Project: flight fare prediction (NEWTON SCHOOL)

NAME: Radharaman Sharma

Introduction

In the present aviation space we have seen how traffic has been raised significantly. Nowadays the price fares of flight even domestic as well as international are quite unpredictable. The prices are so not have any pattern which can be seen by our naked eyes but its so hard to predict what will the price of a flight next month even after following up the prices it changes drastically in a single day or two.

This project is all about resolving this issue and finding insights and patters in the dataset provided so that we can see a pattern and find out a suitable way to predict the flight fares.

Problem statement (provided overview)

Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, it will be a different story. We might have often heard travellers saying that flight ticket prices are so unpredictable. Huh! Here we take on the challenge! As data scientists, we are gonna prove that given the right data anything can be predicted. Here you will be provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities. Size of training set: 10683 records Size of test set: 2671 records FEATURES: Airline: The name of the airline. Date_of_Journey: The date of the journey Source: The source from which the service begins. Destination: The destination where the service ends. Route: The route taken by the flight to reach the destination. Dep_Time: The time when the journey starts from the source. Arrival_Time: Time of arrival at the destination. Duration: Total duration of the flight. Total_Stops: Total stops between the source and destination. Additional_Info: Additional information about the flight Price: The price of the ticket.

Proposed Solution

The solution proposed to take the required input of user from the created interface and process all the provided data to meet the requirements of the machine learning model and finally display the output saying so and so amount is the predicted cost.

Solution Improvements

We can even predict the cost of ticket considering whether is it a weekday, holiday season or other social reasons. But considering from the perspective of business, if we process such data and predict the cost of the discounted ticket it will bring some loss to the airlines company. Hence this method is not considered.

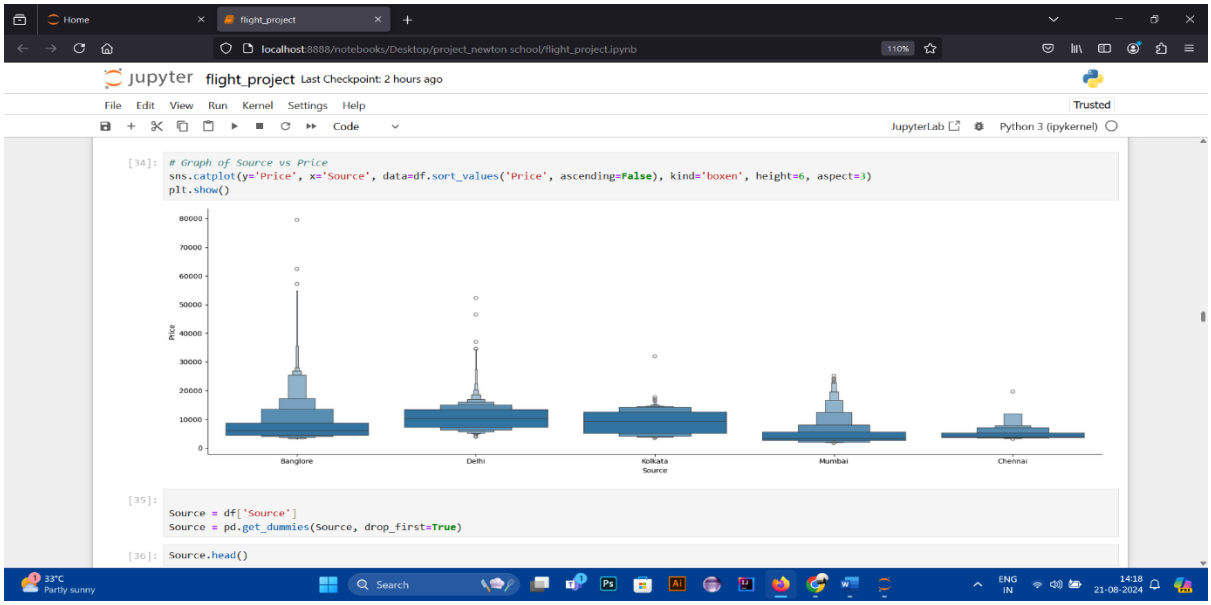
Data Description

There are about 10k+ records of flight information such as airlines, data of journey, source, destination, departure time, arrival time, duration, total stops, additional information, and price.

