

```
In [3]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

```
In [4]: df = pd.read_excel('Data_Train.xlsx')
```

```
In [5]: df.head(10)
```

Out[5]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	Null	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	Null	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	Null	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	Null	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	Null	13302
5	SpiceJet	24/06/2019	Kolkata	Banglore	CCU → BLR	09:00	11:25	2h 25m	non-stop	Null	3873

checking null values

```
In [12]: factor
```

Out[12]: 0.0

```
In [13]: factor = len(df[(df['Additional_Info'] == 'Null ')|(df['Additional_Info'] == 'Null')])/len(df['Additional_Info'])
percentage = factor*100
percentage
```

Out[13]: 78.14284377047646

drop Additional_Info columns in preprocessing!

```
In [14]: df[(df['Additional_Info'] == 'Null ')|(df['Additional_Info'] == 'Null')]
```

Out[14]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	Null	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	Null	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	Null	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	Null	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	Null	13302
...
10678	Air Asia	9/04/2019	Kolkata	Banglore	CCU → BLR	19:55	22:25	2h 30m	non-stop	Null	4107
10679	Air India	27/04/2019	Kolkata	Banglore	CCU → BLR	20:45	23:20	2h 35m	non-stop	Null	4145

```
jupyter Airfareforecast (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)
In [15]: df.duplicated().sum()
Out[15]: 220
```

Feature Engineering and Preprocessing

```
In [16]: def preprocess(data):
'''Function preprocesses the data and make it model ready. Simply push dataframe in the function.
!!! Use only after treating null values or when null values are less enough to drop. It returns two dataframes,
one for eda and one for model.'''

data.dropna(inplace = True)

df.drop_duplicates(inplace = True)

data['Date_of_Journey'] = pd.to_datetime(data['Date_of_Journey'])
data['day'] = pd.DatetimeIndex(data['Date_of_Journey']).day
data['month'] = pd.DatetimeIndex(data['Date_of_Journey']).month
data['year'] = pd.DatetimeIndex(data['Date_of_Journey']).year
data['weekday'] = pd.DatetimeIndex(data['Date_of_Journey']).weekday

data['Total_Stops'] = data['Total_Stops'].str.replace('non-stop', '0')
data['Total_Stops'] = data['Total_Stops'].str.replace('stops', '')
data['Total_Stops'] = data['Total_Stops'].str.replace('stop', '')
data['Total_Stops'] = data['Total_Stops'].str.replace(' ', '')
data['Total_Stops'] = data['Total_Stops'].astype(int)

data['Destination'] = np.where(data['Destination']=='New Delhi','Delhi',data['Destination'])
data['Airline'] = np.where(data['Airline']=='Jet Airways Business','Jet Airways',data['Airline'])

Arrival_Time = []
for i in data['Arrival_Time']:
    Arrival_Time.append(i[:5])
data['Arrival_Time'] = Arrival_Time
```

```
jupyter Airfareforecast (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)
BUILDING MODEL
In [21]: X = data_model.drop('Price', axis = 1)
y = data_model['Price']

In [22]: from sklearn.model_selection import train_test_split

In [23]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state = 42)

In [24]: from sklearn.ensemble import ExtraTreesRegressor

In [25]: extractor = ExtraTreesRegressor(random_state = 42)

In [26]: extractor.fit(X_train,y_train)
Out[26]: ExtraTreesRegressor(random_state=42)

In [27]: x_columns = X_train.columns
feature_rank = pd.DataFrame({'feature':x_columns,'importances':extractor.feature_importances_})

In [28]: feature_rank = feature_rank.sort_values('importances',ascending = False)
```

FINAL Barchart :

