Flight Fare Prediction

Submitted by:

Nidhi Charde.

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

```
modal  LinearRegression()
R2 score:  0.4286728635258734
Mean Absoulte Error:  2126.3258767008783
Mean Sqaure Error:  7564551.99078129
Root Mean Sqaure:  0.4286728635258734
Score:  [0.44754724 0.46283776 0.34418357 0.36958928 0.42440457 0.36135626
 0.44940168 0.40780988 0.33934649 0.20323763]
corss val score mean:  0.3809714369131537
diffrence between r2 score - cross val score:  0.0477014266127197
--------------------------------------------------
modal  RandomForestRegressor()
R2 score:  0.7464969361281693
Mean Absoulte Error:  1233.2288308846516
Mean Sqaure Error:  3356460.7456163717
Root Mean Sqaure:  0.7464969361281693
Score:  [0.55804564 0.6578938  0.6924067  0.69825277 0.64863875 0.61203154
 0.690369   0.70582535 0.34257004 0.26969462]
corss val score mean:  0.5875728219200923
diffrence between r2 score - cross val score:  0.15892411420807695
--------------------------------------------------
modal  DecisionTreeRegressor()
R2 score:  0.5098024159101582
Mean Absoulte Error:  1548.111451495259
Mean Sqaure Error:  6490371.056917092
Root Mean Sqaure:  0.5098024159101582
Score:  [-0.10400389  0.37404138  0.47095891  0.46475228  0.15438802  0.31904286
  0.29301935  0.31088154 -0.08618736 -0.2778158 ]
corss val score mean:  0.1919077285312098
diffrence between r2 score - cross val score:  0.3178946873789484
--------------------------------------------------
modal  KNeighborsRegressor()
R2 score:  0.4956832263171249
Mean Absoulte Error:  1951.8746608315096
Mean Sqaure Error:  6677313.592857768
Root Mean Sqaure:  0.4956832263171249
Score:  [0.27753047 0.46052668 0.39841263 0.47239889 0.47045628 0.37203198
 0.48954163 0.419308   0.19530243 0.0524367 ]
corss val score mean:  0.3607945692492735
diffrence between r2 score - cross val score:  0.13488865706785141
```

```
--------------------------------------------------
modal  GradientBoostingRegressor()
R2 score:  0.6559106590708979
Mean Absoulte Error:  1593.009358593074
Mean Sqaure Error:  4555851.705198563
Root Mean Sqaure:  0.6559106590708979
Score:  [0.5684912  0.6652369  0.60533635 0.61242974 0.62901653 0.57870846
 0.65377529 0.65368681 0.39853    0.34272776]
corss val score mean:  0.5707939047853475
diffrence between r2 score - cross val score:  0.08511675428555032
--------------------------------------------------
modal  Ridge()
R2 score:  0.42838268993308914
Mean Absoulte Error:  2127.3969613557924
Mean Sqaure Error:  7568393.980928153
Root Mean Sqaure:  0.42838268993308914
Score:  [0.44774696 0.46243301 0.34408638 0.36929801 0.42455644 0.36238552
 0.44989361 0.40764183 0.33871746 0.20350707]
corss val score mean:  0.381026628507874
diffrence between r2 score - cross val score:  0.047356061425215146
--------------------------------------------------
modal  SVR()
R2 score:  0.03781250103285516
Mean Absoulte Error:  2854.874569120757
Mean Sqaure Error:  12739666.814594595
