A mathematical study of machine learning and deep learning algorithms and its effectiveness in prediction and classification of data

Shashank H S
Department Of Biotechnology
Indian Institute of Technology, Madras
Chennai, India
shashank.hiremath@smail.iitm.ac.in

***Abstract*—This document aims to study the mathematical properties of various machine learning and deep learning algorithms and demonstrate its effectiveness in using the features in the given datasets to perform classification and prediction.**

This paper will serve as a part of the requirements to be fulfilled for receiving credits in the course **EE4708: Data Analytics Laboratory**.

I. Linear Regression

Introduction

This section seeks to explore one of the fundamental modelling tools used for data analysis - **Linear Regression**. Once we have acquired data with multiple variables, one important task is to understand how the variables are related. Regression is a statistical method used to determine the strength of the relationship between one or more independent variables and a dependent variable. Linear regression, one of the well-known machine learning techniques, makes the assumption that the variables have a linear relationship and accordingly gauges the relationship between the variables. Despite its simplicity, linear regression has proven itself a powerful tool for analyzing data and revealing trends in the data. Its simplicity is evident in its representation:

$$y = \beta_0 + \beta_1 X$$

In the above equation, y denotes the independent variable and X denotes the dependent variables (s) where multiple dependent variables can be succinctly expressed as a vector. The parameters of this model are β_0 and β_1 which represent the intercept/constant and the slope/scaling factor(s) associated with the dependent variable(s). For a given data point X and a given set of parameter values β_0 and β_1, we calculate the estimated value of y. The objective of linear regression is to identify the parameter values for which the sum total of errors for all data points is minimum. These values provide us a quantitative description of the relationship between the variables.
To demonstrate the effectiveness of linear regression, we use standard socioeconomic variables like income and health insurance access in a population to identify putative correlations with the incidence of cancer and mortality caused by cancer in the same population. The pertinent data has been collected from [1]. Before performing data analysis using linear regression, we clean and visualise the data.

Linear Regression

It was stated in the introduction section that despite its simplicity, linear regression has proven itself a powerful tool in data analysis. Inside its simplicity lies certain assumptions that must be mentioned before we delve into the mathematical details of linear regression.

A. Assumptions

- The first assumption is the independence of observations which means that there is no relationship between different independent variables. To be certain that this assumption is appropriate for the model we seek, one must look at the data collection process and see if it is collected without bias. Correlations between independent variables will imply redundancy and this could result in overfitting of the model
- The second assumption, which has been mentioned in the introduction, is the linear relationship between the independent and dependent variables. The assumed linearity is parametric linearity and not variable linearity. Hence, equations such as $y = \beta_0 + \beta_1 x^2 + \beta_2 x^3$ are allowed because y is linearly dependent on the parameters β_0, β_1 and β_2, and variables x^2 and x^3.
- The third assumption is that the error terms have constant variance which is also called homoskedasticity. If error terms have varying variance called as heteroskedasticity, the model will be accurate in some parts of the dataset.

B. Errors

The objective of linear regression is to identify the parameter values for which the sum total of errors for all data points is minimum. There are different ways to measure the error:

1) Mean Absolute Error (MAE) = $\frac{1}{n}\sum_{i=1}^{n}|y_{pred} - y_{actual}|$
2) Mean Squared Error (MSE) = $\frac{1}{n}\sum_{i=1}^{n}(y_{pred} - y_{actual})^2$
3) Root Mean Squared Error (RMSE) = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{pred} - y_{actual})^2}$
4) Mean Percent Error (MPE) = $\frac{100\%}{n}\sum_{i=1}^{n}(\frac{y_{actual}-y_{pred}}{y_{actual}})$
5) Mean Average Percent Error (MAPE) = $\frac{100\%}{n}\sum_{i=1}^{n}(\frac{y_{actual}-y_{pred}}{y_{actual}})^2$

MPE is used to check if the model's performance is symmetric because it does not take absolute values or squares of errors.

Another metric used to evaluate the model's performance is coefficient of determination R^2 which is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_{actual} - y_{pred})^2}{\sum_{i=1}^{n}(y_{actual} - y_{mean})^2}$$

The values of R^2 lie between -1 and 1. If the R^2 of the linear regression model is closer to 1, then it is able to explain a higher proportion of the variance in y and consequently performs better in estimating y.

C. Model Parameters

There are two broad types of linear regression: simple and multiple regression.

- In simple regression, there is one input variable X and two parameters β_0 and β_1 that decide the estimated value of y.
- In multiple regression, there are multiple input variables X_i, each of which have a β_i scaling factor. The scaling factors and the constant β_0 along with the X_is determine the estimated value of y.

The coefficients/scaling factors represent the slope of the line for a particular variable in the hyperplane. A positive value for β_i indicates that X_i is positively correlated with y. Similarly, a negative value for β_i indicates that X_i is negatively correlated with y. β_0 is the prediction of the mean value of y in the state space for $X_i = 0$.

Does socioeconomic status determine cancer risk?

In this section, we see the correlations and relationships revealed by the linear regression models between socioeconomic status and cancer incidence, mortality. We want to see whether poor sections of the population and those without health insurance are likely to get cancer. Two important socioeconomic variables used are:

1) Incidence rate: defined as the number of cancer cases recorded in a year per 100,000 people at risk. The population is taken to be the denominator assuming that cancer is not related to age.
2) Mortality rate: defined as the number of deaths in a population in a given year per 100,000 people.
3) Median income in each county
4) Population with and without insurance

D. Preparing the data for analysis

Data collection is rarely perfect and since this data has been collected from thousands of counties across the United States, some pre-processing has to be done before visualising the data. In the given dataset, there are a lot of missing data entries owing to confidentiality measures. The missing values were imputed using the mean of the columns. The following **13 new features** were created using the given data:

- Total population = sum of all with insurance and all without insurance.
- Population of Male and Female individuals.
- Poverty percentage = all those in poverty divided by the total population.
- Poverty percentage for Male and Female individuals.
- Insurance percentage = all those having insurance divided by the total population.
- Insurance percentage for Male and Female individuals.
- Incidence rate width which was calculated using the difference between the upper 95% and lower 95% confidence interval.
- Incidence rate trend width which was calculated using the difference between the upper 95% and lower 95% confidence interval of the recent 5 year trend.
- Mortality rate width which was calculated using the difference between the upper 95% and lower 95% confidence interval.
- Mortality rate trend width which was calculated using the difference between the upper 95% and lower 95% confidence interval of the recent 5 year trend.

E. Visualisation

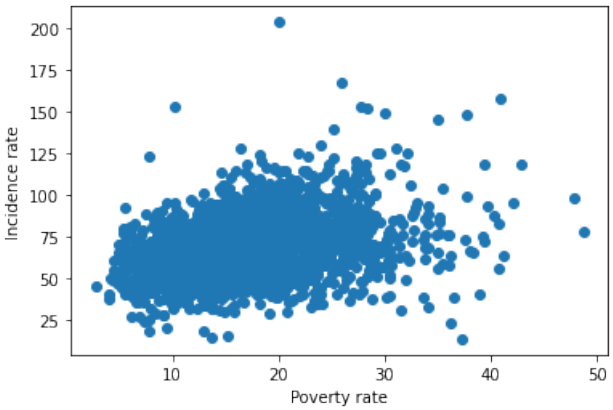


Figure 1: Plot of cancer incidence rate against poverty rate.

The picture itself shows a positive correlation between these two variables.

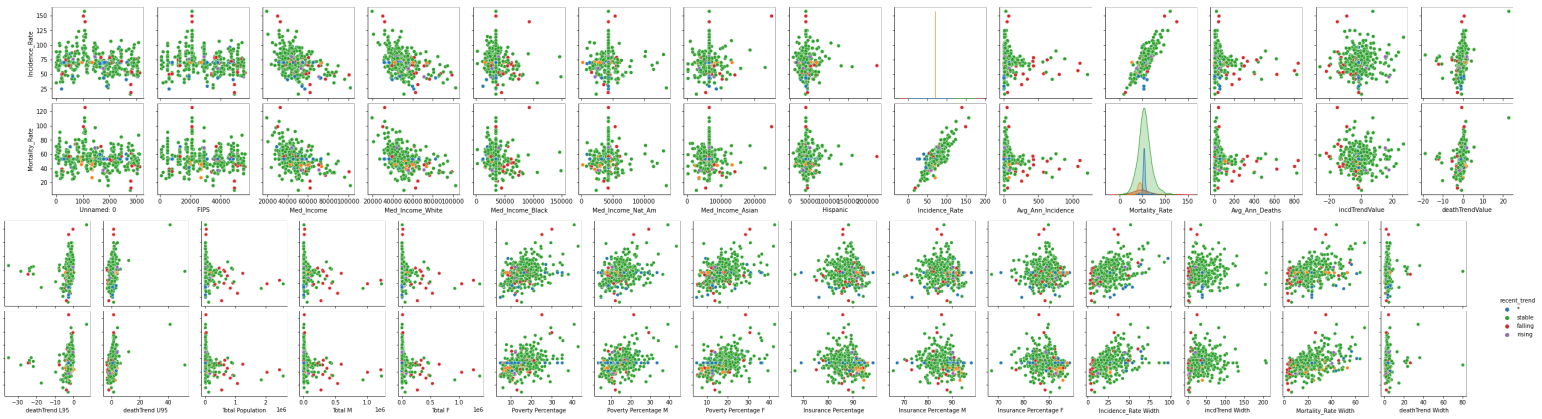


Figure: pair plot of the features plotted against incidence rate and mortality rate with the recent trend used to color the data points.

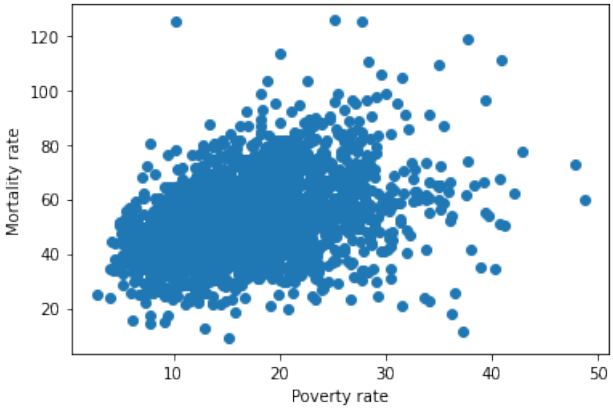


Figure 2: Plot of cancer mortality rate against poverty rate.

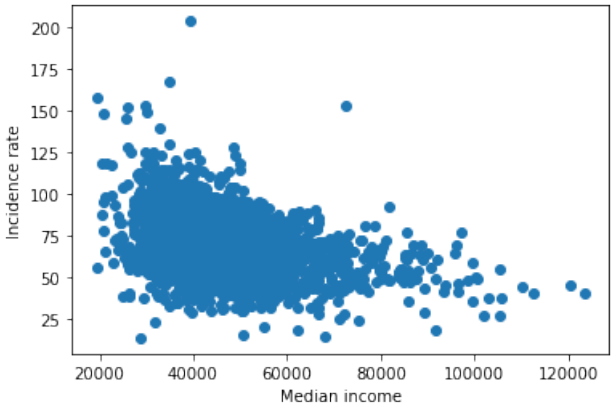


Figure 3: Plot of cancer incidence rate against median income.

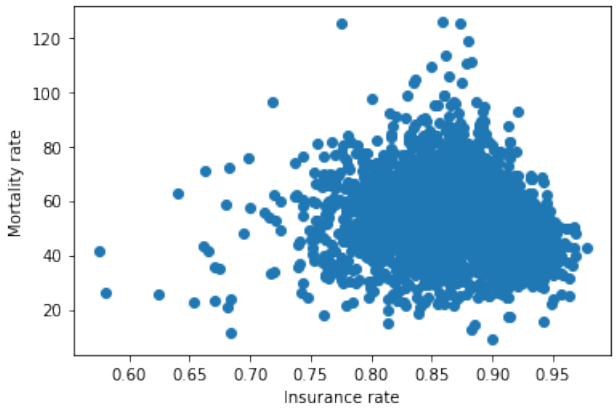


Figure 4: Plot of health insurance rate against cancer mortality rate.

Second, plot the cancer mortality rate against poverty rate. Again, there is a positive correlation between these two variables. The plot of cancer incidence rate against median income shows clear negative correlation and so does the plot between cancer mortality rate and median income (not shown here). The plots of cancer mortality rate against health insurance rate (Fig. 4) and cancer incidence rate against health insurance rate (not shown) do not show a clear correlation between these variables. The visual data shows that the poor people are likely to get cancer or die because of it. Now we have to show it with statistical proof using a linear regression model.

F. The linear regression model

Simple linear regression models were built using the various features. From the quantitative results obtained and the previous visualisation plots, we can see that the distribution is not really homoscedastic. As discussed, this can really hinder the performance of a linear regression model. In each of the below plots, the X - axis shows the dependant variable and the Y - axis shows the independent variable. Each plot shows the regression model built using the variables in the plots. The R^2 score is given in the title of the plot.

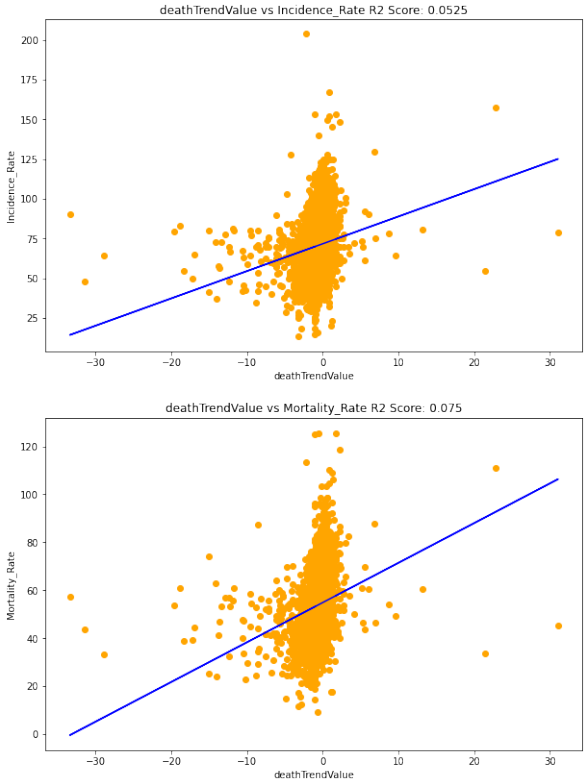


Figure 6: Linear Regression using death trend value against incidence rate and mortality rate.

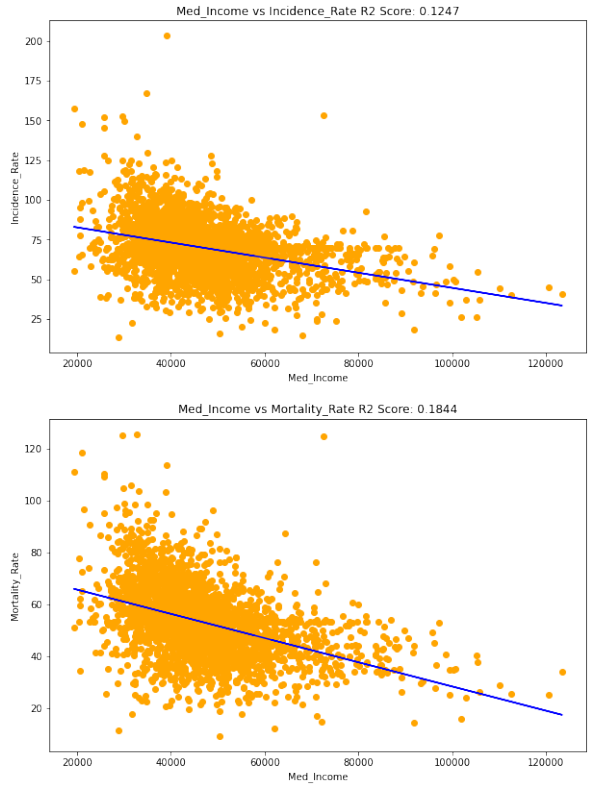


Figure 5: Linear Regression using median income against incidence rate and mortality rate.

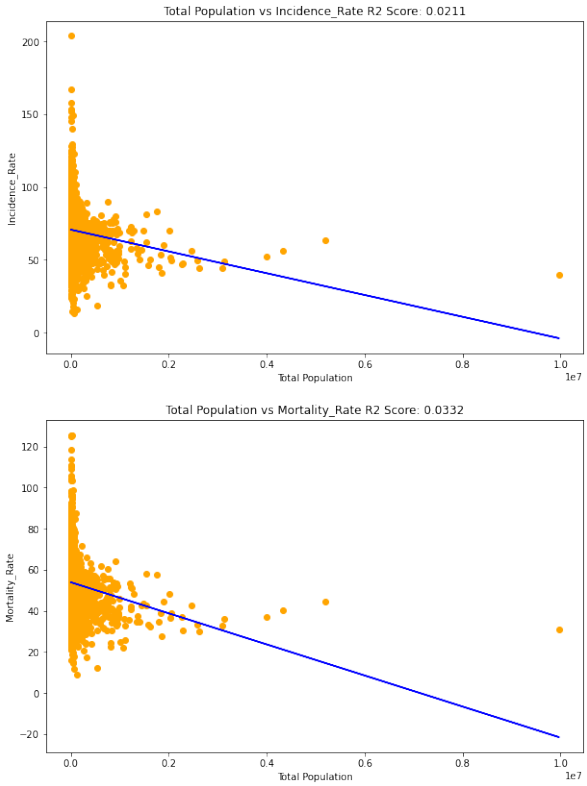
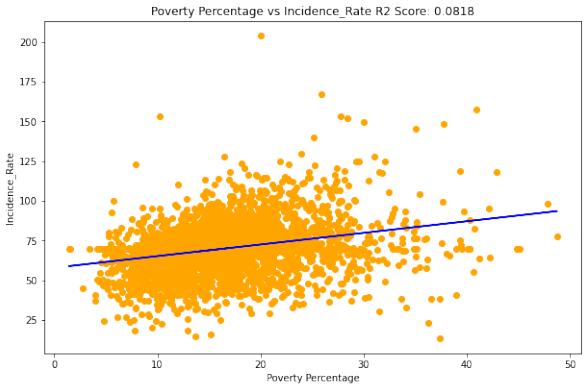


Figure 7: Linear Regression using total population against incidence rate and mortality rate.



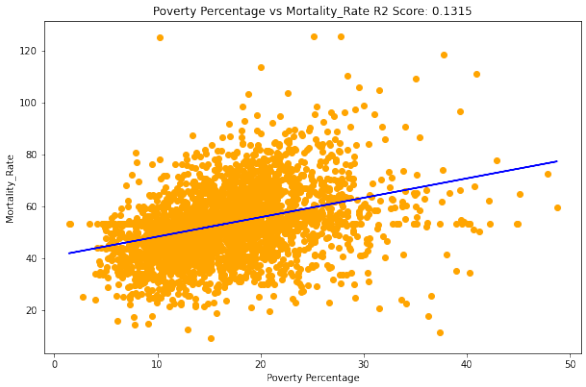


Figure 8: Linear Regression using poverty percentage against incidence rate and mortality rate.

