FLIGHT PRICE PREDICTION PROJECT

Submitted by:

MIENGANDHA SINHA

ACKNOWLEDGEMENT

I would like to express my gratitude to the Company to give this project to me. In making of this project I hereby used to take help from the references which is given by the company as sample documentation and details related to project and professionals and SME guided me a lot in the project and the other previous projects helped me and guided me in completion of the project.

INTRODUCTION

- **Flight Price Prediction Problem**

This project aims to predict the fare of flight. As we'll know that anyone who booked a flight ticket 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- Time of purchase patterns (making sure last-minute purchases are expensive), 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, the prediction of the price of a flight by taking its Company name, its flight price, its flight code and many other parameters it can be predicted. It required to the making of model of the price of flights with the available independent variables. It will be used by the management to understand how exactly the prices vary with the independent variables to meet certain price levels.

- **Conceptual Background of the Domain Problem**

A primary objective of this project is to estimate used flight fare by using some attributes that are highly correlated with a fares. As it is said in the statement that flight ticket prices vary unexpectedly over time. So, to work on a project where collect the data of flight fares with the other features and work to make a model to predict fares of flights. By doing some research on this project, are able to train the model and predicting things make the previous background to work efficiently.

- **Review of Project**

With the help of analysing the flight fare using various machine learning by using essential data analysis techniques then will draw some predictions about the price of the flight based on some features such as what type of airline it is, what is the arrival time, what is the .departure time, what is the duration of flight, source, destination and many other parameters. From the collecting data phase gives important independent variables. The model building do all the data visualizations, data pre-processing steps, evaluating the model, data cleaning and selecting the best model for the project.

- **Motivation for the Problem Undertaken**

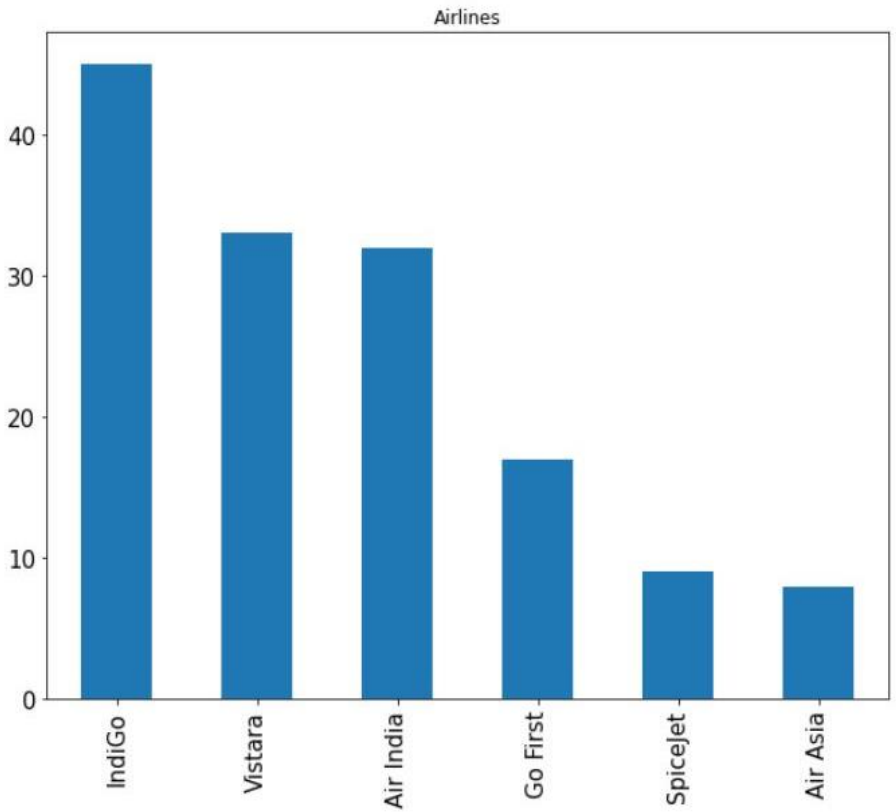
  Here, the datasets have total of 144 entries with 10 columns, no null values, EDA has to be performed to see whether it gain or loss in the independent variable and its compare to the price among every aspects, to build machine learning models, to determine the optimal values of Hyper parameters and the selection of the best model, by predicting of the value can help to the clients and for the further change in the data.