Approach

The main part of the project is here:

- We have started with data preprocessing and removed null values or handled null values with proper filling of missing values.
- Convert the data into the required formats such as conversion of arrival time into proper format that is mixed which is used there.
- Afterwards we have handled categorical data using one-hot encoder.
- Used seaborn visualization to represent the different aspects of our categorical dataset.



- At the end of it we have to print the heatmap of the categorical dataset in this field.
- Feature engineering came into play using scikit-learn in which we used extratreesRegressor().
- Our model was based on Random Forest Regressor on which our model was fitted.
- Afterwards we have got our r2 score and done hyperparameter turing with cross-validation approach.
- Two methods used in hyperparameter turing were 1. RandomizedSearchCV 2. GridSearchCV
- Lastly we obtained the values for MSE, MAE AND RMSE LOSS Functions.

Insights

1. Price Trends and Seasonality:

- **Seasonal Trends:** Identifying patterns in pricing related to seasons, holidays, and special events. For instance, prices may increase during peak travel seasons (e.g., summer, holidays) and decrease during off-peak times.
- **Day of the Week Patterns:** Fares might fluctuate based on the day of the week, with certain days like weekends being more expensive.

2. Demand-Supply Dynamics:

- **Demand Fluctuations:** Insights into how demand affects pricing, such as higher fares when demand exceeds supply or during last-minute bookings.
- **Booking Patterns:** Understanding when customers are more likely to book flights (e.g., weeks in advance versus last-minute).

3. Impact of External Factors:

- **Fuel Prices:** The correlation between fuel prices and fare changes.
- **Economic Indicators:** How factors like inflation, currency exchange rates, and economic conditions impact flight fares.
- **Competitor Pricing:** Analyzing how competitor pricing strategies influence fare adjustments.

4. Customer Behavior Insights:

- **Booking Lead Time:** The relationship between the time of booking and fare prices. Early bookings might lead to lower fares, while last-minute bookings could be more expensive.
- **Class of Service Preferences:** Understanding how different customer segments prefer economy, business, or first class, and how pricing affects their choices.

5. Route-Specific Analysis:

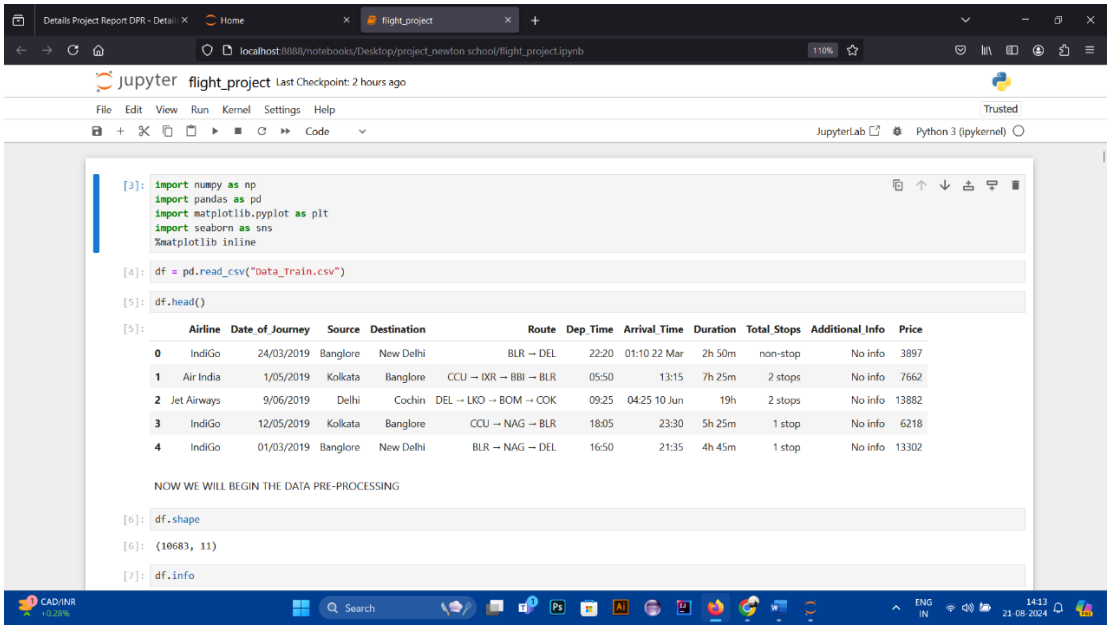
- **Popular Routes:** Identifying which routes have the most stable or volatile prices.
- **Distance vs. Fare:** Understanding how distance impacts fares and whether other factors (e.g., competition, route popularity) have a larger impact.
- **Hub Airports:** Analyzing how fares differ for flights to/from major hub airports versus smaller airports.

