EE559-Mathematical Pattern Recognition

Final Course Project

APS FAILURE AT SCANIA TRUCKS DATA SET

-submitted by

RAKSHITHA PANDURANGA

rpandura@usc.edu

USC ID - 7890-1614-34

-guided by

PROF. KEITH JENKINS

Contents

1. ABSTRACT -

The Air Pressure System Failure at Scania Trucks deals with a major drawback in expenditure due to the failures in the Trucks that are caused.

This drawback can be dealt by designing a predictor which can predict the failures and consequently, the expenditure that the company will endure and thus, helping the company to take precautions in mitigating the loss.

In this project, APS data set is used and the main aim is to develop such an efficient Pattern Recognition System which can consider all the previous patterns and base them with sound statistical reasoning as to why and how much the cost will be endured by the company.

Naïve Baye's classifier, Support Vector Machine, Multi-Layer Perceptron, Random Forest Classifier, **KNN** Classifier are used for the classification. The code is written in Python with various libraries like Scikit Learn, Pandas with various metrics like F1 scores, data set accuracies are considered to give a much more detailed perspective into the problem at hand, as well as the solution. The results led to the conclusion that KNN classifier is the best classification model and gave a F1 score of **0.9930** and the Training Data Set Cost of **1020.0**$ and Testing Data Set Cost of **2060.0$.**

2. INTRODUCTION –

2.1 Problem Statement and Goals –

Problem Statement –

- Scania Trucks deals with a major drawback in expenditure due to the failures in the Trucks that are caused due to APS systems of the Scania trucks. The issue here is to find whether the components belonged to APS or from an external source.

Goals –

- To develop a classifier that can classify the failure of the APS data set based on whether the components were manufactured by APS or another external source to mitigate the loss incurred by SCANIA.
- To develop a best model for classification with least error rate so that the cost endured by the company is lesser.

2.2 Literature Review –

- I have searched a few online resources and found the best classifiers that can be used. I have used those in my project.
- I have read a good amount on the various metrics that need to be considered for classification and used those for getting the best model

3. APPROACH AND PROCEDURE –

3.1. Data Analysing and Feature Importance -

- The given data set was analysed using various techniques like –
  1. Estimated PDF method
  2. Heat maps
  3. Box plots
  4. Variance Thresholding
  5. Missing value percentages

3.2. Data pre-processing -

- The features with less importance were dropped.
- The missing values were imputed using two methods-
  1. Mean of the particular feature
  2. KNN

3.3. Data transformation/Normalisation -

- The training and the testing data sets were normalised with the feature means of the training data set.

3.4. Dimensionality Reduction -

- The normalised data was then sent to the PCA for the reducing the dimensions and keeping the features with utmost importance.

3.5. Classification -

- The data was classified using 5 classifiers without PCA, with PCA and with Cross-Validation.

Explaining in detail,

3.1. Data Analysing and Feature Importance -

I ran a few codes online and these were the results I obtained.

Data analysis was done by using five methods –

1. Histograms and Estimated PDF –

- Plotting the univariate distribution of a few important features of the dataset and their estimated PDF is plotted as shown below.
- The more the variance of the PDF, the better it is for the training of the model and it shows that it has more information.

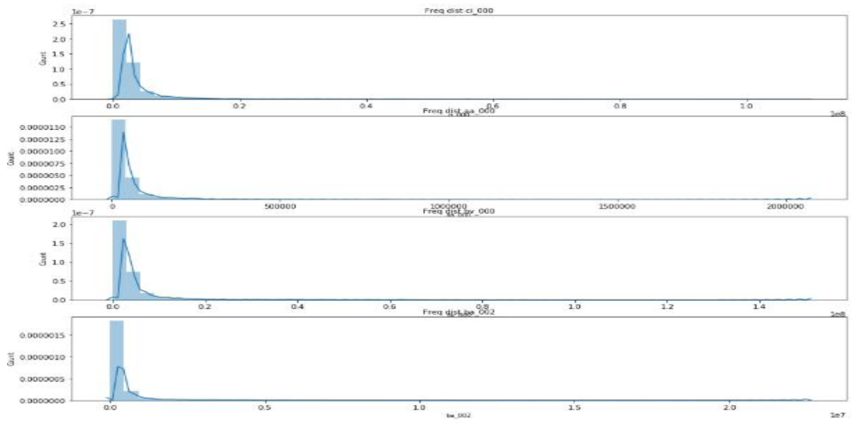
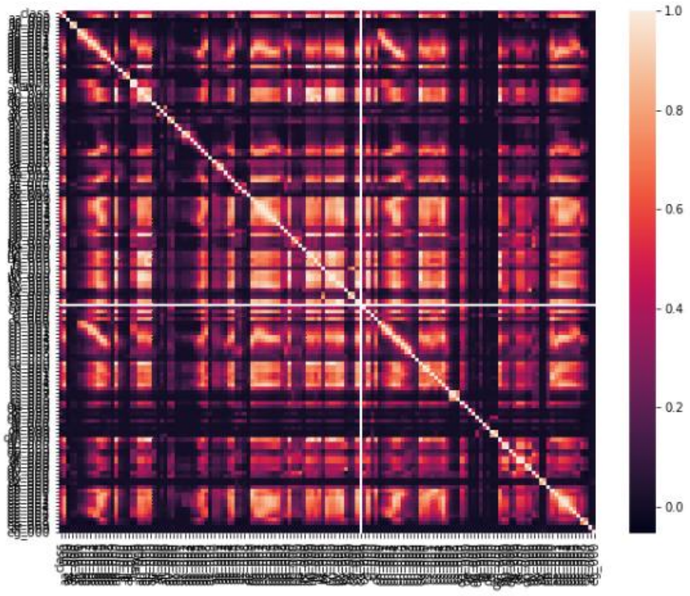


Figure: Estimated PDFs of the most important features

2. Taking the correlation matrix into account –

- The Correlation matrix shows each of the correlation coefficient of the features with each other.
- From these, we can analyse the best set of data that can be used for the best and most efficient way of training the model.
- These can be further utilised to understand which of the attributes contributed the best for the training of the model.



3. Plotting the Box plot for each of the features -

- Taking the box plot into consideration for getting the importance of each feature and the amount of information they contain.

