FLIGHT PRICE PREDICITON PROJECT

PROBLEM STATEMENT:

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)

UNDERSTANDING:

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.

EDA STEPS AND VISUALIZATIONS:

1. Data Collection

```
1 df=pd.read_csv(r'C:/Users/user/Desktop/FRTech Internship Project/FLIGHT PRICE CASE STUDY/flight price prediction.csv')
2 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

2. Data Cleaning

```
1 total=df.isnull().sum().sort_values(ascending=False)
2 percent = (df.isnull().sum()/df.isnull().count()).sort_values(ascending=False)
3 missing = pd.concat([total, percent], axis=1, keys=['Total' , 'Percent'])
4 missing.head()
```

	Total	Percent
Airline Company	0	0.0
Flight Code	0	0.0
Departure Time	0	0.0
Source City	0	0.0
Arrival Time	0	0.0

3. Univariate Analysis

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

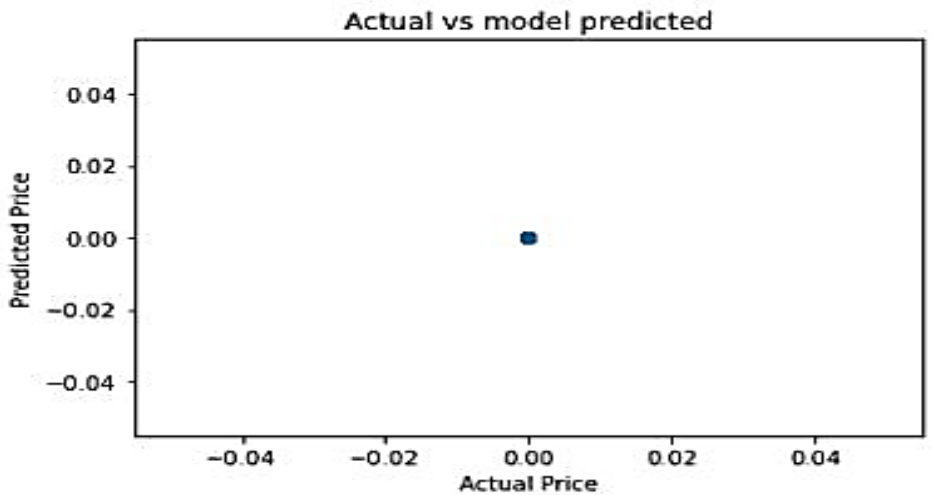
```
array([[-1.14636101,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [-1.14636101,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [-1.14636101,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.       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

