

Prediction of Failures in the Air Pressure System of Scania Trucks



Author:

Naresh Kumar Nagaraj

Student ID: 21250702

Supervisor:

DR. DAPENG DONG

Head of the Department:

DR. ADAM WINSTANLEY

A dissertation submitted in partial fulfilment of the requirements

for the degree of

MSc in Data Science and Analytics

2021-2022

in the

Department of Computer Science,

Maynooth University

August 8, 2022

ABSTRACT

MSc in Data Science and Analytics

Prediction of failures in the Air Pressure System

of Scania Trucks by

by NARESH KUMAR

NAGARAJ

The air pressure system is a crucial component of heavy vehicles like Scania trucks. This vehicle's braking system depends on air pressure, so the air pressure system must function properly. The automotive industry uses predictive maintenance to reduce maintenance costs and improve vehicle performance. There are both manual and automatic methods available for accomplishing this. The manual predictive maintenance process involves human interference and is subject to error. Using artificial intelligence techniques, automatic predictive maintenance uncovers the root cause of air pressure system failures in Scania trucks. Machine learning approaches are used to investigate predictive maintenance of the trucks based on the state of the air pressure system. Data analysis is used in this report to reduce overall maintenance costs for Scania trucks' air pressure systems. In heavy-duty trucks and buses, air pressure braking systems (APS) use compressed air to apply pressure to brake pads in order to stop the vehicle. There is an inextricable link between road safety and APS reliability. Repair and maintenance costs for heavy vehicles' air pressure systems are high. For Scania Truck, the task at hand is to create neural network models using PyTorch or TensorFlow to predict APS failure and reduce associated maintenance expenses. After pre-processing the data, machine learning neural network algorithms like Basic Neural Network, Recurrent Neural Network, Multilayer Perceptron, and Artificial Neural Network are implemented and the accuracy of the classifiers is analyzed. Experimental results show that Artificial Neural Network with both layers outperforms other models in both scenarios considering that without removing correlated columns and by removing correlated columns with an accuracy more than 94%, highest the AUC value and lower the maintenance cost.

Keywords: Air Pressure Braking Systems, Predicting failure, Neural Network

Acknowledgement

In appreciation for the opportunity to work under Dr. Dapeng Dong, my primary supervisor, as well as for his patience and advice throughout the project work, I extend my sincere gratitude. This project would not have been possible without his insightful recommendations and insightful criticism. I would like to acknowledge the technical and support staff at Maynooth University's Computer Science department for their assistance. In addition, I would like to express my deep gratitude to my supervisors for helping me complete my project.

Declaration

I hereby attest that the content I am submitting for evaluation as part of the MSc in Data Science and Analytics qualification program is entirely original work of mine and has not been adapted in any way, save for instances where it has been cited and acknowledged within the text of my work.

Signed: NARESH KUMAR NAGARAJ

Date: 08-08-2022

Table of Contents

Chapter 1 Introduction.....	10
1.1 Scania Trucks	10
1.2 Anti-lock braking system	10
1.3 Motivation.....	11
1.4 Objective	14
Chapter 2 Background Literature Review	15
Chapter 3 Data Science	17
3.1 Overview	17
3.2 Methodologies Used	17
3.2.1 Machine Learning.....	17
3.2.2 Classification Models	17
3.2.3 Neural Network	17
3.2.4 Artificial Neural Networks	18
3.2.5 Recurrent Neural Network.....	19
3.2.6 Multi-Layer Perceptron.....	20
3.2.7 Long Short-Term Memory Recurrent Neural Networks.....	21
3.3 Metrics.....	22
3.3.1 Feature Scaling	22
3.3.2 Imputation.....	22
3.3.3 Dimensionality Reduction.....	23
3.3.4 Confusion Matrix	23
3.3.5 AUC	23
3.3.6 Accuracy	24
3.4 Tools Used.....	24
3.4.1 Python as a programming language	24
3.4.2 Python Libraries.....	25
Chapter 4 Data Exploration and Analysis	28
4.1.3 Feature Class Imbalance.....	30
4.1.4 NAN and Zero Analysis	31

4.1.5 Feature Correlation.....	33
Chapter 5 Data Preparation & Cleansing.....	34
5.1 Target class distribution.....	34
5.2 Missing Value Analysis.....	35
5.3 Correlation.....	37
5.4 Dimensionality Reduction and balancing the data	39
5.5 Train/Validation/Test data.....	40
5.6 COST CALCULATION.....	42
Chapter 6 Model Implementation	43
6.1 Basic Neural Network.....	44
6.1.1 Without Removing Correlated Columns.....	45
6.1.2 Removing Correlated Columns.....	48
6.2 Recurrent Neural Network	50
6.2.1 Without Removing Correlated Columns.....	51
6.2.2 Removing Correlated Columns.....	53
6.3 Multi-Layer Perceptron	56
6.3.1 Without Removing Correlated Columns.....	56
6.3.2 Removing Correlated Columns.....	58
6.4 Artificial Neural Network [Layers 60,30,15 ,1]	60
7.4.1 Without Removing Correlated Columns.....	61
7.4.2 Removing Correlated Columns.....	63
6.5 Artificial neural network [Layers 64,32,16,1]	66
7.5.1 Without Removing Correlated Columns.....	66
7.5.2 Removing Correlated Columns.....	68
Chapter 7 Model Evaluation	71
7.1 Without Removing Correlated Columns	71
7.2 Removing Correlated Columns	72
Chapter 8 Conclusion	73
References.....	74

List of Figures

Figure 1. Scania Trucks	10
Figure 2. Model of Air Pressure System	12
Figure 3. Missing values (NA) in the data set	13
Figure 4. Deep Neural Network Layers	18
Figure 5. An Artificial Neuron flow representation	19
Figure 6. Model of A Recurrent Neural Network	20
Figure 7. Layers of Multi-Layer Perceptron	21
Figure 8. LSTM Unit	21
Figure 9. Area under the ROC Curve	24
Figure 10. Data type of the Training Data	29
Figure 11. Null Ratio for features	30
Figure 12. Target class distribution for Training Set	31
Figure 13. Predictors in the data set with highest missing percentage of NaNs	32
Figure 14. Feature Correlations	33
Figure 15. Target Class Distribution	35
Figure 16. Correlation Matrix	38
Figure 17. Data type of test data	41
Figure 18. Steps involved in developing a machine learning model.	43
Figure 19. Basic Neural Network model for Validation Data	45
Figure 20. Basic Neural Network model for Validation Data	45
Figure 21. Confusion Matrix for Basic Neural Network model	46
Figure 22. Loss Curve Diagram for Basic Neural Network model	46
Figure 23. ROC curve for the Basic Neural Network model	47
Figure 24. Basic Neural Network model for Validation Data	48
Figure 25. Basic Neural Network model for Test Data	48
Figure 26. Confusion Matrix for Basic Neural Network model	49
Figure 27. Loss Curve Diagram for Basic Neural Network model	49
Figure 28. ROC curve for the Basic Neural Network model	50
Figure 29. Recurrent Neural Network model for Validation Data	51
Figure 30. Recurrent Neural Network model for Test Data	52
Figure 31. Confusion Matrix for Recurrent Neural Network model	52
Figure 32. Loss Curve Diagram for Recurrent Neural Network model	52
Figure 33. ROC curve for the Recurrent Neural Network model	53

Figure 34. Recurrent Neural Network model for Validation Data.....	53
Figure 35. Recurrent Neural Network model for Test Data	54
Figure 36. Recurrent Neural Network model for Test Data	54
Figure 37. Loss Curve Diagram for Recurrent Neural Network model	55
Figure 38. ROC curve for the Recurrent Neural Network model	55
Figure 39. Multi-layer perceptron model for Validation Data	56
Figure 40. Multi-layer perceptron model for Test Data	57
Figure 41. Confusion Matrix for Multi-layer perceptron model.....	57
Figure 42. ROC curve for the multi-layer perceptron model	58
Figure 43. Multi-layer perceptron model for Validation Data	58
Figure 44. Multi-layer perceptron model for Test Data	59
Figure 45. Confusion Matrix for Multi-layer perceptron model.....	59
Figure 46. ROC curve for the Multi-layer perceptron model.....	59
Figure 47. ANN model[layers 60,30,15,1] for Validation Data	61
Figure 48. ANN model[layers 60,30,15,1] for Test Data.....	61
Figure 49. Confusion Matrix for ANN model[layers 60,30,15,1]	62
Figure 50. Loss Curve Diagram for ANN model[layers 60,30,15,1]	62
Figure 51. ROC curve for the ANN model[layers 60,30,15,1].....	63
Figure 52. ANN model[layers 60,30,15,1] for Validation Data	63
Figure 53. ANN model[layers 60,30,15,1] for Test Data.....	64
Figure 54. Confusion Matrix for ANN model[layers 60,30,15,1].....	64
Figure 55. Loss Curve Diagram for ANN model[layers 60,30,15,1]	65
Figure 56. ROC curve for the ANN model[layers 60,30,15,1].....	65
Figure 57. ANN model[layers 64,32,16,1] for Validation Data	66
Figure 58. ANN model[layers 64,32,16,1] for Test Data.....	66
Figure 59. Confusion Matrix for ANN model[layers 64,32,16,1].....	67
Figure 60. Loss Curve Diagram for ANN model[layers 64,32,16,1]	67
Figure 61. ROC curve for the ANN model[layers 64,32,16,1].....	68
Figure 62. ANN model [layers 64,32,16,1] for Validation Data	68
Figure 63. ANN model[layers 64,32,16,1] for Test Data.....	69
Figure 64. Confusion Matrix for ANN model[layers 64,32,16,1]	69
Figure 65. Loss Curve Diagram for ANN model[layers 64,32,16,1]	70
Figure 66. ROC curve for the ANN model[layers 64,32,16,1].....	70

List of Tables

Table 1. Model Selection (without removing correlated columns).....	72
Table 2. Model Selection (Removing correlated columns).....	72

Chapter 1

Introduction

1.1 Scania Trucks

Scania is a global leader in the manufacture of trucks and buses for heavy transport applications, along with a wide range of product-related services. With Scania, customers can focus on their core business while they provide financing, insurance, and rental services. Besides engines for industrial and marine applications, Scania is also a leading provider of these products.



Figure 1. Scania Trucks

1.2 Anti-lock braking system

Anti-lock Braking Systems (ABS) are safety anti-skid brakes that can be found on aircraft as well as on land vehicles including cars, motorcycles, trucks, and buses. The ABS system prevents the wheels from locking up during braking, allowing the driver to maintain more control over the vehicle by maintaining tractive contact with the road surface. An automated braking system uses threshold braking and cadence braking principles, which were once used by skilled drivers before ABS became widespread. Many drivers would not be able to handle ABS at the same speed and effectiveness as it operates. When braking on loose gravel or snow-covered surfaces, ABS may significantly increase braking distances, while still improving steering control, even though it generally improves vehicle control and decreases stopping distances on dry surfaces and some slippery surfaces. With the

introduction of ABS in production vehicles, such systems have become more sophisticated and effective.

1.3 Motivation of Project

Manufacturing process failure analysis and process predictions have a significant impact on product quality and process reliability. A number of studies are currently being conducted on failure cause analysis and classification using deep learning or machine learning techniques, which will improve the quality and reliability of products.

Situation: For instance, the use of heavy vehicles as the main form of road transportation is extremely important in industrial sectors. They operate regularly in every industry and are the most adaptable and affordable form of transportation. The secret to preventing any unintended breakdowns and thereby saving money and time is a well-done maintenance. It is crucially important in this situation that all of the vehicle's parts are routinely maintained. If a specific level of system air pressure is not reached its brakes might not operate precisely, which results in fatal accidents. This project focuses on quality prediction of air compressors and pressure systems used in vehicles. One such crucial element is the Air Pressure System (APS). In order to perform various operations like braking and gear changing, the APS produces pressurized air, which requires maintenance.

Complication: There are a lot of subsystems in the truck. APS, or Air Pressure System, is one of these subsystems. The figure 2 below illustrates how an air compressor compresses air that is initially at atmospheric pressure, then regulates the compressor to maintain the ideal pressure. Compressed air is received in the air reservoir. By applying the brakes, the brake valve closes, forcing compressed air from the air reservoir through the line and applying pressure to the brake chamber.

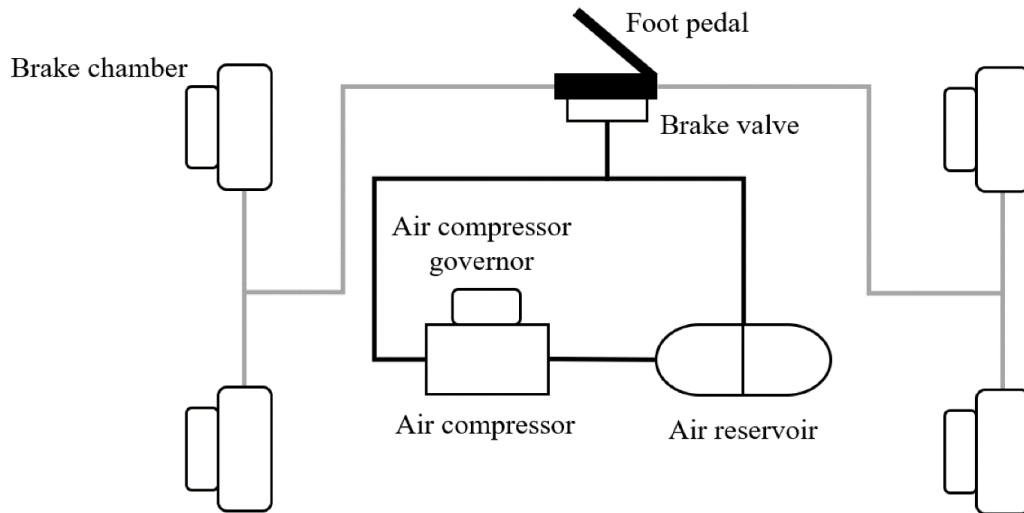


Figure 2. Model of Air Pressure System

The main function of APS is to maintain the required air pressure inside the truck. Instead of the conventional hydraulic brakes used in other lighter vehicles, modern heavy haul trucks use air brakes. For the vehicle to continue moving, these brakes require a continuous flow of pressurized air to remain disengaged. The truck will come to a stop if for any reason this source of pressurized air is compromised, preventing the brakes from disengaging. When this occurs, the owners are required to send a repair vehicle to identify and fix the problem. However, the APS, which is made up of numerous pipes and is present throughout the truck and the trailer, is what actually supplies this air. Because of this, determining whether the issue is related to the APS or not is very difficult to do. It's critical to anticipate APS flaws and malfunctions. A thoughtful analysis of the data produces accurate predictions of potential APS failures, reduces maintenance costs, and minimizes risks. For this study, failure data came from APS operations, and quality data came from Scania trucks. In this data set, the operating sensor attributes of a damaged Scania truck are included. Anonymization has been applied to the attributes. In order to determine the reasons behind any APS failure, these attributes and their data are analysed using a variety of statistical and machine learning techniques.

There are, however, a number of values missing from the data due to sensor failures. There may be insufficient data to analyse using current techniques and it may be difficult to pinpoint the exact cause of failure if data is missing. Unequally distributed data lowers the classification accuracy of models trained on properly proportioned data. Figure 2 illustrates the problem of missing values in the Scania trucks data set. Missing values are marked as unavailable (NA).

In order to analyse a vehicle's failure, the information is used. A positive class (pos) indicates an APS failure, while a negative class (neg) indicates an unrelated failure.

In APS failure analysis, classification accuracy is highly dependent on data completeness. As more missing values are present in the data, the level of completeness decreases. Scania APS data has quite a few missing values. There might also be a lack of data for analysis due to these missing values. It is therefore crucial to have a framework that deals with incomplete data. Missing values and data imbalances lead to less accurate quality predictions. Additionally, the problems with missing values prevent this data set from undergoing a number of multivariate statistical analyses. The most crucial pre-processing steps in this situation are missing value estimations. Numerous studies that have already been conducted have suggested solutions to this data imbalance problem. The various missing value estimation techniques currently in use along with the applications. The majority of research techniques use data imputation, which ignores a data set with missing values. The figure 3 represents the NA values in the training data set.

class	cs_007	cs_008	cs_009	ct_000	cu_000	cv_000	cx_000	cy_000	cz_000	da_000	db_000	dc_000	dd_000	de_000
neg	56426	1626	2	1302	2170	1974392	212462	158	5344	0	0	2388370	2384	566
neg	2908	38	0	na	na	na	na	na	na	na	na	na	58	0
neg	1144	2	0	4	10	2270	40	0	10	0	0	2966	66	48
neg	6980	30	0	na	na	na	na	na	na	na	na	na	4528	290
neg	18568	218	0	580	492	2361668	42868	0	69392	0	0	2427552	2016	214
neg	7834	22	0	232	190	1241578	589558	0	150	0	0	1243120	2146	754
neg	21022	138	0	844	690	1959520	7400	0	558	0	16	2202026	1602	108
neg	3926	0	0	na	na	na	na	na	na	na	na	na	na	na
neg	22022	16	0	na	na	na	na	na	na	na	na	na	2356	198
neg	33422	8	0	na	na	na	na	na	na	na	na	na	19392	376
neg	28704	222	0	136	228	428798	72962	38	125876	0	0	2340884	1952	82
pos	12286	2	0	na	na	na	na	na	na	na	na	na	7146	1380
neg	na	na	na	na	na	na	na	na	na	na	na	na	7284	na
neg	1080	26	0	22	84	7416	580	0	0	0	0	7418	122	46
neg	23756	24	0	1690	476	2552692	66626	0	198	0	36	2638892	2872	286
neg	4846	2	0	na	na	na	na	na	na	na	na	na	1264	84
neg	22350	104	0	388	1082	2153658	86778	40	55878	0	0	2250822	1584	152
neg	11240	4	0	56	130	1061968	269440	0	876	0	0	1100612	1154	262
neg	0	0	0	0	10	1930	40	0	0	0	0	2340	52	68
neg	3928	12	0	na	na	na	na	na	na	na	na	na	1970	176
neg	4244	60	0	370	412	130184	76280	0	2	0	30	130224	142	58
neg	1464	40	0	68	24	14538	2650	0	2	0	16	14652	64	62
neg	68618	306	0	2004	1250	7152788	52954	0	385752	0	10	7287292	6296	582
neg	na	na	na	na	na	na	na	na	na	na	na	na	496	138
neg	13864	80	0	296	94	1953440	62890	0	4052	0	26	2208398	1740	248

Figure 3. Missing values (NA) in the data set

proposed solution: As a result, having a machine learning algorithm or system that can determine whether a truck fault is caused by the APS or not will be very helpful for the concerned parties as it can, to some extent, reduce downtimes and the overall cost associated with breakdowns.