Errors

- Not able to generate the heatmap due to problem in the environment and categorical data is represented as true and false not in 0 and 1 due to that pandas dataframe created few problems and while solving this matplotlib stopped working.
- Matplotlib inline which was the ingeneral solution for these errors not working.

Outcomes

After working with normal methods we have observed some values and afterwards during the hyper parameter turing we got different values which improved our r2 score.



jupyter flight_project Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[4]:
```
df = pd.read_csv("Data_Train.csv")
```

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

[5]:

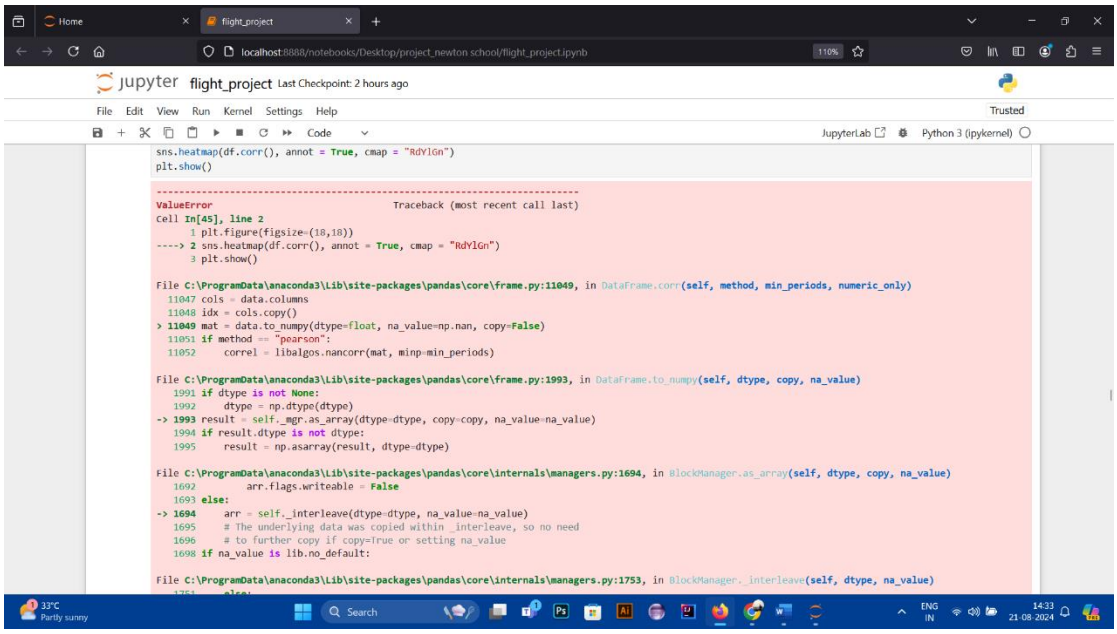
	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

NOW WE WILL BEGIN THE DATA PRE-PROCESSING

[6]:
```
df.shape
```

[6]:
```
(10683, 11)
```

[7]:
```
df.info
```



jupyter flight_project Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

```
sns.heatmap(df.corr(), annot = True, cmap = "RdYlGn")
plt.show()
```

