

Flight Fare Prediction

Submitted by:

Amritesh Kumar.

# Acknowledgement

I would like to express my gratitude to my guide Shubham Yadav (SME, Flip Robo) for his constant guidance, continuous encouragement and unconditional help towards the development of the project. It was he who helped me whenever I got stuck somewhere in between. The project would have not been completed without his support and confidence he showed towards me.

Lastly, I would l like to thank all those who helped me directly or indirectly toward the successful completion of the project.

# Introduction

## Business Problem Framing

Machine Learning is a field of technology developing with immense abilities and applications in automating tasks, where neither human intervention is needed nor explicit programming.

The power of ML is such great that we can see its applications trending almost everywhere in our day-to-day lives. ML has solved many problems that existed earlier and have made businesses in the world progress to a great extent.

Today, we’ll go through one such practical problem and build a solution(model) on our own using ML.

We are about to deploy an ML model for Flight fare prediction and analysis. This kind of system becomes handy for many people.

So, to be clear, this model will provide you will the approximate fare for your flight based on the flight name, timing of the flight, detinaiton and soruce, the number of stops,

## Conceptual Background of the Domain Problem

The goal of this statistical analysis is to help us understand the relationship between flight features and how these variables are used to predict flight fare.

## Review of Literature

From the dataset I get to know that it is a Regression problem . And there are many features which help to find it.

## Motivation for the Problem Undertaken

I am doing this for practice, to get more hands-on data exploration, Feature extraction and Model building.

# Analytical Problem Framing

## Mathematical/ Analytical Modelling of the Problem

I have used Log transformation for transforming the continuous numerical variable containing non-zero elements only as during analysis I found that these variables were not normally distributed, so transformed them using log normal transformation so that the features will be close to normal distributed. I have done some testing separately to check the importance of categorical variables with respect to the fare of flight. Use of Mean, Median to replace the Missing Values in features. Use of Correlation matrix to check the importance and correlation of numerical variables with respect to target variable Fare and Feature scaling using Min Max scaler as we have positive data points.

## Data Sources and their formats

Data I collected from paytm website using web scrapping.

There are more than 7034 observations and 10 features including the target feature fare in dataset.

## Data Pre-processing Done

I have handled the missing values in data set. Based on the Data description I have imputed the missing data. which were described as absence of feature in data description

## Data Inputs- Logic- Output Relationships

I have found out that with continuous numerical variable there is a linear Relationship with the flight fare. And for categorical variable, I have used Boxplot for each categorical feature that shows the relation with the median fare fare for all the sub categories in each categorical variable. For continuous numerical variables I have used scatter plot to show the relationship between continuous numerical variable and target variable.

## Hardware and Software Requirements and Tools Used

The system requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view. System requirements are all of the requirements at the system level that describe the functions which the system as a whole should fulfil to satisfy the stakeholder needs and requirements, and is expressed in an appropriate combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

**Hardware requirements**: -

1. Processor — core i5 and above

2. RAM — 8 GB or above

3. SSD — 250GB or above

**Software requirements**: -

Anaconda

**Libraries**: -

**From sklearn.preprocessing import StandardScaler**

As these columns are different in **scale**, they are **standardized** to have common **scale** while building machine learning model. This is useful when you want to compare data that correspond to different units.

**from sklearn.preprocessing import Label Encoder**

 Label Encoder  and One Hot Encoder. These two encoders are parts of the SciKit Learn library in Python, and they are used to convert categorical data, or text data, into numbers, which our predictive models can better understand.

**from sklearn.model\_selection import train\_test\_split,cross\_val\_score**

Train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make random partitions for the two subsets.

The algorithm is trained and tested K times, each time a new set is used as testing set while remaining sets are used for training. Finally, the result of the K-Fold Cross-Validation is the average of the results obtained on each set.

**from sklearn.neighbors import KNeighborsRegressor**

K Nearest Regressor (KNN) is a very simple, easy to understand, versatile and one of the topmost machine learning algorithms. KNN used in the variety of applications such as finance, healthcare, political science, handwriting detection, image recognition and video recognition

**from sklearn.linear\_model import LinearRegression**

The library sklearn can be used to perform linear regression in a few lines as shown using the LinearRegression class. It also supports multiple features. It requires the input values to be in a specific format hence they have been reshaped before training using the fit method.