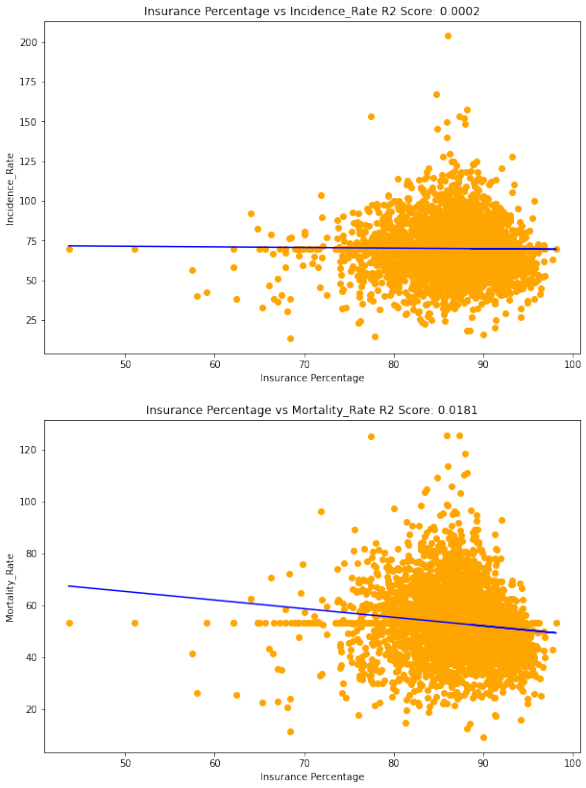
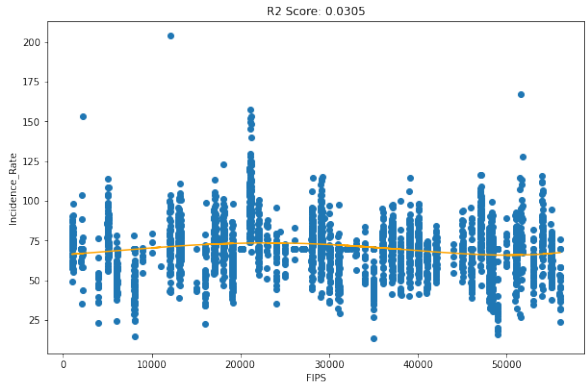


Figure 9: Linear Regression using insurance percentage against incidence rate and mortality rate.



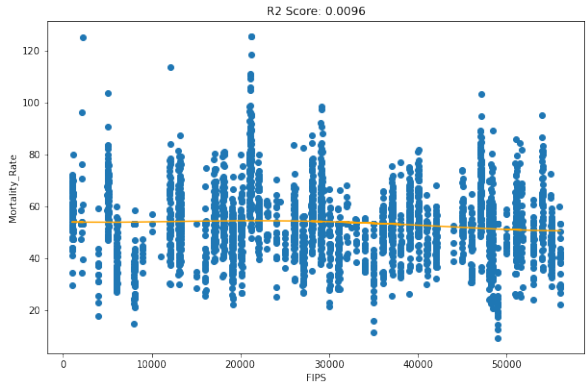


Figure 10: Linear Regression using FIPS against incidence rate and mortality rate.

When the following features were used to train a linear regression model against incidence rate and mortality rate, here are the coefficient values and lowest RMSE scores:

```
__________________
Incidence_Rate
__________________
Med_Income  Coeff:  -2.873657256646401
deathTrendValue  Coeff:  3.6519602983464825
Total Population  Coeff:  0.17793182531792018
Poverty Percentage  Coeff:  3.341590444051891
Insurance Percentage  Coeff:  1.6414448048437214
FIPS  Coeff:  -2.7677008878098004
R2 Score:  0.1761  RMSE:  14.1111

__________________
Mortality_Rate
__________________
Med_Income  Coeff:  -2.353669413217932
deathTrendValue  Coeff:  4.317005393308705
Total Population  Coeff:  -0.16294435081805475
Poverty Percentage  Coeff:  2.5184660130271515
Insurance Percentage  Coeff:  -0.3477807466728721
FIPS  Coeff:  -0.379061420495279
R2 Score:  0.277  RMSE:  10.953
```

Figure 11: Linear Regression feature coefficients and RMSE.

Conclusion

- While the model's performance was successfully demonstrated, both visually and numerically, that there exists a negative correlation between incidence rate and median income, and between mortality rate and median income. The results of the linear regression model are not satisfactory so the relationships in question are not, strictly speaking, linear. This can be attributed to flaws in the data collection or the simplicity of the model.

- After seeing the plots, I suggest a logistic regression model will serve the data better. The heteroscedastic nature of the incidence vs median income and mortality vs median income plots may also suggest the need to take into account additional features such as race or sex, but race specific incidence and death rates must be obtained first.
- For the purposes of fundraising for the NGO, it can be argued that visual proof is more compelling than statistical evidence. Hence, while we can discuss the nature of the relationship between income and rates of incidence or mortality, it is clear that the mortality and incidence rates decrease with an increase in income. The figures make a compelling case for the health authorities to prioritise improving the health standard of the poorer section of the American society.

II. Logistic Regression

Introduction

This section seeks to explore one of the fundamental modelling tools used for data analysis and classification - **Logistic Regression**. Once we have acquired data with multiple variables , one important task is to understand how the variables are related. Regression is a statistical method used to determine the strength of the relationship between one or more independent variables and a dependent variable. Logistic regression, one of the well-known machine learning techniques, makes the assumption that the independent variables are linearly related to the log of odds. Despite its simplicity, logistic regression has proven itself a powerful tool for analyzing data and making predictions of the value of the dependent variable. Logistic regression is usually used for classification when the dependent variable is binary. Its simplicity is evident in its representation:

$$log(\frac{p}{1-p}) = \beta_0 + \beta_1 X$$

where p = $P(y = 1|X)$. In the above equation, y denotes the dependent variable and X denotes the independent variables(s) where multiple independent variables can be succinctly expressed as a vector. The parameters of this model are β_0 and β_1 which represent the intercept/constant and the slope/scaling factor(s) associated with the independent variable(s). For a given data point X and a given set of parameter values β_0 and β_1, we calculate the estimated probability that y = 1. The objective of logistic regression is to identify the parameter values for which the sum total of errors for all data points is minimum. These coefficient values provide us a quantitative description of the relationship between the independent variables and the binary dependent variable.

To demonstrate the effectiveness of logistic regression, we use socioeconomic variables like age, gender and cabin class among Titanic passengers to identify putative correlations with the survival of those passengers. The pertinent data has been collected from [2]. Before performing data analysis and classification using logistic regression, we clean and visualise the data.

Logistic Regression

It was stated in the introduction section that despite its simplicity, logistic regression has proven itself a powerful tool in classification and data analysis. Inside its simplicity lies certain assumptions that must be mentioned before we delve into the mathematical details of linear regression.

A. Assumptions

- The first assumption, which has been mentioned in the introduction, is the linear relationship between the independent variables and log of odds ($log(\frac{P(y=1|X)}{1-P(y=1|X)})$). The assumed linearity is parametric linearity and not variable linearity. Hence, equations such as $log(\frac{p}{1-p}) = \beta_0 + \beta_1 x^2 + \beta_2 x^3$ are allowed because $log(\frac{p}{1-p})$ is linearly dependent on the parameters β_0, β_1 and β_2, and variables x^2 and x^3.
- The second assumption is that the dependent variable is binary. In our case, the passenger has either survived (y = 1) or not (y = 0).
- The third assumption is the independence of observations which means that there is no relationship between different observations. In our data, each data point is a unique passenger.
- The fourth assumption is that there is minimal or no multicollinearity among the independent variables. Correlations between independent variables will imply redundancy and this could result in overfitting of the model.
- Another assumption is that the data set is large. Logistic regression usually requires a large sample size to predict properly.

B. Errors

The objective of logistic regression is to identify the parameter values for which the sum total of errors for all data points is minimum. There are different ways to measure the error:

1) Mean Absolute Error (MAE) = $\frac{1}{n}\sum_{i=1}^{n}|y_{pred} - y_{actual}|$
2) Mean Squared Error (MSE) = $\frac{1}{n}\sum_{i=1}^{n}(y_{pred} - y_{actual})^2$
3) Root Mean Squared Error (RMSE) = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{pred} - y_{actual})^2}$
4) Mean Percent Error (MPE) = $\frac{100\%}{n}\sum_{i=1}^{n}(\frac{y_{actual} - y_{pred}}{y_{actual}})$
5) Mean Average Percent Error (MAPE) = $\frac{100\%}{n}\sum_{i=1}^{n}(\frac{y_{actual} - y_{pred}}{y_{actual}})^2$

MPE is used to check if the model's performance is symmetric because it does not take absolute values or squares of errors.

Another metric used to evaluate the model's performance is coefficient of determination R^2 which is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_{actual} - y_{pred})^2}{\sum_{i=1}^{n}(y_{actual} - y_{mean})^2}$$

The values of R^2 lie between -1 and 1. If the R^2 of the logistic regression model is closer to 1, then it is able to explain a higher proportion of the variance in y and consequently performs better in predicting y.

C. Model Parameters

Whether there are multiple input variables X_i or one independent variable X, the independent variable(s) have a β_i scaling factor. The scaling factors and the constant β_0 along with the X_is determine the estimated value of log of odds for y = 1. Instead of comparing y and X, logistic regression models $P(y = 1|X)$ vs X.

The regression coefficients describe the size and direction of the relationship between a predictor (independent variable) and the response variable y. The coefficients in a logistic regression are log odds ratios. Negative values mean that the odds ratio is smaller than 1 and it is likely that y = 0, and vice versa.

$$log(\frac{p}{1-p}) = \beta_0 + \beta_1 X$$

The above mathematical expression of logistic regression can also be expressed as:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}} = sigmoid(\beta_0 + \beta_1 X)$$

This expression for $P(y = 1|X)$ is a sigmoid function and has an S-shaped curve.

Does socioeconomic status determine probability of survival aboard the Titanic?

In this section, we see the relationships revealed by the logistic regression model between socioeconomic status and survival. We want to see whether poor people and a certain gender among the passengers were likely to survive. The important socioeconomic variables used are:

1) Age in years of the passenger
2) Gender of the passenger
3) Family size of the passenger = 1 + siblings/spouse count + parents/children count.
4) Port of embarkation which can be Cherbourg, Queenstown or Southampton.
5) Cabin Class

D. Preparing the data for analysis and visualisation

Data collection is rarely perfect and since this data was collected a century ago, some pre-processing has to be done before visualising the data. In the given dataset, there are a lot of missing data entries owing to difficulty in obtaining records. Most of these missing entries occur in the fields of Cabin and age. Hence, using the columns that provide additional information,data imputation was performed.

E. Visualisation

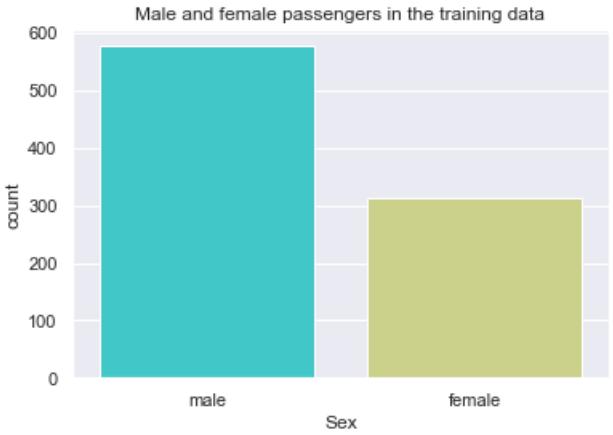


Figure 12: Count of male and female passengers. There were 577 male passengers and 314 female passengers.

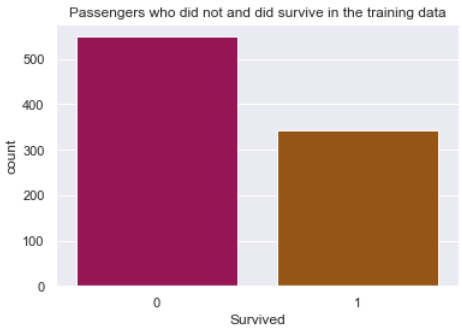


Figure 13: Count of passengers who survived and those who didn't. There were 342 passengers who survived and 549 passengers who did not survive.

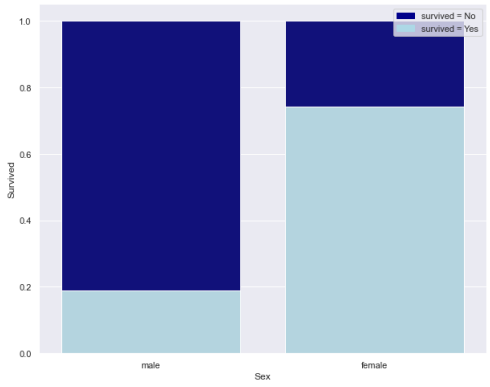


Figure 14: Proportion of male and female passengers who survived and those who didn't. Of the 577 male passengers, less than 20 % survived. Of the 314 female passengers, nearly 80% of them survived. Clearly, gender plays a role in probability of survival.

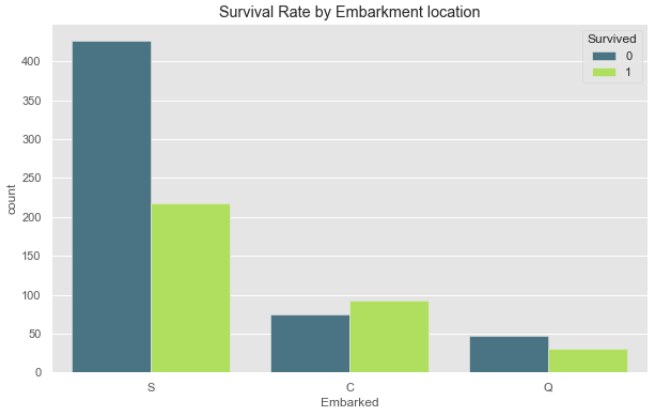


Figure 15: Survival rate by location of embarking.

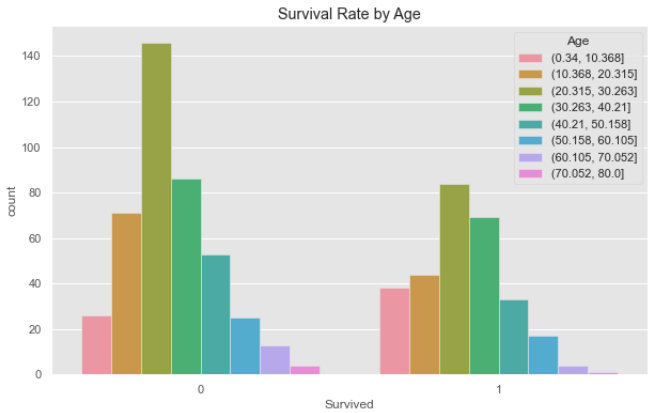


Figure 16: Survival rate by age.

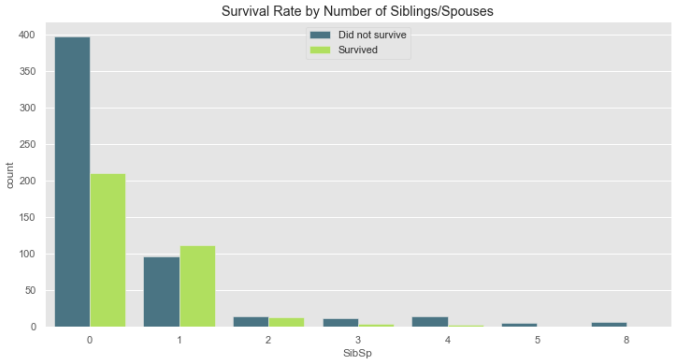


Figure 17: Survival rate by number of siblings/spouses.

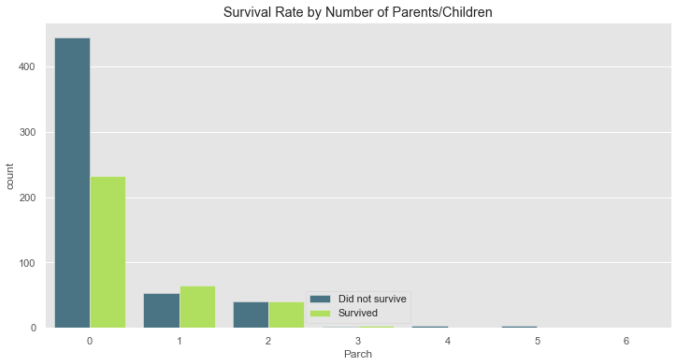


Figure 18: Survival rate by number of parents/children.

Figure 19: Number of missing datapoints per feature. One-hot encoding was used for Cabin and so missing entries were marked as 'U'. The missing values of age were imputed using the median value of the passenger according to their title in their name. The 2 datapoints with missing port of embarkation were discarded.

One-hot encoding was used for the categorical data column of name based on the title present in the name. Even gender was encoded using one-hot encoding. After the performing the necessary changes in encoding mentioned previously, the categorical columns of PassengerId, Name, Sex, Ticket,

Cabin, and Port of Embarkation were discarded. A new column named family size was created where each row = number of siblings/spouses + number of parents/children + 1.

F. The logistic regression model

The previous logistic regression model was trained using 85% of the data. 15% of the data was used for testing / cross-validation. Accuracy on 85% of training data was 84.37086092715231%. Accuracy on 15% of training data (test / cross-validation set) = 84.32835820895522%.

The new logistic regression model was trained on 84% of the data and it had the following changes:

- The penalty was elasticnet i.e. both L1 and L2 penalty terms were present as opposed to only L2.
- The SAGA solver, which is a variant of SAG that also supports the non-smooth penalty L1 option (i.e. L1 Regularization), was used for training.
- The L1 ratio which accounts for the ratio of penalty between L1 and L2 was set to 0.4.

Accuracy on 84% of training data was 84.45%. Accuracy on 16% of training data (test / cross-validation set) = 84.615%. The new model clearly has better accuracy.

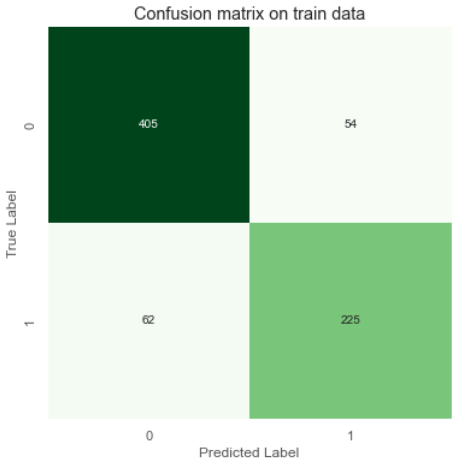


Figure 20: Confusion matrix of the new logistic regression model on the training data.

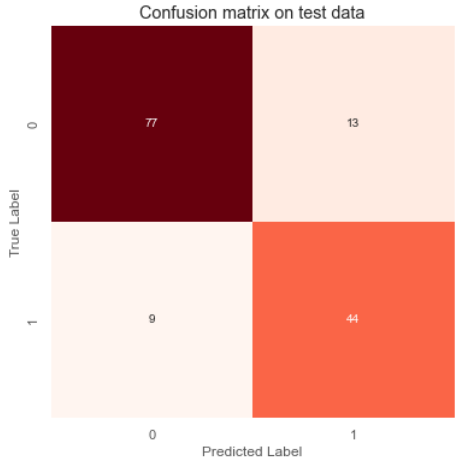


Figure 21: Confusion matrix of the new logistic regression model on the testing / cross-validation data.

The weights of the model are:

```
Pclass     -1.5568495275534602
Age     -2.0010625889452105
SibSp     -1.4718584534197172
Parch     -0.5218307724032638
Fare     0.3636360943539882
Familysize     -1.7572282274933808
Embarked_C     0.2851373648719263
Embarked_Q     0.0
Embarked_S     -0.25127369536320293
Cabin_A     -0.17265976713070955
Cabin_B     0.0
Cabin_C     -0.0451247043606101
Cabin_D     0.7088177433250322
Cabin_E     0.7909680437188372
Cabin_F     0.0
Cabin_G     0.0
Cabin_T     0.0
Cabin_U     -0.4766764727555683
Title_Capt     0.0
Title_Col     0.008115634031067069
Title_Don     0.0
Title_Dr     0.0
Title_Jonkheer     -0.10713665998856309
Title_Lady     0.0
Title_Major     0.0
Title_Master     1.5686337618043955
Title_Miss     0.0
Title_Mlle     0.0
Title_Mme     0.0
Title_Mr     -1.0108311551154257
Title_Mrs     0.8569662073101463
Title_Ms     0.0
Title_Rev     -0.6290171481048461
Title_Sir     0.0567442587827512
Title_the Countess     0.0
Sex_female     0.8457698598482243
Sex_male     -0.84080834089021
```

Figure 22: Logistic regression model weights of the parameters.