1.4 Objective

Numerous businesses are having trouble adapting to the realities of AI implementation as industry continues to attract media attention. Predictive maintenance has a number of highly strategic advantages, including the ability to assess equipment condition and foresee when maintenance needs to be done. ML-based solutions can thus result in significant savings, improved predictability, and increased system availability.

Predicting Scania Air Pressure System (APS) failure in trucks was a challenge because doing so would allow for preventive maintenance, which would lower maintenance costs. Due to confidentiality concerns, the dataset has been anonymized and contains binned values. The objective is to cut down on expenses related to unneeded inspections performed by a mechanic and ignoring a defective truck, which could lead to a future breakdown. However, using neural networks in predicting and reducing the cost of failures related to these reading combinations will be the main goal.

Chapter 2

Background Literature Review

Predictive maintenance is an integral part of the extensive vehicle management system. Due to the advanced brake system on Scania trucks, air pressure systems must have automatic failure detection. If the APS fails, the Scania truck will not perform as well. Scania truck maintenance consequently adopts predictive maintenance as a standard process. Data collected from the vehicle's performance is analysed using artificial intelligence techniques. Machine learning techniques are hindered by unbalanced real-world data, which must be pre-processed before classification algorithms can be applied.

Unbalanced data classification is a crucial problem in information mining and artificial intelligence. An unbalanced dataset is likely to misclassify minority class cases. Most of the real-world dataset exhibits class regularity and can be distributed uniformly using resampling techniques. Various literatures have examined data imbalance for a variety of applications.

The same problem was addressed by Christopher Gondek, Daniel Hafner, and Oliver R. Sampson [3] (2016) using random forest. To fill in the gaps, the median is used instead of normalizing the data. The distance to other distributions, the mean distribution of positive examples, the mean distribution of negative examples, the normal distribution with a mean of 5 and a standard deviation of 1.5, and the mirrored normal distribution were calculated as new features. In order to address the issue of class imbalance, a threshold value of 95% was found to be the most effective. They calculated an average cost of 0.6 by dividing the total maintenance expense by the number of trucks. As an example, they showed how histogram features could be used to enhance prediction. Further, they demonstrated how a threshold on the confidence of a Random Forest can be used to adapt forecasts to cost functions. Finally, the main cost was significantly reduced compared to naive approaches.

The problem was solved by Vitor Cerqueira, Fabio Pinto, Claudio Sa, and Carlos Soares[4] (2016) using boosting trees with meta-features and oversampling. Meta-features were created using box plots, clustering-based outlier ranking, and local outlier finding based on density. SMOTE oversampling techniques were used to resolve class imbalance, and 10-fold cross validation was used to tune the parameters.

To address the problem associated with rarity of classes, G. Weiss [5] (2004) discussed cost-sensitive learning and boosting. The model is trained using gradient boosting trees and cost-sensitive learning.

According to Aman Savaria[6], simple techniques like comparable results using only mean and median imputation. It is discovered during the modeling phase that every model had bias toward the negative class. However, adjusting the prediction threshold significantly increased the models' performance. In addition, he tried models like SMOTE Boost and RUS Boost, which also enhanced performance but reduced misclassification costs at the expense of precision. Finally, He done the conclusion that the XG Boost model would work best with the ideal prediction threshold value.

To reduce costs, Mrugank M. Akarte[7] compared over- and under-sampling strategies. By predicting faults that could lead to breakdowns, machine learning can save businesses a lot of money. By tuning parameters and using a cost-sensitive loss function, bias in an unbalanced class can be reduced and results can be significantly improved. To reduce the cost, different algorithms can be used, such as random forests, neural networks, or ensemble models. Studying advanced sampling techniques like SMOTE and developing new features based on existing features can aid in cost reduction.

By the above referred papers, Median imputation was taken into consideration to replace the missing values in the column. The various model algorithms which were used related to tree based such as random forest, however the requirement of the project is to perform the model using Neural Networks. I have read the papers, which helped me to perform the pre-processing of the data and to conclude on cost aspects for the failure in air pressure system for the Scania trucks. And its way out in implementing the models in neural networks.

Chapter 3

Data Science

3.1 Overview

In data science, scientific techniques and algorithms are used to extract knowledge from structured or unstructured data. In this process, artificial intelligence, machine learning, statistics, and computer science are all applied. In order to produce insights and predictive outcomes, this project combined a variety of statistical and machine learning techniques and algorithms with programming languages like R and Python.

3.2 Methodologies Used

3.2.1 Machine Learning

The ability for computers to learn from data without explicit programming is provided by machine learning, which employs statistical techniques. For the purpose of facilitating predictions, it is frequently used to design models and algorithms. Supervised and Unsupervised machine learning techniques are the two major categories. In order to create prediction models for this project, supervised algorithms were used. Supervised learning involves working with datasets where each set of predictor values has an associated response. Brief descriptions of the various machine learning methods used in this study is provided in this section.

3.2.2 Classification Models

Classification models are a subset of supervised machine learning. Using some input, a classification model produces an output that categorizes the input. It is possible to classify both structured and unstructured data. Classification involves categorizing data into a predetermined number of categories. In a classification problem, the main objective is to determine what category or class new data will fall into.

3.2.3 Neural Network

In deep learning algorithms, neural networks are also referred to as artificial neural networks (ANNs) or simulated neural networks (SNNs), which are subsets of machine learning. In the

same way that biological neurons communicate, the structure and name of these devices are modelled after the human brain.

An artificial neural network (ANN) consists of an input layer, one or more hidden layers, and an output layer represented below in Figure 4. Each node, or artificial neuron, is connected to others and has its own weight and threshold. A node whose output exceeds the threshold value is activated and begins sending data to the uppermost layer of the network. In this case, no data is transmitted to the next layer of the network.

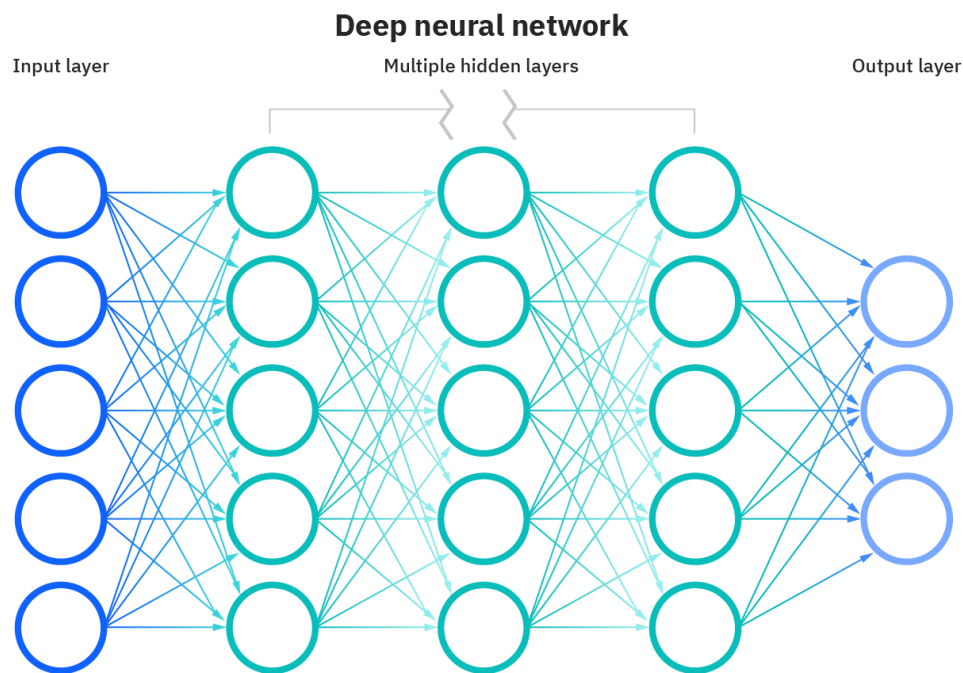


Figure 4. Deep Neural Network Layers

Developing and enhancing neural networks over time requires training data. Once these algorithms are adjusted for accuracy, they become effective tools in computer science and artificial intelligence for classifying and clustering data. Compared to manual identification by human experts, speech recognition or image recognition can be completed in minutes instead of hours. Google's search algorithm uses a neural network, one of the most well-known algorithms.

3.2.4 Artificial Neural Networks

Artificial neural networks (ANNs) are flexible computational models that adapt to the nature of the problem at hand. The architecture of the ANN is said to have been influenced by networked brain cells, or neurons. These neurons' nervous impulses form a processing network

that frequently sends information through different connections. This enables information that has been separately weighted to enter the cell body where it is processed and summed. The connection point then sends the information on to the following neuron after receiving the output. Only a specific set of values is propagated through the network, which results in the generation of an impulse only when a certain excitation threshold is reached, and over time, a distinct configuration that accurately represents the task is established.

The artificial neuron, which serves as the basis of an ANN, is a mathematical representation of the biological neuron. Figure provides a representation of a synthetic neuron.

Figure 5, An artificial neuron is depicted graphically where the inputs are represented by x_i for $i = 1, \dots, m$ and the output is represented by y according to equation.

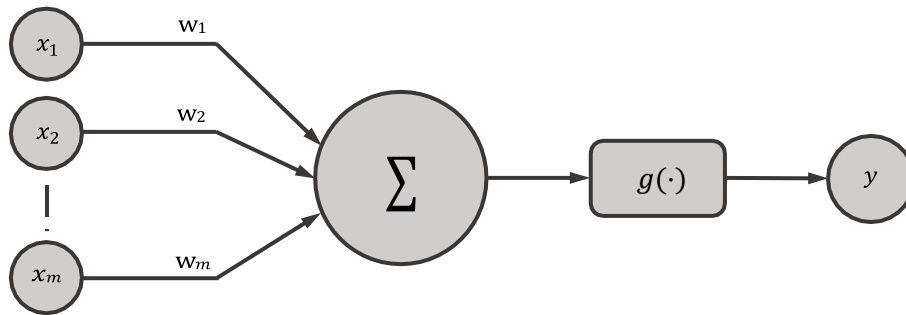


Figure 5. An Artificial Neuron flow representation

3.2.5 Recurrent Neural Network

A recurrent neural network (RNN) is an artificial neural network that uses sequential or time-series data. RNNs are frequently used for ordinal or temporal problems, including language translation, natural language processing (NLP), speech recognition, and image captioning. Like feedforward and convolutional neural networks (CNNs), recurrent neural networks (RNNs) learn from training data. Due to their "memory," these systems can use data from previous inputs to affect current inputs and outputs. Traditional deep neural networks assume that inputs and outputs are independent of one another, but recurrent neural networks depend on the previous elements in the sequence. While unidirectional recurrent neural networks are useful for predicting outputs, they cannot take into account future events in their predictions.

Deep learning techniques for sequential characters were generalized to create the recurrent neural network (RNN). With a dynamic memory component for its input sequences, the RNN differs from earlier ANN architectures. The output from a neuron at a previous time step

influences the calculations in the current neuron as opposed to processing each data point separately. RNNs typically outperform static ANNs on issues based on temporally dependent time series because of this dynamic memory feature.

Conventional RNNs have a dynamic looping mechanism that creates hidden states in their internal memory. Through recurrent connections from the previous output unit onto itself, these can be changed, effectively forming a folded network. An unrolled RNN with a single layer and its corresponding features are shown in Figure 6.

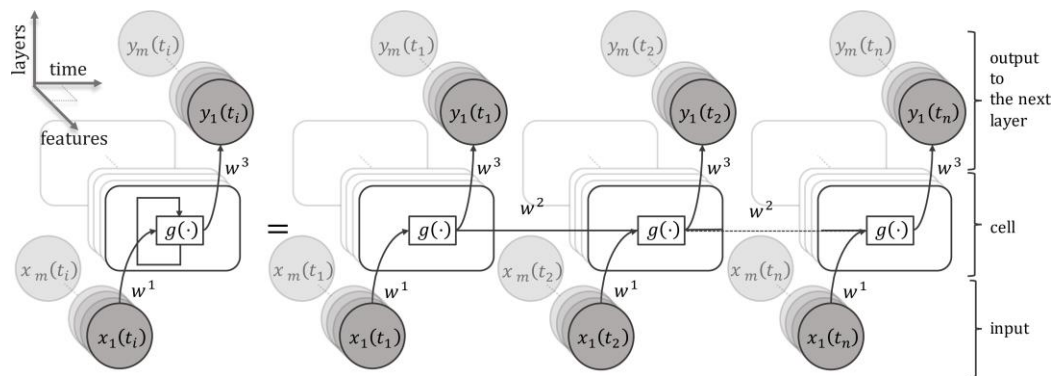


Figure 6. Model of A Recurrent Neural Network

3.2.6 Multi-Layer Perceptron

A multilayer perceptron (MLP) is added to the feed forward neural network. Three types of layers are shown in below Figure 7, the input layer, the output layer, and the hidden layer. Input signals are received at the input layer for processing. In the output layer, tasks such as classification and prediction are completed. The MLP's real computational engine is composed of an arbitrary number of hidden layers sandwiched between the input and output layers. Much like a feed forward network, data flows from the input to the output layer of an MLP. MLP neurons are trained using the back propagation learning algorithm. By approximating any continuous function, MLPs can resolve issues that are not linearly separable. MLP is mainly used for pattern classification, recognition, prediction, and approximation.

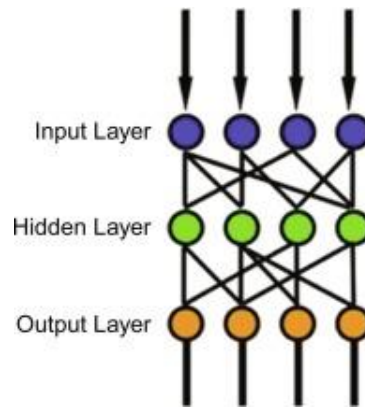


Figure 7. Layers of Multi-Layer Perceptron

3.2.7 Long Short-Term Memory Recurrent Neural Networks

Recurrent neural networks use their internal state or memory to process input sequences. A directed graph is formed by the connections between different nodes in an RNN. LSTM networks are recurrent neural networks (RNNs) with LSTM units. RNN layers are built using LSTM units. The LSTM cell adds three extra gates as well as the cell state $c(t)$, a secondary memory state. This enables the LSTM unit to decide whether it should store the memory that has passed through these specific gates or completely disregard it.

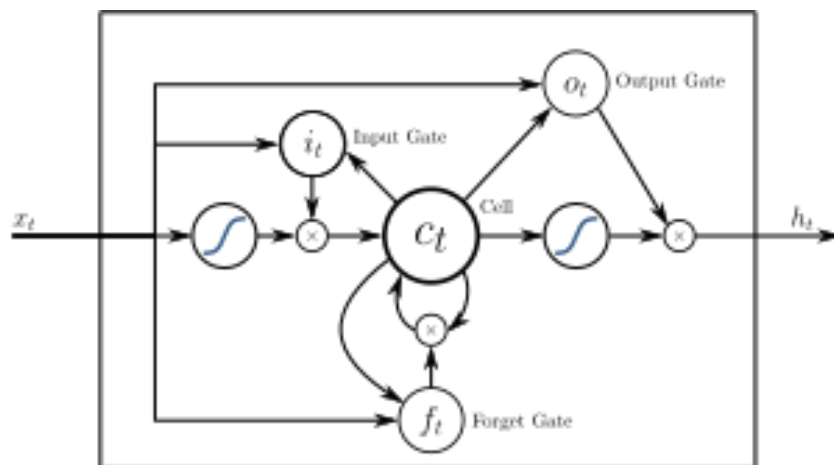


Figure 8. LSTM Unit

LSTM units consist of cells, input gates, output gates, and forget gates. The three gates calculate a weighted sum, and the cell memorizes the results. The output gate decides whether the input should affect the output during the current step, the forget gate decides

whether to delete unimportant information, and the input gate decides whether to allow the input in. With LSTMs, RNNs can memorize inputs for a longer period of time. LSTM networks are excellent for classification and time series prediction. LSTMs don't suffer from disappearing gradients, and training is quick.

