Data Analytics Laboratory Final Exam

Aryan Pandey
Department of Ocean Engineering
Indian Institute of Technology, Madras
Chennai, India
na19b030@smail.iitm.ac.in

***Abstract*—A stock exchange is a marketplace where consumers can purchase and sell shares in publicly traded corporations. It provides a platform that allows for smooth stock exchange. The purchasing and selling of shares takes place over the internet. Different forecasting software and methods were utilised in this paper to forecast the stock market price of various organisations. From 2019 through 2021, we examine the stock markets of Cognizant, HDFC, HCL, Infosys, SBI, and ICICI. We shall use kNN and neural networks to tackle this problem set**

***Index Terms*—Stock Exchange, forecasting**

I. Introduction

A stock is a financial instrument that reflects a proportionate claim on a company's assets and earnings. Stocks are sometimes known as shares or equity in a corporation. Ours is an economic time marked by unpredictability. In a few years, a company that is making a lot of money today might not even be in the running. Because of the huge returns it promises, the stock market is currently attracting the most attention from all imaginable sectors. Trading has been dominated by computers in recent years, and algorithms are responsible for determining split-second trading decisions. The KNN model can be used to solve both classification and regression problems. Its most common application is to solve categorization difficulties. The model employs 'feature similarity' to forecast the values of new data points, assigning a value to a new point based on how closely it resembles the points in the training set. It's a single layer neural network called a feed forward neural network. There is only one layer of input nodes in this single hidden layer form that sends weighted inputs to a later layer of receiving nodes. Using lagged values of the time series as input data, the function model technique leads to a non-linear autoregressive model. From 2019 to 2021, the shares of several firms such as HCL, Infosys, ICICI, SBI, Cognizant, and HDFC have been given. A company's share is very essential. The goal is to forecast these firms' future share/stock prices. The theory of kNN regression and Neural Networks, as well as the mathematics underpinning these methods, are covered in this paper. Different companies' share data has been provided in CSV format. We compare the returns on these firms' shares and forecast future share prices using several models

II. Datasets

Date, highest/lowest value on that day, opening/closing value on that day, and volume of 6 separate firms make up the share data. Cognizant, HCL, HDFC, ICICI, Infosys, and SBI are among these firms. ICICI Bank, Cognizant. All other firms have data from May 2020 to 2021, except HDFC and SBI, which have data from 2019 to 2021. The goal is to forecast future share value forecasts for these companies. For projections, the values of the columns "Date" and "Close" were used. For visualisations, a new column called "return" is added to log(Close share value on current day/Close share value on previous day). Except for the conscious dataset, each dataset has one null row deleted. For visualisations, the columns "Close," "Volume," and "Return" are used.

Fig. 1. Cognizant Close Share Value

The stock market and the log return, as expected, are extremely volatile. The first wave of COVID-19 is likely to blame for a significant drop in share prices between March and May 2020 (Fig 2). Share prices fell during the second wave of COVID-19, from March to May 2021, but the effect was far smaller than the first wave. Aside from that, the share prices haven't changed significantly. In general, stock prices appear to be rising. The number of shares traded (Fig 3) does not appear to have changed significantly. There was a lot of dealing and exchange going on in May of this year. Cognizant's log return value is depicted in Figure 4 below. The return on investment is generally good. In the month of May 2021, there was a significant loss.

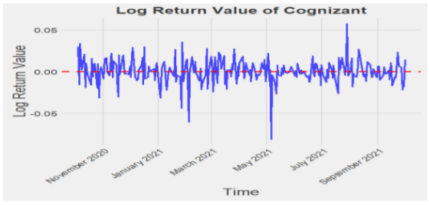


Fig. 2. Cognizant Log Return

The share prices of HCL (Fig 5) are on the rise. During the time span under consideration, no significant decreases were seen. There was little change during COVID-19 waves as well. There has been a slight drop in the amount of shares

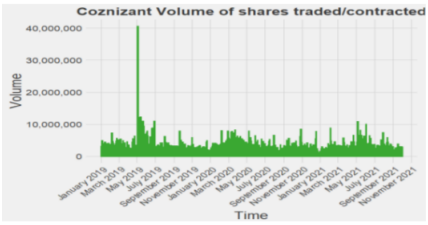


Fig. 3. Cognizant Volume

Fig. 4. HCL Close Share Value

are traded/exchanged on a regular basis. Figure 7 displays the log return HCL's worth. In general, the return value is positive. Between January 2021 and May 2021, there was a significant negative return period. The share prices of HDFC (Fig 8) are also on the rise, however there was a significant drop from March to May 2020, most likely owing to COVID-19. Because to COVID-19, the number of shares exchanged (Fig 9) increased dramatically from March to July 2020. Prices fell, and individuals acquired more shares. During the period March 2021-May 2021, the log return value (Fig 10) tends to be positive and is extremely volatile. The value of ICICI's shares (Fig 11) likewise exhibits an upward trajectory, with a significant drop in prices during COVID-19's initial wave. During the second wave of COVID- 19, there is also a slight decrease (March 2021 - May 2021 period). The share exchange (Fig 12) remained nearly stable, although there was a significant surge in December 2019. In comparison to other corporations, ICICI's log return value (Fig 13) is less variable over time. The share prices of Infosys (Fig 14) are similarly rising, with no significant increases or decreases in sight. Currently, the number of share exchanges (Fig 15) appears to be de- creasing. In the graph, there were several unexpected surges in exchanges. In November 2020, there was a significant increase. The volatility of infosys' log return value (Fig 16) appears to have decreased recently.

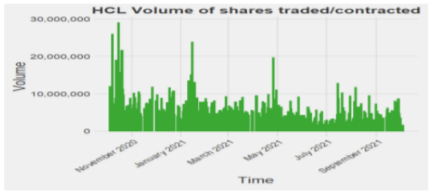


Fig. 5. HCL Volume

Fig. 6. HCL Log Return

Fig. 7. HDFC Close Share Value

The value of SBI's share prices (Fig 17) is currently increasing. During the period of March 2020 to May 2020, there was a significant drop in value, as well as a slight surge in January 2021. In the period March 2020 to December 2020, there was a significant increase in share exchanges (Fig 18), which thereafter declined, but abrupt spikes in exchanges are common. SBI's log return value (Fig 19) does not appear to be very volatile. Overall, all companies' share prices are rising, with a significant drop in prices between March and May 2020, most likely due to COVID-19. There was a fall in pricing during the second wave of COVID-19 (March 2021- June 2021), but it was significantly smaller than the first decline. Every company's log return value was erratic, with some being less volatile and others being quite volatile. In comparison to service organisations, banking institutions like as ICICI, HDFC, and SBI have higher share exchanges.

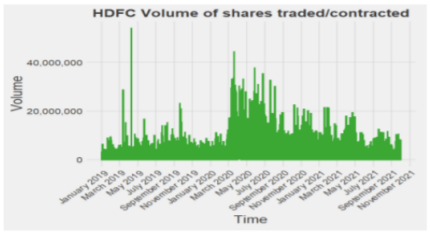


Fig. 8. HDFC Volume

III. Models

Long short term memory in an artificial recurrent neural network (RNN) is an architecture of deep learning. Unlike any feedforward neural network, LSTM has feedback connections. Therefore, it can predict values for point data and can predict sequential data like weather, stock market data, or work with

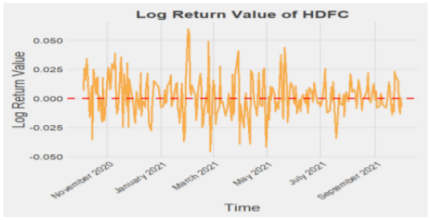


Fig. 9. HDFC Log Return

audio or video data, which is considered sequential data. A most common LSTM network unit consists of a cell, an input gate, an output gate, and a forget gate. A cell remembers values over an autocratic time interval. The input gate manages the flow of information coming inside of the cell. The output gate manages the flow of information going outside. Similarly, forget gates manage the flow of information that is not required or not required. LSTM are useful for making predictions, classification and processing sequential data. We use many kinds of LSTM for different purposes or for different specific types of time series forecasting. The input vector at time t is connected to the LSTM cell of time t by a weight matrix U, the LSTM cell is connected to the the LSTM cell of time t-1 and t+1 by a weight matrix W, and the the LSTM cell is connected to the output vector of time t by a weight matrix V. The matrices W and U are divided in submatrices $(W_f, W_i, W_g, W_o; U_f, U_i, U_g, U_o)$ that are connected to different elements of the LSTM unit, as shown in the Figure below. All the weight matrices are shared across time. The cell state transfers the relevant information during processing, so that also the information from the previous time steps arrives at each time step, reducing the effects of shortterm memory. During training over all the time steps, the gates learn which information is important to keep or to forget, and add them to the cell state, or remove them from it. In this way LSTM allows the recovery of data transferred in memory, solving the vanishing gradient problem. LSTM are useful for classifying, processing, and predicting time series with time lags of unknown duration.

IV. Modelling

Here, I aim to predict the stock price using the previous 60 values. So, to prepare the training dataset, I append the last 60 values of closing price of HDFC stock as the training data, and the current stock price as the test data. These two lists are converted to numpy arrays. The model has 2 LSTM layers having 128 and 64 neurons in the hidden layer respectively. Then it is followed by a fully connected layer having 25 neurons and the output layer. This is optimized by Adam optimizer and trained for 1 epoch with batch size = 1. The mean squared error is the loss function which is to be minimised. Similar preprocessing is done on the test set as well to predict the values. The obtained test RMSE is 22.89, which is pretty good considering the fact that the stock price varies in the range of 1000's.

V. Conclusion

This is my first time working on time series data and I had a good learning experience. The pandemic had a devastating effect on the life and livelihood of the people and the effect is also seen in the stock prices, as most of the stocks reduced to unprecedented levels. The volume traded reduces when the stock reduces, gets a massive boost when the stock prices begin to increase and saturate thereafter. Sophisticated deep learning models such as LSTM does a great job in predicting stock prices, even when there is such a wide variability.

A Mathematical Essay on Support Vector Machines

Aryan Pandey
Department of Ocean Engineering
Indian Institute of Technology, Madras
Chennai, India
na19b030@smail.iitm.ac.in

***Abstract*—This document is an overview of the mathematical aspects of Support Vector Machine as well as its application on a sample data set. The algorithm has been applied on a data set of stars and the prediction task is to predict if a star is a pulsar start or not - which is a rare type of Neutron star that produces radio emissions detectable here on Earth**

***Index Terms*—Support Vector Machine, Pulsar**

I. Introduction

SVMs (short for Support Vector Machines) are machine learning algorithms that are used for classification and regression. SVMs are a type of machine learning method that can be used for classification, regression, and outlier detection. SVM classifiers create a model that allocates new data points to one of the predetermined categories. As a result, it can be considered a binary linear non-probabilistic classifier. In 1963, Vladimir N Vapnik and Alexey Ya. Chervonenkis created the first SVM algorithm. The algorithm was still in its early stages at the time. Drawing hyperplanes for linear classifiers is the sole option. Bernhard E. Boser, Isabelle M Guyon, and Vladimir N Vapnik proposed using the kernel method on maximum-margin hyperplanes to generate non-linear classifiers in 1992. Corinna Cortes and Vapnik proposed the current standard in 1993, and it was published in 1995. In this paper, the aim is to apply Support Vector Machines algorithm to predict if a star is a pulsar start or not. The paper systematically goes through first the mathematical details of SVM, the nature of the data set which has been given to us, then the problem we have in hand and how it has been solved, and finally the conclusions which were drawn. Useful insights and figures have been presented whenever necessary.

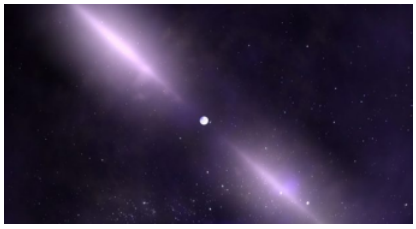


Fig. 1. Pulse Star

II. Datasets

The problem asks us to predict if a star is a pulsar start or not. This section walks you through the entire process followed to make the predictions

A. Outline

We will follow the standard procedure followed to build any Machine Learning model. We will first analyze and visualize the data. We will then try to figure out which features are to be kept intact and which features are to be dropped. Then we will convert the features into the form which the algorithm understands. Then we will split the train data into train/test sub-parts in order to figure out the most optimum parameters. And then finally we will fit the original train and test data to produce the predictions.