**from sklearn.tree import DecisionTreeRegressor**

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy

# Model/s Development and Evaluation

## Identification of possible problem-solving approaches (methods)

For feature transformation I have used Log normal transformation to make the continuous non zero variables close to normal distributed. Use of Annona test to check the importance of categorical features. Use of Pearson’s correlation coefficient to check the correlation between dependent and independent features. Use of Min Max scaler to scale down the features and one label encoding to encode categorical features in numeric.

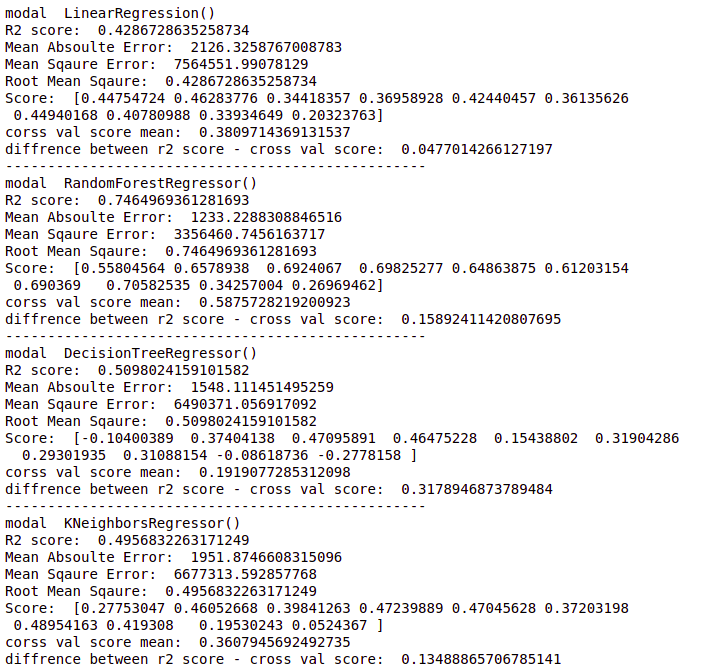
## Testing of Identified Approaches (Algorithms)

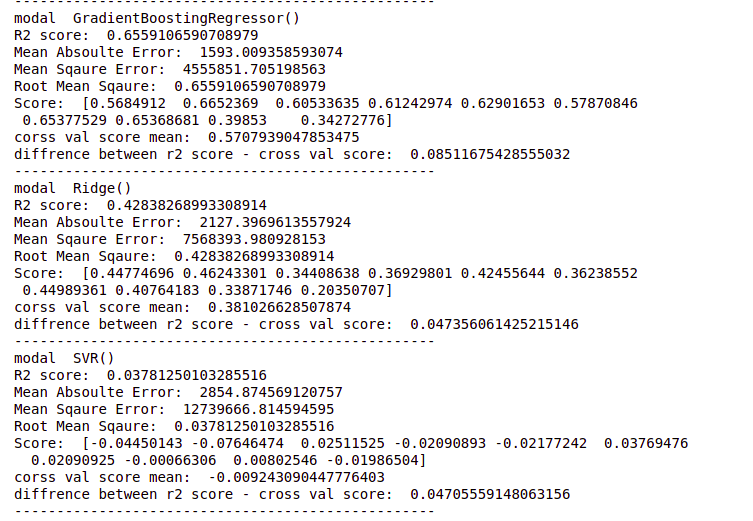
Listing down all the algorithms used for the training and testing.