Root Mean Sqaure:  0.03781250103285516
Score:  [-0.04450143 -0.07646474  0.02511525 -0.02090893 -0.02177242  0.03769476
  0.02090925 -0.00066306  0.00802546 -0.01986504]
corss val score mean:  -0.009243090447776403
diffrence between r2 score - cross val score:  0.04705559148063156
--------------------------------------------------
```

```
model  KNeighborsRegressor()
R2 score: 0.4261330099761089
Mean Absoulte Error:  346215.2375629406
Mean Sqaure Error:  918269512405.0607
Root Mean Sqaure: 0.4261330099761089
Score: [0.49476313 0.50380475 0.29380468 0.58503587 0.56844215 0.31260534
 0.46433207 0.45129611 0.63297159 0.41603055]
cross val score mean:  0.4723086232526013
diffrence between r2 score - cross val score:  -0.04617561327649239
--------------------------------------------------
model  GradientBoostingRegressor()
R2 score: 0.5741458992711275
Mean Absoulte Error:  278138.1629313522
Mean Sqaure Error:  681427655240.6637
Root Mean Sqaure: 0.5741458992711275
Score: [0.60621103 0.68253529 0.32824045 0.73542432 0.63779914 0.52323369
 0.7516615  0.76036327 0.79216272 0.74469759]
cross val score mean:  0.6562328998300534
diffrence between r2 score - cross val score:  -0.08208700055892593
--------------------------------------------------
model  Ridge()
R2 score: 0.2894636298655261
Mean Absoulte Error:  475688.62412464066
Mean Sqaure Error:  1136960127506.6926
Root Mean Sqaure: 0.2894636298655261
Score: [0.2510352  0.32465006 0.15671457 0.40969833 0.35672408 0.30426873
 0.3282089  0.32985214 0.35726959 0.37903057]
cross val score mean:  0.31974521641113657
diffrence between r2 score - cross val score:  -0.030281586545610473
--------------------------------------------------
```

```
# Hyper parameter tunning
```

```
param_grid = {
    'max_depth' : range(10,20),
    'criterion' : ['mse'],
    'max_features': ['auto', 'sqrt'],
    'min_samples_leaf': range(2,6),
}
gridSearchCV = GridSearchCV(RandomForestRegressor(),param_grid=param_grid,refit=True,verbose=3)
```

```
gridSearchCV.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 80 candidates, totalling 400 fits
[CV 1/5] END criterion=mse, max_depth=10, max_features=auto, min_samples_leaf=2; total time=   1.4s
[CV 2/5] END criterion=mse, max_depth=10, max_features=auto, min_samples_leaf=2; total time=   1.4s
[CV 3/5] END criterion=mse, max_depth=10, max_features=auto, min_samples_leaf=2; total time=   1.4s
[CV 4/5] END criterion=mse, max_depth=10, max_features=auto, min_samples_leaf=2; total time=   1.4s
[CV 5/5] END criterion=mse, max_depth=10, max_features=auto, min_samples_leaf=2; total time=   1.4s
[CV 1/5] END criterion=mse, max_depth=10, max_features=auto, min_samples_leaf=3; total time=   1.4s
[CV 2/5] END criterion=mse, max_depth=10, max_features=auto, min_samples_leaf=3; total time=   1.4s
```

```
y_pred = gridSearchCV.best_estimator_.predict(X_test)
```

```
r2_score(y_test,y_pred)
```

```
0.7365317902092343
```

```
# saving the model
```