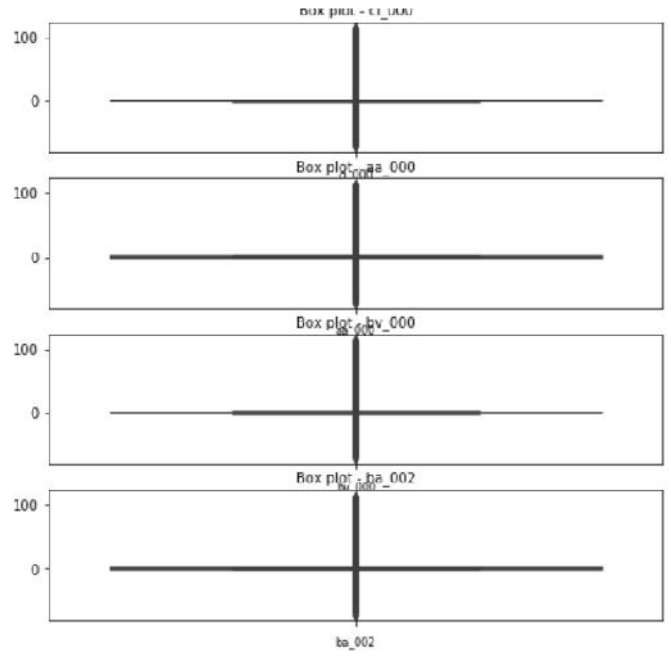
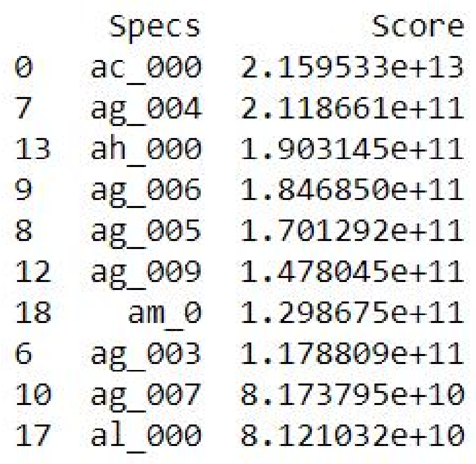


Figure: The box plot indicating the amount of information from each feature

<u>4. Variance Threshold</u>

- In this method, the different features and their variances are taken into consideration.
- Higher the variance, the more irrelevant the feature is.

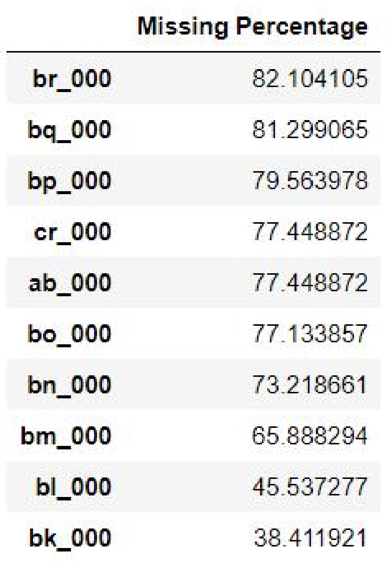


```
       Specs         Score
0     ac_000  2.159533e+13
7     ag_004  2.118661e+11
13    ah_000  1.903145e+11
9     ag_006  1.846850e+11
8     ag_005  1.701292e+11
12    ag_009  1.478045e+11
18      am_0  1.298675e+11
6     ag_003  1.178809e+11
10    ag_007  8.173795e+10
17    al_000  8.121032e+10
```

Figure: Shows the top ten features and their variance scores

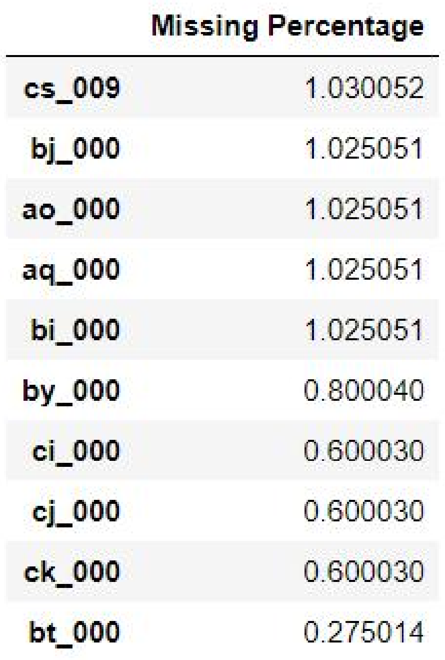
<u>5. Missing percentage method –</u>

- This method helps to remove the features with a particular threshold of missing percentage that can be computed as follow.



	Missing Percentage
br_000	82.104105
bq_000	81.299065
bp_000	79.563978
cr_000	77.448872
ab_000	77.448872
bo_000	77.133857
bn_000	73.218661
bm_000	65.888294
bl_000	45.537277
bk_000	38.411921

Figure: Shows the features with the maximum number of Missing values



	Missing Percentage
cs_009	1.030052
bj_000	1.025051
ao_000	1.025051
aq_000	1.025051
bi_000	1.025051
by_000	0.800040
ci_000	0.600030
cj_000	0.600030
ck_000	0.600030
bt_000	0.275014

Figure: Shows the features with the least Missing values and of more importance

3.1. Feature Importance –

- Feature Importance is very useful in selecting the right features for training as the most important features have the maximum amount of information that is present which will help in building a much more effective model.
- Feature importance was done using two methods –

1. First method was done using model.feature_importances and plotting it against the features that were considered as shown below.

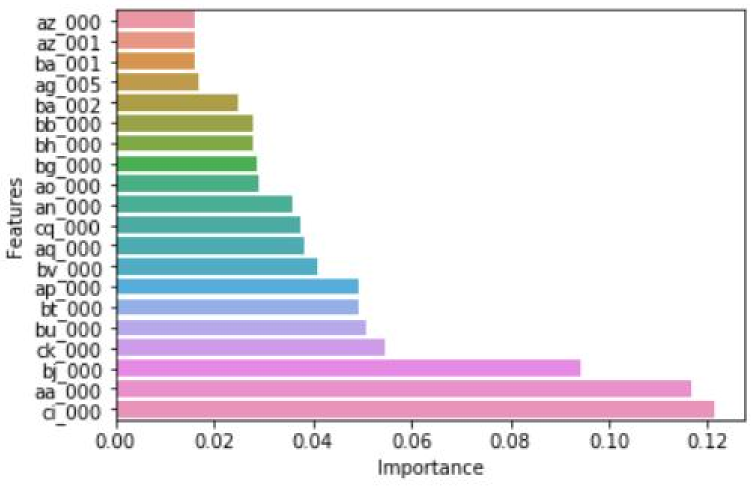


Figure: Feature Importance of a few features

2. The second method was done using the Recursive Feature Elimination Method and this was done later concatenating the feature names and their top twenty ranks.

Second method –

['bi_000', '1'],
['al_000', '2'],
['ag_001', '3'],
['ee_004', '4'],
['class', '5'],
['ak_000', '6'],
['ee_001', '7'],
['ag_002', '8'],
['ay_004', '9'],
['ao_000', '10'],
['ag_003', '11'],
['ag_000', '12'],
['ee_006', '13'],
['ce_000', '14'],
['cs_003', '15'],
['ay_006', '16'],
['cj_000', '17'],

```
['ee_007', '18'],
['cs_001', '19'],
['ba_004', '20'],
```

3.2. Data pre-processing -

One of the major Data Mining techniques involves Data Pre-processing. It is the process of transformation of the input raw and unorganised data to a much better, interpretable and understandable format. Data, usually has a lot of incompleteness to it – missing values of Attributes, including Outliers and Errors, Inconsistent with a lot of Discrepancies.

The different processes involved in pre-processing are-

1. Data Cleaning –

Data Cleaning is the method of identifying and removal of the data that is irrelevant or inaccurate and contributes very little towards training of a very efficient model.

Data Cleaning is the most time – consuming and the most important part of an data modelling process since the data that is fed in for training determines how efficient the training model and helps in a good model with a very good classification rate and less time consumption.

1. In our project, for the first step I made use of the following algorithm where I used a particular threshold for the data points for each feature.

The algorithm is as follows-

1. Iterate through each feature

2. If any feature, has less than Threshold = 2500 data points, remove that particular feature.

Advantage –

1. Reduction in the number of features– Training dataset /Testing Datset– Reduced from 171 to 143.

2. Unwanted data points with less information are removed

2. Data Integration -

Imputation-

The missing values in the data set – 'na'. Since, a lot of the features had this as a data point, it is impossible to pass the data for training this way. Therefore, I tried two imputation techniques in this –

1. Replacing the missing values with the means of the features –

In this method I did the following steps –

1. Calculated the mean of each feature of non 'na' values in the dataset

2. Replacing the ‘na’ or missing values with this mean that is computed.

2. Replacing the missing values with the KNNs of the features.

In this method, I made use of knn.impute

1. Takes the k = 5 nearest neighbours and generalises the data of these neighbour points

2. This value is then imputed to the ‘na’ values.

3.3 Data Transformation/Data Normalisation -

Data Transformation is the method of converting the data from one format of data to another.

Label Encoding for the Labels –

- The training and the test labels were represented by “pos” and “neg”
- I reduced this to 1 and 0 respectively to represent pos and neg.
- This was done using Label Encoding
- The advantage of this is it is very useful for using the label data for the training of the model.

Data Normalisation for the Training and Testing dataset –

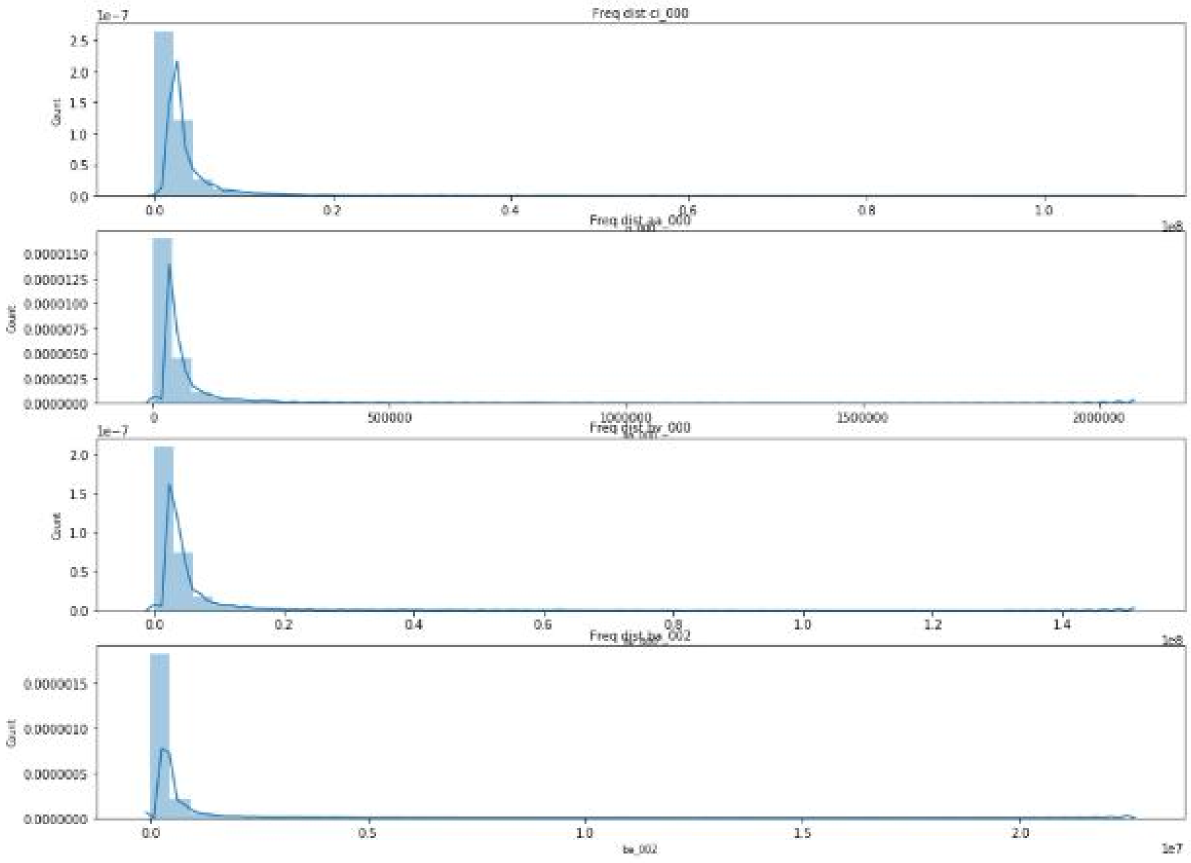
- The training data and the testing data were feature normalised using the feature means of the training data
- I made use of StandardScaler to make compute the normalised features of both the training and the test data.

1. Compensation for unbalanced data –

- There was UNBALANCED data present this particular data set.
- It was gotten rid by taking the SMOTE function on the training data.
- That is, the training data was generated synthetically to compensate for the unbalanced dataset.

3.4. Dimensionality Reduction –

- After Balancing the data, it was given to PCA for the dimensionality reduction
- The Principal Components taken for analysis at this step were 10.
- The best advantage of doing this, is the reduced time in computation.
- PCA doesn’t necessarily have any effect on increasing the accuracies of the training and the testing data.



Data Usage –

Initial Dataset –

Training Dataset = (19999,171)

Test Dataset = (16000,171)

Training Labels = (19999,1)

Test Labels = (16000,1)

After Thresholding –

Training Dataset = (19999,142)

Test Dataset = (16000,142)

Training Labels =(19999,1)

Test Labels =(16000,1)

After SMOTE –

Training Dataset = (39332,142)

Test Dataset = (16000,142)

Training Labels = (39332,1)

Test Labels = (16000,1)

After PCA–

Training Dataset = (39332,10)

Test Dataset = (16000,10)

Training Labels = (39332,1)

Test Labels = (16000,1)

Libraries and Functions Involved –

Libraries -

```
import numpy as np

import pickle

import pandas as pd

import matplotlib.pyplot as plt

import math as mt

import seaborn as sns

from sklearn import preprocessing

from sklearn.preprocessing import StandardScaler

from imblearn.over_sampling import SMOTE

from sklearn.decomposition import PCA

from sklearn.metrics import confusion_matrix

from sklearn.metrics import roc_curve,auc,roc_auc_score

from sklearn.metrics import classification_report

from sklearn.metrics import precision_recall_curve

from sklearn.feature_selection import RFE
from sklearn.linear_model import Lasso

import time

from sklearn.feature_selection import SelectKBest

from sklearn.feature_selection import chi2

from sklearn.naive_bayes import GaussianNB

from sklearn import svm

from sklearn.linear_model import Perceptron

from sklearn.neural_network import MLPClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier
```

Functions –

Functions	Functionality
data_cleaning	Used for removing the unwanted features
Missing_imputation1	Replaces the missing values with the class means
Missing_imputation1	Replaces the missing values with the KNN values
Label_encoding	Encodes the labels to 0 and 1
data_std	Standardises the data
data_SMOTE	Synthetically generates data to make it balanced
PCA_dr	Reduced the Dimensions by using PCA
data_acc	Finds the accuracy of the classifier
data_conf_matrix	Finds the confusion matrix of the data given
data_rocauc	Finds the ROC and AUC scores and plots the ROC graph
data_report	Gives the classification report
data_prcurve	Gives the Precision-Recall Curve
cost_function	Finds the cost of loss by the misclassification
NB_classifier	Naïve Baye's classification

SVM_classifier	SVM classification
MLP_classifier	MLP classification
RF_classifier	Random forest classification
KNN_classifier	KNN classification

Comparison of Learning Variables and Constraints –

1. Number of Learning Variables = 171

2. Number of Constraints = 40,000

Number of Constraints > (3-10) * Number of Learning Variables.

3.5. Training and Classification –

The below table shows the various classifiers that are used in this –

Classifiers	With PCA	Without PCA	With Cross-validation	Used Parameter Tuning
Naïve Baye's classifier	Yes	Yes	Yes	No
Support Vector machine	Yes	Yes	Yes	Yes
Multi-layer Perceptron	Yes	Yes	Yes	Yes
Random Forest	Yes	Yes	Yes	Yes
KNN classifier	Yes	Yes	Yes	Yes

Table: Different classifiers that are used in this project

3.5.1. Naïve Baye's Classifier –

3.5.1.1. Model Description

- It is one of the probabilistic classifiers. It is derived from the Baye's theorem. In this classifier, the features are supposedly strongly independent of each other.
- The main feature of a Naïve Baye's classifier is that it is scalable.
- They are widely used for text classification for spam detection.
- Since, it follows the Baye's theorem, the model can be formulated as –

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \qquad P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Training of a Naïve Baye's classifier is somewhat different compared to the other classifiers since, the only task that needs to be done is the finding of the parameters which have good individuality to them.

3.5.1.2. Parameter Selection – Baseline system

Parameters selected –

- No special parameters were given
- GNB from sklearn was used for the classification purpose

3.5.2. Support Vector Machine –

3.5.2.1. Model Description –

- Support Vector Machine is one of the discriminative types of classifiers. This is usually defined by the hyperplane which divides it. It is a type of supervised learning.
- The output of an SVM is a hyperplane which categorizes the data according to the respective classes.
- The parameters are very usual and require good tuning to the best model possible.
- It is used for Classification, linear regression models.
- It has two metrics –

1. Margin – Should be large
2. Classification Error – Very less

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\vec{w}\cdot\vec{x}_i - b)\right)\right] + \lambda\|\vec{w}\|^2,$$

-

3.5.2.2. Parameter Selection – Baseline system –

Parameters Selected	Value/metrics
gamma	'scale'(default)
kernel	rbf
C	1

Table: Different parameters used in base line model of Naïve Baye's classifier

3.5.3. Multi-Layer Perceptron

3.5.3.1. Model Description -

- An artificial Neural Network is nothing but a Multi-layer perceptron. Many number of perceptrons are combined to give a Multi-layer perceptron.
- MLPs have the capability to approximate a continuous function which is used to model the correlation between the outputs and their relations to the inputs.
- So, the back propagation takes the responsibility of making the bias and the weight adjustments which are used for the minimizing of the error.
- The error weights are given by –

$$\mathcal{E}(n) = \frac{1}{2}\sum_{j} e_j^2(n).$$

- Which are further used to find the derivative of any of the activation functions.

3.5.3.2. Parameter Selection – Baseline system-

Parameters Selected	Value/metrics
Solver	'Adam'
Activation	'relu'
alpha	1e-5
Hidden layer sizes	(100,50)
Learning rate	'constant'

Table: Different parameters used in base line model of SVM's classifier

3.4.5. Random Forest Classifier-

3.5.4.1. Model Description

- Random forest classifier is usually a combination of one or more types of classifications. It usually creates a set of decision trees that are randomly picked by the training set which has a subset.
- They are the ensemble type of classifiers that are used for the classification of the models.
- It is a very flexible that is used for the training of a MPR model.
- Random forests usually takes the columns, creates a forest and makes it random so that the model doesn't get too used to similar kind of data.
- Used in most of the machine learning these days, which is classification and all the regression problems.

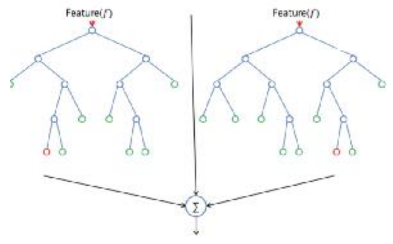


Figure: Random Forest Classification

<u>3.5.4.2. Parameter Selection – Baseline model-</u>

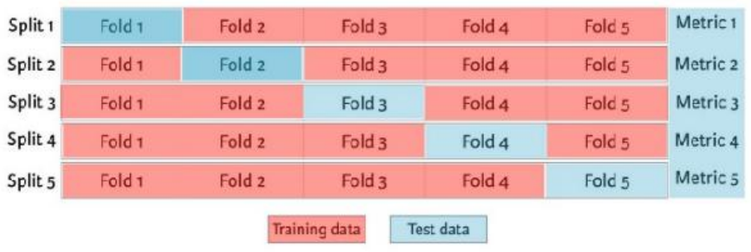
Parameters Selected	Value/Metrics
Number of estimators	100
Max_depth	2
Random_state	0

Table: Different parameters used in base line model of Random Forest's classifier

<u>3.5.5. KNN Classifier -</u>

<u>3.5.5.1. Model Description</u>

- KNN is one of the most simplest algorithms that are used for the storage of all the previously available cases that are present and their statistical estimations.
- KNN is the non-parametric learning algorithm of the others.
- KNN is basically dependent on how the model is trained on the data to learn and how it maps the unknown data to it
- KNN is also used for classification and regression problems.



5-fold cross validation (image credit)

Table : KNN classification

<u>3.5.5.2. Parameter Selection – Baseline system</u>

Parameters selected	Value/Metric chosen
N Neighbours	5

Table: Different parameters used in base line model of KNN's classifier

4. COMPARISON, RESULTS, INTERPRETATION

4.1 Metrics/ Results of classifiers without using Dimensionality Reduction (PCA) -

PARAMETERS	NAÏVE BAYE'S CLASSIFIER	SUPPORT VECTOR MACHINE	MULTI LAYER PERCEPTRON	RANDOM FOREST CLASSIFIER	KNN CLASSIFIER
Training Accuracy	93.60	97.43	99.9059	94.2616	99.1991
Testing Accuracy	96.56	98.1	98.7875	93.5687	97.775
Precision (for minority class)	0.40	0.57	0.78	0.26	0.51
Recall (for minority class)	0.92	0.80	0.67	0.97	0.90
F1 – score (for minority class)	0.56	0.66	0.72	0.42	0.66
AUC score	0.97	0.98	0.8738	0.9846	0.9584
F1 Score	0.5557	0.62	0.7212	0.4150	0.4145
Sensitivity	0.39	0.5668	0.7819	0.2637	0.2624
Specificity	0.9979	0.9952	0.9920	0.9993	0.9996
FPR (False Positive Ratio)	0.0332	0.01472	0.00448	0.06521	0.06656
TPR (True Positive Ratio)	0.9173	0.8026	0.66933	0.97333	0.98666

Table: Comparison of different metrics of all the classifiers without PCA

4.1 Cost Metrics/ Results of classifiers without using Dimensionality Reduction (PCA) -

TOTAL COST INCURRED	NAÏVE BAYE'S CLASSIFIER	SUPPORT VECTOR MACHINE	MULTI LAYER PERCEPTRON	RANDOM FOREST CLASSIFIER	KNN CLASSIFIER
Train Dataset	20090	7800	1020	20410	3580
Test Dataset	20690	39300	62700	15190	12900

Table: Comparison of the cost incurred by SCANIA with the usage of different data sets and classifiers

<u>4.1 Confusion Matrix, ROC Curve, Precision- Recall Curve Results of classifiers without using Dimensionality Reduction (PCA) -</u>

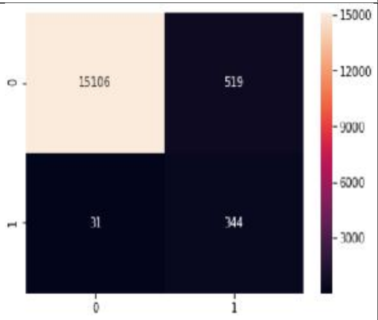
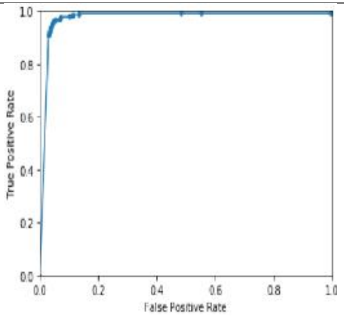
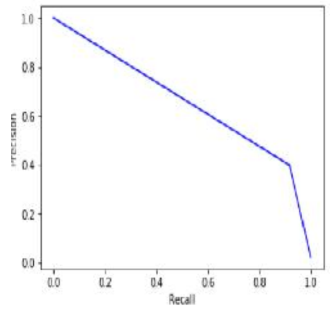
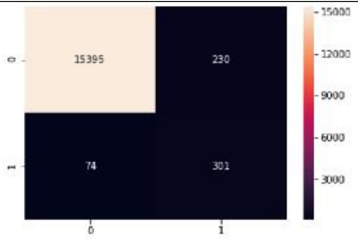
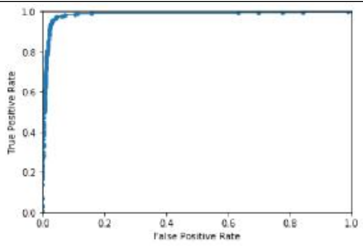
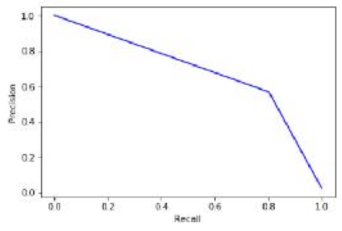
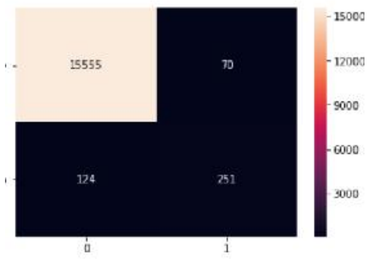
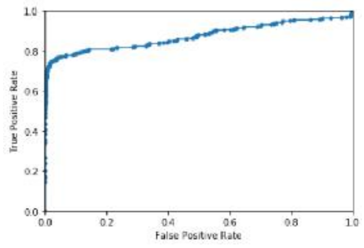
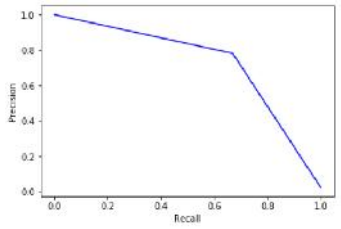
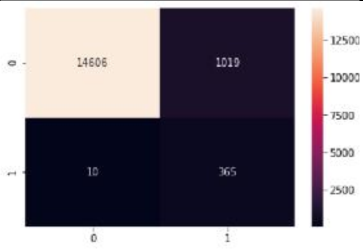
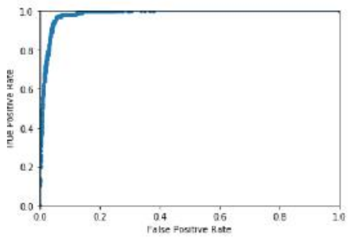
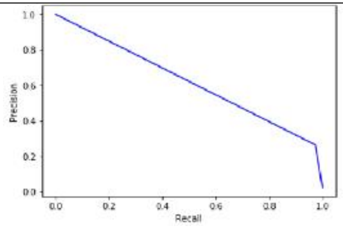
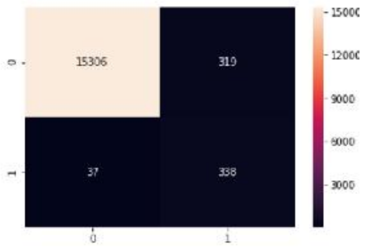
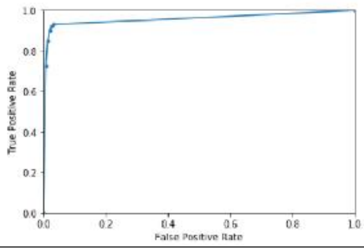
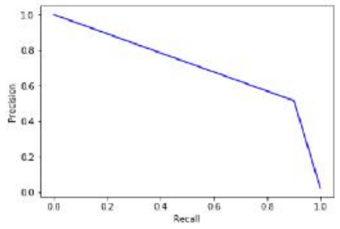
PLOTS	Confusion Matrix	ROC Curve	Precision-Recall Curve
NAÏVE BAYE'S CLASSIFIE R			
SUPPORT VECTOR MACHINE			
MULTI LAYER PERCEPTR ON			
RANDOM FOREST CLASSIFIE R			
KNN CLASSIFIE R			

Table: Comparison of the Confusion Matrix, ROC curves and the Precision-Recall Curves for all the classifiers

Discussion-

- The three tables compare the metrics of each classifier when the data fed to them is not reduced in dimensionality.
- The time taken for running each classifier was higher than the one without PCA (explained more above).
- Without PCA, the Random forest classifier performed the best with a less F1 score, lesser sensitivity.
- As a result, the Random Forest classifier was supposed to give us the least cost.
- But, the KNN classifier gave the least of the costs incurred by the company.
- This could be an anomaly as the KNN classified the test data more precisely than the Random Forest Classifier.
- The confusion matrices show how each of the classes is correctly classified or not, and the Random Forest seems to have misclassified the minority class as wrong.

4.2 Metrics/ Results of classifiers by using Dimensionality Reduction (PCA)

PARAMETERS	NAÏVE BAYE'S CLASSIFIER	SUPPORT VECTOR MACHINE	MULTI LAYER PERCEPTRON	RANDOM FOREST CLASSIFIER	KNN CLASSIFIER
Training Accuracy	89.68	97.1854	99.9237	94.1523	99.181
Testing Accuracy	96.3875	98.02499	98.5437	93.46875	97.75625
Precision (for minority class)	0.38	0.55	0.70	0.26	0.51
Recall (for minority class)	0.87	0.82	0.66	0.99	0.90
F1 – score (for minority class)	0.53	0.66	0.68	0.41	0.65
AUC score	0.9775	0.9851	0.8476	0.9825	0.9542
F1 Score	0.5308	0.6616	0.6795	0.4145	0.6538
Sensitivity	0.3815	0.5527	0.7017	0.2624	0.5120
Specificity	0.9968	0.9957	0.9918	0.9996	0.9976
FPR (False Positive Ratio)	0.03392	0.016	0.00672	0.0665	0.020672
TPR (True Positive Ratio)	0.872	0.824	0.6586	0.9866	0.904

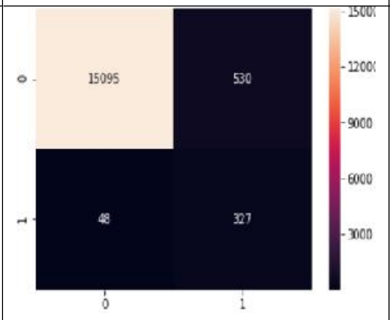
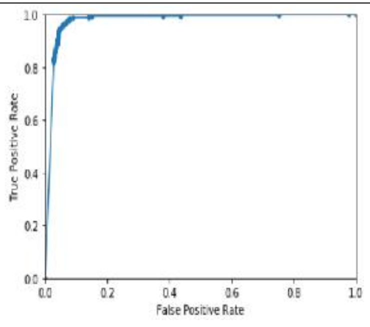
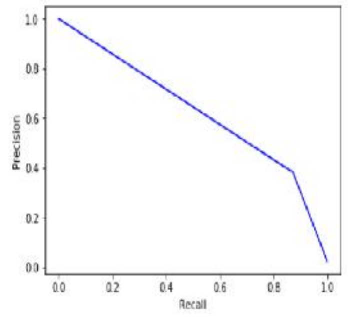
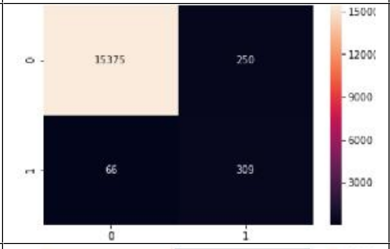
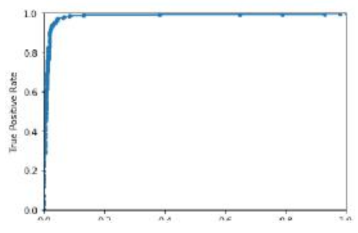
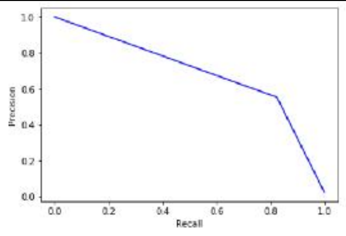
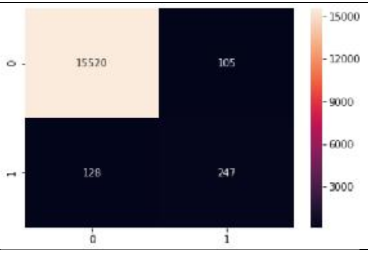
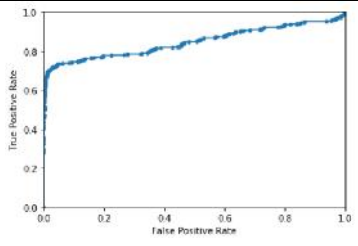
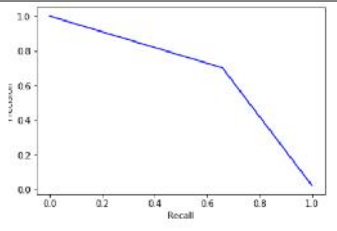
Table: Comparison of different metrics of all the classifiers with PCA

4.2 Cost Metrics/ Results of classifiers with using Dimensionality Reduction (PCA) -

TOTAL COST INCURRED	NAÏVE BAYE'S CLASSIFIER	SUPPORT VECTOR MACHINE	MULTI LAYER PERCEPTRON	RANDOM FOREST CLASSIFIER	KNN CLASSIFIER
Train Dataset	27010	8100	570	19160	3620
Test Dataset	29300	35500	65050	12900	21230

Table: Comparison of the cost incurred by SCANIA with the usage of different data sets and classifiers

4.2 Confusion Matrix, ROC Curve, Precision- Recall Curve Results of classifiers using Dimensionality Reduction (PCA) -

PLOTS	Confusion Matrix	ROC Curve	Precision-Recall Curve
NAÏVE BAYE'S CLASSIFIER			
SUPPORT VECTOR MACHINE			
MULTI LAYER PERCEPTR ON			

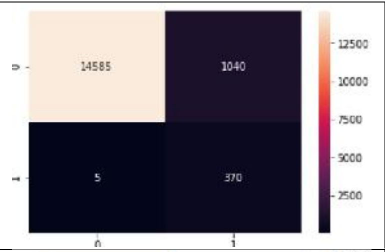
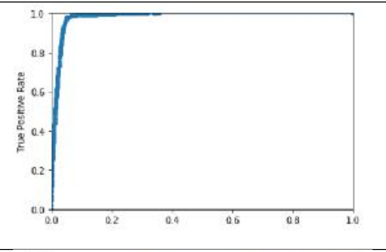
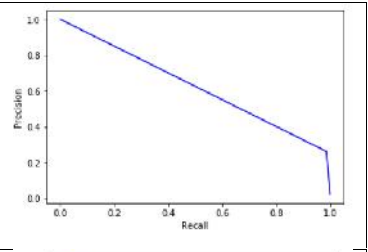
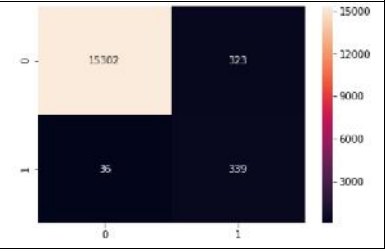
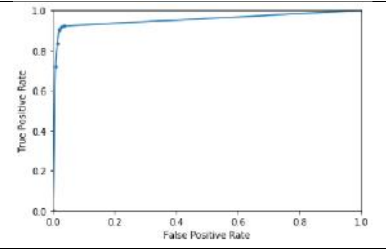
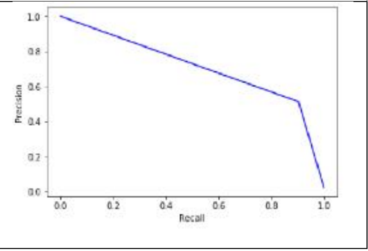
RANDOM FOREST CLASSIFIER			
KNN CLASSIFIER			

Table: Comparison of the Confusion Matrix, ROC curves and the Precision-Recall Curves for all the classifiers

Discussion:

- The time taken for each classifier to run was much lesser compared to the ones without PCA that is because the number of features has been reduced.
- Since, the number of features has been reduced, the accuracies of each of the classifiers increased significantly and the F1 scores decreased and therefore, the cost incurred by SCANIA increased significantly.
- Again. KNN classifier performed the best by giving the least cost incurred by the company.
- But, the Random Forest Classifier was able to correctly classify the minority class better than the other classifiers and this can be seen in the confusion matrix shown in the above table.

4.3 Metrics/ Results of classifiers for Average Cost using Cross- Validation

PARAMETERS	NAÏVE BAYE'S CLASSIFIER	SUPPORT VECTOR MACHINE	MULTI LAYER PERCEPTRON	RANDOM FOREST CLASSIFIER	KNN CLASSIFIER
Training Accuracy	89.6591	89.6973	99.9224	94.2406	99.0923
Testing Accuracy	89.6598	89.6979	99.6364	94.1879	98.7160
Recall	0.8293	0.807	0.998	0.95504	0.9987
F1 Score	0.8891	0.8895	0.9963	0.9426	0.98731
Sensitivity	0.9582	0.9587	0.9941	0.9305	0.97610
Specificity	0.8495	0.8498	0.9986	0.9538	0.99875

FPR (False Positive Ratio)	0.03615	0.0356	0.0058	0.0712	0.02445
TPR (True Positive Ratio)	0.82935	0.829	0.9986	0.9550	0.99877

Table: Comparison of different metrics of all the classifiers with PCA

<u>4.3 Cost Metrics/ Results of classifiers with using Cross-Validation -</u>

TOTAL COST INCURRED	NAÏVE BAYE'S CLASSIFIER	SUPPORT VECTOR MACHINE	MULTI LAYER PERCEPTRON	RANDOM FOREST CLASSIFIER	KNN CLASSIFIER
Train Dataset	1348580.0	1345030.0	9750.0	362200.0	7850.0
Test Dataset	337022.0	336404.0	2932.0	91204.0	3360.0

Table: Comparison of the cost incurred by SCANIA with the usage of different data sets and classifiers

Discussion:

- The addition of cross validation is to make the model run on all possible combinations of the inputs and hence, be able to validate the given data set by every other subset of the data set.
- Without the Cross Validation, the model can have the chances of performing excellently on the training set but, can decline when it comes to the test set.
- Therefore, with cross-validation helps in the verification of the error performance of the given model.
- In order to test the model's success in the classification with the samples that are not made use of while building the classification model.
- Therefore, cross- validation kind of gives the actual estimates of the predictions of the model on the new data.
- Above, you can see that the cost of the test data has significantly reduced from that of the training data and this is because of the validation of the main dataset is done by validating all the other subsets of the data.

4.4 Model Selection/Parameter Tuning

<u>Naïve Baye's-</u>

- No Parameter Selection required. Since, sklearn for Naïve Baye's doesn't have any parameters that can be modified significantly to cause an increase in the performance of the model.

Support Vector Machine –

BEST MODEL

PARAMETERS	MODEL 1 C=1	MODEL 2 C=25	**MODEL 3 C=50**
Sensitivity	0.980	0.9846	**0.9859**
Specificity	0.9550	0.9673	**0.9732**
Precision	0.9809	0.9846	**0.9859**
Recall	0.9537	0.9666	**0.9727**
F1 Score	0.9671	0.9755	**0.9792**
FPR	0.0185	0.0150	**0.0138**
TPR	0.9537	0.9666	**0.9727**
Training Data COST	302070.0	218380.0	**177830.0**
Testing Data COST	152720.0	110320.0	**90190.0**

Table: Comparison of the three models considered by varying the C-value

Discussion:

- In SVM, the C-value or the Penalty parameter of the error term is significantly increased.
- For the Baseline system above, the C-value was give as default – 1.
- So, this C-value was increased to 25 and the 50 and the performance was noted down.
- For evaluation of the performance I considered the F1 score and the Training data set cost and the test data set cost.
- The F1 score basically gives us the test dataset's accuracy and this is best if it is more. Excellent when it is 1.
- The training and the testing costs have to be as less as possible.
- As you can see above that with increase in the values of C, the F1 scores have increased and the costs incurred have reduced.
- Reason – The C-value hyperparameter tells the system the amount of misclassification to be avoided. Therefore, setting a smaller margin for the classification and consequently, the improvement in performance.

Multi-Layer Perceptron–

BEST MODEL

PARAMETERS	**MODEL 1 Number of Layers = [200,100], Baseline model =[100,50]**	MODEL 2 Solver = lgdfs Baseline model= adam	MODEL 3 Activation = tanh Baseline model = relu
Sensitivity	**0.9935**	0.9919	0.990
Specificity	**0.99862**	0.9982	0.998
Precision	**0.99352**	0.9919	0.9903

Recall	**0.99862**	0.9982	0.998
F1 Score	**0.99606**	0.9950	0.9942
FPR	**0.0065**	0.0081	0.0097
TPR	**0.998627**	0.9982	0.9981
Training Data COST	**8870.0**	7590.0	11400.0
Testing Data COST	**4930.0**	6200.0	6800.0

Table: Comparison of the three models considered by varying the number of hidden layers, type of solver used, type of activation used

Discussion-

- In MLP, I have varied three parameters – number of hidden layers, type of solver used, and the type of activation that is used.
- The base model had 100,50 hidden layers. Increasing the hidden layer size will increase the performance of the model as the number of hidden layers increases, the model undergoes optimization which leads to the better performance.
- With the change in the solver to lgdfs, there wasn't an increase in the performance. They are used for weight optimization. The baseline model with the adam was the best because since the dataset considered here is pretty high and adam performs well for the larger datasets
- The lgdfs solver converges faster and performs better on smaller datasets.
- The activation type was changed from relu to tanh and there was a decrease in the performance of the model because relu is the best activation type since it is a rectified linear unit function and has better error rectification than any other activations.
- The learning rate was not chosen to vary because the base line model already had the least learning rate and the best performance since learning rates is the parameter which sets a schedule for the weight updates.

Random Forest Classifier-

BEST MODEL

PARAMETERS	MODEL 1 Number of estimators = 1 Baseline model = 100	MODEL 2 Number of estimators = 25 Baseline model = 100	**MODEL 3 Number of estimators = 500 Baseline model = 100**
Sensitivity	0.90940	0.9186	**0.922**
Specificity	0.74867	0.8513	**0.885**
Precision	0.90940	0.9186	**0.922**
Recall	0.68722	0.8213	**0.865**
F1 Score	0.78281	0.8621	**0.888**
FPR	0.06849	0.0713	**0.071**

TPR	0.68722	0.8213	**0.8657**
Training Data COST	2057030.0	1177630.0	**886400.0**
Testing Data COST	1029650.0	590260.0	**444630.0**

Table: Comparison of the three models considered by varying the number of Estimators

Discussion –

- In RF classifier, I tried t vary the number of estimators given.
- First, I reduced the number of estimators from 100 to 1 and tried to slowly increase it to 500.
- With the increase in the number of estimators, the performance got better.
- With the increase in the number of estimators, the number of trees increases and the Random forest ensemble model averages over all the trees considered.
- The performance, however is not very applaudable since the cost is still significantly higher.

<u>KNN –</u>

BEST MODEL

PARAMETERS	**MODEL 1 Number of Neighbours = 1**	MODEL 2 Number of Neighbours = 25	MODEL 3 Number of Neighbours = 500
Sensitivity	**0.9866**	0.9739	0.9677
Specificity	**0.9996**	0.9964	0.9914
Precision	**0.9866**	0.9739	0.9677
Recall	**0.99964**	0.9965	0.9917
F1 Score	**0.9930**	0.9851	0.9795
FPR	**0.0135**	0.0267	0.0331
TPR	**0.9996**	0.9965	0.9917
Training Data COST	**1020.0**	21040.0	53670.0
Testing Data COST	**2060.0**	12920.0	29340.0

Table: Comparison of the three models considered by varying the number of neighbours

Discussion –

- In this, the number of neighbours have been decreased and then been increased.
- Smaller values of k will give a much more flexible fit but having a very low bias and a very high variance.
- Therefore, the best performance for the Number of Neighbours = 1

- The model with more number of neighbours has a much more tighter fit than all the others, Therefore, the reduced performance.
- But, among all the other classifiers, KNN performed the best but it was executed with the most amount of time since the observations were made over a larger data set.

5. CONTRIBUTIONS OF EACH TEAM MEMBER / DIFFICULTY SCORE-

- I have chosen the APS dataset and performed the training and testing of a classifier model by choosing 5 classifiers and their hyper-parameters to choose the best model.
- I have done this project myself with a lot of help from the lectures and online resources.

6. SUMMARY AND CONCLUSIONS –

- I have used five different methods for Data Analysis as stated in the previous sections
- I have done three types of Feature Importance techniques which give out the best features that need to be considered.
- The Data cleaning and Imputation was done using three methods as stated in the previous sections along with the names and procedure.
- I have used 5 different classifiers and noted all their possible metrics and how they behave without PCA, with PCA, with Cross-Validation.
- I have done model selection for all the classifiers by varying their hyper parameters
- The best model that could be chosen for classifying the APS dataset is –

Classifier - KNN

- **Number of Neighbours – 1**
- **Data Imputation – Mean**
- **F1 score - 0.9930**
- **Training Data COST - 1020.0$**
- **Testing Data COST - 2060.0$**

REFERENCES –

https://towardsdatascience.com/building-a-k-nearest-neighbors-k-nn-model-with-scikit-learn-51209555453a

https://stats.stackexchange.com/questions/118268/is-knn-best-for-classification

https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

https://www.researchgate.net/post/How_to_determine_the_number_of_trees_to_be_generated_in_Random_Forest_algorithm

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

https://stats.stackexchange.com/questions/338255/what-is-effect-of-increasing-number-of-hidden-layers-in-a-feed-forward-nn/338374

https://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel

https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

https://en.wikipedia.org/wiki/F1_score

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

https://www.researchgate.net/post/What_is_the_purpose_of_performing_cross-validation

https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

https://towardsdatascience.com/a-feature-selection-tool-for-machine-learning-in-python-b64dd23710f0

https://scikit-learn.org/stable/modules/feature_selection.html

https://kite.com/blog/python/data-analysis-visualization-python?fbclid=IwAR2Pm1FekrrbRQzGrBy79NrKPUtv0GJQajiL6NAD_GYnLrguoCjD5OrXPmQ

https://towardsdatascience.com/visualizing-your-exploratory-data-analysis-d2d6c2e3b30e?fbclid=IwAR202sNzWxx_PbFepwk3AbHTMQNgbXsecjwSmykW7WQatM7ohwtqyywXcow