3.3 Metrics

3.3.1 Feature Scaling

By feature scaling, independent features in the data are uniformly distributed over a predetermined range. Data are pre-processed in this way to deal with extremely variable magnitudes, values, or units. Machine learning algorithms prioritize larger values over smaller ones without feature scaling, regardless of the unit of measurement. In data pre-processing, it is also referred to as data normalization. An algorithm's calculations are sometimes accelerated by it.

$$z = \frac{x - \mu}{\sigma}$$

A wide range of magnitudes, units, and ranges are present in the real-world dataset. When a feature's scale is unimportant or deceptive, normalization should be used rather than normalization when the scale is meaningful. Magnitudes have an impact on algorithms that use Euclidean distance measures. Here, feature scaling aids in achieving parity in feature weighting.

Formally, if a feature in the dataset has a large scale relative to other features, it will become dominant in algorithms that measure Euclidean distance and will need to be normalized.

3.3.2 Imputation

Imputation is a method for estimating and filling in missing values in a data set. When errors in the data set are considered to be of no value in the correction process, the data values are set to missing and a plausible estimate is imputed. Alternatively, missing values may already exist in the data, and imputation may be performed to produce a complete dataset.

3.3.3 Dimensionality Reduction

Feature selection reduces the number of variables by selecting those that will help the model predict the target the most accurately. Up until a certain point, adding more features would aid the model's ability to make accurate predictions. The performance of the model would improve as the number of features used is increased; this is referred to as a Curse of Dimensionality. However, once the feature count has passed its peak, the performance will start to suffer. Because of this, features that are truly predictive should be chosen.

It is the goal of feature selection and dimensionality reduction techniques to reduce the number of features, but they are fundamentally different. While dimensionality reduction creates a projection of the data and generates entirely new input features, feature selection chooses which features to keep or remove.

3.3.4 Confusion Matrix

A supervised learning algorithm's performance is tabulated and displayed using a confusion matrix. An error matrix is another name for it. While the columns represent the categories in the actual response, each row represents the various categories in the predictions.

3.3.5 AUC

For classification issues at various threshold settings, the AUC-ROC curve can be used as a performance indicator. Separability is measured by AUC, and probability is measured by ROC. A model's ability to distinguish between classes is indicated by this parameter. The higher the AUC, the more accurate the model is at classifying 0 classes as 0, and 1 classes as 1. To represent the ROC curve, TPR and FPR are plotted on the y-axis and x-axis, respectively in Figure 9.

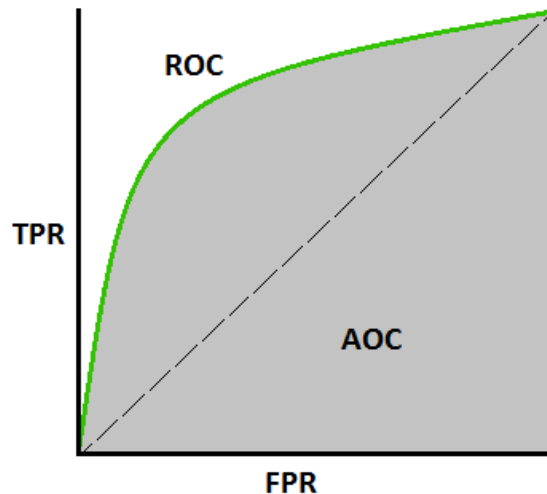


Figure 9. Area under the ROC Curve

3.3.6 Accuracy

The effectiveness of machine learning models for classification is mainly assessed using these three metrics. These values are computed using the following terms [2],

True positive (TP) = Number of positive cases correctly classified as True

False positive (FP) = Number of negative cases incorrectly classified as True

True negative (TN) = Number of negative cases correctly identified as False

False negative (FN) = Number of positive cases incorrectly identified as False Accuracy,

Proportion of true positive and true negative = $(TP+TN)/(TP+TN+FP+FN)$

3.4 Tools Used

3.4.1 Python as a programming language

Guido van Rossum developed Python in the late 1980s as a general-purpose programming language. In data science and engineering, it is widely used and has a large and expanding community of users. Python can be written as both object-oriented code and functional and procedural code.

The following are some key benefits of using Python over other programming languages:

Simple – Comparatively speaking, Python is a very straightforward and user-friendly programming language.

Compatibility – Python can be used effectively on a variety of hardware and operating systems at the same time because it is compatible with such a wide range of them.

Open Source – An open-source programming language is Python. This indicates that it is open source and has a large user base, both of which have sped up the language's evolution.

Object Oriented – Python's object-oriented nature facilitates the creation of server-side and web-based applications by programmers.

Support for multiple libraries – Over the years, Python has been used to create many libraries focused on data science and machine learning. Among the most popular libraries are TensorFlow, Pandas, NumPy, SciPy, NumPy, and Scikit-Learn. The libraries are easy to use and can be easily integrated into existing applications.

It is very easy to install packages with Python package management tool, also known as PIP. Moreover, Python allows us to create separate virtual environments for each of the applications, which makes it easier for us to maintain separate package.

3.4.2 Python Libraries

There are many different Python libraries used in this project. While some of these libraries can be installed automatically using the Python interpreter, others require manual installation through the Python PIP package manager.

Pandas: Pandas is a must-have library for data scientists. Machine learning library that offers flexible high-level data structures and a variety of analysis tools. The program simplifies the process of manipulating, cleaning, and analyzing data. Pandas supports sorting, re-indexing, iteration, concatenation, data conversion, visualization, and aggregate operations.

Numpy: Numpy is short for "Numerical Python." It's the one that gets used the most. Large matrices and multi-dimensional data are supported by this well-liked machine learning library. It has built-in mathematical functions for quick calculations. Even libraries like TensorFlow employ Numpy internally for a number of tensor operations. One of the main characteristics of this library is the Array Interface.

OS: Python's OS module provides tools for communicating with the operating system. Python's standard utility modules include OS. This module provides a portable method to access operating system-specific functionality. The os and os.path modules provide numerous

functions to interact with the file system.

Tensorflow: TensorFlow is an open-source software library. Google's Brain Team developed TensorFlow as part of its Machine Intelligence research organization to conduct machine learning and deep neural network research, but the system is sufficiently general to be useful in other domains as well.

Keras: Numpy's utility library offers functions for working with arrays. By using the method `to_categorical()`, a numpy array (or) vector of integers representing various categories can be transformed into a numpy array (or) matrix with binary values and columns equal to the number of categories.

Scikit-learn: Scikit-learn is an open-source Python library that provides a unified interface for machine learning, pre-processing, cross-validation, and visualization algorithms.

Scikit-salient learn has the following characteristics:

- Data mining and data analysis tools that are easy to use and effective. These algorithms include support vector machines, random forests, gradient boosting, k-means, and other classification, regression, and clustering algorithms. Easily accessible and reusable in a variety of contexts.
- Based on Matplotlib, SciPy, and NumPy
- Commercially usable, open source - BSD license.

Plotly: An open-source library called the Python Plotly Library can be used to quickly and easily visualize data and comprehend it. Plotly supports a number of different plot types, including line charts, scatter plots, histograms, and cox plots.

Any outliers or anomalies in a large number of data points can be identified by using Plotly's hover tool capabilities.

- It appeals to a wide range of audiences.
- Graphs can be fully customized, which enhances their significance and readability for others.

MinMaxScaler: MinMaxScaler scales all data features in the range $[-1, 1]$ if there are negative values in the dataset. This scaling compresses all inliers in the restricted range $[0, 0.005]$. StandardScaler does not ensure balanced feature scales in the presence of outliers,

due to the impact of the outliers during the computation of the empirical mean and standard deviation. As a result, the range of feature values is reduced.

Matplotlib: Python's Matplotlib library is fantastic for plotting arrays in 2D. Based on NumPy arrays, Matplotlib is a multi-platform data visualization library that works with the larger SciPy stack. John Hunter presented it for the first time in 2002. Visualization gives us access to vast amounts of data in easy-to-understand forms. In Matplotlib, can plot lines, bars, scatters, histograms, etc.

Seaborn: Statistical graphics can be plotted using Python's Seaborn visualization library. The default styles and color schemes enhance the appeal of statistical plots. It is built on top of the Matplotlib library and tightly integrated with Pandas data structures. Seaborn puts visualization at the center of data exploration and comprehension. It provides dataset-oriented APIs that allow us to switch between various visual representations for the same variables to better understand the dataset.

Chapter 4

Data Exploration and Analysis

The exploratory data analysis phase follows the preparation of the business hypothesis. Input data used to train the model determines the model outcome. In exploratory data analysis, attributes are identified, data is pre-processed, and features are engineered. Aspect identification involves identifying predictor variables (inputs) and target variables (outputs), along with their data types (string, numeric, or datetime), and classifying features into continuous and categorical variables, which aids the algorithm in applying the proper treatment to the variable while building the model. As a part of data pre-processing, missing values and outliers are identified and filled in by computing the mean or median for quantitative attributes and the mode for qualitative attributes. Outliers' tendency to increase the mean and standard deviation can be eliminated by using the natural log value, which lessens variation caused by extreme values.

4.1 Data Collection

In this dataset, data is collected from heavy Scania trucks in everyday use. The Air Pressure System (APS) generates pressurized air for various functions in a truck, including braking and gear changes. This dataset contains component failures for a specific component of the APS system. In the negative class, trucks with failures related to components other than APS are included. Experts selected a subset of all available data. Scania CV AB released the dataset to the UCI Machine Learning Repository. There are 60000 rows in the training set, 1000 of which belong to the positive class, and 171 columns, one of which is the Class column. All the attributes are numeric, except Class that is a Boolean. There are features with up to 81% of missing values (0's and NaN's). Almost all features have sparse Na's and 0s.

4.1.1 Data types of features

Data types are a method of categorizing variables that specify the kinds of values they can store and the kinds of mathematical, relational, or logical operations that can be performed on them without producing errors. Knowing the appropriate datatypes for independent and dependent variables is crucial for machine learning. Considering That It Provides the Foundation for Classification. As a result, the wrong data types are identified, resulting in incorrect modeling, which leads to the wrong solution.

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 60000 entries, 0 to 59999  
Columns: 171 entries, class to eg_000  
dtypes: float64(169), int64(1), object(1)  
memory usage: 78.3+ MB
```

Figure 10. Data type of the Training Data

Major conclusions are all features are numeric values. There are around 171 entries which are the column values in the data set as mentioned in Figure 10. There are totally 169 columns with float64(169) and one int64(1) column that is the binary classification class column.

Anything that is represented by numbers is referred to as numerical data (floating point or integer). Everything else is typically referred to as categorical data, and discrete labelled groups in particular are frequently singled out. These two main classifications, categorical and numerical. Here, one column will be used for binary classification, and the other numerical columns.

4.1.2 Null Ratio

Many datasets from the real world have missing values. In machine learning models, missing values may skew the results and/or lower their accuracy. Data that are not stored (or not present) for one or more variables in the provided dataset are referred to as missing data. Machine learning, the mainstay of data analysis and information extraction, frequently encounters missing values. There are a number of reasons why missing values occur, including completely random missing, random missing, or not at all random missing. A system failure during data collection or a human error during data pre-processing could cause all of these errors. It is important to deal with missing values before analysing the data, since doing so may lead to inaccurate or biased results. Literature has provided a number of proposals for handling missing values. Missing values are indicated by blanks in the dataset. NaN is typically used to represent missing values in Pandas. Some values in the data may be missing for a variety of reasons. Methods for handling missing data depend on the causes of the missing data. Therefore, it's important to understand the potential causes of missing data. Some of the causes are listed below:

- Poor maintenance could result in corrupted historical data.
- Observations are not recorded for some fields. Values could be missed due to human error.

There is a high null ratio in the data set, mentioned in below Figure 11. Many columns have null values, as can be seen. The data analysis can be performed by dropping the columns with a high null ratio.

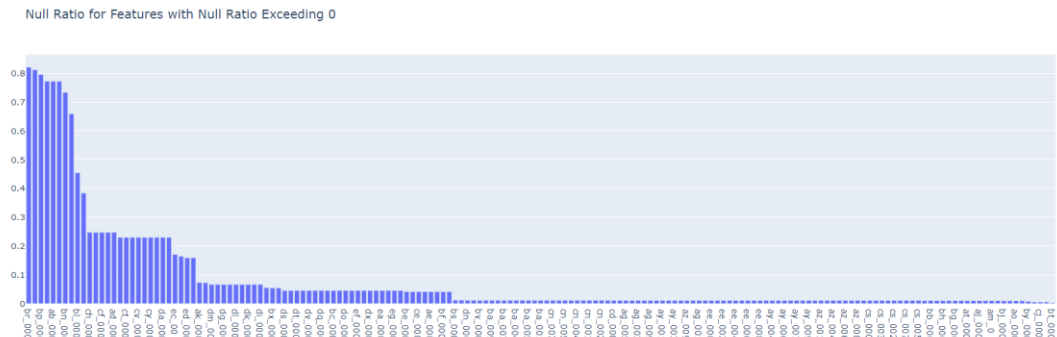


Figure 11. Null Ratio for features

4.1.3 Feature Class Imbalance

Machine learning and statistics use classification as a supervised learning technique in which the computer program makes new observations or classifications based on the data it is provided with. Approximating the mapping function between discrete input variables and output variables is called classification predictive modeling. Using training data, the Classification algorithm classifies new observations. Classification involves using a dataset or provided observations to learn how to classify new observations. Labels, targets, and categories can be used to describe classes. The main objective is to identify the class or category in which the new data belongs. Each piece of input data is categorized using an algorithm called classification. Based on the input data, the classification model predicts the class or category for the training data, which also draws conclusions. Characteristics of a phenomenon under observation can be quantified as features. There are only two possible outcomes for binary classification, such as true or false.

Both structured and unstructured data can be classified. A classification is the process of categorizing data into categories. The first step in the process is to predict the class of the data points provided. Classes are often referred to as targets, labels, and classes. Classification predictive modeling involves approximating the mapping function from discrete input variables to output variables. The main objective is to determine which category or class the new data belongs to. Classification involves predicting the class of a set of data points. Classes

are also known as targets, labels, and categories. Classification predictive modeling (y) involves estimating a mapping function (f) from input variables (X) to discrete output variables (Y). In supervised learning, the input data are also available to the targets during classification.

Classification involves recognizing, comprehending, and categorizing things into predetermined categories. With the help of these pre-categorized training datasets, machine learning programs classify upcoming datasets into appropriate and pertinent categories. The likelihood or probability that the data that follows will fall into one of the predetermined categories is determined by machine learning classifiers using training data.

There is a great deal of imbalance in the class column. The following graph shows the percentage distribution of positive and negative classes in the training set. In the Figure 12 below, Class 0 (in blue) represents a negative class and Class 1 (in red) represents a positive class.

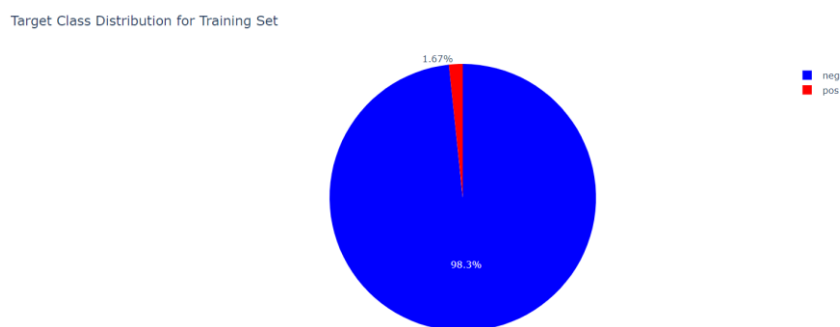


Figure 12. Target class distribution for Training Set

4.1.4 NaN and Zero Analysis

The numeric data type NaN (Not a Number) designates an undefined value or a value that cannot be represented, particularly the outcomes of floating-point calculations. For instance, NaNs can be the square root of a negative, infinity, the result of dividing by zero, or missing values (which is imaginary, whereas a floating-point number is real). The missing value in a cell is represented by the special value NaN, or Not a Number, in DataFrame and numpy arrays. They are also depicted in programming languages; for instance, in Python, they are depicted as values that are None. Due to the fact that None (or NaN) values signify a lack of a value, might believe that they are simply zeroes. However, there is a catch: the difference between

zero and None is that zero is a value (for instance, an integer or float), whereas None denotes the absence of that value.

NaN values are risky in two different ways.

- Changing some metrics' mean or median values, which results in scientists receiving inaccurate information.
- Algorithms implemented in Sklearn cannot operate on datasets with such values.

There are several remedies:

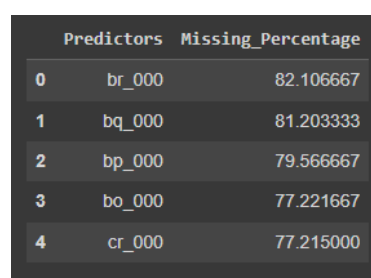
- To remove any rows with NaN values. However, this is not a wise decision because, especially when working with small datasets, information is lost in this manner.
- To use particular techniques or values to impute NaN values.

There are many ways to impute these gaps, but these are the ones that are used the most frequently:

- Assign specific values to input.
- Impute using unique metrics, like mean or median.
- MICE or KNN are two techniques for impute.

By examining each method's operation and its impact on the dataset, the predictors which have highly missing data are listed out. It was found the below columns have highest count of missing data, for which the missing value percentage is listed below Figure 13. There are features with up to 82% of missing values (0's and NaN's). The following aspects can be considered to further analysis of the missing data:

- Removing columns with more than 35% of missing values.
- For the rest of NaNs and 0s, replacement (median) values can be used.
- Feature scaling can be used to scale the columns on the data.
- For NaNs an imputation algorithm can be used.



A screenshot of a table with a dark background and light-colored text. The table has two columns: 'Predictors' and 'Missing_Percentage'. It lists five predictors with their corresponding missing percentages.

	Predictors	Missing_Percentage
0	br_000	82.106667
1	bq_000	81.203333
2	bp_000	79.566667
3	bo_000	77.221667
4	cr_000	77.215000

Figure 13. Predictors in the data set with highest missing percentage of NaNs

4.1.5 Feature Correlation

Data quality determines the quality of machine learning models. By pre-processing and cleaning the data to only include features that will have the greatest impact on the final model. Feature selection is a method of choosing the attributes that will increase the accuracy of the predicted variable or removing the irrelevant attributes that will reduce the model's quality and accuracy. Through data correlation, can discover the relationships between variables and attributes in the dataset. The dataset's features may not all be useful in creating a machine learning model that can make the required prediction. Predictions may even be inaccurate if some of the features are utilized. When creating a machine learning model, selecting the right features is crucial. Correlation can teach some things, such as:

- One or more attributes may be a result of, or dependent upon, another attribute.
- With other attributes, one or more attributes may be connected.

The statistical concept of correlation, which is frequently used to describe how nearly linear a relationship exists between two variables. The main reasons for considering correlation is,

- One attribute can be predicted from another using correlation (Great way to impute missing values).
- Correlation can (on occasion) show that a causal connection exists.
- A fundamental quantity for many modeling techniques is correlation.

This evaluation formula is combined with a heuristic search strategy and a correlation measure. The figure 14 shows the feature correlation of the data.

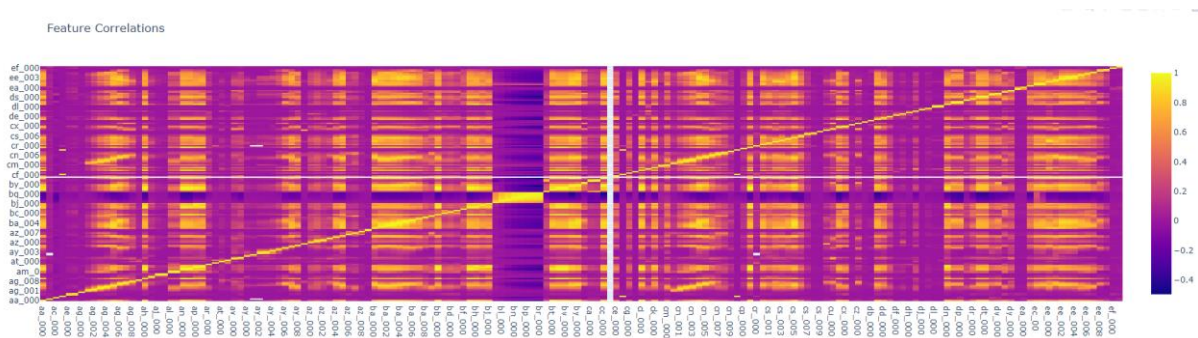


Figure 14. Feature Correlations

Chapter 5

Data Preparation & Cleansing

The pre-processing involved data cleaning process such as removing the larger missing data columns, imputation, feature scaling. The following steps were performed in pre-processing the data:

5.1 Target class distribution

Machine learning refers to algorithms that continuously pick up new information from different examples and then apply it to real-life problems. A machine learning task assigns a label value to a class and then determines whether a given type belongs to that class. Typically, classification problems involve predicting a specific type of class label from input data. To determine which configuration and algorithm will deliver the best performance for a given task, there is no overarching theory. Classification predictive modeling compares the outputs of various algorithms. The classification accuracy of a model can be used to assess its performance based on the various predicted class labels. Even though it may not be the best parameter, classification accuracy is a good starting point for most classification tasks.

Classification predictive modeling predicts a class label for an observation. An imbalanced classification problem is one in which the distribution of examples among the recognized classes is biased or skewed. It is possible for the distribution to be mildly biased or severely unbalanced. Predictive modeling is challenged by imbalanced classifications because most machine learning algorithms assume that each class should have an equal number of examples. Thus, models perform poorly in terms of prediction, particularly for minority groups. The issue is more sensitive to errors in classification for the minority class than for the majority class because the minority class is generally more significant. An unequal distribution of examples between classes in a classification predictive modeling problem. There is an inherent data imbalance in the real world. Long-tailed skewed distributions are common in data.

Based on the binary column of the training data set, the class distribution is calculated. A value of 0 corresponds to a negative value, a value of 1 corresponds to a positive value. Brakes that fail due to air pressure systems are represented by one, while brakes that fail due to other components are represented by zero. This data set is highly imbalanced, as the negative value

is around 59000 and the positive value is 1000. The class imbalance can be handled by resampling.