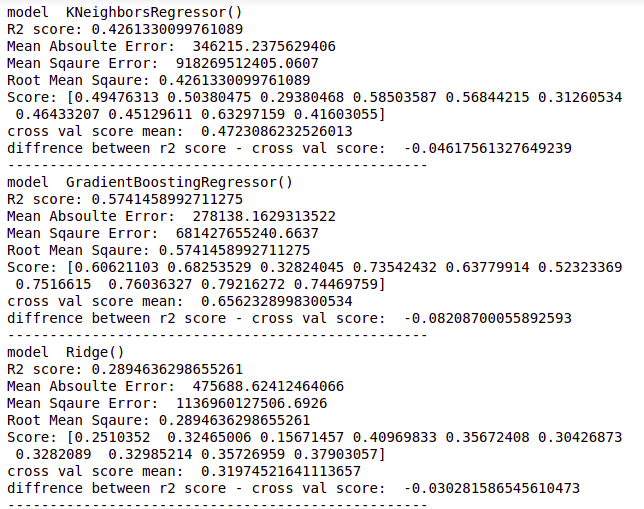
* KNeighborsRegressor ()
* LinearRegression ()
* SVR ()
* DecisionTreeRegressor ()
* RandomForestRegressor ()

I applied all these algorithms in the dataset.