Clearly, if a passenger is female, she is more likely to survive and the men are, unfortunately, likely to die aboard the HMS Titanic.

Those who embarked the ship at Southampton were more likely to die. The rest of the parameters quantify the respective likelihood of survival.

CONCLUSION

- The logistic regression model's performance was successfully demonstrated, both visually and numerically, that there exists a correlation between gender and survival, and between cabin class and survival.
- The addition of the L1 regularization term and the use of the SAGA solver resulted in a small increase in accuracy. After an extensive exploration of parameter values, it is safe to assume that this is the best possible accuracy on the given dataset.

III. NAIVE BAYES CLASSIFIER

INTRODUCTION

This section seeks to explore one of the important modelling tools used for classification and prediction - **Naive Bayes classifier**. Once we have acquired data with multiple features, one important task is to understand how the variables are related. Classification is a statistical method in which the relationship between one or more independent variables and a dependent variable is used to identify the mathematical expression to assign a given data point to a class. The Naive Bayes classifier, one of the well-known machine learning techniques, makes the assumption that the input feature variables are conditionally independent of each other given the target variable.

Despite its simplicity, the Naive Bayes classifier has proven itself a powerful tool for making predictions and classifying a given data point into a particular class. Naive Bayes classifier can be used for binary classification or multi-class classification. Its simplicity is evident in the representation of its inference step:

$$\begin{aligned} p(C_k|x_1, x_2) &\propto p(x_1, x_2|C_k)p(C_k) \\ &\propto p(x_1|C_k)p(x_2|C_k)p(C_k) \\ &\propto \frac{p(C_k|x_1)p(C_k|x_2)}{p(C_k)} \end{aligned}$$

In the above equation, C_k refers to the kth class. x_1 and x_2 are the input features or conditionally independent variables.

This classifier is a generative model, which means that it models the joint probability distribution of the input features and the target variable after the assumption of conditional independence given the target variable. The first step is the inference step, which determines p(X, t) where X is the vector of input features and t is the target variable. The next step is the decision step where the model has to predict the value of the target variable for a given data point by choosing the class with the highest probability or by using some optimal decision criteria.

To demonstrate the effectiveness of the Naive Bayes classifier, we use socioeconomic variables like age, gender and education among adults to classify the annual income of adults as less than or greater than $\$50000$. The pertinent data has been collected by Ronny Kohavi and Barry Becker [3]. Before performing data analysis and classification using the Naive Bayes Classifier, we clean and visualise the data.

NAIVE BAYES CLASSIFIER

It was stated in the introduction section that despite its simplicity, the Naive Bayes classifier has proven itself a powerful tool in classification and data analysis.

Inside its simplicity or naivety lies the assumption that given the target variable t, the input features x_i are conditionally independent of each other. This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, as we shall soon see, this model performs surprisingly well on data where this assumption does not hold.

A. Probabilities

The representation for Naive Bayes is probabilities. A list of probabilities are stored to file for a trained Naive Bayes model. This list contains:

- Class Probabilities: The probabilities of each class in the training dataset. In our previously mentioned nomenclature, this refers to p(C_k).
- Conditional Probabilities: The conditional probabilities of each input value given each class value. This refers to p($x_i|C_k$) for different x_i and C_k.

B. Decision Theory and learning a Naive Bayes model

For any given classifier, there are two key steps in the learning process - Inference and Decision.

- During the inference step, a classifier models p(X, t) or p(t — X) where X and t are the features and target variable respectively. Since Naive Bayes classifier is a generative model, it models the joint probability distribution of the X and t.
- The decision step is when the model has to predict the value of the target variable for a given data point by using some optimal decision criteria. The criteria can be a loss matrix or a standard maximum a posterior (MAP) method where the class with the highest predicted probability is chosen.

To learn a Naive Bayes model, we need to calculate the class probabilities and the conditional probabilities.

- Class probabilities: The class probabilities are simply the frequency of data points that belong to each class divided by the total number of data points.
- Conditional probabilities: The conditional probabilities are the frequencies of each attribute value for a given class value divided by the frequency of data points with that class value.

Once we have performed the inference step and calculated the probabilities based on an underlying model distribution assumption, we make decisions for new data points.

$$p(C_k|x_1, x_2) \propto \frac{p(C_k|x_1)p(C_k|x_2)}{p(C_k)}$$

Using the above equation, we calculate the probabilities of all classes given a new data point. Then, we predict that the data point falls into class k based on decision criteria.

Usually, one would use the Maximum A Posteriori (MAP) probability and predict that the data point X falls into the class with the highest probability. Sometimes, different misclassifications can have different costs. Hence, a loss matrix L can also be used such that the predicted class is chosen as

$$h(x) = argmin_j \sum_t L_{tj}\, p(t|X)$$

h(x) is the classifier and L_{tj} refers to the element in the loss matrix L where t is the true class t and j is the decision.

For a Naive Bayes classifier, the test error will never be zero because the prediction is probabilistic. However, assuming that we have complete knowledge of the posterior distribution p($t|X$), choosing the MAP estimate results in Naive Bayes classifier being the most optimal classifier among all the classifiers.

Model performance can be measured using the standard Mean Squared Error (MSE) = $\frac{1}{n}\sum_{i=1}^{n}(y_{pred} - y_{actual})^2$

C. Gaussian Naive Bayes Classifier

When we calculate the conditional probabilities, we must be able to find the values for new data points. Hence, we must have a standard probability density, characterised by probability density parameters.

For the Naive Bayes classifier, we make the assumption that the values associated with each class are distributed according to the Gaussian or Normal distribution. For example, suppose the training data contains a continuous attribute X. We first segment the data by the class, and then compute the mean and variance of x in each class. Suppose we have some observation value xi . Then, the probability distribution of xi given a class can be computed by the following equation

$$p\left(x_i \mid y_j\right) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_i-\mu_j)^2}{2\sigma_j^2}}$$

The conditional independence assumption in the multivariate input features case is simply expressed in the Gaussian expression with an identity covariance matrix.

CAN SOCIOECONOMIC FACTORS BE USED TO ACCURATELY DETERMINE AN ADULT'S INCOME RANGE?

In this section, we see the predictions and relationships revealed by the Naive Bayes model between socioeconomic status and income. We want to see whether we can accurately classify an adult's income as above or below $50000 and we want to see the role of socioeconomic variables such as education, gender in determining income. The list of socioeconomic variables in the input data are:

1) Age in years of the adult
2) Work class (private, government, etc.)
3) Gender of the adult
4) Level of education
5) Number of years of education
6) Marital status
7) Occupation
8) Relationship
9) Race
10) Sex
11) Capital gain
12) Capital loss
13) Hours per week of work
14) Native country
15) A continuous variable fnlwgt (final weight)

D. Preparing the data for analysis and visualisation

Data collection is rarely perfect and since this data was collected by a census bureau 27 years ago, some pre-processing has to be done before visualising the data. In the given dataset, there are a lot of missing data entries owing to difficulty in obtaining records. These missing entries occur in the fields of workclass, occupation and native country. Hence, using the columns that have entries, data imputation was performed.

E. Visualisation

Simple bar plots and histograms were made for several variables to understand the frequency of certain attributes within that variable.

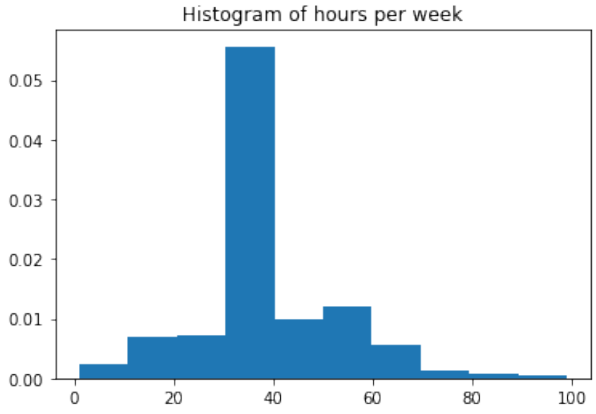


Figure 25: A histogram of hours per week at work. It is shaped like a gaussian curve with very low variance and mean at 37 hours per week.

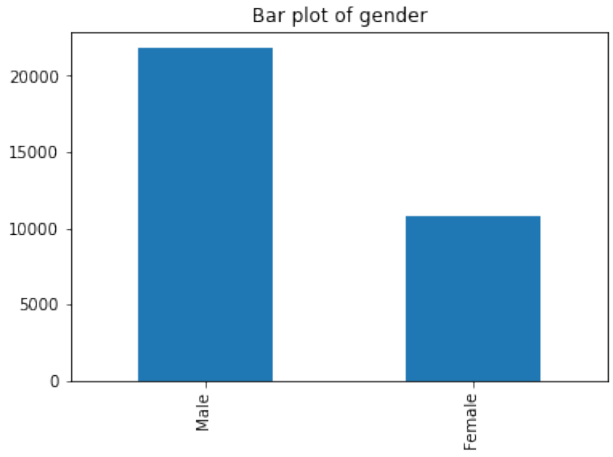


Figure 23: Count of male and female adults. There are 21790 male adults and 10771 female adults.

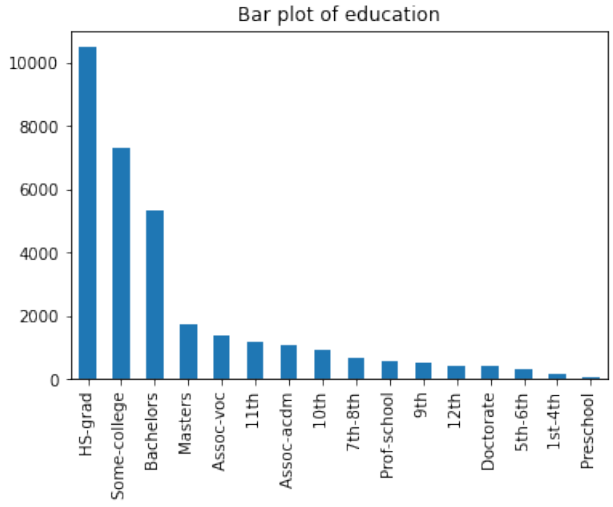


Figure 24: Bar plot of adults with different levels of education. The majority of adults (10501) have only been educated till high school. Several adults have been educated till their Bachelors (5355) and Masters (1723). Very few (414) continued their education till doctorate studies.

The bar plot of the dataset based on native country shows that an overwhleming majority of the adults are from the United States.

One-hot encoding was used for the categorical data columns. The categorial variables are workclass, education, marital status, occupation, relationship, race, sex and native country.

Workclass, occupation and native country are the only columns that had missing values marked as ' ?'. Hence, data imputation was conducted such that the mode of the pertinent columns were used to fill in the missing values.

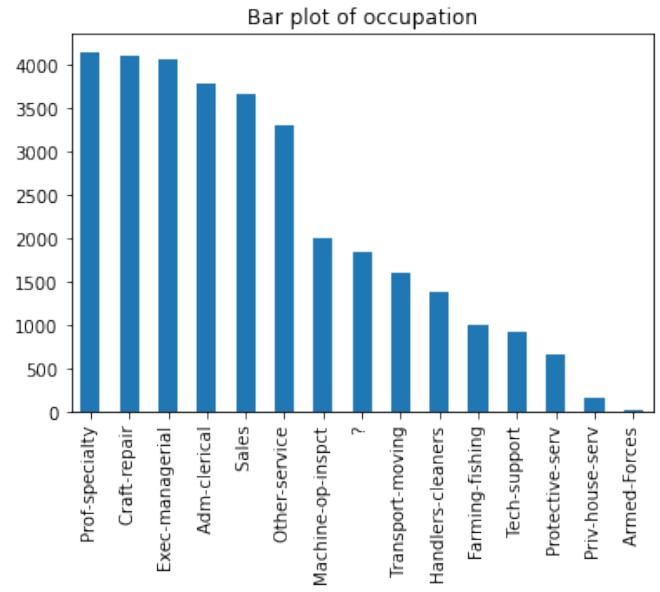


Figure 26: Bar plot of adults with different occupations.

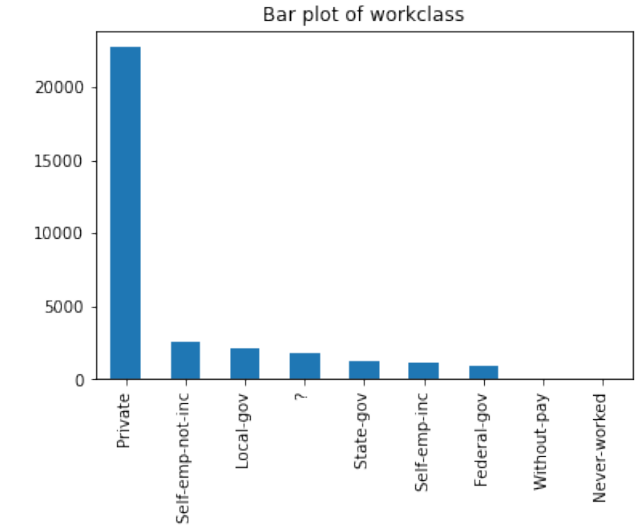


Figure 27: Bar plot of adults with different work class. The overwhelming majority of adults in the dataset work in the private sector.

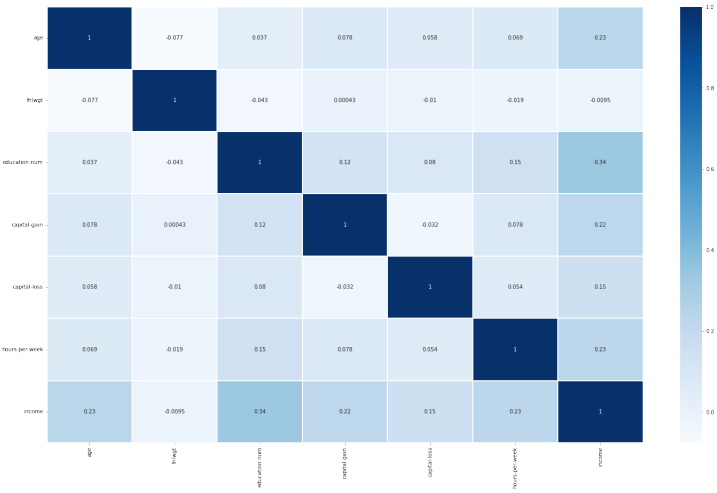


Figure 28: There is absence of conditional independence in this real-world dataset.

Now we will look at the bar plots of different socioeconomic variables by mean income. We defined annual income less than $50000 as 0 and more than $50000 as 1 for making these plots.

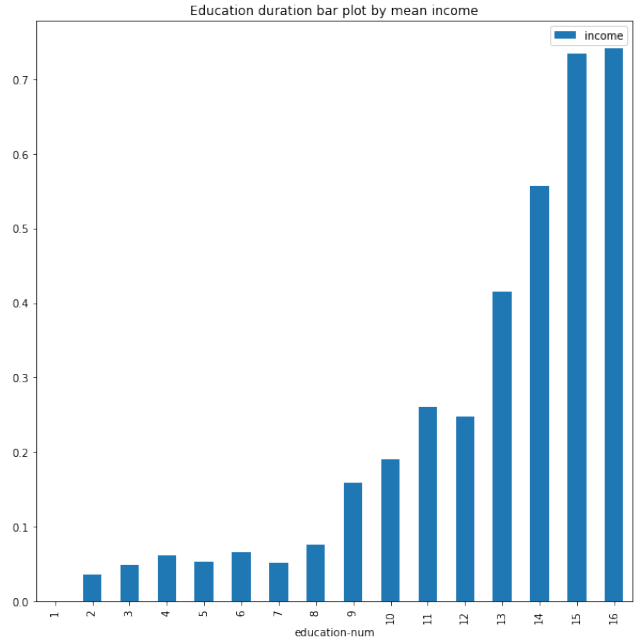


Figure 29: Bar plot of education duration using mean income. We can see that a higher duration of education means that an adult is likely to have a higher income.

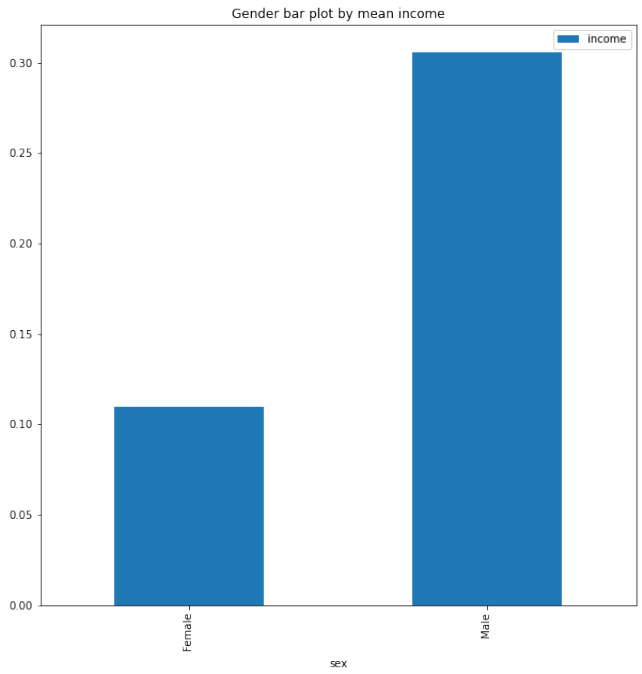


Figure 30: Bar plot of gender using mean income. We can see that men have a higher mean income than women.

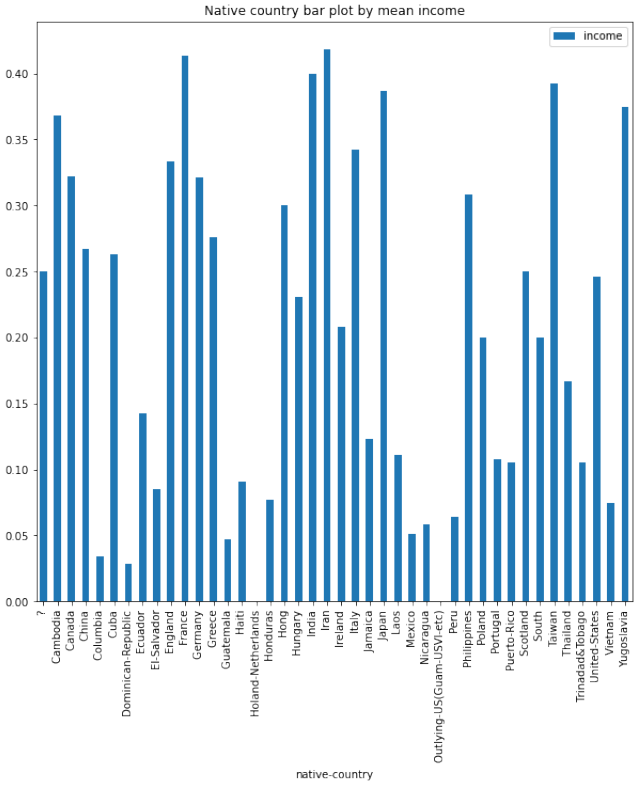


Figure 31: Bar plot of native country using mean income. We can see that adults from countries like India, Iran, France and Taiwan earn a high mean income, although they may have low representation.

