorithm for Binary Classification	64
5	Decision Tree Algorithm	66
6	Support Vector Machine (SVM) Training Algorithm	68
7	Random Forest for Binary Classification	72
8	Binary Classification with Neural Network	74

Acronyms

PdM: Predictive Maintenance

ML: Machine Learning

DL: Deep Learning

DA: Data Analysis

ELM: Extreme Learning Machine

PHM: Predictive Maintenance

CBM: Condition-based maintenance

OEM: Original Equipment Manufacturer

RUL: Remaining Useful Life

CAN: Predictive Maintenance

ECU: Electronic Control Units

ANN: Artificial Neural Network

MLP: Multi Layer Perceptron

CNN: Convolution Neural Networks

LSTM: Long Short Term Memory

MRO: Maintenance, Repair, and Overhaul

PHM: Prognostics Health Management

DT: Decision Tree

RF: Random Forest

LR: Logistic Regression

SVM: Support Vector Machine

FC: Feature Correlation

Chapter 1

Introduction

1.1 Introduction

The use of data-driven methods like machine learning (ML) is rapidly becoming a norm in the automotive sector. As per (Magargle et al. 2017), beyond the challenges of developing complex products, companies are actively looking to monitor and manage the performance of developed products in operation to enhance safety, performance, and consumer satisfaction. As argued by (Theissler et al. 2021), the topic ranges from predictive maintenance (PdM) to predictive quality, including safety analytics, warranty analytics, as well as plant facilities monitoring. As per (Cachada et al. 2018) and (Bokrantz et al. 2020) various terms, including E-maintenance, Prognostics, digital twin, and Health Management (PHM), Maintenance 4.0, or Smart Maintenance, are used to describe the development of methods that analyze, predict, or anticipate performance issues that could compromise the safety and integrity of automotive components, products, and systems. As argued by (Becker et al. 2017) and (Kim et al. 2013) that the increasing demand for cost-efficient technical solutions is driven by the developments towards automated driving and the transformation of the drive-train. Ensuring vehicles' functional safety and reliability over their lifetime is crucial.

A significant investments directed towards industrial machinery and vehicle fleets, maintenance plays a crucial role in facilitating their extended utilization and maximizing return on investment. Nonetheless, the current maintenance processes in place exhibit a

lack of efficiency, creating room for enhancement. To this end, companies are increasingly embracing digital technologies such as the Internet of Things (IoT) and predictive analytics to harness the data streams from these assets and transform them into value. By leveraging predictive algorithms to process the data, companies can proactively identify potential asset failures and take corrective actions. This provides an opportunity to boost utilization and productivity, while at the same time improving the consumer experience argued (Milojevic et al. 2018).

The airline is another major form of mobility that has been affected extensively by predictive maintenance. In accordance with recent studies by (Meyendorf et al. 2018), airline processes, specifically maintenance, are the primary cause of approximately 42 % of delayed flights. To improve their performance, airlines aim for better maintenance, repair, and overhaul (MRO) with improved quality, cost, and turnaround times within budget and schedule. A significant percentage of companies (over 90%) have reported that their current maintenance processes are inefficient. However, it remains to be seen whether these companies are prepared to undertake measures to optimize these processes said, (Milojevic et al. 2018).

As per (Theissler et al. 2021), there has been considerable research on predictive maintenance (PdM) and machine learning (ML) for automotive systems, but there is a notable research gap in the current state-of-the-art solutions. Their extensive review paper search concludes that none of the existing solutions have covered Condition-based damage prediction for vehicle clutches. Thus, Analyzing clutch damages and predicting their failure in vehicles using statistical data has been untouched as far as we know. This paper aims to address this research gap by proposing a novel solution for predicting clutch failure using PdM and ML techniques. The model is trained using data collected from various vehicles, including parameters such as clutch shifting, engine mileage, and retard shifting and counters of these features. The model aims to predict the potential clutch damage before it occurs, thereby reducing maintenance costs and minimizing downtime. The model is developed using machine learning algorithms and validated through extensive testing. The results show that the model can accurately predict clutch damage with high accuracy, demonstrating its potential to improve vehicle maintenance

and reduce operating costs.

The paper is structured as follows: In section 1.1.1, we provide definitions for the key terms discussed in this paper. The Literature Review section 2.1 examines the latest research and developments related to maintenance, machine learning (ML), and their application in the automotive industry. This section is based on a survey that explores the current state of the industry. We identify research gaps in section 2.2, and then outline our proposed methodology for predicting damages in the Predictive Maintenance (PdM) domain in section 3.1. Finally, we conclude this paper in section 6.5.

1.1.1 Predictive Maintenance — terminology and taxonomy

As per (Werbińska-Wojciechowska 2019) and (Theissler et al. 2021) the maintenance strategies can be subdivided in various ways, a commonly used categorization is:

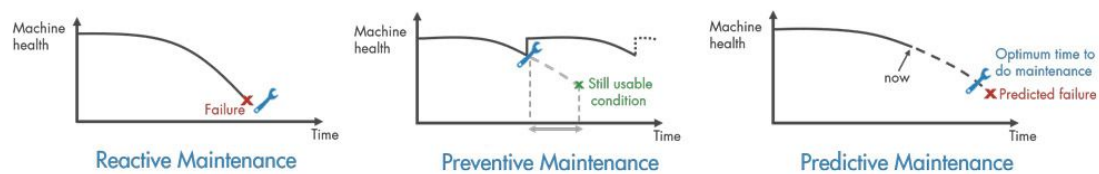


Figure 1: Types of predictive maintenance (MathWork 2021)

1. **Corrective maintenance:** The approach known as corrective maintenance, which is also referred to as reactive maintenance, fix-upon-failure, or run-to-failure, is implemented to repair a system or its components only after they have already experienced a failure.
2. **Preventive maintenance:** Preventive maintenance relies on pre-scheduled maintenance intervals, typically utilizing fixed time intervals and occasionally incorporating a system's usage (e.g. the mileage of vehicles). The primary objective of preventive maintenance is to perform repairs on a system before any failure occurs, without considering the current health status of the system.

3. **Predictive maintenance (PdM):** The primary objective of PdM is to anticipate the ideal timing for maintenance actions, utilizing information about the system's health state and/or past maintenance data. As per (Xiang et al. 2018) this approach seeks to prevent premature and costly repairs to a system, while also ensuring that repairs are performed promptly before any failure. Advanced techniques within PdM strive to predict the anticipated time of a failure, thereby providing an estimate of the remaining useful life (RUL).

As per (Chen et al. 2020) while condition-based maintenance (CBM) is often used as a synonym for predictive maintenance, CBM is viewed as a subcategory of PdM, subdividing PdM into:

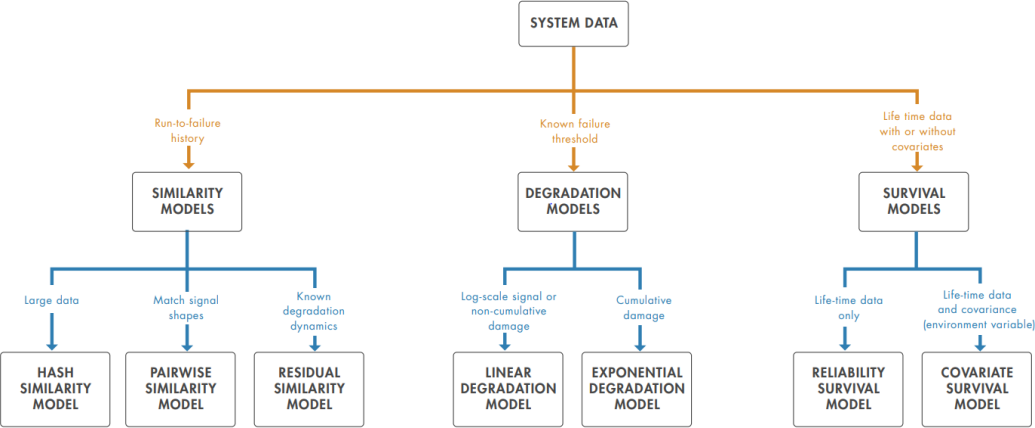


Figure 2: Various RUL approaches (MathWork 2021)

1.1.2 Statistical Predictive Maintenance (PdM)

Statistical Predictive Maintenance (PdM) is a data-driven approach used in various industries, including automotive, to optimize the maintenance schedules of vehicles and equipment. Unlike traditional preventive maintenance, which relies on fixed intervals for service, statistical PdM leverages historical and aggregated data to predict potential failures or maintenance needs before they occur. What sets statistical PdM apart from other types of predictive maintenance is that it relies on data not directly tied to the state of an individual vehicle, but rather on broader datasets, such as historical maintenance records or data gathered from a fleet of vehicles or an entire vehicle population.

The foundation of statistical PdM lies in its use of large datasets that capture patterns of wear, failure rates, and maintenance needs across many vehicles. These datasets often include information such as the frequency of repairs, typical time-to-failure for specific components, and the outcomes of previous maintenance activities. By analyzing these historical trends, statistical models can be developed to forecast when a vehicle or a specific component is likely to require maintenance, even if the vehicle in question has not yet shown any signs of malfunction.

One key advantage of statistical PdM is that it allows manufacturers, fleet managers, and service providers to anticipate issues that may arise based on the experiences of other vehicles in the same category. For example, data from a fleet of similar vehicles operating under similar conditions can provide valuable insights into the lifespan of specific parts, such as brake pads or transmission systems. By analyzing this data, it becomes possible to predict with a high degree of accuracy when a particular part in a given vehicle will likely need to be replaced, thus preventing unexpected breakdowns and reducing downtime.

Furthermore, statistical PdM allows for more effective resource allocation, as it enables organizations to schedule maintenance activities only when necessary, rather than adhering to rigid time-based intervals. This approach not only reduces costs associated with unnecessary maintenance but also ensures that vehicles are serviced at the optimal time, thus extending their operational lifespan and enhancing overall efficiency.

Another significant advantage of statistical PdM is its applicability to entire vehicle populations or fleets. This is especially beneficial for industries that operate large fleets of vehicles, such as logistics companies, public transportation systems, or rental car services. By leveraging aggregated data from the entire fleet, these organizations can implement predictive maintenance strategies that minimize the risk of sudden failures and maximize vehicle uptime. For instance, if a particular make and model of vehicle is found to have a higher failure rate for a specific component after a certain mileage, fleet managers can proactively schedule maintenance for those vehicles before issues arise, thus avoiding costly repairs and disruptions.

However, it is important to note that statistical PdM is not without its limitations. Since it relies on historical and population-level data, it may not always account for the unique conditions or usage patterns of an individual vehicle. For instance, a vehicle that operates in extreme climates or under unusually demanding conditions may experience wear and tear at a different rate compared to the broader fleet. In such cases, statistical PdM may not provide the most accurate predictions, as it is inherently based on generalizations derived from the larger dataset.

Despite this limitation, statistical PdM offers a powerful tool for improving maintenance practices in the automotive industry. By harnessing the power of big data and statistical analysis, it enables a shift from reactive maintenance approaches to proactive and data-driven strategies. This transition not only enhances the reliability and performance of vehicles but also delivers significant cost savings and operational efficiencies for businesses and consumers alike.

In conclusion, statistical PdM represents a significant advancement in the field of automotive maintenance. By utilizing historical maintenance data and insights from entire fleets or vehicle populations, it allows for more accurate predictions of maintenance needs, reducing the likelihood of unexpected breakdowns and extending the lifespan of vehicles. While it may not account for every individual vehicle's unique conditions, its ability to draw from large datasets provides a robust framework for optimizing mainte-

nance schedules and improving overall vehicle performance.

1.1.3 Condition-Based Predictive Maintenance (PdM)

Condition-Based Predictive Maintenance (PdM) is a sophisticated and highly effective maintenance strategy that relies on real-time data to assess the current health of a system and make informed maintenance decisions. Unlike traditional maintenance approaches, which depend on predetermined intervals, or statistical PdM, which draws from historical or population-level data, condition-based PdM continuously monitors the actual operating conditions and performance of individual vehicles. This approach allows for a more precise and timely maintenance intervention, enhancing vehicle reliability, reducing downtime, and minimizing costs.

The central concept behind condition-based PdM is the real-time monitoring of key components and systems within a vehicle. Using a variety of sensors, the system collects data on parameters such as temperature, vibration, pressure, fluid levels, and wear patterns. This data is then analyzed to assess the current health of each component and determine whether maintenance or repairs are needed. By doing so, condition-based PdM enables maintenance teams to address issues as they arise, based on actual usage and condition, rather than relying on preset schedules or generalized data.

One of the most important advantages of condition-based PdM is that it provides insights specific to each individual vehicle, ensuring that maintenance decisions are tailored to the unique operating conditions and performance of that vehicle. For example, a vehicle operating in harsh environments, such as extreme heat or cold, will experience different wear and tear compared to a vehicle operating in more moderate conditions. The real-time data collected through sensors can detect early signs of component fatigue, fluid degradation, or excessive wear, allowing for immediate maintenance actions before a failure occurs. This level of precision ensures that vehicles receive the necessary care exactly when they need it, reducing the risk of unexpected breakdowns and enhancing operational efficiency.

Real-time data analysis in condition-based PdM also enables continuous optimization of maintenance strategies. With the ability to detect minor anomalies or gradual performance deterioration, maintenance teams can take a proactive approach to addressing issues before they escalate into significant problems. For example, if sensors detect abnormal vibrations in the engine or drivetrain, this could indicate the early stages of mechanical wear. Rather than waiting for the part to fail completely, maintenance can be scheduled to replace or repair the affected component, thereby preventing a larger and more expensive repair later.

In addition to improving vehicle reliability, condition-based PdM offers considerable cost savings. Traditional preventive maintenance, which follows fixed schedules, often results in unnecessary service appointments and premature replacement of components that may still have significant operational life remaining. Conversely, condition-based PdM ensures that maintenance is only performed when needed, based on the actual condition of the vehicle. This targeted approach reduces the frequency of maintenance interventions, lowers material costs, and extends the lifespan of vehicle components by avoiding premature replacements.

Furthermore, condition-based PdM plays a crucial role in enhancing safety. Real-time monitoring can detect critical issues, such as brake system wear, tire pressure anomalies, or engine overheating, that may compromise vehicle safety if left unaddressed. By alerting operators or maintenance personnel to these issues immediately, condition-based PdM helps prevent accidents caused by component failure or malfunction. This is especially valuable for commercial fleets or public transportation vehicles, where safety is a top priority.

The implementation of condition-based PdM also enables better planning and resource allocation. Real-time insights into vehicle health allow for more precise scheduling of maintenance activities, reducing unscheduled downtime and improving fleet availability. For businesses that rely on vehicle fleets, such as logistics companies, delivery services, or public transportation, minimizing downtime is essential for maintaining operational efficiency and meeting service commitments. With condition-based PdM,

vehicles can be taken out of service for maintenance only when necessary, and downtime can be planned around operational needs, minimizing disruption.

However, condition-based PdM does present certain challenges. The initial investment in sensor technology, data processing systems, and analytics platforms can be significant, particularly for organizations with large vehicle fleets. Additionally, the continuous flow of real-time data requires robust infrastructure and expertise to manage and analyze the information effectively. Ensuring that the data is accurate and interpreted correctly is essential for making the right maintenance decisions, which may require skilled personnel and advanced analytical tools. Moreover, integrating condition-based PdM into existing maintenance operations may require changes in workflow and processes, which can pose challenges during the implementation phase.

Despite these challenges, the benefits of condition-based PdM are substantial. The ability to monitor the actual condition of vehicles in real time and make data-driven maintenance decisions significantly improves vehicle reliability, safety, and cost efficiency. It allows organizations to move away from reactive maintenance models, where issues are addressed only after they have occurred, and toward a proactive maintenance strategy that maximizes vehicle uptime and operational performance.

In conclusion, condition-based predictive maintenance represents a significant advancement in automotive maintenance practices. By leveraging real-time data to assess system health, it enables more precise, timely, and cost-effective maintenance decisions. The adoption of condition-based PdM not only enhances the reliability and safety of individual vehicles but also contributes to broader operational efficiencies, particularly for organizations managing large fleets. As sensor technology and data analytics continue to evolve, condition-based PdM is poised to play an increasingly important role in the future of automotive maintenance, driving improvements in both vehicle performance and business outcomes.

Chapter 2

Literature Review

2.1 Literature Review

In contrast to statistical predictive maintenance (PdM), condition-based PdM leverages operational data from individual vehicles to determine the status of the overall system or specific components. This approach facilitates component-specific maintenance decisions. The detection of faults is a critical strategy for predicting failures. Identifying faults early can halt their progression, and appropriate measures can be taken to avoid breakdowns (Theissler et al. 2021). Thus increasing customer satisfaction and cost saving. This also provides original equipment manufacturers (OEM) with a view of their particular automotive parts and the cause of their failures.

2.1.1 Current Challenges for Industry

Let's have a look at some surveys to understand the current state-of-the-art and the status of maintenance strategies in industries. Figure 3 depicts a report by CXP Group where (Milojevic et al. 2018) shows the challenges faced by the industry they interviewed. The figure is divided into two parts. Namely, major challenges and minor challenges. See the figure below for references.

Figure 3 by (Milojevic et al. 2018) shows that there are various challenges faced by European businesses. Such as unexpected downtime and emergency repairs (90%),

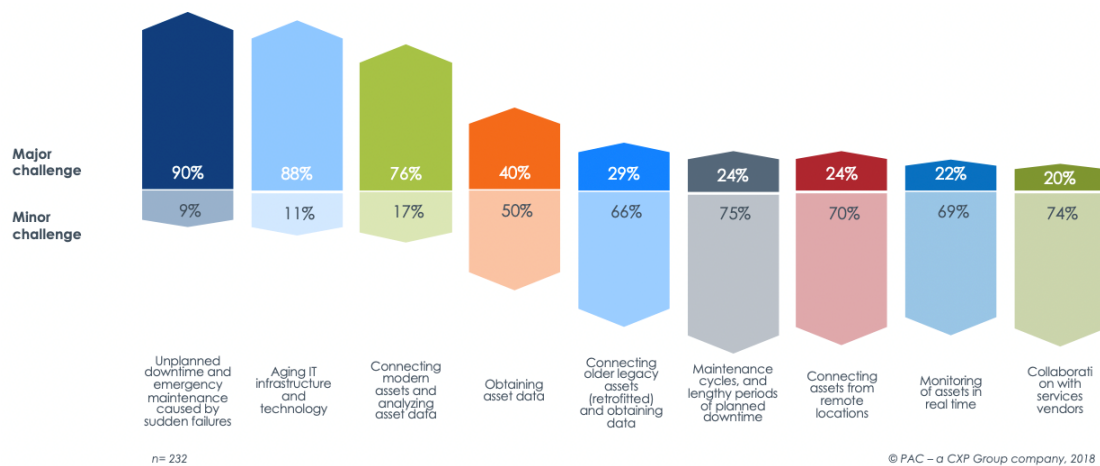


Figure 3: CXP Report: Current Challenges for Industries

outdated IT infrastructure and technology (88%), integration of modern assets and data analysis (76%), acquiring asset data (40%), integration of older legacy assets and obtaining data (29%), maintenance cycles (24%), integration of assets in remote locations (24%), real-time asset monitoring (22%), and collaboration with vendors (20%).

Upon examining the outcomes of the report by (Milojevic et al. 2018), it can be observed that the market is highly dynamic, with 55% of businesses have initiated pilot projects for predictive maintenance (See figure 4). The transportation industry has taken the lead, with 62% of companies in this field executing these initiatives. In addition to this, 55% of the companies are beyond the planning and evaluation stage of predictive maintenance initiatives.

2.1.2 Time Spent on Maintenance

Based on a survey conducted by (Mcleman et al. 2021) on a sample of plants, it was found that approximately 31% of the plants spend less than 20 hours per week on maintenance-related tasks. See the figure 5 below. The figure indicates that a significant proportion of plants allocate relatively fewer hours to maintenance activities. On the other hand, the average time spent on such tasks among the surveyed plants was found to be around 33 hours per week, which suggests that the majority of plants invest a

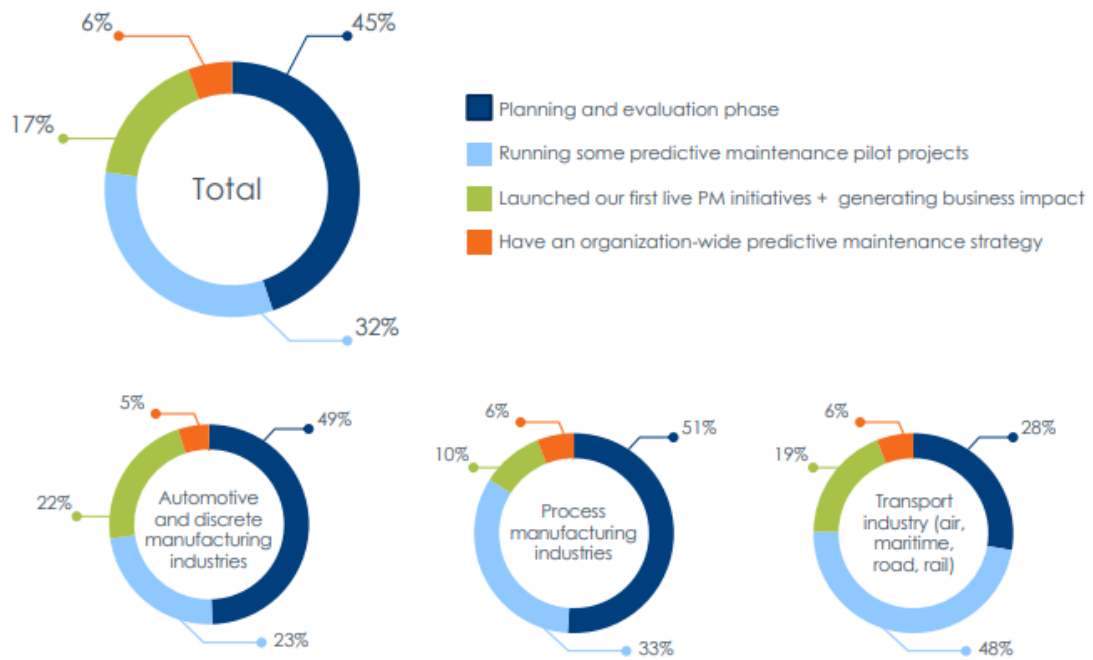


Figure 4: CXP Report: Status of your predictive maintenance initiatives

considerable amount of time in maintenance-related activities. It is important to note that these findings are based on a specific sample and may not be representative of the entire population of plants.

As per the survey by (Mcleman et al. 2021) approximately 52%, utilize a computerized maintenance management system to monitor and manage their maintenance tasks. Furthermore, the figure 6 below also revealed that other maintenance systems that are popularly employed include in-house created spreadsheets and schedules, which were utilized by approximately 49% of the plants surveyed. Additionally, Preventive maintenance was found to be in use by around 88% of the plants included in the study.

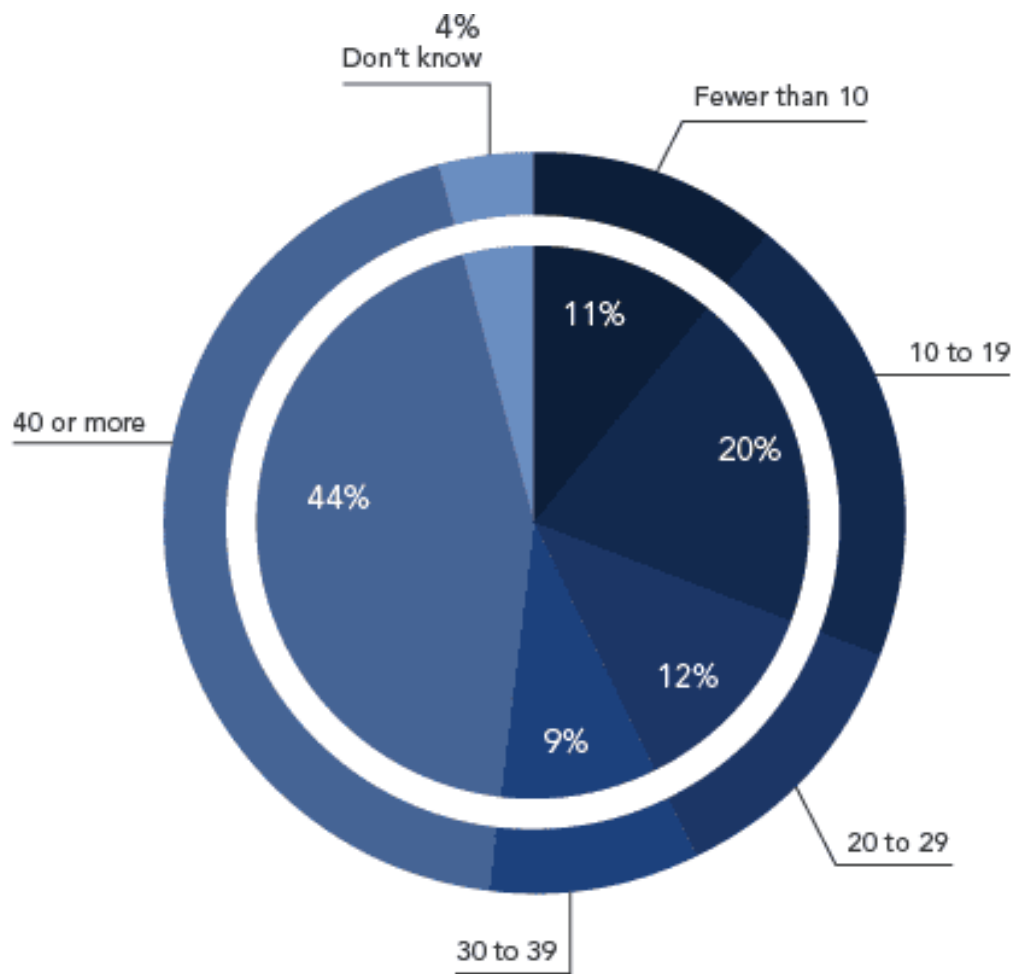


Figure 5: 2021 CFE Media Report on time spent on maintenance

According to a survey (figure 7), a majority of plants (56%) have plans to upgrade their equipment as a means of addressing unscheduled downtime. Additionally, 47% of the plants plan to improve training programs and increase training frequency, while 45% are considering adopting a predictive maintenance strategy (Mcleman et al. 2021).

Which of the following maintenance strategies and tools are present within your plant?

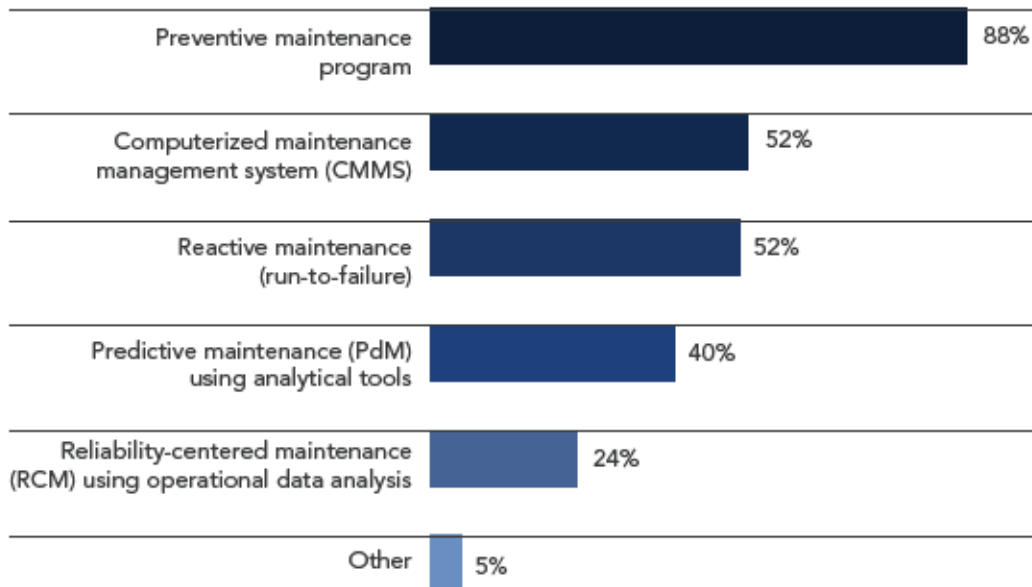


Figure 6: 2021 CFE Media Report on maintenance strategies in use

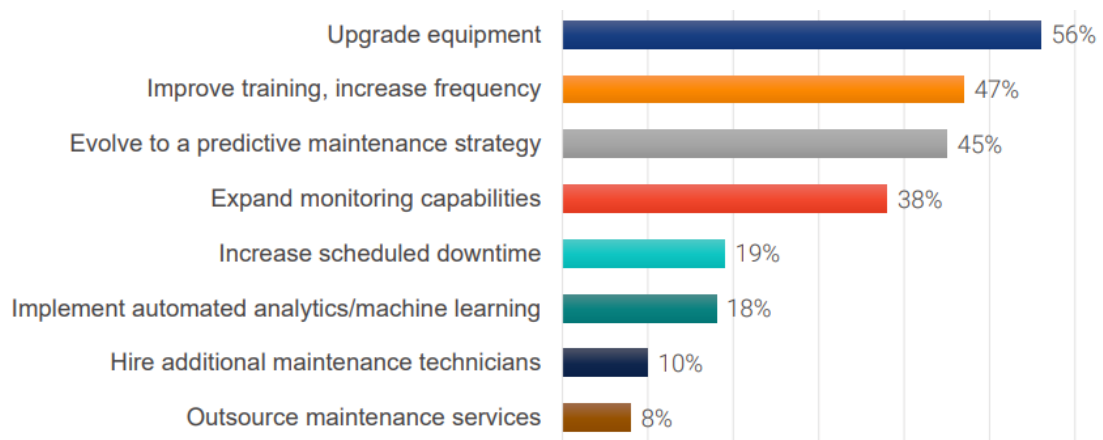


Figure 7: 2021 CFE Media Report on solution to downtime

Condition-based PdM typically employs anomaly detection or classification methods to accomplish this objective. The specific approach (unsupervised, semi-supervised, or supervised learning) used will depend on the data and label availability. As (Wong et al. 2016) Propose a method for the detection of faults in engines focusing on simultane-

ous faults, i.e. multiple single faults occurring concurrently. They used an ensemble of Bayesian extreme learning machines (ELM) in a supervised Machine Learning fashion. Similar efforts were carried out by others such as (Zhong et al. 2018) and (Wolf et al. 2018). However, most of these researches focus on a part of the automotive vehicle. Some focus on breaks (Jegadeeshwaran et al. 2015), some on battery (Sankavaram et al. 2012), and some on the power train or steering.

These are a few notable findings by (Milojevic et al. 2018):

- A considerable proportion of companies (55%) are currently piloting predictive maintenance initiatives, with a notable 23% generating measurable business impact.
- Almost half (49%) of the companies have already invested in predictive maintenance initiatives and plan to increase their investment in the next two years.
- Data security and privacy concerns are major inhibitors of predictive maintenance developments for 89% of the companies, while a significant lack of internal capabilities also poses a challenge.
- To overcome these obstacles, companies seek assistance from vendors to facilitate their journey toward enhanced operational efficiency.

As argued by (Theissler et al. 2021) in their survey paper, the most commonly employed machine learning (ML) methods are Artificial Neural Networks (ANNs), which encompasses standard neural networks such as Multi-Layer Perceptrons (MLPs) and their variations. Moreover, an escalating number of research articles adopt neural network models, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), Extreme Learning Machines (ELMs), and autoencoders.

The survey paper is available here: ¹

¹<https://www.sciencedirect.com/science/article/pii/S0951832021003835>

2.1.3 PdM Satisfaction Among Industries

Plant Services (Wilk 2022) conducted its PdM Survey in March 2022, gathering insights from maintenance and reliability practitioners across the industry. The survey received an overwhelming response, with participation from more than 100 professionals who shared their perspectives and experiences. The results of this year’s PdM Survey provide valuable insights into the current state of the industry, shedding light on the trends, challenges, and advancements in the field of predictive maintenance. By analyzing the survey data, we can gain a comprehensive understanding of the prevailing practices and emerging approaches adopted by maintenance professionals. In this research paper, we will delve into the findings of the PdM Survey, exploring the key takeaways and implications for the industry. Through a detailed examination of the survey results, we aim to provide a comprehensive overview of the state of the industry, offering valuable insights that can inform decision-making, drive improvements, and contribute to the advancement of maintenance and reliability practices.

	2014	2016	2018	2020	2022
Not effective	15.5%	15.6%	12.5%	16.9%	8.8%
Needs some improvement	40.3%	49.4%	45.3%	32.5%	42.5%
Satisfactory	24.8%	18.2%	21.9%	20.8%	26.3%
Effective	15.5%	14.3%	15.6%	23.4%	17.5%
Very effective	3.9%	2.6%	4.7%	6.5%	5.0%

Figure 8: Plant Services Media Report on how satisfied industry is with PdM

In a notable shift, the recent survey conducted by Plant Services witnessed a significant decrease in the proportion of respondents expressing dissatisfaction with their PdM program, marking the first time this figure has fallen into the single digits, standing at approximately 9%. This positive development is a marked improvement from previous surveys, where the dissatisfaction rate consistently exceeded 15% in most years (see figure 8).

Conversely, the percentage of respondents indicating satisfaction with their PdM program experienced a decline, falling below the 50% threshold. Specifically, only 48.7% of participants considered their program to be satisfactory or better. Notably, program sentiment this year shifted toward a middle ground, as nearly 70% of respondents categorized their program as either "satisfactory" or "in need of improvement."

