FLIGHT PRICE PREDICTION

Submitted by:

Komal Vijay Ghatvilkar

ACKNOWLEDGMENT

The success & outcome of this project were possible by the guidance and support from FlipRobo.

It was not possible to done without research from different machine learning sites on Google.

I referred DataTrained material for more information that helped me completion of the project.

Thank you.....!!!

The flight ticket buying system is to purchase a ticket many days prior to flight takeoff so as to stay away from the effect of the most extreme charge. Mostly, aviation routes don't agree this procedure. Plane organizations may diminish the cost at the time, they need to build the market and at the time when the tickets are less accessible. They may maximize the costs. So the cost may rely upon different factors. To foresee the costs this venture uses AI to exhibit the ways of flight tickets after some time. All organizations have the privilege and opportunity to change it's ticket costs at anytime. Explorer can set aside cash by booking a ticket at the least costs. People who had travelled by flight frequently are aware of price fluctuations. The airlines use complex policies of Revenue Management for execution of distinctive evaluating systems. The evaluating system as a result changes the charge depending on time, season, and festive days to change the header or footer on successive pages. The ultimate aim of the airways is to earn profit whereas the customer searches for the minimum rate. Customers usually try to buy the ticket well in advance of departure date so as to avoid hike in airfare as date comes closer.

I have collected the data from one of the sites 'cleartrip.com' which deals in hotels, flights booking activities for all over world.

We have done the following analysis of the dataset where we Imported necessary libraries so that we can work on datasets with the Jupyter notebook.

```
- import numpy as np
- import pandas as pd
- import matplotlib.pyplot as plt
- import seaborn as sns
- import warnings
- warnings.filterwarnings('ignore')
```

Data contains 2034 entries each having 11 variables. After reading the dataset I proceed with the EDA.

```
- df=pd.read_excel(r'C:\Users\HP\Desktop\FlightPricePredictionData.xlsx')
```

I checked the description of data with .info() method.

```
- df.info()
```

I removed unwanted or unnecessary columns to perform further tasks.

```
- df=df.drop(['Unnamed: 0','Departure Time','Arrival Time'],axis=1,inplace=True)
```

To perform further task replcaed some values in Price column and tried with conversion of String data into Integers for equalize the data type for process with LabelEncoder.

```
- df['Price'] = df["Price"].str.replace("₹","")
- df['Price'] = df["Price"].str.replace(",","")
- df['Price'] = df['Price'].astype(int)

- from sklearn.preprocessing import LabelEncoder
- LE=LabelEncoder()
- df['Airline Name']=LE.fit_transform(df['Airline Name'])
- df['Source']=LE.fit_transform(df['Source'])
- df['Duration']=LE.fit_transform(df['Duration'])
- df['Total Stops']=LE.fit_transform(df['Total Stops'])
- df['Route']=LE.fit_transform(df['Route'])
- df['Destination']=LE.fit_transform(df['Destination'])
- df['Date Of Journey']=LE.fit_transform(df['Date Of Journey'])
```

After .describe() done found the statistical description of data & found no null values so performed the task further.

```
- df.describe()
- df.isnull().sum()
```

With the correlation among all the columns checked the correlation and found some of the data is positively correlated and some is negatively correlated with each other.

```
- df.corr()
```

In data visualization done the following visualizations :

First used Correlation Matrix for showing the correlation between all columns with Heatmap.

From the output of correlation matrix, we can see that it is symmetrical i.e. the bottom left is same as the top right and negatively correlated.

```
- corr_mat=df.corr()
- # Size of the canvas
- plt.figure(figsize=[30,10])
- #Plot Correlation Matrix
- sns.heatmap(corr_mat,annot=True) # annot represnts each value encoded in heatmap
- plt.title('Correlation Matrix')
- plt.show()
```

The result of the Correlation Matrix is on following GitHub link.

https://github.com/komalghatvilkar/Internship/blob/main/Flight%20Price%20Prediction%20Project/Correlation%20Matrix.png

Second, used Density plot & Histogram both for all dataset for visualizing the data individually.

The visualization plot shows that each variable distributed differently and as we can see some data has categorical values, so used histogram also for better understanding to show the distribution.

```
- df.plot(kind='density',subplots=True,layout=(5,10),sharex=False,fontsize=1,figsize
  =(30,20))
- plt.show()
- #plot histogram data vizualization
- df.hist(bins=20,figsize=(30,20))
- #plot showing
- plt.show()
```

The result of the Density plot & Histogram is on following GitHub link.

https://github.com/komalghatvilkar/Internship/blob/main/Flight%20Price%20Prediction%20Project/Density%20Plot.png

https://github.com/komalghatvilkar/Internship/blob/main/Flight%20Price%20Prediction%20Project/Histogram.png

Then I checked the skewness of the data & removed the same. After that checked the outliers if any, found very less so didn't removed the outliers. I used boxplot to check the outliers in dataset.

```
- df.skew().sort_values(ascending=False) # For descending
- from sklearn.preprocessing import power_transform
- df_new=power_transform(df)
- df=pd.DataFrame(df_new,columns=df.columns)
- df.skew().sort_values(ascending=False) # For descending

- df.boxplot(figsize=[20,15])
- plt.subplots_adjust(bottom=0.25)
- plt.show()
```

The result of the Boxplot is on following GitHub link.

https://github.com/komalghatvilkar/Internship/blob/main/Flight%20Price%20Prediction%20Project/Boxplot-Outliers.png