ANALYTICAL PROBLEM FRAMING

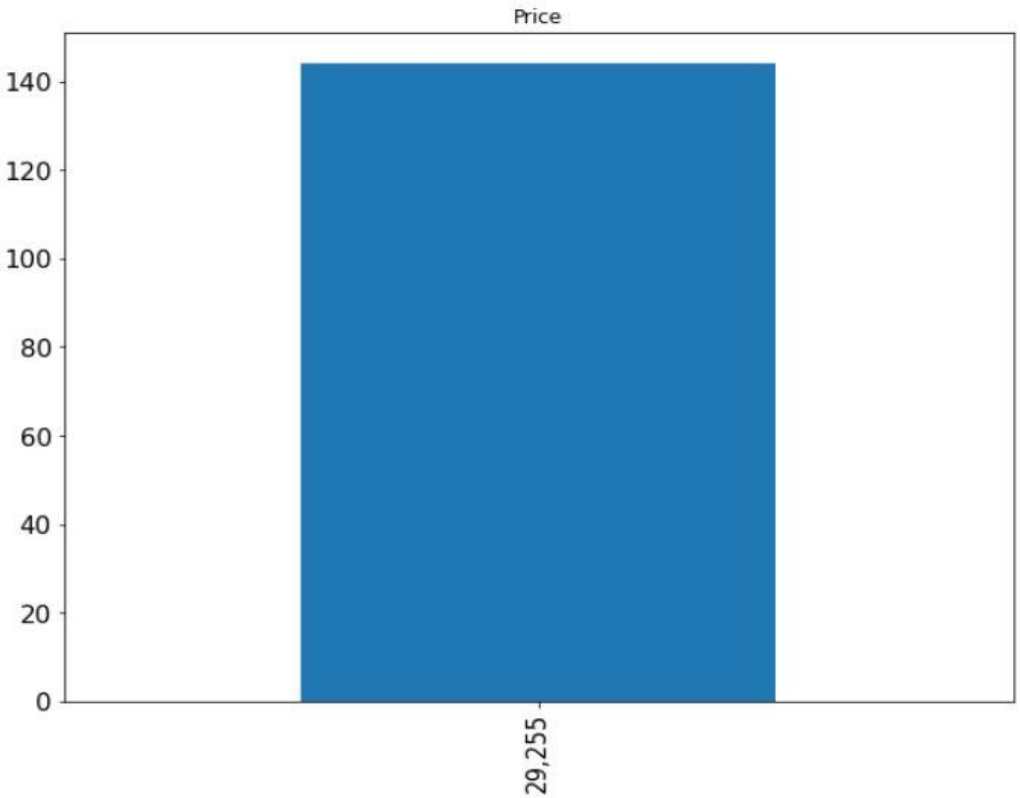
- **Mathematical/Analytical Modelling of the Problem**

  By checking the null values found that having no null values in the datasets, the type of data frame is in pandas, data frame info tells that int64(1 variable), object(9 variables), by using the data visualization they are:

1. Indigo becoming as a most popular Airline

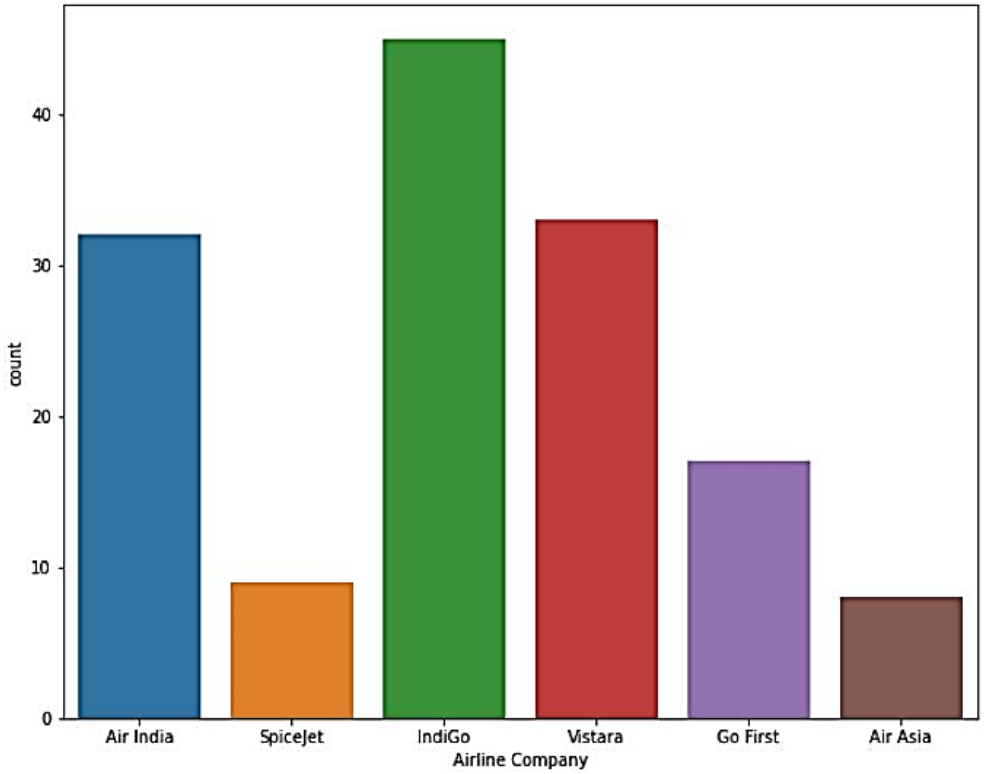


2. It is about the fare of flight from Delhi to Mumbai

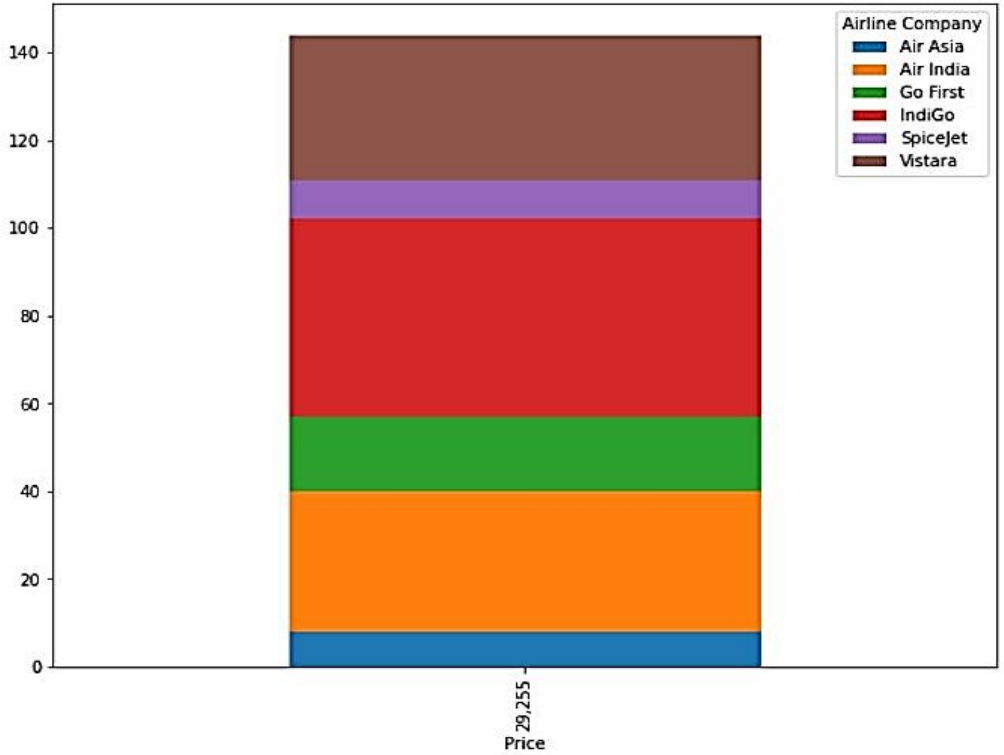


3. As it is shown below:

- **IndiGo has Maximum Price range**
- **Vistara and IndiGo have maximum price when compared to others.**
- **SpiceJet, AirAsia, GoFirst and AirIndia has some similarities in prices.**