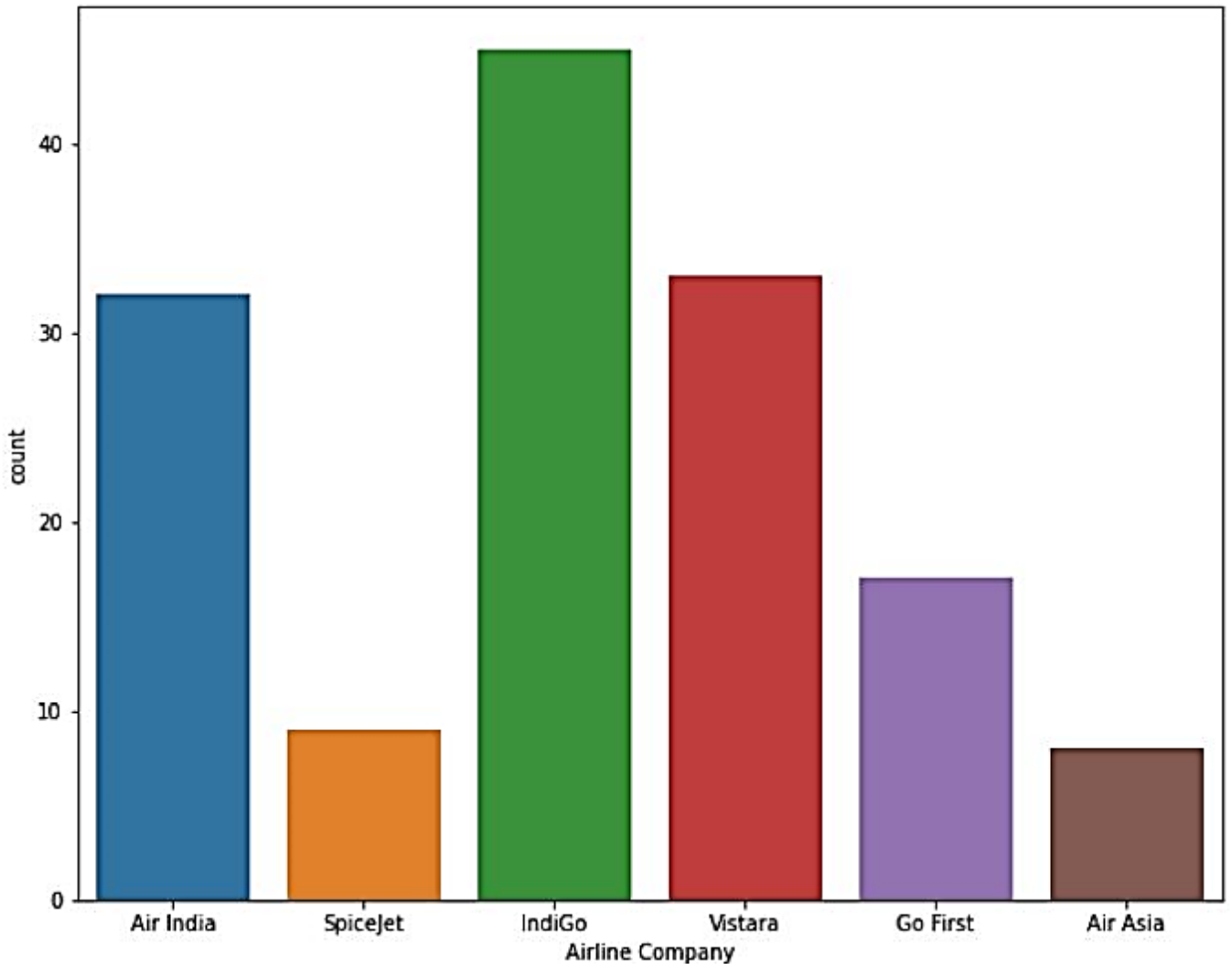
4. Bivariate Analysis

BIVARIATE ANALYSIS

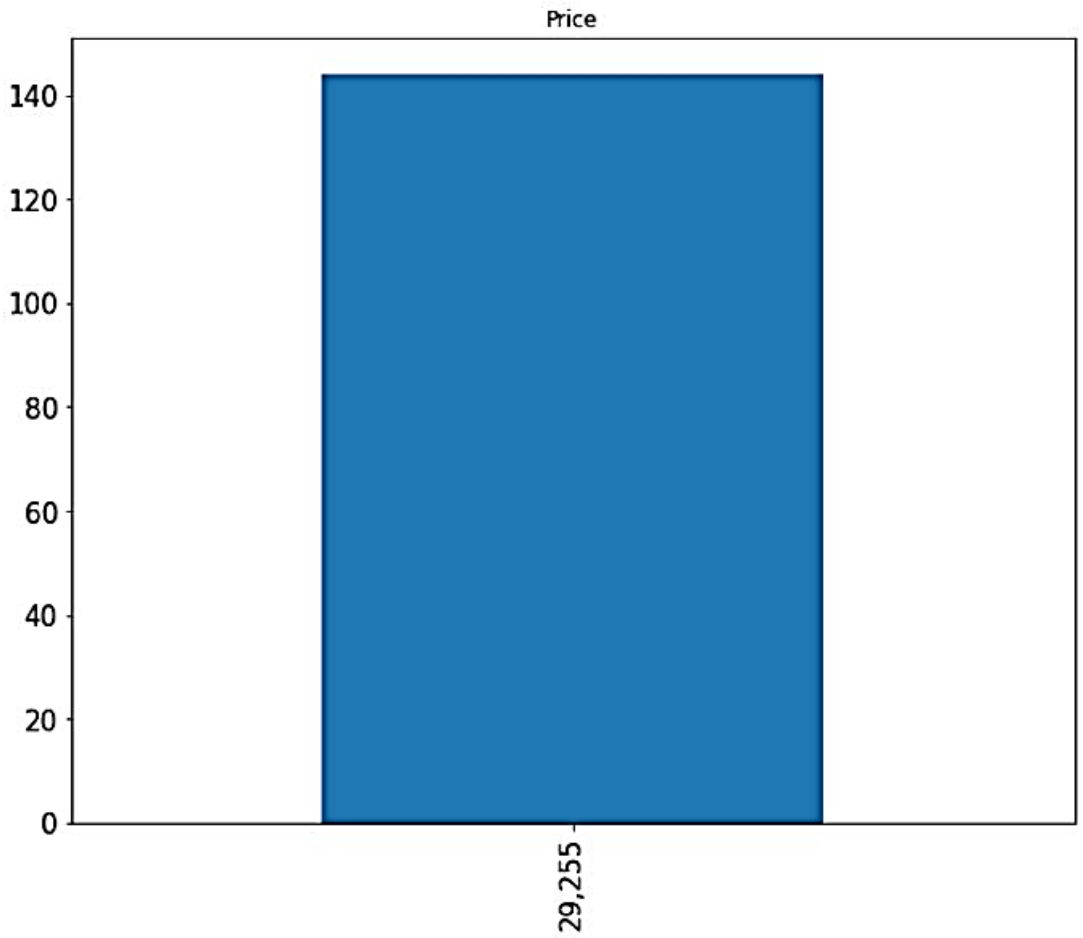
```
1  plt.scatter(y_test,y_pred)
2  plt.xlabel('Actual Price')
3  plt.ylabel('Predicted Price')
4  plt.title('Actual vs model predicted')
5  plt.show
```