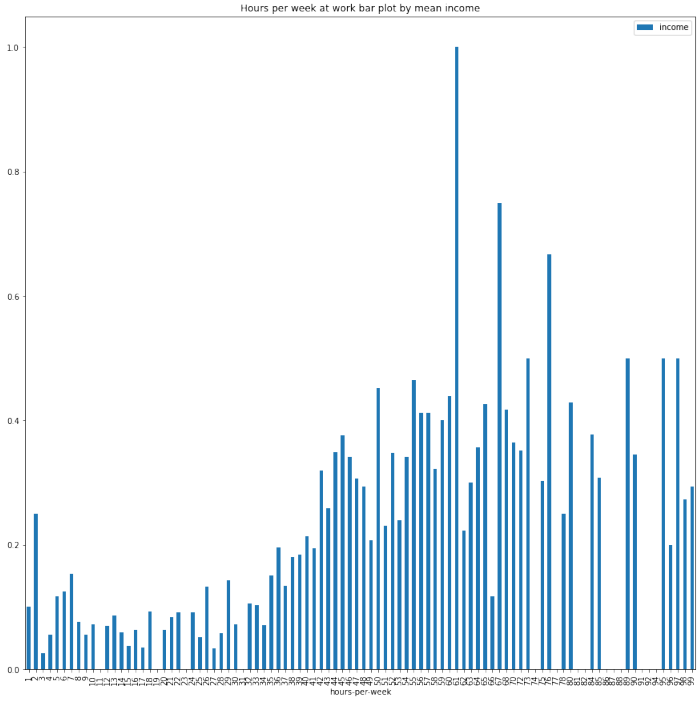


Figure 32: Bar plot of hours per week at work using mean income. There is a variation in income between different hours. More hours at work need not mean higher income.

F. The Naive Bayes classifier model

Using scikit-learn, a Naive Bayes model was trained using 80% of the data. 20% of the data was used for testing / cross-validation. A robust scaler was used scales features using statistics that are robust to outliers. Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set.

Accuracy on the training data was 80.74%.
Accuracy on test data (test / cross-validation set) = 80.04%.

The null accuracy of the test data set is 75.51 %. Null accuracy is the accuracy that could be achieved by always predicting the most frequent class. Since our model's test accuracy is higher, we conclude that the model is performing well and is accurate at classifying an adult's income.

A new Naive Bayes Classifier model was trained with a different value of the var_smoothing parameter. var_smoothing, artificially adds a user-defined value to the distribution's variance (whose default value is derived from the training data set). This essentially widens or makes the curve smoother and accounts for more samples that are further away from the distribution mean. The default value for var_smoothing is 1e-09. In our new model we set the value to be 3.2e-09. Here are the results from the new model:

Accuracy on the training data was 82.97%.
Accuracy on test data (test / cross-validation set) = 82.88%.

When 10 fold cross-validation was used on the training data, the training accuracy was found to be 82.95%. This value is close to the previous training accuracy. The training accuracy is also very close to the test accuracy. These two observations provide strong evidence that the model is **not overfitting** to the data.

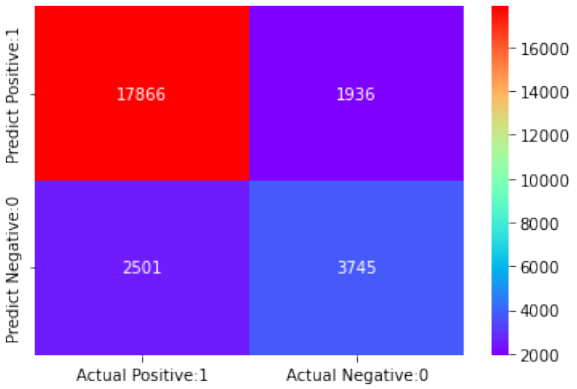


Figure 33: Confusion matrix of the Naive Bayes model on the training data.

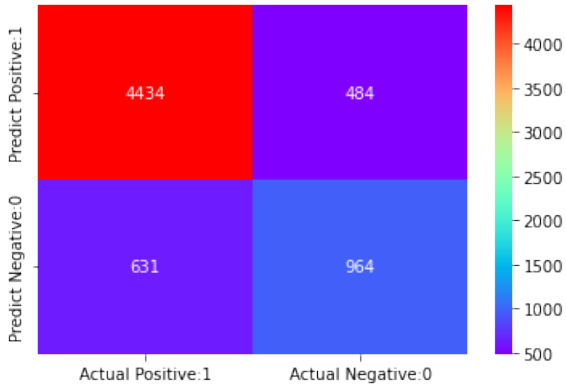


Figure 34: Confusion matrix of the Naive Bayes model on the test / cross-validation data.

The area under the ROC curve for the old model was found to be 0.8947. The area under the ROC curve for the new model was found to be 0.8828. Though there is a slight decrease in the area under the ROC curve, the improvement in accuracy is important.

We have analysed the model performance using the various numerical measures. It would be tedious to examine the probabilities of the various attributes because the one-hot encoding resulted in the creation of more than 100 attributes.

Conclusion

1) The Naive Bayes classifier's performance was successfully demonstrated, both visually and numerically, that the given socioeconomic data can be effectively used to predict whether an adult earns above or below $50000 annually.
2) The classifier had a training accuracy of 82.97% and a test data accuracy of 82.88%, which is an improvement on the old model.
3) The k fold cross validation accuracy, the close values of training and test accuracy, and the improvement over null accuracy provide strong evidence that the model is accurately predicting income without overfitting.

IV. Decision Tree Classifier

Introduction

This section seeks to explore one of the important modelling tools used for classification and prediction - **Decision Tree classifier**. Once we have acquired data with multiple features, one important task is to understand how the variables are related to a target variable. Classification is a statistical method in which the relationship between one or more independent variables and a dependent variable is used to identify the mathematical expression or a set of mathematical rules to assign a given data point to a class. The Decision Tree classifier, one of the well-known machine learning techniques, uses a tree like structure and their possible combinations to solve a particular problem and classify a data point.

Decision Tree models are general, predictive modelling tools that have applications spanning several areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions or mathematical rules. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

By using certain criterion, we can estimate the information contained by each attribute in the dataset. Here are the expressions for 2 common attribute selection measures:

$$E = \sum_{i=1}^{C} -p_i * log_2(p_i)$$

$$G = 1 - \sum_{i=1}^{C} (p_i)^2$$

In the above equation, E refers to the entropy, which measures the impurity in the given dataset. C is the number of classes in the target variable and p_i is the probability associated with the ith class.

G is the gini index. Again C is the number of classes in the target variable and p_i is the probability associated with the ith class.

The attribute selection measures are used to select the splitting attribute used at the node. The model consists of a set of mathematical rules used to split the dataset into different parts as in a tree structure. The structure of a decision tree is shown below.

To demonstrate the effectiveness of the Decision Tree as a classifier, we use the standard features of a car like buying price, maintenance cost and luggage boot size to classify the safety level of cars as unacceptable, acceptable, good or very good. The pertinent data has been collected by Marko Bohanec and Blaz Zupan [4]. Before performing data analysis, model building and classification using the Decision Tree Classifier, we visualise the data.

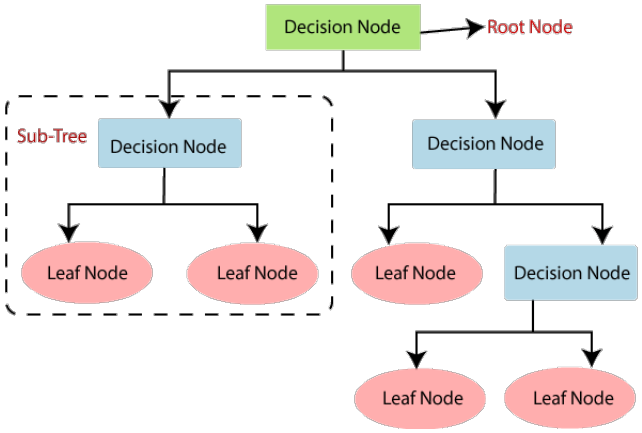


Figure 35: The general structure of a decision tree. **source**

Decision Tree Classifier

It was stated in the introduction section that Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. For our dataset, we focus on decision trees as a classifier.

A decision tree is a tree structure that consists of a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node. Hence, each decision tree model constitutes a precise set of mathematical rules that dictate the class of a data point based on its attributes. Here are some of the assumptions of this model.

A. Assumptions

- Attribute values need to be categorical. If the values are continuous then they are discretized prior to building the model.
- Prior to building the model, the whole training dataset is considered as the root.
- The order of placing attributes as root or internal node of the tree and setting the hierarchy of the importance of the attributes in the mathematical rule set is done by using a statistical approach (using information gain measures).

We shall now examine the important terms pertaining to decision tree models.

B. Decision Tree terminology

In a decision tree model, there is a tree like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the previous test, and each leaf node represents a class label for the target variable. The paths from the root node to leaf node represent the

mathematical rules that perform the classification. The terms involved in decision tree models are as follows:

- Root Node - it represents the entire dataset or sample set. This gets further divided into two or more homogeneous sets of branches.
- Splitting - the process of dividing a node into two or more sub-nodes based on the test for the attribute. When a sub-node splits into further sub-nodes, then it is called a decision node.
- Terminal Node - nodes that do not split are called terminal nodes.
- Pruning - When sub-nodes of a decision node are removed, this process is called pruning. It is the opposite process of splitting.
- Parent and Child Node - A node, which is divided into sub-nodes is called the parent node of sub-nodes where the sub-nodes are the children of the parent node.
- Branch/Sub-Tree - a sub-section of an entire tree is called a branch or sub-tree.

The process of classification is as follows:

1) For each attribute in the dataset, the decision tree model forms a node. The most important attribute is placed at the root node.
2) For evaluating a given data point, we start at the root node and we work our way down the tree by following the corresponding nodes that meets our tests / rules.
3) This tree traversal process continues until a terminal node is reached. It contains the prediction or the outcome of the decision tree model.

To learn a Decision Tree model, we need to identify the attributes in the dataset which we consider as the root node and the attributes whose values are tested at each level of the tree. This process is known as attribute selection. There are different attribute selection measures to identify the attribute which can be at each level of the tree. There are 2 popular attribute selection measures:

- Information gain: assume attributes to be categorical. Information gain is the decrease in entropy. Information gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values. The attribute with the highest information gain is chosen as the splitting attribute at the node.

$$E = \sum_{i=1}^{C} -p_i * log_2(p_i)$$

  In the above equation, E refers to the entropy, which measures the impurity in the given dataset. C is the number of classes in the target variable and p_i is the probability associated with the ith class.
- Gini index: assume attributes to be continuous.

$$G = 1 - \sum_{i=1}^{C} (p_i)^2$$

  G is the gini index. Again C is the number of classes in the target variable and p_i is the probability associated with the ith class. The gini index says that if we randomly select two items from a population, they must be of the same class and probability for this is 1 if the population is pure. Steps to calculate gini index for a split:
  1) Calculate G for sub-nodes, using formula sum of the square of probability for success and failure (p2+q2).
  2) Calculate G for split using weighted G score of each node of that split.

  In case of a discrete-valued attribute, the subset that gives the minimum G for that chosen is selected as a splitting attribute. In the case of continuous-valued attributes, the strategy is to select each pair of adjacent values as a possible split-point and point with smaller G chosen as the splitting point. The attribute with minimum G is chosen as the splitting attribute.

CAN STANDARD FEATURES OF A CAR BE USED TO DETERMINE THE SAFETY LEVEL OF A CAR USING A DECISION TREE?

In this section, we see the predictions and relationships revealed by the decision tree model between physical car features and car safety. We want to see whether we can accurately classify a car as unacceptable, acceptable, good or very good. The list of attributes in the dataset are:

1) Buying price as very high, high, medium or low
2) Maintenance cost as very high, high, medium or low
3) Number of doors as 2, 3, 4 or ¿=5
4) Number of persons to carry as 2, 4 or more
5) Size of luggage boot as small, medium or big
6) Safety level as low, medium or high

C. Visualisation of the dataset

There were no missing data points and hence, no imputation was required. Simple bar plots and correlation plots were made for several variables to understand the frequency of certain attributes within that variable.

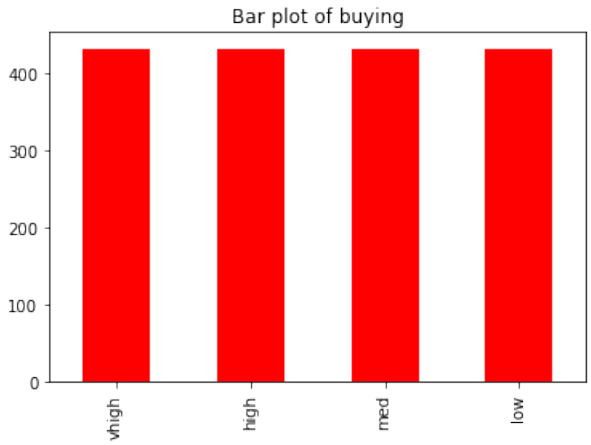


Figure 36: Barplot of buying price.

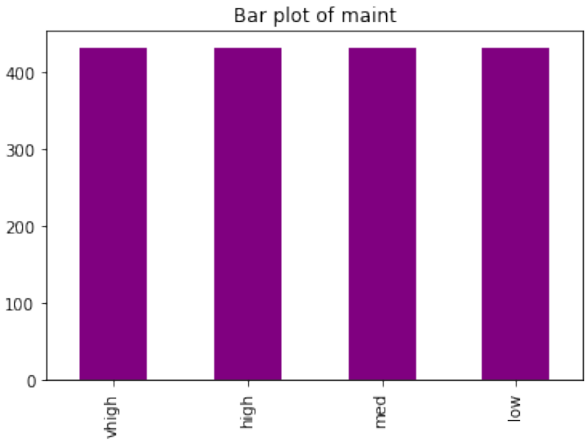


Figure 37: Barplot of maintenance cost.

Figure 38: Barplot of doors.

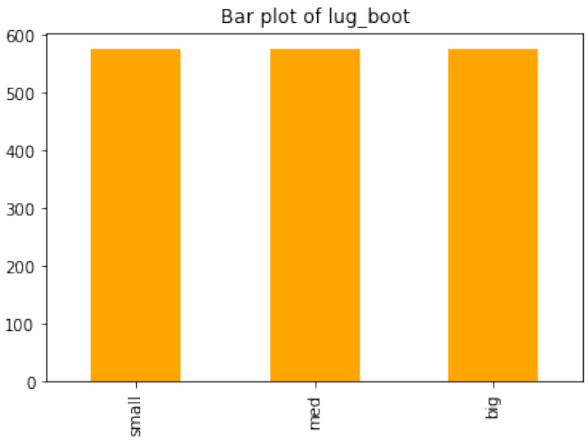


Figure 40: Barplot of luggage boot size.

The bar plot of the attributes show that all the values in each attribute are equal in number. Only the target variable is skewed with unacceptable and acceptable cars having more data points than the good and very good cars.

Figure 39: Barplot of persons to carry.

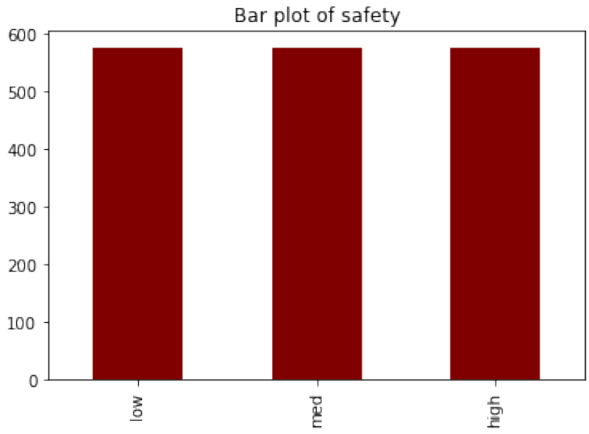


Figure 41: Barplot of estimated safety level.

Figure 43: Correlation matrix of the attributes. One can easily see that there is absence of correlation between the attributes in this dataset.

The categorical variables were encoded using ordinal variables.

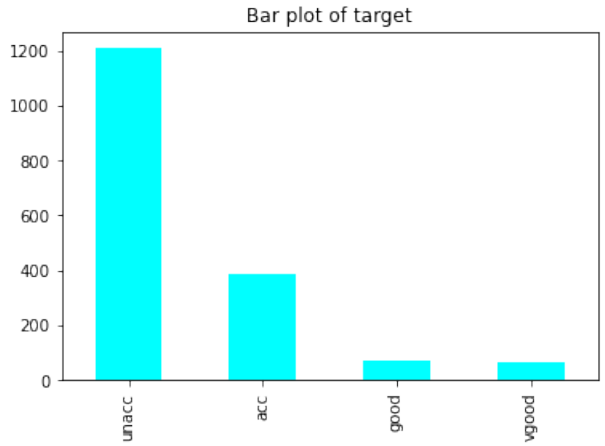


Figure 42: Barplot of target variable.

D. The Decision Tree classifier

Using scikit-learn, a Decision tree model was trained using 85% of the data. 15% of the data was used for testing / cross-validation.

Trees of depths 1 to 30 were trained and tested with information gain and gini index measures to see the performance.

Two models were trained with depth of 10 each and information gain and gini index measures and the performance is:

Gini index model:
Accuracy on the training data was 99.39%.
Accuracy on test data (test / cross-validation set) = 97.69%.

Information gain model:
Accuracy on the training data was 99.32%.
Accuracy on test data (test / cross-validation set) = 97.69%.

The 2 final models were trained exactly as before but this time, the number of features to consider when looking for the best split (max_features) was set as 5 (instead of all 6 variables). Here are 2 plots showing the performance of these models against the depth of trees.

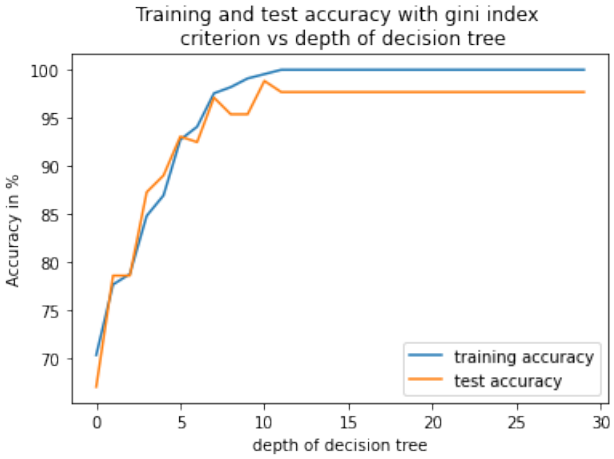


Figure 44

Figure 45

Two final models were trained with depth of 11 for the gini index model and depth of 10 for the entropy model. The performance of these models are:

Gini index model:
Accuracy on the training data was 99.55%.
Accuracy on test data (test / cross-validation set) = 98.84%.

Entropy model:
Accuracy on the training data was 98.46%.
Accuracy on test data (test / cross-validation set) = 98.84%.

The training accuracy is very close to the test accuracy in both cases and the accuracy is higher than the old model. This observation shows strong evidence that the model is **not overfitting** to the data.

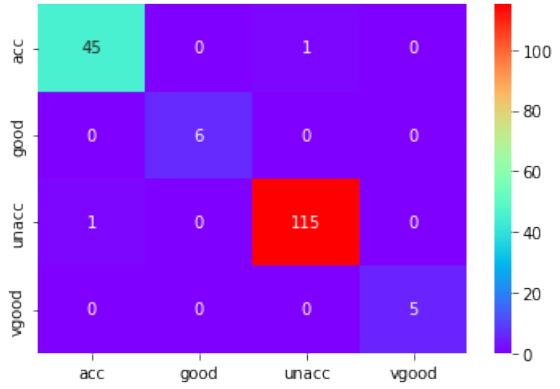


Figure 46: Confusion matrix of the decision tree model trained on gini index.