4. Flight price stacked with their flight name



• Data Sources and their formats

The data sources and their formats are from .csv file.

```
df=pd.read_csv(r'C:/Users/user/Desktop/FRTech Internship Project/FLIGHT PRICE CASE STUDY/flight price prediction.csv')
df.head()
```

	Unnamed: 0	Airline Company	Flight Code	Departure Time	Source City	Arrival Time	Destination City	Duration	Total Stops	Price
0	0	Air India	-6238/5371	08:30	New Delhi	14:45	Mumbai	6h 15m	1 Stop	29,255
1	1	Air India	-6238/5371	08:30	New Delhi	14:45	Mumbai	6h 15m	1 Stop	29,255
2	2	Air India	-6238/5371	08:30	New Delhi	14:45	Mumbai	6h 15m	1 Stop	29,255
3	3	Air India	-6238/5371	08:30	New Delhi	14:45	Mumbai	6h 15m	1 Stop	29,255
4	4	Air India	-6238/5371	08:30	New Delhi	14:45	Mumbai	6h 15m	1 Stop	29,255

• Data Pre- Processing Done

The steps followed for the cleaning of the data is Label Encoder after the importing pre-processing there transform the target columns into features then lastly it has to set for the data frame.

DATA PRE-PROCESSING

```
from sklearn import preprocessing
features=df.drop(['Duration','Price','Total Stops'],axis=1)
target=df['Price']
col_names=list(features.columns)
scaler=preprocessing.StandardScaler()
features=scaler.fit_transform(features)
features=pd.DataFrame(features,columns=col_names)
features.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Airline Company	144.0	7.999003e-17	1.00349	-1.78619	-1.146361	0.133298	0.773127	1.412957
Flight Code	144.0	0.000000e+00	0.00000	0.00000	0.000000	0.000000	0.000000	0.000000
Departure Time	144.0	0.000000e+00	0.00000	0.00000	0.000000	0.000000	0.000000	0.000000
Source City	144.0	0.000000e+00	0.00000	0.00000	0.000000	0.000000	0.000000	0.000000
Arrival Time	144.0	0.000000e+00	0.00000	0.00000	0.000000	0.000000	0.000000	0.000000
Destination City	144.0	0.000000e+00	0.00000	0.00000	0.000000	0.000000	0.000000	0.000000

- **Data Inputs-Logic-Output Relationships**

The relationships between inputs and outputs can be studied extracting weights of the trained model. Regression is that relationships between them can be blocky or highly structured based on the training data. It requires the data scientist to train the algorithm with both labeled inputs and desired outputs.

- **State the set of assumptions(if any) related to the problem under consideration**

Presumptions are by using regression label encoding, classifier, selection of the best models, various machine learning to predict that it means the relationship between the dependent and independent variables look fairly linear. Thus, our linearity assumption is satisfied.

- **Hardware and Software Requirements and Tools Used**

By importing many libraries are:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import pickle
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score

import warnings
warnings.filterwarnings('ignore')
```

MODEL/s DEVELOPMENT AND EVALUATION

- **Identification of possible problem-solving approaches(methods)**

  The collection and interpretation of data in order to uncover patterns and trends. It is a component of data analytics. Statistical analysis can be used in situations like gathering research interpretations, statistical modelling or designing surveys and studies. The approaches/methods of identification are descriptive and inferential statistics which are describes as the properties of sample and population data, and inferential statistics which uses those properties to test hypotheses and draw efficient conclusions in terms of outputs.