```
neg    59000  
pos     1000  
Name: class, dtype: int64
```

Figure 15. Target Class Distribution

5.2 Missing Value Analysis

Outliers and missing values are frequently found during the data collection phase of observational or experimental studies. There are many reasons for missing values, including data loss, participant dropouts, and nonresponses. As a result of missing values, the sample size is smaller than intended, which adversely affects the validity of the study's findings. It can also lead to biased results when conclusions about a population are based on such a sample, undermining their validity. During the pre-treatment process, missing data are either ignored for simplicity's sake or substituted with values calculated statistically. Missing data analysis takes into account the complexity of handling missing data, as well as the bias between missing and observed values. Second, outliers are extreme values that deviate from a distribution's typical pattern. Data entry errors and incorrect participant responses can cause outliers. Data points that are far from the majority of the other data points in a distribution of variables are known as outliers. As a result of outliers in sample data, statistical estimates like mean values are either underestimated or overestimated. It is essential to deal with outliers before analysing the data set containing the outlier. As a result, outliers must either be modified after their sources have been found or substituted. Outliers and missing values can significantly alter data analysis outcomes depending on how they are handled. Analysing missing data and outliers requires the proper handling.

Normalization is a data preparation technique frequently used in machine learning. In normalization, all columns in a dataset are scaled to the same value. There are different types of data normalization. Consider a dataset X with N rows (entries) and D columns (features). Features i and j are represented by $X[:,i]$ and entry j by $X[j,:]$. The data range is rescaled to $[0,1]$ using the min max normalization. In normalization, features are scaled down to a similar size. As a result, the model's functionality and training stability are enhanced.

$$\hat{X}[:, i] = \frac{X[:, i] - \min(X[:, i])}{\max(X[:, i]) - \min(X[:, i])}$$

Many machine learning algorithms may encounter missing values in datasets. Most machine learning models do not handle missing value data. Before modeling the prediction task, should find and replace any missing values in each column of the input data. This is called imputing or imputation of missing data. Imputation of data involves calculating a statistic for each column (such as mean or median) and substituting that statistic for all missing values in that column. The method is popular because the statistic is easy to compute using the training dataset and it usually performs well.

The missing values are imputed using median imputation, which uses the median value across the entire feature column. When the data is skewed, it is wise to think about replacing the missing values with the median value. Keep in mind that only numerical data can be used to impute missing data using the median value. Simple to use and quicker than other methods of filling in the missing values in the entire dataset is median imputation.

The scikit-learn class SimpleImputer is useful for handling missing data in the dataset. In place of NaN values, it inserts a placeholder. It is implemented using the SimpleImputer() method, which accepts the following arguments.

SimpleImputer(missing_values, strategy, fill_value)

missing_values: The placeholder that needs to be filled in with missing values. NaN by default.

strategy: The information that will be used to replace any NaN values in the dataset. The values for the strategy argument are mean (the default), median, most frequent, and constant.

fill_value : the constant value that will be applied using the constant strategy to NaN data.

Almost all the features have sparse NaNs and Zeros. Removing columns with more than 35% of missing values has been tried on both the training and test data. The min max scaler is used for the normalization, and the scaling to unit variance is performed. Data frame is scaled on data sets. For the rest of NaNs and 0s, replacement (median) has been performed. Replacing with median value most of the times. For this an imputation of missing values using

sklearn.impute.SimpleImputer has been performed. Once it is performed, the predictors are around 160 entries.

```
#Dropping Columns that have more than 35% missing values
dropped = list(miss_val.loc[miss_val['Missing_Percentage'] >
35, 'Predictors'])

tr_features.drop(columns = dropped, inplace = True)
tst_features.drop(columns = dropped, inplace = True)


#Feature Scaling
scaler = MinMaxScaler()
scaler.fit(tr_features)

tr_features = pd.DataFrame(scaler.transform(tr_features),
columns=tr_features.columns)

tst_features = pd.DataFrame(scaler.transform(tst_features),
columns=tst_features.columns)


#Imputing Missing Values
imp = SimpleImputer(strategy='median', missing_values=np.nan)
imp.fit(train_features)

tr_features = pd.DataFrame(imputer.transform(tr_features),
columns=tr_features.columns)

tst_features = pd.DataFrame(imputer.transform(tst_features),
columns=tst_features.columns)
```

5.3 Correlation

In statistics, correlation measures how closely two variables are related linearly. This technique is commonly used when describing straightforward connections without explicitly stating cause and effect. This bivariate analysis measure explains the relationship between variables. In most business situations, it is helpful to discuss a subject in terms of its broader context. If two variables are highly correlated, prediction can be done one from the other. With correlation, it is easier to identify the variables that are dependent on other variables. A number of modeling techniques are based on it. Data comprehension is improved by an accurate correlation analysis. Understanding causal relationships is aided by correlations.

There may be a positive correlation between two characteristics (variables). As one variable rises, the value of the other variable or variables rises as well. There can be a negative correlation between two characteristics (variables). As one variable's value rises, the value of the other(s) variable(s) falls. There is no correlation between two features (variables). The other variable(s) do not increase or decrease when the value of the first variable does. Sample correlation coefficients, or r , measure the strength of relationships. Correlations are also statistically significant. Based on the correlation matrix, the figure shows the heat map for feature correlation.

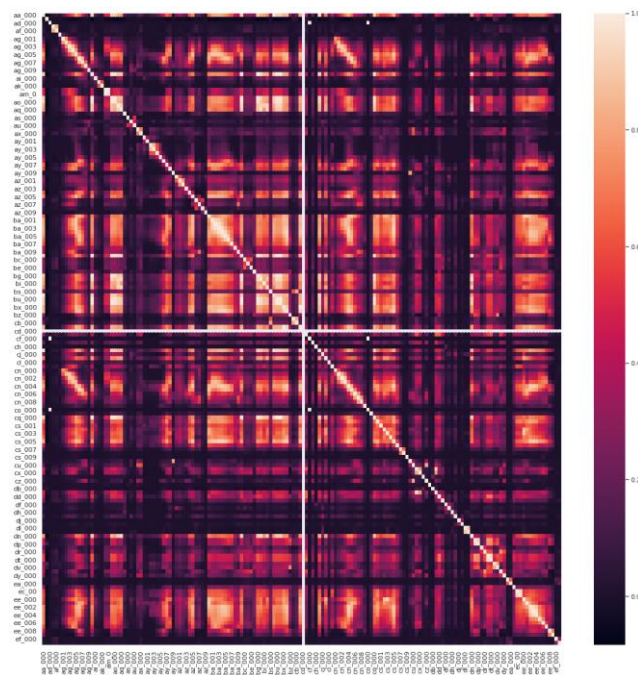


Figure 16. Correlation Matrix

To perform the further analysis, the remaining columns are removed by setting the correlation coefficient threshold at 0.8 to remove highly correlated features. The data set contains around 102 entries after removing correlated columns.

```
msk = np.triu(np.ones_like(corr_matrix, dtype=bool))
tr_df = corr_matrix.mask(mask)
to_drop = [c for c in tr_df.columns if any(tr_df[c] > 0.8)]
train_imp_features = train_features.drop(train_features[to_drop],
axis=1)
```

5.4 Dimensionality Reduction and balancing the data

Techniques for reducing the number of input variables in training data are referred to as "dimensionality reduction." There are frequently too many variables to consider when performing regression or classification tasks using machine learning. These parameters are also referred to as features. The "curse of dimensionality" states that the more features there are, the harder it is to model them. Data are essentially converted from a high-dimensional feature space to a low-dimensional feature space through the process of dimensionality reduction. Additionally, it is crucial that any valuable properties retained in the data after transformation do not disappear. Dimensionality reduction is frequently used in machine learning or deep learning techniques to facilitate the task at hand as well as in data visualization to comprehend and interpret the data.

SelectKBest selects the features according to the k highest score. The dimensionality reduction, statistical technique of reducing the amount of random variables in a problem by obtaining a set of principal variables is performed. Further 2500 samples are taken into consideration and the data is been balanced. This method is applied for both classification and regression data by modifying the 'score func' parameter. When large dataset is prepared for training, choosing the best features is a crucial process. It enables us to remove the less crucial portions of the data and shorten the training period. Using the feature selection technique, the data features that have the greatest impact on the target variable is selected. Top predictors are picked for the desired outcome. For feature selection/dimensionality reduction on sample sets, classes in the sklearn.feature selection module can be used to improve estimators' accuracy scores.

Reduces Overfitting: It is less likely that decisions will be based on noise or redundant data when there is less redundant data.

Improves Accuracy: Reducing misleading data increases modeling accuracy.

Reduces Training Time: Less data allows algorithms to learn more quickly.

```

#Dimensionality Reduction using SelectKBest
selectKBest = SelectKBest(chi2, k=160)
selectKBest.fit(train_features, train_labels)
best_train_features = selectKBest.transform(train_features)
idxs_selected = selectKBest.get_support(indices=True)
best_train_features = train_features.iloc[:,idxs_selected]
best_test_features = test_features.iloc[:,idxs_selected]

#Balancing the dataset
number_samples = 2500
idxs_pos = train_labels[train_labels==1].index
idxs_neg = train_labels[train_labels==0].sample(n=number_samples,
replace=False, random_state=0).index
idxs_balanced = np.concatenate((idxs_pos,idxs_neg))

best_train_features_balanced =
best_train_features.loc[idxs_balanced]

train_labels_balanced = train_labels.loc[idxs_balanced]
print(f'Proportion balanced: {number_samples/1000}/1')

```

5.5 Train/Validation/Test data

The model's training course is determined by the validation set metric. The machine learning model is assessed using the validation set at the end of each epoch. The corresponding loss terms are determined using the metrics from the validation set, and the hyperparameters are adjusted as a result. Data used for actual training is called training data. Validation split improves the model's performance by fine-tuning it after each epoch. The test set provides information about the model's overall accuracy after the training phase.

Training Set: This is a dataset that is fed into the model in order to learn about potential patterns and relationships. The dataset used to train the model is called the training set. The model uses this dataset to discover any underlying relationships or patterns that will allow for future prediction. The training set needs to be as representative of the population that is used to model as possible. Furthermore, must exercise caution and make sure that it is as impartial

as possible because any bias introduced at this point could be carried over into inference.

Validation Set: A variety of models and hyperparameter choices are used in this dataset to evaluate the model's performance. If the same exact dataset is used for both training and tuning, the model will overfit and be unable to generalize well. Nevertheless, comparing the accuracy of each model on the training set can be performed. The validation set comes into play here; it serves as a separate, unbiased dataset for contrasting the effectiveness of various algorithms trained on the training set.

Test Set: The dataset will be used by us as a test set to approximate the unbiased accuracy of the model. After the validation set has been used to determine the algorithm and parameter choices it can be used in production, the test set is used to estimate the models' true performance in the wild. It is the last step in assessing how well the model performs on unobserved data. In general, choosing a model should never, ever take into account the performance of the test set. A form of overfitting is peaking at the test set performance in advance, which will probably result in inaccurate performance expectations in production. The validation set should only be used to determine the best model, and then it should be checked as the final form of evaluation.

10% of the training data set is split into validation data sets. There are 16000 entries and 171 columns in the test data set as shown in Figure 17.