2.1.4 PdM Team Size in Industries

Despite the multitude of challenges faced by plant teams since early 2020, one notable aspect that remained remarkably consistent is the size of the maintenance and reliability team (see figure 9). This finding is significant considering the persistent hiring and retention issues reported by many plants, further highlighting the resilience and stability exhibited by maintenance and reliability teams in the face of various pressures.

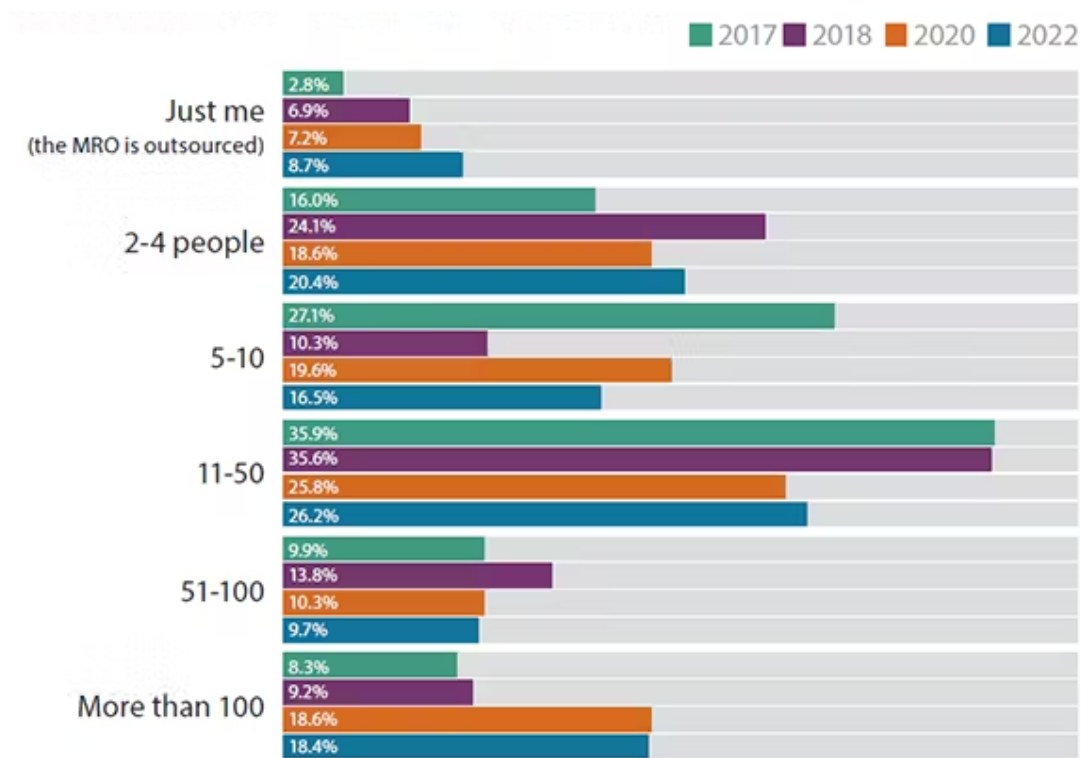


Figure 9: Plant Services Report on how big PdM team is in Industry

Within the survey, participants were queried about the specific predictive main-

tenance technologies they utilize. Notably, this year’s results reaffirm the consistent prominence of five key technologies that have consistently topped the survey rankings in previous years: vibration analysis, infrared thermography, ultrasound testing, oil analysis, and electrical motor testing.

2.1.5 Which Components are Focus of PdM

In addition to exploring the adoption of predictive maintenance technologies, the survey also inquired about the types of assets managed through PdM solutions. Notably, rotational assets and electrical systems emerged as the leading categories (see figure 10). These findings indicate the significance of managing these specific asset types and suggest a focus on the optimization and maintenance of rotational machinery and electrical systems within organizations utilizing PdM practices.

	Using now				In this year's budget				Within 3 years				No plans			
	2014	2018	2020	2022	2014	2018	2020	2022	2014	2018	2020	2022	2014	2018	2020	2022
Vibration	60.0%	64.1%	70.1%	59.5%	5.8%	7.8%	5.2%	12.7%	12.9%	12.5%	10.4%	15.2%	21.3%	15.6%	14.3%	12.7%
Ultrasound	45.5%	60.9%	44.7%	41.6%	5.2%	6.3%	7.9%	9.1%	16.9%	7.8%	21.1%	20.8%	32.5%	25.0%	26.3%	28.6%
Acoustic	24.7%	21.9%	21.1%	28.2%	6.5%	4.7%	10.5%	6.4%	14.3%	15.6%	18.4%	23.1%	54.5%	57.8%	50.0%	42.3%
Corrosion	33.8%	28.6%	39.5%	28.6%	7.8%	11.1%	14.5%	10.4%	14.9%	14.3%	9.2%	16.9%	43.5%	46.0%	36.8%	44.2%
Infrared	65.8%	71.4%	56.6%	55.1%	3.9%	3.2%	19.7%	15.4%	15.5%	6.3%	7.9%	11.5%	14.8%	19.0%	15.8%	17.9%
Oil analysis	62.3%	74.6%	63.6%	59.5%	4.5%	6.3%	13.0%	11.4%	15.6%	4.8%	5.2%	11.4%	17.5%	14.3%	18.2%	17.7%
Predictive modeling software	17.5%	11.1%	15.6%	14.3%	6.5%	6.3%	13.0%	11.7%	25.3%	33.3%	22.1%	28.6%	50.6%	49.2%	49.4%	45.5%
Electric motortesting	50.0%	42.9%	44.2%	50.0%	5.8%	9.5%	15.6%	12.8%	14.9%	17.5%	13.0%	12.8%	29.2%	30.2%	27.3%	24.4%

Figure 10: Plant Services Report on which PdM technologies deployed

2.1.6 Obstacle Faced in PdM

The prevailing trend observed in various categories aligns with our findings, with one exception - the "limited engineering resources" category. Interestingly, respondents indicated that this particular factor posed a greater stressor in 2022 (66.2%) compared to 2020 (60.9%). While the strain caused by limited budgets may be a persistent stressor, plant teams are currently experiencing the significant impact of constrained staffing resources just as intensely (see figure 11). This underscores the growing significance of addressing challenges related to limited headcount and its implications for maintenance and reliability operations.

	Not a factor				Low				Medium				High			
	2014	2018	2020	2022	2014	2018	2020	2022	2014	2018	2020	2022	2014	2018	2020	2022
Budget constraints	5.8%	10.9%	4.7%	7.0%	14.3%	20.3%	7.8%	26.8%	37.7%	48.4%	35.9%	40.8%	42.2%	20.3%	51.6%	25.4%
Undefined financial benefits	9.1%	17.2%	6.3%	12.7%	18.8%	29.7%	23.4%	25.4%	46.1%	35.9%	32.8%	38.0%	26.0%	17.2%	37.5%	23.9%
Undefined operational benefits	16.9%	15.6%	7.8%	14.1%	26.6%	29.7%	29.7%	31.0%	39.6%	40.6%	37.5%	42.3%	16.9%	14.1%	25.0%	12.7%
Limited engineering resources	16.2%	12.5%	9.4%	11.3%	22.7%	28.1%	29.7%	22.5%	42.9%	32.8%	35.9%	46.5%	18.2%	26.6%	25.0%	19.7%
Poor program execution	24.0%	17.2%	7.8%	22.5%	32.5%	37.5%	39.1%	33.8%	30.5%	29.7%	29.7%	35.2%	13.0%	15.6%	23.4%	14.1%
Lack of executive support	24.2%	20.3%	9.4%	25.4%	23.5%	29.7%	23.4%	29.6%	35.9%	29.7%	32.8%	26.8%	16.3%	20.3%	34.4%	18.3%

Figure 11: Plant Services Report on obstacle faced by industries

2.1.7 Data Collection Methods in PdM

In their continuous exploration of data collection methods, author (Wilk 2022) have consistently pondered the potential transition from analog paper-based systems to digital alternatives. However, the results of this year’s survey indicate that we have not yet reached the tipping point. Notably, 62.3% of respondents continue to rely on paper-based systems as their primary method of data collection, as depicted in (see figure 12). Interestingly, the use of consumer-grade smartphones experienced a slight increase since 2020, with a modest rise to 27.5%. Surprisingly, both wireless and internet-enabled sensors witnessed a decline in usage among respondents, deviating from expectations. These findings highlight the persistence of paper-based systems in data collection and suggest that while progress is being made with the integration of consumer-grade smartphones, the transition to fully digital methods has yet to materialize

	2016	2018	2020	2022
Paper-based system	64.9%	66.1%	62.5%	62.3%
Handheld data collector	55.8%	62.7%	60.9%	59.4%
Embedded sensors	53.2%	57.6%	54.7%	50.7%
Wireless sensors	23.4%	28.8%	42.2%	31.9%
Internet-enabled sensors	18.2%	18.6%	26.6%	15.9%
Industrial-gradesmartphone	9.1%	10.2%	10.9%	10.1%
Consumer-gradesmartphone	20.8%	15.3%	21.9%	27.5%
Industrial-grade tablet	15.6%	25.4%	29.7%	21.7%
Consumer-grade tablet	23.4%	13.6%	18.8%	18.8%
Industrial-grade PC	37.7%	33.9%	37.5%	39.1%
Consumer-grade PC	40.3%	42.4%	40.6%	40.6%

Figure 12: Plant Services Report on Data Collection Methods

2.1.8 PdM Adaptation to Monitoring Tools

Research team conducted a comprehensive survey exploring the prevalence and utilization of performance monitoring systems (PdM) within industrial settings. Among the questions posed to participants were several queries related to the degree of integration between their PdM infrastructure and external systems. We hypothesized that many such connections would exist based on prior reports indicating interoperability benefits across domains like healthcare, logistics, and transportation infrastructures said (see figure (Wilk 2022)).

	Using now		In this year's budget		Within 3 years		No plans	
	2020	2022	2020	2022	2020	2022	2020	2022
Reliability solutions	36.7%	27.5%	5.0%	10.1%	16.7%	18.8%	41.7%	43.5%
Historian	41.7%	23.2%	6.7%	7.2%	8.3%	10.1%	43.3%	59.4%
EAM/CMMS system	48.3%	31.9%	13.3%	13.0%	10.0%	26.1%	28.3%	29.0%
EH&S system	13.3%	8.7%	13.3%	7.2%	13.3%	21.7%	60.0%	62.3%
ERP system	26.7%	23.2%	11.7%	4.3%	16.7%	15.9%	45.0%	56.5%
Cloud-based analytics	15.0%	21.7%	11.7%	10.1%	23.3%	23.2%	50.0%	44.9%

Figure 13: Plant Services Report on PdM adaptation to monitoring tools

Consequently, we were surprised by collected data by (Wilk 2022), which exposed a drastic decrease in connectivity for most categories under consideration – specifically for tools used to collect process data (data historians; -44%) as well as software applications intended to manage assets through condition monitoring (EAM/CMMS systems; -34%) (see figure 13). These percentages suggest pronounced downturns in user adoption compared to earlier estimations, thereby urgently necessitating further probes into the causes behind these trends. By divulging these patterns in detail, our article intends not only to summarize current knowledge but also to prompt follow-up investigations focused on restoring the balance between autonomous PdM functionality and seamless coordination with complementary tools. Such research may eventually guide practitioners toward better decision-making concerning their technology investments – ultimately improving operational efficiency and competitiveness.

The survey paper is available here: ²

²<https://www.sciencedirect.com/science/article/pii/S0951832021003835>

2.2 Research Gap

The remarkable progress of machine learning (ML) can be attributed to three main factors, namely the accessibility of data, the breakthroughs in algorithm development, and the advancements in computational power. The selection of ML methods for maintenance modeling is determined by the specific application requirements and, consequently, the available data (Theissler et al. 2021).

Research on predictive maintenance in the automotive sector has primarily focused on technical aspects such as sensor technologies, machine learning algorithms, and fault detection. However, there is a research gap in understanding the business perspective of implementing predictive maintenance in the automotive industry. Specifically, there is a need to explore the business case for predictive maintenance, including the potential ROI, cost-benefit analysis, and the organizational changes required for successful implementation. Additionally, the research could focus on identifying the key success factors and barriers to adoption from a business perspective, including factors such as organizational culture, change management, and the role of leadership in driving adoption. Few other gaps we found in existing research are as follows:

- **Lack of standardized metrics:** While there is growing interest in using predictive maintenance in the automotive sector, there is a lack of standardized metrics to assess the effectiveness of such programs. Without clear metrics, it is difficult for businesses to assess the Return On Investment (ROI) of implementing predictive maintenance programs and compare the effectiveness of different approaches.
- **Limited understanding of customer needs:** While predictive maintenance has the potential to improve customer satisfaction by reducing downtime and improving the reliability of vehicles, there is a lack of research on customer needs and preferences related to predictive maintenance. Businesses need to better understand what features and services customers are willing to pay for in order to design effective predictive maintenance programs that meet customer needs.

- **Integration with existing IT systems:** Predictive maintenance systems require integration with existing IT systems in order to collect and analyze data from vehicles. However, there is a lack of research on the challenges and opportunities associated with integrating predictive maintenance systems with existing IT systems, particularly in the context of legacy systems that may not be compatible with modern predictive maintenance approaches.
- **Real world data:** Many studies in the field of predictive maintenance use simulated or laboratory data to evaluate their methods. However, there is a lack of research that evaluates the performance of predictive maintenance methods using real-world data from the automotive industry and evaluates the model performance with customer feedback.
- **Data privacy and security:** Predictive maintenance programs rely on the collection and analysis of large amounts of data from vehicles, which raises concerns about data privacy and security. There is a need for research on how to design predictive maintenance systems that protect customer data while still providing accurate and effective predictive maintenance services.

In contrast to the named reviews, we (a) focus specifically on ML condition-based damage prediction for vehicle clutches on real-world data, (b) tackle class imbalance problems in the field as it is often the case, and (c) as a key contribution, identify open challenges and research directions in the field. To the best of our knowledge, there is no current research on condition-based damage prediction for vehicle clutches and tackling class imbalance in predictive maintenance.

Chapter 3

Proposed Methodology

3.1 Proposed Methodology

As argued by (Marchetti et al. 2016), the use of software has become ubiquitous in modern vehicles, with nearly all activities, including critical safety functions like steering, braking, traction, and cruise control, being controlled by software. This reliance on software is evident in city cars, which typically have around 80 Electronic Control Units (ECUs) that communicate with one another via a Controller Area Network (CAN) (Steve et al. 2002). The Controller Area Network (CAN) is a standard protocol used in the automotive industry to collect and transmit data from various sensors and devices within a vehicle. CAN is a message-based protocol, which means that data is transmitted between different nodes on the network in the form of messages. These messages contain information about the state of various systems within the vehicle, including engine speed, temperature, and fuel level. CAN operates on a bus topology, which allows all the nodes on the network to receive the same message simultaneously. This enables real-time monitoring and data acquisition, which is essential for predictive maintenance applications in the automotive sector.

This data can also tell us more about the driver's Clutch shifting pattern. And if the data had been collected by having the objective to detect the failure in clutch then we can model the pattern to fit the ML model on it. The idea is to collect real-world data where the clutch had been failed/Healthy conditioned on other features. Such as shifting the clutch, the mileage, retard shifting, the counter retard shifting, etc. We will also collect the labels for each of these features. Finally, the whole process breaks down to the Machine Learning problem.

We will show different kind of Machine Learning models and their performance on automotive statistics data depending on the nature of the data. As is the case with most data in this domain we will also tackle the problem of class imbalance in the data and report our findings on what and how the current class imbalance works on real-world automotive data (Elrahman et al. 2013). Section 3.1.3 talks about how we are going to address the class imbalance problem in automotive data for PdM. Section 3.1.4 throws light on how we can validate the result with ground truth. Within section 3.1.1, we expound upon the foundational principles of this ML paradigm and delineate the various classifications of machine learning, thereby establishing a rudimentary comprehension of the subject matter.

3.1.1 Machine Learning

Machine learning is a vast field that encompasses numerous disciplines such as information technology, statistics, probability, artificial intelligence, psychology, and neurobiology. By creating a model that accurately represents a chosen dataset, machine learning can effectively solve problems. It has evolved from an initial focus on teaching computers to mimic human brain functions, resulting in a broad discipline that generates fundamental statistical computational theories for the learning processes (Nasteski et al. 2017).

Machine learning is all about creating algorithms that allow the computer to learn. Learning is a way of finding statistical regularities or irregularities in data. As argued by (Muhammad et al. 2015) the primary aim of machine learning is to enable computers to utilize data or past experiences to resolve a given problem. Numerous successful applications of machine learning exist, such as email classifiers that can distinguish between

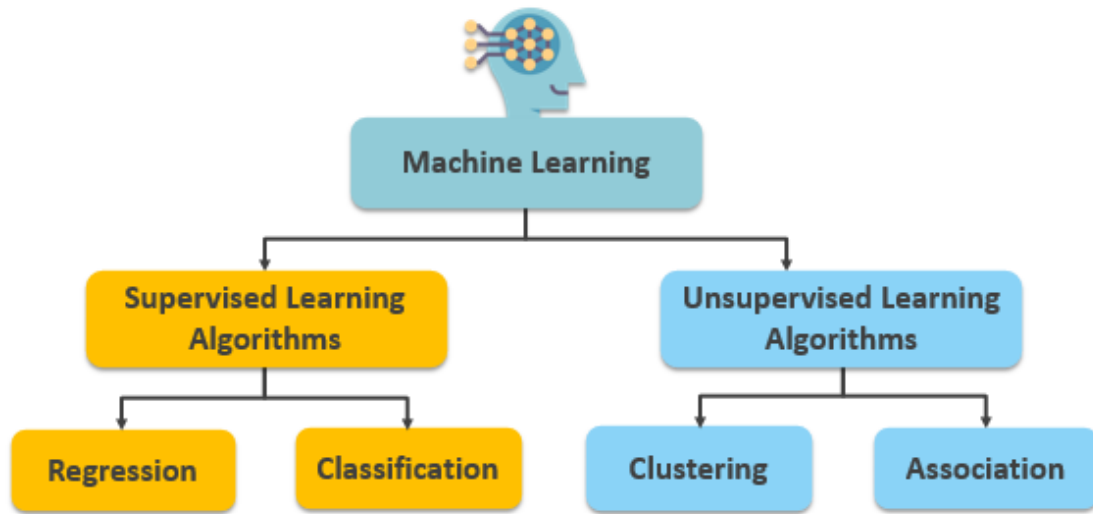


Figure 14: Basic types of Machine Learning (Shiksha 2022)

spam and non-spam messages, sales data analysis systems that can predict customer buying behavior and fraud detection mechanisms. Machine learning can also be implemented for association analysis using supervised, unsupervised, and reinforcement learning techniques (Lindholm et al. 2019).

3.1.1.1 Supervised Machine Learning

In machine learning, supervised learning refers to a type of algorithm that generates a function capable of mapping inputs to desired outputs (Burkart et al. 2021). A common example of supervised learning is the classification problem, where the algorithm is trained to approximate the behavior of a function that assigns a vector to one of several pre-defined classes. This is achieved by analyzing multiple examples of input-output pairs of the function (Nasteski et al. 2017).

The aim is to map a feature vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ to a target $y \in \mathcal{Y} \subseteq \mathbb{R}$. For this purpose, a set of training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ is used for the learning process of the model. Supervised ML can be divided into the tasks of classification and regression. For classification, the target y is a discrete value often called a label. For instance, if $y \in \{0, 1\}$ or $y \in \{-1, 1\}$ one speaks about binary classification. The task of regression is to predict a continuous target value $y \in \mathbb{R}$. Linear regression problem can be defined as follows:

Given a set $\mathcal{D}^{\text{train}} := \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \subseteq \mathbb{R} \times \mathbb{R}$ called training data, compute the parameters $(\hat{\beta}_0, \hat{\beta}_1)$ of a linear regression function

$$\hat{y}(x) := \hat{\beta}_0 + \hat{\beta}_1 x \quad (3.1)$$

s.t. for a set $\mathcal{D}^{\text{test}} \subseteq \mathbb{R} \times \mathbb{R}$ called test set the test error

$$\text{err}(\hat{y}; \mathcal{D}^{\text{test}}) := \frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{(x,y) \in \mathcal{D}^{\text{test}}} (y - \hat{y}(x))^2 \quad (3.2)$$

is minimal.

Note: $\mathcal{D}^{\text{test}}$ has (i) to be from the same data-generating process and (ii) not to be available during training.

In other words, supervised machine learning involves using a labeled dataset to train a model to make predictions on new, unseen data. This can be represented mathematically as:

$$y = f(x; \beta) \quad (3.3)$$

where y is the predicted output, x is the input data, β are the model parameters, and f is the function that maps the inputs to the outputs. The goal of supervised learning is to find the optimal values of β such that the predicted outputs are as close as possible to the true labels in the training data.

3.1.1.2 Unsupervised Machine Learning

Dubbed as unsupervised learning, these approaches operate autonomously without the need for a designated teacher, unlike supervised learning methods. Since there is no single correct answer to be provided, the algorithms are granted the freedom to explore and unearth the noteworthy patterns and relationships concealed within the data (Ma-hesh et al. 2020). These unsupervised learning algorithms are designed to extract salient features from the input data. Subsequently, when novel data is introduced, the algorithm employs the learned features to classify and assign the data to a particular group. Typically employed in clustering and feature reduction applications, unsupervised learning holds tremendous potential for discovering meaningful insights from complex datasets (Bengio et al. 2012).

K-means represents a straightforward and rudimentary unsupervised learning algorithm that tackles the widely recognized clustering problem. The algorithm's core approach revolves around categorizing a given dataset into a specified number of clusters. At the heart of the K-means algorithm lies the concept of defining k centers, one for each cluster. The critical aspect of this step lies in the positioning of these centers, as different placement strategies can lead to disparate outcomes. The K-means can be defined as follows:

Given a set \mathcal{X} called data space, e.g., $\mathcal{X} := R^M$, a set $X \subseteq \mathcal{X}$ called data, a function

$$D : \bigcup_{X \subseteq \mathcal{X}} \text{Part}(X) \rightarrow R_0^+ \quad (3.4)$$

called distortion measure where $D(P)$ measures how bad a partition $P \in \text{Part}(X)$ for a data set $X \subseteq \mathcal{X}$ is, and a number $K \in N$ of clusters, find a partition $P = \{X_1, X_2, \dots, X_K\} \in \text{Part}_K(X)$ with K clusters with minimal distortion $D(P)$.

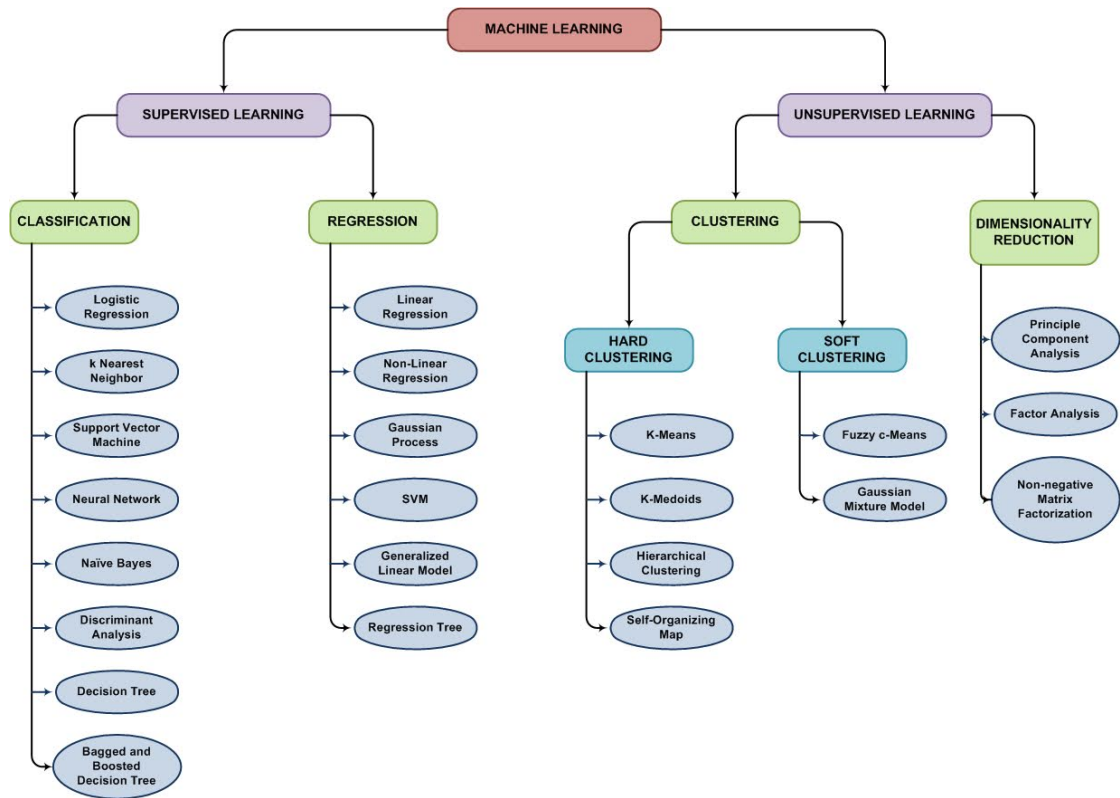


Figure 15: Machine Learning Algorithms (Akshay et al. 2021)

k-means is usually initialized by picking K data points as cluster centers at random:

$$\mu_k := x_n, \quad n := \arg \max_{n \in \{1, \dots, N\}} \sum_{\ell=1}^{k-1} \|x_n - \mu_\ell\|^2, \quad k = 2, \dots, K \quad (3.5)$$

(1) pick the first cluster center μ_1 out of the data points at random and then (2) sequentially select the data point with the largest sum of distances to already chosen cluster centers as next cluster center

To optimize the algorithm's efficacy, it is recommended to place these centers as far apart from each other as feasible. One can envisage that such an approach may prove to be effective in predicting failures from unlabelled data through clustering healthy component in one side and the faulty one on the other.

3.1.2 Types of Machine Learning Predictive Modelling

In recent years there has been a shift towards using machine learning techniques to perform Predictive Maintenance (PdM) tasks involving rolling bearing condition assessment through analysis of their vibration signals. Conventional reliability-centered maintenance strategies have shown their limitations when it comes to identifying faults at an early stage before significant damage occurs. Hence, researchers have focused on developing novel techniques capable of providing accurate detection rates while minimizing false alarms.

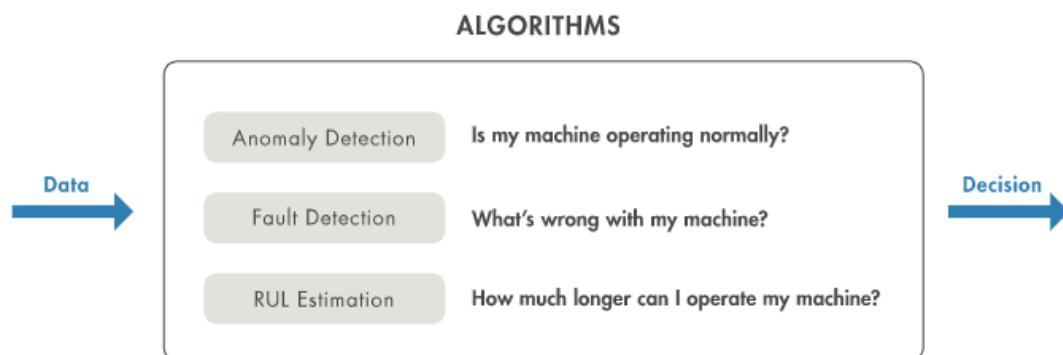


Figure 16: PdM: types of approaches (MathWork 2021)

Despite promising results being reported, many PdM systems still face challenges due to factors such as sensor noise, variability in operating conditions, and the presence of multiple failure modes. Thus, improving system robustness remains a key concern for engineers and scientists involved in the field of predictive maintenance of machines subject to wear and tear (Aburakhia et al. 2022). There are several kinds of Predictive Maintenance approaches when it comes to machine learning (Refer to figure 16). The following section talks about major three approaches in ML for PdM.

3.1.2.1 Similarity Based Models

Data-driven techniques are extensively employed in smart manufacturing to monitor the condition and diagnose faults in rotating machinery. Typically, supervised learning

is used, where a classifier is trained on labeled data to classify the machine's different operational states. Nevertheless, labeled data is frequently inadequate in terms of quantity and quality for many industrial applications, making it unsuitable for training purposes.

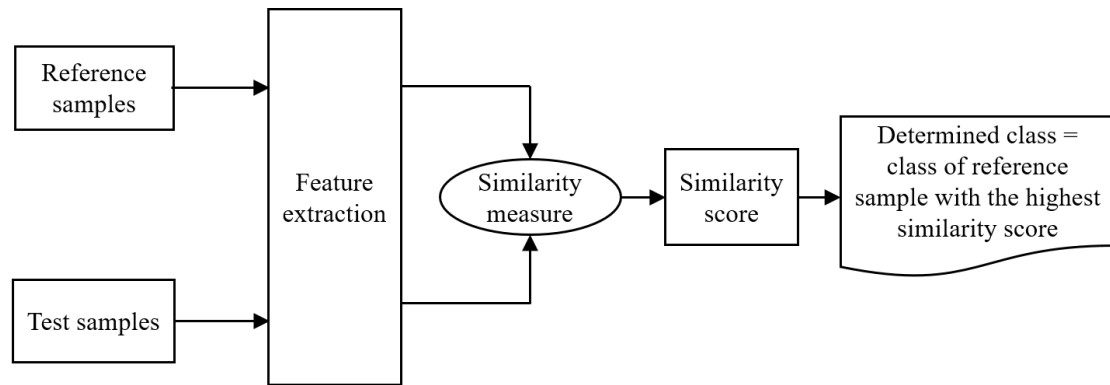


Figure 17: Similarity Based PdM Approach (Aburakhia et al. 2022)

To overcome this challenge, the classification task is reframed as a similarity measure to a reference sample rather than a supervised classification task. Utilizing similarity-based approaches reduces the need for large quantities of labeled data, making it an ideal solution for industrial applications where data is limited. The similarity-based classification framework is depicted in flowchart 17. Discriminative features are extracted from labeled reference and test samples in the initial stage. The similarity between reference and test samples is then computed in the feature space. Ultimately, the classification of various operational states is accomplished by assessing the resultant similarity scores.

3.1.2.2 Survival Based Models

Survival analysis was originally developed for the examination of life tables. Nonetheless, the concept of an 'event' forming the basis of survival analysis is not restricted to biological organisms or human mortality; it encompasses a much broader scope, including occurrences such as student graduation, criminal activity, or divorce. Therefore, the adaptability of survival analysis is significant and accounts for its widespread adoption. Additionally, this versatility is the reason behind the use of various synonyms for survival analysis by various fields, such as event history analysis, duration analysis, or

reliability analysis (Yang et al. 2022).

There are some limitations of such an approach too. Such as censoring. Censoring is a significant issue in the study of survival data or time-to-event data, where an event's time is not observed for various reasons. Unlike other missing data types, censoring is not due to any error or mistake but is an inherent feature of the phenomenon being investigated. For example, patients may withdraw from a clinical trial or service contract customers may terminate the contract before the component failure, leading to missing time-to-event data. A survival function is a statistical measure that defines the probability of a population surviving beyond a particular time point.

$$S(t) = \Pr(T > t) \quad (3.6)$$

It is typically denoted by $S(t)$, where t represents the time of interest. The survival function is a fundamental concept in survival analysis and is commonly used to analyze time-to-event data. The function calculates the probability that a given subject or group of subjects will survive beyond a specific time point. The survival function provides important information for predicting the probability of an event, such as mortality or failure, occurring over a given time period.

where $T > 0$ is a random variable denoting the time of the failure of a component. According to the definition of a cumulative distribution function, a variable T smaller or equal to t can be written as:

$$F(t) = \Pr(T \leq t) \quad (3.7)$$

Since $S(t)$ is a probability, there exists a probability density function f with:

$$S(t) = \int_t^{\infty} f(u)du \quad (3.8)$$

3.1.2.3 Trend Based Models

In contrast to traditional condition-based maintenance (CBM) methods that rely solely on condition monitoring (CM) data from a system or component under surveillance, the sensory-updated degradation-based maintenance (SUDM) policy employs a combination of population-based degradation characteristics and real-time monitoring information to forecast the remaining useful life (RUL). Initially, a generic degradation model is utilized to compute the RUL of a partially degraded system, and an initial maintenance schedule is established based on the preliminary RUL estimates. Subsequently, the RUL estimates are continuously updated in real-time using on-site degradation signals, which are used to adjust the corresponding maintenance actions according to the most recently updated RUL predictions, said (Kaiser et al. 2009).

The foundation of the degradation modeling framework is founded on the premise that the functional expression of a degradation signal is linked to the fundamental physical phenomena that manifest during the degradation process. The functional expression is represented as a stochastic model characterized by continuous-time and continuous-state dynamics. Typically, the magnitude of the degradation signal for the i th component at time t_j is denoted as follows:

$$S(t_{ij}) = \eta(t_{ij}; \Phi_{im}, \Xi_{ik}, B_{il}) + \varepsilon(t_{ij}) \quad (3.9)$$

The degradation signal's trajectory is encapsulated by the functional form $\eta(\cdot)$, while Φ_m is a vector comprising of m deterministic parameters that represent constant degradation attributes shared among all members of the population. The unit-to-unit variability, such as degradation rates, across the population is captured by the vector $\Xi_{ik} = (\theta_{i1}, \dots, \theta_{ik})$, which comprises of k stochastic parameters. In other words, these degradation models estimate RUL by predicting when the condition indicator will cross the set threshold.