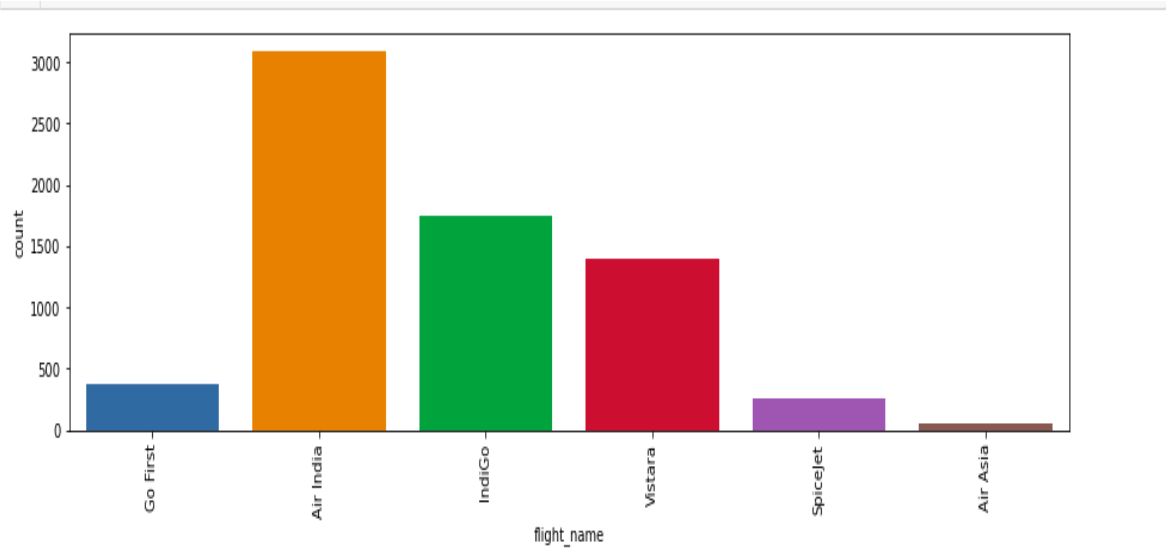
```
joblib.dump(gridSearchCV.best_estimator_,'RandomForestRegressor.modal')
```

