



A project on flight price prediction



Submitted by-  
Sankalp Mahapatra  
Internship 29

# ACKNOWLEDGMENT

I would like to express my sincere thanks of gratitude to my SME as well as “Flip Robo Technologies” team for letting me work on “Flight Price Prediction” project also huge thanks to my academic team “DataTrained”. Their suggestions and directions have helped me in the completion of this project successfully. This project also helped me in doing lots of research wherein I came to know about so many new things.

Finally, I would like to thank my family and friends who have helped me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

# TABLE OF CONTENTS:

## **1. Introduction**

- 1.1 Business Problem Framing
- 1.2 Conceptual Background of the Domain Problem
- 1.3 Review of Literature
- 1.4 Motivation for the Problem Undertaken

## **2. Analytical Problem Framing**

- 2.1 Mathematical/ Analytical Modelling of the Problem
- 2.2 Data Sources and their formats
- 2.3 Data Pre-processing Done
- 2.4 Data Inputs- Logic- Output Relationships
- 2.5 Hardware & Software Requirements & Tools Used

## **3. Model/s Development and Evaluation**

- 3.1 Identification of possible Problem-solving approaches (Methods)
- 3.2 Visualizations
- 3.3 Testing of Identified Approaches (Algorithms)
- 3.4 Run and Evaluate Selected Models
- 3.5 Key Metrics for success in solving problem under consideration
- 3.6 Interpretation of the Results

## **4. Conclusion**

- 4.1 Key Findings and Conclusions of the Study
- 4.2 Learning Outcomes of the Study in respect of Data Science
- 4.3 Limitations of this work and Scope for Future Work

# 1. INTRODUCTION

## 1.1 Business Problem Framing

Airline industry is one of the most sophisticated in its use of dynamic pricing strategies to maximize revenue, based on proprietary algorithms and hidden variables. That is why the airline companies use complex algorithms to calculate the flight ticket prices. There are several different factors on which the price of the flight ticket depends. The seller has information about all the factors, but buyers are able to access limited information only which is not enough to predict the airfare prices. Considering the features such as departure time, arrival time and time of the day it will give the best time to buy the ticket.

Nowadays, the number of people using flights has increased significantly. It is difficult for airlines to maintain prices since prices change dynamically due to different conditions. That's why we will try to use machine learning models to solve this problem. This can help airlines by predicting what prices they can maintain. It can also help customers to predict future flight prices and plan their journey accordingly.

**Business goal:** The main aim of this project is to predict the price of flight tickets based on various features. The purpose of the paper is to study the factors which influence the fluctuations in the airfare prices and how they are related to the change in the prices. Then using this information, build a system that can help buyers whether to buy a ticket or not. So, we will deploy an Machine Learning model for flight ticket price prediction and analysis. This model will provide the approximate selling price for the flight tickets based on different features.

## 1.2 Conceptual Background of the Domain Problem

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.

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).

Here we are trying to help the buyers to understand the price of the flight tickets by deploying machine learning models. These models would help the sellers/buyers to understand the flight ticket prices in market and accordingly they would be able to book their tickets.

## **1.3 Review of Literature**

Literature review covers relevant literature with the aim of gaining insight into the factors that are important to predict the flight ticket prices in the market. In this study, we discuss various applications and methods which inspired us to build our supervised ML techniques to predict the price of flight tickets in different locations. We did a background survey regarding the basic ideas of our project and used those ideas for the collection of data information by doing web scraping from [www.yatra.com](http://www.yatra.com) website which is a web platform where buyers can book their flight tickets.

This project is more about data exploration, feature engineering and pre-processing that can be done on this data. Since we scrape huge amount of data that includes more flight related features, we can do better data exploration and derive some interesting features using the available columns. Different

techniques like ensemble techniques, and decision trees have been used to make the predictions.

The goal of this project is to build an application which can predict the price of flight tickets with the help of other features. In the long term, this would allow people to better explain and reviewing their purchase in this increasing digital world.