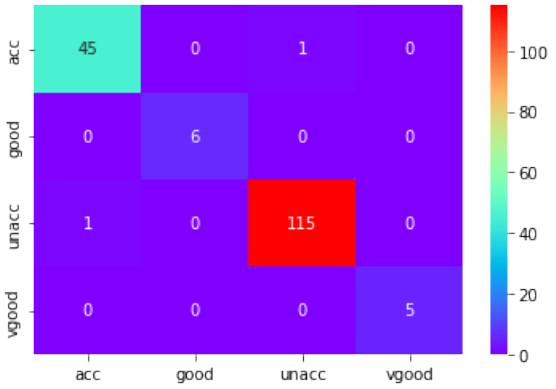


Figure 47: Confusion matrix of the decision tree mode trained on information gain.

	precision	recall	f1-score	support
acc	0.98	0.98	0.98	46
good	1.00	1.00	1.00	6
unacc	0.99	0.99	0.99	116
vgood	1.00	1.00	1.00	5
accuracy			0.99	173
macro avg	0.99	0.99	0.99	173
weighted avg	0.99	0.99	0.99	173

Figure 48: Classifier report for decision tree model trained on gini index.

	precision	recall	f1-score	support
acc	0.98	0.98	0.98	46
good	1.00	1.00	1.00	6
unacc	0.99	0.99	0.99	116
vgood	1.00	1.00	1.00	5
accuracy			0.99	173
macro avg	0.99	0.99	0.99	173
weighted avg	0.99	0.99	0.99	173

Figure 49: Classifier report for decision tree model trained on information gain.

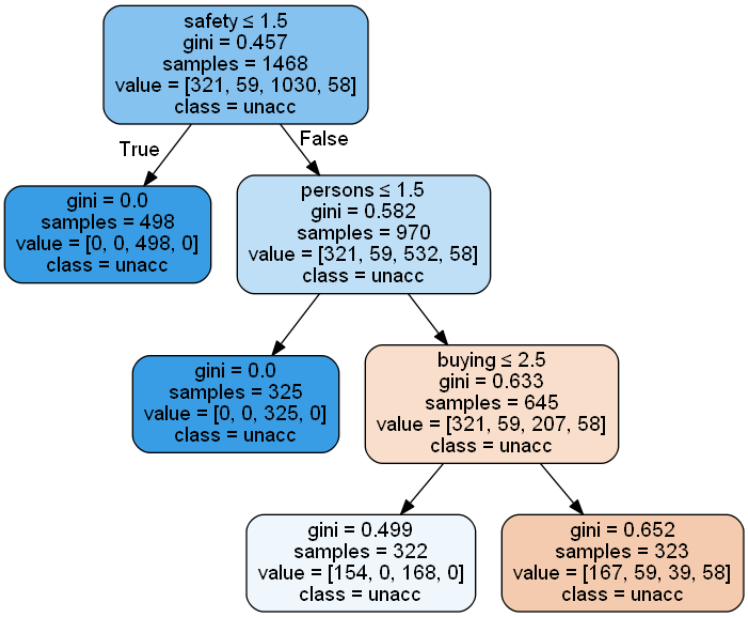


Figure 50: Visualisation of the tree structure and its mathematical rules when depth is 3. Trees of depth 10 and 11 are difficult to visualise in one image since it would be a large image.

Conclusion

1) The Decision Tree classifier's performance was successfully demonstrated, both visually and numerically, that the given car feature data can be effectively used to predict car safety and acceptability.
2) The final classifiers had a training accuracy of 99.55% and 98.46%, and a test data accuracy of 98.84%. These accuracy values are higher than the previous model.
3) The close values of training and test accuracy, the classifier report and the confusion matrix provide strong evidence that the model is accurately predicting car acceptability without overfitting.

V. Random Forest Classifier

Introduction

This section seeks to explore one of the important modelling tools used for classification and prediction - **Random Forest classifier**. Once we have acquired data with multiple features, one important task is to understand how the variables are related to a target variable. Classification is a statistical method in which the relationship between one or more independent variables and a dependent variable is used to identify the mathematical expression or a set of mathematical rules to assign a given data point to a class. The Random Forest classifier, one of the well-known machine learning techniques, uses an ensemble of decision tree classifiers. The theory is that a large number of uncorrelated trees will create more accurate predictions than one individual decision tree. Decision trees have a tree like structure and the rules at each branch are used to solve a particular problem and classify a data point. To understand Random Forest classifiers, we shall first focus on Decision Trees.

Decision Tree models are general, predictive modelling tools that have applications spanning several areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions or mathematical rules. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method. Decision Trees, and by extension Random Forest classifiers, are used for both classification and regression tasks. Decision trees create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Random forest classifiers combine different models such that the final result arises through a voting mechanism from the trees. By using certain criterion, we can estimate the information contained by each attribute in the dataset. Here are the expressions for 2 common attribute selection measures:

$$E = \sum_{i=1}^{C} -p_i * log_2(p_i)$$

$$G = 1 - \sum_{i=1}^{C} (p_i)^2$$

In the above equation, E refers to the entropy, which measures the impurity in the given dataset. C is the number of classes in the target variable and p_i is the probability associated with the ith class.

G is the gini index. Again C is the number of classes in the target variable and p_i is the probability associated with the ith class.

The attribute selection measures are used to select the splitting attribute used at the node. The model consists of a set of mathematical rules used to split the dataset into different parts as in a tree structure. The structure of a decision tree is shown below.

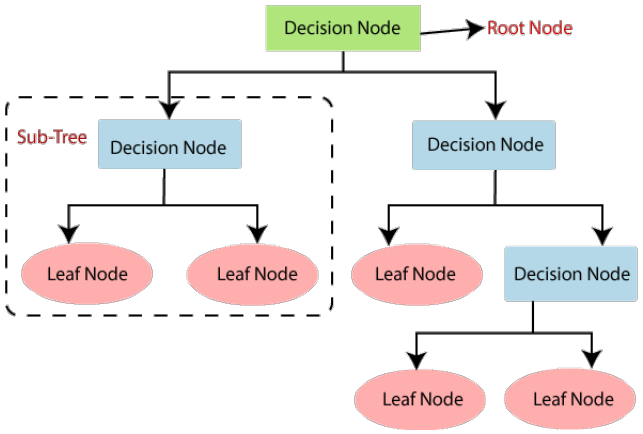


Figure 51: The general structure of a decision tree. **source**

To demonstrate the effectiveness of the Random Forest as a classifier, we use the standard features of a car like buying price, maintenance cost and luggage boot size to classify the safety level of cars as unacceptable, acceptable, good or very good. The pertinent data has been collected by Marko Bohanec and Blaz Zupan [4]. Before performing data analysis, model building and classification using the Random Forest Classifier, we visualise the data.

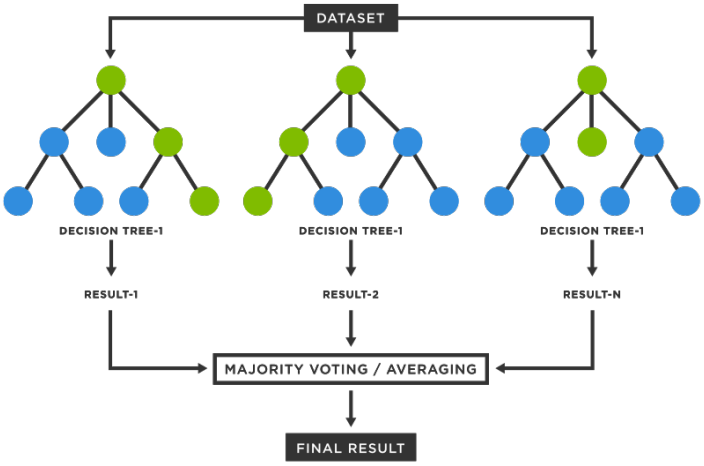


Figure 52: The general structure of a Random Forest classifier. **source**

A Random Forest classifier is a bagging algorithm that is an ensemble of Decision Tree classifiers. Hence, one should look at the important terminologies in Decision Trees in the previous section before reading about Random Forests.

RANDOM FOREST CLASSIFIER

The random forest algorithm is a bagging algorithm. Bagging, also known as bootstrap aggregating, is the aggregation of multiple versions of a prediction model. Each model is trained individually, and combined using an averaging process. The primary focus of bagging is to achieve less variance than any model has individually. Random forest has nearly the same hyperparameters as a decision tree.

The random forest draws random bootstrap samples from the training set. The random forest also adds randomness to the model while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds like a normal decision tree classifier does.

Another advantage of the random forest classifier is that it is easy to measure the relative importance of each feature on the prediction. Sklearn contains a useful tool for this that measures a feature's importance by looking at how much the tree nodes that use that feature reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results so the sum of all importance is equal to one.

Advantages:

- Robust to outliers.
- Works well for non-linear data.
- Low risk of overfitting.
- Runs efficiently on large datasets.

Disadvantages:

- Slow training.
- Can be biased if categorical variables are used.

CAN STANDARD FEATURES OF A CAR BE USED TO DETERMINE THE SAFETY LEVEL OF A CAR USING A DECISION TREE?

In this section, we see the predictions and relationships revealed by the random forest classifier between physical car features and car safety. We want to see whether we can accurately classify a car as unacceptable, acceptable, good or very good. The list of attributes in the dataset are:

1) Buying price as very high, high, medium or low
2) Maintenance cost as very high, high, medium or low
3) Number of doors as 2, 3, 4 or ¿=5
4) Number of persons to carry as 2, 4 or more
5) Size of luggage boot as small, medium or big
6) Safety level as low, medium or high

A. *Visualisation of the dataset*

There were no missing data points and hence, no imputation was required. Simple bar plots and correlation plots were made for several variables to understand the frequency of certain attributes within that variable.

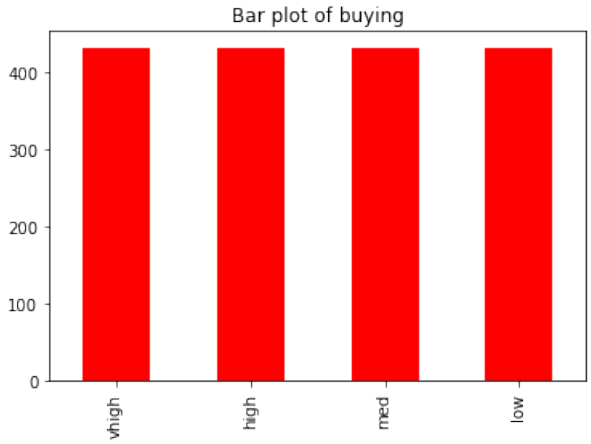


Figure 53: Barplot of buying price.

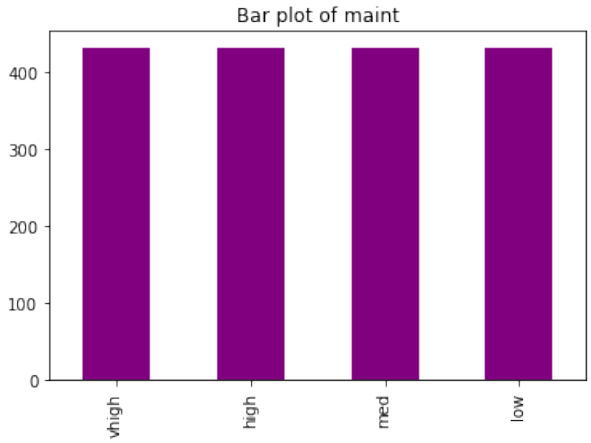


Figure 54: Barplot of maintenance cost.

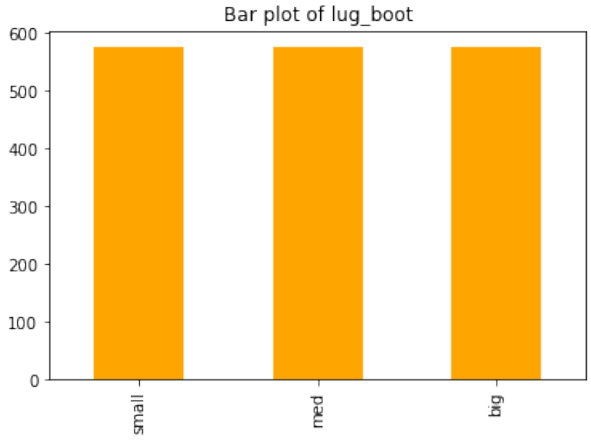


Figure 57: Barplot of luggage boot size.

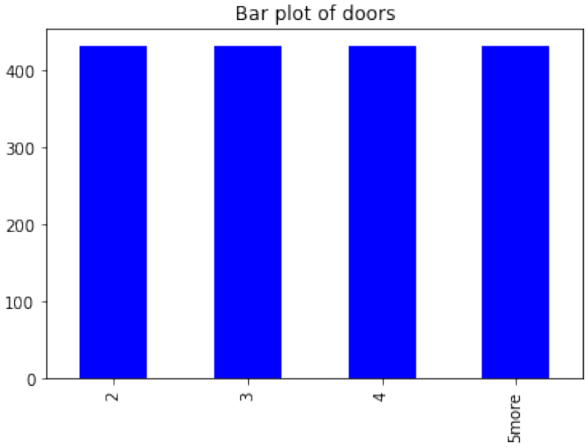


Figure 55: Barplot of doors.

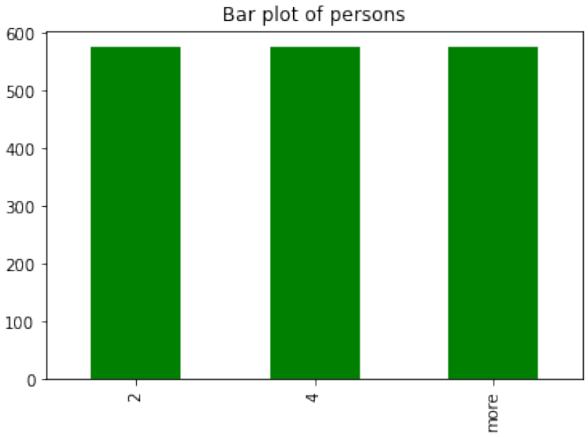


Figure 56: Barplot of persons to carry.

The bar plot of the attributes show that all the values in each attribute are equal in number. Only the target variable is skewed with unacceptable and acceptable cars having more data points than the good and very good cars.

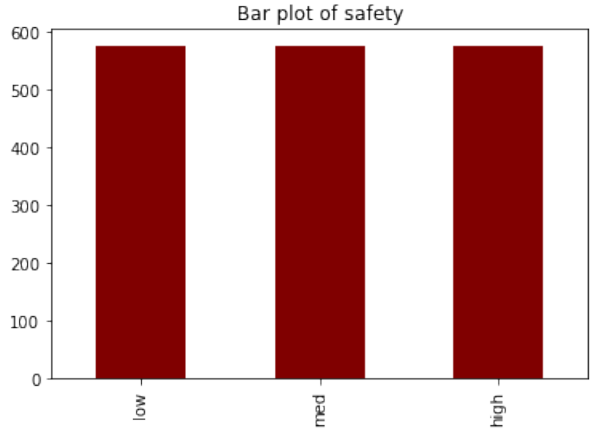


Figure 58: Barplot of estimated safety level.

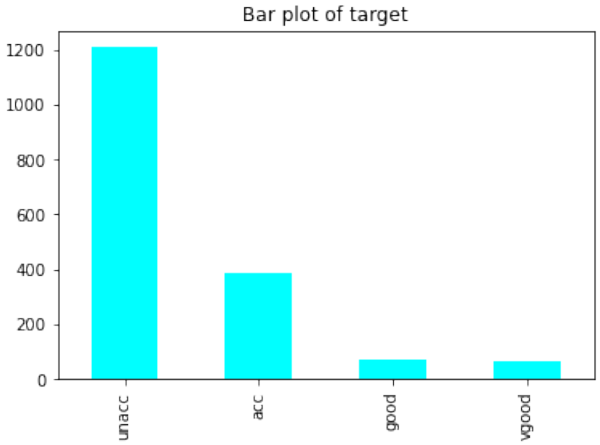


Figure 59: Barplot of target variable.

Figure 60: Correlation matrix of the attributes. One can easily see that there is absence of correlation between the attributes in this dataset.

The categorical variables were encoded using ordinal variables.

B. The Random Forest classifier

The initial random forest classifier models were trained using 85% of the data. 15% of the data was used for testing / cross-validation. In this case, the models were trained with varying number of decision tree models. Models containing 1 to 100 decision trees were trained and tested and accuracy was measured to see the performance.

The plots showed that the model with 27 trees had the highest accuracy. Next, we looked at the importance of training data and trained a random forest classifier with 27 trees using different amounts of training data. The training data set size varied from 1% to 99%. Once again, accuracy was measured to see the performance.

shows that the model trained with 79% of data and using 21% of data as test data has the highest accuracy. Thus, the initial model was trained using 79% of the data and 27 decision tree models.

The initial model's performance is:

Accuracy on the training data was **100%**.
Accuracy on test data set was **97.7961%**.

The new model was trained such that only 5 features were used to determine the best split, instead of all 6. Here are the plots for the new model:

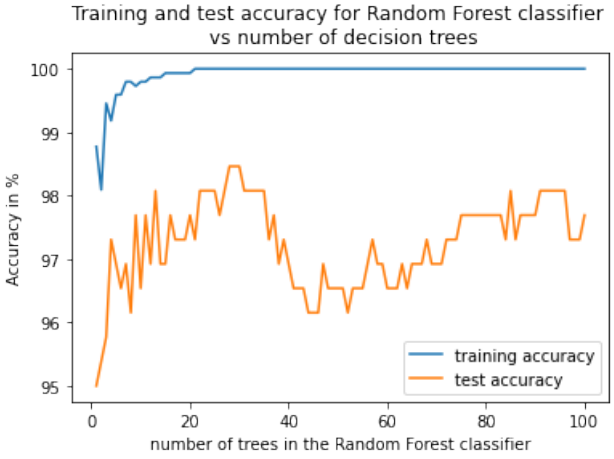


Figure 61: Training and test accuracy of random forest classifiers trained on 85% of the data plotted against number of decision trees.

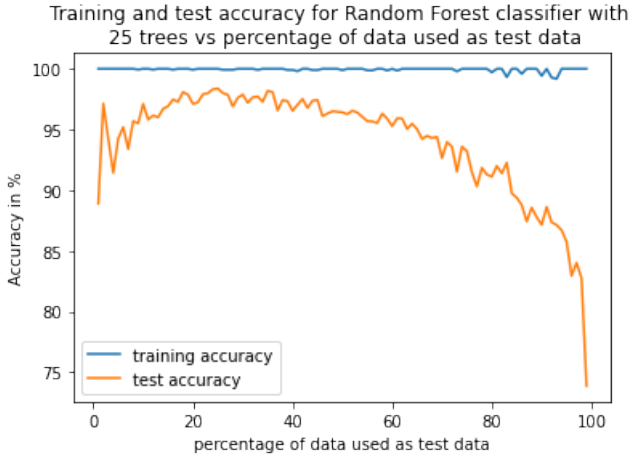


Figure 62: Training and test accuracy of random forest classifiers with 27 decision trees plotted against percentage of data used as test data.

The final model's performance is:

Accuracy on the training data was **100%**.
Accuracy on test data set was **98.3796%**.

The test accuracy is higher than the previous model. The training accuracy is very close to the test accuracy. This observation shows strong evidence that the model is **not overfitting** to the data.

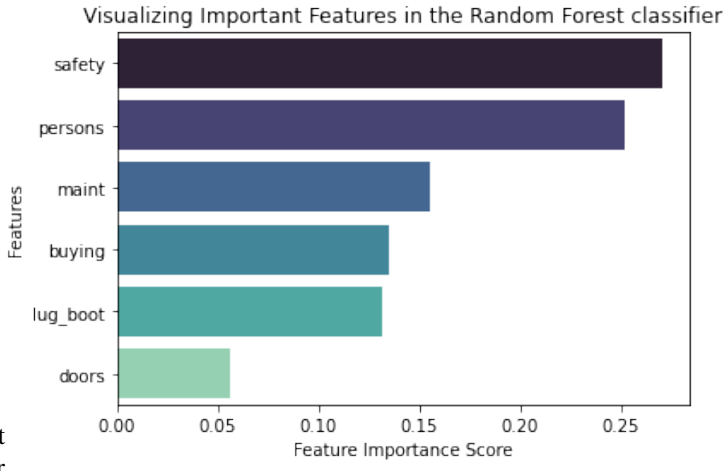


Figure 63: Feature importance of the different features in the data set.