B. Data Visualization

All of the features appear to be continuous, which is to be expected.

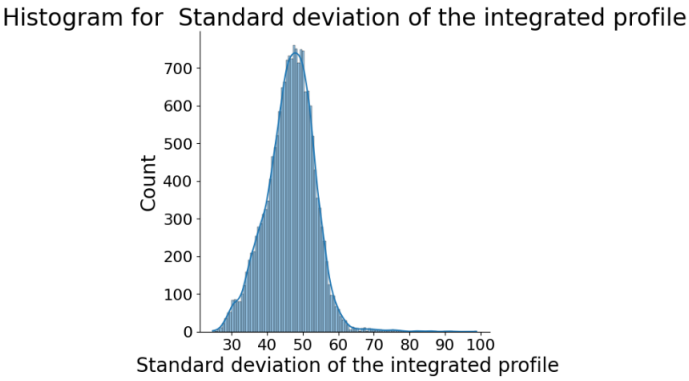


Fig. 2. Histogram for the Standard Deviation of the Integrated Profile

Most features appear to be skewed to some degree, with some containing a large number of outliers. It's also worth noting that the variables are on various scales, which should be taken into account if we're going to employ regularisation techniques (hint: we won't). It's also worth noting that some characteristics are related to one another; for example, both kurtosis and skewness are functions of a distribution's mean and variance, and their functional definitions are extremely similar.

A pair plot of the combined profile features is shown in Fig.3. The mean and kurtosis, as well as the mean and skewness, have decreasing non-linear connections. The link between standard deviation and kurtosis, as well as standard deviation and skewness, is similar, however the patterns are slightly noisier.

In addition, skewnees and kurtosis have a definite link. Let us summarize the dataset all at once:

- There are 9 numerical variables in the dataset
- 8 are continuous variables and 1 is discrete variable
- The discrete variable is targetclass variable. It is also the target variable

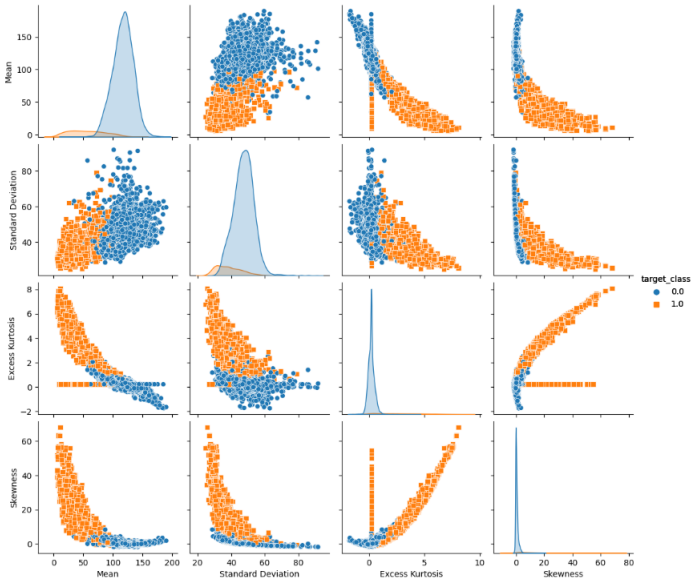


Fig. 3. Pair Plot for features of the integrated Profile separated by target class

On closer inspection, we can suspect that all the continuous variables may contain outliers.

III. Models

This section discusses the mathematical and conceptual aspects of the SVM algorithm.

A. Intuition

Now, we should be familiar with some SVM terminology

- Hyperplane - A hyperplane is a decision boundary that divides a set of data points with differing class labels into two groups. The SVM classifier uses a hyperplane with the most margin to separate data points. The maximum margin hyperplane and the linear classifier it specifies are known as the maximum margin hyperplane and maximum margin classifier, respectively.
- Support Vectors - The sample data points nearest to the hyperplane are called support vectors. By calculating margins, these data points will better define the separation line or hyperplane.
- Margin - A margin is the distance between the two lines on the data points that are closest to each other. It is determined as the perpendicular distance between the line and the nearest data points or support vectors. We strive to maximise this separation distance in SVMs to get the most margin.

The graphic below visually depicts these notions

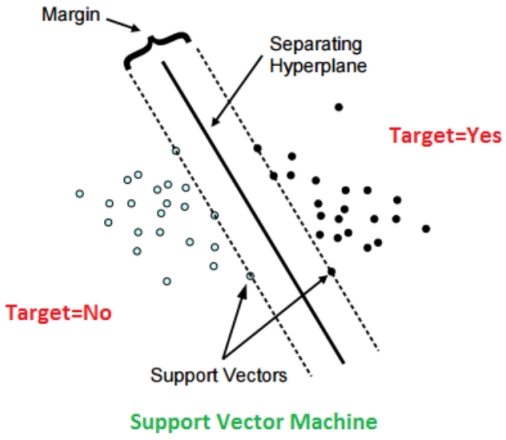


Fig. 4. Margins in a Support Vector Machine

The primary goal of SVMs is to find a hyperplane with the greatest feasible margin between support vectors in a given dataset. In the following two-step method, SVM looks for the largest margin hyperplane

- Create hyperplanes that separate the classes as much as possible. There are numerous hyperplanes that might be used to categorise the data. The optimal hyperplane that reflects the widest separation, or margin, between the two classes should be chosen.
- As a result, we choose the hyperplane with the greatest distance between it and the support vectors on each side. If such a hyperplane exists, it is referred to as a maximum margin hyperplane, and the linear classifier it defines is also referred to as a maximum margin classifier.

The diagram below clearly explains the concepts of maximum margin and maximum margin hyperplane

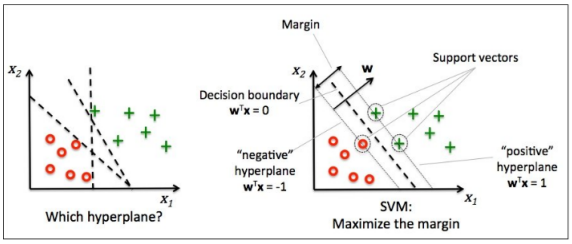


Fig. 5. Maximum Margin Hyperplane

B. The Algorithm

Let's say we have some data and we (the SVM algorithm) are asked to distinguish between males and females by first analysing the features of both genders and then appropriately labelling the unseen data as male or female. In this case, the qualities that would aid in gender differentiation are referred to as features in machine learning. We comprehend x's domain

when we define it in real space, and we receive range and co-domain when we map a function for y = f(x). As a result, we are provided the data to be split by the algorithm at the start. The data to be separated/classified is represented as a single point in space, with each point being represented by a feature vector x.

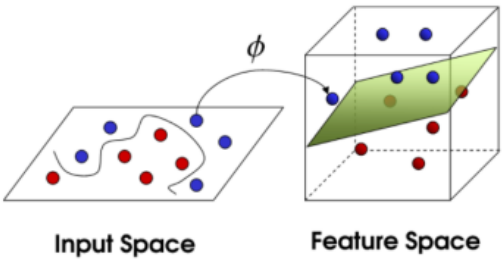


Fig. 6. Input Space gets mapped to the Feature Space

Further, mapping the point on a complex feature space x. The transformed feature space for each input feature mapped to a transformed basis vector (x) can be defined as $\phi(x) = R^D \rightarrow R^M$ Now that we've physically depicted our points, the next step is to divide them with a line, which is where the phrase "decision boundary" comes into play. Decision The key separator for splitting the points into their different classes is the boundary. The equation of the main separator line is called a hyperplane equation.

The hyperplane equation dividing the points (for classifying) can now easily be written as $H : w^T(x) + b = 0$

Here: b = Intercept and bias term of the hyperplane equation In D dimensional space, the hyperplane would always be D -1 operator. Now that we've seen how to represent data points and how to fit a separating line between them, let's look at how to fit a separating line between them. However, while fitting the dividing line, we obviously want one that can segregate the data points in the best feasible way with the fewest mistakes/misclassification errors. So, in order to have the fewest errors in data point categorization, we must first determine the distance between a data point and the separating line.

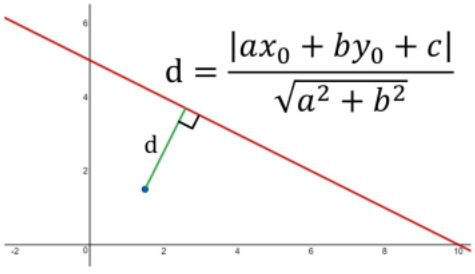


Fig. 7. Distance Measure from the Hyperplane

The distance of any line, ax + by + c = 0 from a given point say, (x0 , y0) is given by d. Similarly, the distance of a hyperplane equation: $w^T(x)+b = 0$ from a given point vector (x0) can be easily written as:

$$d_H(\phi(x_0)) = \frac{|w^T(\phi(x_0)) + b|}{||w||_2}$$

Fig. 8. Distance of a Hyperplane from a point

Here $||w||_2$ is the Euclidean norm for the length of w Finding a hyperplane with the greatest margin (margin is a protected space around the hyperplane equation) and attempting to have the greatest margin with the fewest points (known as support vectors). "The goal is to maximise the minimum distance," to put it another way. If the point from the positive group is substituted in the hyperplane equation while generating predictions on the training data that was binary classified as positive and negative groups, we will get a value larger than 0. (zero) The product of a predicted and actual label would be greater than 0 (zero) on correct prediction, otherwise less than zero. For perfectly separable datasets, the optimal hyperplane classifies all the points correctly, further substituting the optimal values in the weight equation.

C. Types of Kernels

Linear Kernel - When the data is linearly separable, a linear kernel is utilised. It indicates that data can be split with only one line of code. It is one of the most often utilised kernels. It is most commonly utilised when a dataset contains a significant number of features. The linear kernel is frequently used for text classification.

Polynomial Kernel - The similarity of vectors (training samples) in a feature space over polynomials of the original variables is represented by a polynomial kernel. To estimate their similarity, the polynomial kernel examines not only the supplied attributes of input samples, but also combinations of input samples.

RBF Kernel - Radial basis function kernel is a general purpose kernel. It is used when we have no prior knowledge about the data. Fig 19 visualizes the RBF Kernel.

Sigmoid Kernel - Sigmoid kernel has its origin in neural networks. We can use it as the proxy for neural networks. Fig 20 visualized the Sigmoid Kernel

IV. Modelling

After running the SVM model with all the 4 kernels mentioned before, we find that at C=100.0, we get the best accuracy with rbf and linear kernel, and the accuracy is 0.9832. Based on the aforementioned study, we can infer that the accuracy of our classification model is excellent. In terms of predicting class labels, our model performs admirably. However, this is not the case. We have an unbalanced dataset here. The issue is that in the imbalanced dataset scenario,

accuracy is an insufficient metric for assessing predictive performance. As a result, we must look into alternative metrics that can help us choose better models. We'd like to know the underlying distribution of values as well as the kind of errors our classifier makes. Confusion matrix is one such statistic for analysing model performance in imbalanced classes problems. Here is the confusion matrix for this case

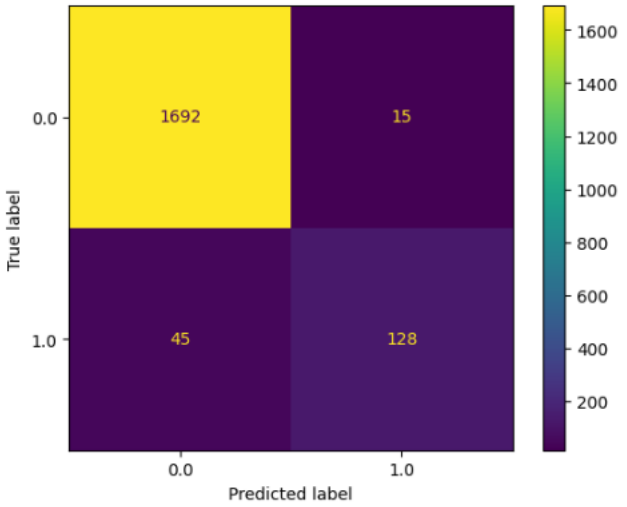


Fig. 9. Distance of a Hyperplane from a point

We can also perform hyperparameter tuning using grid-searchCV. Our original model test accuracy is 0.9832 while GridSearch CV score on test-set is 0.9835. So, GridSearch CV helps to identify the parameters that will improve the performance for this particular model.

