Project Flight Price Prediction

Submitted By:

Aman Saxena

ACKNOWLEDGMENT

- https://scikit-learn.org/stable/ - For the libraries used in the project.
- Rest project is done by myself only.

INTRODUCTION

- Business Problem Framing

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on -
1. Time of purchase patterns (making sure last-minute purchases are expensive)
2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases)

So, you have to work on a project where you collect data of flight fares with other features and work to make a model to predict fares of flights.
.

- Conceptual Background of the Domain Problem

For more understanding we can simply correlate it with the project of car price prediction.

- Motivation for the Problem Undertaken

In this project we have to build the model that will predict price of the flight ticket from source to the destination using different independent variables.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  First of all we scraped the data from different websites and create an excel file by storing the data into it.

  Then we load the excel file from the local. Then we check the shape of the dataset and then describe the dataset from that we have gather some information. Then we check the datatypes of the columns and found all of the columns are of object datatype. After that we check for the columns which have '-' values but we did not them. Then we check the correlation of the columns with the target variable. Then we change the datatype of the price column to int as we have to predict it so it has to in int datatype.Then the visualization of the categorical columns were done.After that we check for the outliers using the boxplot we found some outliers in the columns but we did not remove them as those were the categorical columns.

  Since the categorical values were present so we have to encode them to make the prediction for that we have used ordinal encoder.

- Then the dataset is divided into dependent and independent variables. Then the data is trained on different models and Random forest regressor is giving the best accuracy and the other

metrices are also good in that so we chose it our best model .Then the hyperparameter tuning is done .

.

- Data Sources and their formats

We load the datatset and check their datatypes so we observe that all the columns are of object datatype and then we do the encoding of the dataset so that we can predict that what is the price of the flight.

Data Preprocessing Done

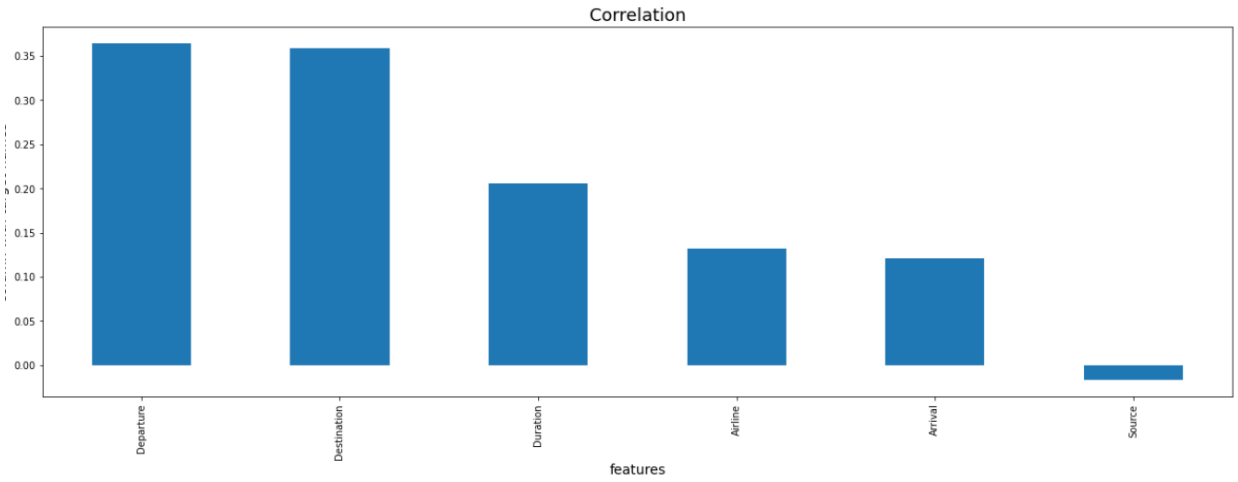
First of all we scraped the data from different websites and create an excel file by storing the data into it.

Then we load the excel file from the local. Then we check the shape of the dataset and then describe the dataset from that we have gather some information. Then we check the datatypes of the columns and found all of the columns are of object datatype. After that we check for the columns which have '-' values but we did not them. Then we check the correlation of the columns with the target variable. Then we change the datatype of the price column to int as we have to predict it so it has to in int datatype.Then the visualization of the categorical columns were done.After that we check for the outliers using the boxplot we found some outliers in the columns but we did not remove them as those were the categorical columns.

Since the categorical values were present so we have to encode them to make the prediction for that we have used ordinal encoder.

Then the dataset is divided into dependent and independent variables. Then the data is trained on different models and Random forest regressor is giving the best accuracy and the other metrices are also good in that so we chose it our best model .Then the hyperparameter tuning is done .