- **Testing of Identified Approaches(Algorithms)**

  There is no outliers

- **Run and Evaluate selected models**

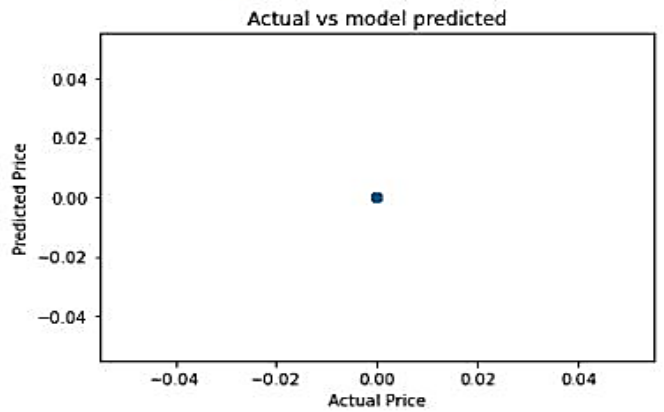
  DATA SCALING

```
from sklearn.preprocessing import StandardScaler
# Data Scaling Formula Z=(x-mean)/std
scaler = StandardScaler()
X_scaled=scaler.fit_transform(x)
X_scaled
```

```
array([[-1.14636101,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [-1.14636101,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [-1.14636101,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       ...,
       [ 0.13329779,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.77312719,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.13329779,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ]])
```

```
plt.scatter(y_test,y_pred)
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.title('Actual vs model predicted')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



MODEL EVALUATION

```
[ ]:
1 from sklearn.metrics import mean_squared_error, mean_absolute_error
2 y_pred=regression.predict(x_test)
```

```
[ ]:
1 mean_absolute_error(y_test,y_pred)
```

```
[ ]: 0.0
```

```
[ ]:
1 mean_squared_error(y_test,y_pred)
```

```
[ ]: 0.0
```

```
[ ]:
1 np.sqrt(mean_squared_error(y_test,y_pred))
```

```
[ ]: 0.0
```

MODEL BUILDING

Model Name: LinearRegression()
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 1.0
Root Mean Squared Log Error (RMSLE): -inf
Mean Absolute Percentage Error (MAPE): nan %
Adj R Square: 1.0

Model Name: DecisionTreeRegressor()
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 1.0
Root Mean Squared Log Error (RMSLE): -inf
Mean Absolute Percentage Error (MAPE): nan %
Adj R Square: 1.0

Model Name: RandomForestRegressor()
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 1.0
Root Mean Squared Log Error (RMSLE): -inf
Mean Absolute Percentage Error (MAPE): nan %
Adj R Square: 1.0

Model Name: KNeighborsRegressor()
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 1.0
Root Mean Squared Log Error (RMSLE): -inf
Mean Absolute Percentage Error (MAPE): nan %
Adj R Square: 1.0

Model Name: ExtraTreesRegressor()
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 1.0
Root Mean Squared Log Error (RMSLE): -inf
Mean Absolute Percentage Error (MAPE): nan %
Adj R Square: 1.0

Model Name: GradientBoostingRegressor(loss='ls')
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 1.0
Root Mean Squared Log Error (RMSLE): -inf
Mean Absolute Percentage Error (MAPE): nan %
Adj R Square: 1.0

Model Name: XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
early_stopping_rounds=None, enable_categorical=False,
eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
importance_type=None, interaction_constraints='',
learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
reg_lambda=1, ...)
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 0.0
Root Mean Squared Log Error (RMSLE): -35.937
Mean Absolute Percentage Error (MAPE): inf %
Adj R Square: -0.059259

Model Name: BaggingRegressor()
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 1.0
Root Mean Squared Log Error (RMSLE): -inf
Mean Absolute Percentage Error (MAPE): nan %
Adj R Square: 1.0

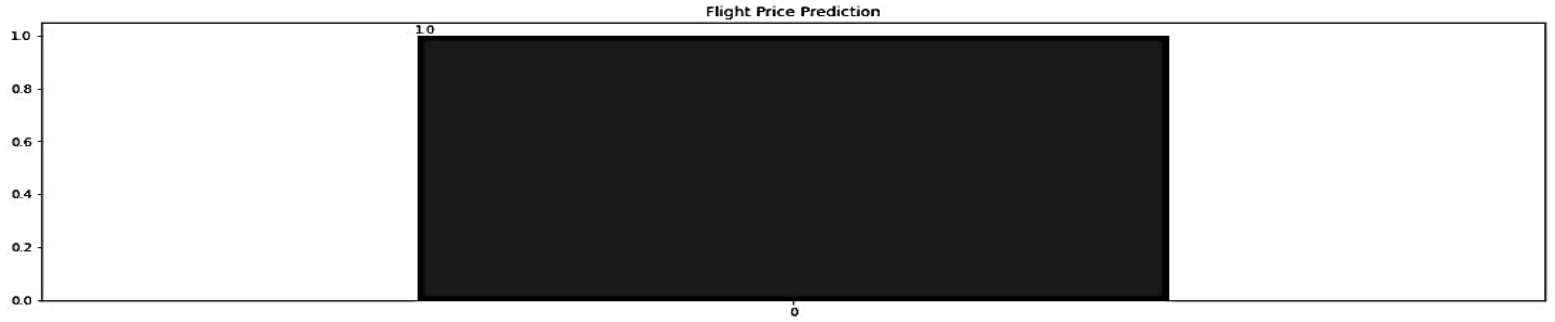
Model Name: Ridge()
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 1.0
Root Mean Squared Log Error (RMSLE): -inf
Mean Absolute Percentage Error (MAPE): nan %
Adj R Square: 1.0

Model Name: Lasso(alpha=0.1)
Mean Absolute Error (MAE): 0.0
Mean Squared Error (MSE): 0.0
Root Mean Squared Error (RMSE): 0.0
R2_score: 1.0
Root Mean Squared Log Error (RMSLE): -inf
Mean Absolute Percentage Error (MAPE): nan %
Adj R Square: 1.0

• Visualizations