V. Conclusion

- Our dataset contains outliers. As I increased the value of C to reduce the number of outliers, the accuracy improved. This is true for several types of kernels
- With C=100.0 and rbf and linear kernel, we get the highest accuracy of 0.9832. As a result, we may conclude that our model does an excellent job at predicting class labels. However, this is not the case. We have an unbalanced dataset here. In the imbalanced dataset problem, accuracy is an insufficient metric for assessing predictive performance. As a result, we must investigate confusion matrices that can help us choose better models
- Our model's ROC AUC is extremely close to 1. As a result, we can conclude that our classifier accurately classifies the pulsar star
- With the linear kernel, I get a better average stratified k-fold cross-validation score of 0.9789, but the model accuracy is 0.9832. As a result, the stratified cross-validation strategy is ineffective in improving model performance.
- The accuracy of our original model test is 0.9832, whereas the GridSearch CV score on the test-set is 0.9835. As a result, GridSearch CV assists in identifying the parameters that will increase the model's performance.

A Mathematical Essay on Random Forests

Aryan Pandey
Department of Ocean Engineering
Indian Institute of Technology, Madras
Chennai, India
na19b030@smail.iitm.ac.in

***Abstract*—This document is an overview of the mathematical aspects of Random Forests as well as its application on a sample data set. The algorithm has been applied on a data set of cars and the prediction task is to classify a car based on its safety**

I. Introduction

Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

In this paper, the aim is to apply Decision Tree to classify a car based on its safety. The paper systematically goes through first the mathematical details of Decision Tree, the nature of the data set which has been given to us, then the problem we have in hand and how it has been solved, and finally the conclusions which were drawn. Useful insights and figures have been presented whenever necessary.

II. Datasets

The dataset given has the following details of multiple cars with the aim of classifying it based on its safety. It has the details of buying price, price of maintenance, number of doors, seating capacity, size of luggage boot and the estimated safety of the car.

The dataset consists of purely categorical columns in which the columns related to buying price, price of maintenance, and estimated safety of the car are categorised into very high, high, medium and low. The target condition of the car has 4 categories which are very good, good, acceptable and unacceptable.

Fig.1 shows that all these columns have a similar split across the categories. This similar observation can be made across all columns except for the target column where it is unbalanced.

		Count
buying	maint	
high	high	108
	low	108
	med	108
	vhigh	108
low	high	108
	low	108
	med	108
	vhigh	108
med	high	108
	low	108
	med	108
	vhigh	108
vhigh	high	108
	low	108
	med	108
	vhigh	108

Fig. 1. Dataset has similar splits

Fig.2 shows us the correlation of the features with each other as well as the correlation of the features with the target variable. We can see that the features have no correlation with each other whereas some features like safety and seating capacity have a good correlation to the target.

III. Models

This section discusses the mathematical and conceptual aspects of the Random Forest algorithm.

A. Intuition

Random forest is a supervised learning algorithm. It has two variations – one is used for classification problems and other is used for regression problems. It is one of the most flexible and easy to use algorithm. It creates decision trees on the given data samples, gets prediction from each tree and selects the best solution by means of voting. It is also a pretty good indicator of feature importance. Random forest algorithm combines multiple decision-trees, resulting in a forest of trees, hence the name Random Forest. In the random forest classifier, the higher the number of trees in the forest results in higher accuracy. Before understanding the working of the random forest we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus a collection

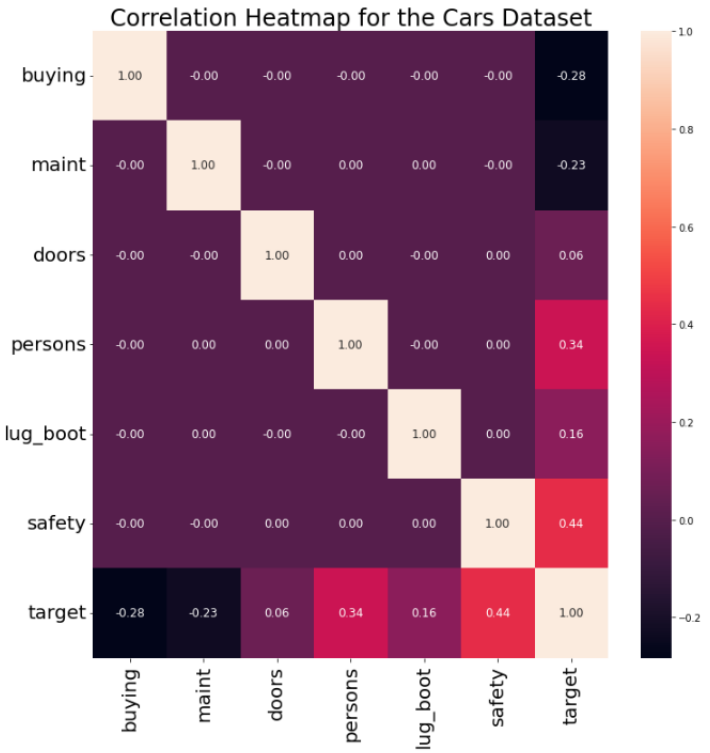


Fig. 2. Correlation Heatmap of features

of models is used to make predictions rather than an indivic model. Ensemble uses two types of models:

- Bagging– It creates a different training subset fi sample training data with replacement the final outpu based on majority voting. For example, Random For
- Boosting– It combines weak learners into strong learr by creating sequential models such that the final mc has the highest accuracy. For example, ADA BOOST, BOOST

As mentioned earlier, Random forest works on the Bagg principle. Now let's dive in and understand bagging in de Bagging Bagging, also known as Bootstrap Aggregation is ... ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

B. The Algorithm

Steps involved in the random forest algorithm:

- In Random forest n number of random records are taken from the data set having k number of records.
- Individual decision trees are constructed for each sample
- Each decision tree will generate an output.
- Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

Going into more detail, we will explain the steps. In the first stage, we randomly select "k" features out of total m features and build the random forest. In the first stage, we proceed as follows:-

- Randomly select k features from a total of m features where $k < m$
- Among the k features, calculate the node d using the best split point
- Split the node into daughter nodes using the best split
- Repeat 1 to 3 steps until l number of nodes has been reached
- Build forest by repeating steps 1 to 4 for n number of times to create n number of trees

In the second stage, we make predictions using the trained random forest algorithm

- We take the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome.
- Then, we calculate the votes for each predicted target.
- Finally, we consider the high voted predicted target as the final prediction from the random forest algorithm.

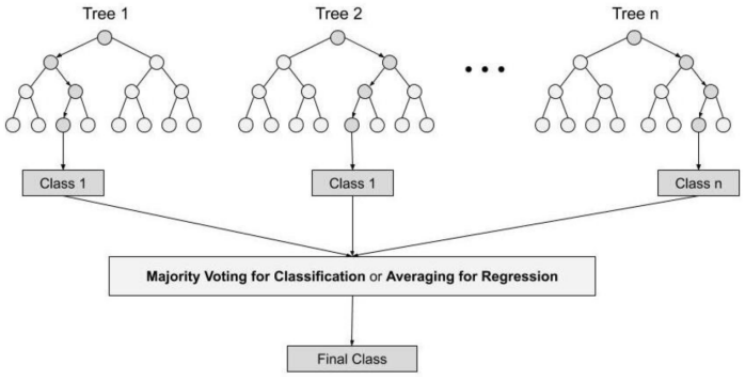


Fig. 3. Depiction of Random Forests

C. Assumptions

A random forest's assumptions are the same as the assumptions an individual decision tree makes which are:

- In the beginning, a part of the training set is considered as the root.
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are distributed recursively on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

D. Advantages

- One of the most accurate learning algorithms available
- It can handle many predictor variables
- Provides estimates of the importance of different predictor variables
- Maintains accuracy even when a large proportion of the data is missing

E. Disadvantages

- Can overfit datasets that are particularly noisy
- For data including categorical predictor variables with different number of levels, random forests are biased in favor of those predictors with more levels
- Therefore, the variable importance scores from random forest are not always reliable for this type of data

IV. Modelling

In this section, we will explore how the model was applied to the dataset at hand and what we can infer from it. The categorical features were first encoded ordinally to ensure that the ranking of the categories was maintained in the encodings. We then split the dataset into a train and validation split of an 85:15 ratio and train the Random Forest Classifier.

We obtain an accuracy of 97% and an F1 score of 0.9 on the dataset. The confusion matrix on the validation set is given in Fig. 3.

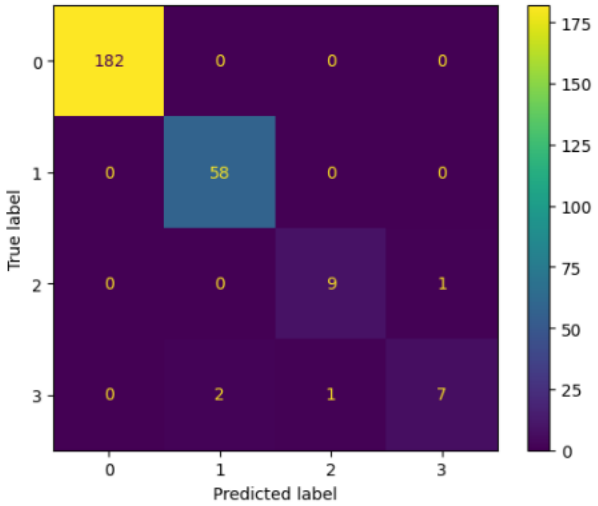


Fig. 4. Confusion Matrix of the Model Predictions

We can also find out the first few layers of the decision tree. This has been represented in Fig.4. As you can see the safety is the primary feature that is taken into account when plotting the tree.

V. Conclusions

Safety was one of the key factors in predicting the target variable for the car. The model is doing a good job of segregating all the categorical features into the different classes. This shows that the dataset has a good amount of separability and the misclassification rate is very low.

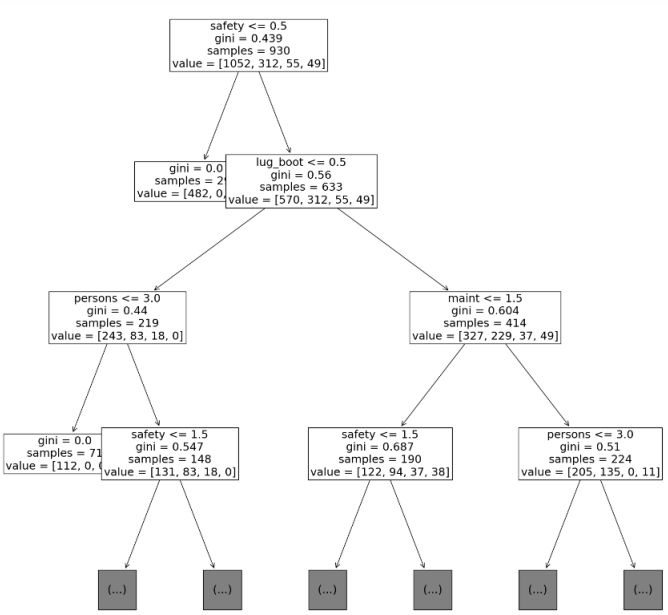


Fig. 5. Plotting the first few layers of the tree

A Mathematical Essay on Decision Trees

Aryan Pandey
Department of Ocean Engineering
Indian Institute of Technology Madras
Chennai, India
na19b030@smail.iitm.ac.in

***Abstract*—This document is an overview of the mathematical aspects of Decision Tree as well as its application on a sample data set. The algorithm has been applied on a data set of cars and the prediction task is to classify a car based on its safety**

I. Introduction

Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

In this paper, the aim is to apply Decision Tree to classify a car based on its safety. The paper systematically goes through first the mathematical details of Decision Tree, the nature of the data set which has been given to us, then the problem we have in hand and how it has been solved, and finally the conclusions which were drawn. Useful insights and figures have been presented whenever necessary.