## **1.4 Motivation for the Problem Undertaken**

Air travel is the fastest method of transport around, and can cut hours or days off of a trip. But we know how unexpectedly the prices vary. So, I was interested in Flight Fares Prediction listings to help individuals and find the right fares based on their needs. And also, to get hands on experience and to know that how the data scientist approaches and work in an industry end to end.

# **2. ANALYTICAL PROBLEM FRAMING**

## **2.1 Mathematical/ Analytical Modelling of the Problem:**

We need to develop an efficient and effective Machine Learning model which predicts the price of flight tickets. So, "Price" is our target variable which is continuous in nature. Clearly it is a Regression problem where we need to use regression algorithms to predict the results. This project is done on three phases:

- **Data Collection Phase:** I have done web scraping to collect the data of flights from the well-known website [www.yatra.com](http://www.yatra.com) where I found more features of flights compared to other websites and I fetch data for different locations. As per the requirement we need to build the model to predict the prices of flight tickets.
- **Data Analysis:** After cleaning the data I have done some analysis on the data by using different types of visualizations.

• **Model Building Phase:** After collecting the data, I built a machine learning model. Before model building, have done all data pre-processing steps. The complete life cycle of data science that I have used in this project are as follows:

- Data Cleaning
- Exploratory Data Analysis
- Data Pre-processing
- Model Building
- Model Evaluation
- Selecting the best model

## 2.2 Data Sources and their formats

We have collected the dataset from the website [www.yatra.com](http://www.yatra.com) which is a web platform where the people can purchase/book their flight tickets. The data is scraped using Web scraping technique and the framework used is Selenium. We scrapped nearly 5303 of the data and fetched the data for different locations and collected the information of different features of the flights and saved the collected data in excel format. The dimension of the dataset is 5303 rows and 9 columns including target variable "Price". The particular dataset contains both categorical and numerical data type. The data description is as follows:

Airline	The Name of airline
Departure_time	The time when the journey starts from the source
Time_of_arrival	Time of arrival at the destination

Duration	Total duration taken by the flight to reach the destination from the source
Source	The source from which the service begins
Destination	The destination where the service ends
Meal_availability	Availability of meals in the flight
Number_of_stops	Total stops between the source and destination
Price	The price of the flight ticket

## 2.3 Data Pre-processing Done

Data pre-processing is the process of converting raw data into a well-readable format to be used by Machine Learning model. Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre-process our data before feeding it into our model. I have used following pre-processing steps:

- Importing necessary libraries and loading collected dataset as a data frame.
- Checked some statistical information like shape, number of unique values present, info, unique (), data types, value count function etc.
- Checked null values and found no missing values in the dataset.
- Taking care of Timestamp variables by converting data types of “Departure\_time” and “Time\_of\_arrival” from object data type into datetime data types.
- Done feature engineering on some features as they had some irrelevant values like “,”, “:” and replaced them by empty space.
- The column Duration had values in terms of minutes and hours. Duration means the time taken by the plane to reach the destination



and it is the difference between the arrival time and Departure time. So, I have extracted proper duration time in terms of float data type from arrival and departure time columns.

- Extracted Departure\_Hour, Departure\_Min and Arrival\_Hour, Arrival\_Min columns from Departure\_time and Time\_of\_arrival columns and dropped these columns after extraction.
- The target variable "price" should be continuous numeric data but due to some string values like ",", it was showing as object data type. So, I replaced this sign by empty space and converted into float data type.
- From the value count function of Meal\_availability we observed "eCash 250" entry which does not belong to meals so I have replaced it as "None" and grouped same categories.
- From the value count function of Number\_of\_stops I found categorical data so replaced them with numeric data according to stops.
- Checked statistical description of the data and separated categorical and numeric features.
- Performed the visualizations using count plots, heatmaps, stripplot, boxplots and regplots. I have found out the relation between the features as well as the relation between the features and labels.
- Identified outliers using box plots and found no outliers.
- Encoded the columns having object data type using Label Encoder method. Used Pearson's correlation coefficient to check the correlation between label and features. With the help of heatmap and correlation bar graph was able to understand the Feature vs Label relativity.
- Separated feature and label data and feature scaling is performed using Standard Scaler method to avoid any kind of data biasness.