```
X_trn, X_tst, y_trn, y_tst =  
train_test_split(best_train_fea_balanced, tr_labls_balanced,  
tst_size=0.1, random_state=3)
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 16000 entries, 0 to 15999  
Columns: 171 entries, class to eg_000  
dtypes: float64(169), int64(1), object(1)  
memory usage: 20.9+ MB
```

Figure 17. Data type of test data

5.6 COST CALCULATION

Misclassification is measured by the challenge metric.

Total cost = Cost 1 * CM. FP + Cost 2 * CM. FN

Where, cost 1 = 10 and cost 2 = 500.

CM - Confusion Matrix

FP – False Positive

FN – False Negative

Machine learning models are needed to reduce misclassification costs. Predicting the cost is done using the formula above.

Chapter 6

Model Implementation

The use of machine learning (ML) in enterprise data analytics has increased significantly in recent years. To develop, test, deploy, and maintain enterprise-grade machine learning models in real-world settings, it is crucial to have an ecosystem in place. A machine learning model is built by gathering data from numerous reliable sources, processing it to make it fit for modeling, selecting a modeling algorithm, building the model, computing performance metrics, and selecting the best model. Model maintenance is crucial after the model has been put into production. Keeping a machine learning model current and relevant in accordance with changes in source data is necessary since the model may eventually become out-of-date. ML model configuration management becomes increasingly important as the number of models increases. In this article, can discuss principles and practices in the industry, including tools and technologies used for the creation, deployment, and maintenance of ML models in enterprises. "ML model lifecycle" refers to a process that begins with the identification of the source data and continues with the development, deployment, and maintenance. In general, ML Model Development and ML Model Operations can be divided into two categories. Figure 18 below shows the steps involved in developing a machine learning (ML) model.

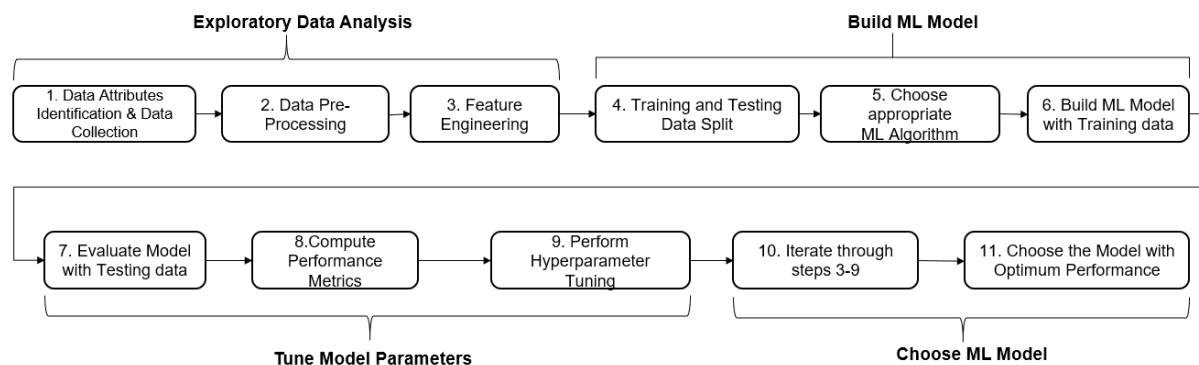


Figure 18. Steps involved in developing a machine learning model.

In the lifecycle of an ML model, data exploration, model building, tuning hyperparameters, and model selection are the steps. Using machine learning, a technique for data analysis, analytical models can be automatically created. Computers can learn from data, recognize patterns, and make decisions without much assistance from humans in this area of artificial

intelligence. As more data is generated, machine learning programs automatically adapt to take into account both the new data and previous processes.

The data must be divided into two sets, such as a "training set" and a "validation set," with a 90:10 ratio. Models are built using the training dataset, and predictions are made using the test dataset. To achieve better performance, deep learning (neural networks) models are preferred over regression models (ML models). The Activation Function (AF) adds an additional layer of nonlinearity to these models. Ensemble learning, model selection. The accuracy of tests is lower than that of validations. The model's hyperparameters have been fine-tuned for the validation dataset, explaining this. For the specified samples, neural network models were used to predict the cost of maintenance for braking in Scania trucks.

Models considered in neural networks are,

- 1 Basic neural network
- 2 Recurrent Neural Network
- 3 Multi-Layer Perceptron
- 4 Artificial neural network [layers 60,30,15 ,1]
- 5 Artificial neural network [layers 64,32,16 ,1]

6.1 Basic Neural Network

The neural network model is used for its fault tolerance and parallel processing capabilities. In a nutshell, a neural network represents how the human brain functions. In order to function, it simulates a large number of connected processing units that resemble abstract representations of neurons. Processing units are arranged in layers. A neural network typically consists of three layers, an input layer with units that represent input fields, a hidden layer, and an output layer with units that represent target fields. Different connection strengths (or weights) exist between the units. Every neuron in the subsequent layer receives values propagated from the neuron in the previous layer. Eventually, a result is delivered by the output layer. Any answers found online are probably absurd since all weights are initially random. The network learns new things through training. Responses provided by the network are compared with known outcomes for examples for which the output is known. As a result of this comparison, the network gradually adjusts its weights based on the feedback it receives. With increasing training, the network replicates the known outcomes more accurately. When trained, the

network can be used in future cases with uncertain outcomes. The process was done with the following parameters:

- epochs=10
- batch_size=10
- activation='relu'
- loss='binary_crossentropy'
- optimizer='adam'

6.1.1 Without Removing Correlated Columns

	precision	recall	f1-score	support
0	0.95	0.97	0.96	253
1	0.91	0.88	0.89	97
accuracy			0.94	350
macro avg	0.93	0.92	0.93	350
weighted avg	0.94	0.94	0.94	350
Total cost is: 6080.0				

Figure 19. Basic Neural Network model for Validation Data

The above Figure 19 shows the results of model for the validation data. Positive predictions are more accurate when their precision is high which is 0.95 for class 0, 0.91 for class 1. A recall also called sensitivity, or hit rate is the percentage of accurately identified positive predictions 0.97 for class 0 and 0.88 for class 1. The F measure, or f1-score, i.e., 0.96 for class 0, 0.89 for class 1 measures precision and recall simultaneously by finding its harmonic mean. An accurate prediction is accurate across all predictions with 0.94. The cost for the validation data set is predicted to be 6080.

	precision	recall	f1-score	support
0	1.00	0.97	0.98	15625
1	0.40	0.95	0.56	375
accuracy			0.97	16000
macro avg	0.70	0.96	0.77	16000
weighted avg	0.98	0.97	0.97	16000
Total cost is: 14810.0				

Figure 20. Basic Neural Network model for Validation Data

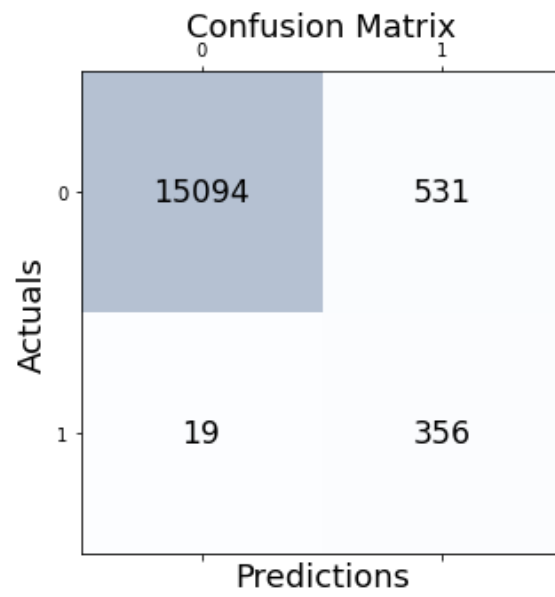


Figure 21. Confusion Matrix for Basic Neural Network model

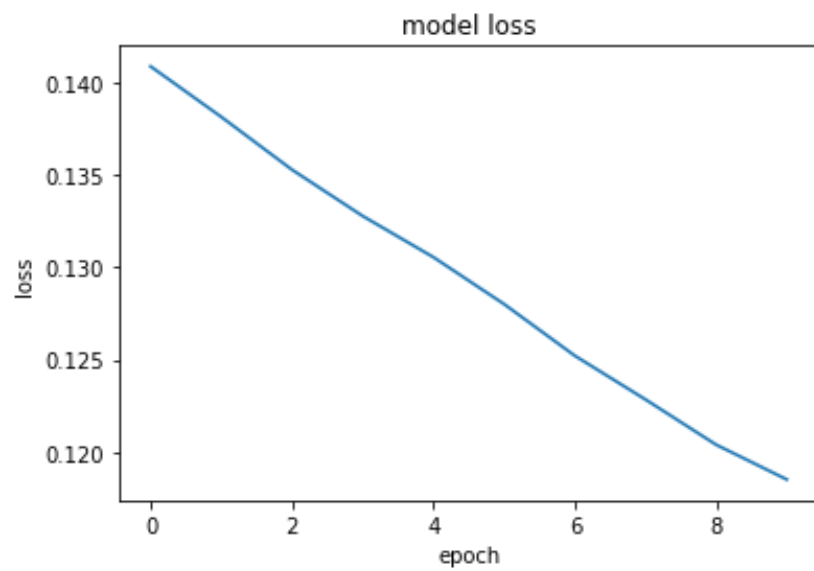
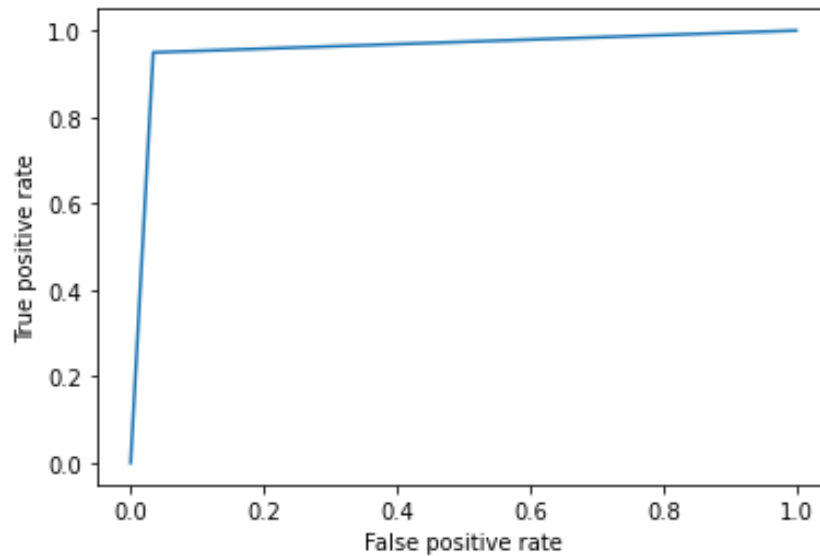


Figure 22. Loss Curve Diagram for Basic Neural Network model



AREA UNDER ROC: 0.9576746666666667

Figure 23. ROC curve for the Basic Neural Network model

The above details provide the modeling results for the test data shown in Figure 20-22. Precision is less for classes due to the imbalance in data. However, this might not be so bad if the false positive is cheap. A recall also called sensitivity, or hit rate is the percentage of accurately identified positive predictions 0.97 for class 0 and 0.85 for class 1. The F measure, or f1-score, i.e., 0.98 for class 0, 0.56 for class 1 measures precision and recall simultaneously by finding its harmonic mean. An accurate prediction is accurate across all predictions with 0.97. Support is the number of occurrences of each class Total maintenance cost from this model is 14810. Here, the misclassification rate is 0.034. A loss function represents one training example. Sometimes it is referred to as an error function. Increasing epochs resulted in a general decrease in loss and an increase in accuracy. Probability is measured by the ROC, while separability is measured by the AUC. A model's ability to discriminate between classes is reflected in this measurement. Increasing AUC increases the likelihood that 0 classes will be predicted as 0 and 1 classes will be predicted as 1. In Figure 23 area under the curve is 0.9577.

6.1.2 Removing Correlated Columns

	precision	recall	f1-score	support
0	0.94	0.97	0.95	253
1	0.91	0.82	0.86	97
accuracy			0.93	350
macro avg	0.92	0.90	0.91	350
weighted avg	0.93	0.93	0.93	350
Total cost is: 8580.0				

Figure 24. Basic Neural Network model for Validation Data

For the validation data, the results of the modeling are shown above in Figure 24. Positive predictions with a high precision, such as 0.94 for class 0, 0.91 for class 1, are more accurate. Recall is also known as sensitivity, or hit rate, and it is 0.97 for class 0 and 0.82 for class 1. Precision and recall are simultaneously measured using the F measure, or f1-score, which is 0.95 for class 0 and 0.86 for class 1. The average accuracy of an accurate prediction is 0.93. The validation data set is estimated to cost 8580.

	precision	recall	f1-score	support
0	1.00	0.97	0.98	15625
1	0.39	0.94	0.56	375
accuracy			0.96	16000
macro avg	0.70	0.95	0.77	16000
weighted avg	0.98	0.96	0.97	16000
Total cost is: 15950.0				

Figure 25. Basic Neural Network model for Test Data

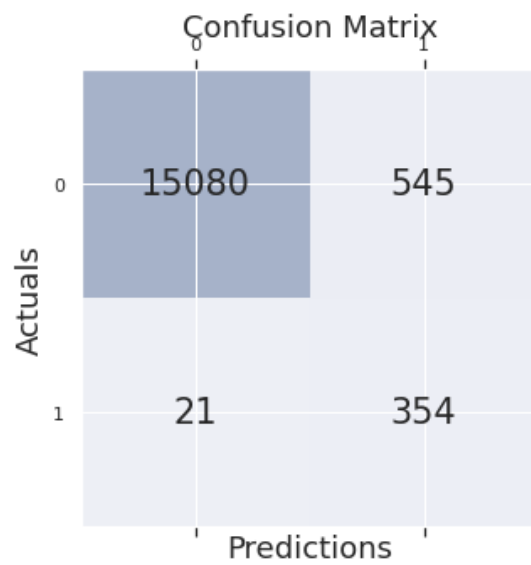


Figure 26. Confusion Matrix for Basic Neural Network model

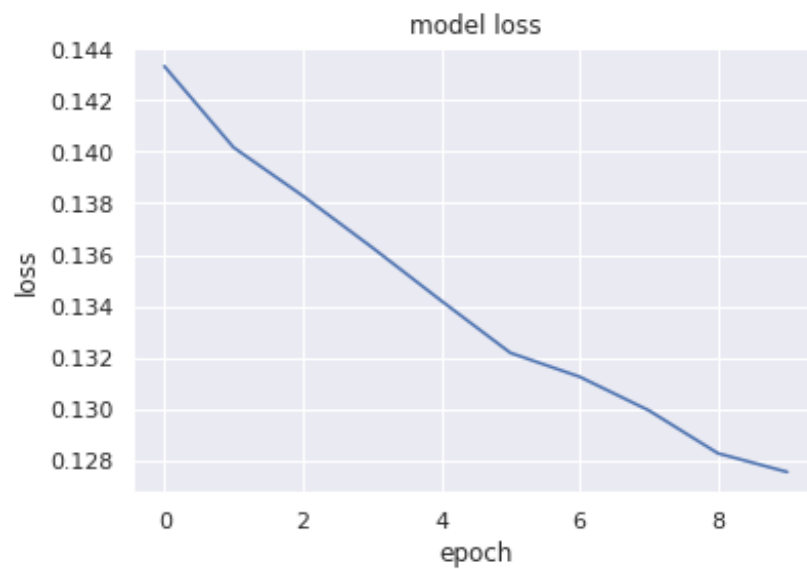
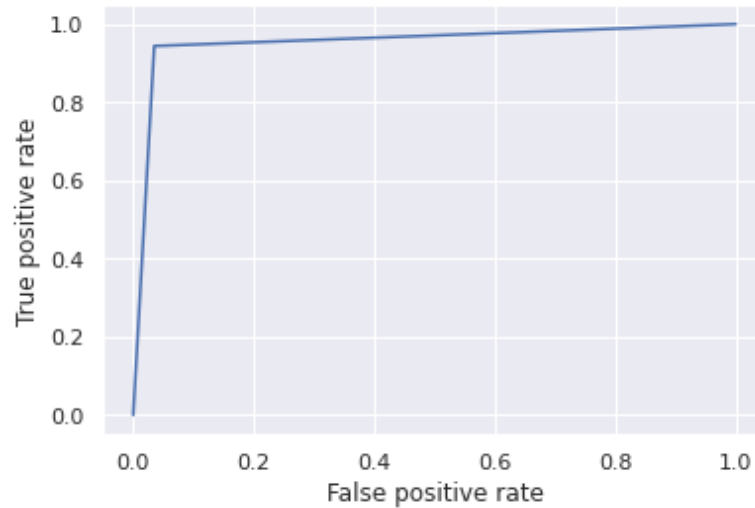


Figure 27. Loss Curve Diagram for Basic Neural Network model



AREA UNDER ROC: 0.95456

Figure 28. ROC curve for the Basic Neural Network model

For the test data, the results of the modeling are shown above Figure 25-28. Classes have less precision due to the imbalance in the data. However, if the false positive is inexpensive, this might not be a bad thing. For class 0, the recall is 0.97, while for class 1, it is 0.94. The F measure, or f1-score, equals 0.98 for class 0 and 0.56 for class 1. Across all predictions, an accurate prediction has an average accuracy of 0.96. The total maintenance cost of this model is 15,950. This case has a misclassification rate of 0.035. As AUC increases, it is more likely that 0 classes will be predicted as 0, and 1 classes will be predicted as 1. The area under the curve is 0.9546.

6.2 Recurrent Neural Network

As the only neural network with an internal memory, RNNs are one of the most promising neural networks currently available. In comparison to other algorithms, recurrent neural networks can develop a much deeper understanding of a sequence and its context. For modeling sequence data, recurrent neural networks (RNN) are useful. The behaviour of RNNs, which are derived from feedforward networks, is similar to that of human brains.