Selecting the appropriate model is contingent upon the nature of an organization's data and the associated attributes. For instance, the identification of the remaining useful life (RUL) of a component or system is contingent upon the presence of timestamp information within the dataset. If the dataset lacks such temporal data, it is not feasible

to infer the RUL of the subject entity.

3.1.3 Tackling Class Imbalance

The continuous surge in data volume across various real-time applications has led to disparate distributions within datasets. Dataset disparity arises when one class contains a significantly higher number of specimens compared to another class. This condition is commonly observed in datasets where the major class denotes specimens as negative, having a greater number of instances, while the minor class represents positive specimens, featuring a lesser count (Shuo et al. 2021). The class's imbalance is characterized by the dominance of majority class specimens over minority class specimens, with class ratios such as 100:1 or 1000:1, and so forth. Datasets comprising only two classes are referred to as binary class, while datasets encompassing more than two classes are termed multi-class. Both binary and multi-class datasets encounter challenges related to imbalanced data, where the skewed distribution poses critical issues for various analytical tasks (Elrahman et al. 2013; Longadge et al. 2013).

In such scenarios, the presence of majority classes often results in classifier bias towards those classes, leading to suboptimal performance in classifying minority classes. Consequently, the classifier predominantly identifies instances as belonging to the majority class, effectively disregarding the minority class. Addressing the challenges arising from class imbalance has been a subject of considerable investigation in the literature (Ali et al. 2019).

In the context of external or data-level processing, pre-classification resampling techniques are employed to address data imbalances. One common strategy involves performing resampling to balance the dataset externally. For instance, in the case of imbalanced data, specimens from the majority class may be randomly removed, while specimens from the minority class are augmented by generating artificial instances, effectively adjusting the class ratio. Alternatively, in the ideal scenario, no specimen is added or removed; instead, informed decisions are made about which specimens to create or eliminate (Soltanzadeh et al. 2021).

As argued by (Ali et al. 2019) an algorithmic approach can be adopted to address the imbalance issue, wherein the learner is explicitly instructed not to favor the majority class, thereby mitigating the overall cost of misclassification. Cost-sensitive methods take into account various types of costs, with particular emphasis on minimizing misclassification costs, aiming to achieve an unbiased classifier while optimizing the total cost incurred. By implementing these strategies, researchers endeavor to create classifiers that effectively manage imbalanced datasets, thus enhancing the accuracy and reliability of classification outcomes while considering the broader implications of misclassification costs.

3.1.3.1 Sampling Approaches for Class Imbalance

The fundamental rationale behind employing sampling approaches lies in the recognition that imbalanced distributions within the training sample can introduce bias into learning systems, leading to solutions that do not align with the user's preference goal. This issue arises because the objective is to achieve predictive accuracy on data that is underrepresented in the sample. Conventional learning systems typically search for models that optimize specific criteria, often related to average performance metrics. However, these metrics tend to reflect the performance on the most common cases, which may not align with the user's objectives (Ali et al. 2019; Shuo et al. 2021). In this context, sampling approaches are devised to alter the data distribution within the training sample, redirecting the focus of the learners towards cases that are of utmost interest to the user. The goal of such approaches is to balance the distribution of the least represented (yet more critical) cases with the more prevalent observations, thus steering the learning process towards capturing the most important patterns and insights. By adjusting the sample distribution, sampling approaches aim to bridge the gap between user preference and model optimization, enhancing the learning system's ability to address the user's specific objectives.

3.1.3.2 Under-Sampling Common Classes

The core concept of under-sampling is to effectively reduce the number of observations associated with the most prevalent target variable values, with the specific objective of achieving a more balanced ratio between these common observations and those with less frequent but more significant target values.

Algorithm 1 SMOTE for Under-sampling

Require: Training set D with majority and minority class samples, k (number of nearest neighbors to consider)

Ensure: Under-sampled training set D'

```
1:  $D' \leftarrow \emptyset$ 
2: for each majority class sample  $x$  in  $D$  do
3:    $N \leftarrow \text{GetNearestNeighbors}(x, D, k)$ 
4:    $n \leftarrow$  randomly choose one nearest neighbor from  $N$ 
5:    $diff \leftarrow n - x$ 
6:    $rand \leftarrow$  random number between 0 and 1
7:    $under\_sampled \leftarrow x + rand \times diff$ 
8:    $D' \leftarrow D' \cup \{under\_sampled\}$ 
9: end for
10: for each minority class sample  $x$  in  $D$  do
11:    $D' \leftarrow D' \cup \{x\}$ 
12: end for
13: function SMOTE FOR UNDER SAMPLING( $x, D, k$ )
14:   for each sample  $s$  in  $D$  do
15:     Calculate distance  $dist$  between  $x$  and  $s$ 
16:      $s.dist \leftarrow dist$ 
17:   end for
18:   Sort  $D$  in ascending order of distances
19:   return first  $k$  samples in  $D$ 
20: end function
21: return  $D'$ 
```

In the context of classification, this process involves randomly obtaining a sample from the training cases containing the frequent (yet less interesting) class values (Fernández et al. 2018). This sample is then combined with the observations featuring the rare target class value, culminating in the formation of the final training set utilized by the selected learning algorithm. Consequently, the resulting training sample de-

rived from this under-sampling approach will be smaller than the original (imbalanced) dataset, offering a more equitable representation of both common and important observations for enhanced learning system performance (Fernández et al. 2018; Soltanzadeh et al. 2021).

3.1.3.3 Over-Sampling Minority Classes

In the realm of addressing class imbalance problems, over-sampling emerges as a significant strategy aimed at mitigating biases and improving the performance of machine learning models. The fundamental concept of over-sampling revolves around increasing the instances of the minority class, which is relatively underrepresented in the original dataset. By synthetically augmenting the number of minority class observations, the data distribution becomes more balanced, thereby enabling the learning algorithm to make better-informed decisions and achieve higher accuracy on the minority class (Mohammed et al. 2020).

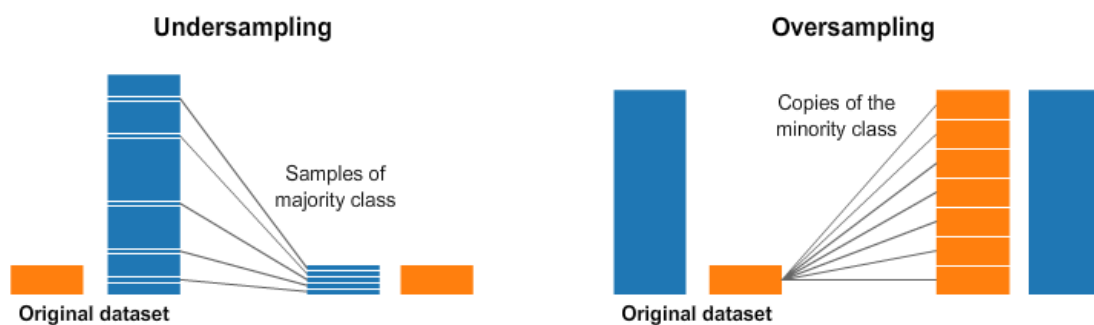


Figure 18: Over and Under Sampling

Several techniques are employed in over-sampling, such as Random Over-sampling and Synthetic Minority Over-sampling Technique (SMOTE). Random Over-sampling involves duplicating existing minority class instances, while SMOTE generates synthetic instances by interpolating between neighboring minority class observations. These techniques effectively enhance the representation of the minority class in the training data, allowing the model to better capture the underlying patterns and reduce the risk of the classifier being biased towards the majority class (Gosain et al. 2017).

Algorithm 2 SMOTE for Over-sampling

Require: Training set D with minority class samples, k (number of nearest neighbors to consider)

Ensure: Synthetic samples S

```
1:  $S \leftarrow \emptyset$ 
2: for each minority class sample  $x$  in  $D$  do
3:    $N \leftarrow \text{GetNearestNeighbors}(x, D, k)$ 
4:   for each nearest neighbor  $n$  in  $N$  do
5:      $diff \leftarrow n - x$ 
6:      $rand \leftarrow$  random number between 0 and 1
7:      $synthetic \leftarrow x + rand \times diff$ 
8:      $S \leftarrow S \cup \{synthetic\}$ 
9:   end for
10: end for
11: function SMOTE FOR OVER SAMPLING( $x, D, k$ )
12:   for each sample  $s$  in  $D$  do
13:     Calculate distance  $dist$  between  $x$  and  $s$ 
14:      $s.dist \leftarrow dist$ 
15:   end for
16:   Sort  $D$  in ascending order of distances
17:   return first  $k$  samples in  $D$ 
18: end function
19: return  $S$ 
```

However, it is crucial to exercise caution when applying over-sampling techniques to avoid potential overfitting issues, as synthetic observations may introduce noise and hinder the model's generalization ability. Careful validation and evaluation procedures are essential to ensure that the model's performance accurately reflects its predictive capabilities on unseen data (Elrahman et al. 2013; Soltanzadeh et al. 2021; Shuo et al. 2021).

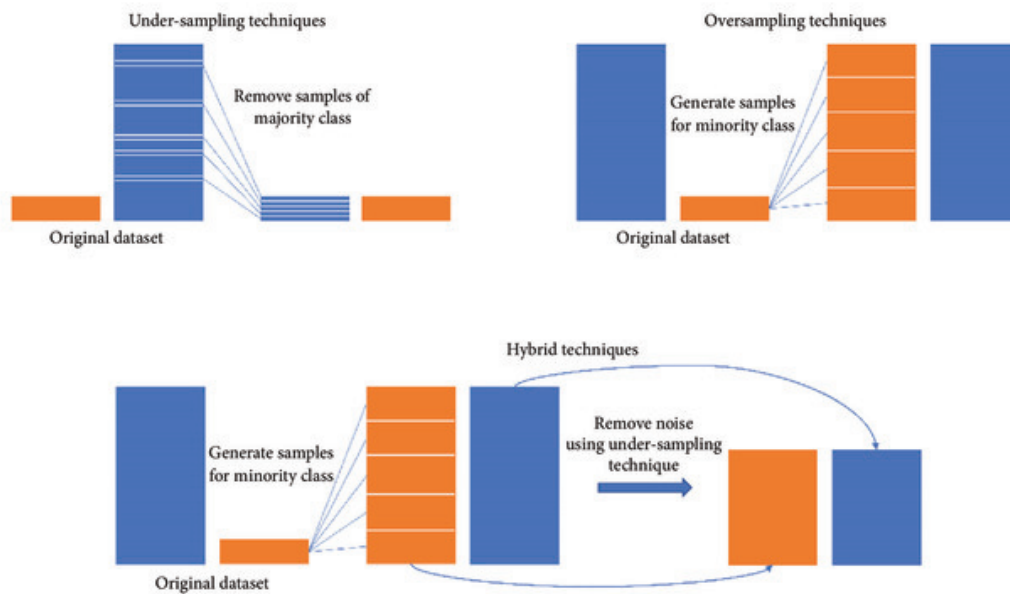


Figure 19: Hybrid Sampling Approach

In conclusion, over-sampling represents a valuable tool in combatting class imbalance problems. By artificially boosting the representation of the minority class, over-sampling empowers learning algorithms to discern and classify minority class instances more effectively, ultimately leading to improved model performance and more equitable data distribution. The appropriate utilization of over-sampling techniques, coupled with rigorous validation and assessment protocols, can significantly enhance the reliability and robustness of machine learning models in addressing class imbalance challenges.

3.1.4 Validating the PdM Results

Validating the results of predictive maintenance in the automotive sector is a critical aspect of ensuring the accuracy and reliability of the predictive models. In this research paper, we adopt several techniques for validation, including data collection, establishing ground truth, cross-validation, and real-world testing. Collaboration with Original Equipment Manufacturers (OEMs) plays a crucial role in enhancing the validation process. By collaborating with OEMs, we gain access to proprietary data sources, encompassing diverse vehicle models and driving scenarios, which enriches the validation dataset. Additionally, OEMs provide historical maintenance records and performance data that serve as reliable ground truth, enabling us to compare the predictive maintenance model's results with actual maintenance events.

3.1.4.1 Data Collection and Preprocessing

The first step in validating predictive maintenance results involves the meticulous collection and preprocessing of data. For this purpose, collaboration with Original Equipment Manufacturers (OEMs) proves to be highly beneficial. OEMs possess extensive data repositories, containing sensor readings, historical maintenance records, performance metrics, and other relevant information from a wide range of vehicle models and driving scenarios. Access to this diverse and extensive dataset ensures that our validation process is representative of real-world conditions. Additionally, OEM collaboration allows us to address potential data quality issues and ensures that the data used for validation is accurate and reliable, enhancing the overall validity of our research.

3.1.4.2 Establishing Ground Truth

Validation of predictive maintenance models necessitates the establishment of a ground truth against which the model's predictions can be assessed. OEM collaboration plays a pivotal role in this aspect by providing historical maintenance records, real-world failure events, and component performance metrics. This ground truth, derived from OEM-provided data, serves as a reliable reference point to validate the accuracy of the model's predictions. By comparing the model's results with actual maintenance events, we can determine the predictive maintenance model's effectiveness and its abil-

ity to anticipate and prevent potential failures.

3.1.4.3 Performance Metrics Selection

Selecting appropriate performance metrics is crucial in gauging the predictive maintenance model's efficacy. Commonly used metrics include precision, recall, F1-score, and receiver operating characteristic (ROC) curve analysis. However, domain-specific metrics that align with automotive maintenance objectives are equally important for accurate assessment. Through collaboration with OEMs, we gain insights into the specific requirements and priorities of the automotive industry, enabling us to select relevant performance metrics that accurately reflect the model's impact on real-world maintenance tasks.

Chapter 4

Experiments

4.1 Data Collection

The automotive industry has witnessed a remarkable evolution in data collection methodologies, particularly through the integration of Controller Area Network (CAN) and Electronic Control Units (ECUs). Among the diverse range of data points, the collection of vehicle mileage, along with a suite of critical features including converter clutch shifting frequency, retard occurrence rate, A clutch shifting instances, shifting time for a clutch, and the presence of clutch failures, holds paramount importance for various operational and analytical purposes. These features collectively offer insights into vehicle performance, driver behavior, and component health, contributing to enhanced decision-making and driving experience optimization (Francis et al. 2022). CAN, the standardized communication network interlinking the vehicle's ECUs, emerges as a central conduit for transmitting these multifaceted data across the vehicle's electronic systems. Dedicated ECUs, tailored for each feature, work cohesively to ensure accurate, real-time data collection. By leveraging the seamless communication capabilities of the CAN bus, data is relayed efficiently, enabling real-time updates and facilitating integration with telematics systems and diagnostic tools (Giobergia et al. 2018).

Furthermore, the integration of CAN and ECUs transcends the domain of individual features, capturing a spectrum of operational data that enriches the vehicle's dig-

ital ecosystem. Beyond the core features, ECUs collect and interpret data on engine performance, braking dynamics, transmission behavior, and more. This comprehensive approach fosters a holistic understanding of vehicle behavior, translating into informed decision-making and proactive maintenance strategies. The interconnectedness of CAN and ECUs also empowers drivers with real-time insights (Turner et al. 2020) and (Giobergia et al. 2018). Certain ECUs, such as those governing the instrument cluster, not only facilitate feature-specific data collection but also visualize crucial vehicle parameters, enhancing the driver's situational awareness. Concurrently, diagnostic ECUs identify potential issues promptly, ensuring minimal disruptions to vehicle operation. The coalescence of CAN and ECUs, therefore, not only facilitates robust data collection but also forms the bedrock for data-driven advancements that encompass vehicle performance, safety, and driver engagement (Maksymova et al. 2018).

4.2 Types of ECU's

In the realm of automotive engineering, Electronic Control Units (ECUs) stand as pivotal elements orchestrating the intricate symphony of vehicular functions. These specialized microcontrollers are meticulously tailored to oversee distinct aspects of a vehicle's operation, shaping its performance, safety, and efficiency. Among the varied categories of ECUs, several play indispensable roles in shaping the behavior of vehicle clutches and their associated activities. The Engine Control Unit (ECU) takes center stage, wielding authority over parameters such as fuel injection, ignition timing, and exhaust emissions to optimize the operation of the powertrain, directly influencing clutch engagement and disengagement. Complementing this, the Transmission Control Unit (TCU) emerges as a crucial contender, directing gear shifts and torque converter lockup, thus intricately influencing clutch interactions within automatic and automated manual transmissions. Additionally, the Anti-lock Braking System (ABS) ECU, with its capacity to prevent wheel lock during braking maneuvers, indirectly influences clutch engagement during deceleration, enhancing stability. Moreover, the Electronic Stability Control (ESC) ECU acts as a sentinel against skidding, a scenario where clutch activity might be pivotal in stabilizing the vehicle.

4.3 Data Overview

We have collected various information related to the clutches of the vehicle using ECU and CAN signals. The data has the following attributes:

Mileage column represents the distance covered by the vehicle (in thousands). It can be a crucial feature for predicting failures, as wear and tear on components, including the clutch, can be influenced by the total mileage. **Converter Clutch Shifting per km** column refers to the frequency of shifting the converter clutch per kilometer. A converter clutch is used in automatic transmissions to improve fuel efficiency and control slipping. Higher values here might indicate more frequent shifting. **Counter of Retard per km** refers to engine braking in this context. This column represents how often the engine braking is used per kilometer. Engine braking can affect components like the clutch as well. **Counter Shifting A Clutch per km** column represents the frequency of shifting the clutch per kilometer. Similar to the previous shifting feature, it could affect wear and tear. **Time in Shifting for A Clutch** column indicates the time taken for shifting the clutch. It's essential for modeling, as longer shift times could indicate potential issues. **Failure of the Clutch** is the target variable for our machine learning model. It is a binary indicator (0 or 1) representing whether a clutch failure occurred or not given the above driving situation. This column was collected and provided as a ground truth.

	serialnummer	mileage	converter_clutch_shifting_per_km	counter_of_retard_per_km	counter_shifting_A_clutch_per_km	time_in_shifting_for_A_clutch	failure_of_the_clutch
0	390791	330.0	2.744701	5.121938	2.142675	8115.702	0
1	419637	101.0	3.081924	6.004832	5.609128	2052.291	0
2	402529	0.0	0.000000	0.000000	0.000000	0.000	0
3	298752	56.0	4.190101	6.693927	5.978263	1571.740	0
4	456176	42.0	1.178854	2.852828	2.923559	0.807	0
5	4366	237.0	4.944449	6.953861	6.359992	6416.444	0
6	390140	278.0	3.919283	6.767968	2.621393	10866.138	0
7	430198	24.0	1.561850	3.590662	2.609622	514.007	0
8	372033	24.0	2.877069	6.982021	5.058199	525.833	0
9	369323	390.0	2.229392	2.962408	1.797570	10012.517	0

Figure 20: Top 10 rows of the data set

The data above shows all the independent features and the last column is our dependent target which we aim to predict. The dependent variable is the primary focus of

our research, and it represents the outcome or response that we are trying to understand, explain, or predict (Reddy et al. 2018). It is the variable that we observe, measure, or analyze to assess the impact of other variables, particularly the independent variable(s).

serialnummer	mileage	converter_clutch_shifting_per_km	counter_of_retard_per_km	counter_shifting_A_clutch_per_km	time_in_shifting_for_A_clutch	failure_of_the_clutch	
1260	348864	211.0	4.678709	5.160544	4.320290	6198.730	1
1261	348899	115.0	3.556275	5.241753	3.166353	2640.000	1
1262	359565	250.0	3.419256	2.411005	4.300579	5541.823	1
1263	360344	222.0	3.926419	3.052833	4.310528	5530.771	1
1264	360361	198.0	3.956091	0.250412	4.153114	4734.406	1
1265	390788	50.0	2.072324	3.445488	1.455295	1122.821	1
1266	401212	42.0	1.948765	3.732642	1.702488	546.596	1
1267	405859	115.0	1.053424	1.068636	1.449403	670.598	1
1268	412612	76.0	0.998201	1.907533	1.453226	512.174	1
1269	425464	153.0	2.899547	6.000143	4.974188	2810.281	1

Figure 21: Bottom 10 rows of the data set

In simpler terms, the dependent variable is what we are trying to explain or understand better through our research. It's the variable that may change in response to variations in other factors. For example, in a study investigating the effect of study time on exam scores, the exam scores are the dependent variable because they depend on how much time students spend studying. The independent variable, on the other hand, is the factor or condition that you manipulate, control, or analyze to observe its effect on the dependent variable (Rajbahadur et al. 2019). It represents the cause or the variable that you believe has an impact on the dependent variable. Independent variables can be categorical (e.g., gender, treatment groups) or continuous (e.g., time, temperature).

Table 1: Dependent and Independent Variables

Dependent Variable	Independent Variables
failure_of_the_clutch	mileage (in thousands) converter_clutch_shifting_per_km counter_of_retard_per_km counter_shifting_A_clutch_per_km time_in_shifting_for_A_clutch

For our use case, table 1 shows which are independent and which is our dependent variable. In Machine Learning terms one can say that the independent variables are the X and the dependent variable is our y which we are going to predict.

Table 2: Data Set Information

Index	Column	Count	Data Type
1	mileage (in thousands)	1270 non-null	float64
2	converter_clutch_shifting_per_km	1269 non-null	float64
3	counter_of_retard_per_km	1269 non-null	float64
4	counter_shifting_A_clutch_per_km	1269 non-null	float64
5	time_in_shifting_for_A_clutch	1270 non-null	float64
6	failure_of_the_clutch	1270 non-null	int64

Table 2 shows the columns' name and their count in the data as well as the data types. One can see that we do have some missing data points and in total, we have only 1270 data points.

4.4 Class Distribution

Checking the class distribution in machine learning (ML) data is a fundamental and crucial step in the data preprocessing phase. It involves examining how the data is distributed among different classes or categories within a dataset. The foremost reason for checking class distribution is to identify whether there is a class imbalance in the dataset. Class imbalance occurs when one class significantly outnumbers the others. It's essential to be aware of this issue because it can profoundly impact model performance. Class imbalance can lead to model bias, where the algorithm tends to favor the majority class. In scenarios with imbalanced data, the model may predict the majority class most of the time to achieve high accuracy, neglecting the minority class. This can result in misleadingly high overall accuracy but poor performance on the minority class.

Understanding the class distribution is crucial for assessing the real-world relevance of the ML problem. In many applications, like fraud detection, disease diagnosis, or rare event prediction, the minority class holds more significant importance. Neglecting the minority class can have severe consequences. That is exactly the case with the damage calculation in Clutches. The minority class holds more importance.

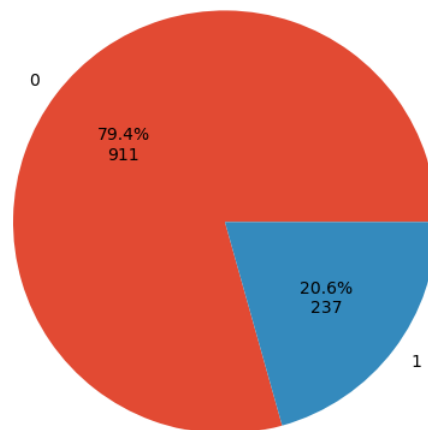


Figure 22: Class Distribution of the Target Variable

The figure 22 showed the distribution of the classes in the data set. Here, we can see that only 237 data points belong to class 1 (failure happened) and 911 data points are coming from class 0 (failure didn't happen). Since the data is skewed towards the class 0 we can safely say that there is a class imbalance in the data. Imbalanced data can lead to poor model generalization. Models trained on imbalanced data-sets may over-fit to the majority class and perform poorly on unseen data that follows a more balanced distribution. Checking and addressing class imbalance can improve model generalization. Moreover, in certain applications, misclassifying instances from the minority class can be more costly than misclassifications from the majority class. For example, in medical diagnosis, failing to detect a rare disease can be life-threatening, making false negatives more costly than false positives. Knowing the class distribution helps in deciding which data preprocessing techniques to apply. Depending on the imbalance level, techniques like oversampling, under-sampling, synthetic data generation (e.g., SMOTE), or cost-sensitive learning may be employed to balance the data (Fernández et al. 2018; Soltanzadeh et al. 2021). Later in this paper we will discuss and tackle this challenge as well.

4.5 Analysis of Independent Variable

The distribution of each feature column in a machine learning (ML) data-set is crucial for several reasons, and analyzing these distributions can provide valuable insights for ML projects. Examining feature distributions helps you gain a deeper understanding of the data you're working with. It allows you to identify the range, spread, central tendency, and other statistical properties of each feature. Detecting anomalies or outliers in feature distributions can be a sign of data quality issues. Outliers may indicate errors in data collection, measurement, or entry, which need to be addressed before building ML models. One such method is to visualize the Histogram of the data. A histogram is a graphical representation of the distribution of data in a data-set. It provides a visual summary of the frequency or count of values within specified bins or intervals. Histograms are widely used in statistics and data analysis to understand the underlying patterns and characteristics of a data-set.

4.5.1 Distribution of Mileage

The x-axis of a histogram represents the range of values in our data-set, divided into discrete bins or intervals. Each bin represents a specific range of values, and data points are grouped into these bins based on their values. The y-axis of a histogram represents the frequency or count of data points falling into each bin. It represents the probability density if the histogram is normalized. In a frequency histogram, the y-axis shows how many data points are in each bin. In a probability density histogram, it shows the probability of a data point falling into each bin.

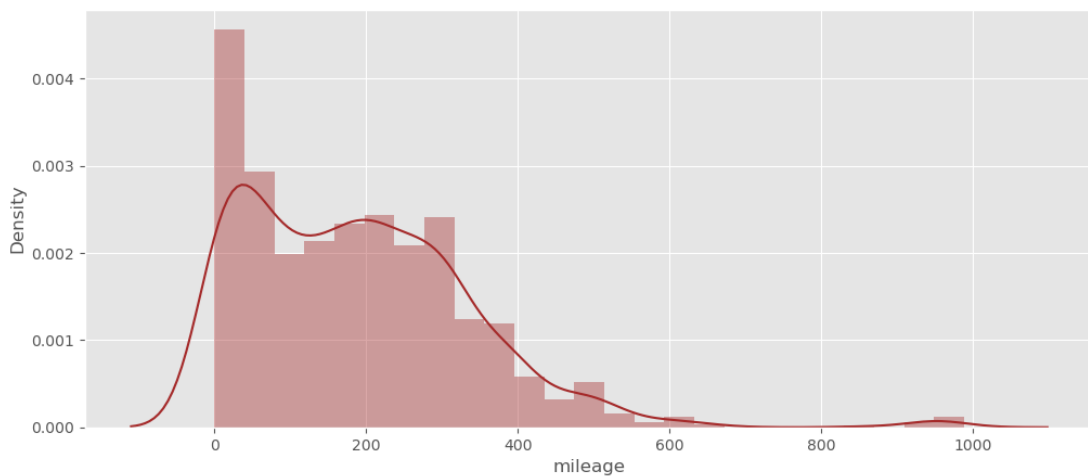


Figure 23: Distribution of Mileage

Mileage stands as a paramount consideration for vehicle owners, often taking precedence even before the initial purchase of a car or motorcycle. In essence, mileage encapsulates the pivotal metric denoting the distance a vehicle can traverse on a single litre of fuel, be it petrol or diesel. The nomenclature for this metric assumes various forms, including 'bike average,' 'bike mileage,' 'car average,' 'gas mileage,' and a plethora of alternatives. Nevertheless, irrespective of the nomenclature employed, the fundamental concept remains unaltered—a quantification of a vehicle's fuel efficiency.

For us, the Mileage is a bit different here. The Mileage column indicates the total distance covered by the vehicle. We can see in the figure 23 that the data follows Skewed Distribution. In other words, when data is not symmetric and has a longer tail on one

side. It can be either positively skewed (long tail to the right) or negatively skewed (long tail to the left).

4.5.2 Mileage and Clutch Failure Relation

The relationship between the total distance covered by a vehicle and the occurrence of clutch failures is an intriguing subject of inquiry within the automotive domain. Clutches are vital components in vehicles, facilitating gear transitions and power transmission. Over time, clutches experience wear and tear due to engagement and disengagement cycles. It is hypothesized that there may exist a correlation between higher cumulative distances traveled and an elevated likelihood of clutch failure. As vehicles accrue greater mileage, the clutch components endure extended usage, potentially leading to increased wear. This wear, in turn, could influence the clutch's performance and longevity.

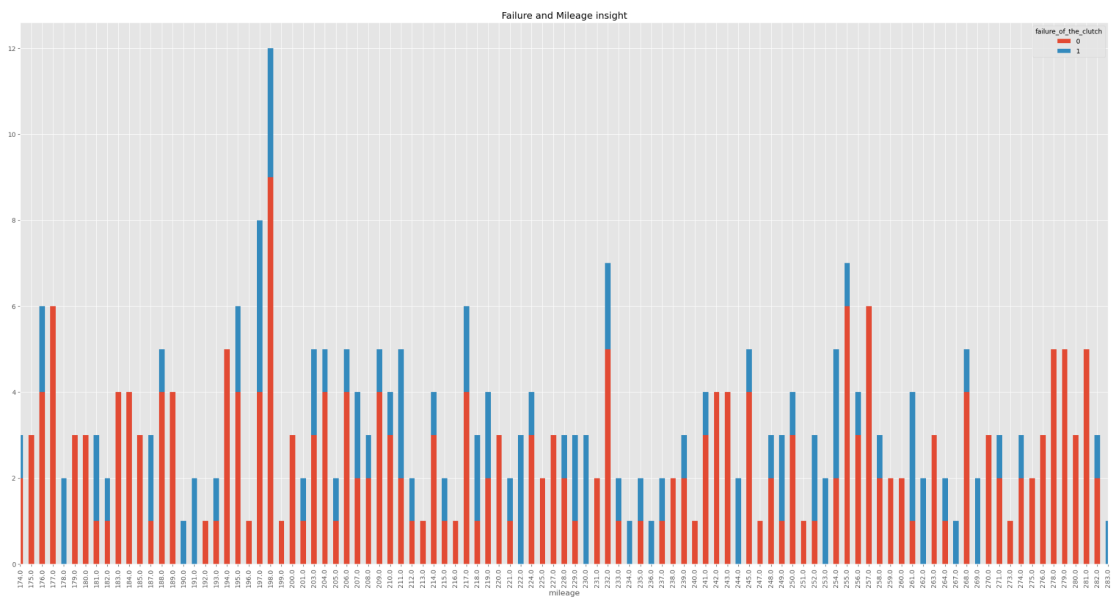


Figure 24: Clutch Failure and Mileage Correlation

As depicted in the figure 24 we can see that it's very hard to say for sure if there exists any definite relationship between the total Mileage covered by the vehicle and its Clutch Failure. In the plot 24 we can see that there exist two keys 0 and 1 where red bars are for the moment where failure didn't happen for a particular mileage. Mileage

can be seen on the x axis. Remember the MILEAGE IS IN THOUSANDS. Important to note is that the plot 24 is only part of a big visualization. However, it gives us an insight that just by looking at the stacked bar chart it's not possible to say that clutch failure indeed is a reflection of vehicle total mileage. There can be other factors at play. The upcoming sections cover a deep analysis of the other features.

4.5.3 Converter Clutch Shifting

The *Converter Clutch Shifting per km* column in our data-set pertains to the frequency at which the converter clutch within a vehicle's transmission system is engaged and disengaged per kilometer traveled. In an automatic transmission, the converter clutch plays a pivotal role in controlling power transfer from the engine to the transmission and, subsequently, to the wheels. The converter clutch can be engaged to optimize fuel efficiency and engine performance. Therefore, the number of times it shifts or toggles its state per kilometer is a crucial metric for evaluating the transmission's behavior. Understanding this parameter allows us to assess how often the clutch is activated during typical driving scenarios, providing valuable insights into transmission performance, and fuel economy, and potentially identifying patterns or anomalies that may impact overall vehicle operation.

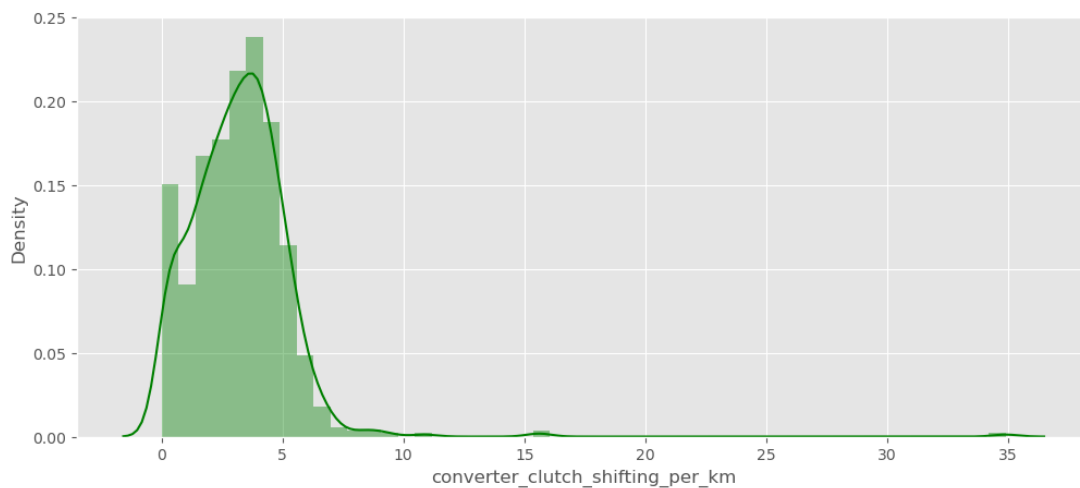


Figure 25: Histogram of Converter Clutch Shifting per km

The figure 25 also shows similar distribution as Mileage 23. It is Skewed Distri-

bution where data is not symmetric and has a longer tail on one side. It can be either positively skewed (long tail to the right) or negatively skewed (long tail to the left). As depicted in plot 25 it is negatively skewed.