II. Datasets

The dataset given has the following details of multiple cars with the aim of classifying it based on its safety. It has the details of buying price, price of maintenance, number of doors, seating capacity, size of luggage boot and the estimated safety of the car.

The dataset consists of purely categorical columns in which the columns related to buying price, price of maintenance, and estimated safety of the car are categorised into very high, high, medium and low. The target condition of the car has 4 categories which are very good, good, acceptable and unacceptable.

Fig.1 shows that all these columns have a similar split across the categories. This similar observation can be made across all columns except for the target column where it is unbalanced.

		Count
buying	maint	
high	high	108
	low	108
	med	108
	vhigh	108
low	high	108
	low	108
	med	108
	vhigh	108
med	high	108
	low	108
	med	108
	vhigh	108
vhigh	high	108
	low	108
	med	108
	vhigh	108

Fig. 1. Dataset has similar splits

Fig.2 shows us the correlation of the features with each other as well as the correlation of the features with the target variable. We can see that the features have no correlation with each other whereas some features like safety and seating capacity have a good correlation to the target.

III. Models

This section discusses the mathematical and conceptual aspects of the Decision Tree algorithm.

A. Intuition

The Decision tree algorithm is a simple yet efficient supervised learning algorithm wherein the data points are continuously split according to certain parameters and/or the problem that the algorithm is trying to solve.

Every decision tree includes a root node, some branches, and leaf nodes. The internal nodes present within the tree describe the various test cases. Decision Trees can be used to solve both classification and regression problems. The algorithm can be thought of as a graphical tree-like structure that uses various tuned parameters to predict the results. The decision trees apply a top-down approach to the dataset that is fed during training.

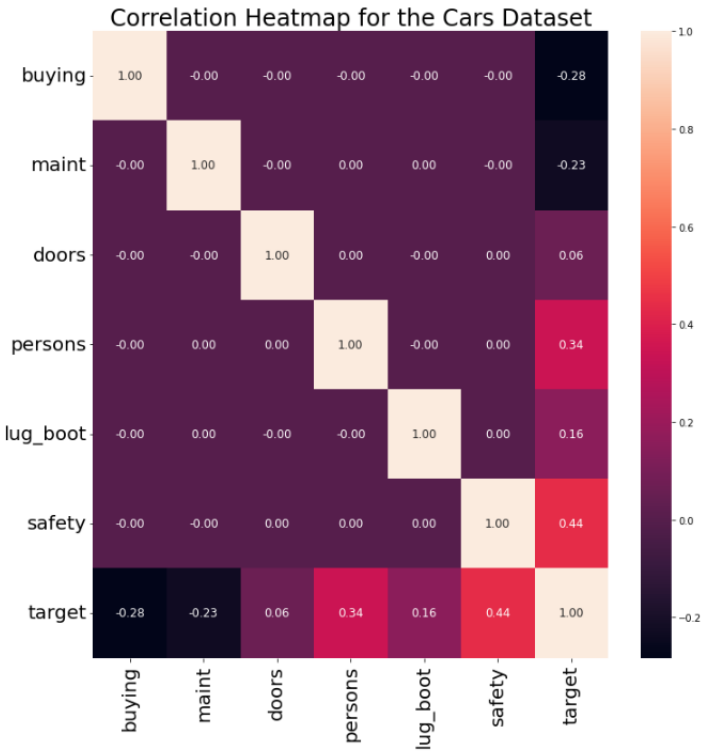


Fig. 2. Correlation Heatmap of features

B. The Algorithm

Entropy: Entropy is the amount of information needed to accurately describe the data. If the data is homogeneous, then the entropy is 0. Mathematically, entropy is written as:

$$Entropy = -\sum_{i=1}^{n} p_i * log(p_i) \tag{1}$$

Gini Index: It measures the impurities in the node. It has a value between 0 and 1. It is the sum of square if the probabilities of each class. It is formulated as:

$$GiniIndex = 1 - \sum_{i=1}^{n} (p_i)^2 \tag{2}$$

- Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- Splitting: It is a process of dividing a node into two or more sub-nodes.
- Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node. Leaf / Terminal Node: Nodes do not split is called Leaf or Terminal node.
- Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
- Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.
- Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

C. Types of Decision Tree Algorithms

There are 2 types of Decision tree algorithm. The 2 types are listed below:-

- Categorical Variable Decision Tree: Decision Tree which has a categorical target variable then it called a Categorical variable decision tree.
- Continuous Variable Decision Tree: Decision Tree has a continuous target variable then it is called Continuous Variable Decision Tree.

D. Assumptions

Below are some of the assumptions we make while using Decision tree:

- In the beginning, the whole training set is considered as the root.
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are distributed recursively on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

E. Disadvantages

- They are not well suited to continuous variables.
- Usually, they provide a lower prediction accuracy than predictive algorithms.
- Over-fitting is a problem if the design of the tree is too complex.

IV. Modelling

In this section, we will explore how the model was applied to the dataset at hand and what we can infer from it. The categorical features were first encoded ordinally to ensure that the ranking of the categories was maintained in the encodings. We then split the dataset into a train and validation split of an 85:15 ratio and train the Decision Tree Classifier.

We obtain an accuracy of 97% and an F1 score of 0.9 on the dataset. The confusion matrix on the validation set is given in Fig. 3.

We can also find out the first few layers of the decision tree. This has been represented in Fig.4. As you can see the safety is the primary feature that is taken into account when plotting the tree.

V. Conclusions

Safety was one of the key factors in predicting the target variable for the car. The model is doing a good job of segregating all the categorical features into the different classes. This shows that the dataset has a good amount of separability and the misclassification rate is very low.

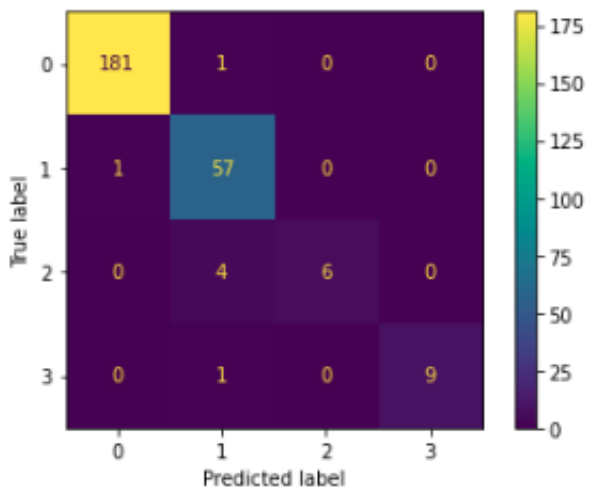


Fig. 3. Confusion Matrix of the Model Predictions

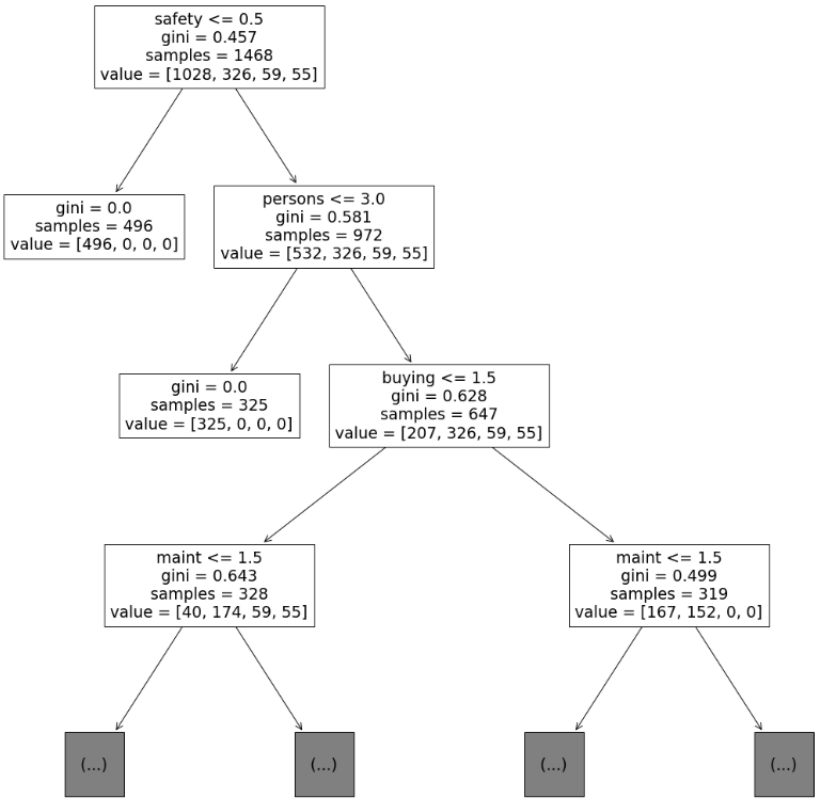


Fig. 4. Plotting the first few layers of the tree

A Mathematical Essay on Naive Bayes

Aryan Pandey
Department of Ocean Engineering
Indian Institute of Technology, Madras
Chennai, India
na19b030@smail.iitm.ac.in

***Abstract*—In this study, we will study the mathematical aspects of the Naive Bayes Algorithm. The dataset that we use is the Census Dataset from the United States of America in the year 1994. Using Visualisation and Modelling techniques, the conclusions we draw from this study are three-fold. First, We see that there is a significant difference in behaviour of people who invest their money versus the ones who don't. Second, Some features like the age of the person and the number of years of education have a significant impact on the classification problem at hand. Third, dealing with the investors and non-investors separately improves performance.**

***Index Terms*—Naive Bayes, Classification**

I. Introduction

One of the key factors to look into when understanding the demography of a country, is to look at the income levels of the people residing in it. In many cases the simplest and fastest way to get an idea of the overall income levels of a country is to look at the number of people who earn above a certain threshold and what are the major traits associated with these kind of people. In this problem, with the help of the US Census data, we aim to try to see these kind of traits in the people who earn more than 50,000 US Dollars a year.

The dataset given consists of some characteristics, like the age, working class, marital status, occupation etc., of the working class of the United States of America. From the problem that we are trying to solve, we get an idea of the important factors that lead to a person making more than 50,000 US Dollars in a year. We also try and find some correlations between other features in general. This helps us in better understanding the demography of the working class.

In order to solve this problem, we use the Gaussian Naive Bayes model that is offered by the sklearn package. We use multiple plotting libraries for the visualisations which have been shown through this paper. In order to tackle this problem, we compare two approaches. The first approach is one which tries to fit a model to the whole dataset in one go. The second one is an approach where we split the dataset into two parts (based on some criteria) and fit a model to each of those splits. The results for the same have been depicted in a later section.

Through this study we hope to better understand the demography of the working class and how we can best represent it using the Naive Bayes model. Section II talks about the dataset which we have used for the study along with some visuals that support some insights from it. Section III dives into the working of the Naive Bayes model and the evaluation metrics that we will be using for this problem. Section IV dives into the implementation details, where we talk about both the approaches and we try to visualise the fit of the model and reason out in which scenarios each approach works best.

II. Datasets

The dataset given to us consists of the data of 32,561 people belonging to the working class of the United States of America. The details given to us are - Their Age, Working Class, Final Weight, Level of Education, Number of Years of Education, Marital Status, Occupation, Relationship, Race, Gender, Capital Gains, Capital Losses, Working Hours per Week, Native Country and Whether or not they earn more than 50,000 US Dollars in a year.

We notice that there's no visible null values on a simple inspection. But when we try seeing the unique values of each column, we see that some columns have a "?" symbol present in an entry. This represents a missing value and we replace all such values with the word "Missing". We then try to see if any of the columns which are continuous in nature have any visible distribution.