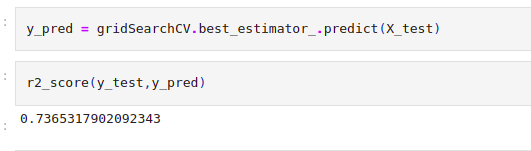
## Run and Evaluate selected models

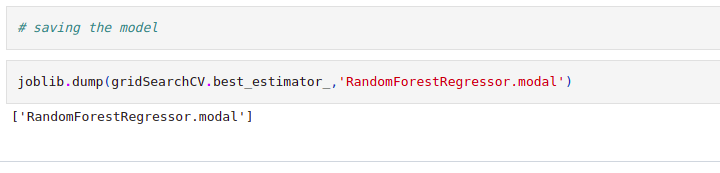








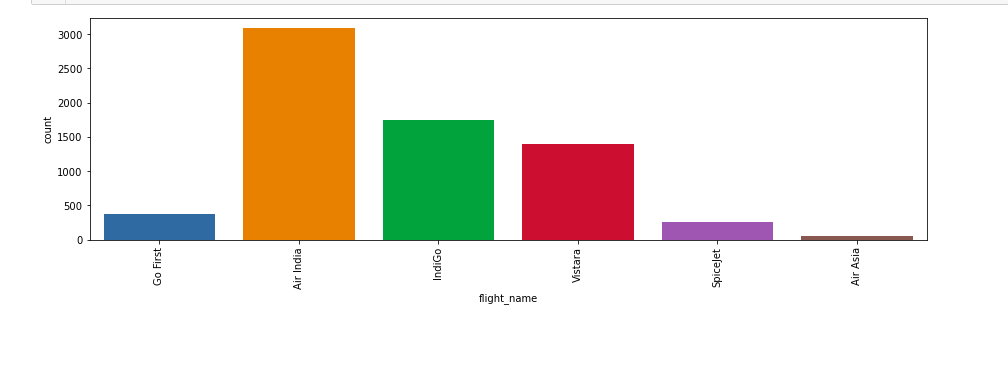


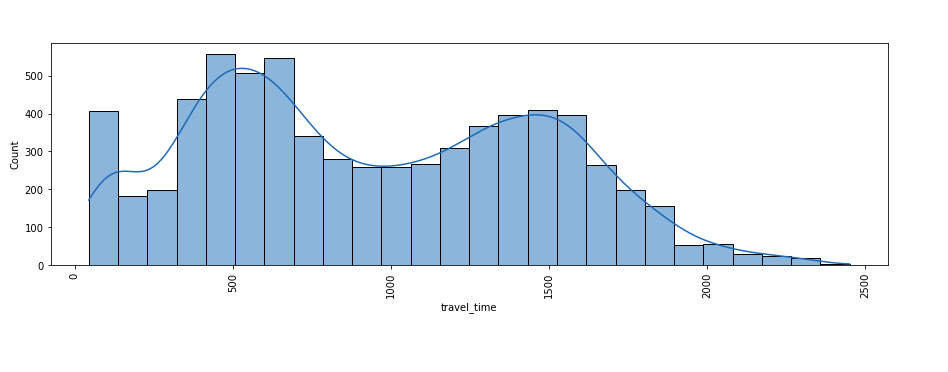


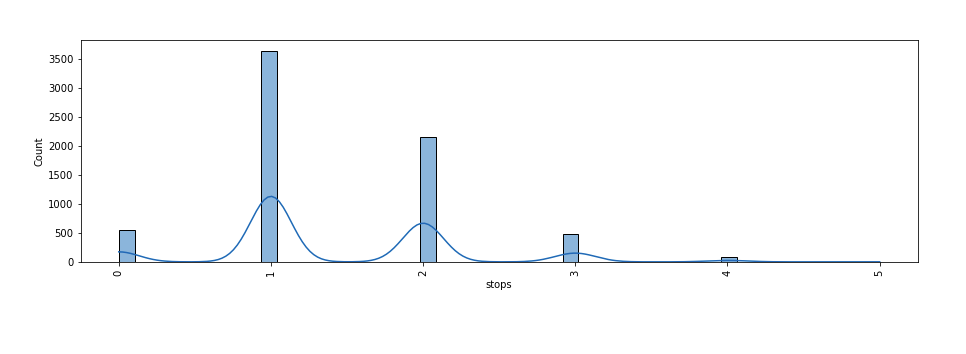
## Key Metrics for success in solving problem under consideration

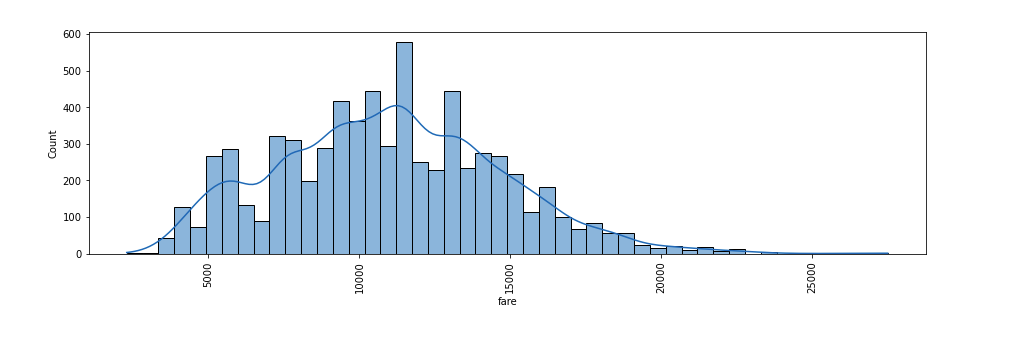
As this is a regression problem, we are required to predict the continuous feature (fare) I have used R2 score, mean absolute error, mean squared error and root mean squared error.