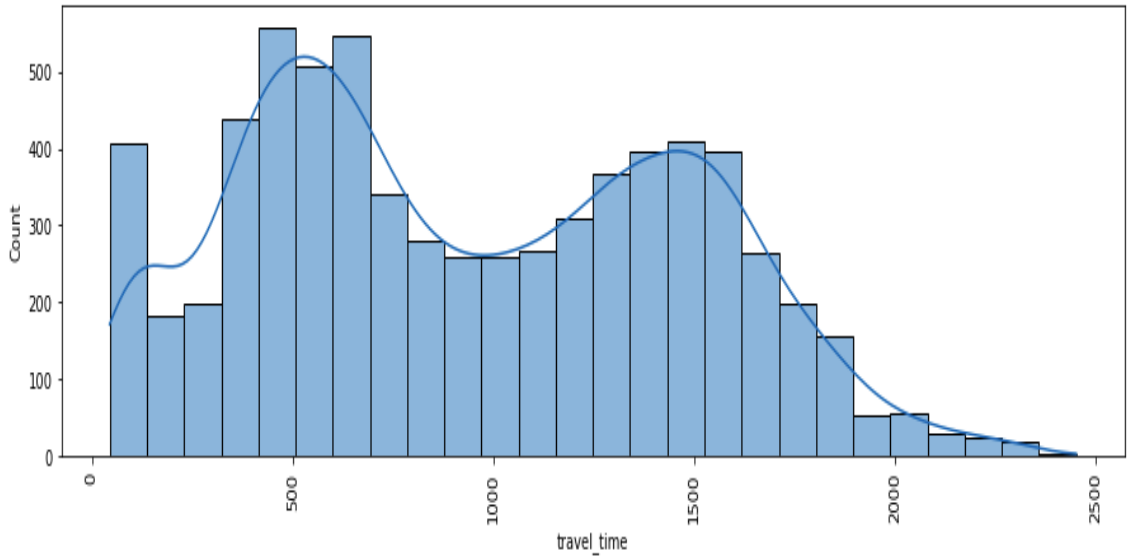
```
['RandomForestRegressor.modal']
```

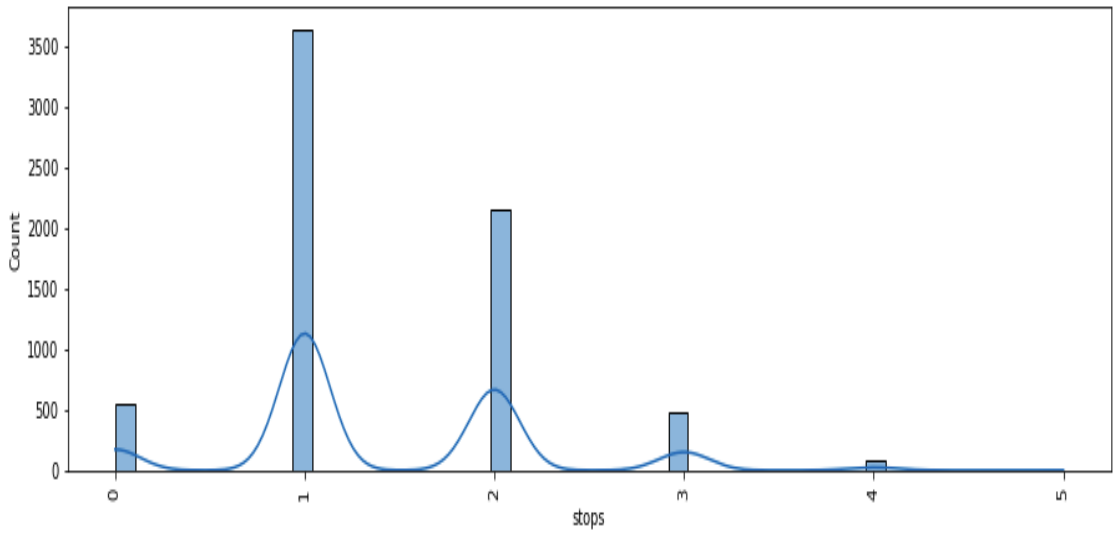
Key Metrics for success in solving problem under consideration

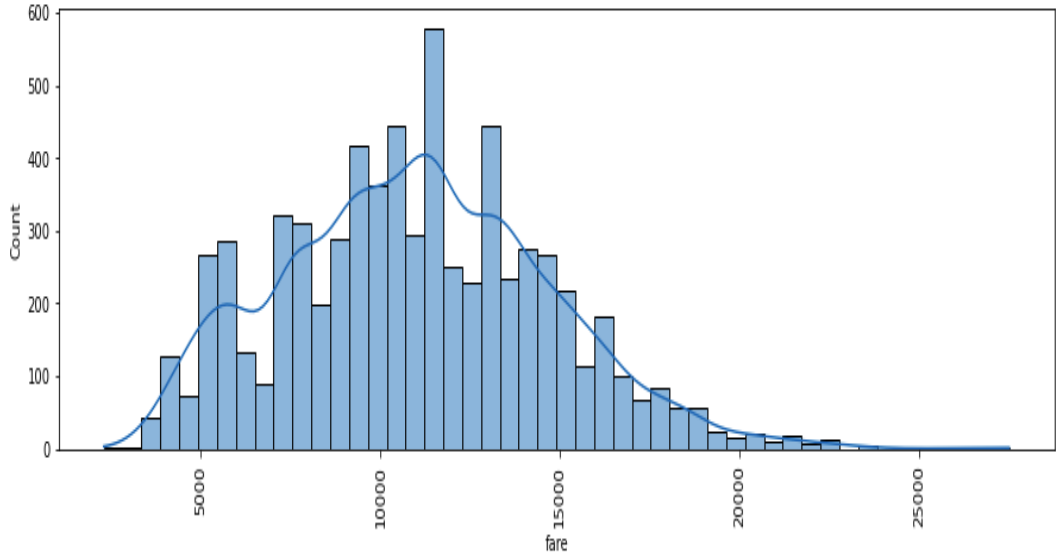
As this is a regression problem, we are required to predict the continuous feature (fare) I have used R2 score, mean absolute error, mean squared error and root mean squared error.