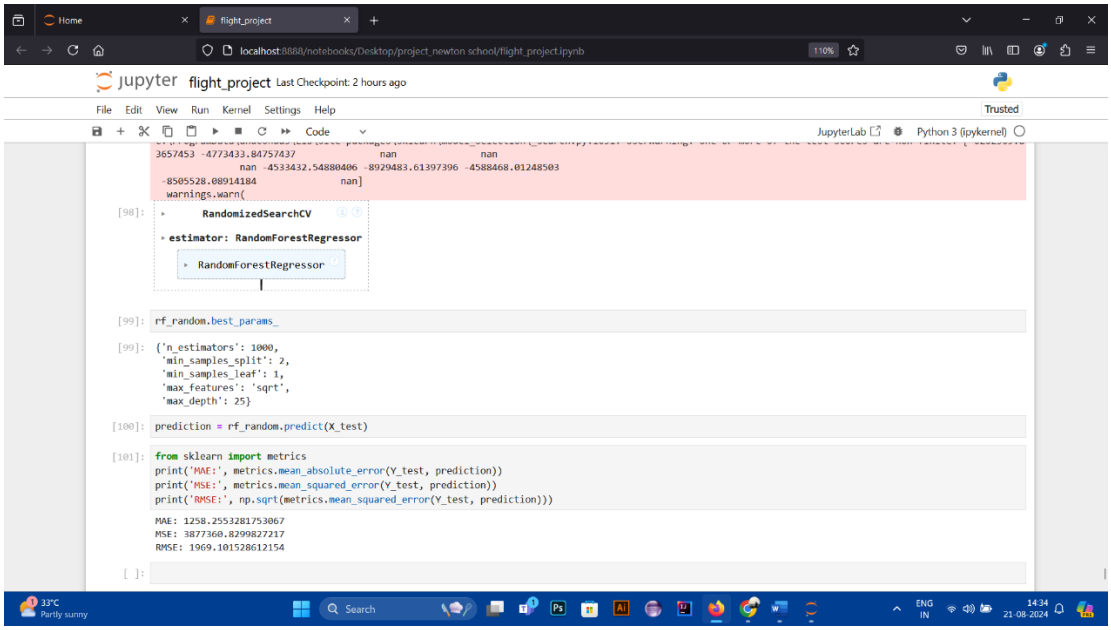
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[45], line 2
      1 plt.figure(figsize=(18,18))
----> 2 sns.heatmap(df.corr(), annot = True, cmap = "RdYlGn")
      3 plt.show()

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\frame.py:11049, in DataFrame.corr(self, method, min_periods, numeric_only)
  11047 cols = data.columns
  11048 idx = cols.copy()
> 11049 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  11051 if method == "pearson":
  11052     correl = libalgos.nancorr(mat, minp=min_periods)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\frame.py:1993, in DataFrame.to_numpy(self, dtype, copy, na_value)
   1991 if dtype is not None:
   1992     dtype = np.dtype(dtype)
-> 1993 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
   1994 if result.dtype is not dtype:
   1995     result = np.asarray(result, dtype=dtype)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1694, in BlockManager.as_array(self, dtype, copy, na_value)
   1692         arr.flags.writeable = False
   1693 else:
-> 1694     arr = self._interleave(dtype=dtype, na_value=na_value)
   1695     # The underlying data was copied within _interleave, so no need
   1696     # to further copy if copy=True or setting na_value
   1698 if na_value is lib.no_default:

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1753, in BlockManager._interleave(self, dtype, na_value)
```


Home
flight_project
localhost:8888/notebooks/Desktop/project_newton school/flight_project.ipynb
jupyter flight_project Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
3657453 -4773433.84757437 nan nan
nan -4533432.54880406 -8929483.61397396 -4588468.01248503
-8505528.08914184 nan]
warnings.warn(
[98]:
RandomizedSearchCV
estimator: RandomForestRegressor
RandomForestRegressor
[99]:
rf_random.best_params_
[99]:
{'n_estimators': 1000,
'min_samples_split': 2,
'min_samples_leaf': 1,
'max_features': 'sqrt',
'max_depth': 25}
[100]:
prediction = rf_random.predict(X_test)
[101]:
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(Y_test, prediction))
print('MSE:', metrics.mean_squared_error(Y_test, prediction))
print('RMSE:', np.sqrt(metrics.mean_squared_error(Y_test, prediction)))
MAE: 1258.2553281753067
MSE: 3877360.8299827217
RMSE: 1969.101528612154
[ ]:
33°C
Partly sunny
Search
ENG
IN
14:34
21-08-2024