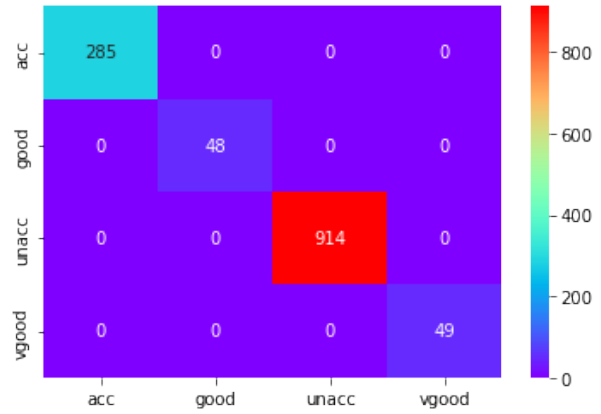


Figure 64: Confusion matrix of the decision tree model based on predictions of the training data set.

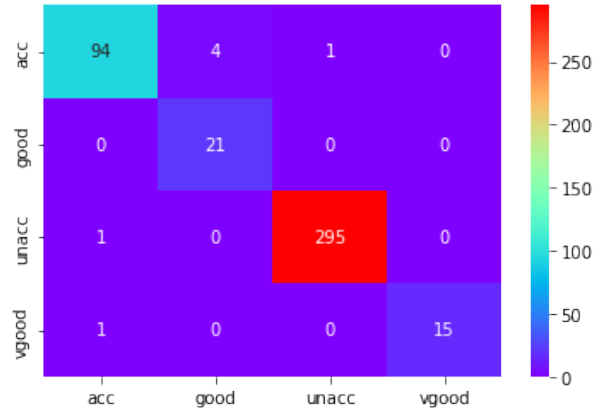


Figure 65: Confusion matrix of the decision tree model based on predictions of the test data set.

	precision	recall	f1-score	support
acc	0.98	0.95	0.96	99
good	0.84	1.00	0.91	21
unacc	1.00	1.00	1.00	296
vgood	1.00	0.94	0.97	16
accuracy			0.98	432
macro avg	0.95	0.97	0.96	432
weighted avg	0.99	0.98	0.98	432

Figure 66: Classifier report for the final random forest classifier model.

Conclusion

1) The Random Forest classifier's performance was successfully demonstrated, both visually and numerically, that the given car feature data can be effectively used to predict car safety and acceptability.
2) The final classifier model trained on 79% of the data using 27 trees had a training accuracy of 100% and a test data accuracy of 98.3796%, which is higher than the previous model.
3) The close values of training and test accuracy, the classifier report and the confusion matrix provide strong evidence that the model is accurately predicting car acceptability without overfitting.
4) The feature importance plot shows that the estimated safety is the most useful feature for prediction, followed by persons (car capacity), buying price, maintenance cost and luggage boot size. The least important feature for prediction is the number of doors.

VI. Support Vector Machines

Introduction

This section seeks to explore one of the important modelling tools used for classification and prediction - **Support Vector Machines** (SVMs). Once we have acquired data with multiple features, one important task is to understand how the variables are related to a target variable. Classification is a statistical method in which the relationship between one or more independent variables and a dependent variable is used to identify the mathematical expression or a set of mathematical rules to assign a given data point to a class. Support Vector Machines, one of the well-known machine learning techniques, uses the data points closest to a potential decision boundary (known as support vectors) to build an optimal hyperplane that separates data points from 2 classes. SVMs can only perform binary classification, but if the data is not linearly separable, it can transform the original data by mapping it to a new space via kernel functions and then find a hyperplane that separates the 2 classes. This makes it a powerful algorithm. SVMs can also be used for regression (Support Vector Regression).

Support vectors are the data points that lie closest to the decision surface (or hyperplane). They are the data points most difficult to classify. They have direct bearing on the optimum location of the decision surface and hence, SVM focuses on these data points in constructing the decision boundary. The distance between the decision boundary and the closest data points is referred to as the margin. SVMs have the following objectives:

- Maximizing the margin around the separating hyperplane.
- Obtaining a hyperplane that correctly separates as many data points as possible i.e. has as high an accuracy as possible.

The objective function for the soft margin SVM can be written in its primal form as follows:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$$

subject to

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, \quad i = 1, 2...N$$

Both the SVM objectives may not be always achieved. The value of C (penalty) is an important hyperparameter for SVMs that influences the number of violations of the margin allowed in the training data and determines the importance of achieving the second objective pertaining to accuracy. A large value of C increases the bias, decreases the variance, provides a small margin and may result in overfitting. A small value of C reduces bias, increases variance, provides a large margin and may result in misclassification of outliers.

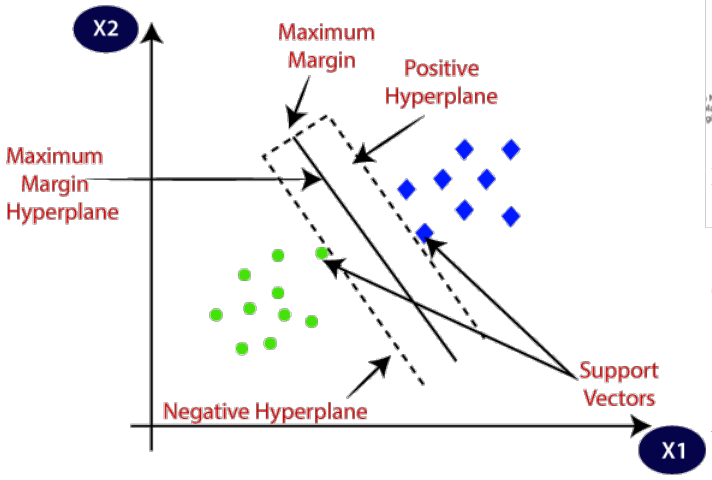


Figure 67: A diagram illustrating the important components in SVM. **source**

To demonstrate the effectiveness of SVMs as a classifier, we use features of the DM-SNR curve and the integrated pulse profile such as mean and standard deviation to classify a star as a pulsar or not. Before performing data analysis, model building and classification using the SVM, we visualise the data and perform feature engineering.

We shall look at SVMs more closely before data visualisation and analysis.

Support Vector Machines

SVMs are powerful supervised learning machine learning algorithms used for classification, regression and outlier detection. In our case, we shall use SVMs as a non-probabilistic binary linear classifier. SVMs can be used for linear classification purposes. As mentioned in the introduction, in addition to performing linear classification, SVMs can efficiently perform a non-linear classification using the kernel trick. It enable us to implicitly map the inputs into high dimensional feature spaces. SVMs are primarily motivated by the principle of optimal separation, the idea that a good classifier finds the largest gap possible between data points of different classes.

Logistic regression is a probabilistic binary linear classifier since it calculates the probability that a data point belongs to one of two classes. Logistic regression attempts to maximize the probability of the classes of known data points according to the model. Hence, it may place the classification boundary arbitrarily close to a particular data point. This violates the commonsense notion that a good classifier should not place a boundary near a known data point, since data points that are close to each other should be of the same class. SVMs solve this problem as they explicitly try to maximise the distance between the decision boundary and the nearest data point.

A. Assumptions

- The margin should be as large as possible.
- The support vectors are the most useful data points because they are the ones most likely to be incorrectly classified.

The second assumption leads to a desirable property of SVMs. After training, the SVM can disregard all other data points, and just perform classification using only the support vectors. Once classification is done, an SVM can predict a data point's class very efficiently, since it only needs to use a handful of support vectors, instead of the entire dataset. This means that the primary goal of training SVMs is to find support vectors in the dataset that both separate the data and find the maximum margin between classes.

B. Support Vector Machine terminology

- Hyperplane / Decision Boundary - the mathematical expression that separates data points having different class labels. Multiple decision boundaries can exist but the one that maximises the margin is chosen unless soft-margin SVM is used.
- Support Vectors - the data points which are closest to the hyperplane. These data points will define the separating line or hyperplane better by calculating margins.
- Margin - the gap between the two lines on the closest data points. It is calculated as the perpendicular distance from the decision boundary to support vectors or closest data points. In SVMs, we try to maximize this separation gap so that we get maximum margin unless the presence of outliers causes us to use soft margin SVM.

If we want to maximise the margin at all costs, we use hard-margin SVM. If there are many outliers and we want to reduce misclassification, we use soft-margin SVM that has a slack hyperparameter that determines how much one is willing to deviate from the maximum margin.

Objective function of the hard-margin SVM:

$$\min_{w} \frac{1}{2} w^T w$$

subject to

$$y_i(w^T\phi(x_i)+b) \geq 1, \quad i = 1, 2...N$$

Objective function of the soft-margin SVM:

$$\min_{w,b,\xi} \frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i$$

subject to

$$y_i(w^T\phi(x_i)+b) \geq 1-\xi_i, \quad i = 1, 2...N$$

$$\xi_i \geq 0, \quad i = 1, 2...N$$

$\phi(x_i)$ in the above equations is the kernel function acting on data point x_i. The kernel function can be the identity function or a Radial Basis Function (RBF) like the Gaussian function. The equation for the RBF kernel:

$$K(x_i, x_j) = exp(\frac{||x_i - x_j||^2}{2\sigma^2})$$

In the case of the RBF kernel, σ plays a role in the amplification of the distance between x_i and x_j. If the distance between x_i and x_j is much larger than σ, then the value of the kernel function tends to zero. Thus, a smaller σ results in a local classifier and a larger σ results in a more general classifier. In terms of bias and variance, the kernel with smaller σ tends to have lower bias and higher variance while larger σ tends to have higher bias and lower variance.

C. Optimization

The previous objective functions are called as the primal form of the SVM problem. The primal form can be converted into a dual form of the SVM.

$$\max_{\alpha}\min_{w,b} \frac{1}{2}w^T w + C\sum_{i=1}^{N}(\alpha_i + 1 - w^T\phi(x_i) + b)$$

subject to

$$0 \leq \alpha_i \leq C, \quad i = 1, 2...N$$

This optimization problem can be simplified (by setting some gradients to 0) to:

$$\max_{\alpha}\sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{N}(y_i\alpha_i(\phi(x_i)^T\phi(x_j))y_j\alpha_j)$$

subject to

$$0 \leq \alpha_i \leq C, \quad i = 1, 2...N$$

As stated by the representer theorem [1]:

$$w = \sum_{i=1}^{N}(y_i\alpha_i\phi(x_i))$$

We learn α_i from the dataset and then obtain the weights. Stating the SVM problem in its dual form is useful because this form allows for faster computation and the use of the kernel trick

CAN THE FEATURES OF THE INTEGRATED PULSE PROFILE AND DM-SNR CURVE BE USED TO IDENTIFY PULSARS USING A SUPPORT VECTOR MACHINE?

In this section, we see the predictions and relationships revealed by the SVM between the statistics of the integrated pulse profile and dm-snr curve and presence/absence of a pulsar. We want to see whether we can accurately classify a star as a pulsar or not. The list of features in the dataset are:

1) Mean of the integrated profile
2) Standard deviation of the integrated profile.
3) Excess kurtosis of the integrated profile.
4) Skewness of the integrated profile.
5) Mean of the DM-SNR curve.
6) Standard deviation of the DM-SNR curve.
7) Excess kurtosis of the DM-SNR curve.
8) Skewness of the DM-SNR curve.

The target is a binary variable - 0 indicates absence and 1 indicates the presence of a pulsar.

D. Visualisation and Imputation of the dataset

There were several missing data points and hence imputation was required.

- 1735 missing data points in the training data set and 767 missing data points in the test data set for Excess kurtosis of the integrated profile.
- 1178 missing data points in the training data set and 524 missing data points in the test data set for Standard deviation of the DM-SNR curve.
- 625 missing data points in the training data set and 244 missing data points in the test data set for Skewness of the DM-SNR curve.

Imputation was performed such that the mean of the remaining data points was used to replace the missing values.

Histogram plots, a pair plot for the pairwise relationships between features, a correlation plot and a bar plot (for target) were made to understand the relationships between features and the spread of the data.

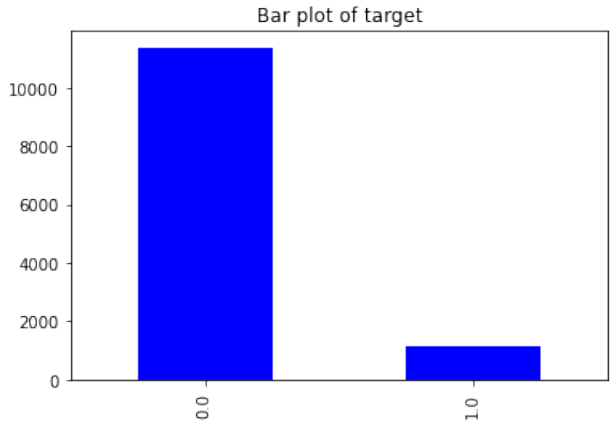


Figure 68: Bar plot of target class. Clearly, there is class imbalance. There are 11375 data points for target = 0 and 1153 data points for target = 1. Class label 0 is represented by 90.79% of the data points.

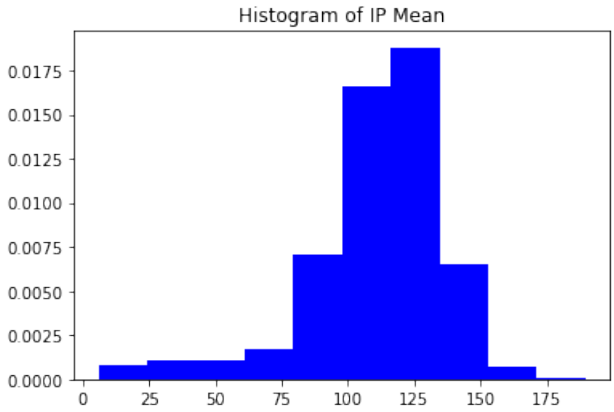


Figure 69: Histogram plot of the mean of the integrated profile.

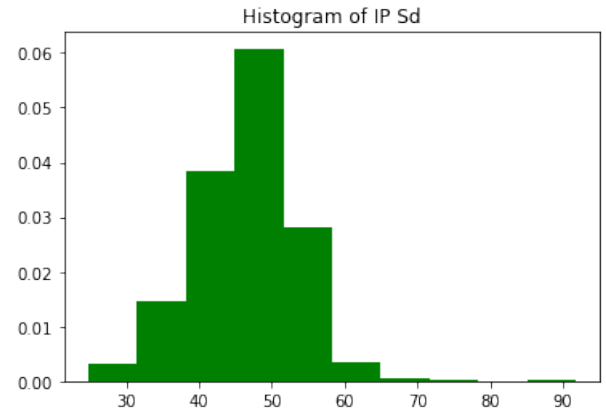


Figure 70: Histogram plot of the standard deviation of the integrated profile.

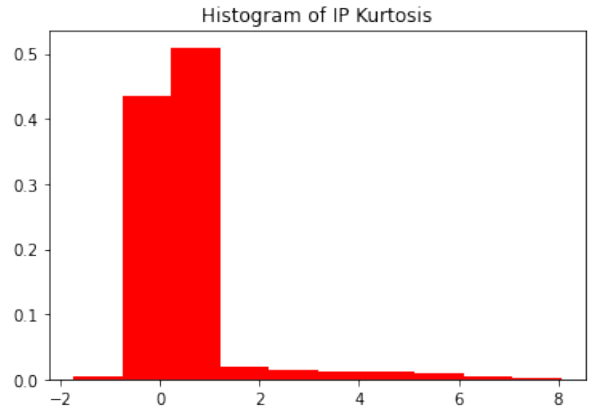


Figure 71: Histogram plot of the excess kurtosis of the integrated profile.

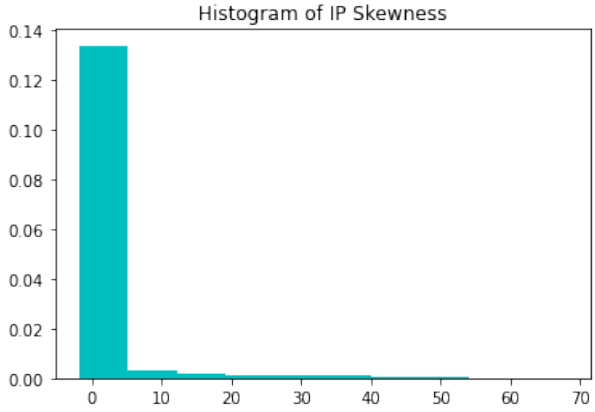


Figure 72: Histogram plot of the skewness of the integrated profile.

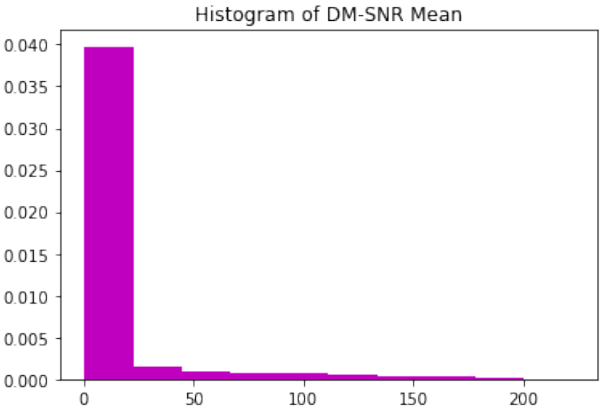


Figure 73: Histogram plot of the mean of the DM-SNR curve.

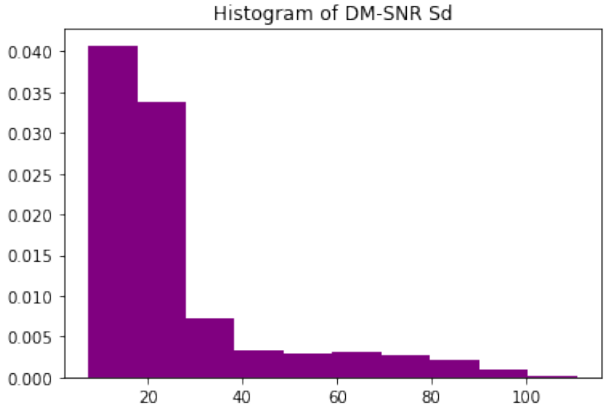


Figure 74: Histogram plot of the standard deviation of the DM-SNR curve.

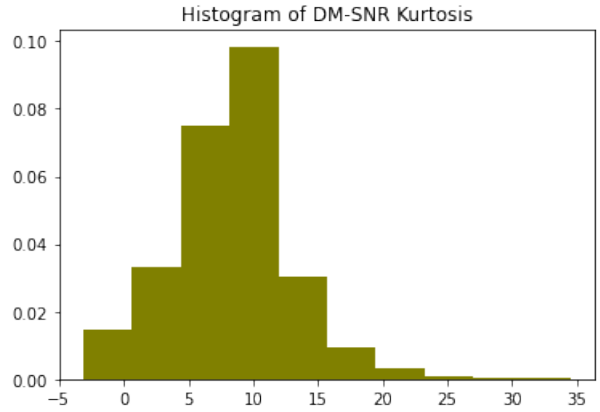


Figure 75: Histogram plot of the excess kurtosis of the DM-SNR curve.

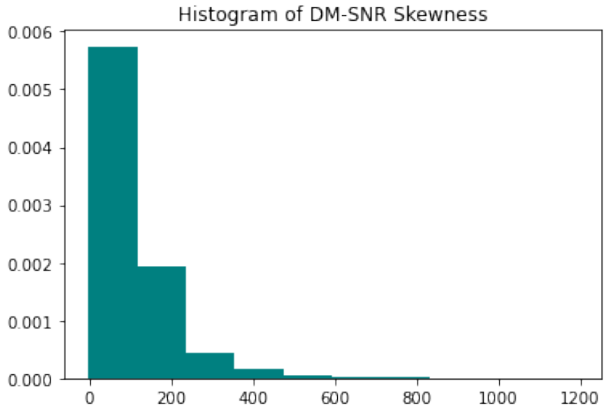


Figure 76: Histogram plot of the skewness of the DM-SNR curve.