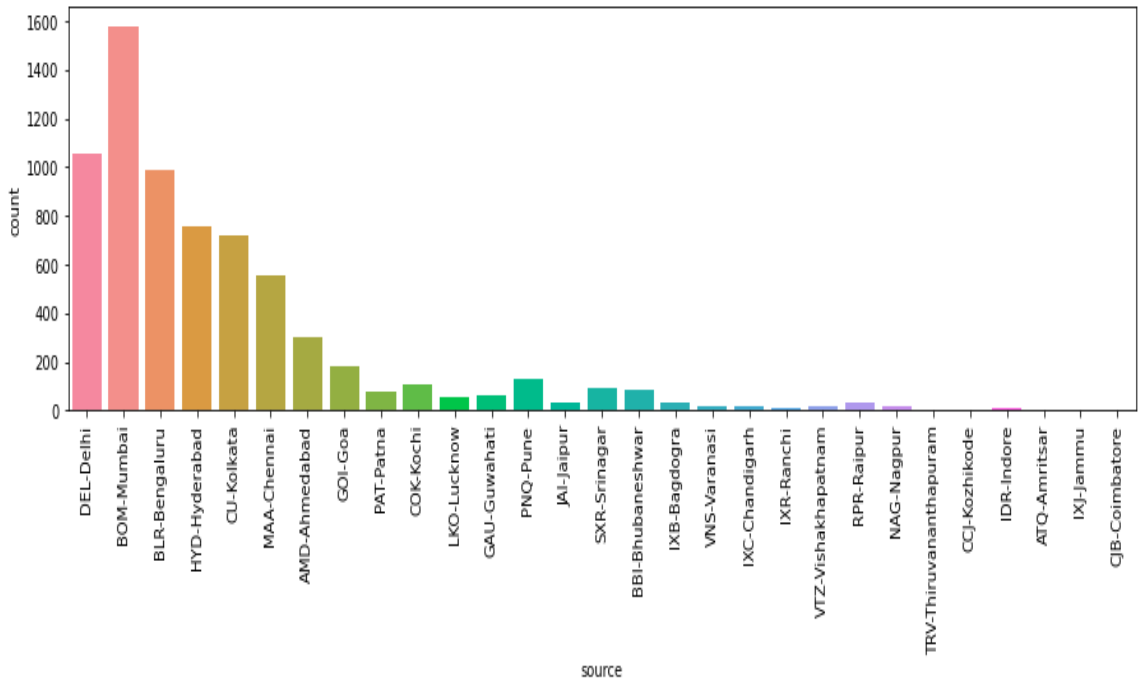
Visualizations

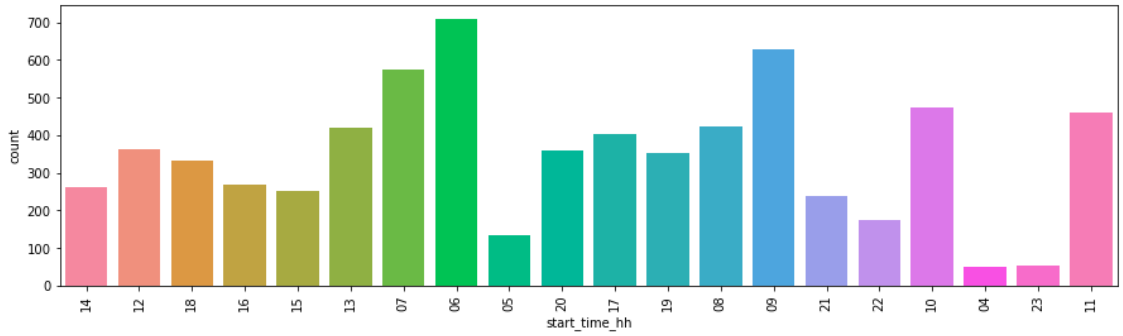


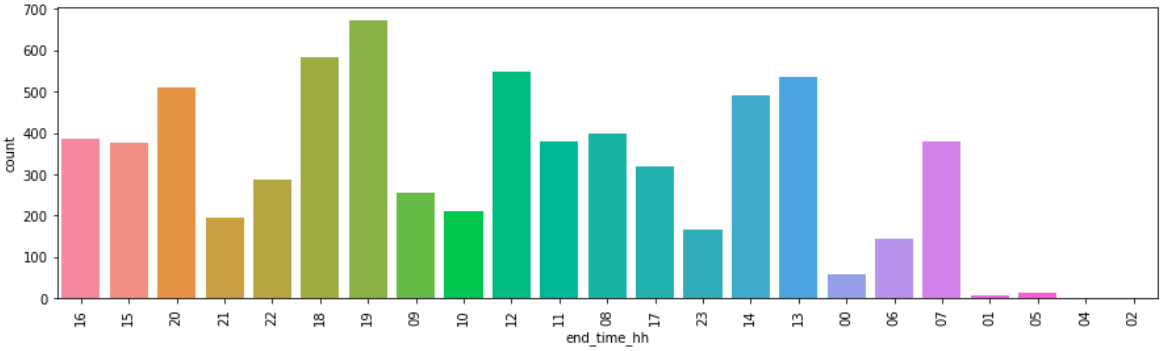

Count
500
400
300
200
100
0
0
500
1000
1500
2000
2500
travel_time

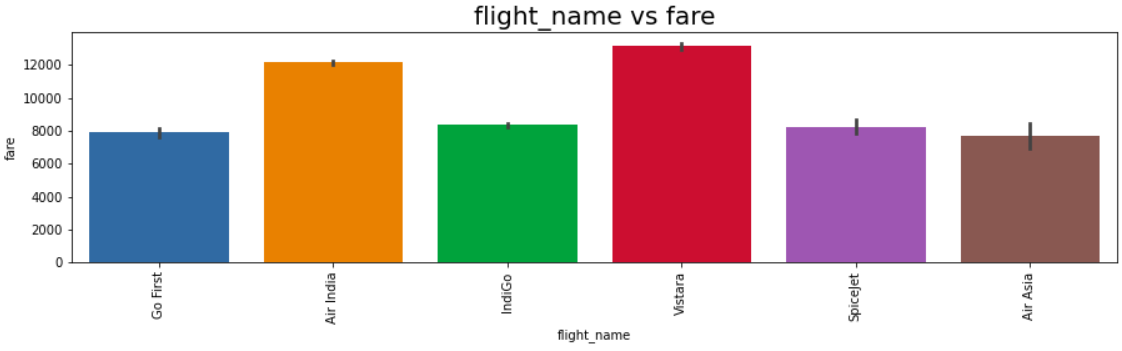

Count
3500
3000
2500
2000
1500
1000
500
0
0