In RNN, each layer's output is saved and fed back into the system's input to predict the output of that layer. RNNs can handle sequential data, accepting inputs from the past as well as inputs from the present. The internal memory of RNNs allows them to remember previous inputs. The recurrent neural network will standardize the activation functions, weights, and biases of each hidden layer in order to ensure that each hidden layer has the same parameters. Instead of

creating multiple hidden layers, it will create one hidden layer and loop over it as many times as necessary.

An RNN can model a collection of records so that each pattern is dependent on the previous one. The process was done with the following parameters:

- Hidden units=2
- Epochs=30
- Loss='mse'
- Optimizer='adam'

6.2.1 Without Removing Correlated Columns

	precision	recall	f1-score	support
0	0.94	0.95	0.95	253
1	0.87	0.85	0.86	97
accuracy			0.92	350
macro avg	0.91	0.90	0.90	350
weighted avg	0.92	0.92	0.92	350
Total cost is: 7620.0				

Figure 29. Recurrent Neural Network model for Validation Data

The results of the modeling are shown above Figure 29 for the validation data. When a positive prediction's precision is high, which is 0.94 for class 0 and 0.97 for class 1, it is more accurate. The percentage of correctly identified positive predictions with a recall of 0.95 for class 0 and 0.85 for class 1 is known as sensitivity. By determining its harmonic mean, the F measure, or f1-score, which is 0.95 for class 0 and 0.86 for class 1. An accurate prediction has an average accuracy of 0.92 across all predictions. Finally, it is estimated that the validation data set will cost 7620.

	precision	recall	f1-score	support
0	1.00	0.93	0.96	15625
1	0.23	0.93	0.37	375
accuracy			0.93	16000
macro avg	0.61	0.93	0.67	16000
weighted avg	0.98	0.93	0.95	16000
Total cost is: 24150.0				

Figure 30. Recurrent Neural Network model for Test Data

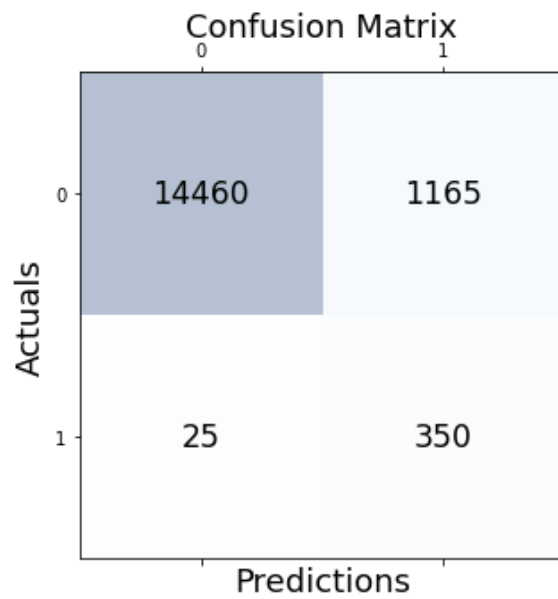


Figure 31. Confusion Matrix for Recurrent Neural Network model

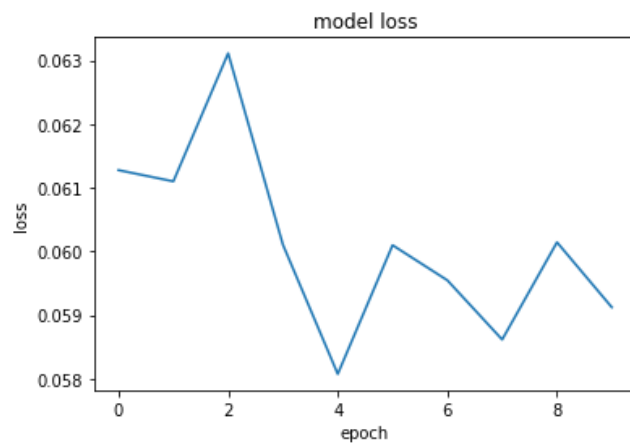
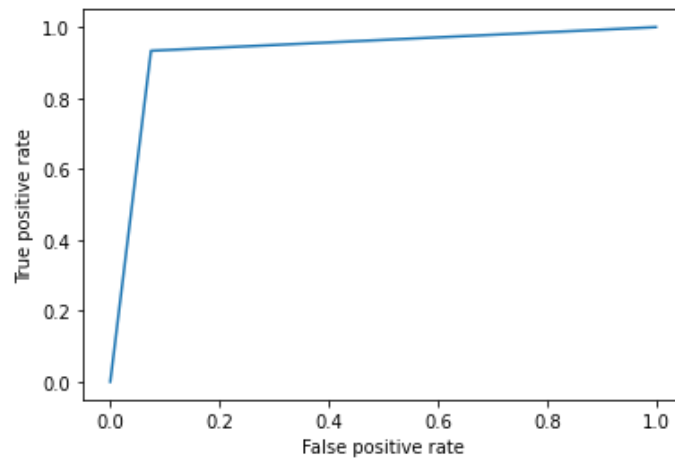


Figure 32. Loss Curve Diagram for Recurrent Neural Network model



AREA UNDER ROC: 0.9293866666666667

Figure 33. ROC curve for the Recurrent Neural Network model

The results of the modeling are shown above Figure 30-33 for the test data. The percentage of correctly identified positive predictions (0.93 for class 0 and 0.93 for class 1) is known as recall, also known as sensitivity, or hit rate. By determining its harmonic mean, the F measure, or f1-score, which is equal to 0.96 for class 0 and 0.37 for class 1, simultaneously measures precision and recall. An accurate prediction has an average accuracy of 0.93 across all predictions. Support is the total number of each class instances. This model has a total maintenance cost of 24,150. The misclassification rate in this case is 0.074. The ROC and AUC measure probability and separability, respectively. It is more likely that 0 classes will be predicted as 0, and 1 classes will be predicted as 1, as AUC increases. AUC in this instance is 0.9294.

6.2.2 Removing Correlated Columns

	precision	recall	f1-score	support
0	0.90	0.97	0.93	253
1	0.90	0.72	0.80	97
accuracy			0.90	350
macro avg	0.90	0.85	0.87	350
weighted avg	0.90	0.90	0.90	350
Total cost is: 13580.0				

Figure 34. Recurrent Neural Network model for Validation Data

The modeling outcomes for the validation data are shown in the aforementioned specifics. Positive predictions have a higher accuracy when their precision is high (0.90 for class 0 and 0.90 for class 1) The proportion of correctly identified positive predictions with a recall of 0.97 for class 0 and 0.72 for class 1. By calculating its harmonic mean, the F measure, also known

as the f1-score, which is equal to 0.93 for class 0 and 0.80 for class 1, simultaneously measures recall and precision. An accurate prediction has an overall accuracy of 0.90. The validation data set is expected to cost 13580 in total.

	precision	recall	f1-score	support
0	1.00	0.96	0.98	15625
1	0.32	0.85	0.46	375
accuracy			0.95	16000
macro avg	0.66	0.90	0.72	16000
weighted avg	0.98	0.95	0.96	16000
Total cost is: 35860.0				

Figure 35. Recurrent Neural Network model for Test Data

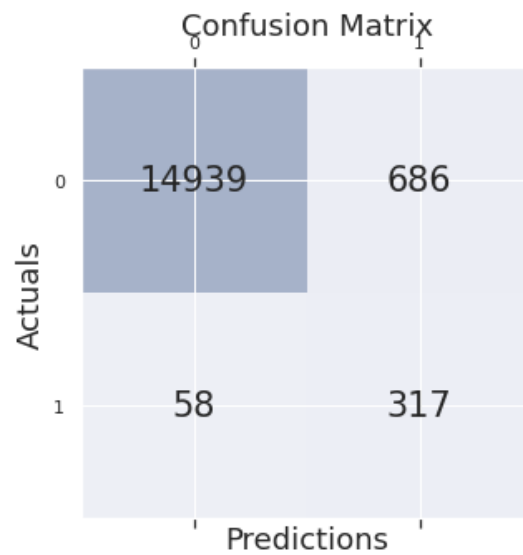


Figure 36. Recurrent Neural Network model for Test Data

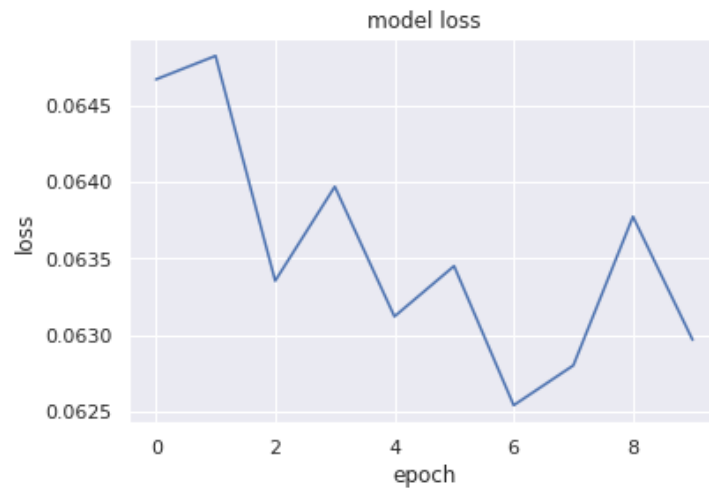
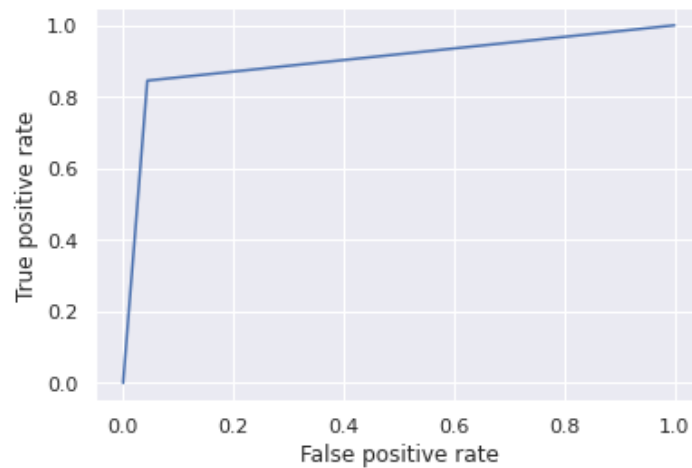


Figure 37. Loss Curve Diagram for Recurrent Neural Network model



AREA UNDER ROC: 0.9007146666666668

Figure 38. ROC curve for the Recurrent Neural Network model

The information above in Figure 35-38 shows the test data modeling results. But if the false positive is inexpensive, it might not be so bad. Recall, is the proportion of correctly identified positive predictions, which is 0.96 for class 0 and 0.85 for class 1, respectively. The F measure, or f1-score, which equals 0.98 for class 0 and 0.46 for class 1. With a 0.95 accuracy rate, a prediction is considered accurate overall. The amount of each class that exists is known as support. This model has a 35860 total maintenance cost. This area has a 0.047 misclassification rate.

6.3 Multi-Layer Perceptron

Multi-layer perception is abbreviated as MLP. It consists of dense, fully connected layers that can convert any input dimension into the desired dimension. Multi-layer perception refers to neural networks with multiple layers. Neural networks are built by combining neurons so that some of their outputs are also inputs. There is one input layer with one neuron (or node) for each input, one output layer with one node for each output, and any number of hidden layers with any number of nodes. A Multi-Layer Perceptron (MLP) is illustrated below. Backpropagation is a supervised learning technique used in MLP.

A multilayer perceptron has three inputs and three input nodes, which equals three hidden nodes. The output layer produces two outputs, so there are two output nodes. The input layer forwards its output to each of the three hidden layer nodes, and the hidden layer processes the data before sending it to the output layer. Nodes in the input layer receive input and forward it for further processing. As part of the multi-layer perception, each node uses the sigmoid activation function. Sigmoid activation functions translate real values into numbers between 0 and 1 using the sigmoid formula.

$$\sigma(x) = 1/(1+\exp(-x))$$

The process was done with the following parameters:

- Hidden_layer_sizes=(6,5)
- Random_state=5
- Verbose='True'
- Learning_rate_init=0.01

6.3.1 Without Removing Correlated Columns

	precision	recall	f1-score	support
0	0.95	0.98	0.96	253
1	0.93	0.88	0.90	97
accuracy			0.95	350
macro avg	0.94	0.93	0.93	350
weighted avg	0.95	0.95	0.95	350
Total cost is: 6060.0				

Figure 39. Multi-layer perceptron model for Validation Data

The information above in Figure 39 shows the validation data modeling results. Positive predictions have a higher accuracy when their precision is high, which is 0.95 for class 0 and 0.93 for class 1, respectively. The recall is 0.98 for class 0 and 0.88 for class 1. The F measure, also known as the f1-score, which is equal to 0.96 for class 0 and 0.80 for class 1, measures both precision and recall at the same time by determining its harmonic mean. With a 0.95 accuracy rate, a prediction is considered accurate overall. A 6060-cost estimate is made for the validation data set.

	precision	recall	f1-score	support
0	1.00	0.98	0.99	15625
1	0.52	0.90	0.66	375
accuracy			0.98	16000
macro avg	0.76	0.94	0.82	16000
weighted avg	0.99	0.98	0.98	16000
Total cost is: 22140.0				

Figure 40. Multi-layer perceptron model for Test Data

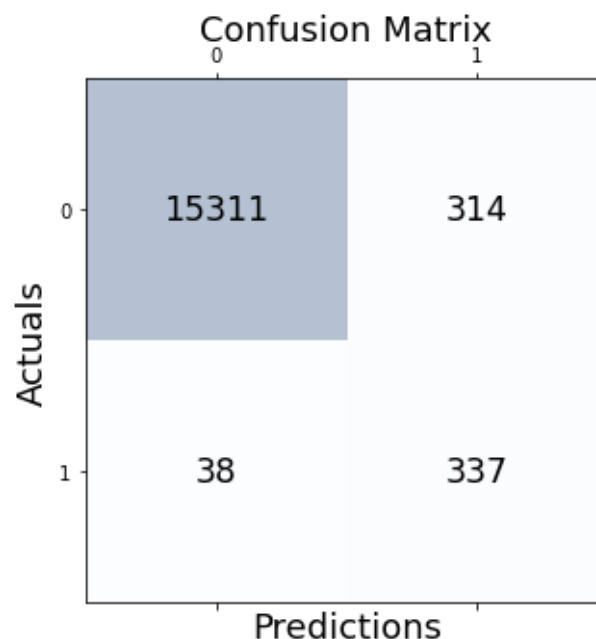
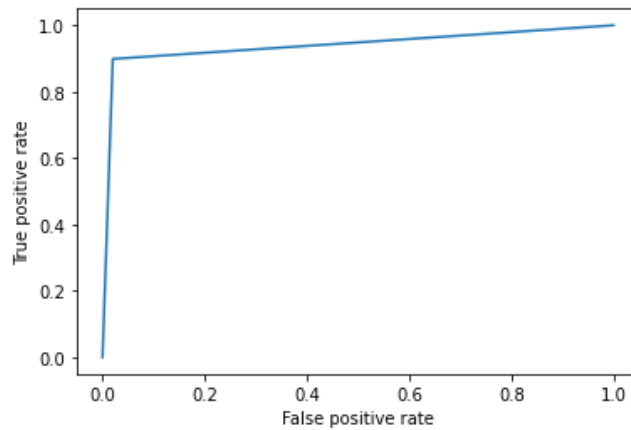


Figure 41. Confusion Matrix for Multi-layer perceptron model



AREA UNDER ROC: 0.9392853333333333

Figure 42. ROC curve for the multi-layer perceptron model

The modeling outcomes for the test data are shown in Figure 40-42, the aforementioned specifics. The recall is 0.98 for class 0 and 0.90 for class 1. The F measure is equal to 0.99 for class 0 and 0.66 for class 1, measures both precision and recall at the same time by determining its harmonic mean. An accurate prediction has an overall accuracy of 0.98. This model has a total maintenance cost of 22140. The rate of misclassification is 0.047 in this case. The likelihood that 0 classes will be predicted as 0, and 1 classes will be predicted as 1, increases with increasing AUC. Area under the curve is 0.9393.

6.3.2 Removing Correlated Columns

	precision	recall	f1-score	support
0	0.98	0.96	0.97	253
1	0.91	0.94	0.92	97
accuracy			0.96	350
macro avg	0.94	0.95	0.95	350
weighted avg	0.96	0.96	0.96	350
Total cost is: 3090.0				

Figure 43. Multi-layer perceptron model for Validation Data

The results of the modeling for the validation data are shown in Figure 43. Positive predictions are more accurate when their precision is high, which is 0.98 for class 0, 0.91 for class 1. The recall value is 0.94 for class 1 and 0.96 for class 0. The F1 score equals 0.97 for class 0 and 0.92 for class 1, simultaneously measures precision and recall. A prediction that is accurate across all predictions has an accuracy of 0.96. It is estimated that the validation data set will cost 3090.

	precision	recall	f1-score	support
0	1.00	0.97	0.99	15625
1	0.45	0.91	0.60	375
accuracy			0.97	16000
macro avg	0.72	0.94	0.79	16000
weighted avg	0.99	0.97	0.98	16000
Total cost is: 20230.0				

Figure 44. Multi-layer perceptron model for Test Data

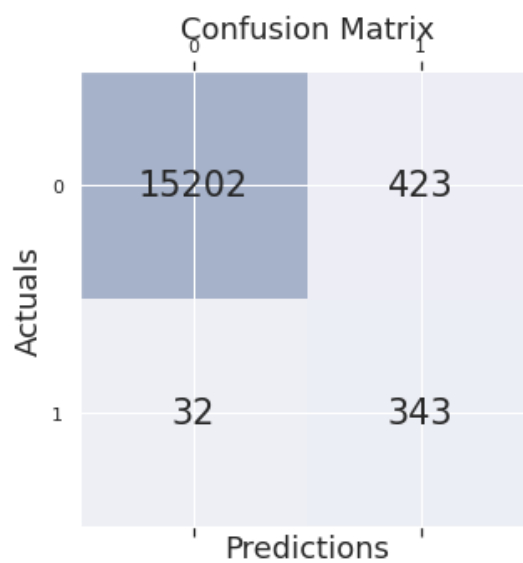
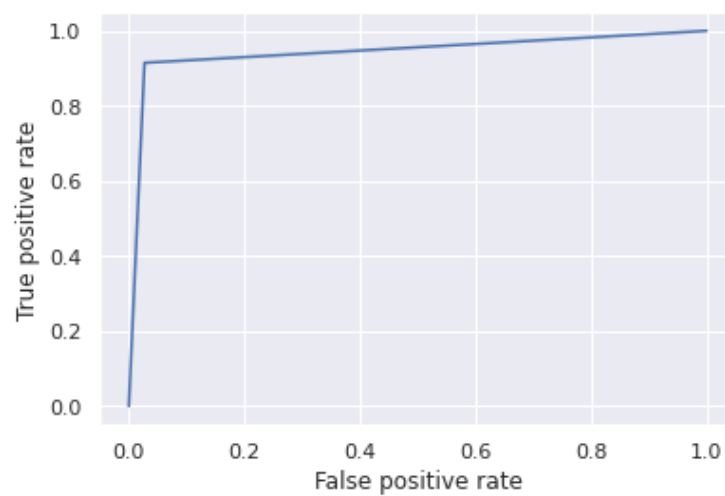


Figure 45. Confusion Matrix for Multi-layer perceptron model



AREA UNDER ROC: 0.9437973333333334

Figure 46. ROC curve for the Multi-layer perceptron model

