**Project Report on Cab Fare prediction based on Machine Learning Techniques**

By Kamaljeet Singh

**Table of Content**

**Content Page No.**

**Chapter1**

* **Problem Statement 03**
* **Data 03**

**Chapter 2**

* **Data Preprocessing 04**
* **Understanding Data 04**
* **Analysis of Numerical Data 04**
* **Analysis of Categorical Data 04**
* **Missing value Analysis 05**
* **Outlier Analysis 05**
* **Feature Selection 07**
* **Feature Scaling 07**
* **Model building 09**
* **Linear Regression 09**
* **Decision Tree Regression 10**
* **Random Forest Regression 10**
* **Cross Validation and Model Tuning 11**
* **Grid Search CV and Hyperparameter Tuning 11**

**Chapter 3**

* **Conclusion 12**

**Chapter 1**

**Problem statement**

You are a cab rental start-up company. You have successfully run the pilot project and

now want to launch your cab service across the country. You have collected the

historical data from your pilot project and now have a requirement to apply analytics for

fare prediction. You need to design a system that predicts the fare amount for a cab ride

in the city. The objective is to predict the fare amount based on the given set of data.

Clearly it is a problem of regression model. We will be applying various regression algorithms to solve the given dataset

**Data**

Based on the problem statement, a regression model is required to guess how much should be the Fare amount for the given set of data

We have a total of 6 independent variables and 1 dependent variables. The training dataset contains 25981 observations.

Our dataset contains fare amount details , no. of passenger counts, pickup location details in the form of pickup latitude and longitude and dropoff location details in the form of dropoff longitude and latitude and pickup\_datetime. So the prime objective would be to get the correct prediction of the fare\_amount for test dataset containing the same features. In the test dataset we do not have the Dependent variable fare\_amount so to overcome this problem we will join both the training set and the test set to form a bigger data set. We will perform all the EDA and data preprocessing techniques on this data set and then split this data set in the Training set and test set and will build model on this data sets.

**Chapter 2**

**Data Preprocessing**

While building any of the Data Science project it needs to be ensured that the data set contains less or no redundant data. To remove all the unwanted data, Data Preprocessing needs to be done. Data preprocessing is a data mining technique that involves transforming raw data into understandable format. Real world data is incomplete, inconsistent and usually contains too many errors. Although complete removal of the errors is not possible but it can be reduced to quite an extent using Data Preprocessing techniques.

**Understanding the data**

In our dataset we have 2 categorical variables (Object type) 5 Numerical Variables (Float Type). Train\_set has 16067 row & 7 columns. Test \_set has 9914 rows & 6 columns.

**Analysis of the Numerical Data**

In the data set 2 categorical variables are present 5 numerical variables (Float). On analyzing the numerical statistics of the given data it is found that the data is in same range in all numerical columns. This means. A simple machine learning algorithm is based on the distance between two points. This means that higher values will dominate the lower values while calculating the distance. If such is the case, FEATURE SCALING is applied. Since in the given dataset there is no such thing as large differences in the distance between two points. Hence feature scaling is not required.

**Analysis of the Categorical Data**

The variable named as “fare\_amount” contains 16043 unique values and the variable pickup\_datetime contains 25981. So feeding this variable will not provide effective results. Moreover while making the dummy variables maximization of the dimension of dataset would be required leading to curse of dimensionality. So it would be better to drop the ‘pickup\_datetime’ and converting the ‘fare\_amount’ to the numeric data.

**Missing Value Analysis**

As per the analysis of our dataset there are nearly 9938 missing values in fare\_amount variable and 55 missing values in passenger count. A total of 9993 missing values were found in our ‘Data’ dataset. After the analysis of various imputation methods median imputation was found to be accurate and so all the missing values were imputed with the median value. After imputation it was found that there were no longer any missing values in our ‘Data’ dataset

**Outlier Analysis**

An outlier is an observation that lies at an abnormal distance from other values in a random sample from a population. Outlier analysis is also called as anomaly detection. Outlier analysis can be a tricky part when the dataset is small. The boxplot method is used for the outlier detection, if any value is greater than (Q3+(1.5\*IQR)) or less than (Q1 –(1.5\*IQR))

Where

Q1 > 25% of data or less than or equal to this value or 25TH %ile

Q2> 50% of data or less than or equal to this value or Median or 50th %ile

Q3 > 75% of data or less than or equal to this value or 75th %ile

IQR( Inter Quartile Range) = Q3 –Q1

Outlier should be removed in such a manner that important information is not lost in the process for correct prediction. Before the removal of outlier, the nature of outlier must be known. Some outliers are just an outcomes of human measurement errors.