4.5.4 Counter of Retard

This parameter represents the number of times the vehicle's transmission system engages in retardation events (e.g., gear shifts or deceleration due to braking) within a distance of one kilometer. In essence, it measures how frequently such events occur during typical driving conditions. Analyzing this data could provide insights into the vehicle's transmission behavior and its relationship to clutch performance. Understanding the relationship between retardation events and clutch behavior can be valuable for predicting and preventing clutch failures. For instance, frequent and abrupt gear shifts or excessive deceleration events may contribute to increased wear and tear on the clutch, potentially leading to premature failures. Therefore, analyzing the "Counter of Retard per km" can be a crucial aspect of studying and predicting clutch failures in the context of this data-set.

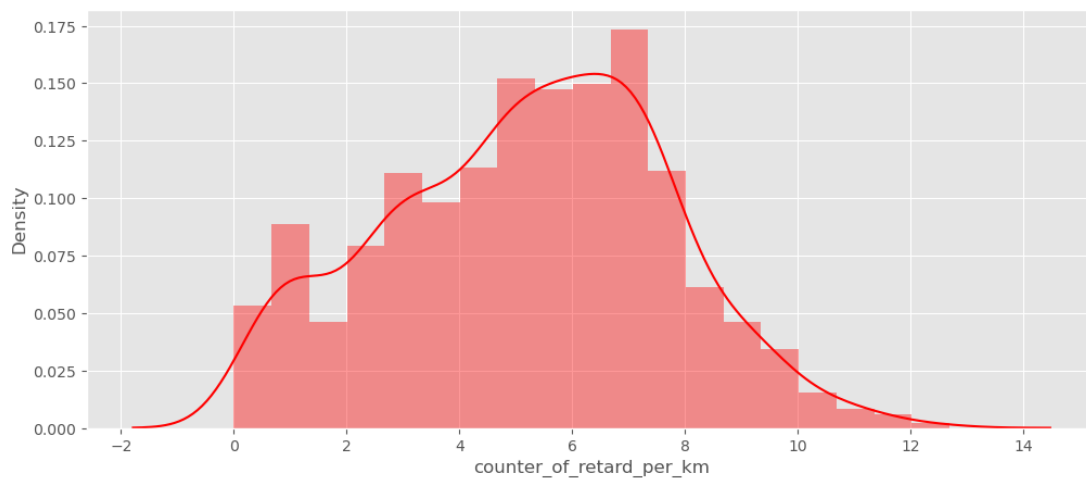


Figure 26: Histogram of Counter of Retard per km

The width of the histogram indicates the spread or dispersion of the data (see figure 26). A wide histogram suggests a greater variability in the data, while a narrow histogram indicates lower variability.

4.5.5 Counter Shifting of Clutch

The "Counter Shifting of a Clutch per km" column within the data-set, specifically collected for the prediction of clutch failures, serves as a significant parameter denoting the frequency of clutch engagement and disengagement events per kilometer traveled in a vehicle. In the intricate mechanics of automotive transmission systems, the clutch plays a pivotal role in the seamless transition between gears, enabling the vehicle to accelerate, decelerate, and maintain various speeds. This column's value indicates how often these clutch-shifting events occur over a standardized distance of one kilometer. The frequency of clutch shifts is a critical metric for understanding the operational dynamics of the vehicle's transmission system. By meticulously tracking the occurrences of clutch engagements and disengagements in relation to the distance traveled, this parameter can unveil essential insights into the clutch's performance, wear patterns, and potential factors contributing to clutch failures. A comprehensive analysis of "Counter Shifting A Clutch per km" offers a valuable perspective on the interplay between clutch behavior and the broader context of vehicle operation, aiding in the prediction and prevention of clutch-related issues, which can be of paramount significance in automotive maintenance and reliability.

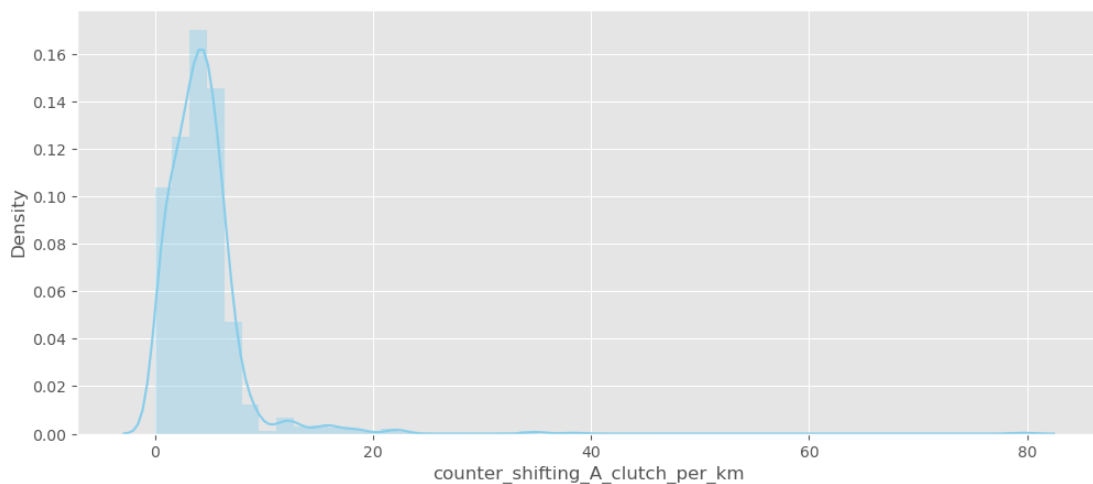


Figure 27: Histogram of Counter Shifting of Clutch

The figure 27 also shows a similar distribution as Mileage 23. It is Skewed Distribution where data is not symmetric and has a longer tail on one side. It can be either

positively skewed (long tail to the right) or negatively skewed (long tail to the left). As depicted in plot 27 it is negatively skewed.

4.5.6 Time in Clutch Shifting

The "Time in Shifting for A Clutch" column within the data-set assumes a pivotal role in the modeling and evaluation of vehicle clutch performance, presenting a crucial metric denoting the duration taken for the clutch-shifting process. In the intricate mechanics of automotive transmission systems, the clutch serves as a linchpin, orchestrating the seamless transition between gears, a process vital for acceleration, deceleration, and maintaining various speeds. This specific parameter encapsulates the temporal aspect of these clutch engagements and disengagements, signifying the time interval required for the shifting process to transpire. Notably, it can be a barometer for assessing the efficiency and efficacy of the clutch operation. Longer shift times may serve as potential indicators of underlying issues within the clutch mechanism, such as wear and tear, misalignments, or sub-optimal performance. Therefore, a comprehensive analysis of the "Time in Shifting for A Clutch" parameter proves indispensable for understanding the intricacies of clutch behavior and pinpointing aberrations or anomalies that could impact vehicle operation. In the context of modeling, it emerges as a valuable feature for predictive maintenance, facilitating the early detection of clutch-related concerns and, subsequently, the preservation of vehicle reliability and performance.

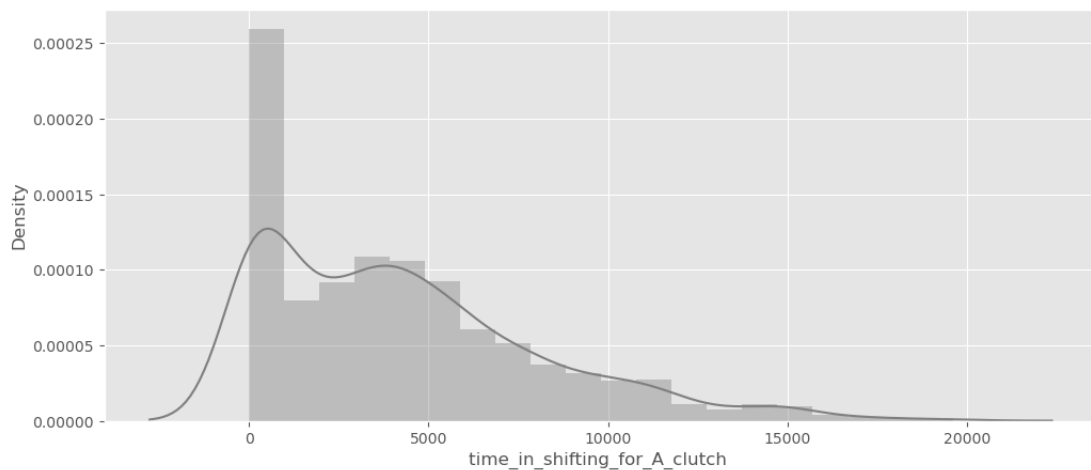


Figure 28: Histogram of Time in Shifting of Clutch

4.5.7 Feature Correlation

Numerous determinants exert influence on the efficacy of machine learning in the context of a specific task. Notably, data quality stands out as a critical factor, wherein the presence of irrelevant, redundant, or noisy information, along with data unreliability, poses formidable challenges to the knowledge acquisition process during training (Cai et al. 2018). Addressing this issue, feature subset selection emerges as a pivotal methodology aimed at judiciously identifying and eliminating superfluous and duplicative data components. It endeavors to enhance the data's informational efficiency by retaining only the most pertinent attributes (Hall et al. 2000). It is noteworthy that machine learning algorithms exhibit variability in their prioritization of feature selection, with some algorithms assigning greater significance to this process than others, underlining its role in optimizing model performance and knowledge extraction (Senan et al. 2021).

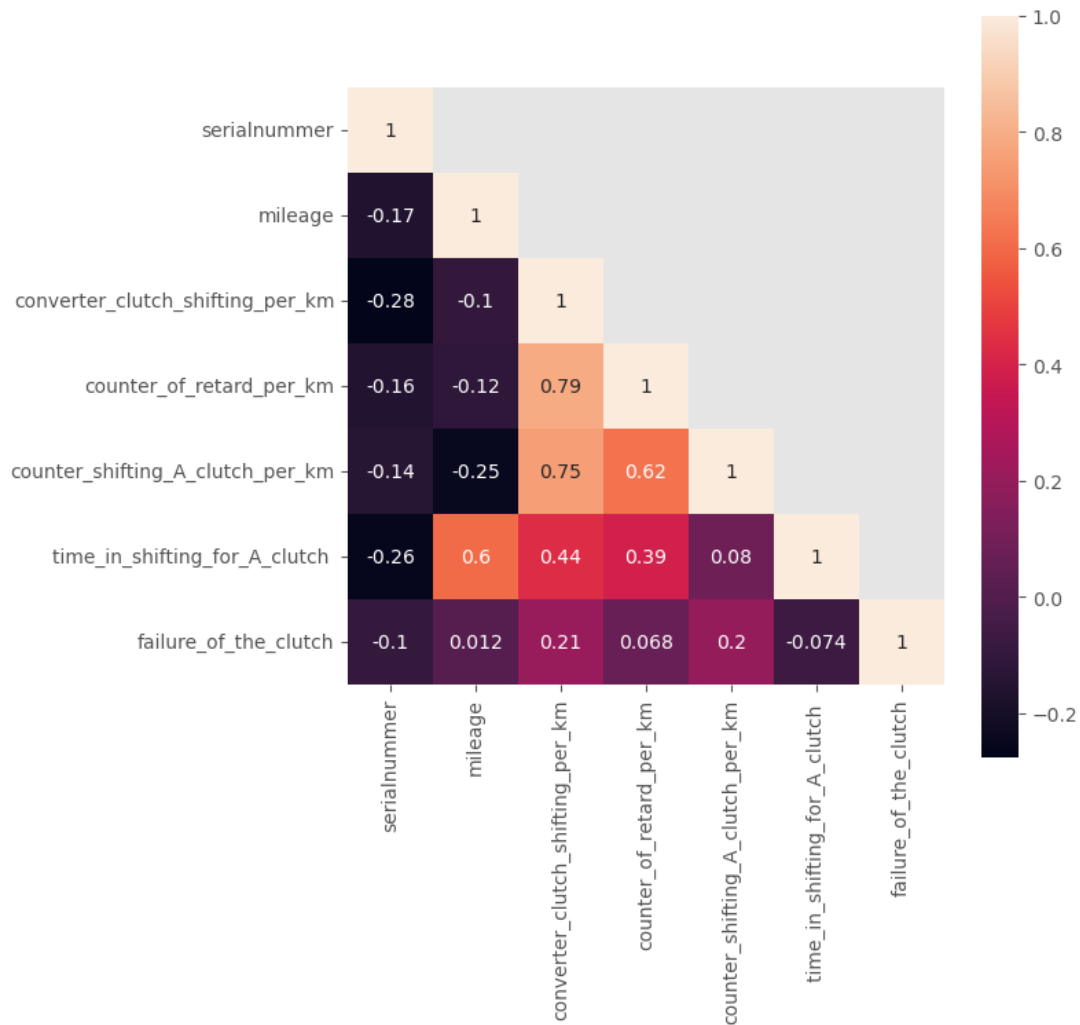


Figure 29: Correlation in Features

The core of the CFS (Correlation-based Feature Selection) algorithm resides in a heuristic designed to systematically assess the value or significance of a subset of features. This heuristic meticulously considers two fundamental aspects: firstly, the utility of individual features in predicting the class label, and secondly, the degree of intercorrelation existing among these features (Williams et al. 2006). The underlying hypothesis guiding this heuristic operation posits that optimal feature subsets exhibit characteristics wherein the constituent features demonstrate strong correlations with the class label while simultaneously maintaining minimal correlations among themselves. This principle underscores the pivotal role of feature selection in enhancing the discriminative

power of machine learning models by selecting feature subsets that strike an ideal balance between class relevance and interfeature independence (Hall et al. 2000; Senan et al. 2021). A correlation plot (see figure 29) in machine learning, often represented as a correlation matrix or heatmap, provides a visual representation of the relationships between variables (features) in a dataset. It is a valuable tool for exploring the degree and direction of linear association between pairs of variables.

”A good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other”. — (Hall et al. 2000)

High correlations between independent features (multicollinearity) can lead to instability in Machine Learning models. A correlation plot can highlight multicollinearity issues, allowing us to address them by selecting one feature from highly correlated pairs or applying dimensionality reduction techniques. Moreover, When building predictive models, knowing the correlations between features can guide us in selecting an appropriate algorithm. For example, linear models assume that features are not highly correlated, while tree-based models can handle correlated features more robustly. If we look at the figure 29, it shows some correlation between the 2 features. However, they are not highly correlated thus, we are keeping all the columns for our modeling.

4.6 Experiment Setup

The ML (Machine Learning) experiment setup is a critical phase in the lifecycle of any machine learning project. It encompasses the systematic arrangement of all essential components, processes, and resources required to conduct an ML experiment effectively. This setup involves defining the problem statement, selecting appropriate data sources, preprocessing and cleaning the data, choosing suitable machine learning algorithms, specifying hyperparameters, splitting the data into training and testing sets, and setting up performance metrics for evaluation. Moreover, it also includes considerations for hardware and software infrastructure, such as selecting the right computing environment and ensuring compatibility with the chosen ML framework. The experiment setup

phase is pivotal in ensuring that the experiment is well-organized, reproducible, and capable of yielding meaningful insights and results, ultimately guiding the development of robust machine learning models. Some of the aforementioned components apply to our use case. Such as data cleaning, choosing the right model, etc.

4.6.1 Data Cleaning

Data cleaning, often referred to as data cleansing or data scrubbing, is an integral and meticulous process within the realm of data preprocessing (Li et al. 2019). It involves the systematic identification, correction, and removal of errors, inconsistencies, inaccuracies, and anomalies present in a dataset. These imperfections can arise from various sources, including data entry errors, sensor inaccuracies, missing values (Hong-hai et al. 2005), duplications, and outliers. The primary objective of data cleaning is to enhance the quality, integrity, and reliability of the data, making it suitable for analysis and modeling. This process typically encompasses tasks such as imputing missing values, correcting typos, handling outliers, and ensuring data consistency. The significance of data cleaning cannot be overstated, as the accuracy and credibility of subsequent data-driven analyses and machine-learning models heavily depend on the cleanliness of the underlying data. It is an indispensable step in the data preparation pipeline, ensuring that insights drawn from data are robust, trustworthy, and actionable (Chu et al. 2016).

4.6.1.1 Missing Value

As shown in the table 2, we have a few missing values. When dealing with missing values, the key is to maintain data integrity while minimizing data loss. Depending on the nature and context of the dataset, we can either remove the affected rows or impute the missing values using straightforward methods. However, only one data point is missing thus, we can easily remove the NaN value row. This is a reasonable strategy when the proportion of missing values is negligible and won't significantly impact your analysis or model training.

4.6.1.2 Data Point Removal

As you can see in the figure 21, there exist rows where Mileage is zero. And when Total Mileage is zero then it means the vehicle didn't move at all. and thus this data point was collected due to an error in the data collection process. We will remove all the rows where Total Mileage is recorded to be zero. There are many reasons to remove such a data point. Such as,

1. Zero values can often skew the distribution of a feature. For some machine learning algorithms, especially those sensitive to feature distributions like linear regression, having a skewed distribution can lead to suboptimal model performance.
2. Zero values can interfere with scaling operations, such as standardization (mean normalization) or min-max scaling. When you have zero values in a feature, it can affect the calculation of mean and standard deviation, potentially distorting the scaled values.
3. Sometimes, zero values are used as placeholders for missing data. In such cases, it's crucial to distinguish between true zero values and missing data. By removing or handling zero values, you can treat missing data more appropriately, either by imputing them or using other techniques.
4. Some machine learning models, especially those based on distances or similarities, can be sensitive to zero values. For example, in clustering or nearest neighbor algorithms, zero values can lead to unpredictable results or distort the notion of similarity.
5. In certain scenarios, zero values might represent noise or irrelevant information. Removing them can simplify the dataset and improve the signal-to-noise ratio, potentially leading to more accurate models.
6. Zero values can affect the interpretability of models. For example, coefficients in linear models may be challenging to interpret when zero values are present.

However, it's important to note that removing zero values should be done thoughtfully and in consideration of the specific problem and dataset. In some cases, zero

values may carry meaningful information, and removing them could result in the loss of important insights. Therefore, the decision to remove or handle zero values should be based on a thorough understanding of the data, the domain, and the goals of the machine learning project. After initial data cleaning our data-set has following properties:

Table 3: Clean Data Set Information

Index	Column	Count	Data Type
1	mileage (in thousands)	1148 non-null	float64
2	converter_clutch_shifting_per_km	1148 non-null	float64
3	counter_of_retard_per_km	1148 non-null	float64
4	counter_shifting_A_clutch_per_km	1148 non-null	float64
5	time_in_shifting_for_A_clutch	1148 non-null	float64
6	failure_of_the_clutch	1148 non-null	int64

We have 1148 data points for each of the independent variables and also for the Target column (dependent variable).

4.6.1.3 Split Training and Test Set

The process of splitting a data-set into training and test sets is a fundamental step in the development and evaluation of machine learning models. This division serves two essential purposes: training and validation. The training set, typically comprising a substantial portion of the data, is used to train the model. It provides the algorithm with examples from which it learns patterns and relationships within the data. On the other hand, the test set is kept separate and serves as an independent data-set for evaluating the model's performance. By assessing the model's predictions on unseen data, the test set helps gauge its generalization capability—how well it can make accurate predictions on new, previously unseen data. The split between training and test sets is a crucial aspect of model development, ensuring that the model is not merely memorizing the training data but can effectively apply what it has learned to make accurate predictions in real-world scenarios. Properly conducted split and evaluation procedures are essential for building robust and reliable machine learning models. We split our data-set where 70%

will be used for training and 30% for testing.

Algorithm 3 Data Splitting: 70% Training, 30% Test

Dataset D with n samples

Training set D_{train} , Test set D_{test}

Shuffle D randomly Calculate the split index: $k = \lfloor 0.7n \rfloor$

Split D into two subsets:

D_{train} containing the first k samples

D_{test} containing the remaining $n - k$ samples

return $D_{\text{train}}, D_{\text{test}}$

The choice of the data split ratio, such as 70:30 or any other ratio, depends on several factors, including the size of the data-set, the nature of the problem, and the specific goals of the machine learning project. There isn't a fixed standard split ratio, but 70:30 (or 80:20) is a commonly used default for several reasons. A 70:30 split provides a reasonable balance between the amount of data used for training and testing. It allows the model to learn from a substantial portion of the data while still having a sufficiently large test set for robust evaluation. With a larger test set (30%), the evaluation results are likely to be statistically more significant and reliable. This is especially important when assessing model performance and making decisions based on the evaluation metrics.

Moreover, in practice, a 70:30 split often strikes a good balance between model training time and evaluation effort. Using a larger training set can be computationally expensive, especially for complex models. The choice of split ratio may also depend on the size of the available data-set. If you have a limited amount of data, you might opt for a larger test set to ensure a more rigorous evaluation. In addition to this, it's important to note that the choice of split ratio is not set in stone and can vary based on the specific context. For very large data-sets, you might use a smaller percentage for the test set (e.g., 90:10) because even a small test set can provide sufficient data for evaluation. Conversely, for very small datasets, you might use a larger test set (e.g., 50:50) to ensure a more representative evaluation.

4.7 Model Selection

Machine learning model selection is a critical step in the development of predictive models, as it entails choosing the most appropriate algorithm or architecture to solve a specific problem. This process involves a delicate balance between various factors, including the nature of the data, the complexity of the problem, computational resources, and the desired model performance. Model selection often begins with an exploration of different algorithms, such as decision trees, support vector machines, neural networks, or ensemble methods, among others (Shiksha 2022). Researchers and practitioners assess how well each model generalizes to unseen data through techniques like cross-validation. They consider factors like model complexity, interpretability, and the potential for over-fitting. Ultimately, the goal of model selection is to identify the model that achieves the best trade-off between accuracy, generalization, and practicality for the given task (Badillo et al. 2020). It's a process that requires domain knowledge, iterative experimentation, and a deep understanding of both the data and the problem domain to make informed choices and build robust machine learning systems (Akshay et al. 2021).

Clutch damage prediction falls into the Supervised Machine Learning arena. Supervised learning, at its core, involves providing a computer system with training data consisting of observed inputs paired with their corresponding known output values (Bengio et al. 2012; Lindholm et al. 2019; Muhammad et al. 2015). The objective is to acquire overarching rules or a "model" that can effectively establish a mapping from inputs to outputs. This learning process enables the system to make predictions for new, previously unseen data instances, where we possess input information but lack knowledge of their associated outputs. Supervised learning broadly falls into two primary categories: (i) classification, wherein the output values are categorical or discrete, and (ii) regression, where the output values take on numeric or continuous values. These categories define the nature of the predictive tasks that the supervised learning algorithms are designed to tackle (Badillo et al. 2020).

4.7.1 Logistic Regression

Logistic regression is a widely used statistical method and supervised learning algorithm primarily employed for binary classification tasks. In binary classification, the goal is to predict one of two possible outcomes, typically represented as 0 and 1, true or false, or positive and negative. Logistic regression accomplishes this by modeling the relationship between a set of input features and the probability of an event occurring (Minka et al. 2001).

The algorithm's name, "logistic," stems from its underlying logistic function, which transforms a linear combination of input features into a value bounded between 0 and 1. This transformed value represents the estimated probability that the instance belongs to the positive class. A key strength of logistic regression lies in its simplicity and interpretability. The model produces coefficients associated with each input feature, indicating their impact on the prediction. The logistic regression model aims to model the probability that a binary outcome variable (e.g., 0 or 1) takes a particular value based on a set of input features. The logistic function (also known as the sigmoid function) is used to transform a linear combination of these features into a probability value:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p)}}$$

Where: - $P(Y = 1|X)$ represents the probability that the outcome variable Y takes the value 1 given the input features X . - $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ are the coefficients associated with each feature X_1, X_2, \dots, X_p . - e is the base of the natural logarithm.

The linear combination $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$ is computed, and the logistic (sigmoid) function maps this value to a range between 0 and 1. If $P(Y = 1|X)$ is greater than or equal to 0.5, the model predicts the positive class (1); otherwise, it predicts the negative class (0). This formula represents the mathematical foundation of logistic regression, which estimates the coefficients β during the training process to make probabilistic predictions for binary classification tasks (Feng et al. 2014).

During training, logistic regression optimizes these coefficients to minimize a loss function, typically the log-likelihood or cross-entropy loss, which quantifies the model's deviation from the true labels. Once trained, the logistic regression model can classify

Algorithm 4 Logistic Regression Algorithm for Binary Classification

```
1: Input: Training dataset  $\{(X^{(i)}, Y^{(i)})\}_{i=1}^m$ , learning rate  $\alpha$ , number of iterations  $N$ 
2: Initialize weights  $\theta_0, \theta_1, \dots, \theta_n$  to small random values
3: for  $k$  from 1 to  $N$  do
4:   for  $i$  from 1 to  $m$  do
5:     Compute the hypothesis:  $h^{(i)} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1^{(i)} + \theta_2 X_2^{(i)} + \dots + \theta_n X_n^{(i)})}}$ 
6:     Compute the error:  $error^{(i)} = h^{(i)} - Y^{(i)}$ 
7:     for  $j$  from 0 to  $n$  do
8:       Update weights:  $\theta_j = \theta_j - \alpha \cdot error^{(i)} \cdot X_j^{(i)}$ 
9:     end for
10:   end for
11: end for
12: Output: Trained logistic regression model with weights  $\theta_0, \theta_1, \dots, \theta_n$ 
```

new data points by evaluating the probability and applying a threshold (often 0.5) to determine the predicted class. Due to its simplicity and effectiveness, logistic regression is not only a valuable tool in binary classification tasks but also serves as a fundamental building block in more complex machine learning algorithms and models. It finds applications in various domains, including medical diagnosis, finance, and marketing, where distinguishing between two classes is a common and critical requirement (Liu et al. 2018).

4.7.2 Decision Tree

A decision tree is a versatile and intuitive machine learning algorithm that is commonly used for both classification and regression tasks. It operates by recursively partitioning the dataset into subsets based on the values of input features, ultimately forming a tree-like structure of decision nodes and leaves. At each decision node, the algorithm selects a feature and a corresponding threshold to split the data, aiming to maximize the separation of classes or minimize variance in the case of regression. The tree's structure is determined through a process called recursive binary splitting, guided by criteria like Gini impurity or entropy for classification and mean squared error for regression (Tangirala et al. 2020). Decision trees are highly interpretable, allowing users to trace

the path of a decision and understand the rules behind predictions. However, they can be prone to overfitting when the tree becomes overly complex, which is why techniques like pruning are used to simplify the model. Decision trees are fundamental in ensemble methods like random forests, which aggregate multiple trees to improve predictive accuracy, making them a valuable asset in the field of machine learning and data analysis (Charbuty et al. 2021).

A decision tree can be mathematically represented as a recursive structure, where each node in the tree makes a binary decision based on a feature. Here's a simplified LaTeX formula to illustrate the concept of a decision tree:

```
if  $X_i \leq \text{threshold}_i$  :  
    go to left subtree  
else:  
    go to right subtree
```

In this representation:

- X_i represents the value of the i -th feature.
- threshold_i is the threshold value for feature X_i .
- The decision is made based on whether X_i is less than or equal to threshold_i .
- Depending on the decision, you navigate to the left or right subtree of the decision tree.

Algorithm 5 Decision Tree Algorithm

```
1: function BUILDDECISIONTREE(data, features)
2:   Create a node  $N$ 
3:   if data is pure or a stopping criterion is met then
4:     Assign the class label to  $N$ 
5:   else
6:     Select the best feature  $F$  and a splitting criterion
7:      $N$ .feature  $\leftarrow F$ 
8:     Split data into subsets based on  $F$ 
9:     for all subsets do
10:      Create child node  $N_c$ 
11:       $N_c$ .threshold  $\leftarrow$  splitting threshold
12:      Recursively call BUILDDECISIONTREE(subset, features  $- \{F\}$ ) and assign result to  $N_c$ 
13:      Attach  $N_c$  to  $N$ 
14:     end for
15:   end if
16:   return  $N$ 
17: end function
```

This formula captures the fundamental logic of a decision tree, where features are examined at each node, and decisions are made based on whether a feature value meets a specific condition. The process is recursive, allowing the tree to partition the data into subsets and make predictions based on these binary decisions. The actual mathematical formulas for determining the thresholds and conditions at each node can vary depending on the specific algorithm used to construct the decision tree (e.g., CART or ID3) (Singh et al. 2014).

4.7.3 Support Vector Machine(SVM)

Support Vector Machines (SVMs) represent a powerful and versatile class of supervised machine learning algorithms with applications spanning various domains, making them a prominent choice in the field of pattern recognition and classification. SVMs are particularly renowned for their effectiveness in both linear and non-linear classification tasks, and they excel in scenarios where complex decision boundaries need to be established to separate data into distinct classes (Gold et al. 2003). SVMs aim to find the optimal hyperplane that maximizes the margin between data points of different classes, effectively enhancing generalization and robustness in classification tasks. This optimal hyperplane is the one that minimizes the classification error while maintaining the maximum separation margin, and it is this unique characteristic that distinguishes SVMs from other classifiers. Furthermore, SVMs can handle high-dimensional data efficiently, mitigating the "curse of dimensionality" problem (Cristianini et al. 2000).

SVMs offer adaptability through the use of various kernel functions, allowing them to handle non-linearly separable data by mapping it into higher-dimensional feature spaces, where linear separation becomes possible. This ability to handle non-linearity makes SVMs well-suited for a wide range of applications, including text classification, image recognition, bioinformatics, and financial forecasting. SVMs have demonstrated excellent performance in scenarios with limited training data, making them robust against overfitting. However, it's important to note that tuning parameters such as the choice of kernel and regularization parameters can significantly impact the model's performance, and careful parameter selection is often necessary (Wu et al. 2006; Mukherjee et al. 1999). In summary, Support Vector Machines have earned their reputation as a reliable and versatile tool in machine learning and data analysis. Their ability to handle both linear and non-linear classification problems, their robustness, and their capacity to work efficiently in high-dimensional spaces have made them indispensable in various research and practical applications, contributing significantly to the advancement of pattern recognition and classification tasks across diverse domains (Ukil et al. 2007; Widodo et al. 2007).

Algorithm 6 Support Vector Machine (SVM) Training Algorithm

Input: Training data $\{(x_i, y_i)\}_{i=1}^N$, where x_i is a feature vector and $y_i \in \{-1, 1\}$ is the class label.

Output: SVM model parameters \mathbf{w} (weight vector) and b (bias).

Initialize \mathbf{w} and b to zeros. Choose a regularization parameter $C > 0$. Compute the kernel matrix \mathbf{K} :

$$K_{ij} = y_i y_j \cdot \langle x_i, x_j \rangle$$

Use a quadratic programming solver to solve the following optimization problem:

$$\text{Minimize: } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

$$\text{Subject to: } y_i(\langle \mathbf{w}, x_i \rangle + b) \geq 1 - \xi_i, \quad i = 1, \dots, N$$
$$\xi_i \geq 0, \quad i = 1, \dots, N$$

Extract the support vectors $\{x_i\}$ for which $\xi_i > 0$. Compute \mathbf{w} as follows:

$$\mathbf{w} = \sum_i \alpha_i y_i x_i$$

Compute the bias b as follows:

$$b = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} (y_i - \langle \mathbf{w}, x_i \rangle)$$

where \mathcal{S} is the set of indices of support vectors.

Input Data and Output: In this step, we define the input and output of the SVM training algorithm. The input consists of a set of training data samples, denoted as $(x_i, y_i)_{i=1}^N$, where each x_i represents a feature vector, and y_i is the corresponding class label, typically -1 or 1. The goal is to determine the SVM model parameters \mathbf{w} (weight vector) and b (bias).

Initialization: Here, we initialize the SVM model's weight vector \mathbf{w} and bias b to zero values. These parameters will be updated during the training process to find the optimal hyperplane that best separates the data into different classes.

Kernel Matrix Computation: The algorithm computes the kernel matrix \mathbf{K} to represent the similarity between data samples. The kernel matrix is calculated using the dot product (inner product) between feature vectors and class labels. It serves as a basis for the optimization process, allowing SVMs to work efficiently in high-dimensional

spaces.

Quadratic Programming Optimization: The heart of SVM training involves solving a quadratic programming problem. This optimization aims to find the hyperplane that maximizes the margin between different classes while minimizing classification errors. The objective function balances the margin width and the classification accuracy, controlled by a regularization parameter C .

Support Vector Identification: After solving the optimization problem, we identify the support vectors. These are data samples that lie on the margin boundaries or misclassified samples. Support vectors are crucial as they define the position of the optimal hyperplane.

Weight Vector Calculation: The weight vector \mathbf{w} is computed by summing the contributions of the support vectors, scaled by their associated Lagrange multipliers α_i and class labels y_i . The weight vector determines the orientation of the hyperplane that best separates the data.

Bias (Intercept) Calculation: The bias term b is calculated to shift the hyperplane away from the origin and align it with the data distribution. It's determined by averaging the differences between the true class labels and the predictions based on the support vectors.

Each of these steps plays a critical role in the SVM training process, collectively allowing the algorithm to find the optimal hyperplane that maximizes the margin while minimizing classification errors, thus creating an effective classifier for various applications in machine learning and pattern recognition.