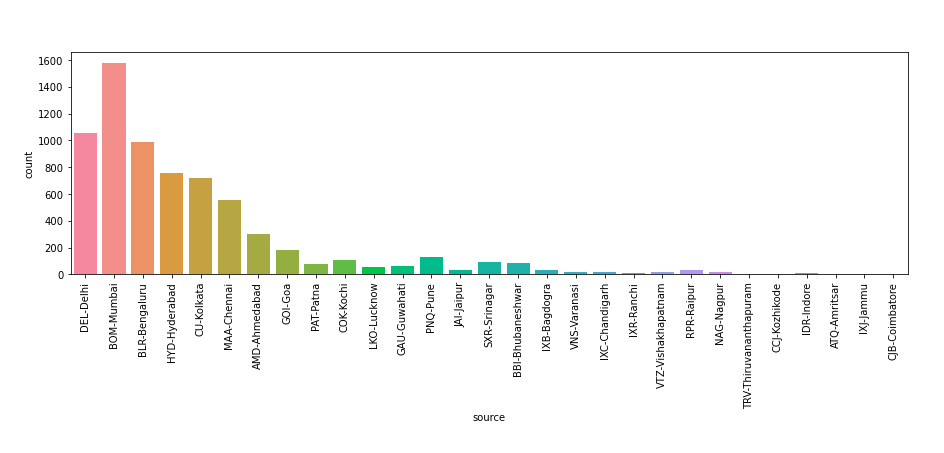
## Visualizations

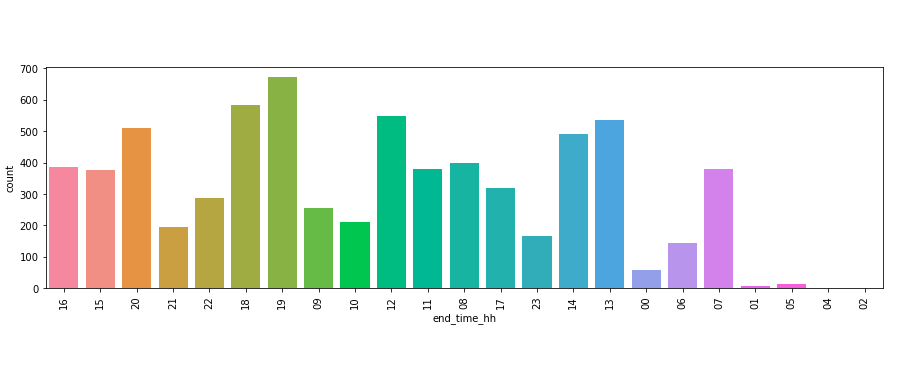
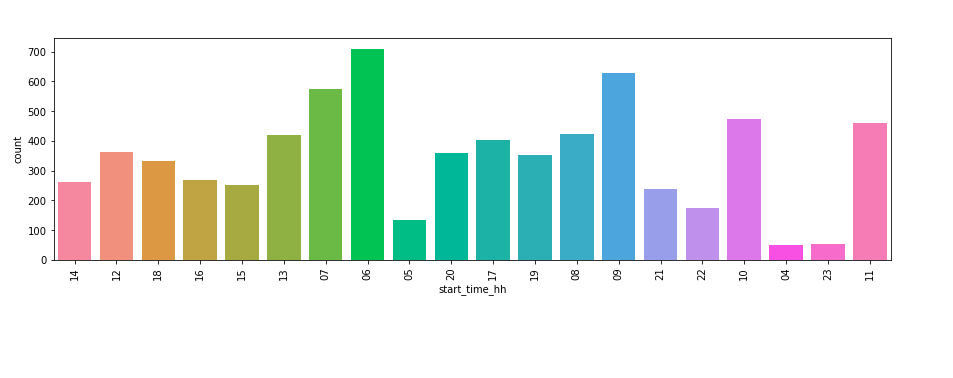
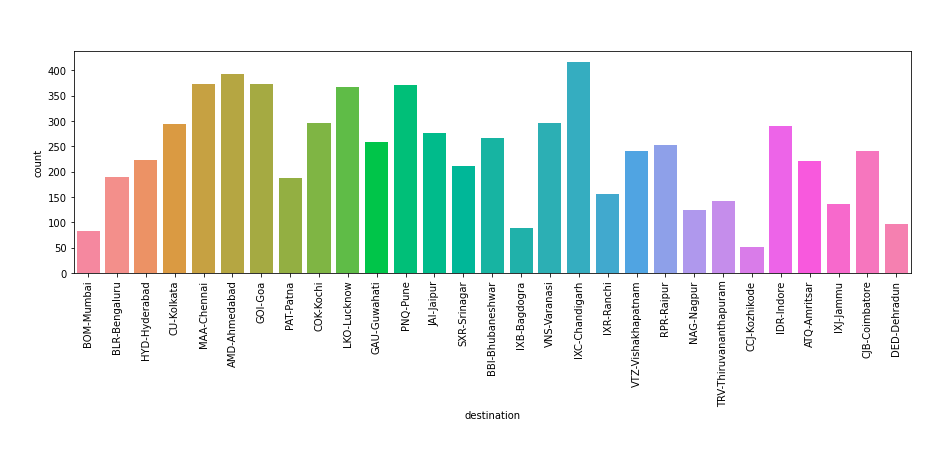