- Data Inputs- Logic- Output Relationships



In the above picture we can clearly see that what features are positively related with the target variable and what features are negatively related with the target variable.

- Hardware and Software Requirements and Tools Used

- import pandas as pd
- import numpy as np
- from sklearn.linear_model import LinearRegression

- from sklearn.linear_model import Lasso,Ridge
- from sklearn.model_selection import train_test_split,GridSearchCV
- from sklearn.preprocessing import StandardScaler
- import matplotlib.pyplot as plt
- import seaborn as sns
- import warnings
- warnings.filterwarnings('ignore')

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

Since the problem is of regression and we have to predict the price of the flight so we have used Linear regression, Lasso regression, Ridge regression and Random forest regressor for our project.

Testing of Identified Approaches (Algorithms)

- Linear Regression

- Lasso Regression

- Random forest regressor
- Ridge regression

- Run and Evaluate selected models
  - Linear Regression

Linear Regression

```
LR.fit(x_train,y_train)
y_pred=LR.predict(x_test)
print(r2_score(y_test,y_pred))
```

```
0.36555518374538454
```

  - Lasso Regression

Lasso

```
parameters={'alpha':[.0001,.001,.01,.1,1,10],'random_state':list(range(0,10)),'max_iter':[100,200,500,1000],'normalize':[Tru
ls=Lasso()
clf=GridSearchCV(ls,parameters)
clf.fit(x_new,y)
print(clf.best_params_)
```

```
{'alpha': 10, 'max_iter': 100, 'normalize': True, 'random_state': 0}
```

```
ls=Lasso(alpha=10,random_state=0,max_iter=100,normalize=True)
ls.fit(x_new,y)
```

```
Lasso(alpha=10, max_iter=100, normalize=True, random_state=0)
```

```
ls_pred=ls.predict(x_test)
lss=r2_score(y_test,ls_pred)
lss
```

```
0.37372751395969217
```

➢ Random Forest regressor

```
from sklearn.ensemble import RandomForestRegressor
```

```
parameters={'criterion':['mse','mae'],'max_features':['auto','sqrt','log2']}
rf=RandomForestRegressor()
clf=GridSearchCV(rf,parameters)
clf.fit(x_train,y_train)
```

```
GridSearchCV(estimator=RandomForestRegressor(),
             param_grid={'criterion': ['mse', 'mae'],
                         'max_features': ['auto', 'sqrt', 'log2']})
```

```
print(clf.best_params_)
```

```
{'criterion': 'mse', 'max_features': 'auto'}
```

```
rf=RandomForestRegressor(criterion='mse',max_features='auto')
rf.fit(x_train,y_train)

```

```
RandomForestRegressor()
```

```
rf_pred=rf.predict(x_test)
rfs=r2_score(y_test,rf_pred)
rfs
```

```
0.8568493594193562
```

➢ Ridge Regression

Ridge

```
parameters={'alpha':[.0001,.001,.01,.1,1,10],'random_state':list(range(0,10)),'max_iter':[100,200,500,1000],'normalize':[Tru
Rg=Ridge()
clfR=GridSearchCV(Rg,parameters)
clfR.fit(x_new,y)
print(clfR.best_params_)
```

```
{'alpha': 1, 'max_iter': 100, 'normalize': True, 'random_state': 0}
```

```
Rgg=Ridge(alpha=1,random_state=0,max_iter=100,normalize=True)
Rgg.fit(x_new,y)
```

```
Ridge(alpha=1, max_iter=100, normalize=True, random_state=0)
```

```
Rg_pred=Rgg.predict(x_test)
Rgs=r2_score(y_test,Rg_pred)
Rgs
```

```
0.2832413074870206
```

- Key Metrics for success in solving problem under consideration

  R2_score → for calculating the accuracy

CONCLUSION

First of all we scraped the data from different websites and create an excel file by storing the data into it.

Then we load the excel file from the local. Then we check the shape of the dataset and then describe the dataset from that we have gather some information. Then we check the datatypes of the columns and found all of the columns are of object datatype. After that we check for the columns which have '-' values but we did not them. Then we check the correlation of the columns with the target variable. Then we change the datatype of the price column to int as we have to predict it so it has to in int datatype.Then the visualization of the categorical columns were done.After that we check for the outliers using the boxplot we found some outliers in the columns but we did not remove them as those were the categorical columns.

Since the categorical values were present so we have to encode them to make the prediction for that we have used ordinal encoder.

- Then the dataset is divided into dependent and independent variables. Then the data is trained on different models and Random forest regressor is giving the best accuracy and the other metrices are also good in that so we chose it our best model .Then the hyperparameter tuning is done .