1
2
3
4
5
stops

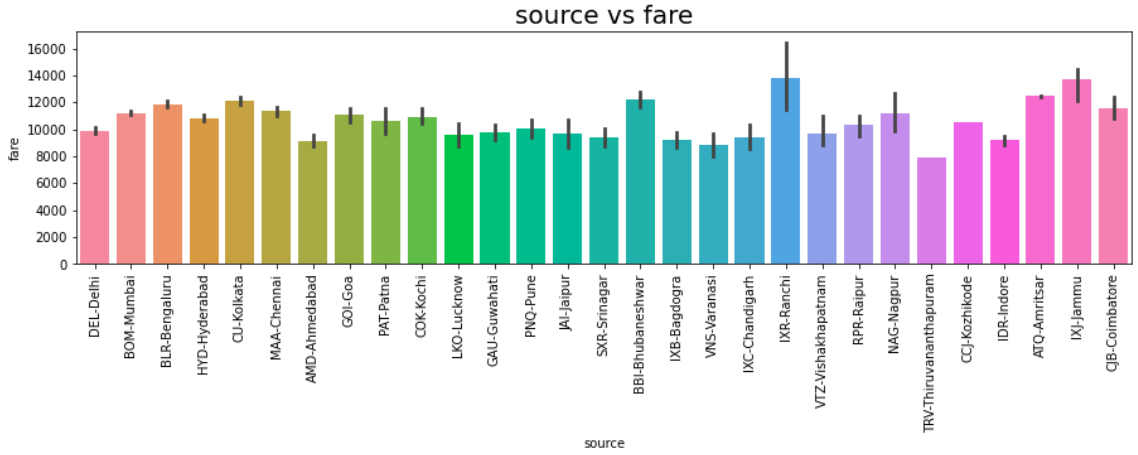

600
500
400
300
200
100
0
Count
5000
10000
15000
20000
25000
fare

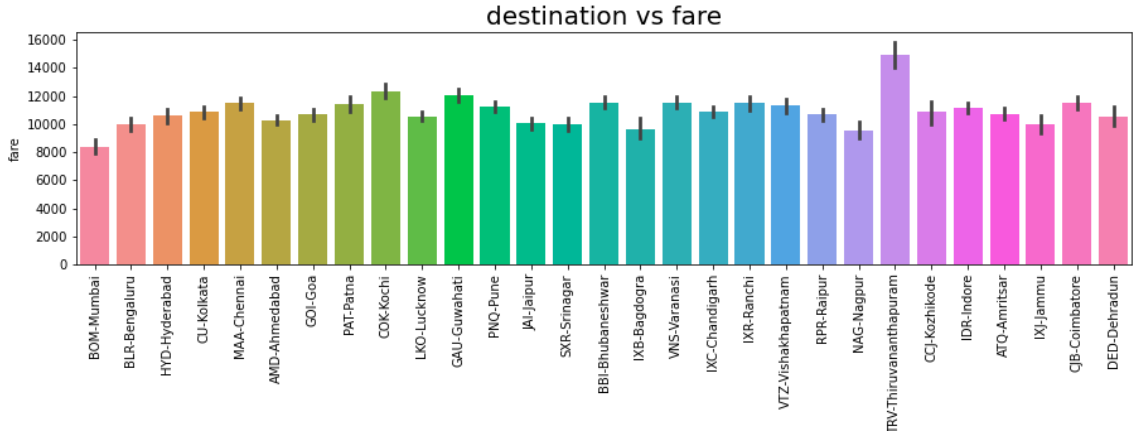

count
1600
1400
1200
1000
800
600
400
200
0
DEL-Delhi
BOM-Mumbai
BLR-Bengaluru
HYD-Hyderabad
CU-Kolkata
MAA-Chennai
AMD-Ahmedabad
GOI-Goa
PAT-Patna
COK-Kochi
LKO-Lucknow
GAU-Guwahati
PNQ-Pune
JAI-Jaipur
SXR-Srinagar
BBI-Bhubaneshwar
IXB-Bagdogra
VNS-Varanasi
IXC-Chandigarh
IXR-Ranchi
VTZ-Vishakhapatnam
RPR-Raipur
NAG-Nagpur
TRV-Thiruvananthapuram
CCJ-Kozhikode
IDR-Indore
ATQ-Amritsar
IXJ-Jammu
CJB-Coimbatore
source