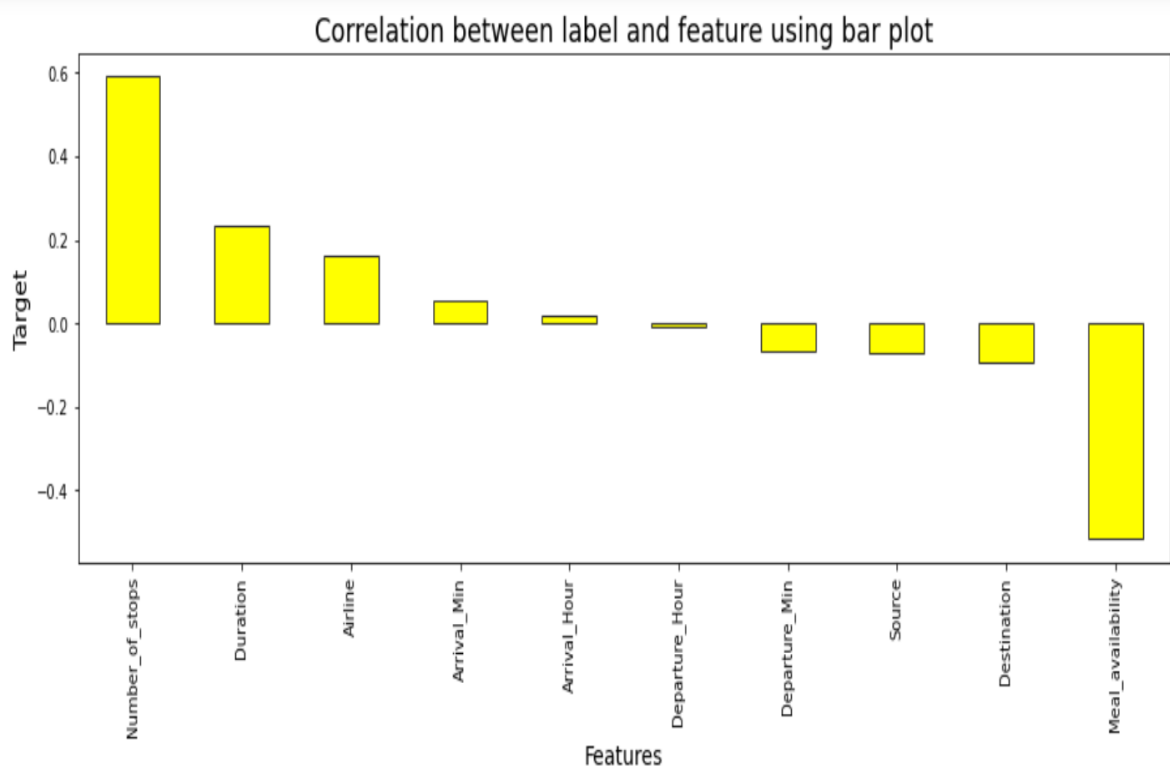
## 2.4 Data Inputs- Logic- Output Relationships

The dataset consists of label and features. The features are independent and label is dependent as the values of our independent variables changes as our label varies.

- Since we had both numerical and categorical columns, I checked the distribution of skewness using dist plots for numerical features and

checked the counts using count plots & pie plots for categorical features as a part of univariate analysis.

- To analyse the relation between features and label I have used many plotting techniques where I found numerical continuous variables having some relation with label Price with the help of categorical and line plot.
- have checked the correlation between the label and features using heat map and bar plot. Where I got both positive and negative correlation between the label and features. Below is the bar graph to know the correlation between features and label.



## 2.5 Hardware & Software Requirements & Tools Used

To build the machine learning projects it is important to have the following hardware and software requirements and tools.