4.7.4 Random Forest

Random Forest is a powerful ensemble learning technique widely employed in machine learning for both classification and regression tasks. This algorithm builds upon the fundamental concept of decision trees, aiming to enhance predictive accuracy and mitigate overfitting issues associated with individual trees. The core idea behind Random Forest lies in constructing a multitude of decision trees during the training phase. Rather than relying on a single tree's prediction, the model aggregates the predictions from multiple trees, ultimately yielding a more robust and reliable outcome. The "random" component in Random Forest manifests in two key ways: feature selection and data sampling. For each tree, a random subset of features is considered at each split, injecting an element of diversity that helps capture various aspects of the data's complexity. Additionally, the training data for each tree is sampled with replacement, introducing variability and preventing the model from being overly sensitive to specific patterns within the dataset (Rigatti et al. 2017).

One of the significant advantages of Random Forest is its ability to handle high-dimensional datasets with numerous features, providing a solution to the "curse of dimensionality." The ensemble nature of Random Forest promotes model stability, reducing the risk of overfitting that may be associated with individual decision trees. Moreover, the algorithm inherently provides a built-in mechanism for feature importance estimation, offering insights into the most influential variables driving the model's predictions (Biau et al. 2016). Random Forest has demonstrated efficacy across diverse domains, from finance to healthcare, owing to its versatility and adaptability to various data complexities. However, it is essential to fine-tune hyperparameters carefully, such as the number of trees and maximum depth, to optimize performance. Overall, Random Forest stands as a formidable tool in the machine learning toolkit, renowned for its versatility, accuracy, and robustness in addressing real-world challenges (Breiman et al. 2001).

To maintain simplicity, our focus is solely on the binary classification challenge, although it's crucial to acknowledge that random forests inherently possess the capability to address problems involving multiple classes. Within this binary classification context, the stochastic response variable Y assumes values in the set $(0, 1)$. Given the predictor variable X , the objective is to predict the corresponding value of Y . A classifier or classification rule, denoted as m_n , represents a Borel measurable function of both X and D_n , aiming to provide an estimate of the label Y based on the information available in X and D_n . Within this framework, the term "consistent" is attributed to a classifier m_n if its conditional probability of error is minimized. In simpler terms, a consistent classifier strives to minimize the likelihood of making errors when predicting the label Y based on the given predictor variable X and auxiliary information D_n (Biau et al. 2016).

$$L(m_n) = P[m_n(\mathbf{X}) \neq Y] \xrightarrow{n \rightarrow \infty} L^*,$$

where L^* is the error of the optimal-but unknown-Bayes classifier:

$$m^*(\mathbf{x}) = \begin{cases} 1 & \text{if } P[Y = 1 \mid \mathbf{X} = \mathbf{x}] > P[Y = 0 \mid \mathbf{X} = \mathbf{x}] \\ 0 & \text{otherwise.} \end{cases}$$

In the classification context, the random forest classifier is obtained via a majority vote among the classification trees, that is,

$$m_{M,n}(\mathbf{x}; \Theta_1, \dots, \Theta_M, \mathcal{D}_n) = \begin{cases} 1 & \text{if } \frac{1}{M} \sum_{j=1}^M m_n(\mathbf{x}; \Theta_j, \mathcal{D}_n) > 1/2 \\ 0 & \text{otherwise.} \end{cases}$$

If a leaf represents region A , then a randomized tree classifier takes the simple form.

Algorithm 7 Random Forest for Binary Classification

Require: Training dataset D , number of trees T , number of features to consider at each split m

Ensure: Random Forest classifier RF

```
1:  $RF \leftarrow \emptyset$ 
2: for  $t \leftarrow 1$  to  $T$  do
3:    $D_t \leftarrow$  Randomly sample with replacement from  $D$ 
4:    $m_t \leftarrow$  Randomly select  $m$  features
5:    $DT \leftarrow$  TrainDecisionTree( $D_t, m_t$ )           ▷ Train a decision tree
6:    $RF \leftarrow RF \cup \{DT\}$                        ▷ Add the decision tree to the forest
7: end for
8: function RANDOMFORESTPREDICT( $RF, x$ )
9:    $predictions \leftarrow \emptyset$ 
10:  for  $DT \in RF$  do
11:     $prediction \leftarrow$  PredictUsingDecisionTree( $DT, x$ )
12:     $predictions \leftarrow predictions \cup \{prediction\}$ 
13:  end for
14:  return MajorityVote( $predictions$ )           ▷ Final prediction by majority voting
15: end function
16: function MAJORITYVOTE( $predictions$ )
17:  if  $\sum_{p \in predictions} p > \frac{T}{2}$  then
18:    return 1           ▷ Classify as positive
19:  else
20:    return 0           ▷ Classify as negative
21:  end if
22: end function
```

4.7.5 Artificial Neural Networks(ANN)

The Artificial Neural Network (ANN) represents a cutting-edge technology rooted in the intricate structure of the human brain, simulating its remarkable learning capabilities. This neural approach holds the promise of deriving valuable insights from past experiences. When an ANN undergoes training with historical data, it gains the proficiency to generate informed outputs based on the knowledge distilled from the data. Numerous research endeavors have unequivocally demonstrated the potency of ANN in the domain of classification. The adoption of ANN for classification tasks is underpinned by a multitude of compelling advantages. To begin, ANNs possess the remarkable ability to adapt to data without imposing prior assumptions on underlying functions (Jeatrakul et al. 2009; Min et al. 2009).

The architecture of neural networks plays a pivotal role in shaping the success and efficacy of binary classification tasks, marking a crucial intersection between model complexity, capacity, and interpretability. The structure of a neural network, characterized by layers, nodes, and connections, governs its capacity to comprehend complex representations, making it particularly well-suited for tasks involving intricate features and non-linear relationships (Jeatrakul et al. 2009).

Furthermore, the architecture influences the network's generalization ability. A well-designed architecture ensures that the model not only performs optimally on the training data but also generalizes effectively to unseen data. Overly complex architectures may lead to overfitting, where the model memorizes training samples but struggles with new, unseen instances. On the other hand, architectures with insufficient complexity may fail to capture essential patterns, resulting in underfitting (Koehrsen et al. 2018). In binary classification tasks, the role of neural network architecture extends beyond mere predictive accuracy. It influences the interpretability of the model, providing insights into the decision-making process (Zhang et al. 2018). Techniques such as attention mechanisms and interpretability-focused architectures contribute to understanding which features are crucial for the classification decision (Lipton et al. 2018).

Additionally, they serve as universal function approximators, capable of closely approximating virtually any function with remarkable precision. Furthermore, ANNs offer the invaluable attribute of nonlinearity, rendering them highly adaptable for complex real-world applications across diverse fields such as industry, business, and science. The

realm of successful applications for ANNs is extensive and includes noteworthy examples like bankruptcy prediction, handwriting recognition, fault detection, and medical diagnosis, illustrating the profound impact of this technology in addressing real-world challenges (Faris et al. 2016).

Algorithm 8 Binary Classification with Neural Network

- 1: **Input:** Training data $\{(x_i, y_i)\}_{i=1}^N$, where x_i is the input feature vector and $y_i \in \{0, 1\}$ is the binary class label.
 - 2: **Output:** Trained neural network parameters θ , which include weights and biases.
 - 3: Initialize neural network architecture: Define the number of layers, neurons per layer, activation functions (e.g., sigmoid, ReLU), and other hyperparameters.
 - 4: Initialize model parameters θ : Initialize the weights and biases for each neuron in the network randomly or using specific initialization methods (e.g., Xavier/Glorot initialization).
 - 5: Choose a loss function: Select a suitable loss function for binary classification tasks, such as binary cross-entropy loss.
 - 6: Choose an optimization algorithm: Select an optimization algorithm (e.g., gradient descent, Adam) to update the model parameters θ in order to minimize the chosen loss function.
 - 7: **for** each training epoch **do**
 - 8: **for** each training sample (x_i, y_i) **do**
 - 9: Perform forward propagation: Compute the output of the neural network by applying the activation functions to the weighted sum of inputs.
 - 10: Compute the loss: Calculate the loss between the predicted output and the true label using the chosen loss function.
 - 11: Perform backward propagation: Compute the gradients of the loss with respect to the model parameters θ using backpropagation.
 - 12: Update model parameters: Update the weights and biases using the chosen optimization algorithm to minimize the loss.
 - 13: **end for**
 - 14: **end for**
 - 15: **Output:** Trained neural network with optimized parameters θ .
-

4.7.6 Models Weakness and Strength

In the domain of machine learning (ML), binary classification is a fundamental task where the objective is to assign one of two possible labels to each data point. Examples of such tasks include spam detection, disease diagnosis, and customer churn prediction. While there are numerous machine learning algorithms designed for binary classification, the choice of the most suitable model is not always obvious. Thus, it is standard practice to experiment with different models during the development process to determine which one yields the best performance. The rationale for this experimentation lies in the fact that each model has distinct underlying mechanisms, assumptions, and capabilities, leading to varying levels of effectiveness depending on the nature of the dataset and the problem being addressed (Lindholm et al. 2019).

The main reason for experimenting with different ML models in binary classification is the "no free lunch" theorem. This principle suggests that no single algorithm performs optimally across all types of problems or datasets. Each classification model has specific strengths and weaknesses, and its performance is often contingent on the characteristics of the data, such as feature distribution, data dimensionality, noise levels, class imbalance, and the relationships between features and target variables. As a result, selecting the right model is highly problem-dependent (Akshay et al. 2021).

Furthermore, different models have varying tendencies when it comes to bias and variance trade-offs. Some models may be prone to overfitting, capturing too much noise in the data, while others may be too simplistic and underfit the data, failing to capture complex relationships. By experimenting with multiple models, practitioners can identify the algorithm that strikes the best balance between bias and variance, thereby achieving better generalization on unseen data.

Additionally, the interpretability of different models plays a crucial role in selecting the appropriate model for binary classification. In certain domains, such as healthcare or finance, interpretability is just as important as accuracy. For example, decision trees provide a level of interpretability by showing the rules that lead to a particular decision, whereas models such as neural networks are often considered "black boxes" due to their complex internal mechanisms. Depending on the domain requirements, experimenting with models allows researchers to choose between high accuracy or higher interpretability.

- **Logistic Regression:** Logistic regression is a linear model that assumes a linear relationship between the input features and the log-odds of the binary outcome. It uses the sigmoid function to map predicted values to a probability between 0 and 1. Logistic regression is widely used due to its simplicity and interpretability. However, its performance is limited when the true relationship between features and the target is highly non-linear. It is also sensitive to outliers and multicollinearity in the data.
- **Support Vector Machines (SVM):** SVM is a powerful classification algorithm that works by finding the hyperplane that best separates the data into two classes. SVM is particularly effective for high-dimensional data and is capable of handling non-linear relationships by using kernel functions. The choice of kernel (linear, polynomial, radial basis function, etc.) significantly affects the model's performance. However, SVMs can be computationally expensive, especially for large datasets, and require careful tuning of hyperparameters such as the regularization parameter and kernel choice.
- **Decision Trees:** Decision trees split the dataset into subsets based on the values of input features. They are highly interpretable, as they provide a visual representation of the decision-making process. Decision trees are also flexible, capable of modeling both linear and non-linear relationships. However, they tend to be prone to overfitting, especially when the tree becomes too deep. Regularization techniques such as pruning or setting a maximum tree depth are often employed to counteract this issue.
- **Random Forest:** Random Forest is an ensemble learning method that builds multiple decision trees during training and outputs the class that is the mode of the classes from individual trees. The model's strength lies in its ability to reduce overfitting by averaging the predictions of various trees, leading to more robust and stable predictions. Random Forests are versatile, handling both categorical and continuous data well. However, their interpretability is lower than individual decision trees, and they may be computationally expensive for large datasets.
- **Artificial Neural Networks (ANN):** Neural networks are a class of models inspired by the human brain, consisting of layers of interconnected neurons that can learn

complex, non-linear relationships in data. ANNs are particularly powerful when there are many features or when the data exhibits complex interactions. However, they require significant computational resources and are less interpretable than other models like logistic regression or decision trees. Neural networks also tend to require large amounts of training data to avoid overfitting and to perform well.

Experimenting with different machine learning models for binary classification is an essential step in the model selection process. The diversity of algorithms, each with its own strengths and weaknesses, reflects the varied nature of datasets and classification problems. By testing multiple models, practitioners can optimize the trade-off between bias and variance, account for data-specific characteristics, and choose a model that best aligns with the problem's requirements—whether that be accuracy, interpretability, or efficiency. This experimentation process ensures that the final model not only performs well on the data but is also suited to the real-world context in which it will be applied.

Chapter 5

Results

5.1 Evaluation Metric

Evaluation metrics for binary classification in machine learning play a pivotal role in assessing the performance and effectiveness of classification models. These metrics provide valuable insights into the model's ability to distinguish between two classes, typically referred to as the positive class (e.g., presence of a disease) and the negative class (e.g., absence of a disease). One of the most fundamental and widely used evaluation metrics is accuracy, which measures the overall correctness of predictions. However, accuracy alone may be misleading, especially in imbalanced datasets where one class vastly outnumbers the other (Canbek et al. 2022). In such cases, metrics like precision, recall (sensitivity), and the F1-score come into play. Precision quantifies the proportion of true positive predictions among all positive predictions, emphasizing the minimization of false positives. Recall, on the other hand, gauges the ability to capture all actual positive instances, thus minimizing false negatives. The F1-score strikes a balance between precision and recall, serving as a harmonic mean of these two metrics, which is particularly valuable when seeking a balanced trade-off between precision and recall (Hossin et al. 2015; Chicco et al. 2020).

In addition to these metrics, the ROC (Receiver operating characteristic curve) curve and AUC (Area Under the Curve) are essential tools for evaluating the classifier's performance across different thresholds, enabling a nuanced understanding of its discrimina-

tive power (Purves et al. 1992; Mandrekarand et al. 2010). Beyond these common metrics, specific applications may necessitate domain-specific evaluation measures, highlighting the flexibility and adaptability of machine learning evaluation. Ultimately, the selection of an appropriate evaluation metric depends on the nature of the problem, the importance of different types of errors, and the overarching goal of the classification task, underscoring the critical role of thoughtful metric choice in assessing the efficacy of binary classification models. We have selected F1-Score and imbalanced accuracy to evaluate model performance.

5.2 Results: Imbalanced data-set

In this section, we delve into the heart of our research findings, presenting a comprehensive evaluation of our machine-learning binary classification model. Here, we scrutinize the model's performance through a range of meticulously selected evaluation metrics, shedding light on its ability to effectively distinguish between two classes - a task of paramount importance in the context of our study. We provide a detailed account of metrics such as accuracy, precision, recall, and F1-score, as well as the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC). These metrics collectively form the foundation of our assessment, offering an in-depth perspective on the classifier's strengths and areas of improvement.

Moreover, we examine the influence of various experimental factors, including hyperparameter tuning, feature engineering strategies, and data pre-processing techniques, on the model's performance. Visual aids, including tables and graphs, will be employed to enhance the clarity of our presentation, allowing readers to grasp the intricacies of our classification results. This Results section represents a crucial juncture in our study, providing empirical validation for our chosen machine-learning approach and paving the way for a deeper understanding of the subject matter at hand. At first we will outline the result with imabalnaced data as it is. In later section 5.3, we will Oversample the data and report the accuracy of balanced data in section 5.4.

Note: The imbalanced data is denoted by **Imb** and the balanced data by **balnc**

throughout this document.

5.2.1 Result Imb Data: Logistic Regression

Here we will outline the result of the Logistic regression with the imbalanced dataset. The outcome of the logistic regression analysis conducted on the imbalanced dataset will be delineated here. This comprehensive outline will encapsulate the findings and insights derived from the regression model, accounting for the inherent imbalances within the dataset. By addressing the intricacies of imbalanced data, the analysis will shed light on the predictive performance, highlighting the nuances of the logistic regression's outcomes, including its efficacy in handling the disparities in class distribution.

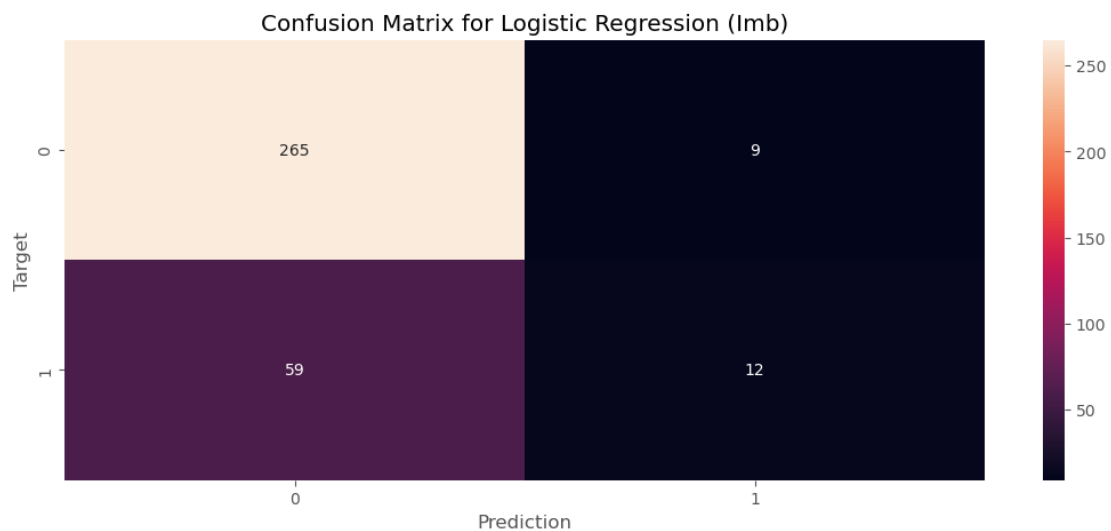


Figure 30: Confusion Matrix for Imb Logistic Regression Prediction

Here in the picture above 30 we can see that the model favors class 0 and achieves higher accuracy and the minority class 1 received disadvantages. The sole reason for such a result can be the class imbalance which we have explained in the earlier chapter 4.4.

While the overall accuracy stands at an impressive 80%, it is essential to note that the metrics portraying recall and F1 score exhibit notably lower values. This discrepancy

Index	Metric	Score
0	Accuracy	0.800000
1	Precision	0.583333
2	Recall	0.098592
3	F1 Score	0.168675

Table 4: Evaluation Metrics for Logistic Regression with Imbalanced Data

arises primarily from the limited data availability for both classes within the dataset. The inadequate representation of these classes skews the evaluation, resulting in diminished performance metrics like recall and F1 score.

Despite the commendable accuracy, the imbalanced nature of the dataset adversely impacts the model’s ability to effectively capture and predict instances from the minority class, leading to a substantial drop in these crucial evaluation metrics. Addressing the imbalance by potentially employing techniques like oversampling, undersampling, or utilizing algorithms designed to handle imbalanced data could potentially improve these specific metrics, ensuring a more comprehensive evaluation of the model’s performance. In the later section, we will also show the results of the balanced data set.

The Receiver Operating Characteristic (ROC) curve is a graphical representation used to illustrate the performance of a binary classification model across various thresholds. The Area Under the Curve (AUC) metric measures the overall performance of this curve (Marchetti et al. 2016; Tangirala et al. 2020). An AUC value of 0.77 signifies that the model has a reasonably good ability to distinguish between the two classes. Specifically, an AUC of 0.77 indicates that the model has a 77% chance of correctly distinguishing between a randomly chosen positive instance and a randomly chosen negative instance. In other words, if you randomly select a positive sample and a negative sample, the model correctly ranks them 77% of the time, based on their predicted probabilities.

An AUC value closer to 1 suggests better overall performance, indicating that the model has a higher true positive rate and a lower false positive rate across various threshold values. However, an AUC of 0.77 still suggests that the model is capable of making

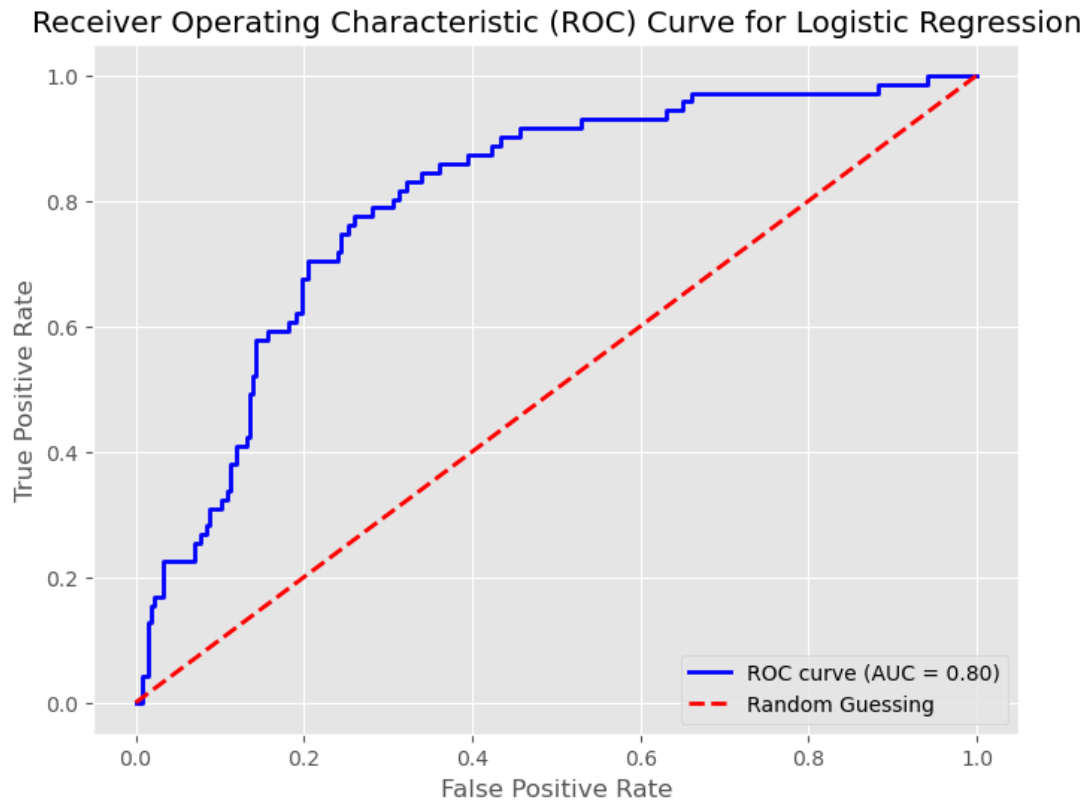


Figure 31: ROC and AUC for Imb Logistic Regression Prediction

reasonably good predictions, although there might be room for further improvement depending on the specific context and requirements of the application. In the figure provided for plotting the ROC curve 31, there is a line that plots a diagonal dashed line from the bottom left to the top right of the graph with the label 'Random Guessing'. This line represents the performance of a random classifier that makes predictions by chance.

In a binary classification problem, a random classifier would essentially make random guesses without considering the input data. It would have no discrimination ability and would randomly assign class labels. This line, which diagonally connects the points (0,0) and (1,1) on the ROC curve, represents the scenario where the model's performance is no better than random guessing. By including this line on the ROC plot, it serves as a reference point to demonstrate the performance of the actual classifier. A good classifier's ROC curve should be positioned as far away as possible from this

diagonal line toward the top-left corner of the plot, indicating superior performance compared to random guessing.

5.2.2 Result Imb Data: Decision Tree

Decision Tree provides valuable insights into the model's performance and its ability to discern between two classes. The Decision Tree algorithm operates by recursively splitting the feature space based on the most discriminative features, creating a hierarchical structure of decision nodes.

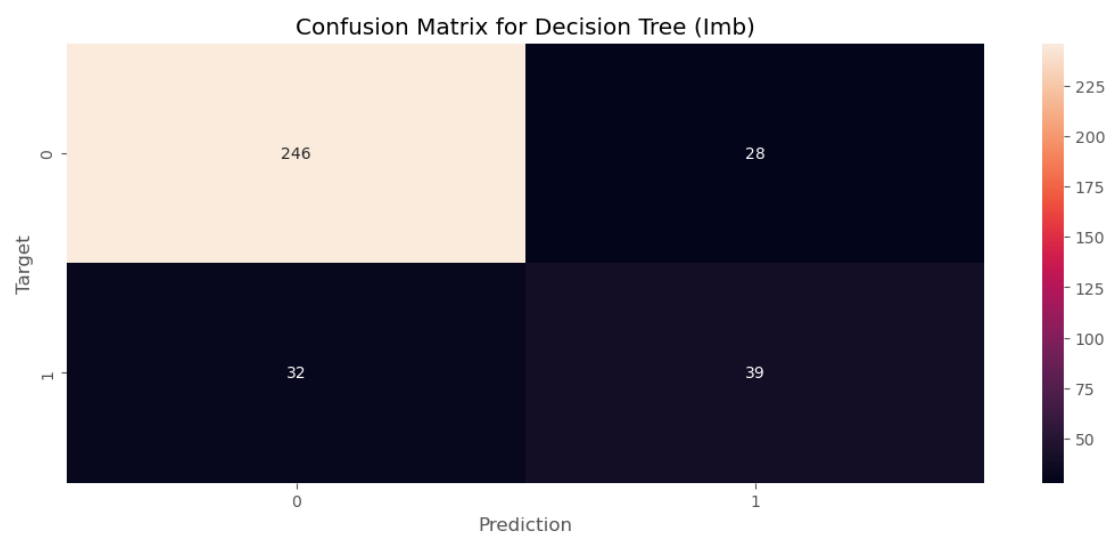


Figure 32: Confusion Matrix for Imb Decision Tree Prediction

The confusion matrix 32 showed that 43 instances were correctly predicted as positive (class 1). These are cases where the model correctly identified (True Positive) instances belonging to the positive class. False Positives where 16 instances were incorrectly predicted as positive. These are cases where the model falsely identified instances as belonging to the positive class when they actually belong to the negative class. True Negatives where 258 instances were correctly predicted as negative (class 0). These are cases where the model accurately identified instances belonging to the negative class. And finally, False Negatives where 28 instances were incorrectly predicted as negative. These are cases where the model falsely identified instances as belonging to the negative

class when they actually belonged to the positive class.

With an accuracy of approximately 82.6%, the model demonstrates a commendable ability to correctly classify instances across both positive and negative classes. Precision, denoting the accuracy of positive predictions, is calculated at approximately 58.2%. This implies that when the model predicts the positive class (clutch failure happened), it is accurate around 58.2% of the time. Furthermore, the recall, or sensitivity, stands at approximately 54.9%, indicating the model's effectiveness in capturing a substantial portion of the true positive instances. The F1 Score, a harmonized metric balancing precision and recall, is calculated at approximately 56.5%, providing a consolidated measure of the model's overall performance.

Index	Metric	Score
0	Accuracy	0.826087
1	Precision	0.582090
2	Recall	0.549296
3	F1 Score	0.565217

Table 5: Evaluation Metrics for Decision Tree with Imbalanced Data

These metrics collectively suggest that while the model exhibits notable accuracy, there is room for improvement in precision and recall. This observation prompts a nuanced interpretation, indicating that the model is reasonably adept at making correct predictions overall, but there is potential for refinement in its ability to precisely identify positive instances and capture a greater proportion of actual positive instances. These insights derived from accuracy, precision, recall, and F1 Score collectively guide further iterations and optimizations to enhance the model's robustness and effectiveness in the specific binary classification task at hand.

The ROC curve (ref: 33), which stands for Receiver Operating Characteristic, is a graphical representation of the trade-off between the true positive rate (sensitivity) and the false positive rate at various thresholds for a binary classification model. The AUC, or Area Under the Curve, is a quantitative measure of the ROC curve's performance, providing a single value to assess the model's ability to discriminate between the positive and negative classes.

In the case of the Decision Tree model we have achieved an AUC of 0.72, the curve suggests that the model performed moderately well in distinguishing between the two classes. An AUC value closer to 1 indicates a stronger ability to differentiate between positive and negative instances, while a value of 0.5 suggests random guessing. Therefore, the AUC of 0.72 implies that the Decision Tree model demonstrated reasonable discriminative power, although there is room for improvement.

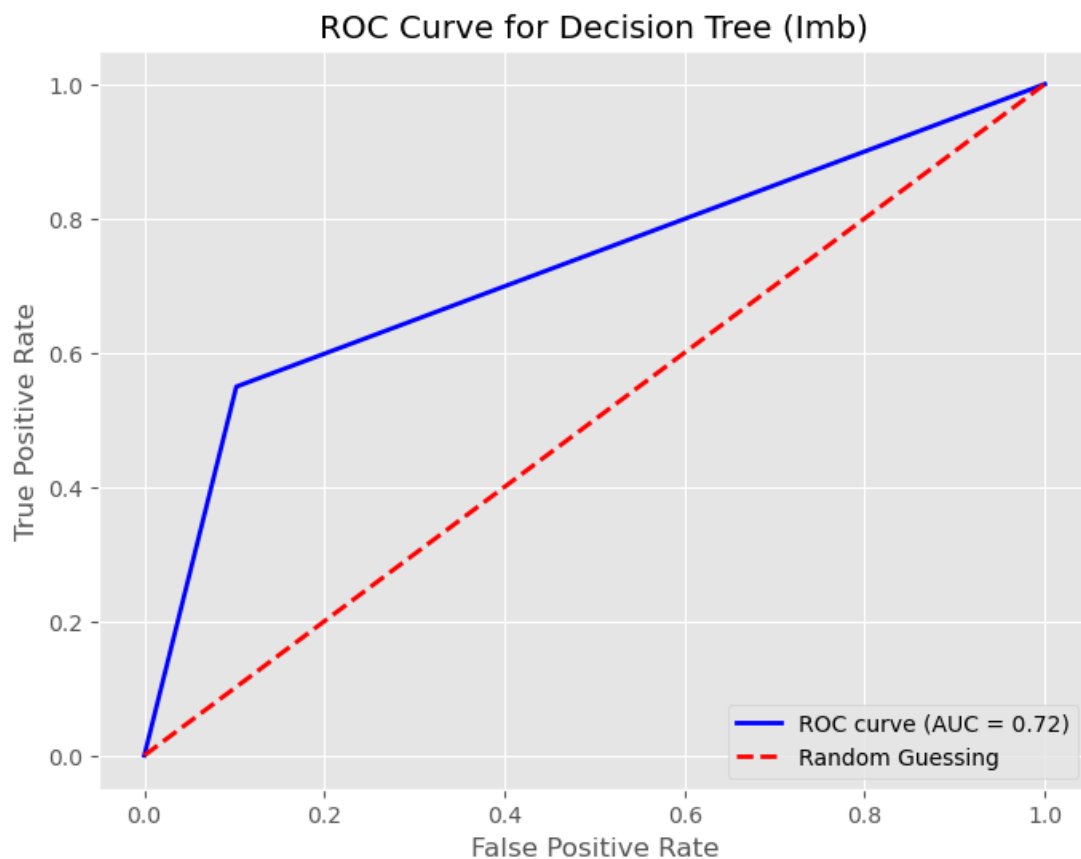


Figure 33: ROC Curve for Imb Decision Tree Prediction

A higher true positive rate (sensitivity) and a lower false positive rate are desirable for our model, as this signifies that the model correctly identifies positive instances while minimizing the misclassification of negative instances. In the context of our Decision Tree model with an AUC of 0.72, there is evidence of satisfactory discrimination, yet potential enhancements in performance could be explored to further improve the

model's ability to distinguish between positive and negative classes.

5.2.3 Result Imb Data: Support Vector Machine

The confusion matrix for the binary classification task conducted using a Support Vector Machine (SVM) is presented in the figure.

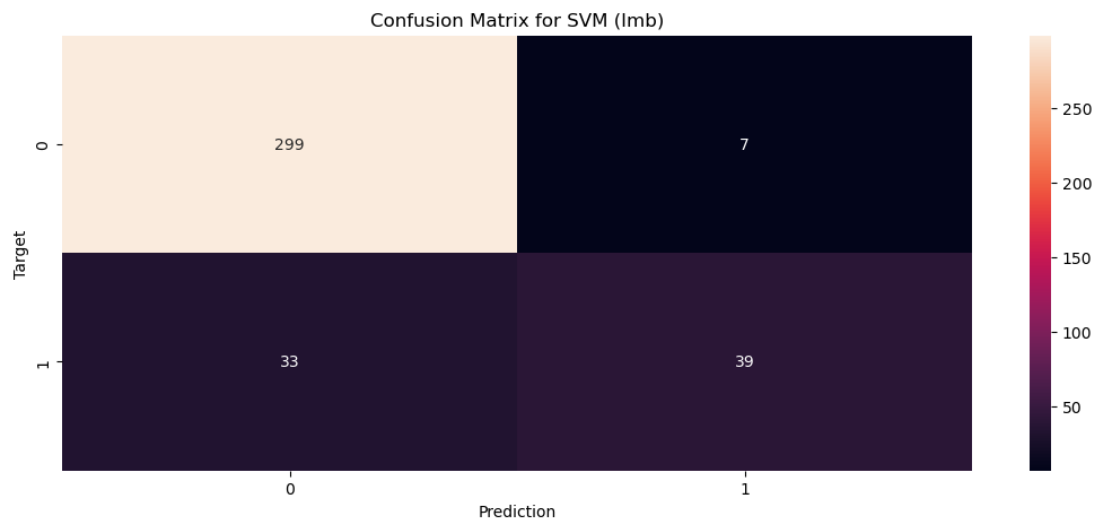


Figure 34: Confusion Matrix for Imb SVM Prediction

This matrix encapsulates the outcomes of the SVM model's predictions, where the rows represent the actual class labels and the columns signify the predicted class labels. In this context, the upper-left quadrant (299) corresponds to True Negatives (TN), indicating instances accurately classified as the negative class. These instances likely represent situations where the SVM correctly identified benign outcomes or the absence of the targeted condition. On the contrary, the bottom-right quadrant (39) represents True Positives (TP), signifying instances correctly identified as the positive class. These instances likely pertain to the accurate detection of the presence or manifestation of the targeted condition.