```
plt.figure(figsize=(20,5))
ax=df.Price.value_counts(normalize=True).plot(kind='bar', color=['black', 'pink'], alpha=0.9, rot=0)
plt.title('Flight Price Prediction')
for i in ax.patches:
    ax.annotate(str(round(i.get_height(),2)),(i.get_x() * 1.01, i.get_height() * 1.01))

plt.show()
```

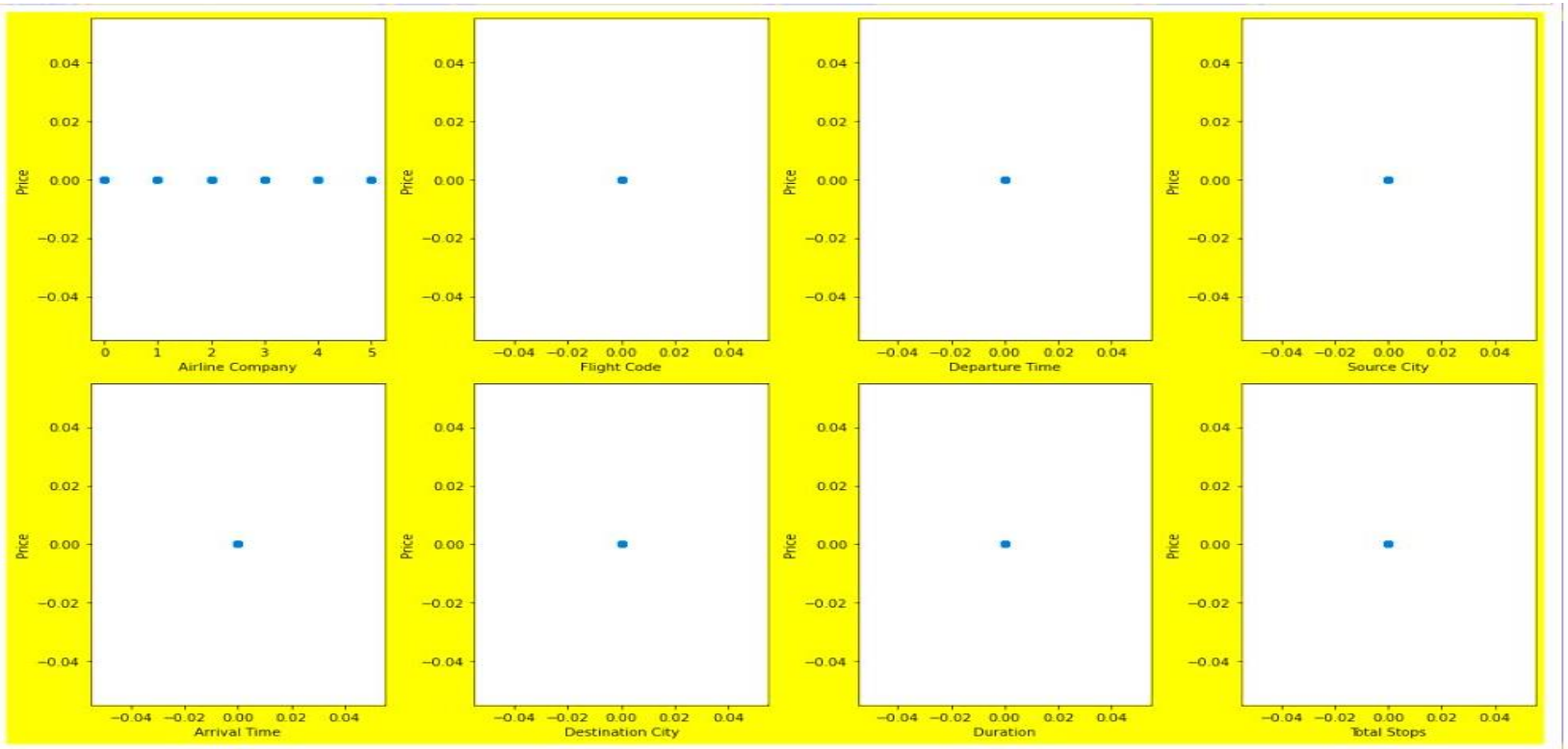


USING THE HEATMAP FOR THE DATASETS