### Hardware required:

- Processor: core i5 or above
- RAM: 8 GB or above

- ROM/SSD: 250 GB or above

### Software required:

- Distribution: Anaconda Navigator
- Programming language: Python
- Browser based language shell: Jupyter Notebook
- Chrome: To scrape the data

### Libraries Used:

```
1 import numpy as np
2 import pandas as pd
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 import os
6 import scipy as stats
7 from scipy.stats import zscore
8 from sklearn.preprocessing import PowerTransformer
9 from sklearn.model_selection import train_test_split
10 from sklearn.metrics import mean_absolute_error
11 from sklearn.metrics import mean_squared_error
12 from sklearn.metrics import classification_report
13 from sklearn.model_selection import cross_val_score
14 from sklearn.metrics import r2_score
15 from sklearn import metrics
16 from sklearn.tree import DecisionTreeRegressor
17 from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
18 from sklearn.ensemble import GradientBoostingRegressor
19 from sklearn.ensemble import BaggingRegressor
20 from sklearn.neighbors import KNeighborsRegressor as KNN
21 from sklearn.model_selection import GridSearchCV
22 import warnings
23 %matplotlib inline
```

## 2. MODEL/S DEVELOPMENT AND EVALUATION

### 3.1 Identification of possible Problem-solving approaches (Methods):

I have used both statistical and analytical approaches to solve the problem which mainly includes the pre-processing of the data also used EDA techniques and heat map to check the correlation of independent and dependent features. Removed skewness using square root transformation. Encoded data using Label Encoder. Also, before building the model, I made sure that the

input data is cleaned and scaled before it was fed into the machine learning models. Checked for the best random state to be used on our Regression Machine Learning model pertaining to the feature importance details. Finally created multiple regression models along with evaluation metrics.

For this particular project we need to predict flight ticket prices. In this dataset, “Price” is the target variable, which means our target column is continuous in nature so this is a regression problem. I have used many regression algorithms and predicted the flight ticket price. By doing various evaluations I have selected RandomForestRegressor() as best suitable algorithm to create our final model as it is giving high R2 score and low evaluation error among all the algorithms used. Performed hyper parameter tuning on best model. Then I saved my final model and loaded the same for predictions.

## **3.2 Testing of Identified Approaches (Algorithms)**

Since “Price” is my target variable which is continuous in nature, from this I can conclude that it is a regression type problem hence I have used following regression algorithms. After the pre-processing and data cleaning I left with 11 columns including target and with the help of feature importance bar graph I used these independent features for model building and prediction. The algorithms used on training the data are as follows:

1. Decision Tree Regressor
2. Random Forest Regressor
3. Linear Regression
4. KNearestNeighbour Regressor
5. SVR
6. Bagging Regressor

## **3.3 Run and evaluate selected models**

### **Model Building:**

```

1 #splitting the data between train and test. the model will be built(trained) on the train data and tested on test data
2
3 x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.25,random_state=488)
4 y_train.head()
5

```

```

1 from sklearn.metrics import mean_squared_error, mean_absolute_error
2 from sklearn.linear_model import LinearRegression, Ridge, Lasso
3 from sklearn.svm import SVR
4 from sklearn.tree import DecisionTreeRegressor
5 from sklearn.ensemble import RandomForestRegressor
6 from sklearn.neighbors import KNeighborsRegressor

```

```

1 regression=LinearRegression()
2 knn=KNeighborsRegressor()
3 rf=RandomForestRegressor()
4 svr=SVR()
5 dtc=DecisionTreeRegressor()
6 BR=BaggingRegressor()
7

```

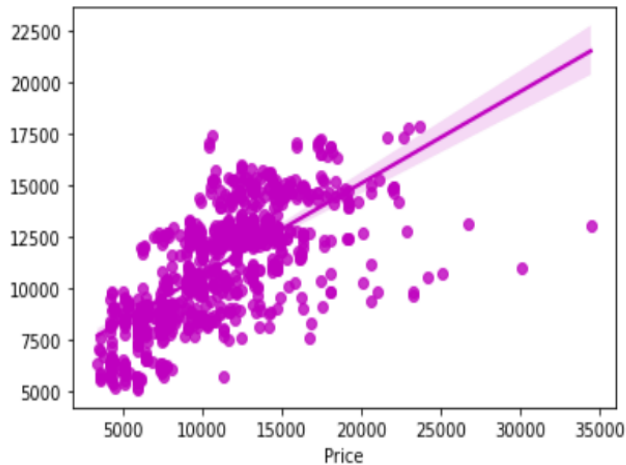
```

models = [regression, knn, svr, rf,dtc,BR]
for m in models:
    print (m)
    m.fit(x_train, y_train)
    y_pred = m.predict(x_test)
    print ('adjusted R2 score for training data-----',m.score(x_train, y_train))
    print ('adjusted R2 score for testing data-----',m.score(x_test, y_test))
    print ("mean absolute error-----",mean_absolute_error(y_test,y_pred))
    print ("mean squared error-----",mean_squared_error(y_test,y_pred))
    print ("root mean squared error-----",np.sqrt(mean_squared_error(y_test,y_pred)))
    sns.regplot(y_test,y_pred,color="m")
    plt.show()