We find that the Age and Final Weight features approximately follow a Gaussian Distribution as shown in Fig 1. Another interesting thing to note is that when we see a histogram of the number of Hours Worked per Week (as shown in Fig. 2), we notice a sharp spike at the number 40. This is because this is the work hour commitment for a normal day job for any company. This is further validated by the fact that most of the people work in the Private Sector and are not self employed or in a kind of Working Class which allows flexible working hours (as shown in Fig. 3). As can be seen in Fig. 4, the majority of the population are centred around a few key occupations like Prof-Speciality, Craft special, Exec-Managerial, Adm-Clerical and Sales. Capital Gain and Capital Loss are two of the features which contain a large number of zeros. On further inspection, we see that we can divide the population into two kinds of people - Investors (Those who have either a non-zero Capital Gain or Capital Loss) and Non-Investors (Those who have Zero Capital Gain and Zero Capital Loss). We make a new feature called Capital Profit and Loss which is calculated as the difference of the Capital Gain and Capital Loss. We split the dataset into two parts based on whether or not the value of the Capital Profit and Loss is zero. Fig 5, 6 and 7 show the correlation heatmap obtained for the whole population, the Investors and the Non-Investors.

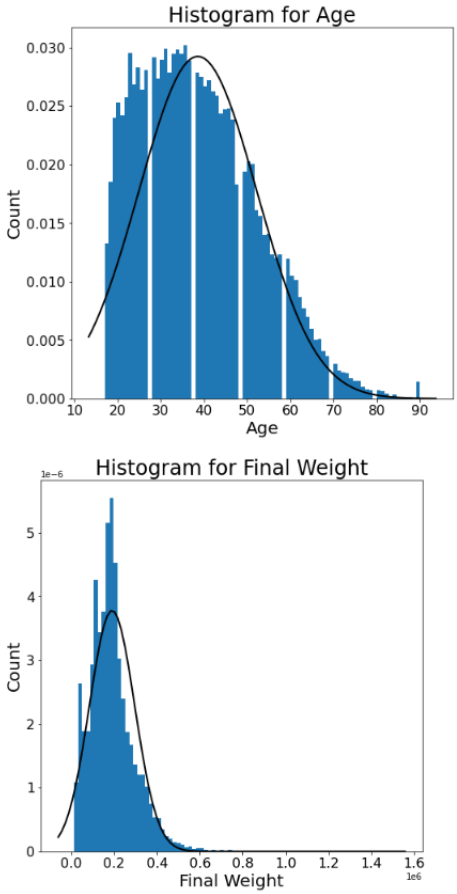


Fig. 1. Gaussian Distributions for Age and Final Weight

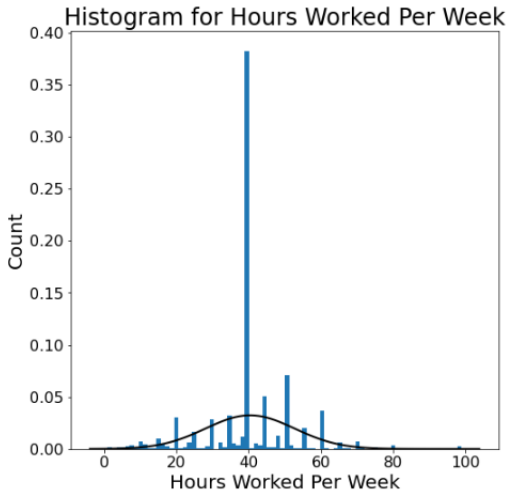


Fig. 2. Histogram for the number of Hours Worked Per Week

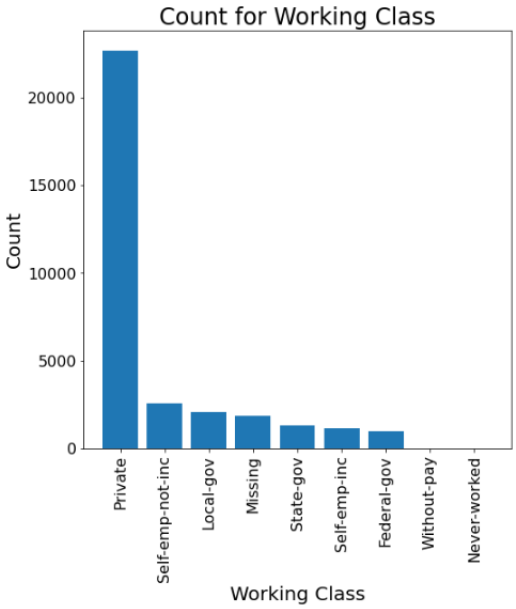


Fig. 3. Working Class Distribution

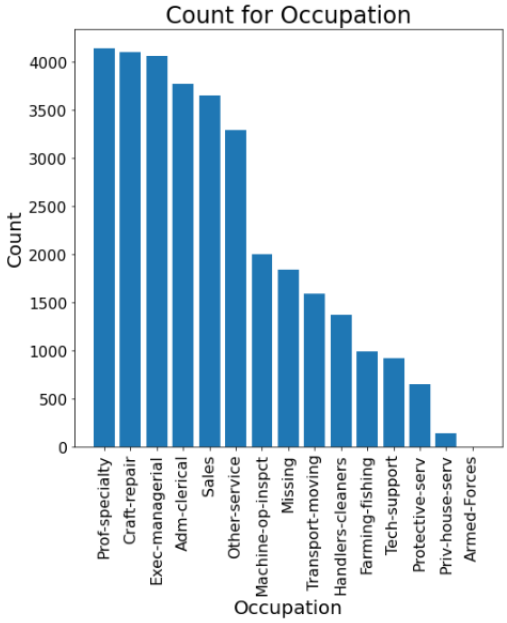


Fig. 4. Occupations taken on by the people

As can be seen in the above graphs, the points that were discussed above are verified. The Age and Final Weight follow a Normal Distribution, there's a huge surge in the Work hour distribution at the 40 Hour mark, this is because of the High Employment in the Private Sector and the majority of the occupations are centred around some key occupations as shown above.

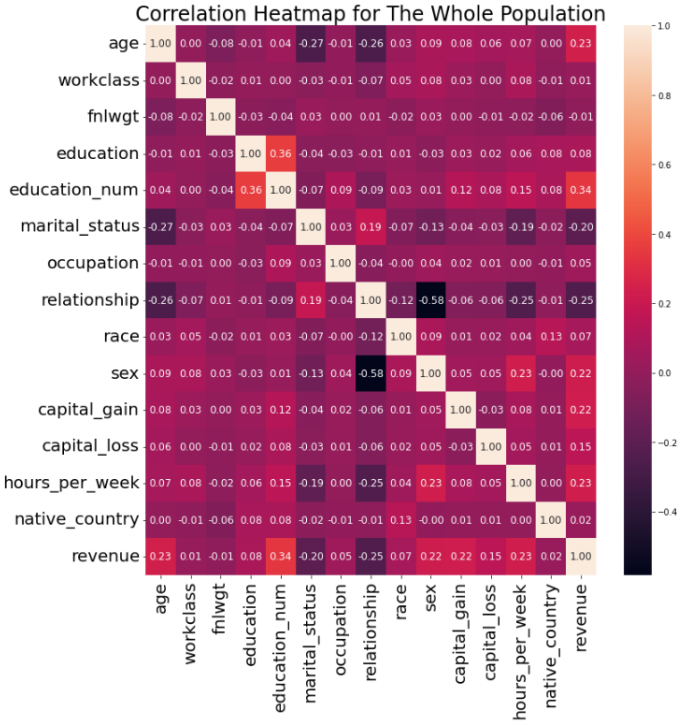


Fig. 5. Population Correlation

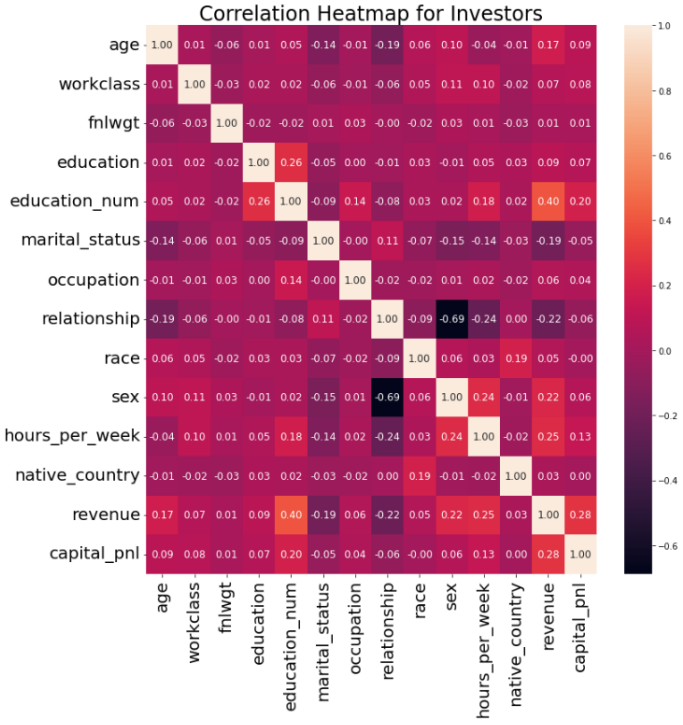


Fig. 6. Investor Correlation

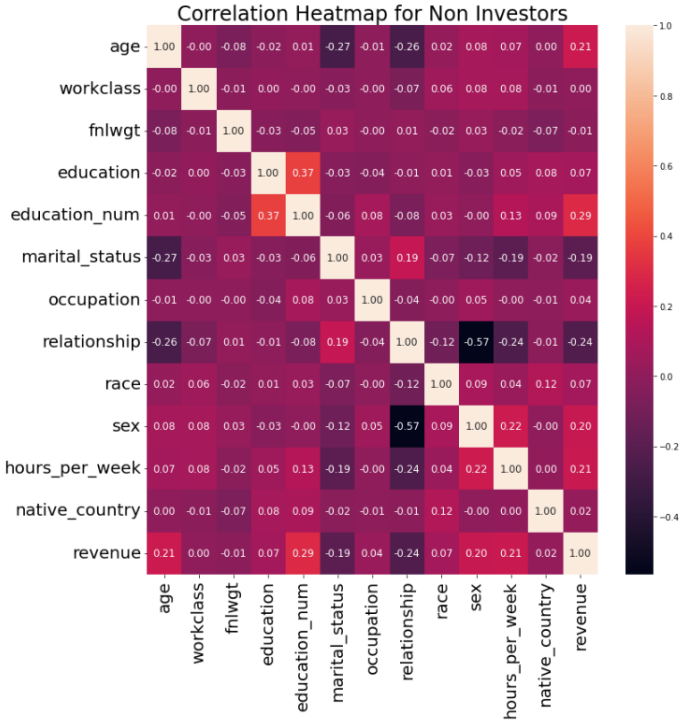


Fig. 7. Non-Investor Correlation

In all the three Figures, we can see that there are some common trends. The Age, Number of Years of Education, Marital Status, Relationship, Gender and Number of Work Hours per Week show a large correlation to the Target Variable which is the Revenue earned.

III. Models

This section takes a deep dive into the mathematical and intuitive concepts involved in the Naive Bayes Algorithm.

A. Intuition

Naive Bayes Classifier uses the Bayes' theorem to predict membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class.

B. The Algorithm

Bayes Theorem states that -

$$P(x|y) = \frac{P(x|y)P(y)}{P(x)} \tag{1}$$

Where,

- P($c|x$): posterior probability of class(c,target) given predictor(x,attributes). This represents the probability of c being true, provided x is true.
- P(c): is the prior probability of class. This is the observed probability of class out of all the observations.

- P($x|c$): is the likelihood which is the probability of predictor-given class. This represents the probability of x being true, provided c is true.
- P(x): is the prior probability of predictor. This is the observed probability of predictor out of all the observations.