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

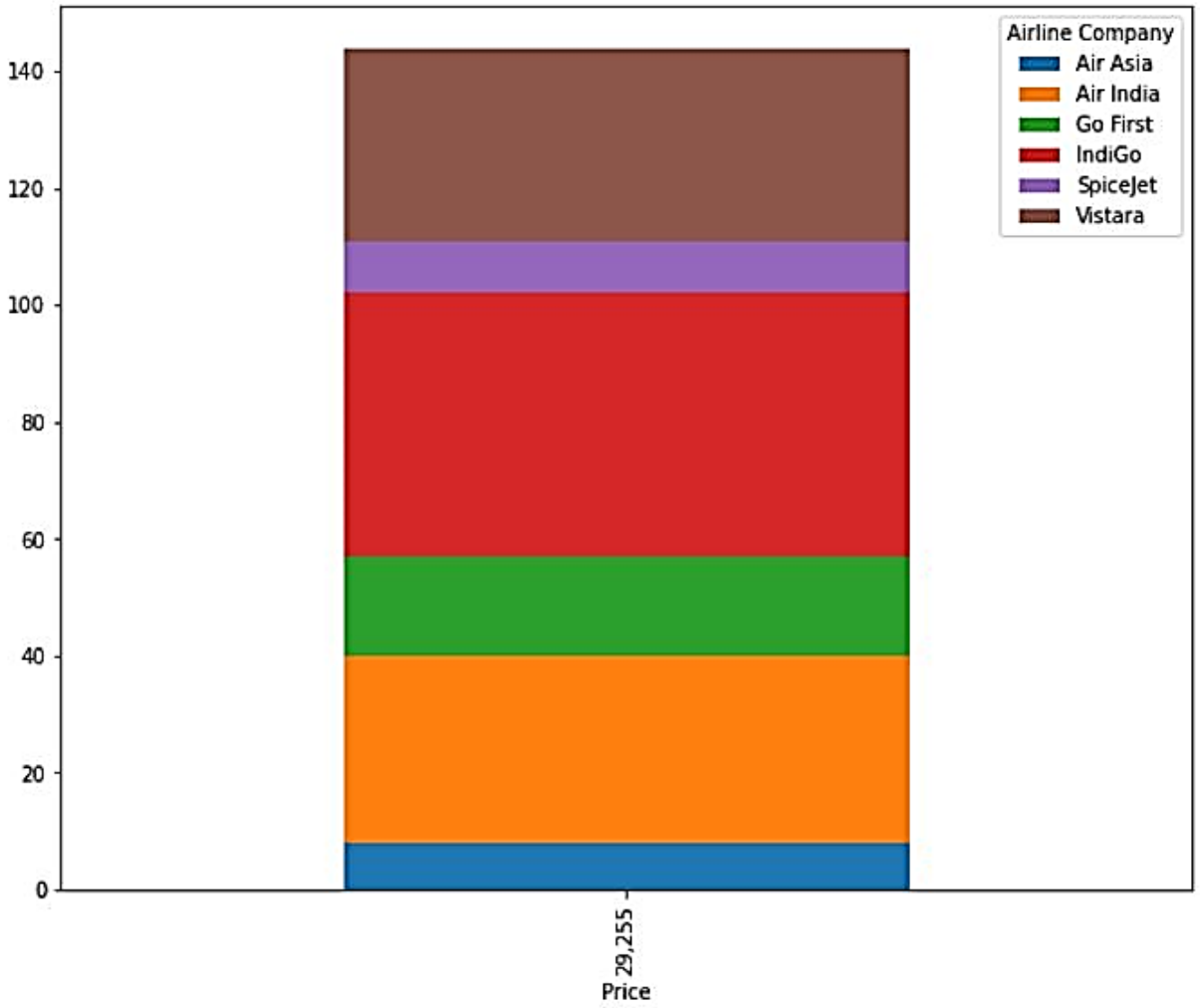




Indigo becoming as a most popular Airline and has maximum price range as compare to others



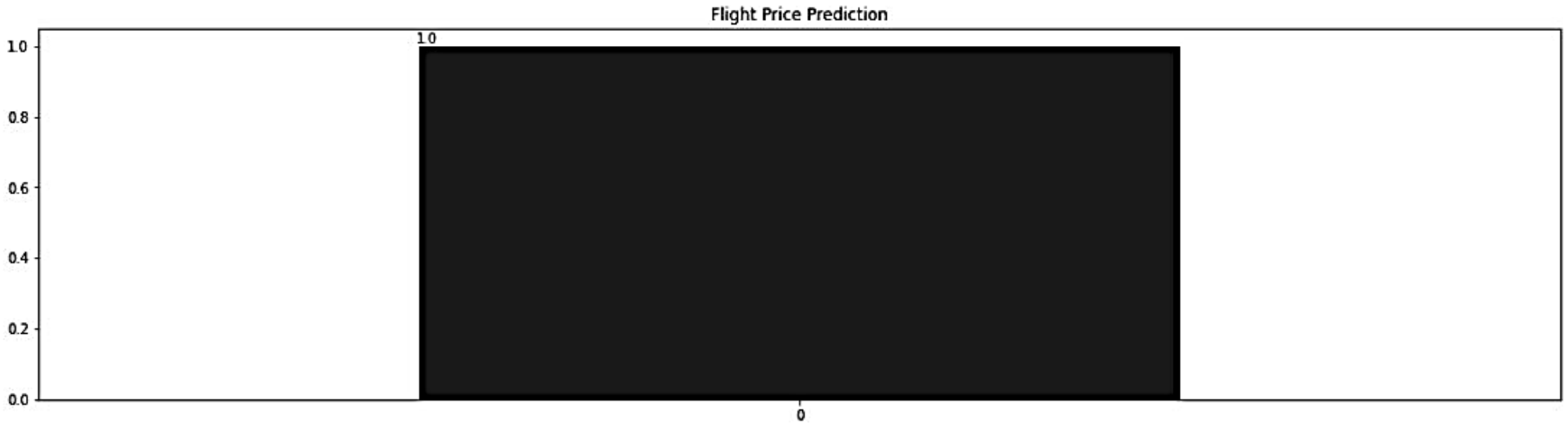
As Flight Ticket booked Delhi to Mumbai the ticket fare showing according to that