For the given dataset with the help of the boxplot method it was found that there were many outliers in the given dataset. In such case all the outliers were detected and were given a value of NA. later these NA values were imputed with the median imputation

**Removal of outliers using box plot**

Outlier removal will be done in following features

* fare\_amount
* pickup\_latitude
* pickup\_longitude
* dropoff\_latitude
* dropoff\_longitude
* passenger\_count

**Analysis**

All the outliers present in the dataset were assigned a value of NA and were later imputed with the median

**Feature Selection**

For the feature selection the correlation between independent variable must be known. For any machine learning model, its performance decreases with multicollinearity, so it must be ensured that two variables carrying same information must be dropped.

**Analysis of numerical variables**

**VIF Analysis**

VIF value for any variable having less than value 10 depicts the multicollinearity. Multi collinear variables should be dropped from the final dataset. As per the VIF analysis was found that none of the variables have a VIF over 10. So none of the variables were dropped from the ‘Data’dataset

**Feature Scaling**

For the given set of data Feature Scaling is not required. Feature scaling is important for those variables for which their values are either too high or too low. Algorithms using distance method, are affected by out of range values. Higher value dominates the lesser values in calculating distance. Since the given dataset has a uniform distribution so feature scaling is not required.

**Building the Machine Learning model using Algorithm Stated in the Coursework**

key terms to avoid any confusion in next steps.

* No\_outlier: training dataset , containing observation which left after outlier removal
* X\_train : containing independent variables of ‘Data’
* y\_train: containing dependent variable (fare\_amount) of ‘Data’

**Model building**

**Model Performance**

For the given problem statement. Regression model is required and then model performance will be used to decide the final model and algorithm.

**Linear Regression**

**Analysis**

By applying the Linear regression Following error metrics were obtained

MAE

1.327820966421147

MSE

2.8854577422488545

RMSE

1.6986635164884347

Where

MAE:- Mean Absolute Error

MSE:- Mean Squared Error

RMSE:-Root Mean Squared Error

**Decision Tree Regression**

**Analysis**

By applying the Decision Tree regression Following error metrics were obtained

MAE

1.5030234183851803

MSE

4.212084935337295

RMSE

2.052336457634882

Where

MAE:- Mean Absolute Error

MSE:- Mean Squared Error

RMSE:-Root Mean Squared Error

**Random Forest Regression**

By applying the Random forest regression Following error metrics were obtained

**Analysis**

MAE

1.2284603285564488

MSE

2.3693851275777704

RMSE

1.5392807176008443

**Cross Validation and Model Tuning**

**GridsearchCV**

Hyperparameter are the parameters which are passed as argument to the building functions, like kernel, criterion, n\_estimators etc. So to get best values of these, gridserchcv is used. In this technique, a list of these different parameters are made and then gridsearchcv build model for every combination of these parameters and then check crossvalidation score and based on score it, further it gives the best combination of hyperparameters.

And then the model is build with the values of hyperparameter given by GridSearchCV.

This is called performance tuning and this would be used to tune the model..

**Hyperparameter tuning**

Hyperparameter tuning is used to find optimum values of arguments used in building models like n\_estimators, max\_depth, kernel etc. so that better result with these tuned parameter can be gained. So hyperparameter tuning will be done to improve our model Random Forest Regression

**Model tuning for random forest regression**

For the tuning of the model GridSearchCV was used. 'min\_samples\_leaf', 'min\_samples\_split', 'n\_estimators' were taken in the parameter Grid. On further running of the code it was found that for the Optimum Tuning of the model 'min\_samples\_leaf' should be 5, 'min\_samples\_split' should be 12 and 'n\_estimators' should be 200 as the best parameters.

Once these parameters were used following changes were obtained in the error metrics

MAE

1.1827316336281253

MSE

2.137185173791558

RMSE

1.4619114794650045

It is clearly evident that the value of RMSE decreased further suggesting that the Random forest model is optimized

**Chapter 3**

**Conclusion**

**Final Model and Training Dataset**

**From the above model we have selected the random forest regression and all the features except pickup\_datetime. The resultant dataset is the y\_pred with all the correct predictions as per the random forest regressor**

**Dataset:**

• First take whole training dataset and test dataset and then merge the dataset into one data

• Drop columns pickup\_date time. Perform EDA. Split the dataset into training set and test set

• Create matrix of features (X\_train, X\_test) and Vector of dependent variable(y\_train,y\_test)

• Build various Regression model using Regression Algorithms

**Model:**

• Use random Forest model and train using dataset which we prepared with above steps.

• Perform hyperparameter tuning.

• Build model using tuned hyperparameter.

• Prediction can be obtained from model