Naive Bayes Classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature. In real world datasets, we test a hypothesis given multiple evidence on features. So, the calculations become quite complicated. To simplify the work, the feature independence approach is used to uncouple multiple evidence and treat each as an independent one.

C. Types of Naive Bayes Algorithms

There are three major types of Naive Bayes Algorithms. Many modifications of these exist, but the crux of the models remain the same. They are:

- Gaussian Naive Bayes: Which assumes that the underlying distribution of the dataset is Gaussian in nature
- Multinomial Naive Bayes: Which cares about counts for multiple features that do occur.
- Bernoulli Naive Bayes: Which cares about counts for a single feature that do occur and counts for the same feature that do not occur

D. Applications of Naive Bayes

Naive Bayes is one of the most used and fastest classification algorithms. It is very well suited for large volume of data. It is successfully used in various applications such as:

- Spam filtering
- Text Classification
- Sentiment Analysis
- Recommendation Systems

It uses the Bayes theorem of Probability for Prediction of Unknown Class.

IV. Modelling

In this section, we contrast and compare two of our approaches. One where we fit a model to the whole population and one where we have two models - One for the investors and one for the Non-Investors. The vanilla model achieves an F1-Score of 0.33. When we split the decision making criteria, we see that the Investor model achieves an F1-score of 0.74 and the Non-Investor Model that of 0.34, both of which are improvements over the vanilla model. This shows that making this split in predictions will help make better decisions regarding the demography. Given in Fig 8 - 12 are the parameter values learned by the Naive Bayes model for the various cases. We fit a Gaussian Naive Bayes for the Vanilla Model and the Investor Model and we use a Multinomial Naive Bayes for the Non-Investors.

	Revenue <= 50K	Revenue >50K
age	36.773244	44.162882
workclass	3.878490	3.908858
fnlwgt	190242.214153	187374.587908
education	10.133019	10.853361
education_num	9.583916	11.616784
marital_status	2.788034	2.078057
occupation	6.330432	6.791397
relationship	1.678604	0.721161
race	3.627859	3.771545
sex	0.611452	0.852609
hours_per_week	38.850002	45.523387
native_country	37.037809	37.460972
capital_pnl	96.340229	3843.405775

Fig. 8. Mean for the Features of the Vanilla Model

	Revenue <= 50K	Revenue >50K
age	2.069172e+02	1.221455e+02
workclass	1.290551e+01	1.351802e+01
fnlwgt	1.126458e+10	1.049606e+10
education	2.816591e+01	1.902911e+01
education_num	1.700481e+01	1.676584e+01
marital_status	1.372054e+01	1.181185e+01
occupation	2.974118e+01	3.037753e+01
relationship	1.347789e+01	1.354272e+01
race	1.186135e+01	1.162498e+01
sex	1.131902e+01	1.120711e+01
hours_per_week	1.622374e+02	1.304965e+02
native_country	5.403683e+01	4.663598e+01
capital_pnl	1.025130e+06	2.157017e+08

Fig. 9. Standard Deviation for the Features of the Vanilla Model

	Revenue <= 50K	Revenue >50K
age	40.707702	45.082812
workclass	3.818960	4.004333
fnlwgt	186038.717577	188183.861338
education	10.271231	10.838710
education_num	9.780777	11.946558
marital_status	2.634628	2.110737
occupation	6.356814	6.868079
relationship	1.468729	0.735195
race	3.691244	3.776119
sex	0.647136	0.848339
hours_per_week	39.830151	45.860857
native_country	37.030283	37.461242
capital_pnl	1336.330481	12227.597978

Fig. 10. Mean for the Features of the Investor Model

	Revenue <= 50K	Revenue >50K
age	2.395507e+02	1.295570e+02
workclass	1.237823e+01	1.242134e+01
fnlwgt	1.035463e+10	1.002059e+10
education	2.483711e+01	1.813632e+01
education_num	1.659863e+01	1.598519e+01
marital_status	1.286211e+01	1.102297e+01
occupation	2.992622e+01	2.865191e+01
relationship	1.262382e+01	1.256549e+01
race	1.083049e+01	1.069575e+01
sex	1.039117e+01	1.029148e+01
hours_per_week	1.728410e+02	1.343625e+02
native_country	5.801970e+01	4.485234e+01
capital_pnl	1.359433e+07	5.841815e+08

Fig. 11. Standard Deviation for the Features of the Investor Model

	Revenue <= 50K	Revenue >50K
age	-8.563115	-8.361959
workclass	-10.802544	-10.783542
fnlwgt	-0.000791	-0.000889
education	-9.847285	-9.759576
education_num	-9.900871	-9.706207
marital_status	-11.134138	-11.418490
occupation	-10.306366	-10.231865
relationship	-11.637592	-12.455309
race	-10.871057	-10.813991
sex	-12.652151	-12.306726
hours_per_week	-8.502400	-8.329458
native_country	-8.548577	-8.525213

Fig. 12. Log Probabilities for the Features of the Non Investor Model

V. Conclusions

In this project, I build a Gaussian Naive Bayes Classifier model to predict whether a person makes over 50K a year. We compare two approaches, one where we fit a model to the whole dataset, called the Vanilla Approach and one where we split the dataset into two parts - One consisting of people who invest their money and one consisting of people who do not invest their money. As seen, the second approach significantly outperforms the first one in terms of our performance metrics. We conclude that this approach is a better one for decision making overall, since the characteristics of the people belonging to these two groups are significantly different. If a single model is used to fit moth, it will miss out on the unique behaviours of these two groups which is exactly what we observe in our experiments.

VI. Avenues For Further Research

One possible area of improvement would be to have a different model for the categorical features (a Multinomial Naive Bayes) and another for the Continuous ones (a Gaussian Naive Bayes). Combining the Predicted probabilities of these models for our prediction, should increase the reliability of our prediction framework.

References

[1] An Article on Gaussian Naive Bayes: https://towardsdatascience.com/gaussian-naive-bayes-4d2895d139a

[2] An Article on Multinomial Naive Bayes: https://www.upgrad.com/blog/multinomial-naive-bayes-explained/

A Mathematical Essay on Logistic Regression

Aryan Pandey
Department of Ocean engineering
Indian Institute of Technology, Madras
Chennai, India
na19b030@smail.iitm.ac.in

***Abstract*—Problems where we need to classify data into a given set of labels is one that is of extreme importance. To handle such problems simple regression techniques are not enough and this has given rise to a new set of ways to handle such problems. The main focus of this study is one such method, Logistic Regression, where we will also look at its application on one of the most famous classification problems - Titanic Survival Prediction. The main aim of this problem is to predict whether or not a given person would have survived the Titanic disaster.**

***Index Terms*—Logistic Regression, Classification**

I. Introduction

The Titanic was a British Passenger ship which sank in the North Atlantic Ocean on 15th April 1912 after striking an iceberg during her maiden voyage from Southampton, UK to New York City, United States. Out of the 2,224 passengers on board, more than 1500 died. In our problem, we aim to better understand the demography of the people on board along with the relation of certain individual traits to their survival chance. Further we move on and try to look into what exactly were the important traits of a person which decided whether or not the person survived.

Trying to model the chances of survival as a binary classification problem is something that would be of interest, since we would be able to figure out what kind of people are the most likely to survive in the event of such a disaster. The main reason why over 1500 people died in this disaster was simply because there weren't enough lifeboats on the ship. It is interesting to try and figure out through our analysis, which sections of society were able to survive this disaster and which ones succumbed to it. The data-set given to us contains details of 891 of the people who were on board the ship along with whether or not they survived the shipwreck. The details given are the ticket class of the passenger, their gender, age and how many siblings or spouses they had onboard the ship, their ticket number, the fare of the journey, the cabin number if the were allotted one and where they embarked on the journey form.

In order to do this whole modelling, we fit a Logistic Regression model to the data-set given using the sklearn package in python. Since this is a classification problem, we try to model this using one of the most basic and simple to understand or interpret classification models. We split the data-set into a training data-set and a validation data-set in a 85:15 ratio in a stratified manner. This ensures that we can validate our results and make sure that the model has not over-fit to it. All the training is done on the training set and any results reported are on the validation set.

For our evaluation we use the F1-score metric which has been explained in later sections of the paper. We achieve an F1-score of 0.7736 using the Logistic Regression model. The confusion matrix for the same has also been shown in the later sections of the paper. Through this paper we better understand the demography of the people travelling onboard the Titanic and also build out a list of features that turned out to be the contributing factor to the survival rates onboard the ship.

Section II gives a detailed view of the data-set and some visualisations of the features in the data-set. In Section III we explain the mathematical aspects of Logistic Regression. Section IV talks about how we applied Logistic Regression to the problem at hand and makes a few statements about the features which were an important contributing factor to the survival of the passengers.

II. Datasets

The dataset provided to us is the classic Titanic Dataset. In this dataset we are given details about whether or not the person survived, the Ticket Class of the person which serves as a proxy for the socio-economic status of the person, the gender of the person, their age, number of siblings and spouses, number of family relations (in terms of mother, father and children), the ticket number, ticket fare, the cabin number if a cabin was allotted and the port form which the passenger embarked.

The age of the person was fractional is the age was less than 1. If the age was an estimated age, it takes the form xx.5 . While counting siblings and spouses, any mistresses and fiances were ignored. While counting the number of family relations, children who were travelling only with a nanny were considered as having no family relations on board. The dataset has a high number of missing values in the Age and Cabin features. In order to solve this, we first find out the median age of the passengers across a passenger class and impute any missing values with the median age of that person's class. Fig. 1 shows a boxplot representing the Age plotted against the passenger class. In order to fill the missing values, any missing values in Passenger Class 1 are imputed with a value of 24, Passenger Class 2 with 39 and Passenger Class 3 with 29. In order to fill the Cabin, the missing values are treated as if there were no cabin allotted to those individuals. Moreover, to ease the preprocessing, we take the first letter to

demonstrate the importance of the cabin assigned. Therefore, the values in the Cabin column are A, B, C and N, representing Cabins in the A-series, B-series, C-series and No Cabin. As another preprocessing step we also extract the designation of the person from their name. These follow as ”Mr.”, ”Mrs.” and so on.

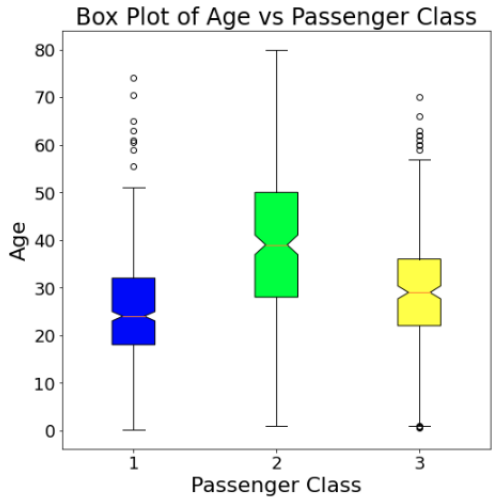


Fig. 1. Age Variation across different Passenger Classes

In order to better understand the data given to us, we first try and understand how the survival rates vary across the different passenger classes. Fig. 2 shows a plot which demonstrates how many people survived and how many didn't across different passenger classes. Clearly from this we can see that the fraction of people who survived in the first class is the highest, whereas it reduces slightly in the second class and the survival rates for the third class are abysmally low.

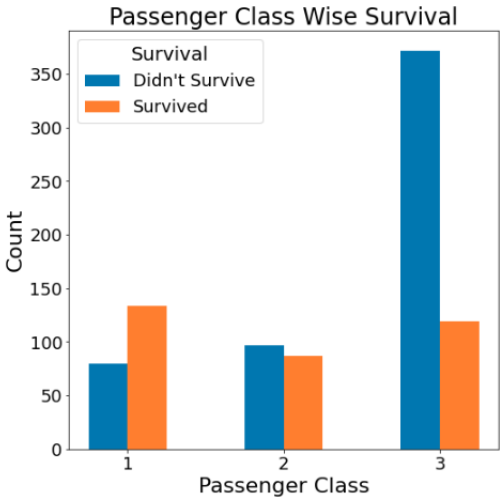


Fig. 2. Passenger Class Wise Analysis of Survival vs Non Survival

We also try to better understand the demography of the people there by seeing the gender distribution across the different passenger classes. Fig 3 shows this distribution. One can see that in general the number of males travelling onboard the ship is higher than the number of females travelling onboard the ship. In First Class, we can see that the ratio of Male to Female passengers is almost one whereas in second class the ratio is almost 1.5 and in third class the ratio is well above 2.