Flight price stacked with their flight name

```
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()
```

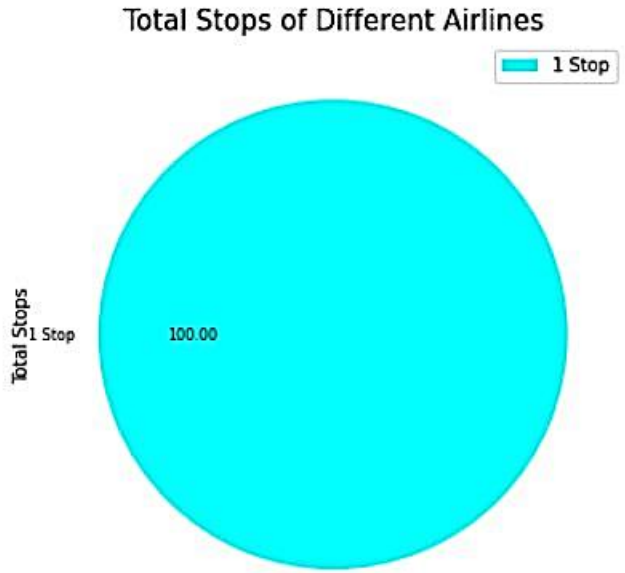



```
plt.figure(figsize=(20,5))
sns.heatmap(df.isnull(),cbar=False,cmap='crest')
```

<AxesSubplot:>

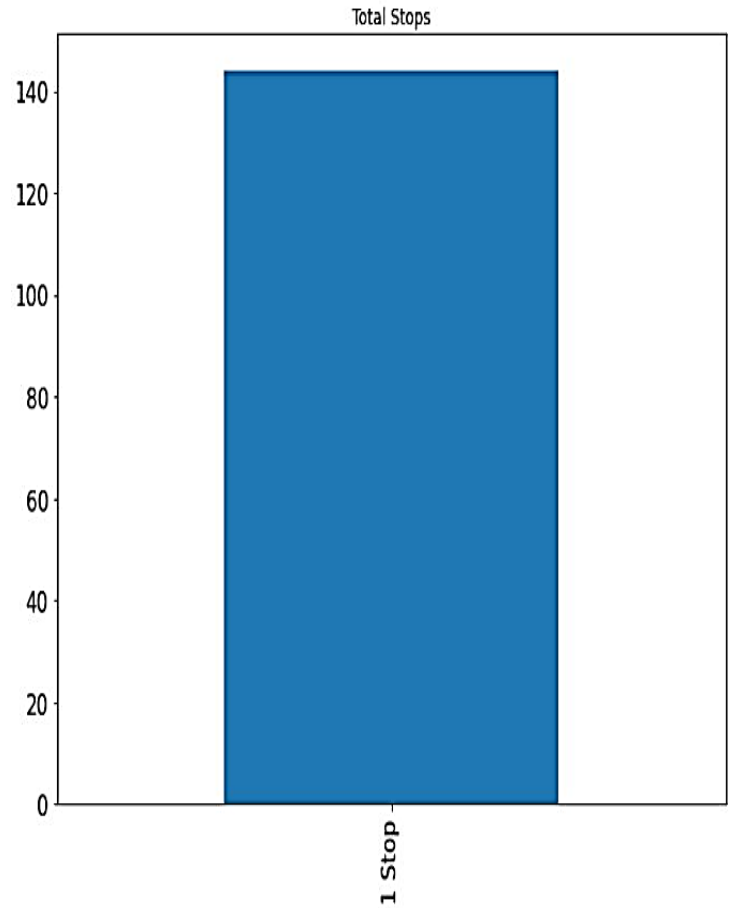
```
df2=df.groupby(['Flight Code' , 'Airline Company' , 'Total Stops'], as_index=False).count()
```

```
plt.figure(figsize=(8,6))
df2['Total Stops'].value_counts().plot(kind='pie',textprops={'color':'black'}, autopct='%0.2f',cmap='cool',fontsize=8)
plt.title('Total Stops of Different Airlines', fontsize=15)
plt.legend(['1 Stop' , 'Non-Stop'])
plt.show()
```