Conversely, the upper-right quadrant (7) denotes False Positives (FP), where instances from the negative class were inaccurately predicted as positive. This introduces

the possibility of misclassification, highlighting areas where the SVM model has falsely identified instances as positive. Similarly, the bottom-left quadrant (33) signifies False Negatives (FN), indicating instances from the positive class that were erroneously classified as negative. This suggests instances where the SVM model failed to detect the presence of the targeted condition. The analysis of this confusion matrix 34 allows for the computation of various performance metrics, such as accuracy, precision, recall, and the F1 Score which are shown in the table 6.

Index	Metric	Score
0	Accuracy	0.894180
1	Precision	0.847826
2	Recall	0.541667
3	F1 Score	0.661017

Table 6: Evaluation Metrics for SVM with Imbalanced Data

Beginning with accuracy, the model achieved an impressive accuracy rate of approximately 89.42%. This metric signifies the proportion of correctly classified instances out of the entire dataset, reflecting the overall correctness of the model's predictions. The high accuracy suggests that the SVM model excels in making correct binary classifications, showcasing its robustness in distinguishing between the two classes under consideration. However, while accuracy offers a holistic view, precision and recall provide a more nuanced understanding of the model's behavior, especially in scenarios with imbalanced class distributions.

Precision, computed at approximately 84.78%, offers a measure of the accuracy of positive predictions among all instances predicted as positive. This indicates that when the SVM model asserts a positive prediction, it is accurate around 84.78% of the time. Precision becomes particularly relevant in scenarios where the cost of false positives is significant. On the other hand, recall, or sensitivity, calculated at approximately 54.17%, gauges the model's ability to capture all positive instances among the actual positives. The relatively lower recall suggests that the model misses a considerable portion of positive instances in the dataset (clutch failure happened). This trade-off between precision and recall is encapsulated by the F1 Score, which harmonizes these metrics, yielding

a value of approximately 66.10%. The F1 Score, being the harmonic mean of precision and recall, provides a balanced assessment, indicating a reasonable compromise between the competing objectives of precision and recall in the SVM model's binary classification performance.

In summary, while the SVM model exhibited high accuracy, a more nuanced examination through precision, recall, and the F1 Score revealed valuable insights into its strengths and limitations, guiding further optimization efforts based on the specific requirements of the binary classification task.

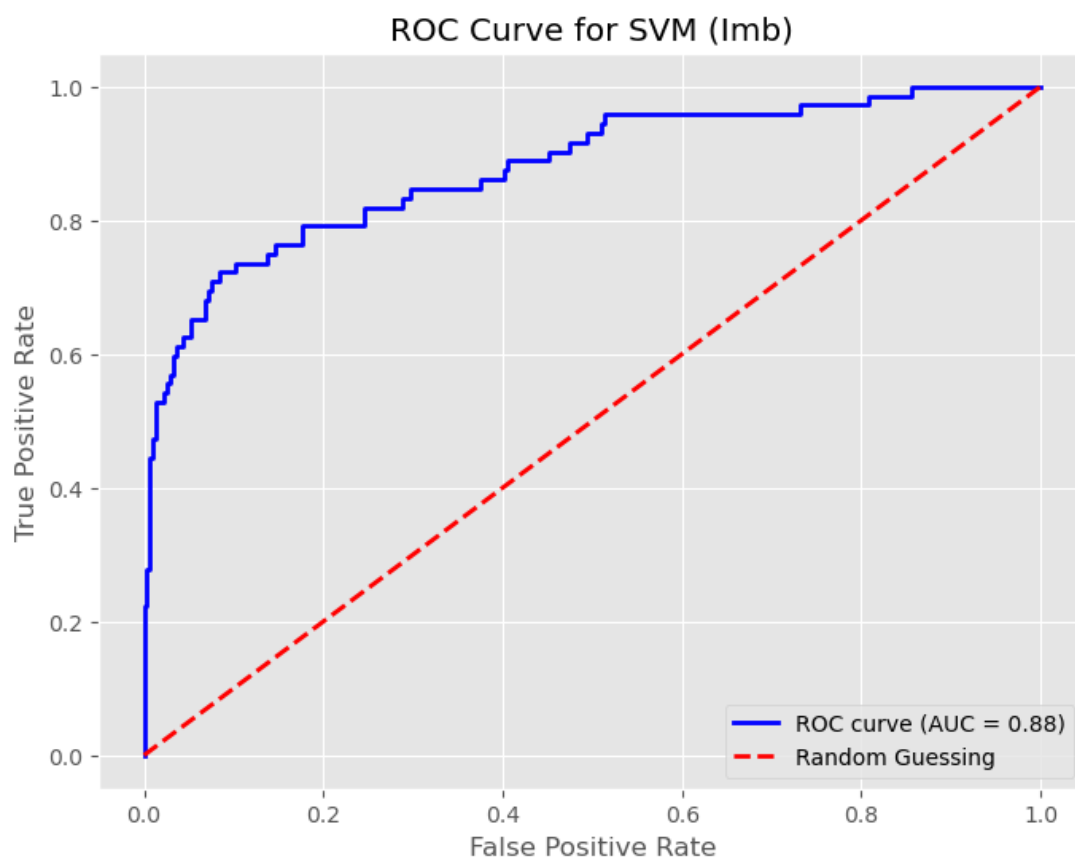


Figure 35: ROC Curve for Imb SVM Prediction

In this context, an AUC of 88% indicates a relatively high level of discriminatory power, as it measures the probability that the SVM model ranks a randomly chosen positive instance higher than a randomly chosen negative instance. The closer the AUC is to

1, the better the model's ability to distinguish between the positive and negative classes. Therefore, an AUC of 88% suggests that the SVM model exhibits strong discriminatory performance, providing confidence in its ability to make accurate predictions in the binary classification task.

5.2.4 Result Imb Data: Random Forest

The provided confusion matrix 36 encapsulates the outcomes of a binary classification task conducted using a Random Forest model. In this matrix, the upper-left quadrant (268) represents True Negatives (TN), indicating instances accurately classified as the negative class. These instances are likely cases where the Random Forest correctly identified situations devoid of the targeted condition or outcomes characterized as benign. Conversely, the bottom-right quadrant (38) represents True Positives (TP), signifying instances correctly identified as the positive class. These instances are indicative of the model's accurate detection of the presence or manifestation of the targeted condition.

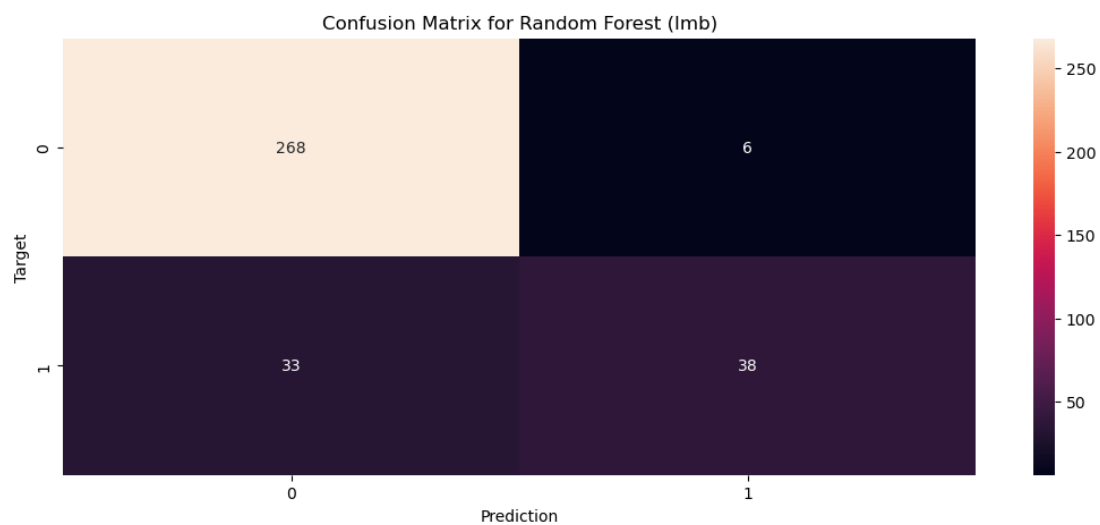


Figure 36: Confusion Matrix for Imb Random Forest Prediction

On the contrary, the upper-right quadrant (6) denotes False Positives (FP), where instances from the negative class were inaccurately predicted as positive. This introduces

the potential for misclassification, suggesting areas where the Random Forest model may have falsely identified instances as positive. Similarly, the bottom-left quadrant (33) signifies False Negatives (FN), indicating instances from the positive class that were erroneously classified as negative. This implies instances where the Random Forest model failed to detect the presence of the targeted condition.

The examination of these elements within the confusion matrix allows for a more detailed understanding of the Random Forest model's classification performance, offering insights into both correct and incorrect predictions and serving as a foundation for further evaluation.

Index	Metric	Score
0	Accuracy	0.886957
1	Precision	0.863636
2	Recall	0.535211
3	F1 Score	0.660870

Table 7: Evaluation Metrics for Random Forest with Imbalanced Data

In terms of performance metrics derived from the confusion matrix, the model achieves an accuracy of approximately 88.71%, reflecting the overall correctness of its predictions. Precision, computed at around 86.36%, signifies the accuracy of positive predictions among all instances predicted as positive. This suggests that when the Random Forest asserts a positive prediction, it is accurate around 86.36% of the time. Furthermore, the recall, or sensitivity, is calculated at approximately 53.52%, indicating the model's ability to capture all positive instances among the actual positives. The F1 Score, which harmonizes precision and recall, is approximately 66.67%. These metrics provide a comprehensive evaluation of the Random Forest model's efficacy in binary classification, considering both its strengths in accurate predictions and areas that may benefit from refinement in capturing positive instances.

5.2.5 Result Imb Data: Artificial Neural Network

In the context of training a neural network model for binary classification, monitoring both training and validation accuracy is crucial for assessing the model's performance and generalization capabilities. During the training phase, the model learned from the provided dataset, adjusting its internal parameters to minimize the discrepancy between its predictions and the actual labels. The training accuracy reflected the proportion of correctly classified instances within the training dataset, serving as an indicator of how well the model is adapting to the training data. However, achieving high training accuracy alone does not guarantee the model's effectiveness on new, unseen data.

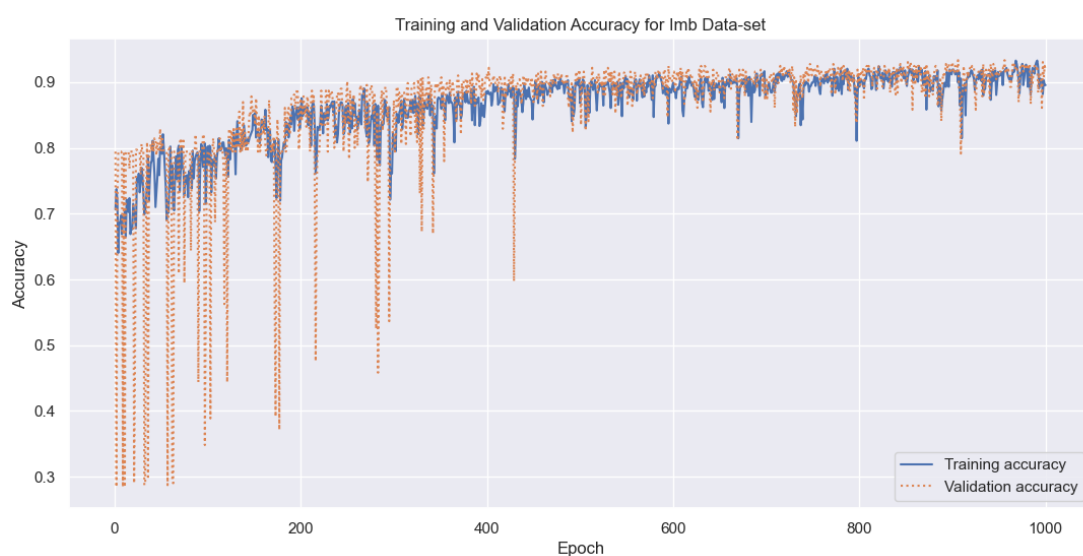


Figure 37: Neural Network Training and Validation Accuracy for Imb Data-set

Validation accuracy, on the other hand, is assessed on a separate dataset that the model has not encountered during training. This dataset functions as a proxy for real-world scenarios, allowing the evaluation of the model's generalization performance. A high training accuracy coupled with a significantly lower validation accuracy might suggest overfitting, where the model becomes too specialized in capturing the training data's nuances but fails to generalize to new instances. Conversely, similar performance on both training and validation datasets indicates that the model is likely learning essential patterns without overfitting. Balancing training and validation accuracy is a key

aspect of optimizing a neural network model for binary classification, ensuring its ability to make accurate predictions on previously unseen data. Regular monitoring, fine-tuning, and the application of techniques like dropout or regularization contribute to enhancing both training and validation accuracy, fostering a robust and well-generalizing neural network model.

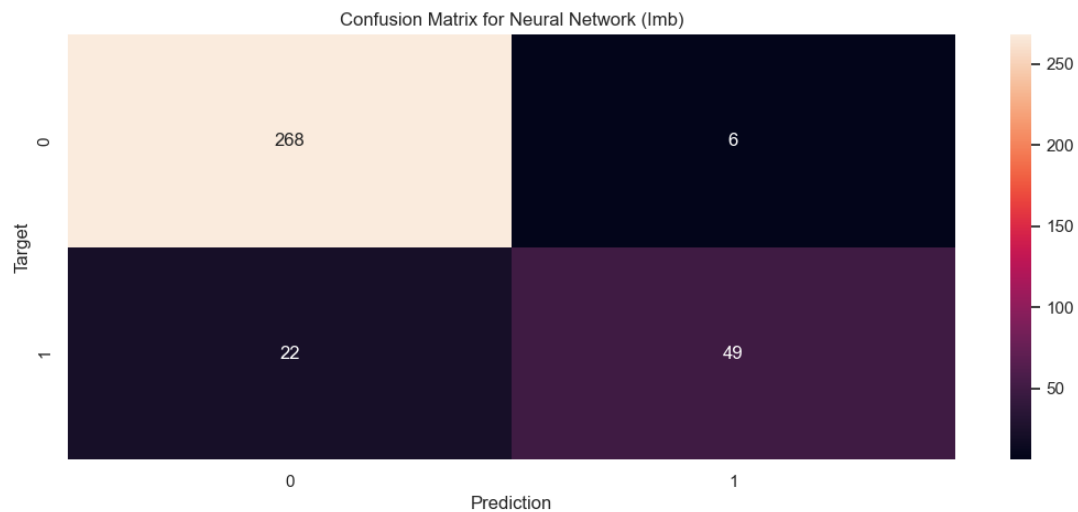


Figure 38: Confusion Matrix for Imb Neural Network Prediction

Starting with the upper-left quadrant (268), these instances represent True Negatives (TN), indicating cases where the model correctly classified instances as belonging to the negative class (Clutch failure didn't happen). In the context of binary classification, this denotes situations where the model accurately identified benign outcomes or conditions absent from the targeted class. The bottom-right quadrant (49) reflects True Positives (TP), signifying instances accurately classified as belonging to the positive class (Clutch failure happened). These instances showcase the model's ability to correctly detect the presence or manifestation of the targeted condition.

On the flip side, the upper-right quadrant (6) signifies False Positives (FP), where instances from the negative class were inaccurately predicted as positive. This introduces the misclassification, suggesting areas where the neural network model has falsely identified instances as positive. Similarly, the bottom-left quadrant (22) denotes False Neg-

atives (FN), indicating instances from the positive class that were erroneously classified as negative. This implies instances where the neural network model failed to detect the presence of the targeted condition. Derived from these elements, several performance metrics can be computed. These metrics collectively offer a comprehensive evaluation of the neural network model's performance in binary classification, considering both correct and incorrect predictions and providing insights into its precision, recall, and overall predictive accuracy. See table 8.

Index	Metric	Score
0	Accuracy	0.92134
1	Precision	0.89192
2	Recall	0.69012
3	F1 Score	0.77789

Table 8: Evaluation Metrics for Neural Network with Imbalanced Data

Achieving a remarkable Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) of 91% for the neural network model 39 underscores its robust discriminatory capacity in binary classification. The AUC is a pivotal metric that evaluates the model's ability to distinguish between the positive and negative classes, providing a comprehensive measure of its overall performance.

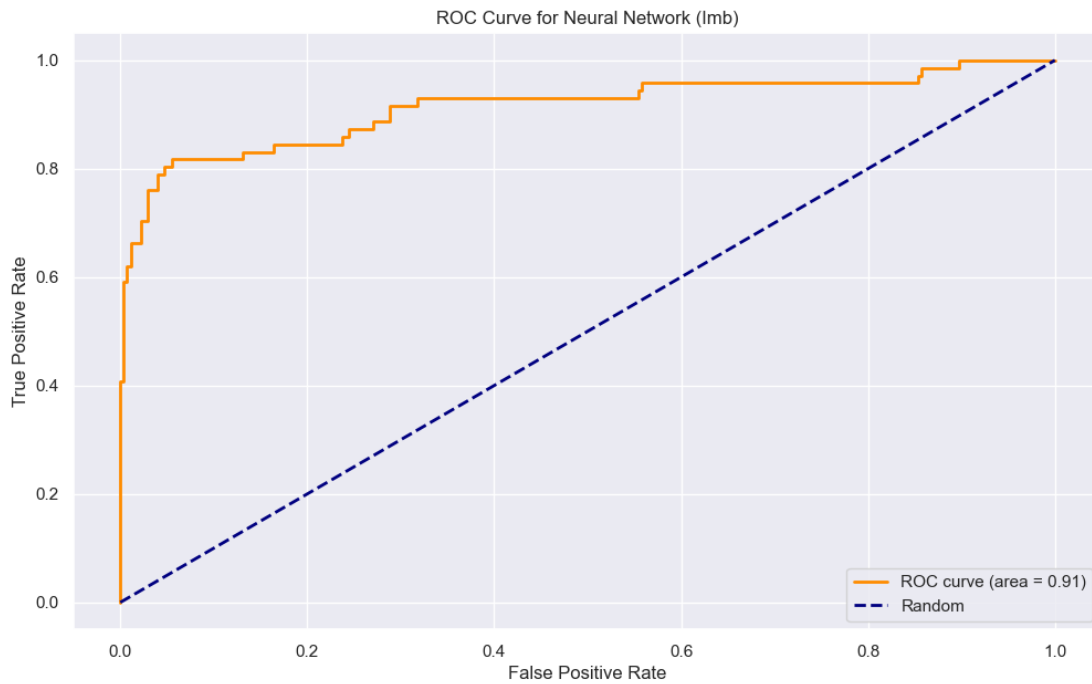


Figure 39: ROC Curve for Imb Neural Network Prediction

In this context, the high AUC signifies a strong capacity to rank positive instances higher than negative ones, demonstrating the model's proficiency in capturing intricate patterns and making accurate predictions. This result indicates not only the effectiveness of the neural network in differentiating between the two classes but also its potential for practical applications where the correct identification of positive instances holds significant importance. The 91% AUC reflected a Neural Network model that excels in both sensitivity and specificity, instilling confidence in its reliability for tasks requiring precise binary classification.

5.3 Sampling the Data-Set

We have already seen in the section 4.4 that our data-set is not balanced. Therefore, sampling of data in the context of imbalanced datasets is a critical aspect of addressing the skewed distribution of class instances, where one class significantly outnumbers the other. Traditional machine learning models tend to be biased towards the majority class, leading to suboptimal performance in accurately predicting the minority class. To mitigate this imbalance, various sampling techniques are employed. One common approach is oversampling the minority class, where instances from the minority class are duplicated or synthetically generated to balance the class distribution. This ensures that the model is exposed to a more equitable representation of both classes during training, preventing it from favoring the majority class. Alternatively, undersampling the majority class involves randomly removing instances from the majority class to create a more balanced dataset. While these techniques address class imbalance, they come with trade-offs. Oversampling can lead to overfitting, especially if the synthetic instances introduce noise, while undersampling risks losing potentially valuable information from the majority class.

Furthermore, advanced sampling methods, such as SMOTE (Synthetic Minority Over-sampling Technique), create synthetic instances for the minority class by interpolating between existing instances. SMOTE helps overcome the limitations of simple oversampling by introducing diversity in the synthetic samples. It's essential to carefully choose the appropriate sampling strategy based on the dataset's characteristics and the specific requirements of the classification task. Evaluating model performance using metrics like precision, recall, F1 score, and area under the ROC curve on both the training and validation sets helps assess the effectiveness of the chosen sampling approach. Striking the right balance between addressing class imbalance and avoiding potential pitfalls is crucial to ensuring the robustness and generalization of machine learning models on imbalanced datasets.

We decided to over-sample the data to deal with imbalanced class distribution. The decision to opt for oversampling of the minority class in the face of a substantially small dataset reflects a strategic approach to mitigating the challenges posed by class imbalance while preserving the available data samples. In scenarios where the minority class is underrepresented and limited in size, oversampling becomes an imperative technique to rectify the imbalance and enhance the model's ability to learn from instances belonging to the minority class.

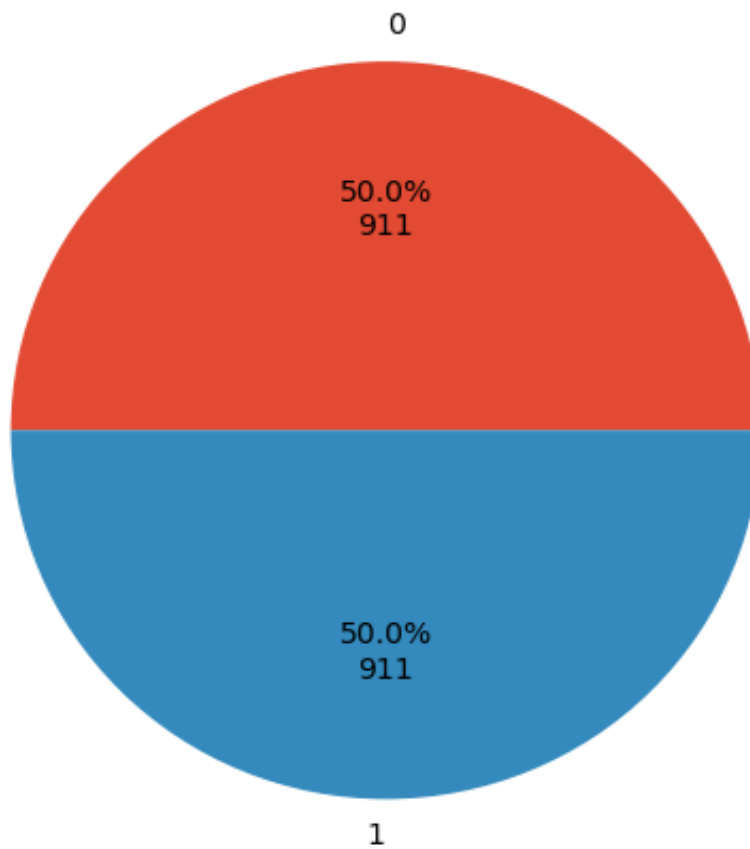


Figure 40: Class Distribution after Over Sampling

By duplicating or synthetically generating samples from the minority class, over-sampling ensures a more equitable representation during model training, preventing biases towards the majority class. However, the decision to choose oversampling over

other techniques, such as undersampling or more complex methods like SMOTE, is often influenced by the need to conserve as much valuable information as possible from the original dataset. Since the dataset is already constrained by its small size, oversampling 2 strikes a balance by addressing class imbalance without sacrificing an undue proportion of the limited available samples. This approach acknowledges the trade-off between addressing imbalance and retaining the entirety of the available data, with the overarching goal of bolstering the model's performance in scenarios where the minority class is crucial but underrepresented.

The transformation achieved through oversampling is vividly depicted in the pie chart 40, where the once skewed distribution has now been harmonized, resulting in an equal representation of both classes. We can see that class 1 has 911 samples where Clutch Failure happened and also 911 samples for class 0 where the clutch was healthy and failure didn't appear. This balanced presentation signifies the successful mitigation of class imbalance, a critical preprocessing step in machine learning. The pie chart's symmetry underscores the deliberate efforts to enhance the model's capability to discern and accurately predict instances from both classes. With a more equitable class distribution, the machine learning model is poised to make informed and unbiased predictions, promoting fairness and robustness in its classification task. The visual manifestation of equal proportions in the pie chart encapsulates the positive impact of oversampling, illustrating a dataset primed for training models that are sensitive to the intricacies of both classes, ultimately contributing to potentially improved overall performance and predictive accuracy.

5.4 Result: Balanced Data-set

With the rebalanced dataset achieved through oversampling, an extensive evaluation of various supervised machine learning algorithms has been conducted to gauge their performance in the context of binary classification. Employing metrics such as accuracy, precision, recall, and F1 score, the models were rigorously assessed to capture nuanced aspects of their predictive capabilities. Each algorithm, ranging from traditional models like logistic regression to more complex ones such as random forest and neural networks, underwent a comprehensive scrutiny.

Our findings revealed in later sections the diverse strengths and weaknesses of each algorithm trained on the balanced data-set. Logistic regression in section 5.4.1, known for its simplicity, showcased commendable recall, while decision tree in section 5.4.2 models demonstrated robust performance too. Random forest in section 5.4.4, with its ensemble nature, exhibited a balanced performance across multiple metrics. Support Vector Machines (SVMs) in section 5.4.3 demonstrated high precision but relatively lower recall. The neural network in section 5.4.5, being inherently flexible, displayed competitive results in accuracy and F1 score. This multifaceted evaluation enables a nuanced understanding of how each algorithm responds to the intricacies of the rebalanced dataset. It serves as a foundation for informed decisions regarding the selection of the most suitable model for the specific requirements of the binary classification task, considering factors like the relative importance of precision and recall based on the domain and application context. The thorough examination of these diverse models not only facilitates optimal model selection but also provides insights into potential areas for further refinement and enhancement in predictive performance.

5.4.1 Result Balnc Data: Logistic Regression

Starting with the upper-left quadrant (192), these instances represent True Negatives (TN), signifying the Logistic Regression model’s accurate classification of instances as belonging to the negative class. In this scenario, these instances likely represent cases where the logistic regression model correctly identified outcomes as benign or falling into the absence of the targeted condition. The bottom-right quadrant (222) reflects True Positives (TP), indicating instances accurately classified as belonging to the positive class. These instances illustrate the model’s capacity to correctly detect the presence or manifestation of the targeted condition.

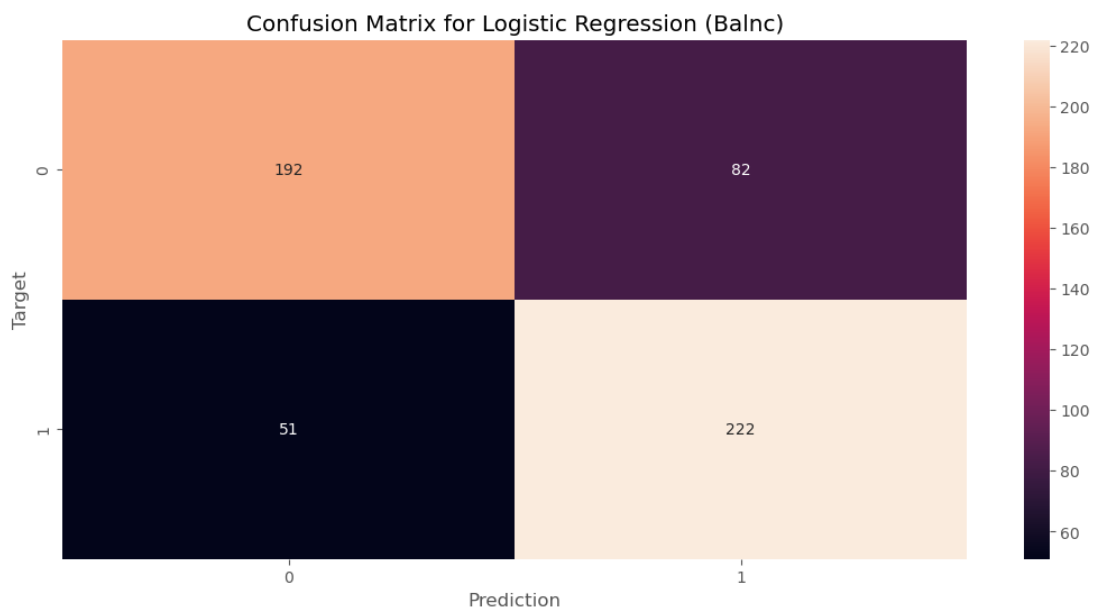


Figure 41: Confusion Matrix for Balanced Logistic Regression Prediction

Conversely, the upper-right quadrant (82) denotes False Positives (FP), where instances from the negative class were inaccurately predicted as positive. This introduces misclassification, highlighting areas where the logistic regression model has falsely identified instances as positive. Similarly, the bottom-left quadrant (51) signifies False Negatives (FN), indicating instances from the positive class that were erroneously classified as negative. This suggests instances where the logistic regression model failed

to detect the presence of the targeted condition. It is safe to say that this very basic ML model still achieved better results on the balanced data set as opposed to the imbalanced data set. Derived from these elements in 41, several performance metrics can be computed. Refer to table 9:

Index	Metric	Score
0	Accuracy	0.7545
1	Precision	0.7306
2	Recall	0.8135
3	F1 Score	0.7796

Table 9: Evaluation Metrics for Logistic Regression with Balanced Data

This comprehensive evaluation table 9 offers insights into the logistic regression model’s effectiveness in binary classification, considering both correct and incorrect predictions and providing a nuanced understanding of its precision, recall, and overall predictive accuracy.

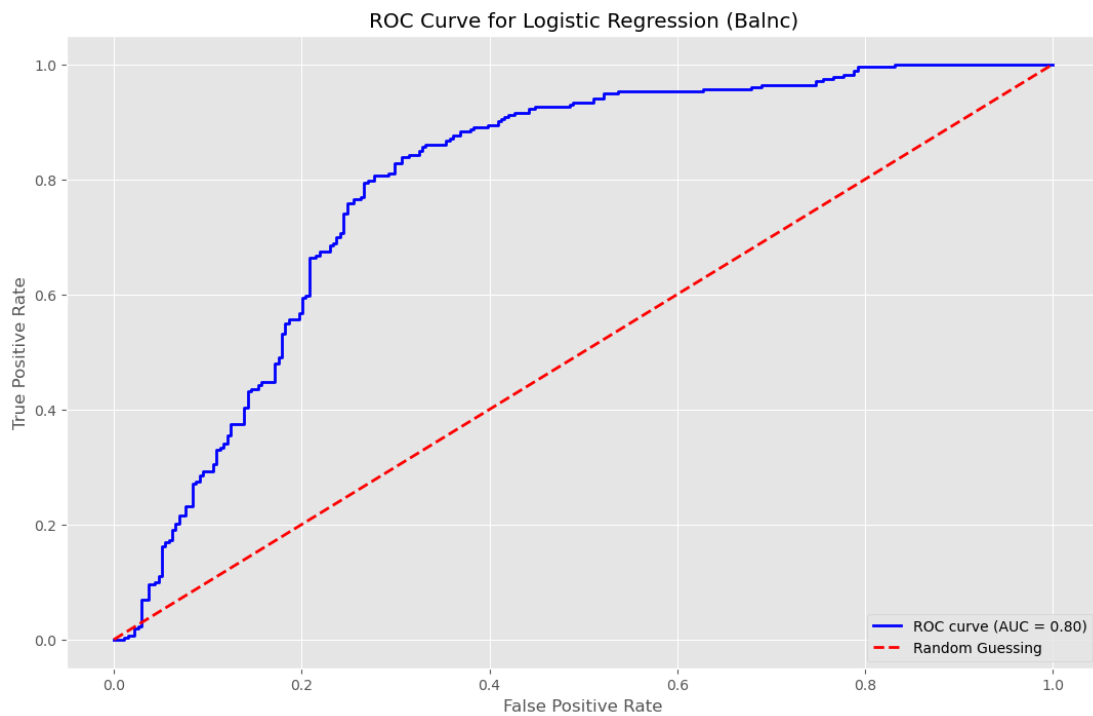


Figure 42: ROC Curve for Balanced data Logistic Regression

Overall, an AUC of 80% signifies a logistic regression model with substantial discriminatory capability and suggests its potential utility in scenarios requiring accurate and balanced predictions between two classes. The AUC for imbalanced data was reported to be 74%. It is safe to say that the balanced data is effective in delivering better results.

5.4.2 Result Balnc Data: Decision Tree

Starting with the upper-left quadrant (244), these instances signify True Negatives (TN), representing cases where the decision tree accurately classified instances as belonging to the negative class. These instances are indicative of situations where the model correctly identified outcomes as benign or not falling within the targeted condition. Moving to the bottom-right quadrant (246), we find True Positives (TP), indicating instances accurately classified as belonging to the positive class. These instances showcase the decision tree's proficiency in correctly detecting the presence or manifestation of the targeted condition.