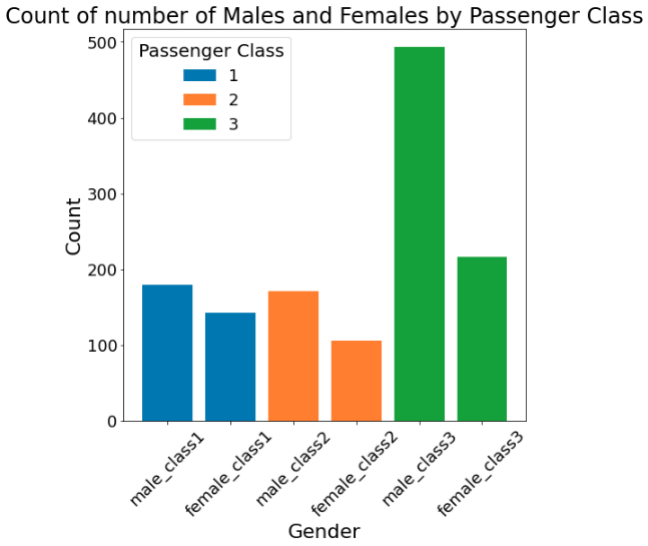


Fig. 3. Number of Males and Females in each Passenger Class

In Fig.4 we see a Pearson Correlation Matrix demonstrated as a heatmap. This shows us the Correlation between any two features. For us the main interest here is to see which features are heavily correlated to the Survival of a person. On inspection we can see that the Gender of a person as well as the Passenger Class and Fare play an important role in deciding the survival chances since they are all correlated to the survival chances. Since no feature here has a severely low correlation to Survival, we won't be dropping any features and will let the model decide via it's training as to which features turn out to be more important. An interesting thing to note here is some other correlations. The Cabin allotted to a person is highly correlated with the Passenger Class. This means that the first class passengers are allotted the best cabins while the third class passengers go cabinless.

III. Models

In this section we will develop a thorough understanding of the Logistic Regression framework, the math behind it, any assumptions and the evaluation metrics being used here.

A. Variables

The independent variables used to train any classical machine learning framework are of two types: Categorical and Continuous. Continuous variables are those variables which are continuous in nature over their whole domain. Categorical Variables are the discrete variables that are present. Some Categorical variables present themselves in textual form (eg. Gender) which we need to convert to a numeric form so that

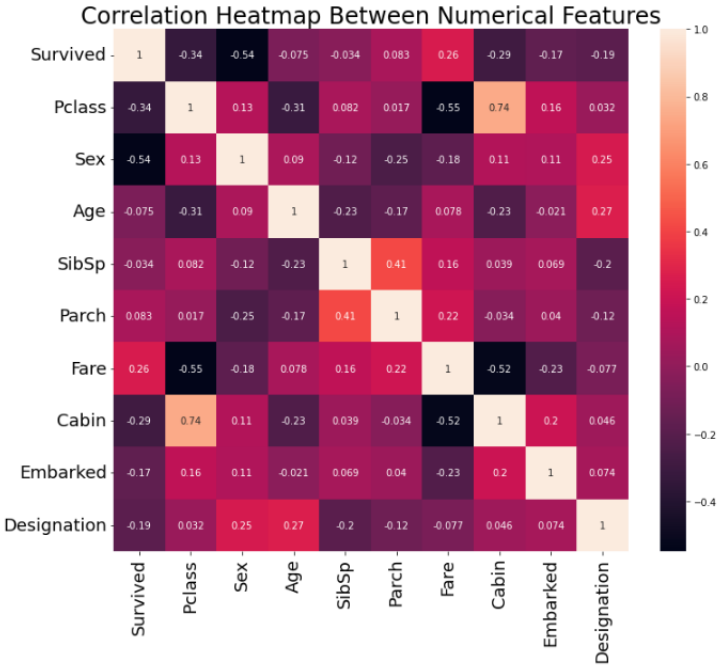


Fig. 4. Correlation Plot for the features in the Dataset

our model can understand it. We do this using a Label Encoder which assigns a single number to represent each unique value of the variable. In case of a Logistic Regression framework, the dependent variable is usually a Binary Variable (meaning that it takes two discrete values).

B. The Algorithm

The logistic function for the Logistic Regression Algorithm to calculate the logits is as shown below:

$$logit(Y) = ln(odds) = ln(\frac{\pi}{1-\pi}) = \alpha + \beta X \tag{1}$$

The above equation assumes that there is a single feature in our dataset. In case we have n features, the formulation changes to as shown below:

$$logit(Y) = ln(odds) = ln(\frac{\pi}{1-\pi}) = \alpha + \beta_1 X_1 + ...\beta_n X_n \tag{2}$$

In the Logistic Regression framework, we try to predict the chances of an event happening as a regression problem and then apply a function to convert these chances to a probability value. In order to do this we will need a function that takes in all real number as its domain and gives a value from [0,1] as it's range. This function is given by the Sigmoid function. The Sigmoid function definition is as given below:

$$S(x) = \frac{1}{1 + exp(-x)} \tag{3}$$

When applied to our Logistic Regression Framework for n features, it evaluates to the expression given below:

$$P(Y = 1|X) = \frac{1}{1 + exp(-(\beta_0 + \sum_{i=1}^{n} \beta_i X_i))} \tag{4}$$

Since this is a binary classification problem:

$$P(Y = 0|X) = 1 - P(Y = 1|X) \tag{5}$$

$$P(Y = 0|X) = \frac{exp(-(\beta_0 + \sum_{i=1}^{n} \beta_i X_i))}{1 + exp(-(\beta_0 + \sum_{i=1}^{n} \beta_i X_i))} \tag{6}$$

All the Coefficients represented by β_i are calculated via the least squares minimisation based on the Maximum Likelihood Estimates.

C. Assumptions of Logistic Regression

Logistic regression does not make many of the key assumptions of linear regression and general linear models that are based on ordinary least squares algorithms – particularly regarding linearity, normality, homoscedasticity, and measurement level. First, logistic regression does not require a linear relationship between the dependent and independent variables. Second, the error terms (residuals) do not need to be normally distributed. Third, homoscedasticity is not required. Finally, the dependent variable in logistic regression is not measured on an interval or ratio scale. However, some other assumptions still apply.

- Binary logistic regression requires the dependent variable to be binary and ordinal logistic regression requires the dependent variable to be ordinal.
- Logistic regression requires the observations to be independent of each other. In other words, the observations should not come from repeated measurements or matched data.
- Logistic regression requires there to be little or no multicollinearity among the independent variables. This means that the independent variables should not be too highly correlated with each other.
- Logistic regression assumes linearity of independent variables and log odds. although this analysis does not require the dependent and independent variables to be related linearly, it requires that the independent variables are linearly related to the log odds.
- Logistic regression typically requires a large sample size. A general guideline is that you need at minimum of 10 cases with the least frequent outcome for each independent variable in your model. For example, if you have 5 independent variables and the expected probability of your least frequent outcome is .10, then you would need a minimum sample size of (10*5 / .10) which is 500.

D. Evaluation Metric

While there are a bunch of metrics which one can use to evaluate a classifciation model, we use the F1-Score to evaluate the Logistic Regression model that we train. The F1-Score is defined as:

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{7}$$

Where Precision is calculated by dividing the true positives by anything that was predicted as a positive. Recall (or True Positive Rate) is calculated by dividing the true positives by anything that should have been predicted as positive.

IV. Modelling

In order to apply the Logistic Regression Framework to the given dataset we use the sklearn package. We first split the dataset into a Training and Validation Split in an 85:15 ratio. After training the model on the Training Split for a maximum of 1000 iterations, we observe that the F1-Score achieved by the model is 0.7736. In order to better understand what the model has learnt, we plot the feature importance of the model which is shown in Fig. 5

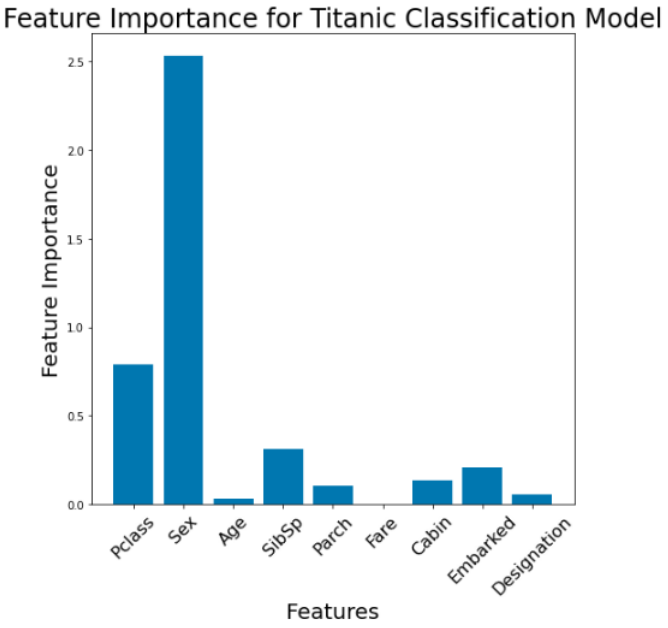


Fig. 5. Feature Importance from the Model

As one can see from the Figure, the model places a heavy emphasis on the Gender of a person, followed by the Passenger Class of the person. In order to understand this better, we also take cases where we change the gender and designation of a person (from male to female and vice versa) keeping all other parameters the same and observe the model's predictions. Shown in Fig. 6 are the results of this experiment

	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked	Designation
0	3.0	Male	22.0	1.0	0.0	7.2500	7.0	2.0	Mr.
1	1.0	Male	38.0	1.0	0.0	71.2833	2.0	0.0	Mr.
2	3.0	Female	22.0	1.0	0.0	7.2500	7.0	2.0	Mrs.
3	1.0	Female	38.0	1.0	0.0	71.2833	2.0	0.0	Mrs.

```
Male Survival:  [0. 0.]
Female Survival:  [1. 1.]
```

Fig. 6. Gender Bias in the Model

As you can see there is some sort of inherent bias present in this model, since we can see that even if we keep all other parameters same and change the gender of the person, the survival of the person changes as predicted by the model.

V. Conclusions

A. Contributions

- A person's gender was one of the key contributing factors in their survival. Due to the protocol of saving Women and Children first, priority was given to them for the limited lifeboats present onboard the ship.
- It is also observed that people travelling in first class had higher survival rates.
- A model trained on this dataset assumes an inherent bias due to these factors when trying to predict the survival of a person onboard the ship.

B. Avenues for Further Research

From this we can see that one possible source of further research is to build a bias free model here while still retaining the essence of the problem. Moreover, we could also make an attempt at fitting non-linear decision boundaries by making the features themselves non-linear

References

[1] Article on the Titanic Disaster: https://en.wikipedia.org/wiki/Titanic
[2] Assumptions of Logistic Regression: https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/assumptions-of-logistic-regression/
[3] F1 Score: https://en.wikipedia.org/wiki/F-score

A Mathematical Essay on Linear Regression

Aryan Pandey
Department of Ocean Engineering
Indian Institute of Technology, Madras
Chennai, India
Email ID: na19b030@smail.iitm.ac.in

***Abstract*—This essay is an overview of the mathematical formulations and applications of Linear Regression. First, the basic aspects of Linear Regression is explained with all the necessary details needed to better understand the study that has been performed. Next the problem statement and the data-set are introduced. The data-set under study is one of Cancer incidence and mortality among different socioeconomic groups in the United States of America population. In order to effectively do this, the data is thoroughly analysed and any insights (both visual and quantitative) have been published in this essay. Finally, the results from the application of the linear regression model is described. The model is applied to the data-set after transforming some of the features in order to better understand which features are important when it comes to correlating the socioeconomic status with the Incidence and Mortality Rate in different counties.**

I. Introduction

Linear Regression is a form of Supervised Machine Learning where one tries to map the feature space to a target (or dependent) variable using a linear equation. Based of off this learned map, one can try to then make predictions about the target variable. In this algorithm, one strives to find the best possible linear fit to the target variable given the feature space.