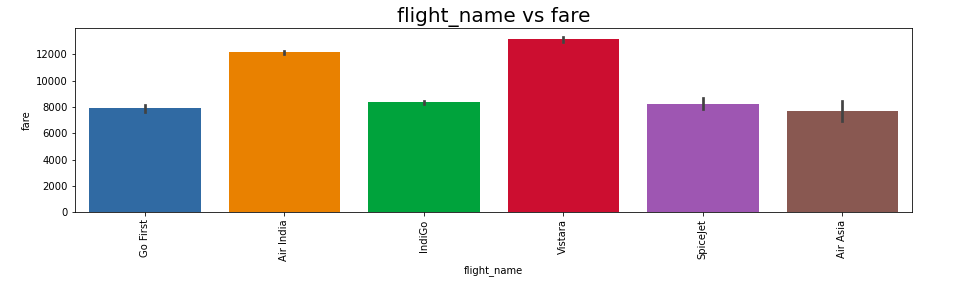


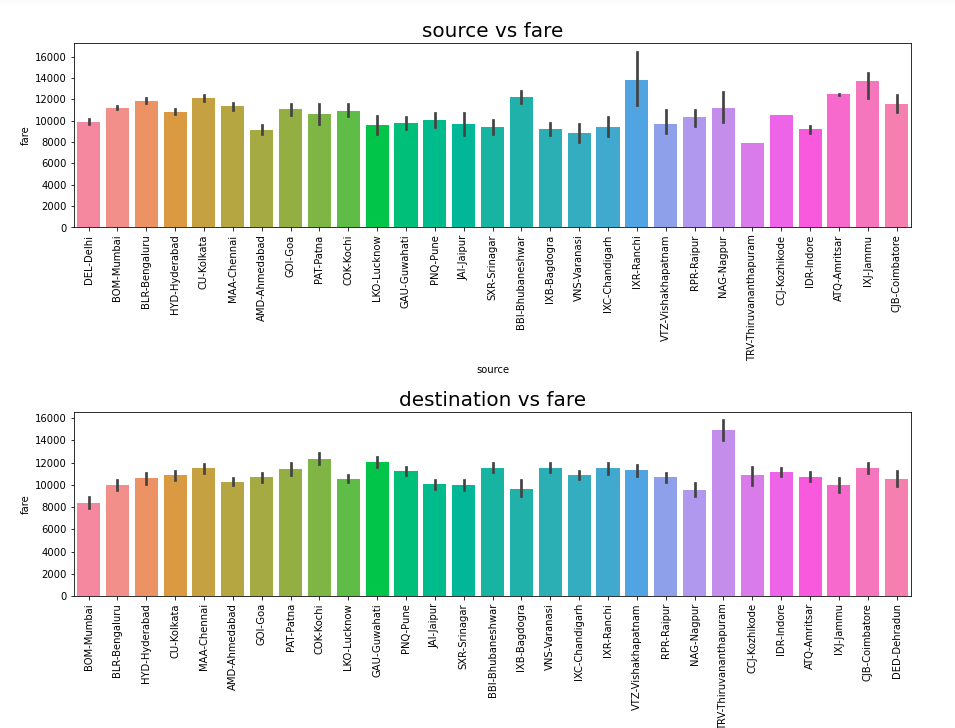


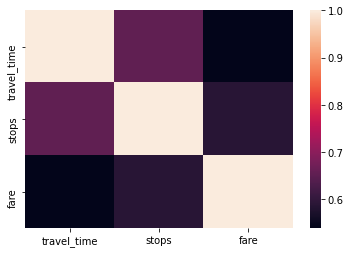












## Interpretation of the Results

Random Forest Regressor R2 score was 73.6%, means 74% of the variance of the dependent variable being studied is explained by the variance of the independent variable.

Higher the R2 score means the model is well fit for the data. However, if R2 score is very high, it might be a case of overfitting. Other metrics Mean Absolute Error, Mean Squared Error and Root Mean Squared Error, with gradient boosting these scores are less then compared to other models. If these errors are less that means the model shows less errors.

# Conclusion

## Key Findings and Conclusions of the Study

From this dataset I get to know that each feature plays a very import role to understand the data. Data format plays a very important role in the visualization and Appling the models and algorithms.

## Learning Outcomes of the Study in respect of Data Science

The power of visualization is helpful for the understanding of data into the graphical representation its help me to understand that what data is trying to say, Data cleaning is one of the most important steps to remove missing value or null value .

Various algorithms I used in this dataset and to get out best result and save that model. The best algorithm is Random Forest Regressor.

## Limitations of this work and Scope for Future Work

Limitations of this project is we have less number of features. If we get interior column, where we will get feature like, food etc. More the number of features, more accuracy we’ll get.

In future, if someone do the proper and detail study of this dataset’s each column than the accuracy will be so high.