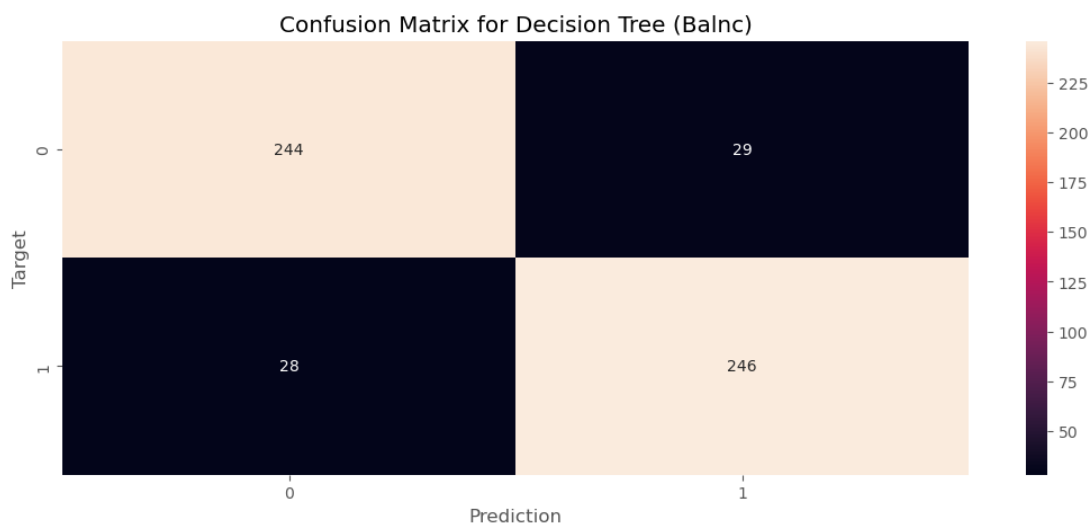


Figure 43: Confusion Matrix for Balanced Decision Tree Prediction

On the contrary, the upper-right quadrant (29) denotes False Positives (FP), where instances from the negative class were inaccurately predicted as positive. This intro-

duces the potential for misclassification, highlighting areas where the decision tree model has falsely identified instances as positive. Similarly, the bottom-left quadrant (28) signifies False Negatives (FN), indicating instances from the positive class that were classified as negative. This suggests instances where the decision tree model failed to detect the presence of the targeted condition. Overall, the performance of the Decision tree stands out better than logistic regression.

In the previous section, we showed the result of the Decision tree while using the imbalanced data set. The reported AUC was 75%. While on the balanced data, we have achieved AUC of 90%.

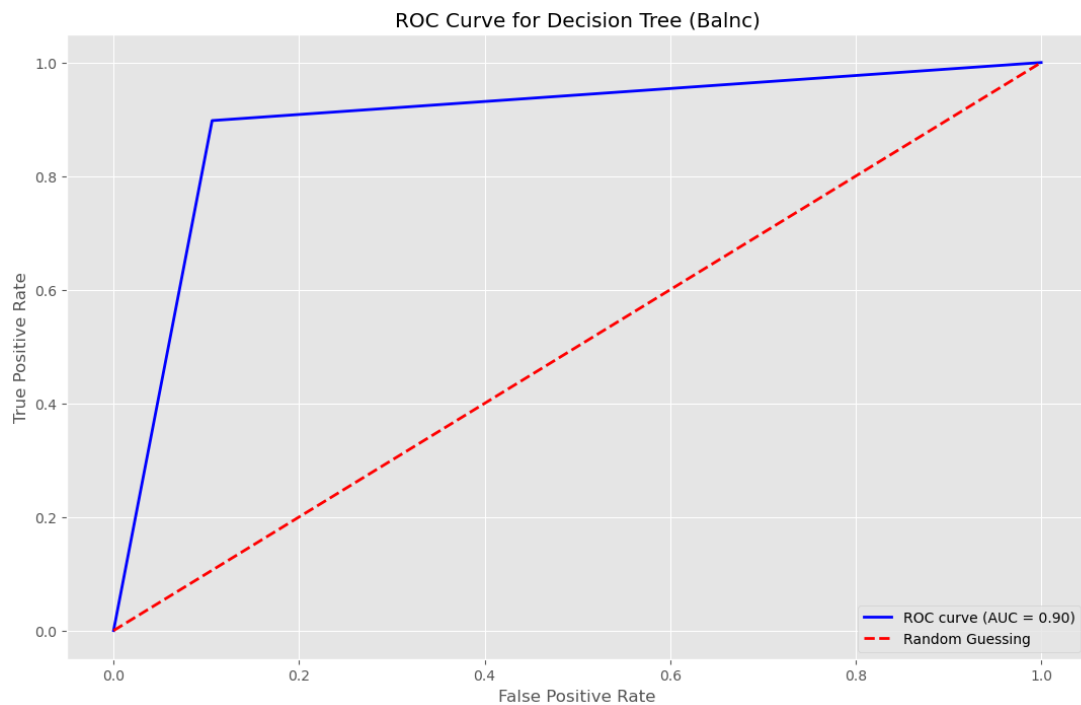


Figure 44: ROC Curve for Balanced data Decision Tree

Since the precision, recall and other metrics have scored similar accuracy points, the plot 44 above showed the ROC curve at 89% and does not follow similar depiction as all other models.

Index	Metric	Score
0	Accuracy	0.895795
1	Precision	0.894545
2	Recall	0.897810
3	F1 Score	0.896175

Table 10: Evaluation Metrics for Decision Tree with Balanced Data

5.4.3 Result Balnc Data: Support Vector Machine (SVM)

For Support Vector Machine (SVM), the presented confusion matrix unveils distinctive aspects of the model's predictive performance. In the upper-left quadrant, the count of 290 signifies True Negatives (TN), highlighting the instances correctly identified as part of the negative class. These instances represent scenarios where the SVM accurately discerned outcomes as lacking the characteristic targeted in the classification task. Conversely, in the lower-right quadrant, the count of 251 reveals True Positives (TP), showcasing the SVM's effectiveness in correctly recognizing instances belonging to the positive class. These instances underscore the SVM's capability in successfully detecting the presence or manifestation of the characteristic under consideration.

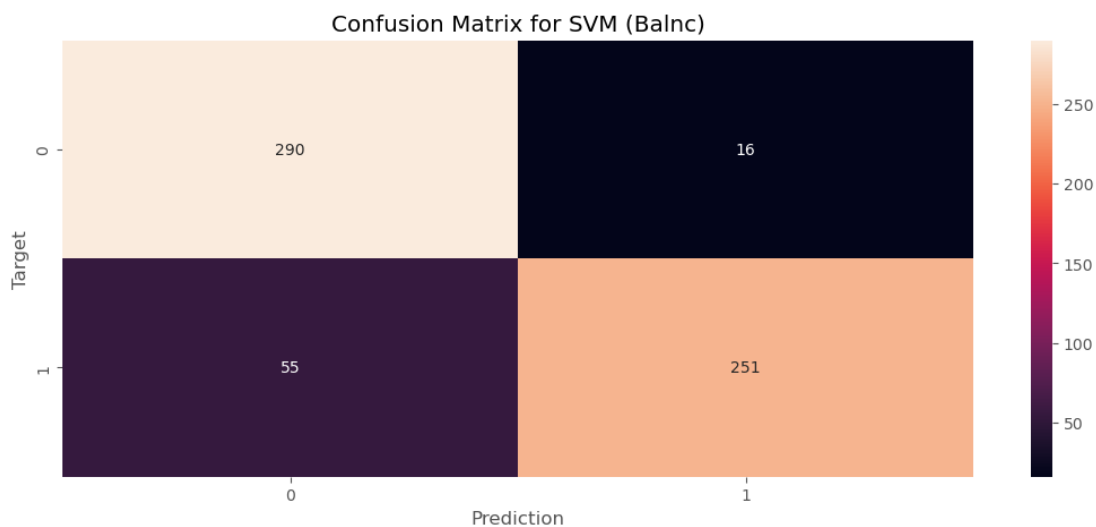


Figure 45: Confusion Matrix for Balanced Data SVM Prediction

Moving to the upper-right quadrant, the count of 16 designates instances character-

ized as False Positives (FP), where the SVM erroneously predicted negative-class instances as positive. This introduces the possibility of misclassification, indicating areas where the SVM may have inaccurately identified instances as positive. Simultaneously, in the lower-left quadrant, the count of 55 represents False Negatives (FN), signifying instances from the positive class that the SVM misclassified as negative. These instances reflect situations where the SVM encountered challenges in capturing the presence of the targeted characteristic. In summary, this comprehensive evaluation results in an accuracy of approximately 89%, precision of 94%, recall of 82%, and an F1 Score of 87%. These metrics collectively provide a nuanced understanding of the SVM's performance in binary classification, shedding light on its strengths and areas for potential improvement in predictive accuracy.

Index	Metric	Score
0	Accuracy	0.883987
1	Precision	0.940075
2	Recall	0.820261
3	F1 Score	0.876091

Table 11: Evaluation Metrics for SVM with Balanced Data

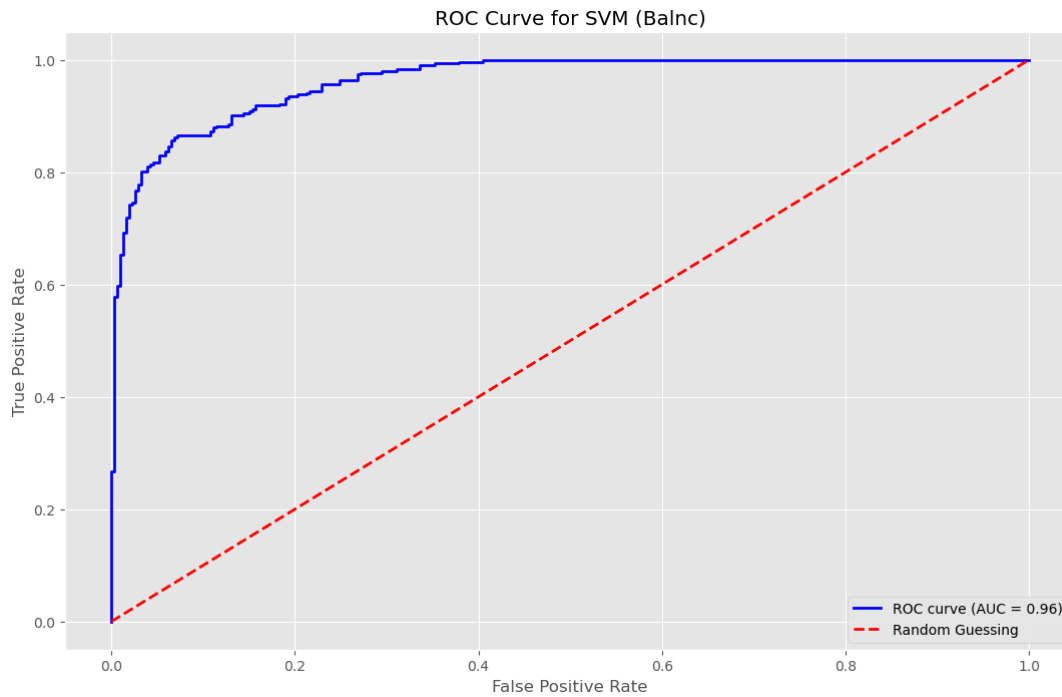


Figure 46: ROC Curve for Balanced data SVM

The AUC score for SVM trained on the balanced data set was recorded to be 96%. While the AUC stands at 89% when trained on the imbalanced data set. Overall, an AUC of 96% signifies that the SVM model has substantial discriminatory capability and suggests its potential utility in scenarios requiring accurate and balanced predictions between two classes.

5.4.4 Result Balnc Data: Random Forest

The presented confusion matrix reflects the outcomes of a random forest model in the domain of binary classification, exhibiting a matrix structure with distinct elements. In the upper-left quadrant, the count of 249 signifies True Negatives (TN), denoting instances accurately classified as part of the negative class. This outcome highlights the model's adeptness in correctly discerning cases absent of the characteristic targeted by the classification task. Conversely, the lower-right quadrant exhibits a count of 261, indicative of True Positives (TP). This element showcases the model's proficiency in accurately identifying instances belonging to the positive class, effectively capturing the presence or manifestation of the targeted characteristic.

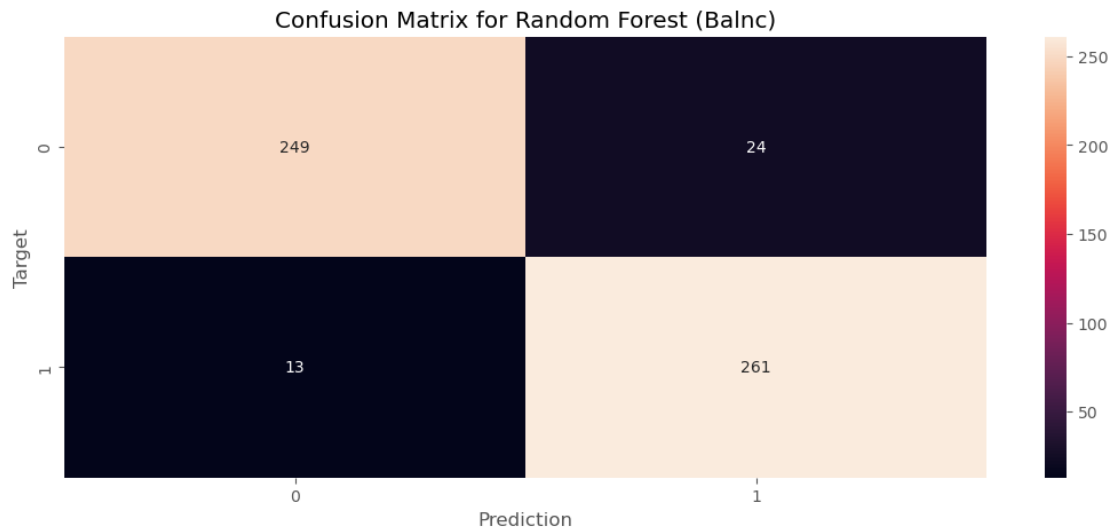


Figure 47: Confusion Matrix for Balanced Data Random Forest Prediction

On the upper-right side, the count of 24 represents False Positives (FP), signifying instances where the model incorrectly predicted negative-class instances as positive. This introduces the prospect of misclassification, delineating areas where the random forest model has identified instances as positive. Simultaneously, the lower-left quadrant discloses a count of 13, symbolizing False Negatives (FN). These instances represent scenarios where the model misclassified instances from the positive class as negative, indicating challenges in capturing the presence of the targeted characteristic.

In a quantitative assessment, the model achieved an accuracy of approximately 93%, precision of 91%, recall of 95%, and an F1 Score of 93%. These metrics collectively provide a nuanced understanding of the random forest model's binary classification performance, emphasizing its strengths in correctly identifying both positive and negative instances while acknowledging potential areas for refinement in minimizing false positives and false negatives.

Index	Metric	Score
0	Accuracy	0.932358
1	Precision	0.915789
2	Recall	0.952555
3	F1 Score	0.933810

Table 12: Evaluation Metrics for Random Forest with Balanced Data

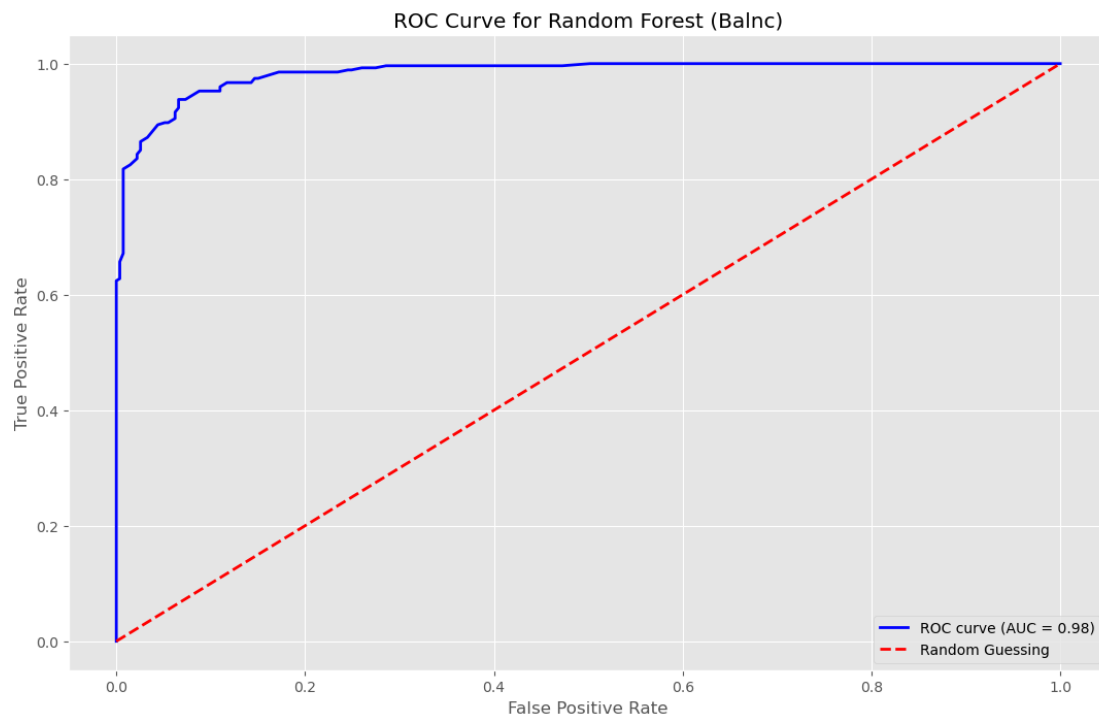


Figure 48: ROC Curve for Balanced data Random Forest

The Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) for the

random forest model stands at an impressive 98%, denoting its exceptional ability to discriminate between positive and negative instances in a binary classification scenario. This metric encapsulates the model's capacity to rank true positives higher than false positives across various classification thresholds. The AUC value near 1 signifies a minimal overlap between the distributions of positive and negative instances, emphasizing the model's robustness in distinguishing between the two classes. The high AUC score underscores the random forest's proficiency in achieving a balance between sensitivity and specificity, showcasing its potential for reliable predictions in scenarios where accurate discrimination between classes is paramount.

5.4.5 Result Balnc Data: Artificial Neural Network

Neural Network performance depends on many factors. Such as, the architecture of the neural network, the parameter, and hyper-parameter tuning, and the size of the training data. Neural Networks are known to be a good choice if the data is not so small. However, in our case, we had a relatively small data set to train. The figure 49 depicts the training and validation accuracy.

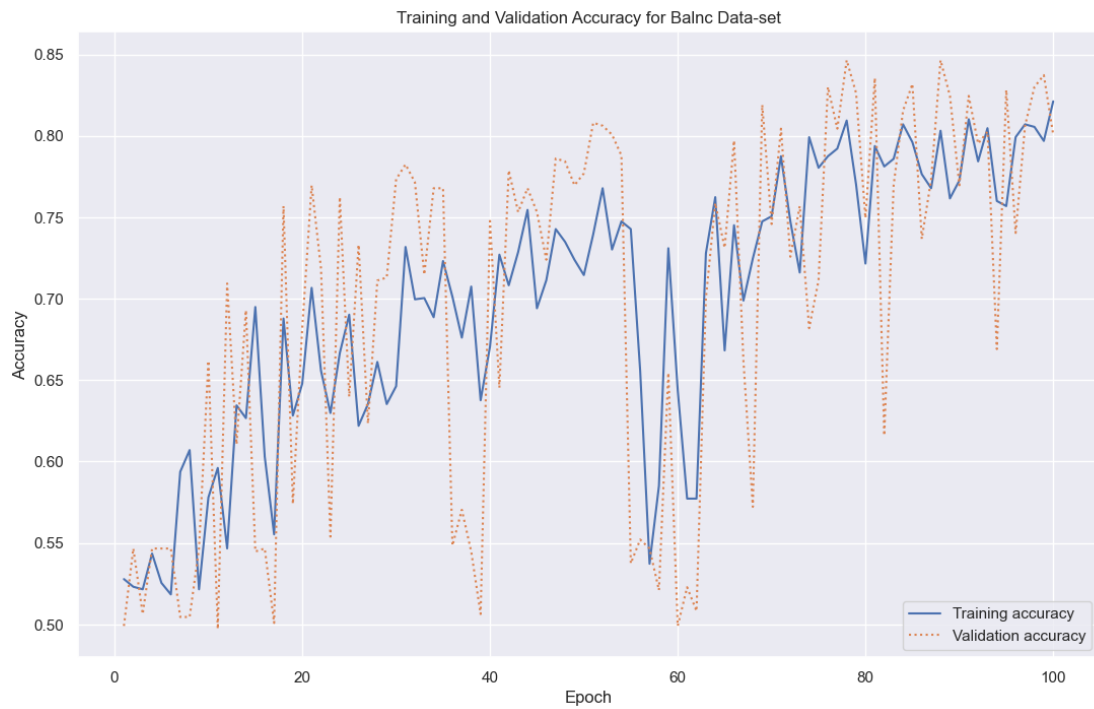


Figure 49: Neural Network Training and Validation Accuracy for Balnc Data-set

The fig 49 revealed a scenario where the achieved training and validation accuracies do not exhibit a considerable level of proficiency. Despite the intricate architecture and adaptability inherent to neural networks, the model's training accuracy falls short of reaching a level of excellence. Additionally, the validation accuracy mirrors this trend, indicating limitations in the model's ability to generalize well to unseen data. The challenges in achieving robust accuracy metrics could potentially be attributed to a constrained dataset, where the neural network may struggle to learn diverse and representative patterns due to limited examples. The relatively small size of the dataset

might impede the model's capacity to discern intricate relationships within the data, leading to suboptimal performance. This underscores the significance of dataset size and diversity in training neural networks, necessitating careful consideration and potential augmentation strategies to enhance the model's learning capabilities and improve its binary classification performance.

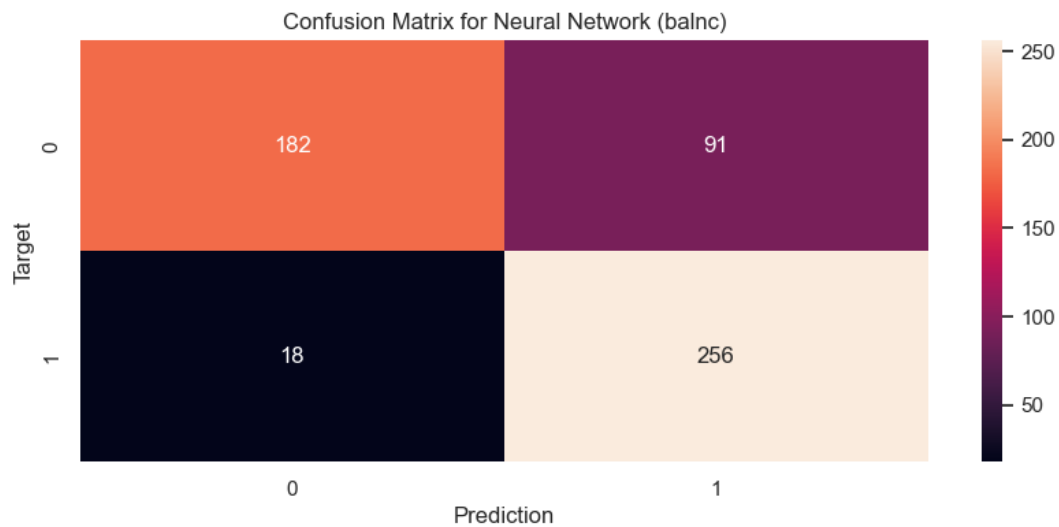


Figure 50: Confusion Matrix of Balanced Data for Neural Network Prediction

The upper-left quadrant (182) represents instances correctly identified as part of the negative class, termed True Negatives (TN). This suggests the Neural Network model accurately recognized cases where the condition being investigated was absent. On the contrary, the lower-right quadrant (256) signifies True Positives (TP), indicating instances correctly classified as part of the positive class, where the condition was indeed present. Moving to the upper-right quadrant (91), these instances denote False Positives (FP), where the model incorrectly predicted negative-class instances as positive. Lastly, the lower-left quadrant (18) represents False Negatives (FN), signifying instances from the positive class that the model erroneously classified as negative. In simple terms, the model exhibited strengths in correctly identifying absence but faced challenges in capturing instances where the condition was present, leading to a balance between accurate and misclassified predictions.

When training a neural network on a balanced dataset, the achieved Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) of 92% reflects the model's commendable ability to distinguish between positive and negative instances. However, a marginal reduction to 91% AUC is observed when the same neural network is trained on an imbalanced dataset. Several factors could contribute to this nuanced behavior. Firstly, in a balanced dataset, the model encounters an equal representation of both classes, fostering a more comprehensive understanding of the underlying patterns associated with each category. Conversely, when trained on an imbalanced dataset, where one class is underrepresented, the neural network might exhibit a bias toward the majority class, impacting its discriminatory performance.

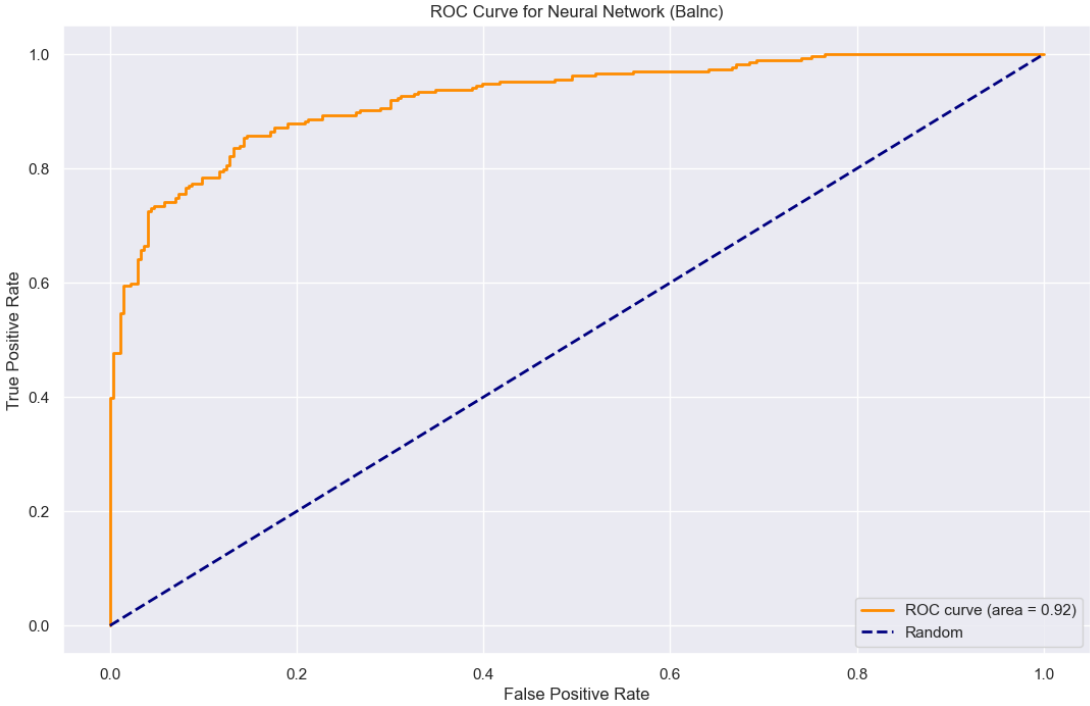


Figure 51: ROC Curve for Balanced data Neural Network

Additionally, the limited instances of the minority class in the imbalanced dataset may result in the neural network struggling to generalize well to such instances during training, leading to a subtle reduction in AUC. The intricate interplay between class distribution and the neural network's learning dynamics emphasizes the importance of

balanced datasets for optimal model performance, as imbalances can introduce challenges in capturing the intricacies of the underrepresented class.

Chapter 6

Discussions & Conclusion

6.1 Discussions

The examination of machine learning model results before and after oversampling presents a compelling narrative on the impact of addressing class imbalance. Before oversampling, the models encountered challenges in effectively learning from the minority class due to its limited representation. This deficiency often translated into imbalanced performance metrics, particularly lower recall and sensitivity. However, after implementing oversampling techniques, which involve synthetically increasing instances of the minority class, a notable transformation is observed. The models exhibit improved balance in their predictive capabilities, demonstrating enhanced sensitivity and recall, crucial for correctly identifying instances of the minority class. This rebalancing act contributes to a more comprehensive and equitable evaluation of the models, with metrics like accuracy, precision, recall, and F1 score reflecting a more accurate depiction of their overall performance.

6.1.1 Result Interpretation: Imbalance Data

The interpretability of each model's behavior, both before and after oversampling, underscores the pivotal role that addressing class imbalance plays in refining the predictive capacity of machine learning models, particularly in scenarios where minority class instances are pivotal yet underrepresented.

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.800000	0.583333	0.098592	0.168675
Decision Tree	0.826087	0.582090	0.549296	0.565217
SVM	0.894180	0.847826	0.541667	0.661017
Random Forest	0.886957	0.863636	0.535211	0.660870
Neural Network	0.921341	0.891922	0.690123	0.777892

Table 13: Evaluation Metrics for all models with **imbalanced Data**

The presented table outlines the performance metrics of different machine learning models employed in a binary classification task. Each model is evaluated based on key metrics such as accuracy, precision, recall, and F1 Score, providing a comprehensive assessment of their predictive capabilities. The Logistic Regression model (Model 1) demonstrates an accuracy of 80%, indicating a reasonable overall correctness of predictions. However, the low recall of 9.86% suggests challenges in effectively capturing positive instances, potentially resulting in a higher number of false negatives. The Decision Tree (Model 2) exhibits a slightly higher accuracy of 82.61% with comparable precision, recall, and F1 Score, reflecting a balanced performance in terms of correctly identifying positive and negative instances.

Moving to the SVM (Model 3), the model achieves an accuracy of 89.42%, showcasing robust overall correctness. The precision of 84.78% indicates a high accuracy of positive predictions, while the recall of 54.17% suggests the model's struggle in capturing all positive instances, leading to a trade-off between precision and recall. Similarly, the Random Forest (Model 4) attains an accuracy of 88.70%, with a well-balanced precision and recall, striking a compromise between accurately predicting positive instances and capturing a substantial portion of them.

The Neural Network (Model 5) emerges as the top performer with an accuracy of 92.13%. This model demonstrates high precision (89.19%) and recall (69.01%), leading to a well-balanced F1 Score of 77.79%. The neural network excels in both accurate positive predictions and capturing a significant proportion of positive instances, making it a strong candidate for scenarios where a balance between precision and recall is

crucial. In conclusion, the analysis of the model performances reveals nuanced trade-offs between accuracy, precision, and recall. The choice of the most suitable model depends on the specific requirements and priorities of the classification task, considering factors such as the consequences of false positives and false negatives in the given context. However, Neural Network outperformed all the other models in all the metrics on imbalanced data set.

6.1.2 Result Interpretation: Balanced Data

Interpreting machine learning model results on a balanced dataset that has been oversampled provides valuable insights into the impact of addressing class imbalance. The process of oversampling involves synthetically increasing instances of the minority class, thereby mitigating the effects of an imbalanced distribution. In this context, the models exhibit improved overall accuracy, as they are now better equipped to handle the intricacies of both classes. Precision and recall metrics also benefit from the balanced dataset, with models demonstrating heightened accuracy in positive predictions and a more comprehensive ability to capture instances of the minority class. The enhanced F1 Score reflects a harmonious balance between precision and recall, showcasing the models' effectiveness in achieving a nuanced equilibrium. Overall, the interpretation underscores the significance of addressing class imbalance through oversampling, leading to more robust and reliable machine learning models in scenarios where accurate representation of both classes is imperative.

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.7545	0.7306	0.8135	0.7796
Decision Tree	0.895795	0.894545	0.897810	0.896175
SVM	0.883987	0.940075	0.820261	0.876091
Random Forest	0.932358	0.915789	0.952555	0.933810
Neural Network	0.800412	0.737234	0.934432	0.825542

Table 14: Evaluation Metrics for all models with **Balanced Data**

The table 14 provides valuable insights into the performance of various machine learning models in a classification task on a Balanced Data Set. The Logistic Regression model demonstrates a modest accuracy of 75.45%, showcasing a reasonable capability

in making correct predictions. In contrast, the Decision Tree model stands out with an accuracy of 89.58%, reflecting its robust ability to capture complex patterns within the data. The SVM model achieves an accuracy of 88.40%, combining high precision (94.01%) with a respectable recall (82.03%), indicating its proficiency in accurate positive predictions while maintaining sensitivity to actual positive instances.

The Random Forest model emerges as a top performer with an accuracy of 93.24%, demonstrating a well-balanced precision (91.58%) and recall (95.26%). The Neural Network, while displaying a lower accuracy of 80.04%, exhibits notable recall (93.44%), emphasizing its strength in capturing positive instances. Overall, these findings underscore the diversity in model performances, emphasizing the importance of considering various metrics such as precision, recall, and F1 Score to comprehensively evaluate their effectiveness in different aspects of the classification task.

6.2 Limitations

In the realm of automotive research and data analysis, ensuring the accuracy and integrity of collected data presents a multifaceted challenge. A variety of factors contribute to the complexity of this task, each of which must be meticulously managed to produce reliable and meaningful insights. Key among these challenges are sensor errors, data entry mistakes, and inconsistencies in measurement methods, all of which can compromise the validity of the data.

6.2.1 Sensor Errors and Data Entry Mistakes

Modern vehicles are equipped with a plethora of sensors that collect data on a range of parameters, from engine performance to driver behavior. However, these sensors are not infallible and can suffer from calibration issues, wear and tear, or even software malfunctions, leading to erroneous data. Similarly, human errors during data entry—whether during the initial recording or subsequent transcription—can introduce inaccuracies that skew results. Such errors are particularly problematic in large datasets where manual review of every entry is impractical.