This is done via an optimisation algorithm. We choose an objective function which measures the distance between the underlying true distribution and the current fit that we have. Via the optimisation algorithm, one aims to minimise the objective function in order to obtain minimum separation between the underlying true distribution and the fit that we have proposed via our linear model. To test the robustness of the model, we create a split in the data, train the model on the first split (the training data) and evaluate it on the second split(the test data).

The problem that we're trying to tackle is to apply the concepts of Linear Regression to better understand the relationship between the different socioeconomic groups in the United States of America and the cancer incidence and mortality rates in them. The data available has details on poverty status, income, health insurance, gender, incidence rates and mortality rates for each county. The study aims to examine whether low income groups or certain sections of society are at a greater risk of being diagnosed and dying from cancer.

Through this essay, one can understand the basics of Linear Regression, dive into the problem that is being examined and understand how the insights produced are backed with the help of visual or quantitative facts.

II. Linear Regression

Linear regression is a statistical analysis which depends on modeling a relationship between two kinds of variables, target (or dependent) and features (independent). The main purpose of regression is to examine if the features are successful in predicting the target and which features are significant predictors of the target. Given below is a discussion on the terminology and the mathematics used in a Linear Regression Approach.

A. Features/Independent Variable

The features or Independent Variables are those set of variables that are being used to predict the target variable. An ideal set of such variables would be a basis set, since this would allow us to map any point in an n dimensional space onto the target variable. The Features or Independent variables can either be discrete (for example, number of sales calls made in a day) or continuous (for example, the time taken by a ball to fall from the top of a building). The discrete features can also be in a textual format (for example, gender of a person). In such cases, care should be taken to get these features into a format which is understandable by a machine (more details in Section III). These set of variables are collectively represented in a matrix often referred to as X.

B. Target/Dependent Variable

Target or Dependent variables can be either continuous or discrete. When we apply Linear Regression to a certain problem statement, the target variable assumes a continuous distribution. This variable is often represented in a vector and if often referred to as Y.

C. Assumptions made in Linear Regression

Following are the assumptions that any Linear Regression framework makes:

- There is a linear relationship present in the underlying true distribution between X and Y
- For any value of X, the variance of the residual is the same (Homoscedasticity)
- Observations are distinct from one another (Independence)
- Y is regularly distributed for any fixed value of X (Normality)

D. Types of Linear Regression

- Uni-Variate Linear Regression: In this kind of Linear Regression we have one predictor and one target variable. The model then finds a linear relationship between this as:

$$Y = w_0 + w_1 X \tag{1}$$

w_0, w_1 are the parameters/weights

- Multi-Variate Linear Regression: In this kind of Linear Regression, we have multiple predictors which combine to form the same target variable. The input is a vector of individual features

$$X = [x_1, x_2, x_3, ..., x_{n-1}, x_n] \tag{2}$$

n is the number of predictors that we have

The linear relationship between the predictors and the targets is then given by:

$$Y = w^T X^* \tag{3}$$

Where X^* represents the X vector and the bias term combined and w is the parameter/weight vector

E. Objective Function (or Loss Function)

The disparity between the underlying true distribution and the predicted values we get from our fit is quantified by an Objective or a Loss function. Most frequently, we use a Root Mean Square Error function which measures the mean of the square of the errors. Such a loss function keeps in mind that the X vector is normally distributed. In certain situations when we would like to assume the prior for X to be a Laplacian Distribution, we would use the Mean Absolute Error in which we take the mean of the Absolute errors.

III. The Problem

We are provided with a data-set that has details about different cancer incidence and mortality in different counties in the United States of America. Our end goal of the study is to establish whether or not there exists a correlation between the socioeconomic status of a person and the cancer incidence or mortality rate. This study also explores the importance given to different features by the linear regression model.

A. Data Description

The data-set consists of county-wise information (represented by each row) on:

- The number of people living in poverty (segregated by gender)
- The median income (segregated by ethnic race)
- Health Insurance (segregated by gender)
- Annual Incidence and Death rates

Information on the State the county is in along with its FIPS code is also present to identify the county with much more ease. For each county, we are also given the recent trend that the country has seen (as a categorical variable, for example, rising, falling etc.). In our study we observe that including the name of the State as a feature was causing the model to create a good fit to the data since the state was one of the major distinguishing factor. One possible analysis that could further be done is to identify if there are some states where there is a higher number of socioeconomic groups that have high cancer incidence and mortality (if at all there is a correlation). For this study, we remove the State feature from our analysis and focus purely on the columns that directly involve a relation to a socioeconomic group. The target variables for our analysis are the incidence rate and mortality rate, since these are numbers that are population normalised.

B. Data Handling

If we take a quick view of the data, we see that missing values exist in the target variables as well as income columns. The way in which the rows with missing values in these columns are handled is as mentioned:

- Income Related Variables: The missing values in these columns (for example, median income overall or median income for a certain ethnic group) has been replaced by the mean of the values of all other counties in the same state. This is possible because the median values are normally distributed, so replacing it with the mean will have no effect on the model's robustness.
- Target Variables: Some states' incidence and mortality rates were not provided either due to confidentiality reasons or because there simply weren't enough cases of incidence or mortality. All counties with such entries were removed from the data-set since any attempt to replace these values with the mean or median could hamper the model's effectiveness. In doing so, we also take care of the missing values in the recent trend feature.

The spacial characters in Incident Rate, Mortality Rate and Average Annual Values of Incidence and Mortality were removed and the remainder, converted to float values. Following this data cleaning we are left with 2618 data points.

C. Exploration

The Pearson Correlation coefficients between the data are examined first. They are a measure of how linearly related any two given variables are. The coefficient of Pearson Correlation between two variables X and Y is given by:

$$\rho_{x,y} = \frac{E[(x - \mu_x)(y - \mu_y)}{\sigma_x \sigma_y} \tag{4}$$

If two variables are the same then the Pearson Correlation Coefficient has a value of 1 and when they are the exact opposite of each other, it has a value of -1. Given figure shows the correlation heatmap between the features of our given data-set

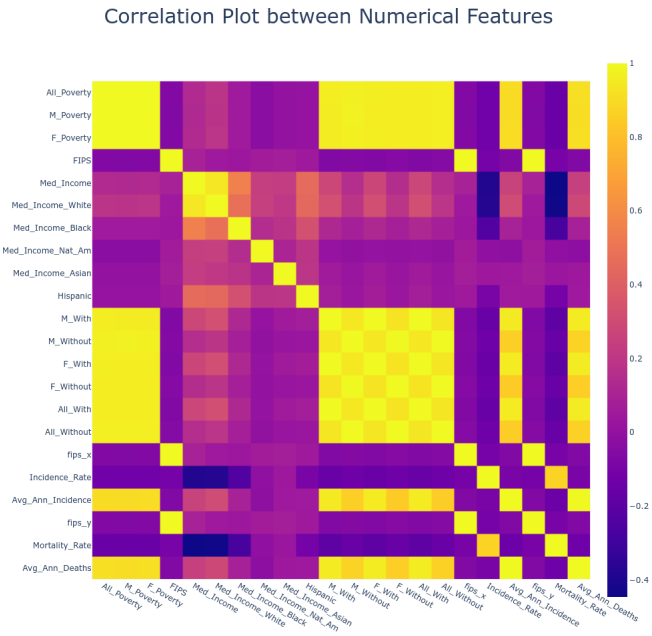


Fig. 1: Correlation Plot between Numerical Features

From this we can find a few relations to Average Annual Deaths and Average Incidence Rate:

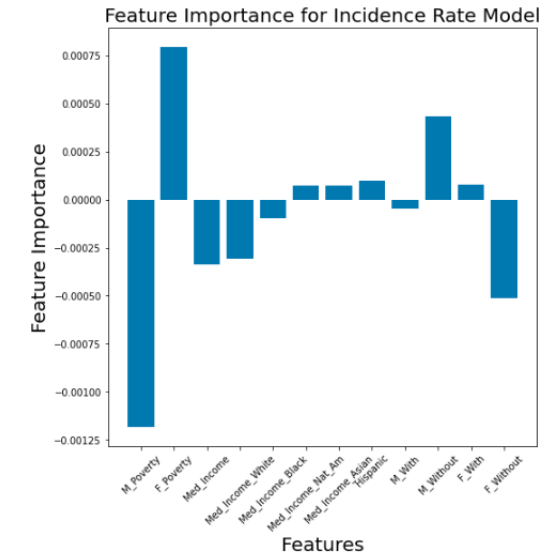
- Not having Health Insurance (which is also highly correlated to poverty) is a big factor in deciding Incidence and death rate
- Poverty also plays a huge role in this regard

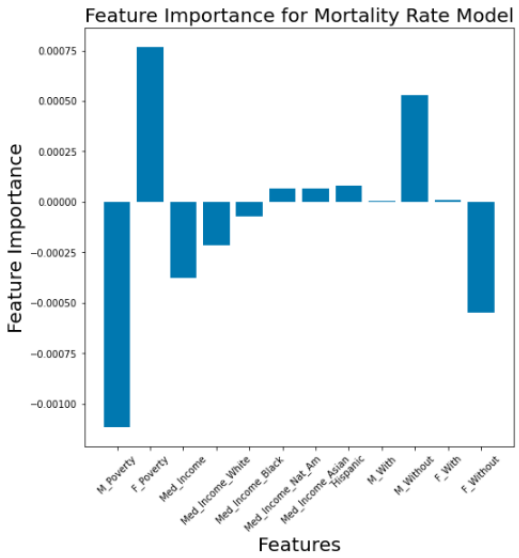
Moreover we can now say that we can remove the overall columns of the Gender and Ethnicity data since we want to get an idea of which groups are impacted the most and these columns are highly correlated with the individual groups.
Given on the right is a scatterplot of the income of different ethnic groups plotted against the Average Annual Incidence.

D. Applying Linear Regression

In order to do this we use the sklearn package in Python. The variable set that we use finally as our features X are M Poverty, F Poverty, Med Income, Med Income White, Med Income Black, Med Income Nat Am, Med Income Asian, Hispanic, M With, M Without, F With and F Without. We fit two linear models to the data, one for predicting Incidence Rates and the other for predicting mortality rates. We use the root mean square error as our evaluation metric after we have split the data-set into 2 splits: the train split and the validation split. On the Validation Split, the linear model for predicting Incidence Rate achieves an RMSE of 16.32 where the Incidence Rates themselves range from 13.5 to 203.7. The linear model for predicting Mortality Rate achieves an RMSE of 13.33 where the mortality rates themselves vary from 9.2 to 125.6. One thing to note here is that these rates are defined as the number of cases per 100000 people in the county.

Given below is a plot of the feature importance brought out by the two models





IV. Conclusion

- Because the correlations with the related overall statistics were quite high, the poverty and insurance data for overall population were redundant.
- Exploratory data analysis found that two overall aspects of socioeconomic status(Poverty Ratio and Median Income) were significantly associated (-0.79) and each of them was correlated with the target variables (Incidence and Mortality Rate).
- The most important characteristics according to Feature Importances are the number of Males in Poverty in a county, followed by number of Females in a county and then cases when each gender does not have health insurance.
- The Insurance Ratio and the incidence/mortality rate were both inversely associated (-0.55)
- While the Insurance Ratio didn't have a very strong relationship with the incidence rate, it had a strong correlation with the mortality rate.
- Socioeconomic Characteristics had a stronger relationship with the Death Rate when compared to the incidence rate.
- The Plots shown visually support all the above quantitative evidence.
- The percentage error in the death rate model was lower when compared to the incidence rate model since the features showed a higher correlation to the death rate than the incidence rate