The histogram plots of the features show that there are a significant number of outliers for several features.

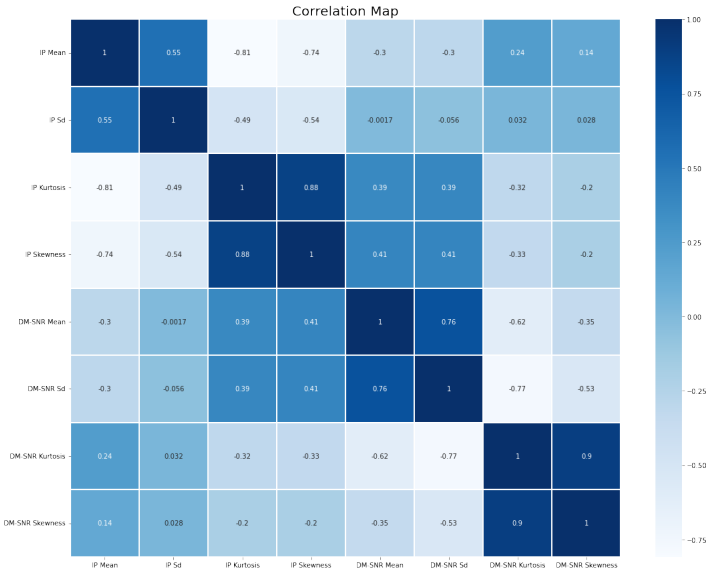


Figure 77: Correlation matrix of the attributes. One can easily see that there is significant correlation between some of the features in this dataset.

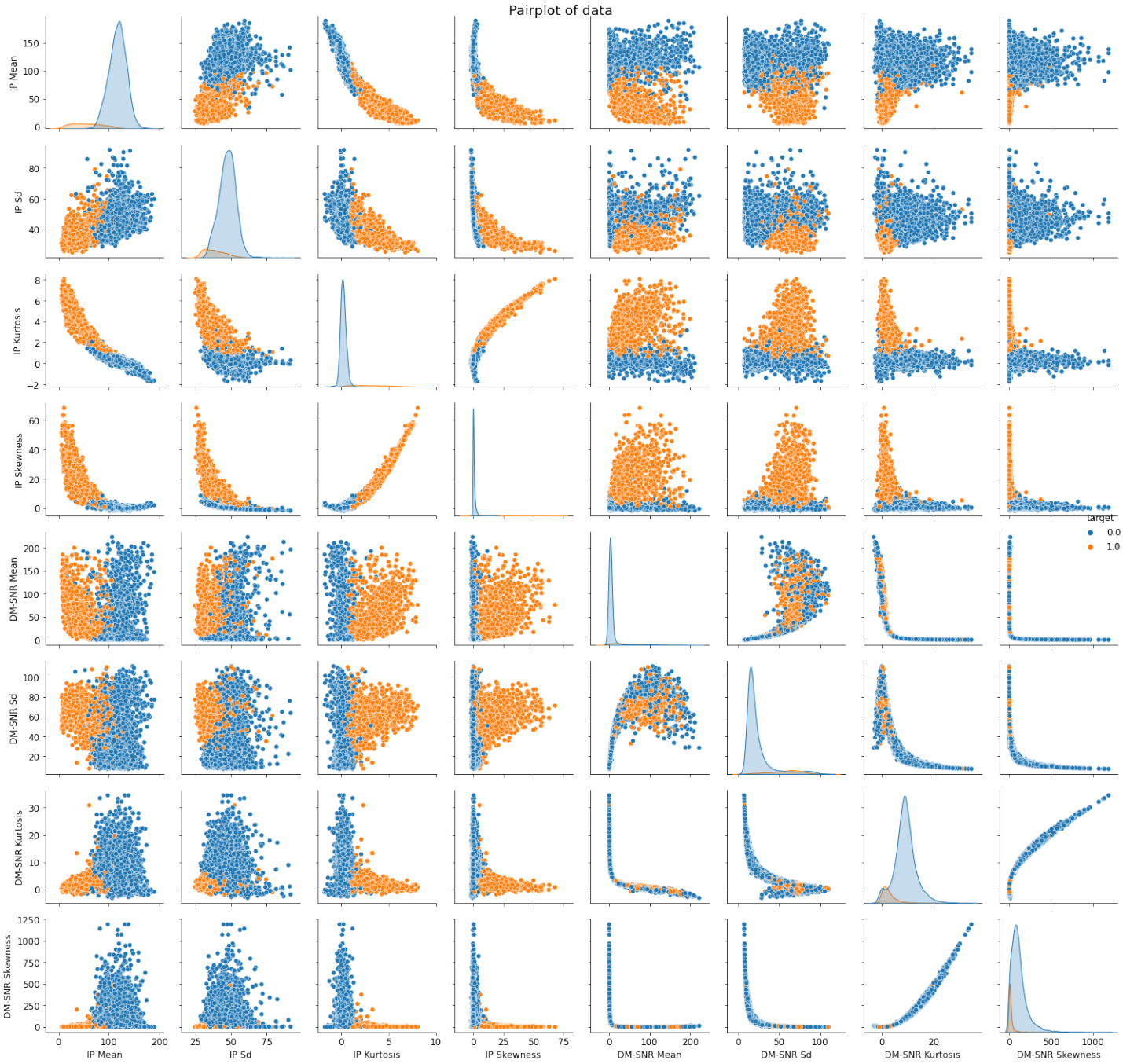


Figure 78: Pair plot of the attributes. One can easily see that there are some features in this dataset which one can easily use to distinguish between a regular star and a pulsar.

E. Support Vector Machine Model

Using scikit-learn, SVM models were built using 85% of the data. 15% of the data was used for testing / cross-validation. In this case, the models were trained with linear and RBF kernel and varying values of the hyperparameter C.

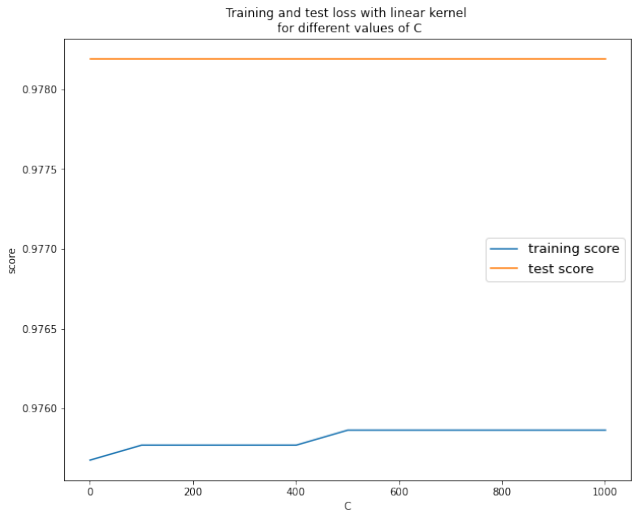


Figure 79: Training and test accuracy of SVMs trained on 85% of the data using a linear kernel and different values of C plotted against the value of C.

Figure 13 shows that the test data performance of the SVM model built using a linear kernel is not affected by the value of the hyperparameter C. The test data accuracy is 0.9782 and the highest training data accuracy is 0.9757.

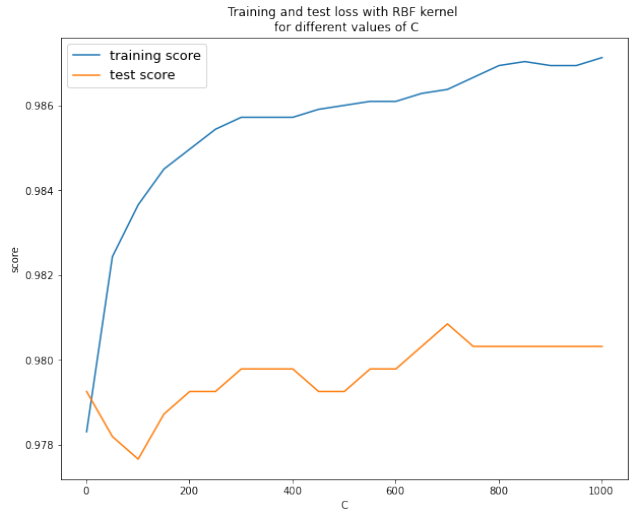


Figure 80: Training and test accuracy of SVMs trained on 85% of the data using the RBF kernel and different values of C plotted against the value of C. The kernel coefficient used for RBF is 1 / (n_features * X.var()) which is the default value.

Figure 14 shows that the test data performance of the SVM model built using the RBF kernel improves as the value of the hyperparameter C increases and reaches its highest value of 0.9808 at C = 701. The training data accuracy for C = 701 is 0.9863.

The final model used was the model built using RBF kernel with C = 701 and default kernel coefficient. Its performance metrics have been mentioned above. The training accuracy is very close to the test accuracy. This observation shows strong evidence that the model is **not overfitting** to the data. The search through the parameter space was widened, which was computationally expensive. However, no parameter values were found which resulted in a better accuracy than that of the aforementioned model.

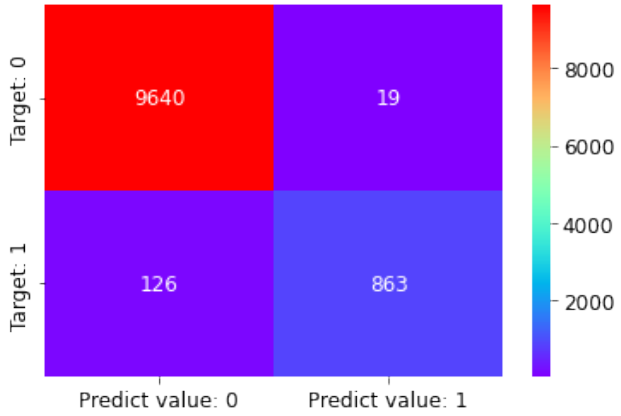


Figure 81: Confusion matrix of the final SVM model based on predictions of the training data set.

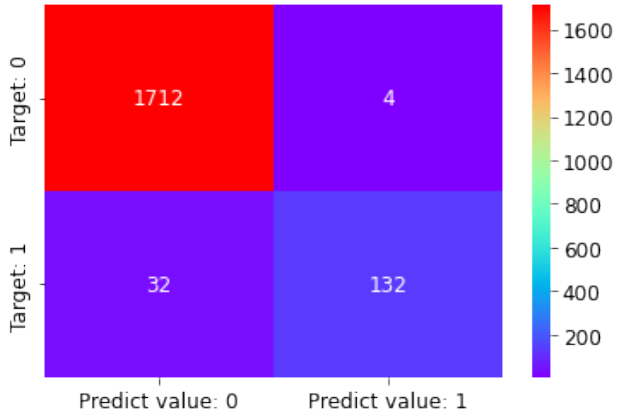


Figure 82: Confusion matrix of the final SVM model based on predictions of the test data set.

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	1716
1.0	0.97	0.80	0.88	164
accuracy			0.98	1880
macro avg	0.98	0.90	0.93	1880
weighted avg	0.98	0.98	0.98	1880

Figure 83: Classifier report for the final SVM model.

The null accuracy score is the baseline accuracy that can be achieved by always predicting the most frequent class. In this case, with the class imbalance, the null accuracy by always predicting class 0 is 91.28%. The final model's accuracy on the test data set is 98.08%, which is much higher than the null accuracy. This demonstartes that the SVM is able to find the optimal decision boundary.

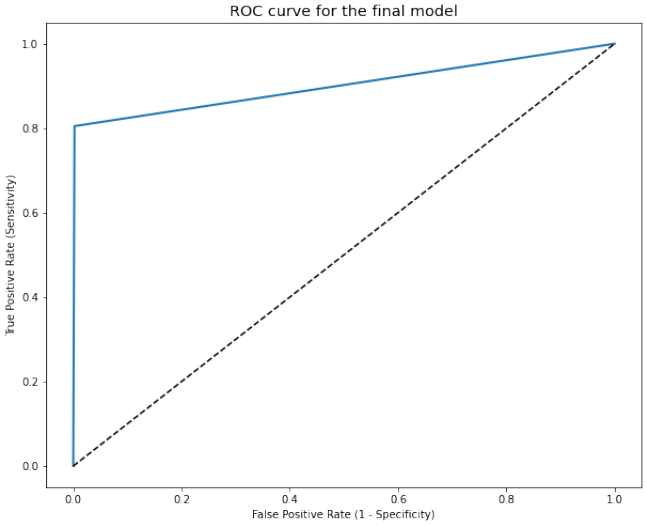


Figure 84: ROC curve for the final SVM model.

The Area Under the ROC Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. The area under the ROC curve for the final model is 0.9013, which means that the model is performing very well.

Conclusion

1) The Support Vector Machine's performance was successfully demonstrated, both visually and numerically, that the presence of pulsars can be effectively predicted using the statistics of the integrated pulse profile and DM-SNR curve.
2) The final SVM model was trained on 85% of the data using RBF kernel and C = 701. accuracy on the training data was **98.63%**. Accuracy on test data set was **98.08%**.
3) The close values of training and test accuracy, the classifier report, the confusion matrices, the accuracy above null accuracy and the ROC curve provide strong evidence that the model is accurately predicting the presence and absence of pulsars without overfitting.

VII. Long Short Term Memory (LSTM) networks

Introduction

This section seeks to explore one of the important deep learning algorithm used for classification and regression - **Long Short Term Memory (LSTM) networks**. Once we have acquired data with multiple features, one important task is to understand how the variables are related to a target variable. When the data is time-series or sequential, then there is information inside the sequential structure of the data. To learn from such data, one must learn in such a way that allows the learned information to persist. LSTMs are a special type of Recurrent Neural Networks (RNNs). RNNs use the previous information to process the current input. This process occurs because RNNs have a loop structure where they take the present input and the recent past to learn about the future.

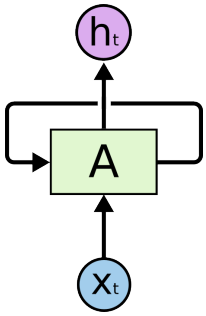


Figure 85: A diagram illustrating the loop structure in RNNs. **source**

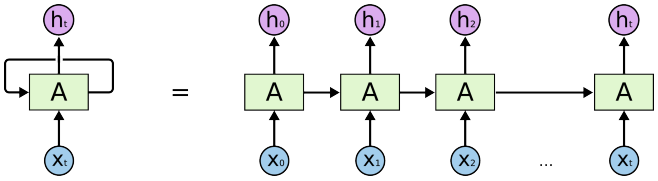


Figure 86: A diagram illustrating the unrolling of the loop structure in RNNs. **source**

The loop structure results in RNNs having a strong advantage over traditional feed-forwards neural networks (FFNs) because of the presence of memory. FFNs will not remember the first data point when processing the fifth data point and this is an ineffective learning paradigm when it comes to sequential data.

To demonstrate the effectiveness of LSTMs, we use time-series data of various stock prices and the USD-INR exchange rate to see if LSTMs can predict the prices / exchange rates effectively. Before performing model building and prediction using the LSTM, we visualise the data and perform feature engineering.

LSTMs

As mentioned in the introduction section, RNNs have a loop structure that allows them to store and use recent information when encountering new information. However, this information storage for standard RNNs is low and they cannot store long-term dependencies. For certain problems, the long-term memory is not needed to effectively solve / predict data. But in the case of our financial dataset, LSTMs are required.

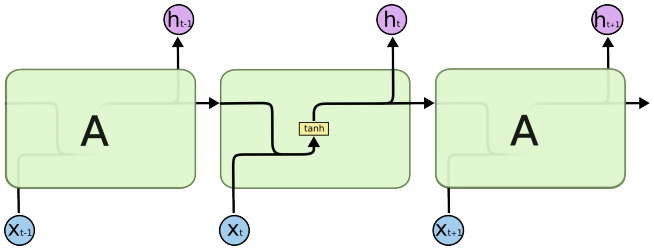


Figure 87: A diagram illustrating the repeating module in RNNs. **source**

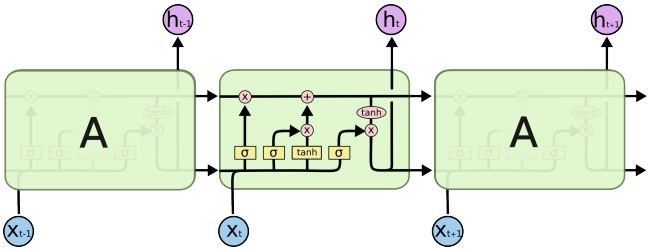
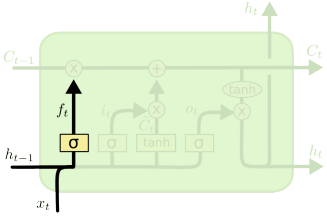


Figure 88: A diagram illustrating the repeating module in LSTMs. **source**

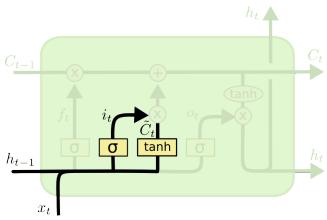
RNNs have a simple repeating module with a tanh activation function. LSTMs have a more sophisticated repeating module with multiple gates. The horizontal line at the top of the LSTM gates is key. It is the cell state to which information is added or removed by the gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

Here are the diagrams that describe the operation in LSTMs:



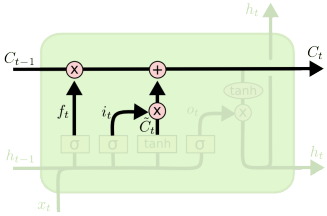
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 89: **source**



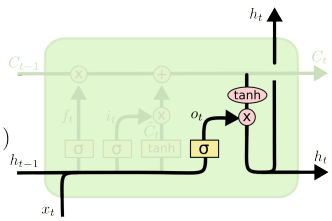
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 90: **source**



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 91: **source**



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 92: **source**

CAN LSTMS BE USED TO PREDICT THE FUTURE STOCK PRICES AND USD-INR EXCHANGE RATES?

In this section, we see the predictions revealed by the LSTM using past data. We want to see whether it can be used to accurately predict the future stock prices and USD-INR exchange rates. The list of features in the dataset are (for each day):

1) Date
2) Opening price of a stock
3) High price of a stock
4) Low price of a stock
5) Closing price of a stock
6) Volume of stock traded
7) Date, Open, High, Low, Close and Adjusted Closing price for exchange rates.

A. Visualisation and Imputation of the dataset

There were several missing data points and hence imputation was required. HCL and Infosys were missing 2 data points. HDFC, SBI and ICICI were missing 2 data points. The USD-INR exchange rate was missing 21 data points. Imputation using the mean would not be ideal because this is sequential data. The missing values were imputed using the mean of the 2 nearest neighbor points.