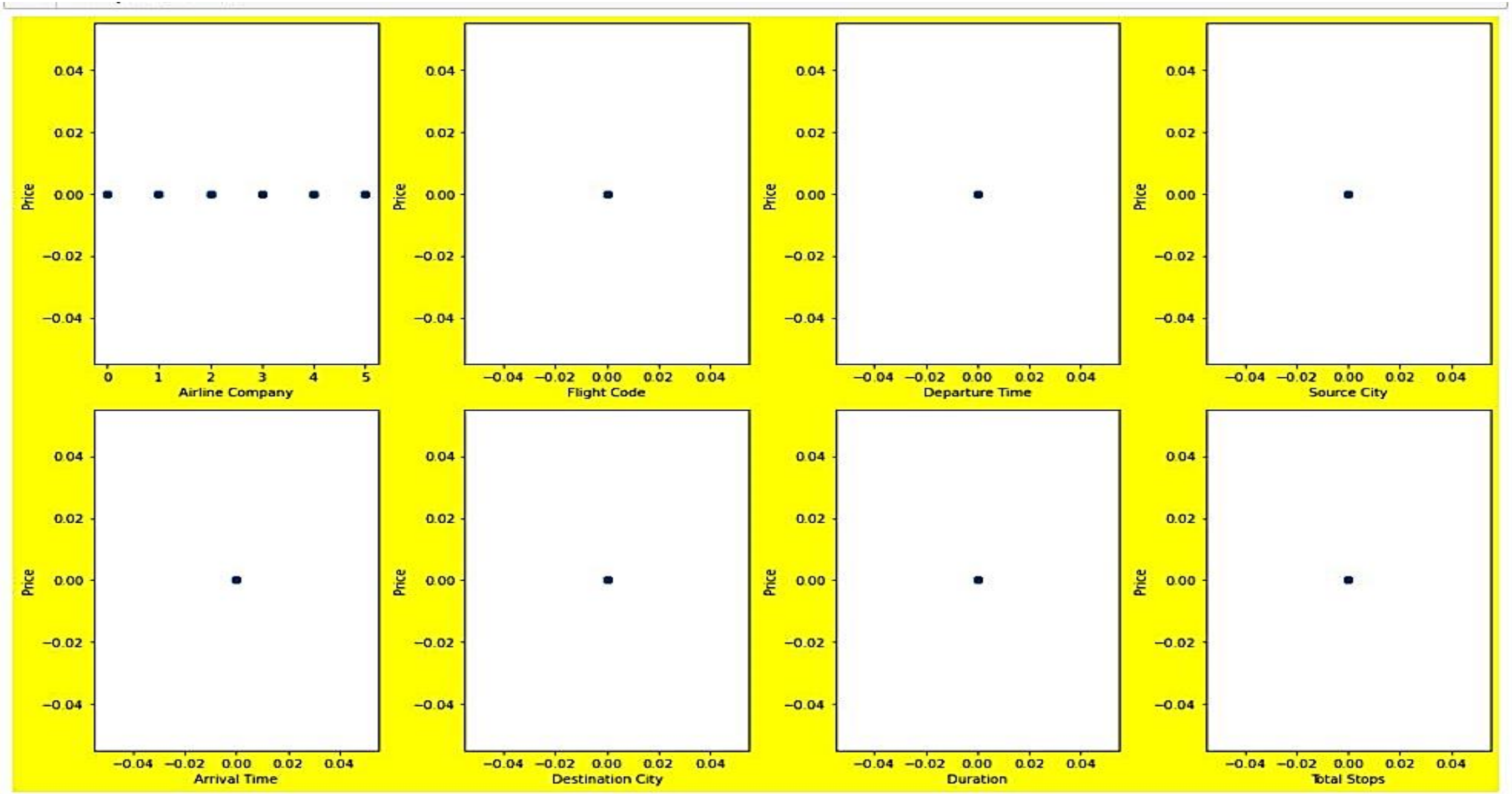



```
plt.subplot()
df['Total Stops'].value_counts().plot(kind='bar',title='Total Stops',figsize=(10,8),fontsize=15)
```

