**comprehensive project Report**

1.we will load data in both R and python .Data has 7 variable ,out of which 6 are independent and 1 is dependent (fare\_amount)

2.converting all data into numeric :=

a. Converting pickup datetime into numeric as the output is in numeric form.

Pre-processing done to impute NA values as totel NA values were less than 30 percent.

3.checked multicollinearity factor using vifcore and dfs4.corr(method = ("pearson")).

|  | **pickup\_datetime** | **pickup\_longitude** | **pickup\_latitude** | **dropoff\_longitude** | **dropoff\_latitude** | **passenger\_count** | **fare\_amount** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **pickup\_datetime** | 1.000000 | 0.008686 | 0.002000 | 0.002271 | -0.002618 | -0.023288 | -0.014127 |
| **pickup\_longitude** | 0.008686 | 1.000000 | 0.671789 | 0.366191 | 0.305677 | -0.012355 | -0.011975 |
| **pickup\_latitude** | 0.002000 | 0.671789 | 1.000000 | 0.284715 | 0.450692 | 0.000521 | -0.009892 |
| **dropoff\_longitude** | 0.002271 | 0.366191 | 0.284715 | 1.000000 | 0.573829 | -0.009709 | 0.006309 |
| **dropoff\_latitude** | -0.002618 | 0.305677 | 0.450692 | 0.573829 | 1.000000 | -0.001005 | 0.001630 |
| **passenger\_count** | -0.023288 | -0.012355 | 0.000521 | -0.009709 | -0.001005 | 1.000000 | -0.000372 |
| **fare\_amount** | -0.014127 | -0.011975 | -0.009892 | 0.006309 | 0.001630 | -0.000372 | 1.000000 |

Dropping the variables that are not contributing. Much.

dfs4 = dfs4.drop(["pickup\_latitude","dropoff\_longitude","dropoff\_latitude"] ,axis =1)

4..Applying different regression models as output we have to do prediction as output is continuous variable ,starting with linear regression as it’s the basic model.

Output comes as :=

For python :=

=============================

|  |  |  |  |
| --- | --- | --- | --- |
| OLS Regression Results | | | |
| **Dep. Variable:** | fare\_amount | **R-squared:** | 0.849 |
| **Model:** | OLS | **Adj. R-squared:** | 0.849 |
| **Method:** | Least Squares | **F-statistic:** | 7.242e+04 |
| **Date:** | Tue, 11 Jun 2019 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 21:44:02 | **Log-Likelihood:** | -34894. |
| **No. Observations:** | 12853 | **AIC:** | 6.979e+04 |
| **Df Residuals:** | 12852 | **BIC:** | 6.980e+04 |
| **Df Model:** | 1 |  |  |
| **Covariance Type:** | nonrobust |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **coef** | **std err** | **t** | **P>|t|** | **[0.025** | **0.975]** |
| **pickup\_longitude** | -0.1173 | 0.000 | -269.102 | 0.000 | -0.118 | -0.116 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Omnibus:** | 2800.280 | **Durbin-Watson:** | 2.018 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 5491.001 |
| **Skew:** | 1.323 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 4.804 | **Cond. No.** | 1.00 |

Warnings:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

**Note := Linear regression model not sufficient as both R-square and Adjusted r-square are more than 80 pecent ,,it indicates independent variables are contributing much in model so model will be practical. Error Rate 2.48**

**5.Applying Decision Tree Regressor :=**

**Applying decision tree regressor model we get MAE of 2.98 ,its less than Linear Regression Model**

**6.Applying Random Forest :=**

**We applied random forest regressor model with 10 trees ,100 ,500, 1000, the best result we got with 1000 tree in python i.e MAE is 2.34**

**So fix the random Forest regressor model is the best model.**