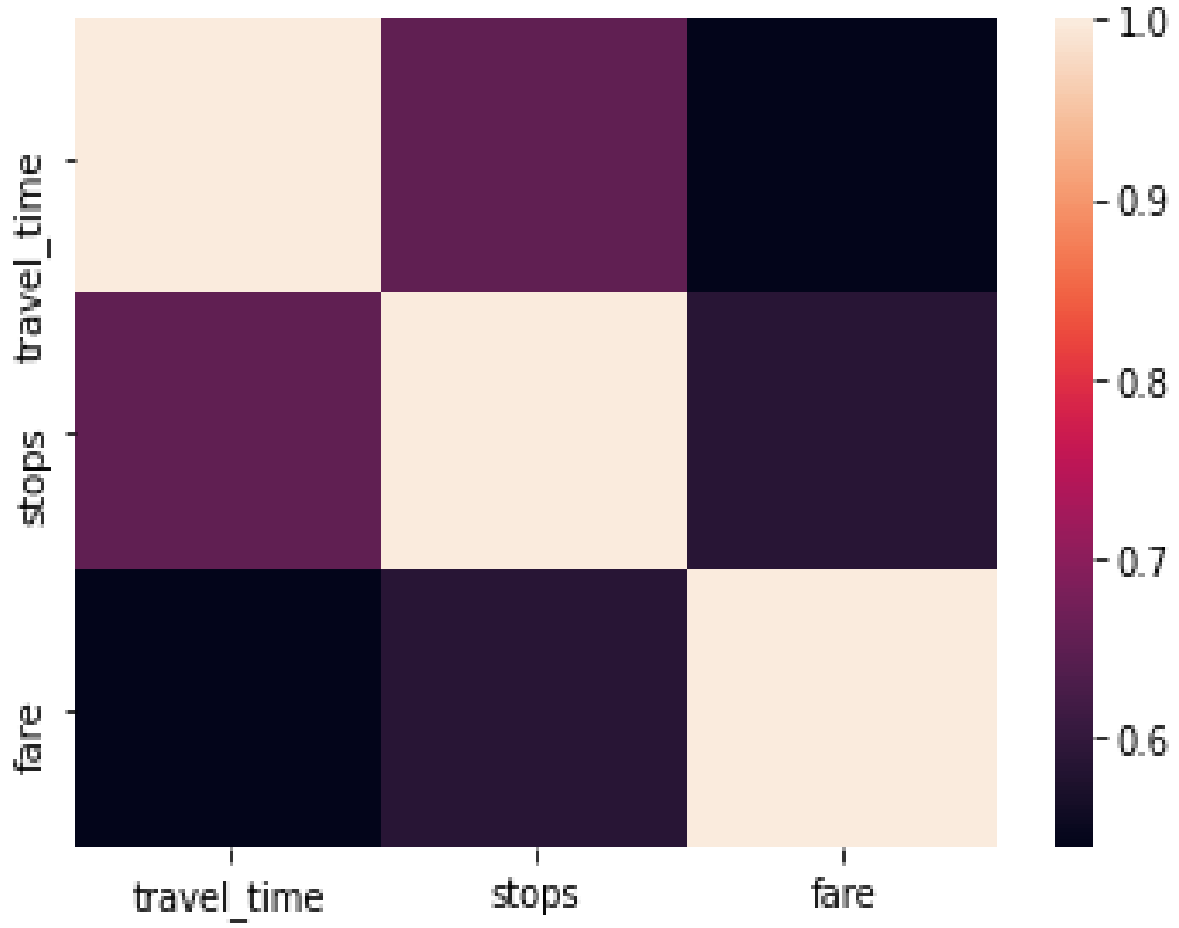

count
start_time_hh
14
12
18
16
15
13
07
06
05
20
17
19
08
09
21
22
10
04
23
11


count
end_time_hh
16
15
20
21
22
18
19
09
10
12
11
08
17
23
14
13
00
06
07
01
05
04
02


flight_name vs fare
fare
flight_name
Go First
Air India
IndiGo
Vistara
SpiceJet
Air Asia


source vs fare
fare
source
DEL-Delhi
BOM-Mumbai
BLR-Bengaluru
HYD-Hyderabad
CU-Kolkata
MAA-Chennai
AMD-Ahmedabad
GOI-Goa
PAT-Patna
COK-Kochi
LKO-Lucknow
GAU-Guwahati
PNQ-Pune
JAI-Jaipur
SXR-Srinagar
BBI-Bhubaneshwar
IXB-Bagdogra
VNS-Varanasi
IXC-Chandigarh
IXR-Ranchi
VTZ-Vishakhapatnam
RPR-Raipur
NAG-Nagpur
TRV-Thiruvananthapuram
CCJ-Kozhikode
IDR-Indore
ATQ-Amritsar
IXJ-Jammu
CJB-Coimbatore


destination vs fare
fare
BOM-Mumbai
BLR-Bengaluru
HYD-Hyderabad
CU-Kolkata
MAA-Chennai
AMD-Ahmedabad
GOI-Goa
PAT-Patna
COK-Kochi
LKO-Lucknow
GAU-Guwahati
PNQ-Pune
JAI-Jaipur
SXR-Srinagar
BBI-Bhubaneshwar
IXB-Bagdogra
VNS-Varanasi
IXC-Chandigarh
IXR-Ranchi
VTZ-Vishakhapatnam
RPR-Raipur
NAG-Nagpur
TRV-Thiruvananthapuram
CCJ-Kozhikode
IDR-Indore
ATQ-Amritsar
IXJ-Jammu
CJB-Coimbatore
DED-Dehradun


travel_time
stops
fare
travel_time
stops
fare
1.0
0.9
0.8
0.7
0.6

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.