<AxesSubplot:title={'center':'Total Stops'}>



VISUALIZING THE RELATIONSHIP WITH PRICE



STEPS AND ASSUMPTIONS TO COMPLETE THE PROJECT:

- The project is basically machine learning & statistic used methods for the implementation of the models & automation.
- 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.
- The model of the independent variables and dependent variables are exactly vary with the variables.
- 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.

- 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.

- 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.

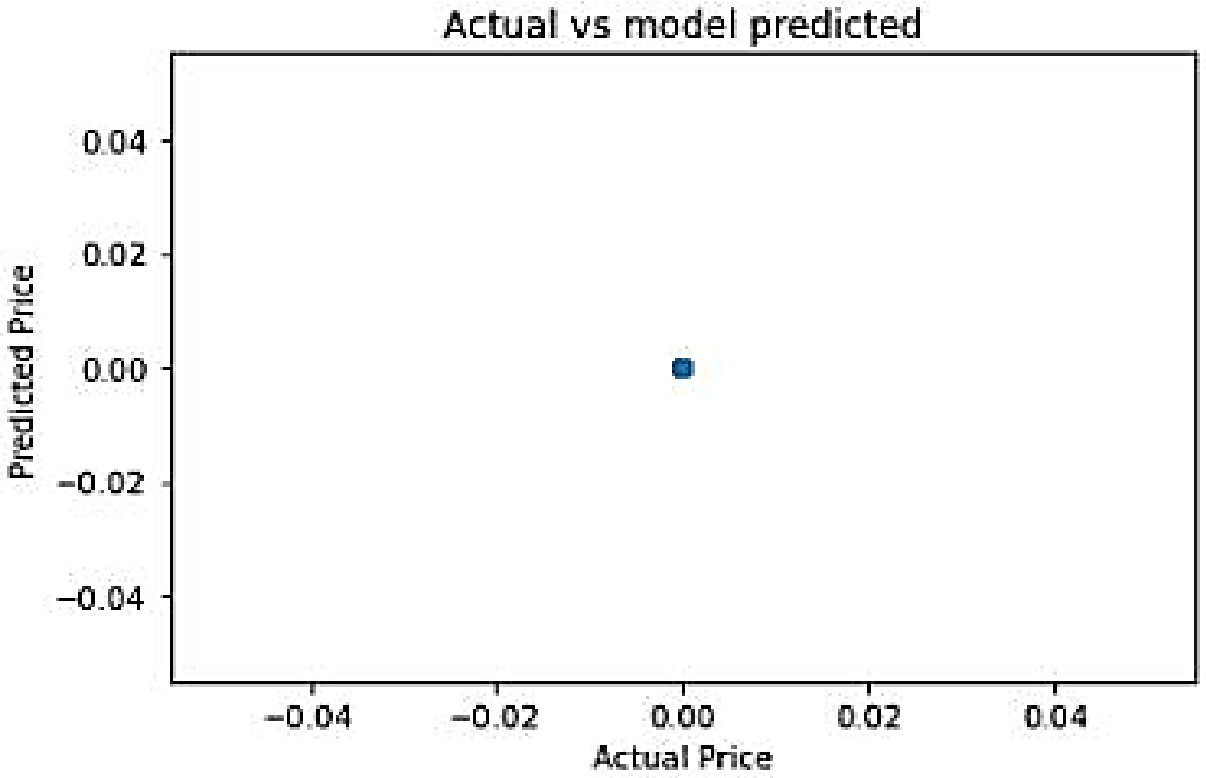
- Data preparation is followed by analysing the data, uncovering hidden trends and then applying various predictive machine learning models on the training set.

- 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.

- 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.

- 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.

MODEL DASHBOARD



It shows the effect of the model

FINALIZED MODEL:

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

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