```
plt.figure(figsize=(20,5))
sns.heatmap(df.isnull(),cbar=False,cmap='crest')
```

<AxesSubplot:>

VISUALIZING THE RELATIONSHIPS



The relationship between the dependent and independent variables look good in linear. Thus, our linearity assumption is satisfied.

• Interpretation of the Results

The results were interpreted from the visualizations, pre-processing and modelling:

1. The project is basically machine learning & statistic used methods for the implementation of the models & automation.

2. To understand and evaluate flight fares and to develop a strategy that utilizes data mining techniques to predict the fares, to guide the individuals booking the flight ticket.

3. The model of the independent variables and dependent variables are exactly vary with the variables.

4. It can accordingly manipulating the strategy of the areas that will yield high returns as it make easier for the individuals in booking the ticket.

5. For any prediction/classification problem, past flight prices for each route collected on a daily basis is needed. Manually collecting data daily is efficient to do work with the research data.

6. After have the data, need to clean & prepare the data accordingly to the model's requirements. In any ML problem, this is the step that is the most important and the most time consuming.

7. Data preparation is followed by analysing the data, uncovering hidden trends and then applying various predictive machine learning models on the training set.

8. By data visualizations there are many things to be noted when it will according to work each other it means that from the aspect it is going to work to predict the flight fares.

9. By pre-processing the data it means that from the help of label encoder helps the dataset column to transfer to fit another column into it.

10. Having built various machine learning models, to test models on testing set and come up with most suitable metric to calculate the accuracy. Moreover, many a times, merging models and predicting a cumulative target variable proves to be more accurate.

CONCLUSION

I would like to conclude here that by doing research in the topic by the time of purchase patterns really matters and keeping the flight as full as it want means by raising prices on a flight which is filling up in order to reduce sales and half back inventory for those expensive last minute purchases and that's why it is unexpectedly the fares vary. The cheapest available ticket on a given flight gets more and less expensive over time. By using some parameters to predict the data according to the statement. There are many things to be noted when it will according to work each other like data pre-processing, cleaning, visualizing, scaling and label encoder helps that dataset column to transform to fit another column into it. There are a few times when an offer is run by an airline because of which the prices drop suddenly. These are difficult to incorporate in our mathematical models, and hence lead to error. Along the Delhi-Mumbai route, we find that the price of flights increases or remains constant as the days to departure decreases. This because of the high frequency of the flights, high demand and also could be due to heavy competition. At last, after doing all this research, the model of the project are ready to analyse the independent and dependent variable.

THANK YOU