These specifics show the modeling results for the test data Figure 44-46. As a result of the data imbalance, classes have less precision. However, if the false positive is inexpensive, it might not be such a bad thing. The proportion of positive predictions that are correctly identified with a recall of 0.97 for class 0 and 0.91 for class 1 is referred to as sensitivity, hit rate, or recall. The F measure, or f1-score, measures recall and precision simultaneously. It is equal to 0.99 for class 0 and 0.60 for class 1. There is an overall accuracy of 0.98 for an accurate prediction. Support refers to the number of instances of each class. The total maintenance costs for this model are 20230. This case has a misclassification rate of 0.028. An example of a training function is a loss function. The addition of more epochs resulted in a decrease in loss and an improvement in accuracy. Probability and separability are measured by the ROC and AUC, respectively. A model's class discrimination capability is measured by this measurement. As AUC increases, it is more likely that 0 classes will be predicted as 0, and 1 classes will be predicted as 1. The area under the curve is 0.9438.

6.4 Artificial Neural Network [Layers 60,30,15 ,1]

The purpose of artificial neural networks is to mimic the behavior of neural networks in humans and other animals' brains. Modeling the behavior of neurons, machine learning acquires the model architecture needed to handle increasingly complex data. Compared to contemporary methods, many early iterations of artificial neural networks appear straightforward. Artificial neural networks are used as the architecture for advanced deep learning models. Depending on the type of artificial neural network, each artificial neuron is connected to other nodes in a different number and density. There are typically layers of nodes between the input and output layers of the network. This multi-layered network architecture is also called a deep neural network due to the thickness of its layers. Data features can be picked up by these various layers in artificial neural network models. Complex concepts or patterns can be understood from processed data using hidden hierarchical layers.

The process was done with the following parameters:

- Layers=[60,30,15,1]
- Optimizer='adam'
- Loss='binary_crossentropy'

6.4.1 Without Removing Correlated Columns

	precision	recall	f1-score	support
0	0.98	0.97	0.97	253
1	0.92	0.94	0.93	97
accuracy			0.96	350
macro avg	0.95	0.95	0.95	350
weighted avg	0.96	0.96	0.96	350
Total cost is: 3080.0				

Figure 47. ANN model[layers 60,30,15,1] for Validation Data

The results of the modeling are shown above for the test data. When a positive prediction has a high precision, such as 0.98 for class 0, 0.92 for class 1, it is more accurate. The percentage of correctly identified positive predictions is known as a recall and is 0.97 for class 0 and 0.94 for class 1. It is also known as sensitivity or hit rate. By determining its harmonic mean, the F measure, or f1-score, which equals 0.97 for class 0 and 0.93 for class 1, simultaneously measures precision and recall. An accurate prediction has an average accuracy of 0.96 across all predictions. The validation data set is expected to cost 3080 in total.

	precision	recall	f1-score	support
0	1.00	0.99	0.99	15625
1	0.65	0.89	0.75	375
accuracy			0.99	16000
macro avg	0.82	0.94	0.87	16000
weighted avg	0.99	0.99	0.99	16000
Total cost is: 23280.0				

Figure 48. ANN model[layers 60,30,15,1] for Test Data

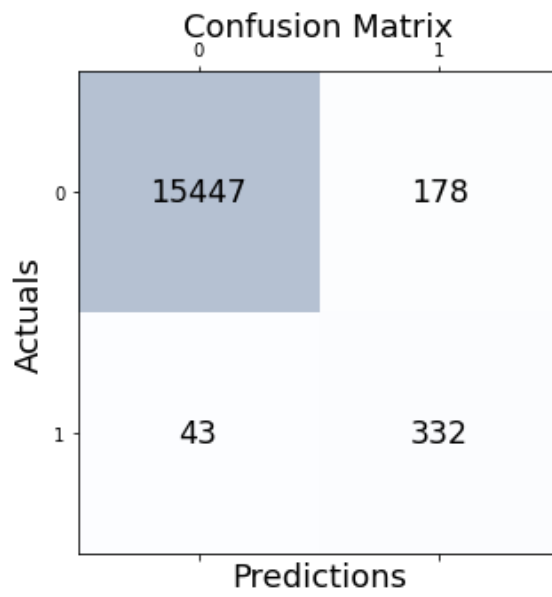


Figure 49. Confusion Matrix for ANN model[layers 60,30,15,1]

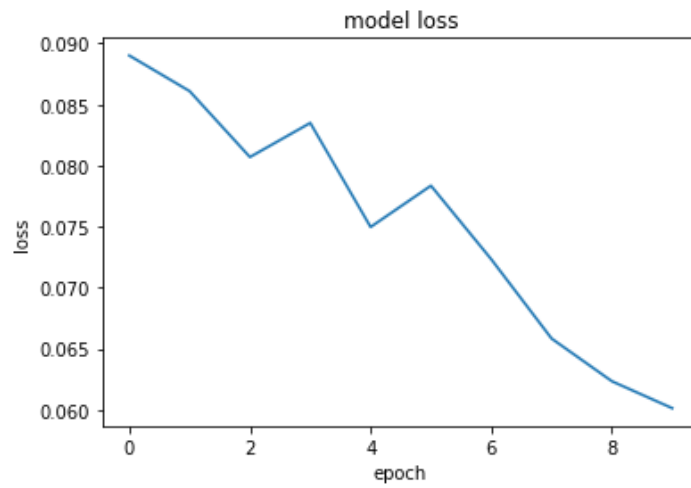
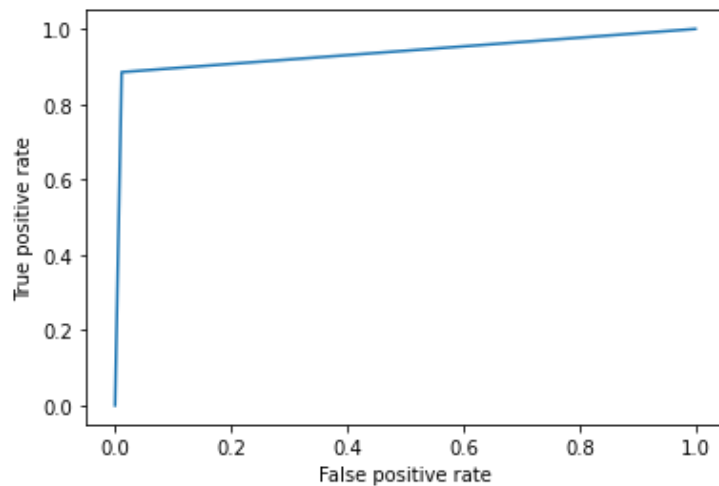


Figure 50. Loss Curve Diagram for ANN model[layers 60,30,15,1]



AREA UNDER ROC: 0.9369706666666667

Figure 51. ROC curve for the ANN model[layers 60,30,15,1]

The results of the modeling are shown above Figure 48-51 for the test data. The percentage of correctly identified positive predictions with a recall of 0.99 for class 0 and 0.89 for class 1 is known as hit rate, or recall. By determining its harmonic mean, the F measure, or f1-score, which is 0.99 for class 0 and 0.75 for class 1, simultaneously measures precision and recall. An accurate prediction has an average accuracy of 0.99 across all predictions. This model has a total maintenance cost of 23,280. The misclassification rate in this case is 0.014. It is more likely that 0 classes will be predicted as 0, and 1 classes will be predicted as 1, as AUC increases. AUC is 0.9370 in this instance.

6.4.2 Removing Correlated Columns

	precision	recall	f1-score	support
0	0.96	0.97	0.97	253
1	0.92	0.91	0.91	97
accuracy			0.95	350
macro avg	0.94	0.94	0.94	350
weighted avg	0.95	0.95	0.95	350
Total cost is: 4580.0				

Figure 52. ANN model[layers 60,30,15,1] for Validation Data

The results of the modeling are shown above Figure 52 for the validation data. When a positive prediction has a high precision, such as 0.96 for class 0 and 0.92 for class 1, it is more accurate. The recall, is 0.97 for class 0 and 0.91 for class 1. By determining its harmonic mean, the F measure, or f1-score, which equals 0.97 for class 0 and 0.91 for class 1, simultaneously

measures precision and recall. An accurate prediction has an average accuracy of 0.95 across all predictions. It is estimated that the validation data set will cost 4580.

	precision	recall	f1-score	support
0	1.00	0.98	0.99	15625
1	0.53	0.91	0.67	375
accuracy			0.98	16000
macro avg	0.76	0.94	0.83	16000
weighted avg	0.99	0.98	0.98	16000
Total cost is: 20050.0				

Figure 53. ANN model[layers 60,30,15,1] for Test Data

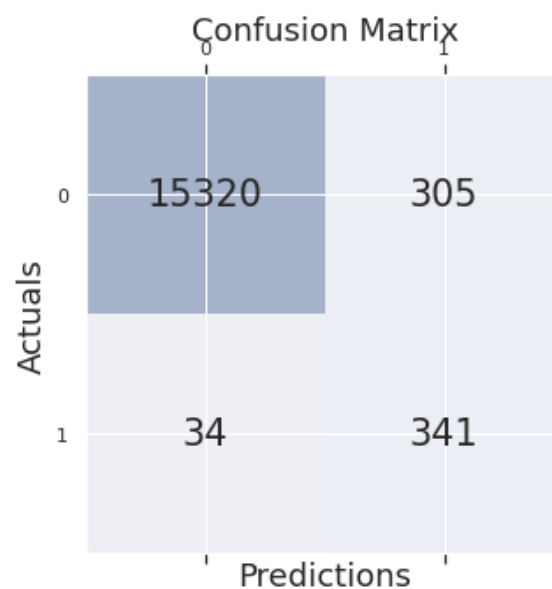


Figure 54. Confusion Matrix for ANN model[layers 60,30,15,1]

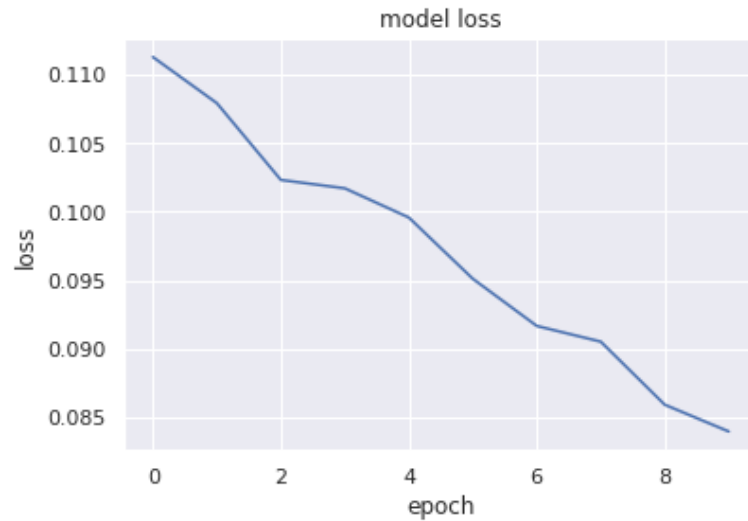
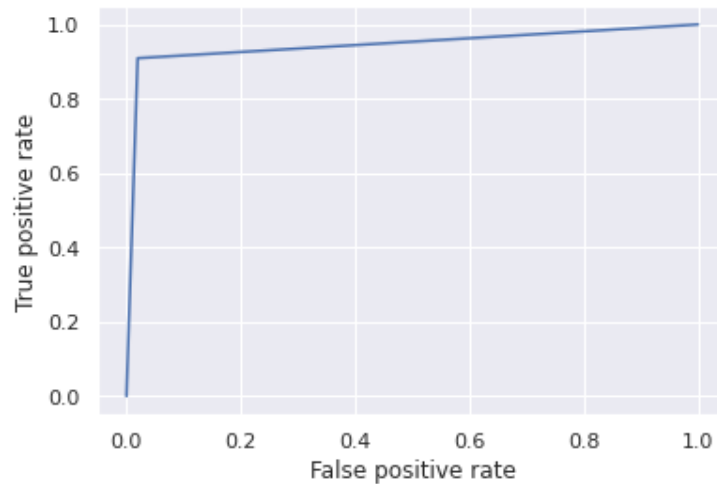


Figure 55. Loss Curve Diagram for ANN model[layers 60,30,15,1]



AREA UNDER ROC: 0.9449066666666667

Figure 56. ROC curve for the ANN model[layers 60,30,15,1]

The results of the modeling are shown above Figure 53-56 for the test data. The recall is 0.98 for class 0 and 0.91 for class 1. By calculating its harmonic mean, the F measure, or f1-score, which is equal to 0.99 for class 0 and 0.98 for class 1. An accurate prediction has an average accuracy of 0.98 across all predictions. This model has a total maintenance cost of 23,280. The misclassification rate in this case is 0.021. One training example is represented by a loss function. In general, loss decreased and accuracy improved as the number of epochs increased. The ROC calculates probability, while the AUC calculates separability. This measurement reflects the class discrimination performance of a model. As AUC rises, there is a greater chance that 0 classes will be predicted as 0, and 1 classes will be predicted as 1. The area under the curve is 0.9449.

6.5 Artificial Neural Network [Layers 64,32,16,1]

Typically, an artificial neural network is organized in layers. Many interconnected "nodes" that each have an "activation function" make up a layer. Hidden layers are responsible for neural networks ability to recognize extremely complex relationships and deliver thrilling results in a variety of tasks.

The process was done with the following parameters:

- Layers= [64,32,16,1]
- Optimizer='adam'
- Loss='binary_crossentropy'

7.5.1 Without Removing Correlated Columns

	precision	recall	f1-score	support
0	0.97	0.98	0.97	253
1	0.94	0.93	0.93	97
accuracy			0.96	350
macro avg	0.95	0.95	0.95	350
weighted avg	0.96	0.96	0.96	350
Total cost is: 3560.0				

Figure 57. ANN model[layers 64,32,16,1] for Validation Data

The results of the modeling are shown above Figure 57 for the validation data. Positive predictions with high precision—which is 0.97 for class 0 and 0.94 for class 1—are more accurate. The recall value is 0.98 for class 0 and 0.93 for class 1. The F measure, or f1-score, which equals 0.97 for class 0 and 0.93 for class 1. An accurate prediction has an average accuracy of 0.96 across all predictions. It is estimated that the validation data set will cost 3560.

	precision	recall	f1-score	support
0	1.00	0.97	0.99	15625
1	0.46	0.96	0.62	375
accuracy			0.97	16000
macro avg	0.73	0.97	0.80	16000
weighted avg	0.99	0.97	0.98	16000
Total cost is: 11720.0				

Figure 58. ANN model[layers 64,32,16,1] for Test Data

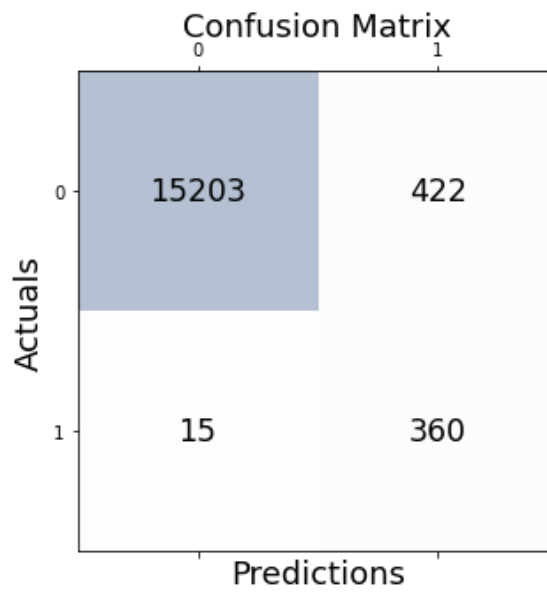


Figure 59. Confusion Matrix for ANN model[layers 64,32,16,1]

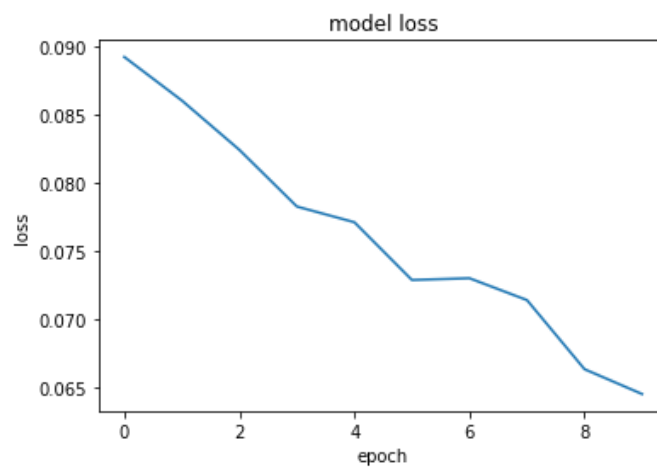
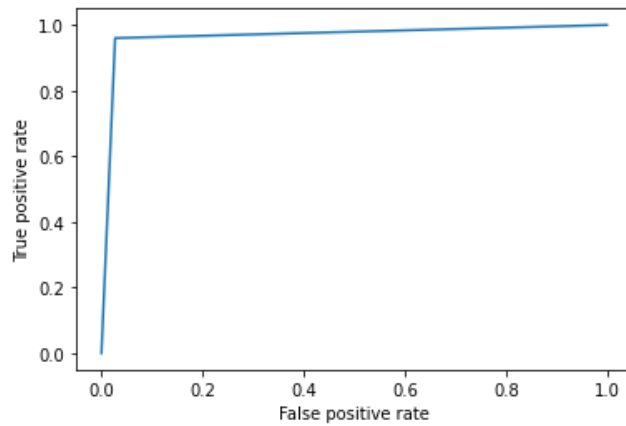


Figure 60. Loss Curve Diagram for ANN model[layers 64,32,16,1]



AREA UNDER ROC: 0.9664959999999999

Figure 61. ROC curve for the ANN model[layers 64,32,16,1]

The results of the modeling are shown above Figure 58-61 for the test data. Recall is the proportion of correctly identified positive predictions that are 0.97 for class 0 and 0.96 for class 1. By determining its harmonic mean, the F measure, or f1-score, which is equal to 0.99 for class 0 and 0.62 for class 1. An accurate prediction has an average accuracy of 0.98 across all predictions. The total cost of maintenance for this model is 11720. The misclassification rate in this case is 0.027. The ROC and AUC measure probability and separability, respectively. This measurement captures the class discrimination capability of a model. It is more likely that 0 classes will be predicted as 0, and 1 classes will be predicted as 1, as AUC increases. The area under the curve is 0.9665.

6.5.2 Removing Correlated Columns

	precision	recall	f1-score	support
0	0.96	0.97	0.96	253
1	0.92	0.89	0.91	97
accuracy			0.95	350
macro avg	0.94	0.93	0.93	350
weighted avg	0.95	0.95	0.95	350
Total cost is: 5570.0				

Figure 62. ANN model [layers 64,32,16,1] for Validation Data

The information above Figure 62 shows the test data modeling results. Positive predictions are more accurate when they have a high precision, which is 0.96 for class 0, 0.92 for class 1. The proportion of correctly identified positive predictions is known as recall, it is 0.97 for class 0 and 0.89 for class 1. The F measure, also known as the f1-score, which is equal

to 0.96 for class 0 and 0.91 for class 1, measures both precision and recall at the same time by determining its harmonic mean. With a 0.95 accuracy rate, a prediction is considered accurate overall. The validation data set's predicted cost is 5570.

	precision	recall	f1-score	support
0	1.00	0.94	0.97	15625
1	0.29	0.97	0.45	375
accuracy			0.94	16000
macro avg	0.65	0.96	0.71	16000
weighted avg	0.98	0.94	0.96	16000
Total cost is: 14680.0				

Figure 63. ANN model[layers 64,32,16,1] for Test Data