6.2.2 Inconsistencies in Measurement Methods

Another significant challenge arises from inconsistencies in measurement methods across different data sources or time periods. For example, the adoption of new sensor technologies or changes in data collection protocols can lead to discrepancies that complicate data analysis. This is especially problematic in longitudinal studies where consistency over time is crucial for drawing valid conclusions. Without standardized methods and protocols, comparing data across different datasets can lead to misleading or inconclusive results.

6.2.3 Data Fragmentation and Access Restrictions

The fragmentation of data across various departments, organizations, or even geographic regions further complicates data consolidation and analysis. In the automotive industry, data may be siloed within specific departments, such as manufacturing, sales, or customer service, each using its own systems and standards. Moreover, access to proprietary data from automotive manufacturers or suppliers is often restricted due to intellectual property concerns or competitive considerations. Even when access is granted, it typically requires navigating complex legal agreements, which can delay or limit research efforts.

6.2.4 Data Privacy Regulations

Beyond technical and logistical challenges, automotive data collection and analysis must also navigate the complex landscape of data privacy regulations, ethical considerations, and potential biases that can undermine the generalizability of research findings. With the advent of stringent data privacy regulations, such as the General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA) in the United States, researchers must exercise extreme caution in handling personal and sensitive data. These regulations impose strict requirements on how data is collected, stored, and shared, often necessitating the anonymization of datasets to protect individual privacy. However, anonymization can sometimes strip the data of critical context, reducing its utility for in-depth analysis.

6.2.5 Ethical Considerations

Collecting and analyzing sensitive data—such as customer information, vehicle usage patterns, or biometric data—requires careful consideration of the ethical implications. Researchers must ensure that their work does not inadvertently harm individuals or groups, whether through the misuse of data, unintended biases in analysis, or breaches of confidentiality. Ethical guidelines and oversight mechanisms are essential to prevent such outcomes, but they can also add layers of complexity and delay to research projects.

6.2.6 Bias and Representativeness

Finally, biases in the data, whether due to regional disparities, market-specific trends, or variations in driver behavior, can limit the generalizability of research findings. For instance, data collected predominantly from urban areas may not accurately reflect driving patterns in rural regions, leading to skewed results. Similarly, the diversity of vehicle types, models, and technologies poses a significant challenge in collecting representative data. Different vehicles may be equipped with varying sensors or may be driven in different ways, further complicating efforts to standardize and compare data across the automotive landscape.

In sum, the accuracy and integrity of automotive data are influenced by a multitude of factors, ranging from technical issues and logistical challenges to regulatory, ethical, and bias-related concerns. Addressing these challenges requires a concerted effort to standardize data collection methods, ensure data privacy and ethical integrity, and mitigate biases to produce research findings that are both reliable and broadly applicable.

6.3 Research Implications

6.3.1 Research Implications for Industries

Predictive maintenance, a data-driven approach to maintenance, has emerged as a powerful tool for industries seeking to reduce downtime and increase overall equipment effectiveness (OEE). By leveraging advanced analytics and real-time data, predictive

maintenance enables organizations to anticipate equipment failures before they occur, thereby minimizing unplanned downtime and optimizing maintenance schedules.

Predictive maintenance involves the following steps: gathering real-time data from sensors and other monitoring devices installed on equipment, applying advanced analytics techniques to analyze the collected data and identify patterns or anomalies that indicate potential equipment failures, continuously monitoring equipment health based on the analyzed data and setting thresholds for critical parameters, developing predictive models that can forecast equipment failures based on historical data and current trends, and using the predictive models to schedule maintenance tasks proactively, addressing potential failures before they lead to unplanned downtime.

The benefits of predictive maintenance include reduced downtime, increased OEE, improved asset management, cost savings, and enhanced safety. Predictive maintenance is applicable across a wide range of industries, including manufacturing, energy, transportation, healthcare, and oil and gas.

By leveraging data-driven insights and advanced analytics, organizations can proactively address potential equipment failures, optimize maintenance schedules, and maximize the value of their assets.

6.3.2 Research Implications for Academia

Predictive maintenance, a data-driven approach to maintenance, has emerged as a significant area of research in academia. Its applications across various industries, from manufacturing to healthcare, have fueled interest in understanding its potential and limitations. This section delves into the key research implications for predictive maintenance in academia.

The development of novel machine learning algorithms, deep learning techniques, and statistical models is crucial for accurately predicting equipment failures. Research into data quality assessment, cleaning, and feature engineering techniques is necessary to ensure the reliability of predictive models. Investigating efficient methods for processing and analyzing large volumes of real-time data from sensors and other monitoring devices is also essential.

Developing probabilistic models to quantify uncertainty in predictions and improve the reliability of maintenance decisions is another important research direction. Study-

ing the sensitivity of predictive models to different input variables and identifying critical factors affecting equipment health is also crucial. Researching methods to calculate confidence intervals for predictions, providing a measure of uncertainty associated with maintenance recommendations, is another area of interest.

Developing techniques to make predictive maintenance models more transparent and understandable to human operators, facilitating trust and adoption, is essential. Designing interactive tools that integrate predictive maintenance insights with human expertise to enable informed decision-making is also crucial. Investigating how humans and machines can effectively collaborate in maintenance tasks, leveraging the strengths of both, is another area of interest.

Conducting case studies and field experiments to evaluate the effectiveness of predictive maintenance in various industrial settings is essential. Assessing the economic benefits of predictive maintenance, including reduced downtime, improved asset utilization, and cost savings, is also crucial. Investigating the scalability and generalizability of predictive maintenance models across different equipment types and industries is another area of interest.

By addressing these research implications, academia can contribute significantly to the advancement of predictive maintenance and its widespread adoption across various industries. Future research efforts should focus on developing innovative methodologies, addressing practical challenges, and ensuring the ethical and responsible application of predictive maintenance technologies.

6.4 Future Work

The field of clutch damage prediction in commercial vehicles using Machine Learning (ML) presents numerous exciting avenues for future research and development. With the automotive industry increasingly transitioning toward data-driven solutions for predictive maintenance, ML offers immense potential for improving the accuracy and timeliness of clutch damage detection. However, to fully harness the power of these technologies, several challenges need to be addressed and opportunities explored.

6.4.1 Expansion of Data Sources and Quality

One of the key areas for future work is the expansion and enhancement of data sources. Current clutch damage prediction models typically rely on sensor data from vehicle telematics, including parameters such as engine torque, clutch engagement times, and vehicle speed. However, the integration of additional data sources, such as vibration analysis, thermal imaging, and acoustic emissions, could greatly improve the robustness and precision of these models. High-quality data from these diverse sources would allow for a more comprehensive understanding of the conditions that lead to clutch damage. Additionally, the introduction of standardized data formats across different vehicle models and manufacturers would facilitate better cross-industry collaboration and algorithm performance.

6.4.2 Advanced Feature Engineering and Selection

Another avenue for future research lies in improving feature engineering and selection techniques. Machine Learning algorithms, particularly deep learning models, require relevant and well-curated features to perform optimally. Future work could focus on the development of advanced techniques for identifying the most critical features related to clutch wear and tear. Additionally, real-time feature extraction and processing, supported by edge computing, could allow for faster and more accurate predictions directly within vehicles. This would lead to the immediate detection of clutch issues and prevent costly failures.

6.4.3 Utilization of Advanced Machine Learning Algorithms

Future work should also explore the application of more advanced ML algorithms, such as ensemble methods and reinforcement learning, to improve prediction accuracy. While current models often use traditional methods like Random Forests or Support Vector Machines, newer techniques such as Long Short-Term Memory (LSTM) networks or Convolutional Neural Networks (CNNs) can capture temporal dependencies and complex relationships in time-series data. Hybrid models, which combine the strengths of multiple algorithms, could be particularly effective for handling the intricacies of clutch damage patterns.

6.4.4 Predictive Maintenance Optimization

Predictive maintenance strategies leveraging ML should be further optimized. Current approaches primarily focus on detecting imminent failures, but future research could explore the optimization of maintenance schedules based on the severity and likelihood of damage. By refining these algorithms, it will be possible to provide more accurate maintenance intervals that balance the cost of early repairs with the risks of clutch failure. This could have a substantial impact on reducing downtime and maintenance costs for commercial vehicle fleets.

6.4.5 Integration of Real-Time Feedback Loops

The integration of real-time feedback loops into ML-based prediction models is another promising area for future work. By continuously updating the predictive model based on new data from vehicles in operation, the system can learn and adapt to changing conditions, improving accuracy over time. This continuous learning process could enable dynamic adjustments to the model based on different driving behaviors, environmental conditions, or variations in vehicle loads, thus improving the generalizability of the model across different scenarios.

6.4.6 Explainable AI and Model Interpretability

As machine learning models become more complex, the need for explainable AI (XAI) is crucial. Future work should focus on making ML models for clutch damage prediction more interpretable to technicians and engineers. This will ensure that predictions are trusted and actionable, enabling end-users to understand why a certain failure is predicted and how to prevent it. Techniques such as SHAP (Shapley Additive Explanations) values and LIME (Local Interpretable Model-agnostic Explanations) could be employed to provide more transparency.

6.4.7 Scalability and Deployment in Real-World Applications

While promising, many of the current clutch damage prediction models are still in the research phase or limited to small-scale testing. Future work should focus on the scalability and deployment of these models in real-world commercial vehicle fleets. This involves overcoming challenges related to the computational requirements of running ML models on limited hardware, such as in-vehicle embedded systems, and ensuring that predictions are delivered in real-time without sacrificing accuracy.

6.4.8 Collaborative Data Sharing and Model Standardization

Collaborative data sharing across manufacturers and fleet operators could lead to significant advancements in clutch damage prediction. The creation of a standardized, industry-wide dataset for clutch wear and failure could enhance model performance and generalizability. By pooling data from various sources, ML models could be trained on more diverse datasets, leading to improved predictions that work across different vehicle types and operational environments.

6.4.9 Ethical and Privacy Considerations

Lastly, as with any data-driven technology, ethical and privacy considerations must be addressed. Future work must explore ways to protect driver and fleet operator data while still enabling the benefits of ML-based predictive maintenance systems. This involves ensuring that data collection complies with privacy regulations and that predic-

tive systems are designed with transparency and fairness in mind.

The future of clutch damage prediction using Machine Learning is filled with opportunities for enhancing vehicle reliability, reducing operational costs, and minimizing downtime. By advancing the integration of diverse data sources, improving feature selection and model interpretability, and scaling the deployment of these systems, the automotive industry can realize the full potential of ML in predictive maintenance. Addressing these future work areas will bring about a new era of intelligent, data-driven fleet management and contribute significantly to the longevity and efficiency of commercial vehicles.

6.5 Conclusion

In conclusion, our analysis has successfully demonstrated the feasibility of predicting failures in vehicle clutches using machine learning models when provided with appropriate features and a substantial amount of data. The diverse set of algorithms employed in our study, ranging from Logistic Regression and Decision Trees to SVM, Random Forest, and Neural Networks, collectively showcase the potential of leveraging advanced analytics for predictive maintenance in automotive systems. The models, through their varying accuracies, precision, recall, and F1 Scores, collectively reinforce the notion that predictive modeling can contribute significantly to identifying potential failures in vehicle clutches. This not only aids in preemptive maintenance but also enhances operational efficiency and safety. The nuanced trade-offs observed in different models highlight the importance of tailoring the choice of algorithms based on the specific priorities and constraints of the application. Overall, this study marks a step forward in leveraging data-driven approaches to enhance the reliability and performance of automotive systems, particularly in anticipating and mitigating potential issues related to vehicle clutches.

The automotive industry stands to benefit significantly from deploying more test vehicles to gather high-quality data, despite the associated heavy costs for companies. As vehicles become increasingly complex, laden with advanced sensors and interconnected technologies, the amount and diversity of data generated during real-world testing become crucial for enhancing safety, performance, and overall reliability. Deploying an extensive fleet of test vehicles allows for the collection of diverse data sets under various driving conditions, enabling a more comprehensive understanding of system behaviors and potential failure modes. This data-driven approach not only facilitates the refinement of existing technologies but also supports the development of innovative features and the identification of potential issues before they escalate. While the upfront costs may seem substantial, the long-term benefits in terms of improved product quality, reduced warranty claims, and enhanced customer satisfaction can ultimately outweigh the initial investment. In an era where data-driven insights are pivotal for innovation and competitiveness, the strategic deployment of test vehicles becomes an invaluable in-

vestment for automotive companies striving to stay at the forefront of technological advancements and ensure the delivery of reliable and cutting-edge vehicles to consumers.

While over-sampling has proven effective in addressing class imbalance within datasets, industries should prioritize acquiring real-world data over relying solely on data augmentation techniques. While techniques like over-sampling can synthetically inflate minority class instances, they may not fully capture the nuanced variability and complexity present in authentic, on-the-road scenarios. Acquiring real-world data ensures that machine learning models are exposed to the diverse and dynamic conditions encountered during actual vehicle operation. Authentic data encompasses a broader spectrum of scenarios, driving patterns, and potential failure modes, providing a more accurate representation of the challenges faced by automotive systems. Emphasizing the collection of real-world data not only enhances the robustness of predictive models but also contributes to their adaptability in addressing unforeseen challenges and anomalies. Therefore, industries should strive for a balanced approach, incorporating both over-sampling techniques and a continuous effort to gather authentic, diverse data from real-world driving conditions to ensure the optimal performance and reliability of predictive models in the automotive domain.

The availability of more open-source datasets from the automotive industry stands as a catalyst for significant advancements in the digital field. OEMs and Automotive industries should find a secure way to share the data to enable advanced development. Currently, there are very few data-set open sourced. Open datasets, generously shared by automotive companies, provide researchers, engineers, and data scientists with invaluable resources to explore and innovate. The diversity of data, ranging from vehicle diagnostics to real-world driving scenarios, facilitates the development and refinement of machine learning models, predictive algorithms, and other digital solutions. These datasets not only contribute to the evolution of autonomous driving technologies but also fuel progress in vehicle safety, energy efficiency, and smart transportation systems. By fostering collaboration and knowledge-sharing, open-source datasets empower the digital community to collectively address challenges, test hypotheses, and propel the automotive industry toward a future marked by technological excellence, safety enhance-

ments, and sustainable innovation.

The comparison of model performances based on the result tables offers a nuanced perspective on the strengths and weaknesses of each algorithm in the classification task before and after the oversampling of the Data Set. In the first set of results 13, the models exhibit varying accuracies, with the Neural Network emerging as the top performer. However, the precision, recall, and F1 Score metrics reveal trade-offs, emphasizing the need for a balanced evaluation beyond accuracy alone. The second set of results 14 further highlights the diversity in model capabilities. The Decision Tree and Random Forest models demonstrate high accuracy, precision, and recall, showcasing their effectiveness in handling complex patterns. The SVM model excels in precision but with a slightly lower recall, suggesting a focus on accurate positive predictions at the expense of potentially missing some positive instances. In contrast, the Neural Network exhibits strong recall but lower precision, indicating its proficiency in capturing positive instances but with a tendency for false positives. Overall, these findings underscore the importance of considering multiple metrics and understanding the specific goals of the classification task when selecting the most suitable model.

While metrics such as accuracy and F1 score provide valuable insights into the performance of predictive models for clutch failures in the automotive domain, the ultimate evaluation should extend beyond numerical values to practical outcomes observed by Original Equipment Manufacturers (OEMs) in real-world settings. Witnessing the models' predictions in action, deployed across a fleet of vehicles, offers a more comprehensive understanding of their efficacy in identifying and mitigating clutch failures. Direct feedback from OEMs, based on the observed reduction in actual failures and the successful prevention of critical issues, becomes the most meaningful metric. This real-world validation ensures that the models not only meet technical benchmarks but also align with the broader goals of improving operational efficiency, reducing maintenance costs, and enhancing overall vehicle reliability. By actively involving OEMs in the assessment process, the automotive industry can bridge the gap between predictive modeling in controlled environments and the dynamic challenges faced on the roads, fostering a more robust and practical evaluation framework.

Bibliography

- [1] Sulaiman, A., Tayeh, T., Myers, R., and Shami, A. (2022). Similarity-Based Predictive Maintenance Framework for Rotating Machinery. *arXiv preprint arXiv:2212.14550*. DOI: <https://doi.org/10.48550/arXiv.2212.14550>.
- [2] Haseeb, A., Mohd, N., Mohd, S., Saedudin, R., Hussain, K., and Muhammad, F. (2019). Imbalance class problems in data mining: A review. *Indonesian Journal of Electrical Engineering and Computer Science*, (pp. 1560–1571), DOI: <http://doi.org/10.11591/ijeecs.v14.i3.pp1552-1563>.
- [3] Badillo, S., Banfai, B., Birzele, F., Davydov, II., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., Zhang, JD. An Introduction to Machine Learning, *Clin Pharmacol Ther. 2020 Apr*; (pp. 871–885), doi: 10.1002/cpt.1796.
- [4] Becker J., Michael H., and Oliver P. (2017). System Architecture and Safety Requirements for Automated Driving. In: *Automated Driving: Safer and More Efficient Future Driving*, (pp. 265–283). ISBN: 978-3-319-31895-0. DOI: https://doi.org/10.1007/978-3-319-31895-0_11.
- [5] Bengio Y., Aaron C., and Pascal V. (2012). Unsupervised feature learning and deep learning: A review and new perspectives. *ArXiv abs/1206.5538* Available at: <https://api.semanticscholar.org/CorpusID:4493778>.
- [6] Biau G., and Erwan S. (2016). A random forest guided tour. *Statistics Surveys*, (pp. 197–227), DOI: <https://doi.org/10.48550/arXiv.1511.05741>.
- [7] Bokrantz J., Anders S., Cecilia B., Thorsten W., and Johan S. (2020). Smart Maintenance: an empirically grounded conceptualization. *International Journal of Production Economics*, ISSN: 0925-5273, DOI: <https://doi.org/10.1016/j.ijpe.2019.107534>.
- [8] Breiman L. (2001). Random forests. *Machine Learning*, (pp. 5–32), DOI: <https://doi.org/10.1023/A:1010933404324>, Accessed (November 29, 2023).

- [9] Burkhart N., and HUber M. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, (pp. 245–317), DOI: <https://doi.org/10.48550/arXiv.2011.07876>.
- [10] Cachada, A., Jose, B., Paulo, L., Carla, G., Leonel, D., Jacinta, C., Carlos, T., João, T., António, M., Pedro, M., and Luís, R. (2018). Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture. In: *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, (pp. 139–146), DOI:10.1109/ETFA.2018.8502489.
- [11] Cai, J., Jiawei, L., Shulin, W., and Sheng, Y. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, (pp. 70–79), DOI: <https://doi.org/10.1016/j.neucom.2017.11.077>.
- [12] Canbek, G., Tugba, T., and Seref, S. (2022). PToPI: A comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics. *SN Computer Science*, DOI: <https://doi.org/10.1007/s42979-022-01409-1>.
- [13] Charbuty, B., and Adnan, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, (pp. 20–28), DOI:10.38094/jastt20165.
- [14] Chen, C., Ying, L., Xianfang, S., Carla, C., and Scott, T. (2020). Automobile Maintenance Modelling Using gcForest. In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, (pp. 600–605), DOI:10.1109/CASE48305.2020.9216745.
- [15] Chicco, D., and Giuseppe, J. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, (pp. 1–13), DOI: 10.1186/s12864-019-6413-7.
- [16] Xu, C., Ilyas, I., Krishnan, S., and Wang, X. (2016). Data learning: Overview and emerging challenges. In: *Proceedings of the 2016 International Conference on Management of Data*, (pp. 2201–2206), DOI: <https://doi.org/10.1145/2882903.2912574>.
- [17] Nello, C., and Shawe-Taylor, J. (2000). Machine learning and kernel methods. *Cambridge University Press*, (pp. 337–404), DOI:10.1017/S0263574700232827
- [18] Saza, M., Elrahman A., and Abraham A. (2013). A review of class imbalance problem. *Journal of Network and Innovative Computing*, (pp. 332–340) Available at: <https://api.semanticscholar.org/CorpusID:10781880>.

- [19] Faris, H., Ibrahim, A., and Seyedali, M. (2016). Training Feedforward neural networks using multi-verse optimizer for binary classification problems. *Applied Intelligence*, (pp. 322–332), DOI: 10.1007/s10489-016-0767-1.
- [20] Feng, J., Huan, X., Shie, M., and Shuicheng, Y. (2014). Robust logistic regression and classification. In: *Advances in Neural Information Processing Systems*, Available at: <https://api.semanticscholar.org/CorpusID:7703595>.
- [21] Fernandez, A., Garcia, S., Herrera, F., and Chawla, N. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, (pp. 863–905), DOI: <https://doi.org/10.1613/jair.1.11192>.
- [22] Francis, L., Pierozan, V., Gracioli, G., & Medeiros, G. (2022). Data-driven Anomaly Detection of Engine Knock based on Automotive ECU. In *2022 Brazilian Symposium on Computing Systems Engineering (SBESC)*, (pp. 1–8), DOI: 10.1109/SBESC56799.2022.9965059.
- [23] Giobergia, F., Baralis, E., Camuglia, M., Cerquitelli, T., Mellia, M., Neri, A., Tricarico, D., & Tuninetti, A. (2018). Mining sensor data for predictive maintenance in the automotive industry. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, (pp. 351–360), DOI:10.1109/DSAA.2018.00046.
- [24] Gold, C., & Peter, S. (2003). Model selection for support vector machine classification. *Neurocomputing*, (pp. 221–249), DOI: [https://doi.org/10.1016/S0925-2312\(03\)00375-8](https://doi.org/10.1016/S0925-2312(03)00375-8).
- [25] Gosain, A., & Sardana, S. (2017). Handling class imbalance problem using over-sampling techniques: A review. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (pp. 79–85), DOI: 10.1109/ICACCI.2017.8125820.
- [26] Hall, M. (2000). Correlation-based feature selection of discrete and numeric class machine learning. In *ICML Proceedings of the Seventeenth International Conference on Machine Learning*, (pp. 359-366).
- [27] Feng, H., Guoshun, C., Cheng, Y., Bingru, Y., & Yumei, C. (2005). A SVM regression based approach to filling in missing values. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, (pp. 581–587), DOI:10.1007/11553939_83.

- [28] Hossin, M., & Sulaiman, N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, DOI:10.5121/ijdkp.2015.5201.
- [29] Jeatrakul, P., & Wong, K. (2009). Comparing the performance of different neural networks for binary classification problems. In *2009 Eighth International Symposium on Natural Language Processing*, (pp. 111–115), DOI:10.1109/SNLP.2009.5340935.
- [30] Jegadeeshwaran, R., & Sugumaran, V. (2015). Brake fault diagnosis using Clonal Selection Classification Algorithm (CSCA) – A statistical learning approach. *Engineering Science and Technology, an International Journal*, 18(1), (pp. 14–23). Available at <https://www.sciencedirect.com/science/article/pii/S221509861400055X>
- [31] Kaiser, K., & Nagi, G. (2009). Predictive maintenance management using sensor-based degradation models. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, (pp. 840–849).
- [32] Kim, J., Ragnathan, R., & Jochim, M. (2013). Towards dependable autonomous driving vehicles: A system-level approach. DOI: <https://doi.org/10.1145/2492385.2492390>
- [33] Koehrsen, W. (2018). Overfitting vs. underfitting: A complete example. *Towards Data Science*, Accessed (November 23, 2023).
- [34] Li, P., Xi, R., Blase, J., Yue, Z., Xu, C., & Ce, Z. (2019). CleanML: A benchmark for joint data cleaning and machine learning [Experiments and Analysis]. *arXiv preprint arXiv:1904.09483*, 75.
- [35] Lindholm, A., Niklas, W., Fredrik, L., & Thomas, B. (2019). Supervised machine learning. *Department of Information Technology, Uppsala University*
- [36] Lipton, Z. (2018). The mythos of model interpretability: In *machine learning, the concept of interpretability is both important and slippery*, (pp. 31–57).
- [37] Liu, L. (2018). Research on logistic regression algorithm of breast cancer diagnose data by machine learning. In *2018 International Conference on Robots & Intelligent System (ICRIS)*, (pp. 157–160).
- [38] Longadge, R., & Dongre, S. (2013). Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*.

- [39] Magargle, R., Lee, J., Padmesh, M., Peyman, D., Omkar, K., Sivasubramani, K., John, B., & Anand, P. (2017). A simulation-based digital twin for model-driven health monitoring and predictive maintenance of an automotive braking system. In *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, 132, (pp. 35–46), Linköping University Electronic Press.
- [40] Mahesh, B., & Batta, A. (2020). Machine learning algorithms: A review. *International Journal of Science and Research (IJSR)*, (pp. 381–386).
- [41] Maksymova, I., Christian, S., & Norbert, D. (2018). Review of LiDAR sensor data acquisition and compression for automotive applications.
- [42] Mandrekar, J., & Jayawant, N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, (pp. 1315–1316).
- [43] Marchetti, M., Dario, S., Alessandro, G., & Michele, C. (2016). Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms. In *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, (pp. 1–6).
- [44] MathWork (2021). Types of Predictive maintenance Approaches. Available at: <https://www.mathworks.com/discovery/predictive-maintenance-matlab.html> Accessed (July 10, 2022).
- [45] Akshay, M., & Juneja, M. (2021). Types of Machine Learning Algorithms. Available at: <https://www.info4eee.com/2021/12/supervised-learning-and-unsupervised.html>, Accessed (June 06, 2022).
- [46] Mclleman, A., Jack, S., & Parker, K. (2021). The maintenance function, like manufacturing itself, is a rapidly changing environment. Available at: <https://www.plantengineering.com>, Accessed (May 20, 2023).
- [47] Meyendorf, M., & Norbert, N. (2018). The role of data fusion in predictive maintenance using digital twin. *AIP Conference Proceedings*
- [48] Milojevic, M., & Franck, N. (2018). Digital industrial revolution with predictive maintenance. *Are European Businesses Ready to Streamline Their Operations and Reach Higher Levels of Efficiency*.
- [49] Min, H., & Chulwoo, J. (2009). A binary classification method for bankruptcy prediction. *Expert Systems with Applications*, (pp. 5256–5263), DOI: <https://doi.org/10.1016/j.eswa.2008.06.073>.

- [50] Minka, T. (2001). Algorithms for maximum-likelihood logistic regression. *Statistics Tech Report 758*.
- [51] Roweida, M., Rawashdeh, J., & Abdullah, M. (2020). Machine learning with over-sampling and under-sampling techniques: overview study and experimental results. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, (pp. 243–248).
- [52] Muhammad, I., & Zhu, Y. (2015). Supervised Machine Learning Approach: A survey. *ICTACT Journal on Soft Computing*, 5(3).
- [53] Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golu, T., Mesirov, J., & Poggio, T. (1999). Support vector machine classification of microarray data. *Tech. rep. AI Memo 1677, Massachusetts Institute of Technology*.
- [54] Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons*, (pp. 51–62), DOI: 10.20544/HORIZONS.B.04.1.17.P05, Accessed (July 23, 2023).
- [55] Purves, R. (1992). Optimum numerical integration methods for estimation of area-under-the-curve (AUC) and area-under-the-moment-curve (AUMC). *Journal of Pharmacokinetics and Biopharmaceutics*, (pp. 211–226).
- [56] Bahadur, R., Krishnan, G., Wang, S., Kamei, Y., & Hassan, A. (2019). Impact of discretization noise of the dependent variable on machine learning classifiers in software engineering. *IEEE Transactions on Software Engineering*, (pp. 1414–1430), DOI: <https://doi.org/10.48550/arXiv.2202.06146>.
- [57] Reddy, S., Thota, A., & Dharun, A. (2018). Machine learning techniques for stress prediction in working employees. In *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, (pp. 1–4), DOI: DOI:10.1109/ICCIC.2018.8782395.
- [58] Rigatti, SJ. (2017). Random Forest. *J Insur Med. 2017*, (pp. 31-39), PMID: 28836909, DOI: 10.17849/in-sm-47-01-31-39.1.
- [59] Sankavaram, C., Pattipati, B., Pattipati, K., Zhang, Y., Howell, M., & Salman, M. (2012). Data-driven fault diagnosis in a hybrid electric vehicle regenerative braking system. In *2012 IEEE Aerospace Conference*, (pp. 1–11), DOI: 10.1109/AERO.2012.6187368.
- [60] Ebrahim, M., Ibrahim, A., Jadhav, M., & Mohamed, S. (2021). Score and correlation coefficient-based feature selection for predicting heart failure diagnosis by using machine learning algorithms. *Computational and Mathematical Methods in Medicine*, DOI: 10.1155/2021/8500314.

- [61] Shiksha (2022). Types of Machine Learning. Available at: <https://www.shiksha.com/online-courses/articles/differences-between-supervised-and-unsupervised-learning/>, Accessed (June 06, 2022).
- [62] Shuo, J., Yu, X., Xiao, Y., Bennin, K., Kabir, A., & Zhang, M. (2021). COSTE: Complexity-based OverSampling Technique to alleviate the class imbalance problem in software defect prediction. *Information and Software Technology*, DOI: <https://doi.org/10.1016/j.infsof.2020.106432>.
- [63] Singh, S., & Gupta, P. (2014). Comparative study ID3, CART and C4.5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, (pp. 97–103).
- [64] Paria, S., & Hashemzadeh, M. (2021). RCSMOTE: Range-Controlled synthetic minority over-sampling technique for handling the class imbalance problem. *Information Sciences*, (pp. 92–111), DOI: <https://doi.org/10.1016/j.ins.2020.07.014>.
- [65] Steve, C. (2002). Introduction to the controller area network (CAN). *Application Report SLOA101*, (pp. 1–17).
- [66] Tangirala, S., & Suryakanthi (2020). Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm. *International Journal of Advanced Computer Science and Applications*, (pp. 612–619), DOI: <http://dx.doi.org/10.14569/IJACSA.2020.0110277>.
- [67] Theissler, A., Pérez-Velázquez, J., Kettelgerdes, M., & Elger, G. (2021). Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. *Reliability Engineering System Safety*, 215, 107864. ISSN: 0951-8320. DOI: <https://doi.org/10.1016/j.ress.2021.107864>, Available at: <https://www.sciencedirect.com/science/article/pii/S0951832021003835>.
- [68] Turner, C., Okorie, O., Emmanouilidis, C., & Oyekan, J. (2020). A digital maintenance practice framework for circular production of automotive parts. *IFAC-PapersOnLine*, (pp. 19–24), DOI: <https://doi.org/10.1016/j.ifacol.2020.11.004>.
- [69] Ukil, A. (2007). Support vector machine. In *Intelligent systems and signal processing in power engineering*, (pp. 161–226), DOI: 10.1007/978-3-540-73170-2-4.
- [70] Werbińska-Wojciechowska, S. (2019). Preventive Maintenance Models for Technical Systems. In *Technical System Maintenance: Delay-Time-Based Modelling* (pp. 21–100), ISBN: 978-3-030-10788-8. DOI: <https://doi.org/10.1007/978-3-030-10788-8-2>.

- [71] Widodo, A., & Bo-Suk, Y. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, (pp. 2560–2574), DOI: 10.1016/j.ymssp.2006.12.007.
- [72] Wilk, T. (2022). 2022 PdM survey results: How does your plant compare? Available at: <https://www.plantservices.com/predictive-maintenance/predictive-maintenance/article/21435521/2022-pdm-survey-results-how-does-your-plant-compare>, Accessed (July 05, 2023).
- [73] Williams, N., Zander, S., & Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Computer Communication Review*, (pp. 5–16), DOI: <https://doi.org/10.1145/1163593.116359>.
- [74] Peter, W., Mrowca, A., Nguyen, T., Bäker, B., & Günnemann, S. (2018). Pre-ignition Detection Using Deep Neural Networks: A Step Towards Data-driven Automotive Diagnostics. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, (pp. 176–183), DOI: <https://doi.org/10.1109/ITSC.2018.8569908>.
- [75] Wong, K., Zhong, J., Yang, Z., & Vong, C. (2016). Sparse Bayesian extreme learning committee machine for engine simultaneous fault diagnosis. *Neurocomputing*, ISSN: 0925-2312, DOI: 10.1016/j.neucom.2015.02.097, Available at: <https://www.sciencedirect.com/science/article/pii/S0925231215011765>.
- [76] Wu, Q., & Zhou, D. (2006). Analysis of support vector machine classification. *Journal of Computational Analysis & Applications*, Available at: <http://www.eudoxuspress.com/JoCAA/v8-06.pdf>.
- [77] Li, X., Ding, Q., & Sun, J. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering System Safety*, 172, (pp. 1–11). ISSN: 0951-8320, DOI: <https://doi.org/10.1016/j.ress.2017.11.021>, Available at: <https://www.sciencedirect.com/science/article/pii/S0951832017307779>.
- [78] Zhen, Y., Kannianen, J., Krogerus, T., & Emmert-Streib, F. (2022). Prognostic modeling of predictive maintenance with survival analysis for mobile work equipment. *Scientific Reports*, (pp. 1–20), DOI: 10.1038/s41598-022-12572-z.
- [79] Zhang, Q., & Zhu, S. (2018). Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, (pp. 27–39), DOI: <https://doi.org/10.48550/arXiv.1802.00614>.

- [80] Zhon, J., Wong, P., & Yang, Z. (2018). Fault diagnosis of rotating machinery based on multiple probabilistic classifiers. *Mechanical Systems and Signal Processing*, (pp. 99–114), ISSN: 0888-3270, DOI: 10.1016/j.ymssp.2018.02.009, Available at: <https://www.sciencedirect.com/science/article/pii/S0888327018300657>.