After checking the outliers We split the data into x & y. I have taken the output as the column "Price" because our problem statement is to find the price of flight booking.

```
- x=df.drop(['Price'],axis=1)
- y=df['Price']
```

And then did the Train Test Split and find the best accuracy & random state which gives me the following results :-

```
- # Training process
- # Min-max scaler
```

```
- from sklearn.preprocessing import MinMaxScaler
- mms=MinMaxScaler()
- from sklearn.linear_model import LinearRegression
- lr=LinearRegression()
- from sklearn.metrics import r2_score
- from sklearn.model_selection import train_test_split
- For i in range(0,100):
- x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.20,random_state=
  i)
- lr.fit(x_train,y_train) # Fitting the data will train the model
- pred_train=lr.predict(x_train) # Predicting the data # Predicted traget variable
- pred_test=lr.predict(x_test)
- print(f'At Random State {i}, the training accuracy is :-
  {r2_score(y_train,pred_train)}')
- print(f'At Random State {i}, the training accuracy is :-
  {r2_score(y_test,pred_test)}')
- print("\n")
```

- At Random State 97, the training accuracy is :- 0.2467505138803625 At Random State 97, the training accuracy is :- 0.2763466453342366
- At Random State 98, the training accuracy is :- 0.24687243880221865 At Random State 98, the training accuracy is :- 0.28344225089136144
- At Random State 99, the training accuracy is :- 0.23047102647669837 At Random State 99, the training accuracy is :- 0.33719557803155153

With the best suitable model building for regression dataset that is Linear Regression performed The task.

Next did Cross Validation with “cross_val_score” for the models used & it shows the output :-

```
- # Cross Validation
- Train_accuracy = r2_score(y_train,pred_train)
- Test_accuracy = r2_score(y_test,pred_test)
- from sklearn.model_selection import cross_val_score
- for i in range(2,8):
- cv_score=cross_val_score(lr,x,y,cv=4)
- cv_mean=cv_score.mean()
- print(f'At cross fold {i} the cv score is  {cv_mean} and accuracy score for
  training is {Train_accuracy} and accuracy score for testing is {Test_accuracy}')
- print('\n')
```

- At cross fold 5 the cv score is -1.4913060902449504 and accuracy score for training is 0.23047102647669837 and accuracy score for testing is 0.33719557803155153

- At cross fold 6 the cv score is -1.4913060902449504 and accuracy score for training is 0.23047102647669837 and accuracy score for testing is 0.33719557803155153
- At cross fold 7 the cv score is -1.4913060902449504 and accuracy score for training is 0.23047102647669837 and accuracy score for testing is 0.33719557803155153

Then with the matplotlib.pyplot showed the distribution of data & the result shows best fit line & relationship between two variables.

```
- import matplotlib.pyplot as plt
- plt.figure(figsize=(8,6))
- plt.scatter(x=y_test,y=pred_test,color='blue')
- plt.plot(y_test,y_test,color='black')
- plt.xlabel('Actual price', fontsize=14)
- plt.ylabel('Predicted price', fontsize=14)
- plt.title('Linear Regression', fontsize=18)
- plt.show()
```

Then checked the regularization with GridSearchCV & With Lasso technique to perform regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

```
- # Regularization
- from sklearn.model_selection import GridSearchCV
- from sklearn.model_selection import cross_val_score
- from sklearn.linear_model import Lasso
- parameters={'alpha':[.0001,.001,.01,.1,1,10],'random_state':list(range(0,20))}
- ls=Lasso()
- clf=GridSearchCV(ls,parameters)
- clf.fit(x_train,y_train)
- print(clf.best_params_)
- ls=Lasso(alpha=10,random_state=0)
- ls.fit(x_train,y_train)
- ls.score(x_train,y_train)
- pred_ls=ls.predict(x_test)
- lss=r2_score(y_test,pred_ls)
- Lss
- cv_score=cross_val_score(ls,x,y,cv=5)
```

- `cv_mean=cv_score.mean()`
- `cv_mean*100`

With AdaBoostRegressor assemble technique to achieve best accuracy of dataset.

- `# Ensemble Technique`
- `from sklearn.ensemble import AdaBoostRegressor`
- `AD = AdaBoostRegressor()`
- `AD.fit(x_train,y_train)`
- `AD.score(x_test,y_test)`

Finally came to the Conclusion as per the results found those are the model is showing the result for the dataset with 70% accuracy but as per the observations the data is not suitable to work with or use because the data is not accurate enough to proceed when we see the accuracy and regularization done.

Please find the GitHub links for Pyplot of Linear regression to refer.

https://github.com/komalghatvilkar/Internship/blob/main/Flight%20Price%20Prediction%20Project/Linear%20Regression.png

Please find the GitHub links for Jupyter Notebook Solution of web scraping of data collected to refer.

https://github.com/komalghatvilkar/Internship/blob/main/Flight%20Price%20Prediction%20Project/Flight%20Price%20Prediction%20-%20Data%20Collection.ipynb

Please find the GitHub links for Jupyter Notebook Solution of dataset to refer.

https://github.com/komalghatvilkar/Internship/blob/main/Flight%20Price%20Prediction%20Project/Flight%20Price%20Prediction%20Project.ipynb

The results shows that the dataset is 70% accurate & we can't proceed with the data accordingly.

Thank you...!!