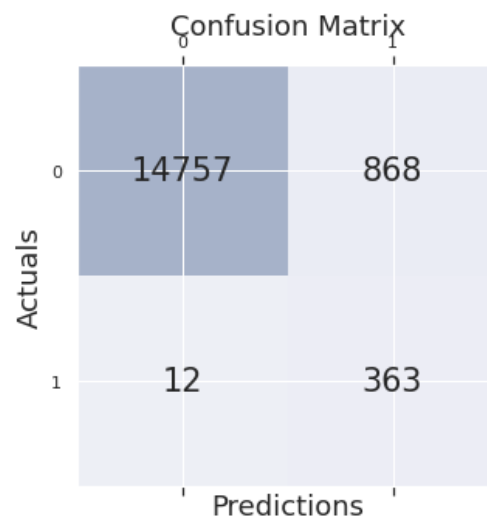


Figure 64. Confusion Matrix for ANN model[layers 64,32,16,1]

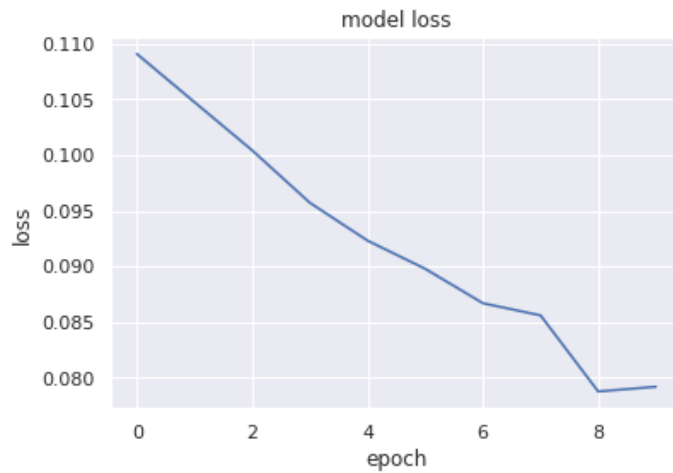
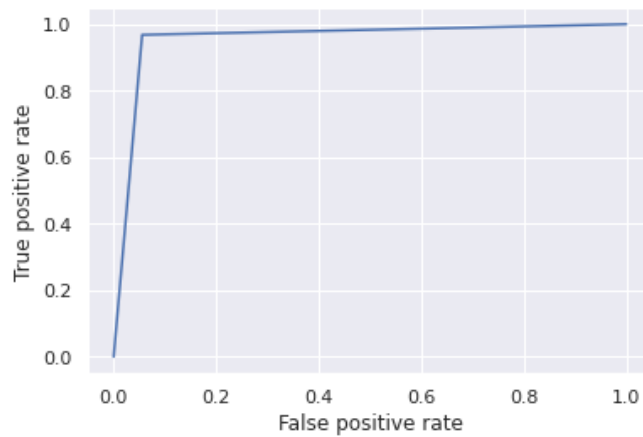


Figure 65. Loss Curve Diagram for ANN model[layers 64,32,16,1]



AREA UNDER ROC: 0.956224

Figure 66. ROC curve for the ANN model[layers 64,32,16,1]

The results of the modeling are shown above Figure 63-66 for the test data. The percentage of correctly identified positive predictions is known as a recall and is 0.94 for class 0 and 0.97 for class 1. By determining its harmonic mean, the F measure, or f1-score, which is equal to 0.97 for class 0 and 0.45 for class 1. An accurate prediction has an average accuracy of 0.94 across all predictions. The total cost of maintenance for this model is 11720. The misclassification rate in this case is 0.027. A general decrease in loss and an improvement in accuracy were the effects of adding more epochs. It is more likely that 0 classes will be predicted as 0, and 1 classes will be predicted as 1, as AUC increases. Area under the curve is 0.9562 in this case.

Chapter 7

Model Evaluation

Machine learning models are built on the principle of helpful feedback. Models are created, metrics are used to provide feedback, changes are made, and the process is repeated until the desired accuracy is achieved. Evaluation metrics explain the effectiveness of a model. Evaluation metrics must be able to distinguish between different model results. In machine learning model evaluation, various evaluation metrics are used to comprehend a model's performance, strengths, and weaknesses.

It is crucial to evaluate a model's effectiveness in the early stages of research. Monitoring a model is also aided by model evaluation. Assess a model's ability to accurately predict the target based on current and upcoming data. Holding up a sample of data from the training data source that has been labelled with the target (ground truth) allows to properly assess a model's accuracy metric on data for which the target answer is already known. Testing a machine learning model's predictive accuracy using the same training data is useless because it rewards models that remember the training data rather than extrapolate from it. ML models are trained by sending held-out observations with known target values to them after they have been trained. Based on the ML model's predictions, compare them with the predetermined target value. Using a summary metric, can determine how well the predicted and true values match.

A cost model for maintaining air pressure systems (APS) was developed based on the samples provided.

7.1 Without Removing Correlated Columns

Models	Cost	Accuracy	AUC
Basic Neural Network	14810	0.97	0.9577
Recurrent Neural Network	24150	0.93	0.9294
Multi-Layer Perceptron	22140	0.98	0.9393

ANN [layers 60,30,15 ,1]	23280	0.99	0.9370
ANN [layers 64,32,16 ,1]	11720	0.97	0.9665

Table 1. Model Selection (without removing correlated columns)

As a result of evaluating the various models, the ANN[layers 64,32,16,1] model is found to have the lowest maintenance cost of 11720, and the highest AUC value that is 0.9665. Nevertheless, the most accurate model is the ANN[60,30,15,1] model, but it was not able to provide good maintenance costs and AUC values. It can be concluded that the ANN[layers 64,32,16,1] model performs better than the other models in general.

7.2 Removing correlated columns

Models	Cost	Accuracy	AUC
Basic Neural Network	15950	0.96	0.9546
Recurrent Neural Network	35860	0.95	0.9007
Multi-Layer Perceptron	20230	0.97	0.9438
ANN [layers 60,30,15 ,1]	20050	0.98	0.9449
ANN [layers 64,32,16 ,1]	14680	0.94	0.9562

Table 2. Model Selection (Removing correlated columns)

Among the various models, ANN[layers 64,32,16,1] provides the lowest cost that is 14680, and highest AUC value of 0.9562 when compared with the other models. As far as accuracy is concerned, the ANN[60,30,15,1] model gives the highest accuracy for which the maintenance cost is a bit higher and the AUC is also less than the ANN[64,32,16,1]. A decent prediction can be made using the ANN[layers 64,32,16,1] model and the basic neural network model in this context.

Chapter 8

Conclusion

A truck company can save a lot of money if an air pressure system failure is discovered in advance. Overall, class imbalance and missing values in the dataset posed the biggest problems for this problem. Additionally, the dataset's features have been made anonymous for confidential reasons. After overcoming the difficulties, a well-performing model can be obtained. To solve the missing value problem, initially should eliminate the features with a high number of missing values before using median imputation to impute the missing values for the remaining features.

During the modeling phase, every model was biased toward the negative class. However, balancing the data set significantly increased the model's performance. New features are added to the median imputed features to indicate missing values of the train and test data.

The various models in neural networks like basic neural network, MLP, ANN, RNN are performed. Though few other models were good in accuracy, ANN with [layers 64,32,16 ,1] gave lower cost for maintenance, highest AUC value and a decent accuracy for both data set that is without removing correlated columns and removing correlated columns respectively. The results were reasonably good, and they have been used.

This specific issue may be resolved using Deep Learning techniques, and its performance on Neural Networks may be assessed using the performance metric. Other imputation techniques, like the Soft-Impute Algorithm, can be applied. A variety of other feature selection and imputation techniques can be used in this situation.

References

Kaggle : <https://www.kaggle.com/uciml/aps-failure-at-scania-trucks-data-set>

- [1] T. Vijayakumar, "Comparative study of capsule neural network in various applications," *Journal of Artificial Intelligence*, vol. 1, pp. 19-27, 2019.
- [2] Mohamed Bekkar, Dr.Hassiba Kheliouane Djemaa, and Dr.Taklit Akrouf Ali- touche. "Evaluation Measures for Models Assessment over Imbalanced Data Sets". In: (2013).
- [3] Gondek, Christopher & Hafner, Daniel & Sampson, Oliver. (2016). Prediction of Failures in the Air Pressure System of Scania Trucks Using a Random Forest and Feature Engineering. 10.1007/978-3-319-46349-0_36.
- [4] Cerqueira V., Pinto F., S C., Soares C. (2016) Combining Boosted Trees with Metafeature Engineering for Predictive Maintenance. In: Boström H., Knobbe A., Soares C., Papapetrou P. (eds) *Advances in Intelligent Data Analysis XV. IDA 2016. Lecture Notes in Computer Science*, vol 9897. Springer, Cham
- [5] G. Weiss. Mining with rarity: A unifying framework. *SIGKDD Explorations*, 6(1):7-19, 2004.
- [6] Aman Savaria "Predicting a Failure in the APS of a Scania Truck using Machine Learning"
- [7] Mrugank M Akarte "Predictive Maintenance of Air Pressure System using Boosting Trees: A Machine Learning Approach"
- [8] H. Nguyen and X.-N. Bui, "Predicting blast- induced air overpressure: a robust artificial intelligence system based on artificial neural networks and random forest," *Natural Resources Research*, vol. 28, pp. 893-907, 2019.
- [9] A. Bashar, "Survey on evolving deep learning neural network architectures," *Journal of Artificial Intelligence*, vol. 1, pp. 73-82, 2019.
- [10] R. A. Hamad, M. Kimura, and J. Lundström, "Efficacy of imbalanced data handling methods on deep learning for smart homes environments," *SN Computer Science*, vol. 1, pp. 1-10, 2020.
- [11] S. Fotouhi, S. Asadi, and M. W. Kattan, "A comprehensive data level analysis for cancer diagnosis on imbalanced data," *Journal of biomedical informatics*, vol. 90, p. 103089, 2019.
- [12] V. Babar and R. Ade, "A novel approach for handling imbalanced data in medical

- diagnosis using undersampling technique," *Communications on Applied Electronics*, vol. 5, pp. 36-42, 2016.
- [13] R. Dhall and V. Solanki, "An IoT Based Predictive Connected Car Maintenance," *International journal of interactive multimedia & artificial intelligence*, vol. 4, 2017.
 - [14] H. A. Nabwey, "A Method for Fault Prediction of Air Brake System in Vehicles," *International Journal of Engineering Research and Technology*, vol. 13, pp. 1002-1008, 2020.
 - [15] K. T. Selvi, R. Thamilselvan, and S. M. Saranya, "Diffusion convolution recurrent neural network—a comprehensive survey," in *IOP Conference Series: Materials Science and Engineering* , 2021, p. 012119.
 - [16] R. Jegadeeshwaran and V. Sugumaran, "Vibration based condition monitoring of a brake system using statistical features with logit boost and simple logistic algorithm," *International Journal of Performability Engineering*, vol. 14, p. 1, 2018.
 - [17] J. S. Raj and J. V. Ananthi, "Recurrent neural networks and nonlinear prediction in support vector machines," *Journal of Soft Computing Paradigm (JSCP)*, vol. 1, pp. 33-40, 2019.
 - [18] R. Raveendran, K. Devika, and S. C. Subramanian, "Intelligent Fault Diagnosis of Air Brake System in Heavy Commercial Road Vehicles," in *2020 International Conference on COMmunication Systems & NETwork S (COMSNETS)* , 2020, pp. 93- 98.
 - [19] W. YU, Y. CHEN, X. SHAO, and L. FU, "Failure Diagnosis for Air Brake System of Some Truck Based on Fault Tree Analysis," *Machine Tool & Hydraulics*, p. 02, 2017.
 - [20] R. Raveendran, A. Suresh, V. Rajaram, and S. C. Subramanian, "Artificial neural network approach for air brake pushrod stroke prediction in heavy commercial road vehicles," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 233, pp. 2467-2478, 2019.
 - [21] S. Zheng, A. Farahat, and C. Gupta, "Generative adversarial networks for failure prediction," *arXiv preprint arXiv:1910.02034*, 2019.
 - [22] M. M. Akarte and N. Hemachandra, "Predictive maintenance of air pressure system using boosting trees: A machine learning approach," in *ORSI*, 2018.

- [23] UCI Machine Learning Repository. 2017. Available online: <https://archive.ics.uci.edu/ml/datasets/APS+Failure+at+Scania+Trucks> (accessed on 8 December 2017)
- [20] Paul, D.A. Missing Data; Sage Publications Inc.: Thousand Oaks, CA, USA, 2002; pp. 27–74.
- [21] Dempster, P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm.