Muhammad Asif Khan
S19004470

Applied Machine Learning Project

1. Data Set

1.1 Data Set Information:

For this project, the database is taken from the University of California Irvine(UCI) Machine Learning Repository. It is called 'The Heart Disease' and can be downloaded by using the following link:

https://archive.ics.uci.edu/ml/datasets/Heart+Disease/

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. The subset version with 14 fields will be used in this project. The names and social security numbers of the patients were removed from the database and replaced with dummy values [1].

1.2 Data Set Attribute Information:

age - Age of the patient

sex - Sex of the patient ~ (1:male , 0: female)

cp - Chest pain type ~ 0 = Typical Angina, 1 = Atypical Angina, 2 = Non-anginal Pain, 3 = Asymptomatic

trtbps - Resting blood pressure (in mm/Hg)

chol - Cholestoral in mg/dl fetched via BMI sensor

fbs - (fasting blood sugar > 120 mg/dl) ~ 1 = True, 0 = False

restecg - Resting electrocardiographic results ~ 0 = Normal, 1 = ST-T wave normality, 2 = Left ventricular hypertrophy

thalachh - Maximum heart rate achieved

oldpeak - Previous peak

slp - Slope

caa - Number of major vessels ~ (0-3)

thall - Thalium Stress Test result ~ (0,3)

exng - Exercise induced angina ~ (1 = Yes, 0 = No)

output - Target variable ~ 0=less chance of heart attack, 1= more chance of heart attack

2. Task

In this project, the aim is to build and train a classification model in MATLAB programming environment by using various Machine Learning (ML) techniques to understand what are the reasons which can affect the heart attack. The ML classification model should be able to predict if a person suffers with a heart condition based on the given evidence.

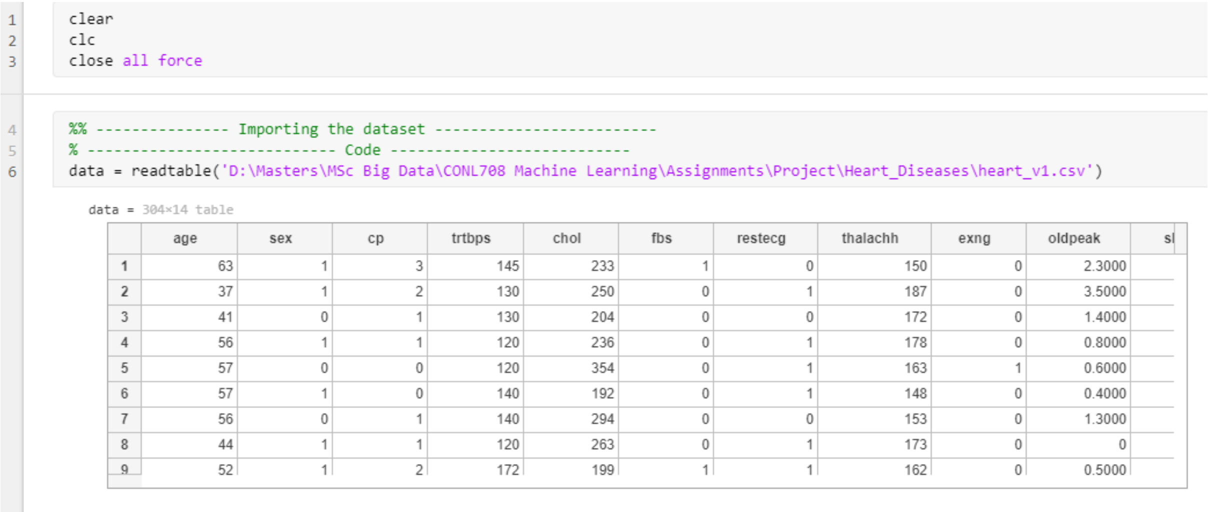
It will be interesting to discover the impact of certain features like age, cholesterol level and blood pressure on the heart attack.

3. Plan

At the first stage, the data will be accessed and loaded in the MATLAB environment. Then, at the second stage, the data is pre-processed. In the third stage, features are extracted from the pre-processed data. At the fourth stage, models are trained using the extracted features in the second stage. At the fifth stage, an iteration is carried out continuously until an optimum or near-optimum model is found and finally the best trained ML model will be deployed in a production system [3].

4. (Stage 1) Access and Load the data

The following code is used to upload the data in MATLAB area as shown in fig 2.



```
clear
clc
close all force
```

```
%% -------------- Importing the dataset ------------------------
% --------------------------- Code --------------------------
data = readtable('D:\Masters\MSc Big Data\CONL708 Machine Learning\Assignments\Project\Heart_Diseases\heart_v1.csv')
```

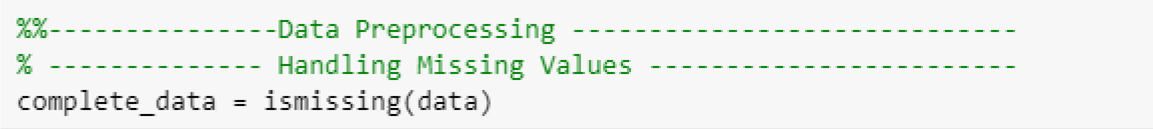
data = 304×14 table

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	sl
1	63	1	3	145	233	1	0	150	0	2.3000	
2	37	1	2	130	250	0	1	187	0	3.5000	
3	41	0	1	130	204	0	0	172	0	1.4000	
4	56	1	1	120	236	0	1	178	0	0.8000	
5	57	0	0	120	354	0	1	163	1	0.6000	
6	57	1	0	140	192	0	1	148	0	0.4000	
7	56	0	1	140	294	0	0	153	0	1.3000	
8	44	1	1	120	263	0	1	173	0	0	
9	52	1	2	172	199	1	1	162	0	0.5000	

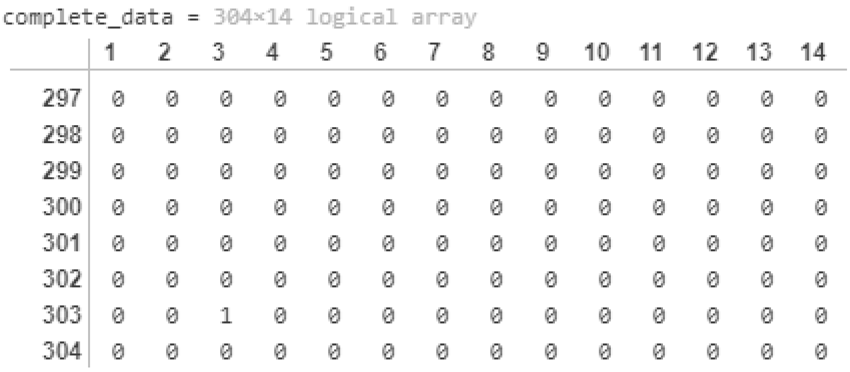
Fig-2

5. (Stage 2) Pre-process the data

5.1 Fill Missing Data



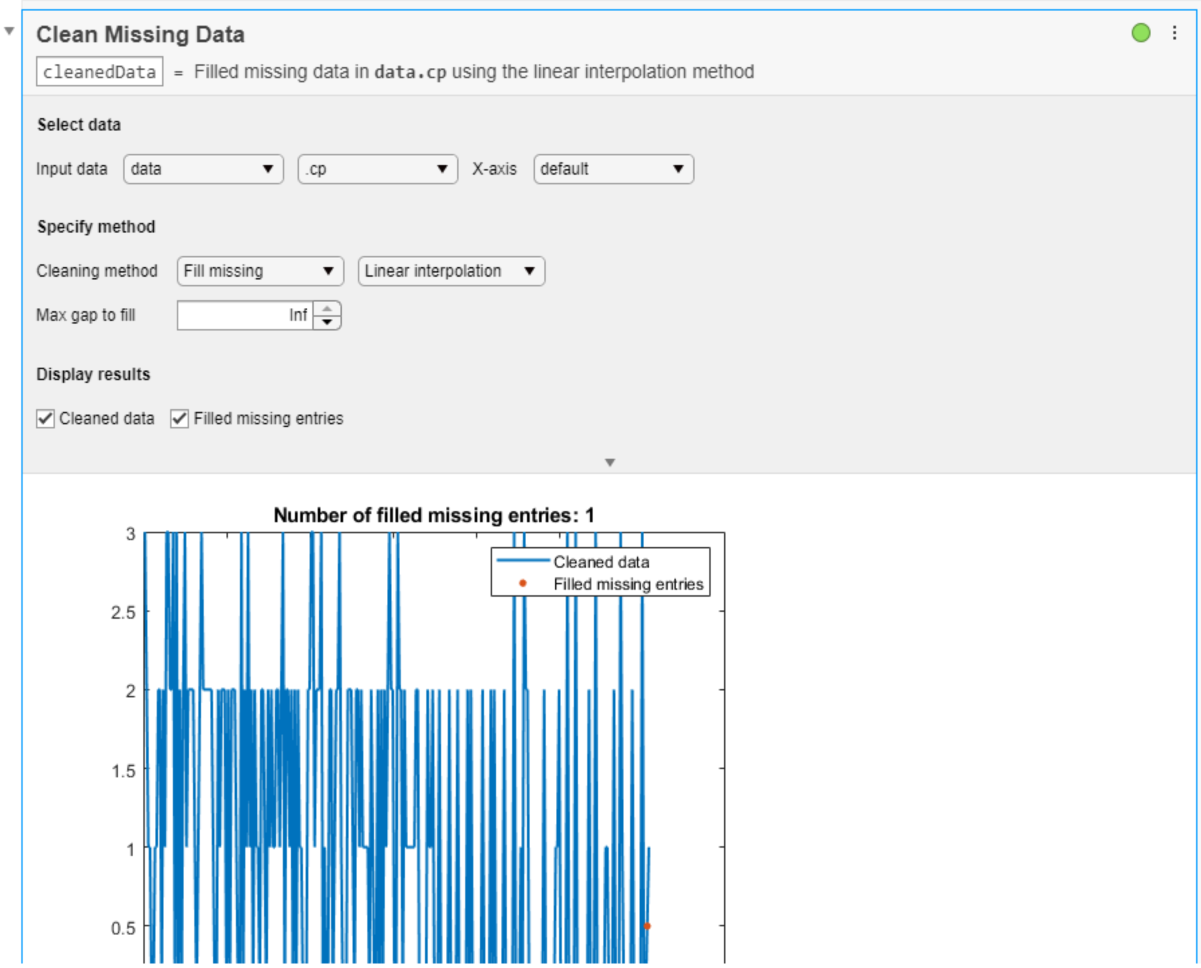
```
%%--------------Data Preprocessing ----------------------------
% -------------- Handling Missing Values ------------------------
complete_data = ismissing(data)
```



```
complete_data = 304×14 logical array
       1   2   3   4   5   6   7   8   9  10  11  12  13  14
 297   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 298   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 299   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 300   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 301   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 302   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 303   0   0   1   0   0   0   0   0   0   0   0   0   0   0
 304   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

Fig - 3

To replace the NaN value at (303,3), type the keyword missing in a code block, and then click Clean Missing Data when it appears in the menu. Select the input data and the cleaning method to plot the filled data automatically as shown below in Fig-4.



Clean Missing Data

cleanedData = Filled missing data in `data.cp` using the linear interpolation method

Select data

Input data data ▼ .cp ▼ X-axis default ▼

Specify method

Cleaning method Fill missing ▼ Linear interpolation ▼

Max gap to fill Inf

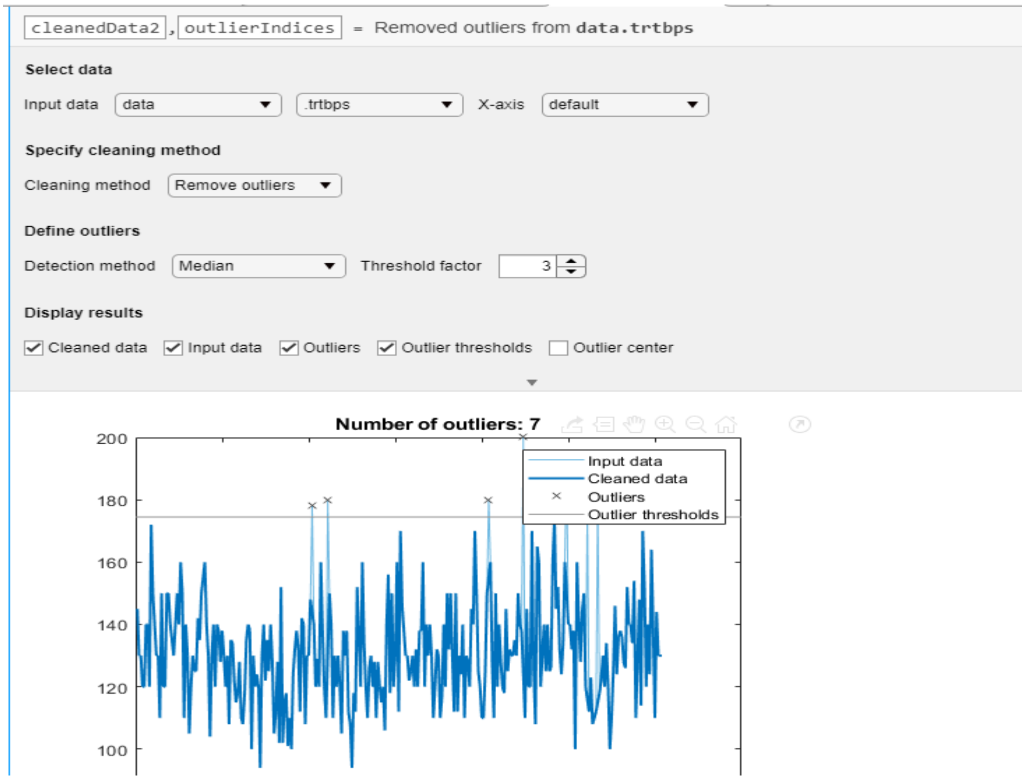
Display results

☑ Cleaned data ☑ Filled missing entries

Fig - 4

5.2 Fill Outliers

In this project, outliers need to be removed because they could seriously impact the training of ML models. To remove the outliers from the given data set, use the Clean Outlier Data task to identify outliers and remove them as shown below in Fig-5.



cleanedData2 , outlierIndices = Removed outliers from data.trtbps

Select data

Input data data .trtbps X-axis default

Specify cleaning method

Cleaning method Remove outliers

Define outliers

Detection method Median Threshold factor 3

Display results

☑ Cleaned data ☑ Input data ☑ Outliers ☑ Outlier thresholds □ Outlier center

Fig - 5

5.3 Data Normalisation

This task is not required for this project because we will also use standardised scales to compare different features in the next stage.

6 (Stage 3) Derive features using the pre-processed data

Now the data has been pre-processed, we need to select features which will help us to improve a machine learning algorithm by focusing on the data features that are most likely to produce accurate results. For this purpose, we will the Parallel Coordinates Plot to compare the feature of several individual observations as shown below in Fig-6 [5].

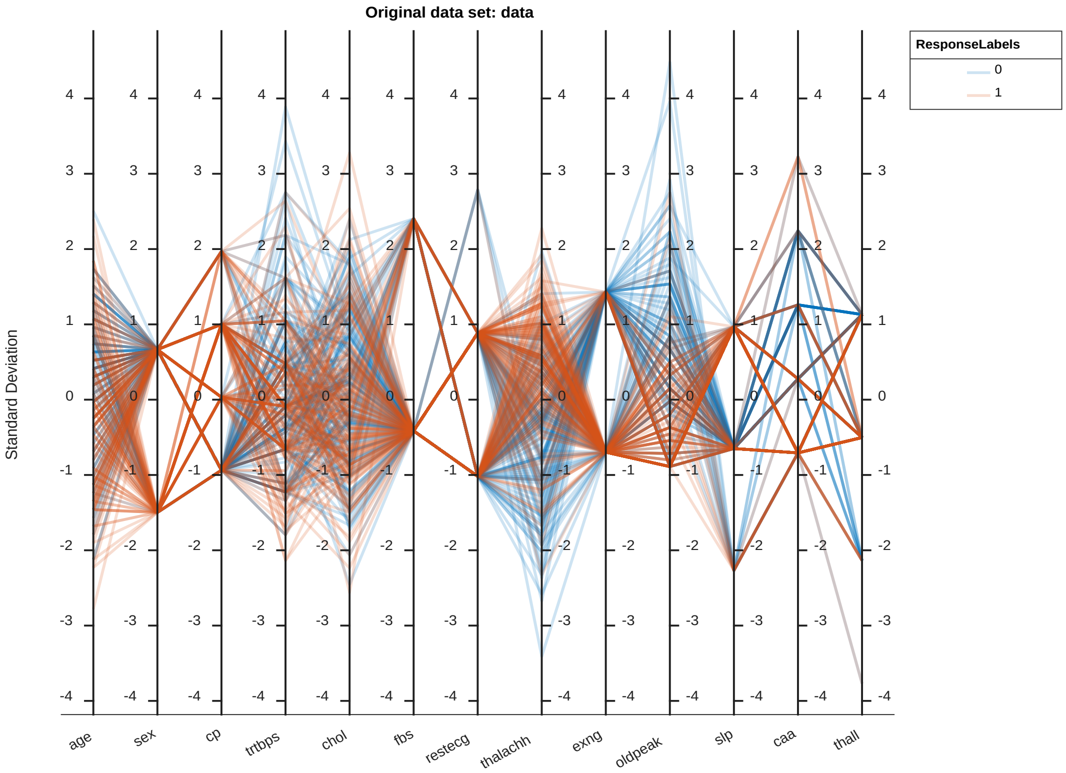


Fig - 6

From the Parallel Coordinates Plot, we have deduced the following features:

- The data consists of more than twice the number of people with sex = 1 (male) than sex = 0 (female).

- People with higher maximum heart rate (***thalachh)*** and lower previous peak (***oldpeak)*** have higher chances of heart attack.

There are some interesting features regarding bivariate analysis revealed by the following Scatter plots as shown in Fig-7.

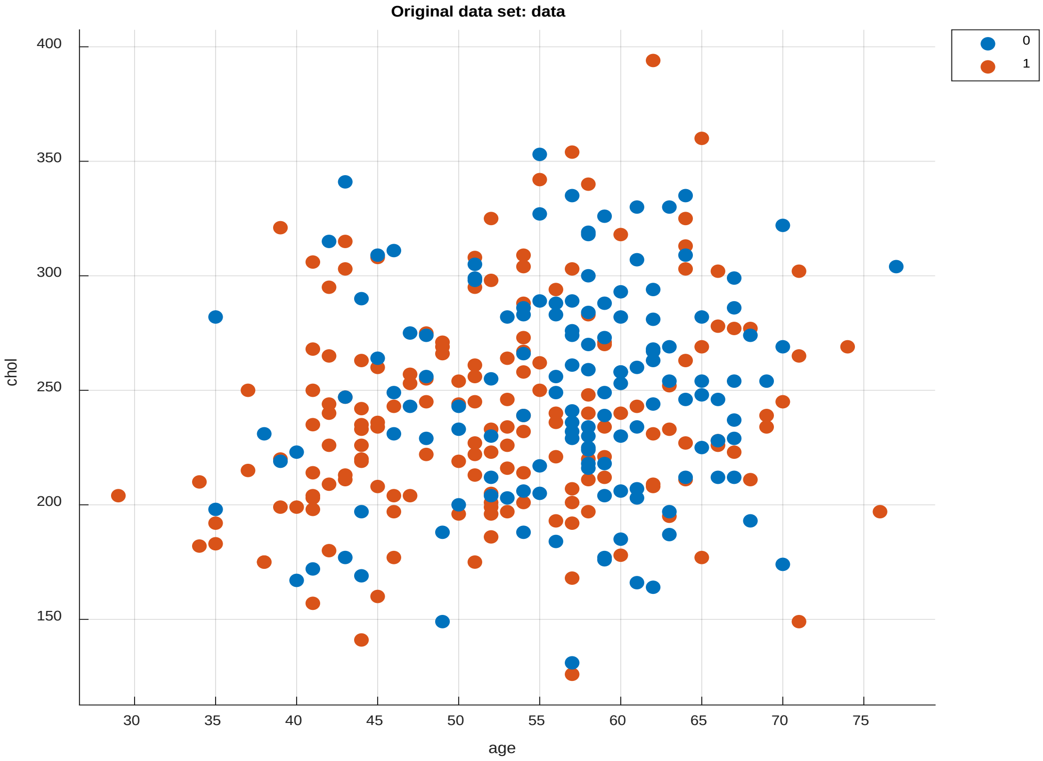


Fig - 7

It is intuitive that elder people might have higher chances of heart attack; as we thought initially, turn out to be incorrect. According to the scatter plot of age with respect to cholesterol in Fig-7, it is the cholesterol level of more than 200mg/dl that is causing heart attacks in 80% of population.

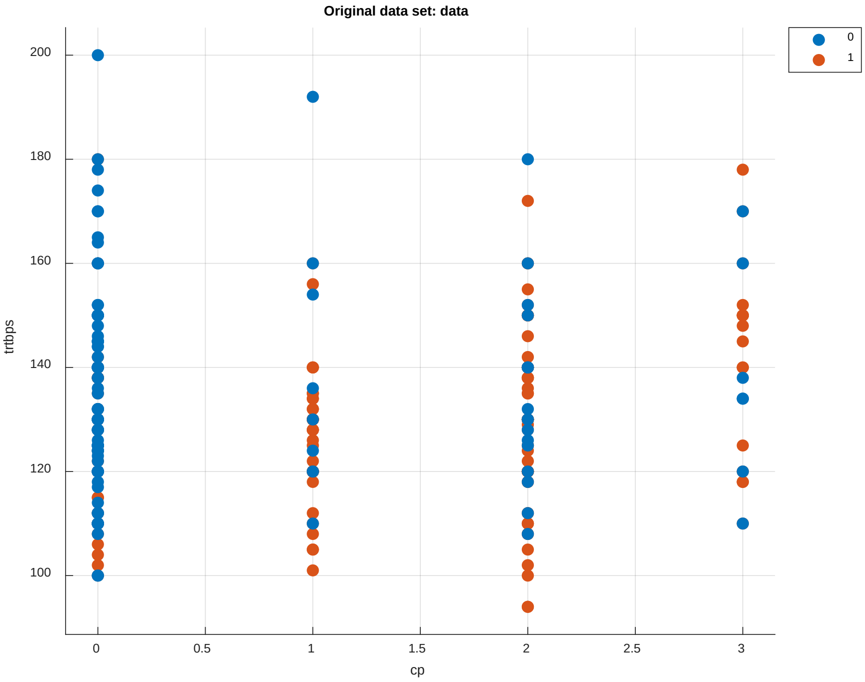


Fig - 8

People with Non-Anginal chest pain, that is with cp = 1 or higher have higher chances of heart attack as shown above in Fig -8.

By using the above exploratory data analysis and the domain knowledge, we have decided to reduce the number of features from 13 to 4 as shown below.

Model Predictors:

cp - Chest pain

trtbps - Resting blood pressure

chol - Cholestoral level

thalachh - Maximum heart rate achieved

Model Class:

output - Target variable ~ 0=less chance of heart attack, 1= more chance of heart attack

7 The selection of Classification Techniques to Build and Train ML Model

The following three ML techniques have been chosen to build and train models to solve the given problem based on the following rationales.

1) **Decision Tree Classification**

   It is chosen because it works very well with categorical featured data like our data set. It provides very good interpretation by using effective visualisation tress with several levels. It is prone to overfitting; therefore, we are going to use cross-validation with 5 folds to avoid this problem [7].

2) **Support Vector Machine (SVM)**

   We have decided using SVM because is very effective when dealing with multiple features with a weak correlation between each other just like in our data set. Furthermore, it is capable of handling data set with high dimension. As there is no linear correlation between key features, we will be using non-linear kernel while optimising the corresponding ML model [8].

3) **K – Nearest Neighbours(KNN)**

   The main rationale to use KNN is because it performs well with data with many features because it uses a nonlinear classifier and hence, the prediction boundary is also non-linear. We will be using Euclidean distance to measure distance between k nearest neighbours.

8 (Stage 4) Train models using the extracted features

To explore classification models interactively, we will use the Classification Learner app. First, let us start with all 13 features and train our model by using Decision Tree algorithm. We are not going to use any of the hyperparameters for optimisation. After training and testing the first model, we have got the following details:

Basic Tree model with no feature selection and optimisation

Model Details	Results	Optimiser Options	Feature Selection and PCA
Type: Fine Tree ***Max. number of splits:*** 100 ***Split Criterion:*** Gini's Diversity Index ***Surrogate Decision Splits:*** Off	***Accuracy:*** 64% ***Total misclassification cost:*** 66 ***Prediction Speed:*** ~7500 obs/sec Training Time: 3.8 sec	***Hyperparameter option:*** Disabled	***No of features being used:*** 13/13 ***PCA:*** Disabled

Fig - 9

The above model has 64% accuracy which is very poor, so we need to use feature selection and optimisation to improve its performance and accuracy.

By using the above values and other key diagnostic measures like scatter plot, confusion matrix chart, Area under the Curve(AUC), and ROC values; we should be able to compare validation results and choose the best model that works for your classification problem [4].

8.1 Decision Tree Classification (Model 1)

Our first model is based on Decision Tree. We have used feature reduction and predictors are reduced to 4 from 13. We are also using auto optimisation to improve accuracy.

Key Diagrams and Statistics

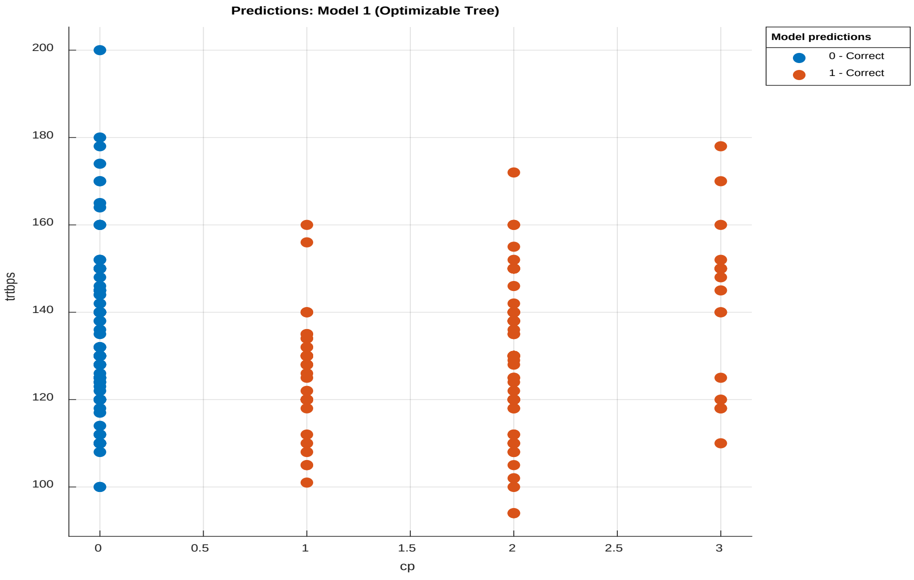


Fig -10

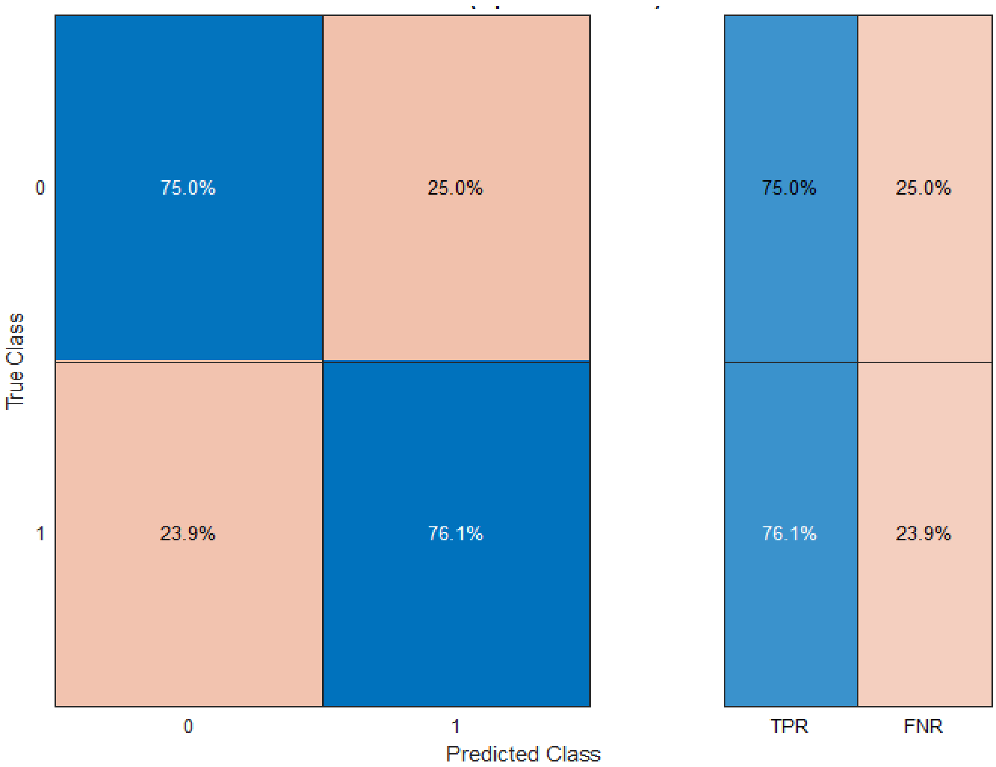

True Class
0
1
75.0%
25.0%
23.9%
76.1%
0
1
Predicted Class
75.0%
25.0%
76.1%
23.9%
TPR
FNR

Fig-11

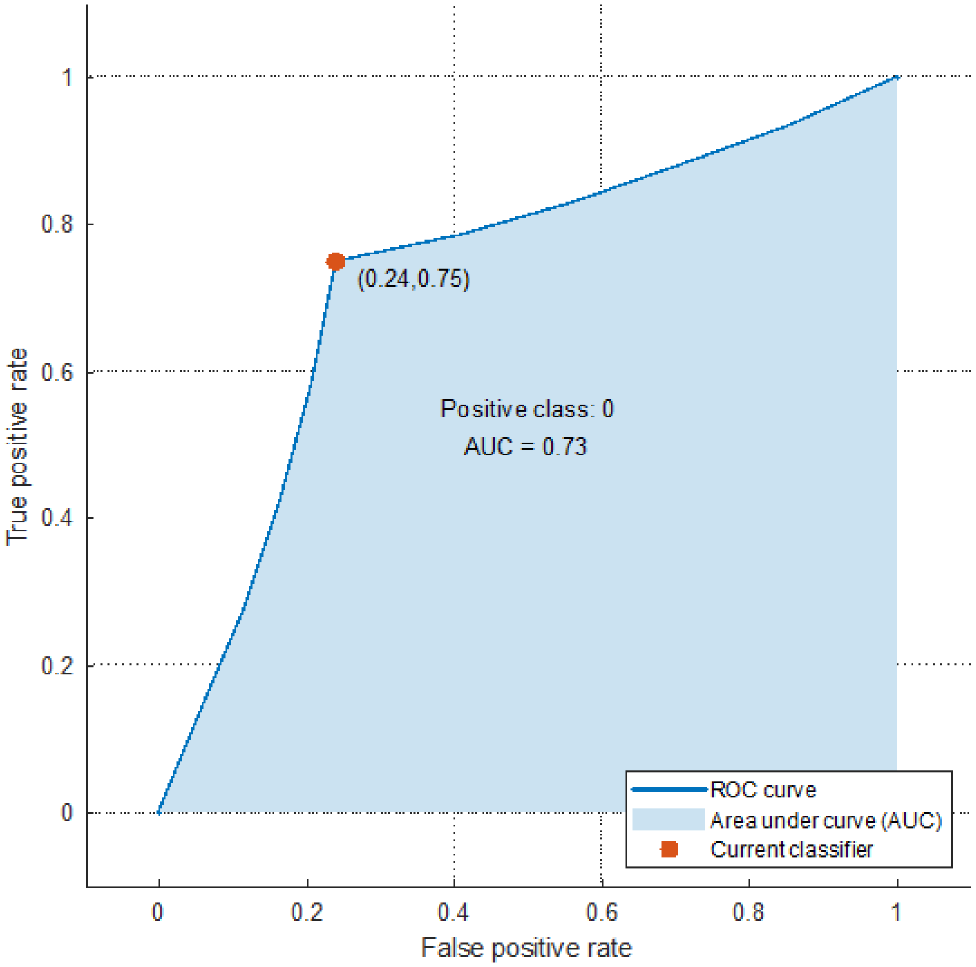

True positive rate
1
0.8
0.6
0.4
0.2
0
(0.24,0.75)
Positive class: 0
AUC = 0.73
ROC curve
Area under curve (AUC)
Current classifier
0
0.2
0.4
0.6
0.8
1
False positive rate

Fig - 12

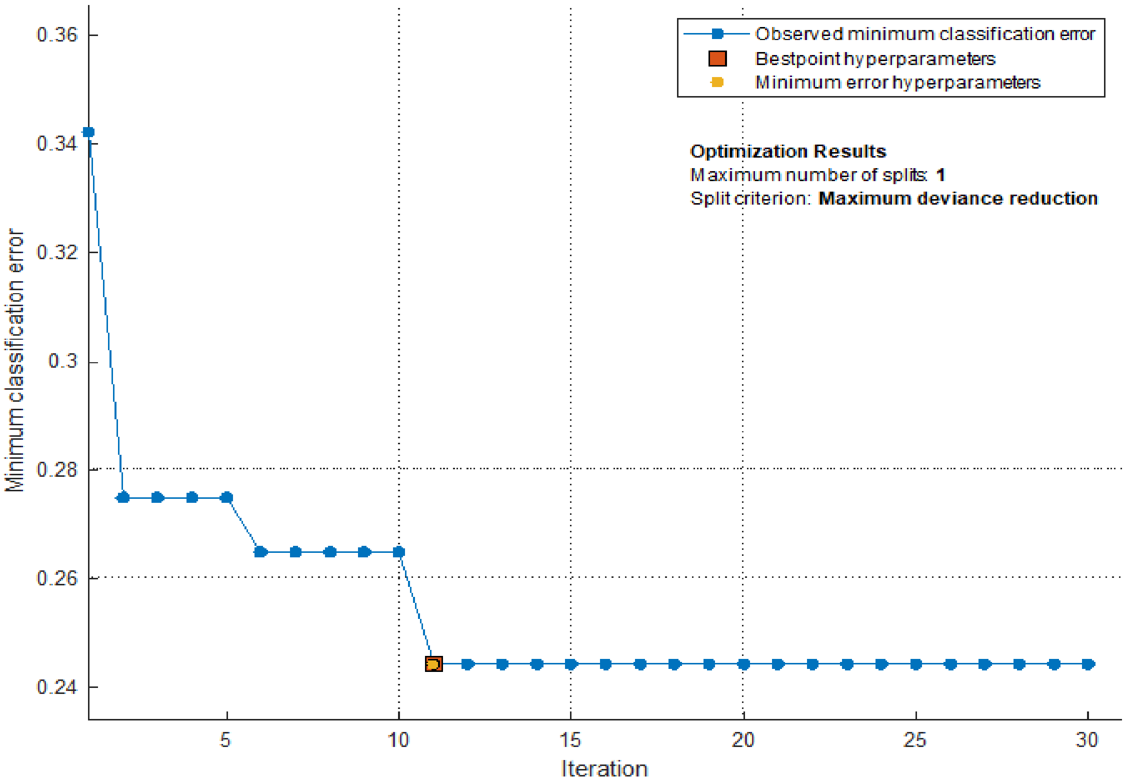


Fig – 13

Model Details	**Results**	**Optimiser Options**	**Feature Selection and PCA**
Type: Optimised Tree ***Max. number of splits:*** 1-298 ***Split Criterion:*** Gini's Diversity Index ***Surrogate Decision Splits:*** Off	***Accuracy:*** 75.6% ***Total misclassification cost:*** 73 ***Prediction Speed:*** ~33000 obs/sec Training Time: 22.344 sec	***Hyperparameter option:*** Enabled (Random Search with 45 iterations)	***No of features being used:*** 4/13 ***PCA:*** Disabled

Fig-14

Model 1 Performance Evaluation

- The accuracy of the Tree model has improved from 64% to 75.6% after using optimisation
- The prediction speed has improved from 35000 obs/sec to 7500 obs/sec because we only used 4 features out of 13 for predicting the class.
- Area under the curve (AUC) is also improved from 0.634 (see Fig-9) to 0.73 (see Fig-12) which increases the probability of a better prediction.
- This model is slower because its training time is 22.344 sec due to 45 iterations to optimise the classifier while the basic tree model had only 3.8 sec of training time (see Fig-9)

8.2 Support Vector Machine (Model 2)

Our second model is based on Support Vector Machine (SVM). Again, feature reduction and auto optimisation will be used to improve accuracy.

Key Diagrams and Statistics

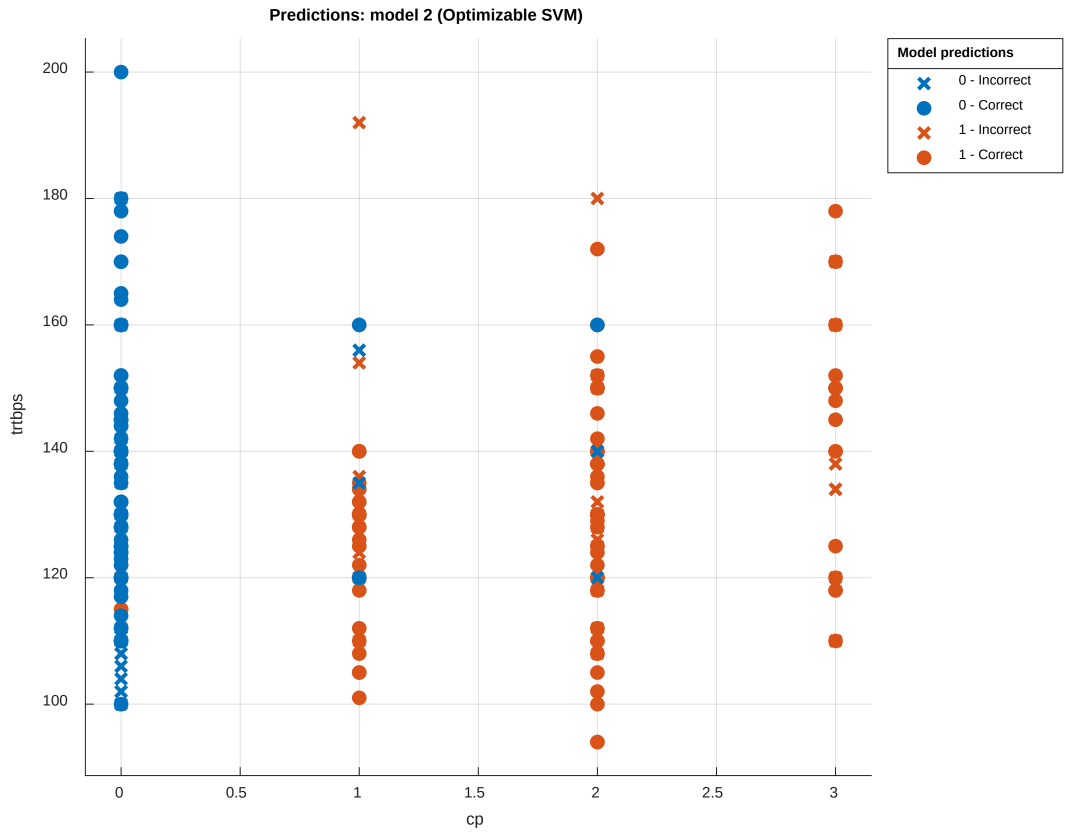


Fig – 15

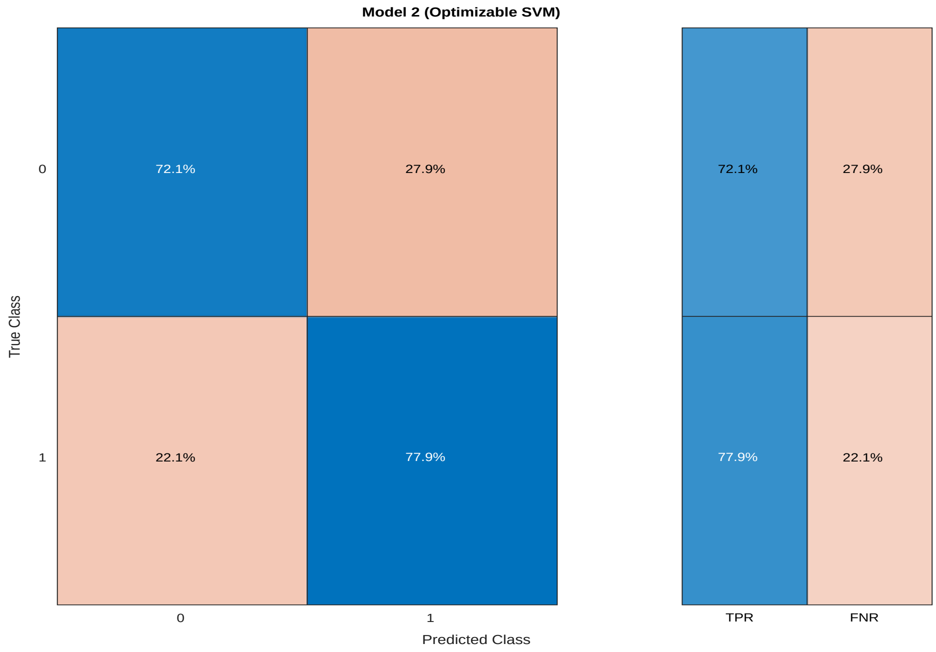


Fig – 16

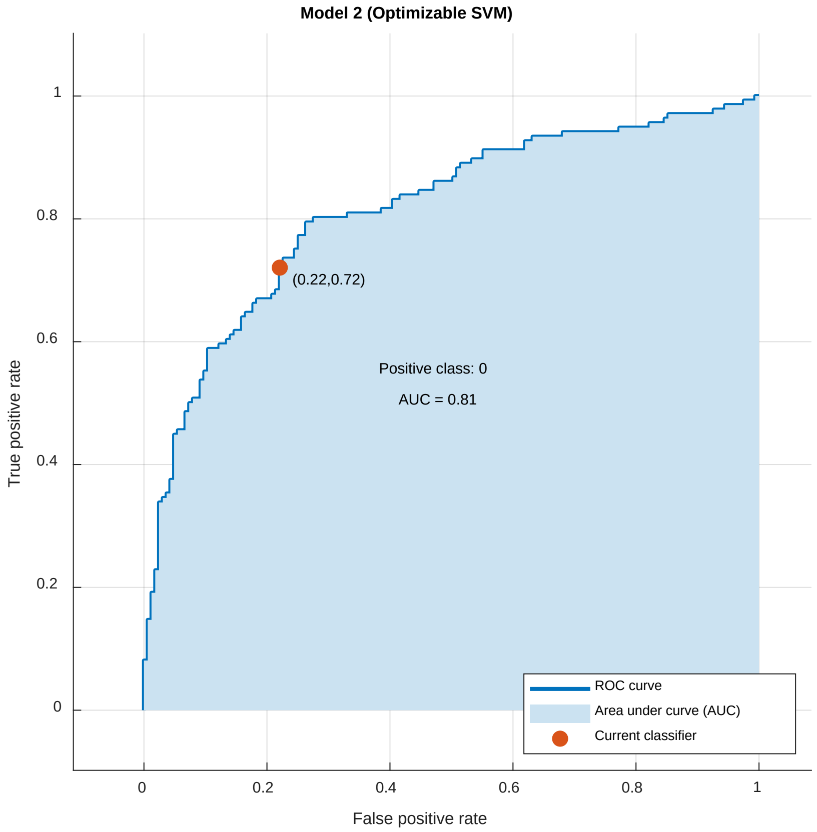

Model 2 (Optimizable SVM)
True positive rate
False positive rate
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
(0.22,0.72)
Positive class: 0
AUC = 0.81
ROC curve
Area under curve (AUC)
Current classifier

Fig – 17

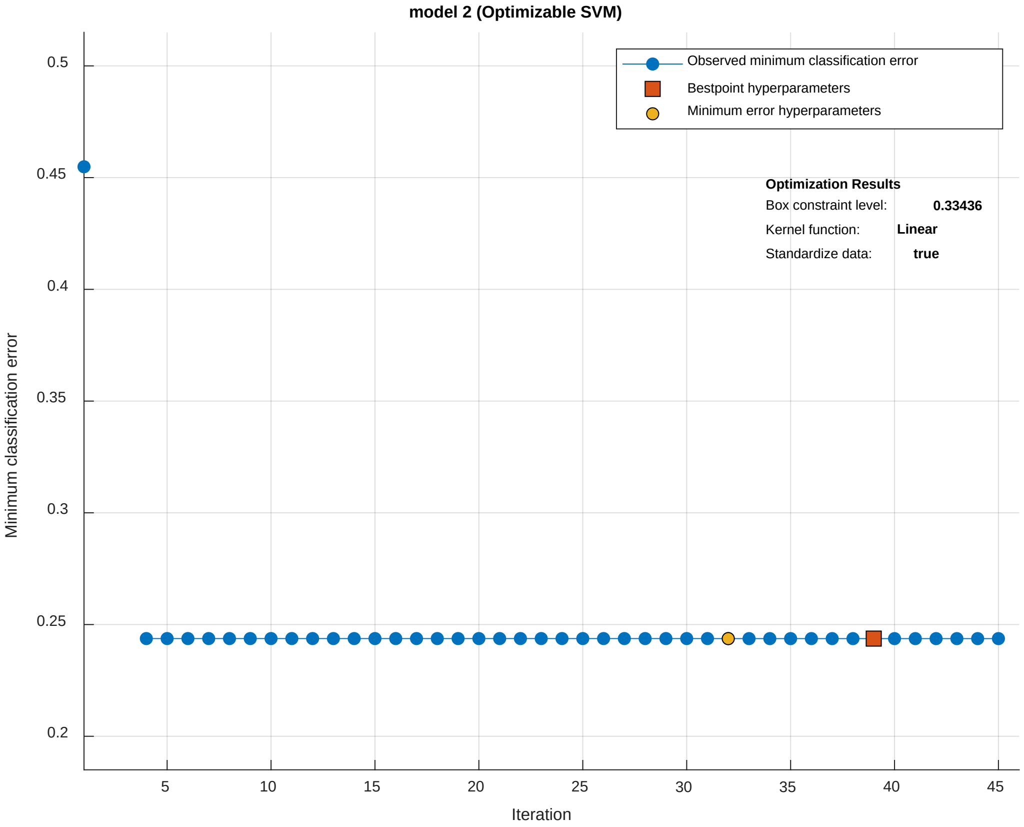

model 2 (Optimizable SVM)
Minimum classification error
Iteration
0.5
0.45
0.4
0.35
0.3
0.25
0.2
5
10
15
20
25
30
35
40
45
Observed minimum classification error
Bestpoint hyperparameters
Minimum error hyperparameters
Optimization Results
Box constraint level: 0.33436
Kernel function: Linear
Standardize data: true

Fig – 18

Model Details	Results	Optimiser Options	Feature Selection and PCA
Type: Optimised SVM ***Kernal Scale:*** 1 with One-vs-One multiclass method ***Kernal function:*** Gaussian Standardise ***Surrogate Decision Splits:*** Off	***Accuracy:*** 75.3% ***Total misclassification cost:*** 74 ***Prediction Speed:*** ~46000 obs/sec Training Time: 134.74 sec	***Hyperparameter option:*** Enabled (Linear with Box constraint with 45 iterations)	***No of features being used:*** ***4***/13 ***PCA:*** Disabled

Fig – 19

Model 2 Performance Evaluation

- Model 2 percentage accuracy is 75.3% which is very similar to the model 1 which was 75.6% (see Fig-14)
- The prediction speed has gone up in the model 2 with 46000 obs/sec as compared to model 1 with 33000 obs/sec.
- Training time for the model 2 has also gone up from 134.74 sec as compared to 22.344 sec for the model 1.
- Area under the curve (AUC) has improved for the model 2 with the value of 0.81 (see Fig-17) against 0.73 for the model 1 (see Fig-12).

8.3 K Nearest Neighbours or KNN (Model 3)

Our last model is based on K Nearest Neighbours(KNN). The feature reduction will be used alongside with a nonlinear classifier for better optimisation.

Key Diagrams and Statistics

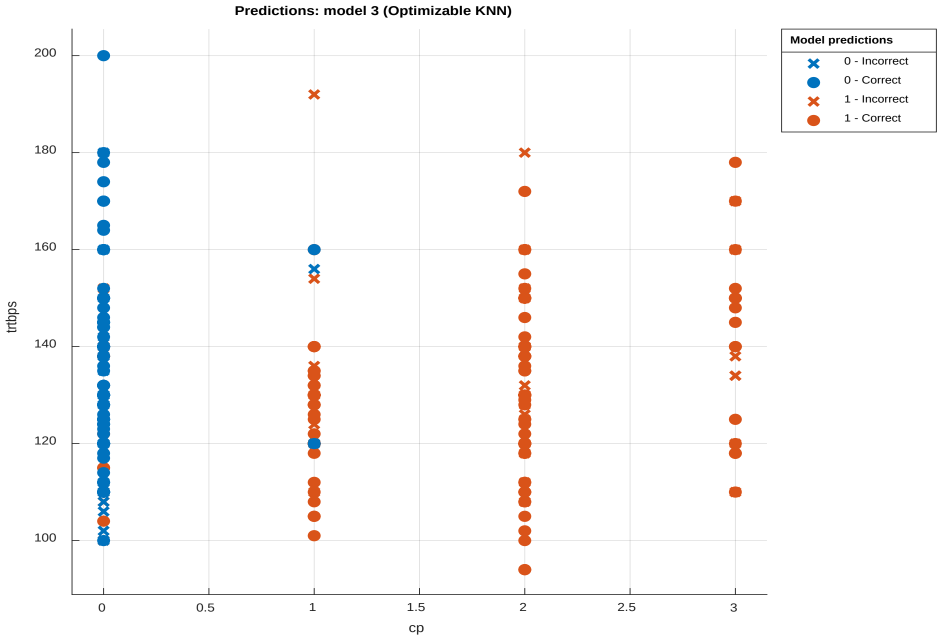


Fig – 20

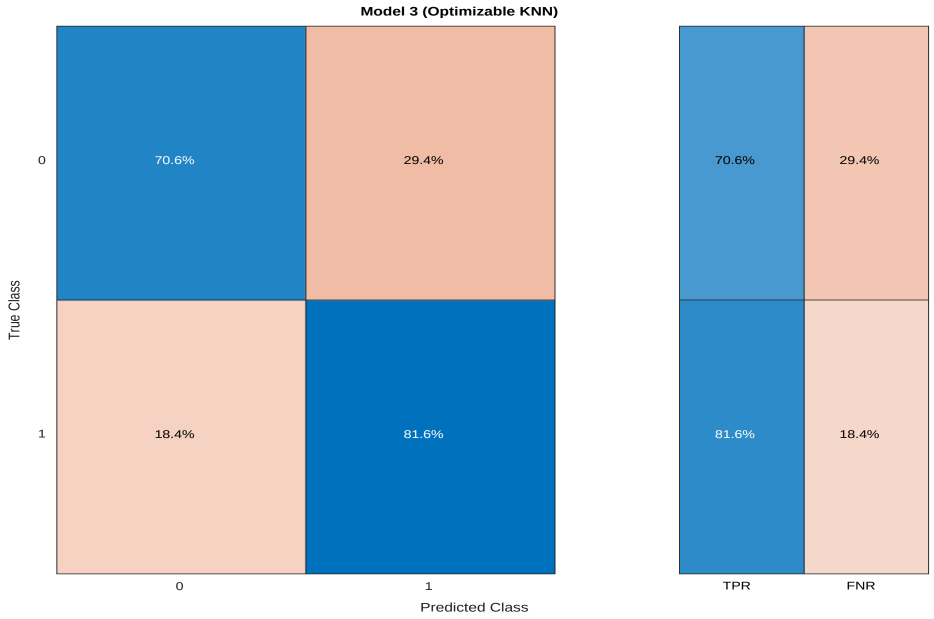

Model 3 (Optimizable KNN)
True Class
0
1
70.6%
29.4%
18.4%
81.6%
0
1
Predicted Class
70.6%
29.4%
81.6%
18.4%
TPR
FNR

Fig – 21

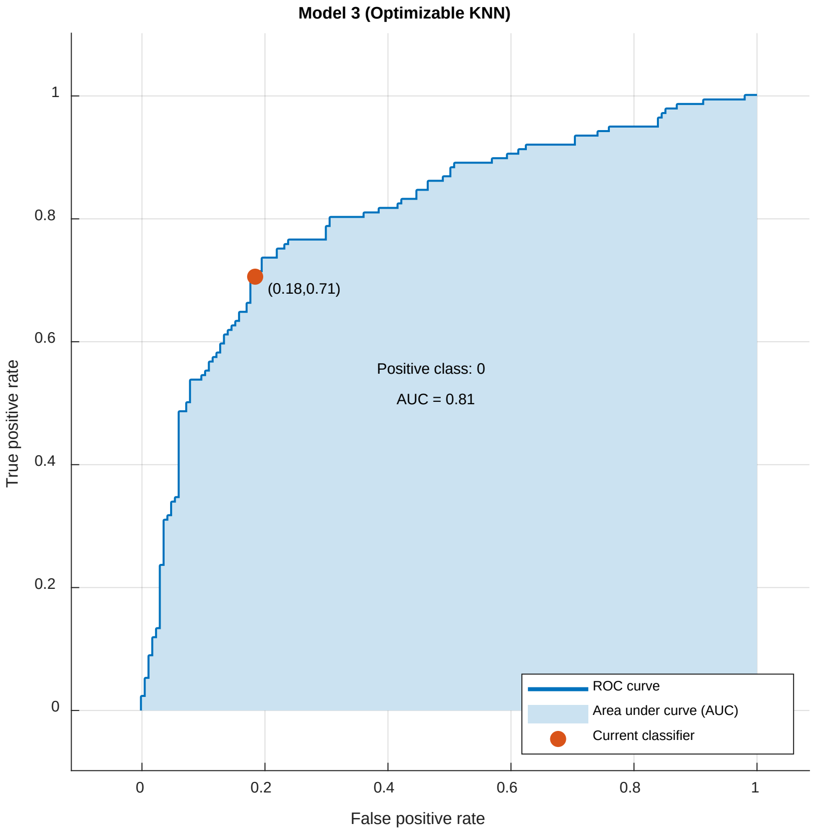

Model 3 (Optimizable KNN)
True positive rate
1
0.8
0.6
0.4
0.2
0
(0.18,0.71)
Positive class: 0
AUC = 0.81
ROC curve
Area under curve (AUC)
Current classifier
0
0.2
0.4
0.6
0.8
1
False positive rate

Fig – 22

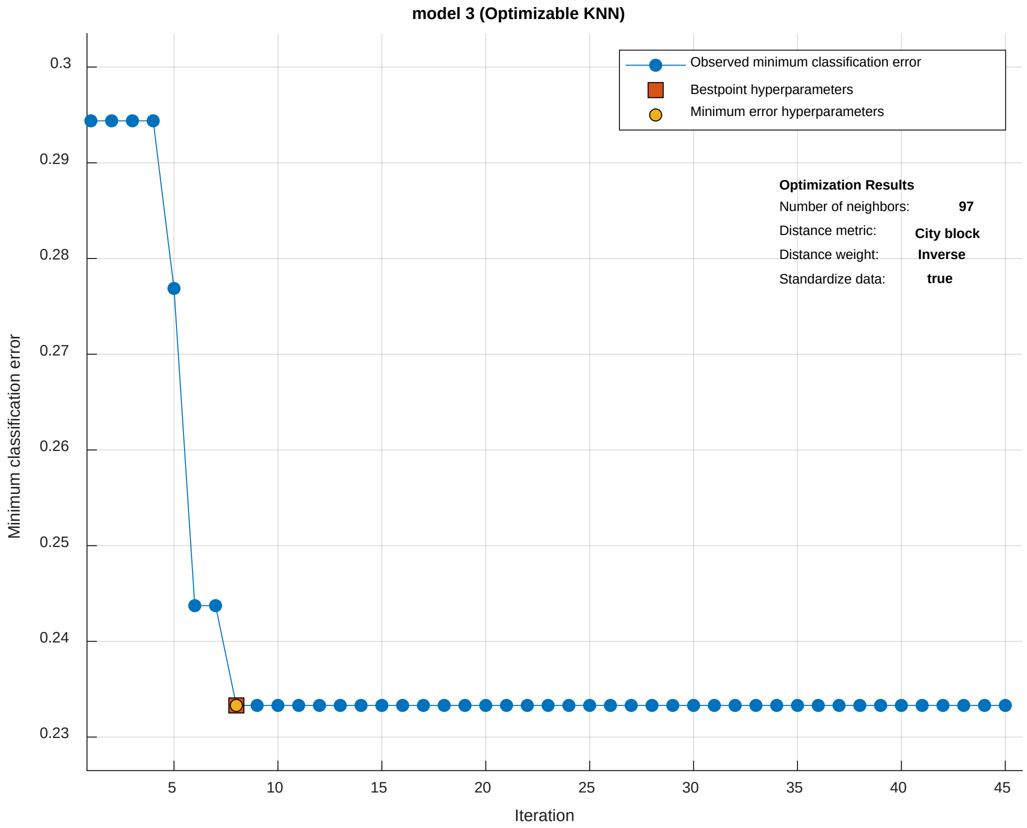


Fig - 23

Model Details	**Results**	**Optimiser Options**	**Feature Selection and PCA**
Type: Optimised KNN ***Surrogate Decision Splits:*** Off	***Accuracy:*** 76.6% ***Total misclassification cost:*** 70 ***Prediction Speed:*** ~18000 obs/sec ***Training Time:*** 20.061 sec	***Hyperparameter option:*** Enabled (Non-Linear with 97 neighbours and City block Matrix with 45 iterations)	***No of features being used:*** ***4***/13 ***PCA:*** Disabled

Fig - 24

Model 3 Performance Evaluation

- Model 3 percentage accuracy (76.6%) is the best in comparison to model 2 (75.3%) and model 1 (75.6%)
- The prediction speed of model 3 is the lowest with 18000 obs/sec. Model 2 is the fastest with 46000 obs/sec and model 1 is the second fastest with a speed of 33000 obs/sec.
- Training time for model 3 is the lowest at 20.061 sec. Model 2 took the longest time of 134.74 sec for training, while the model 1 is the second quickest with 22.344 sec.
- The model 3 has the same Area under the curve (AUC) as the model 2 with the value of 0.81 (see Fig-22). The model 1 has a lower AUC value of 0.73 (see Fig-12).

9 Comparisons between three Machine Learning Models

Below is the quick summary of statistics to compare three ML models.

	Percentage Accuracy	**Prediction Speed**	**Training Time**	**AUC Probability**	**False Negative Rates FNR(0)**	**False Negative Rates FNR(1)**
	%	**obs/sec**	**sec**	**(0-1)**	**%**	**%**
Model 1 **(Optimised Tree)**	75.6	33000	22.344	0.73	**25.0**	23.9
Model 2 **(Optimised SVM)**	75.3	**46000**	134.74	**0.81**	27.9	22.1
Model 3 **(Optimised KNN)**	**76.6**	18000	**20.061**	**0.81**	29.4	**18.4**

Performance Matrix (Fig – 25)

10 Recommendations and Conclusions

By looking at the performance matrix in Fig – 25, it is very clear that the model 3 (optimised KNN) is the best ML model because it has scored well in most categories.

It has the highest accuracy of 76.6 %. It also takes the least time of 20.061 sec to get trained. The Area under the curve (AUC) and ROC curves are very important measures for performance for classification problems as they are used to distinguish between various output of the resulting class [7]. In our case, both model 2 and model 3 has the AUC value of 0.81, which means that there is a probability of 0.81 or 81% that both models will be able to predict if a given patient will get a heart disease (1) or not(0).

However, the most parameter in the domain of health care is the False Negative Rate (FNR) value which shows the statistical classification of false negative diagnosis [8]. In our use case, it means that percentage of patients who had a heart condition, but the given ML model fails to diagnose it correctly. Based on this argument, we believe that the model 3(KNN) is the best choice as it has the lowest FNR(1) value of 18.4%.

Based on above discussion, we would recommend using the model 3 which is an optimised KNN classification Machine Learning model to predict if a person suffers with a heart condition based on the given data set.

References

[1] https://archive.ics.uci.edu/ml/datasets/Heart+Disease/

[2] https://uk.mathworks.com/help/stats/machine-learning-in-matlab.html

[3] A. Burkov, “The Hundred-page Machine Learning Book”, Self-published, 2019.

[4] Alpaydin E. (2014) Introduction to Machine Learning, 3rd ed, MIT press

[5] https://www.data-to-viz.com/graph/parallel.html

[6] https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f

[7] https://en.wikipedia.org/wiki/False_positives_and_false_negatives

[8] https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5