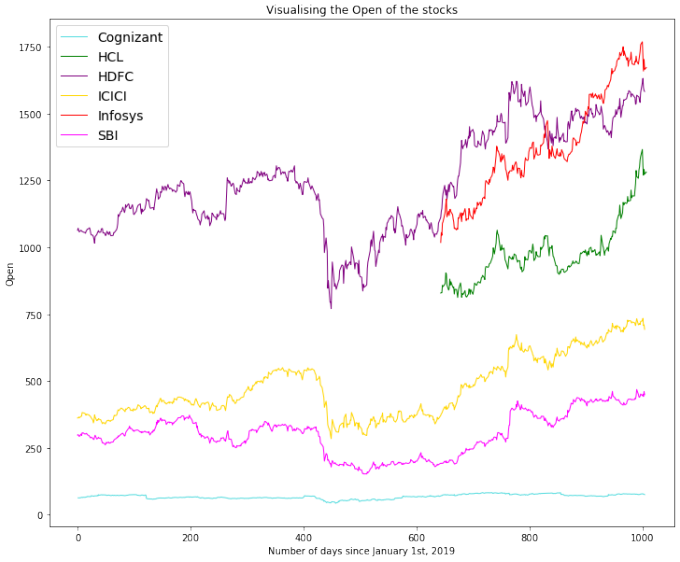


Figure 93

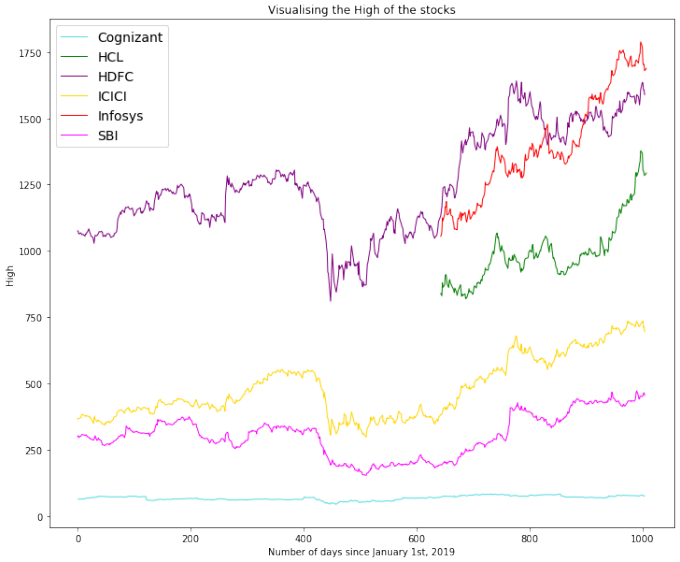


Figure 94

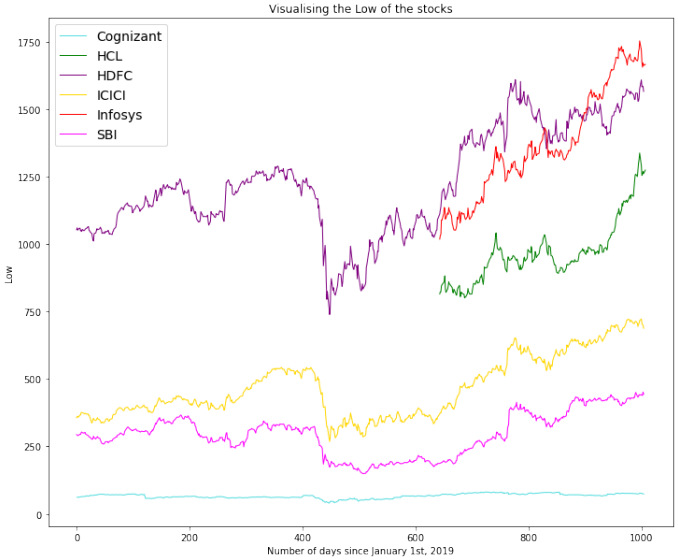


Figure 95

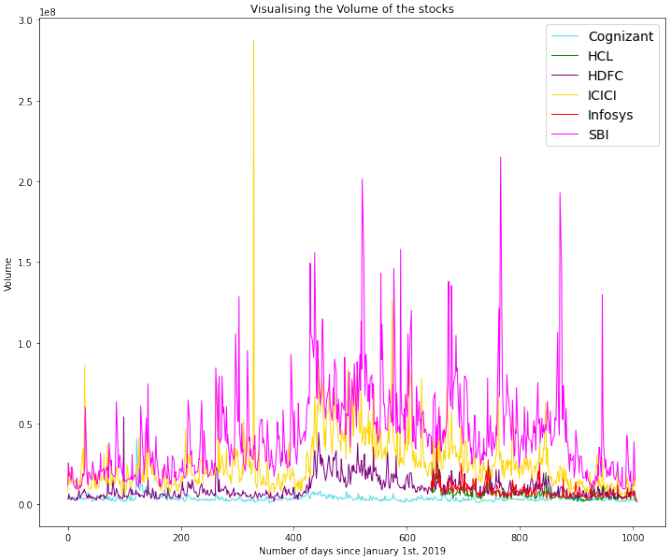


Figure 97

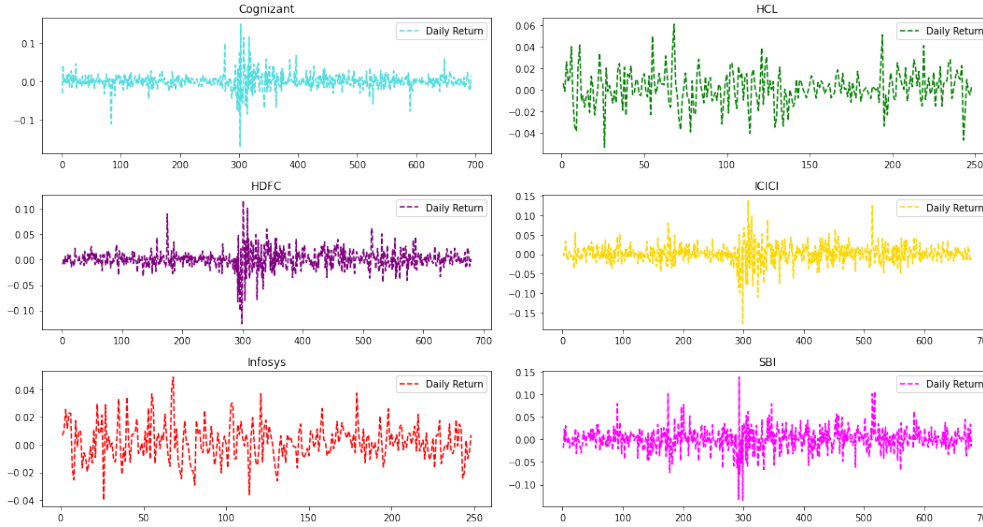


Figure 98: Daily returns

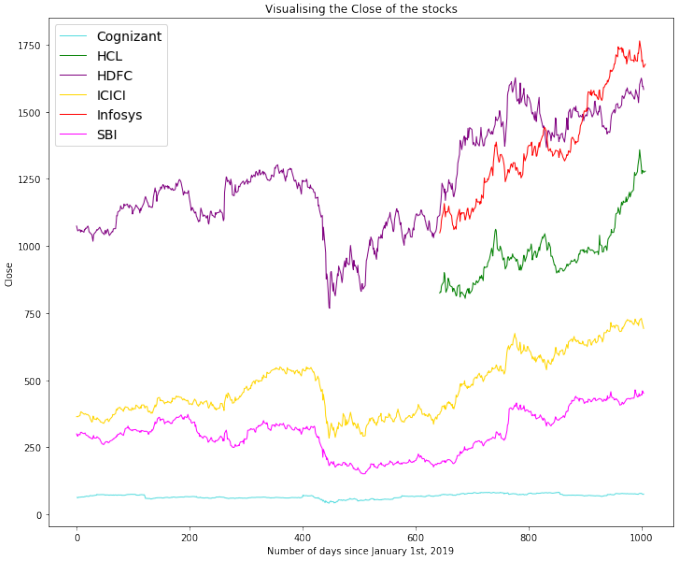


Figure 96

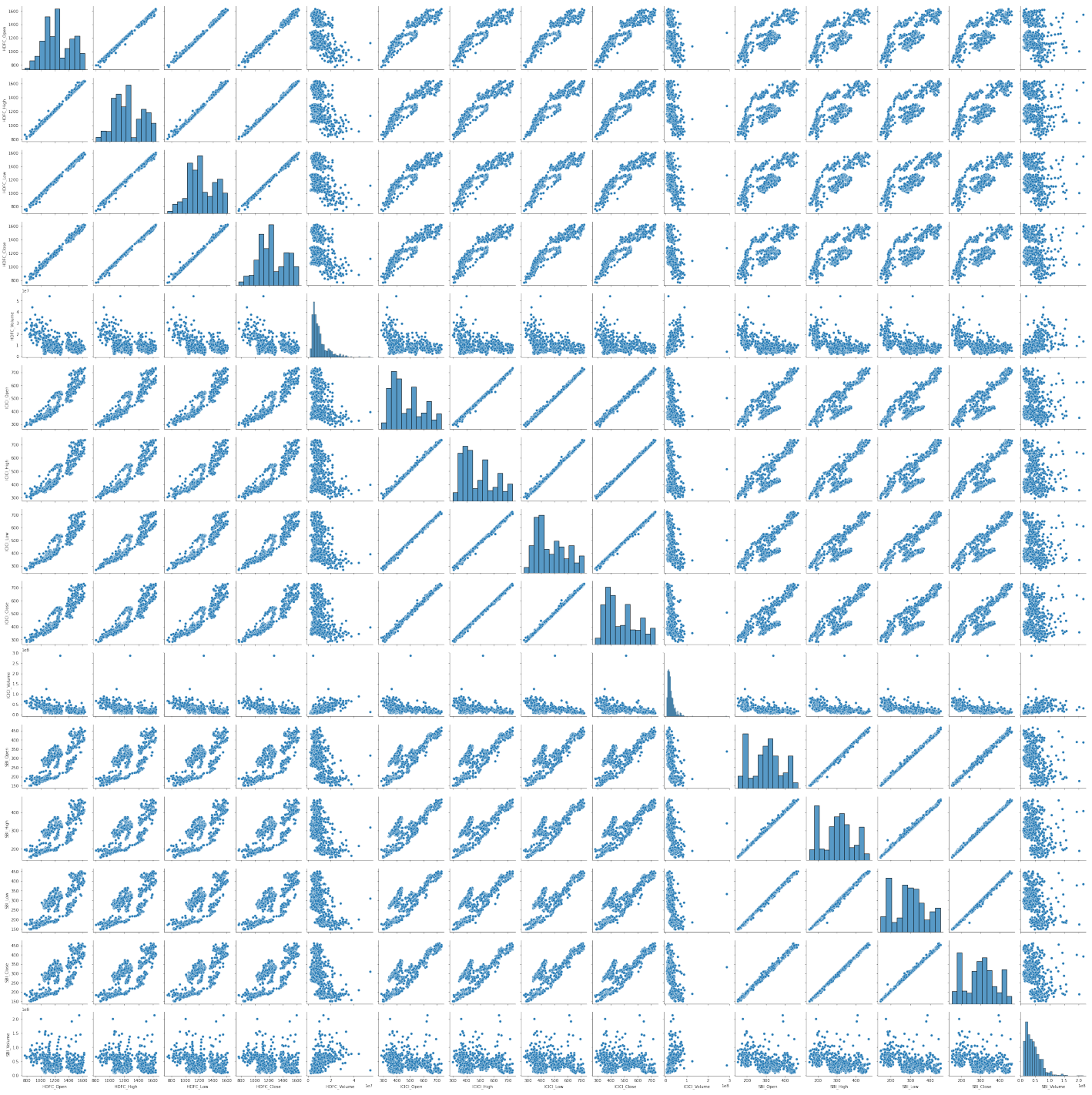


Figure 99: Pair plot of the attributes for each stock.

B. LSTM Model

The data was first scaled using min=max scaler. Using Keras, LSTM models were built such that the previous 60 data points were used as training and the 61st data point was predicted and used as test data. The LSTM had 128 modules followed by 64 modules followed by a 2 layer FFN (25 units and 5 units). The solver was Adam and the loss was the mean squared error. After 3 epochs, here is the performance on stocks.

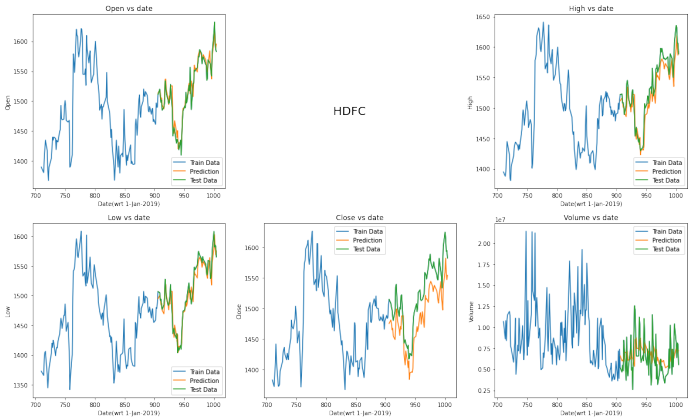


Figure 102: Prediction for HDFC.

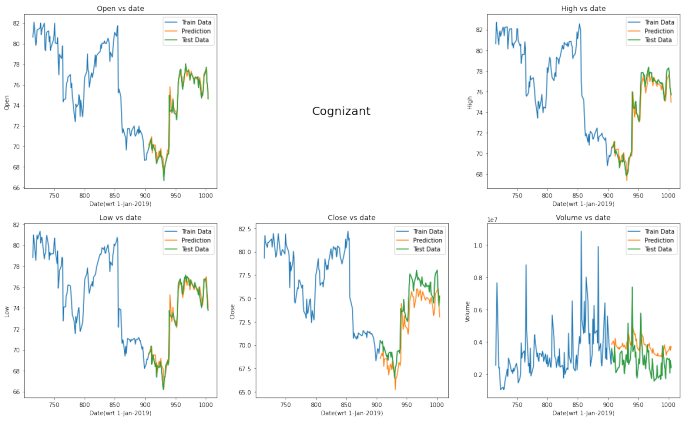


Figure 100: Prediction for Cognizant.

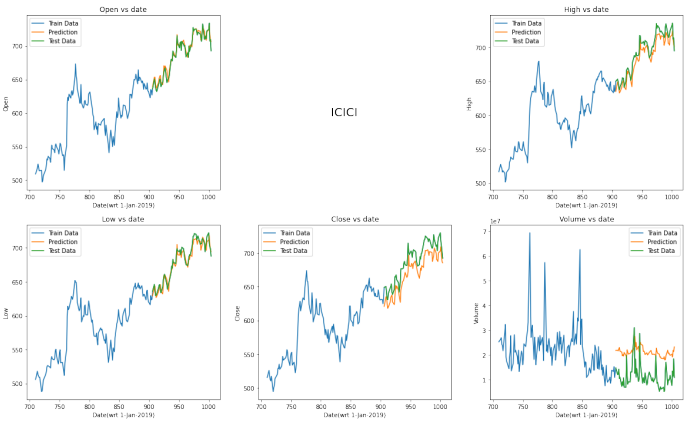


Figure 103: Prediction for ICICI.

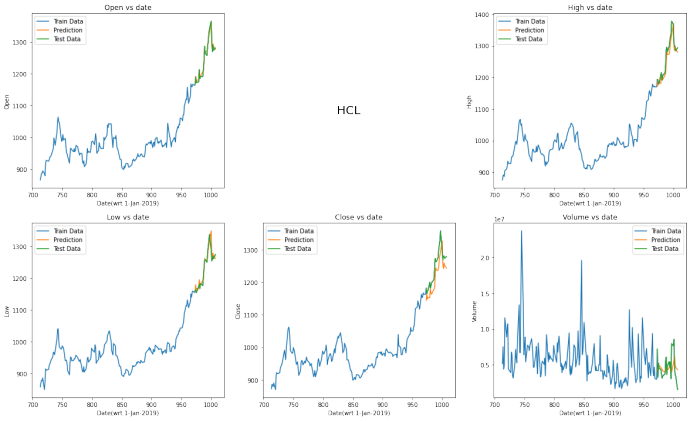


Figure 101: Prediction for HCL.

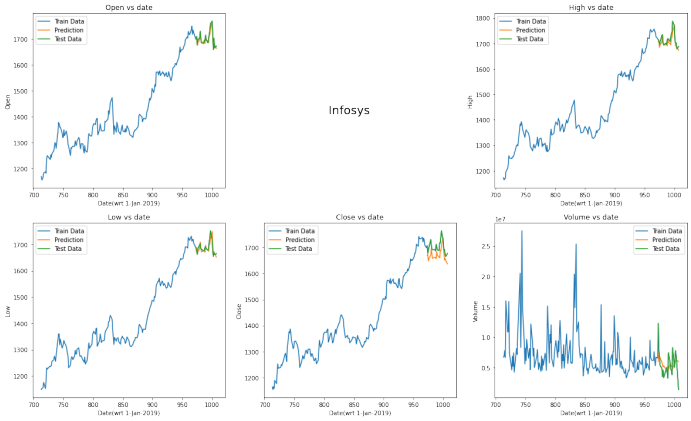


Figure 104: Prediction for Infosys.

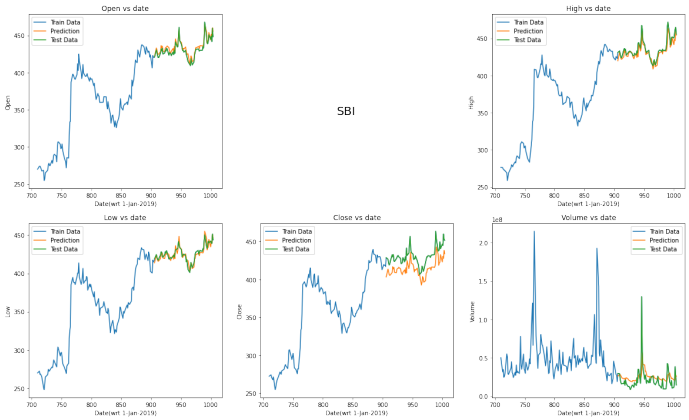


Figure 105: Prediction for SBI.

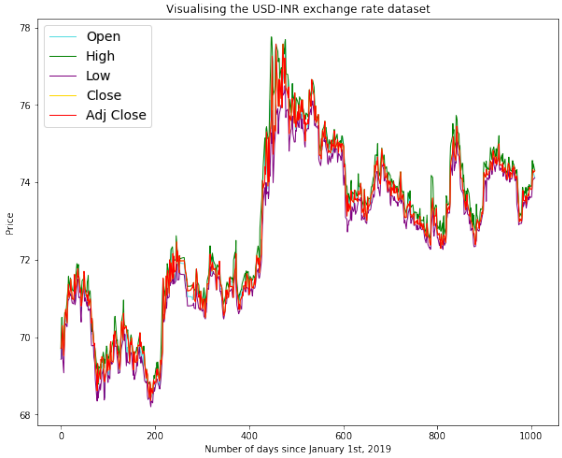


Figure 106: Visualising the USD-INR exchange rate dataset.

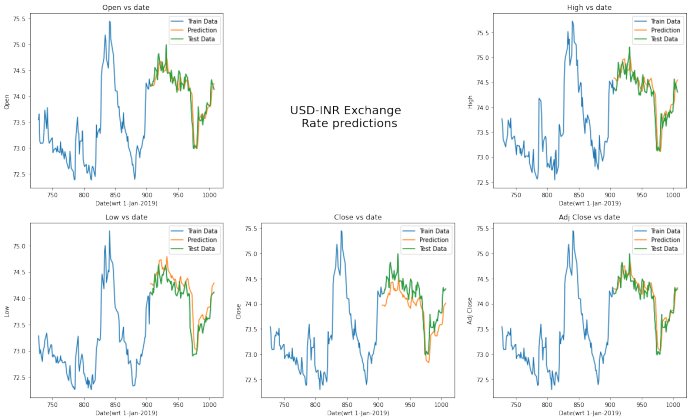


Figure 107: Prediction for the USD-INR exchange rate.

Clearly, the LSTM model is performing well in terms of predicting future values of the stock prices and exchange rates. Both the LSTMs were trained for 3 epochs and they reached a loss of 0.0023 in both cases. Training for further epochs did not result in an improvement as shown by the loss function.

Conclusion

1) The LSTM is a modified RNN with long-term memory which is able to keep track of the important dependancies in the financial data. This model is able to effectively predict the future prices of stocks and USD-INR exchange rate.
2) The LSTM model after 3 epochs had a loss of 0.0023.

References

[1] "State Cancer Profiles", Statecancerprofiles.cancer.gov, 2021. [Online]. Available: **Link**.
[2] HMS Titanic Passenger Data obtained from **Link**.
[3] 1994 Census Bureau provided by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics). **Link**.
[4] Car Evaluation Data Set provided by Marko Bohanec and Blaz Zupan. **Link**.