```

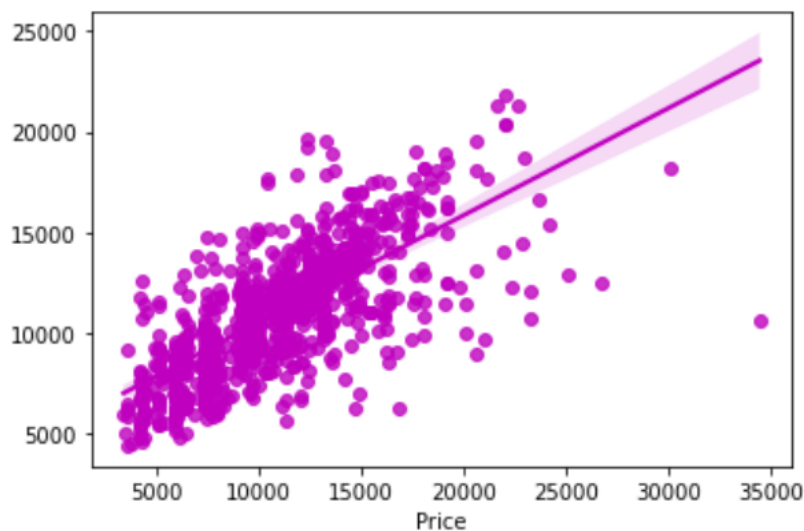
## Linear Regression Model

```
LinearRegression()  
adjusted R2 score for training data----- 0.4652970579570842  
adjusted R2 score for testing data----- 0.4634316364015756  
mean absolute error----- 2178.5920229148605  
mean squared error----- 9134561.742033688  
root mean squared error----- 3022.343749812997
```



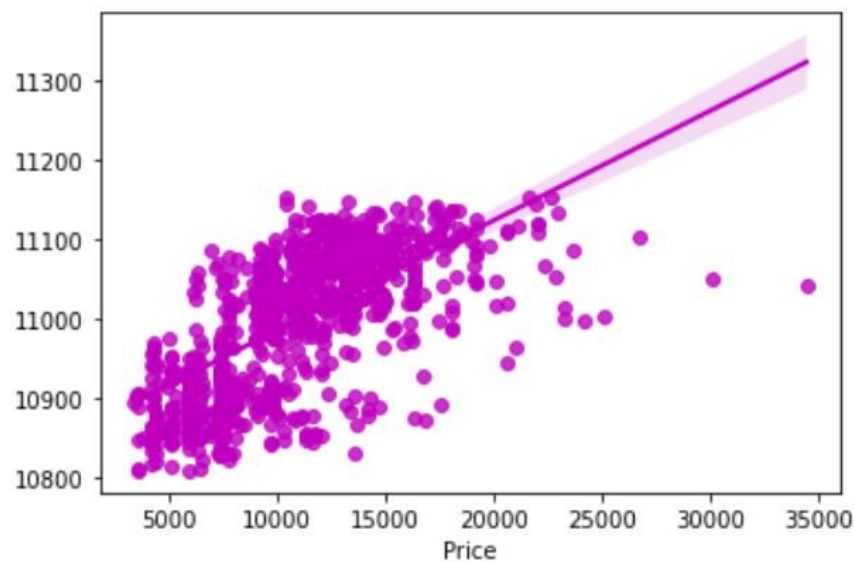
## KNearestNeighbour Regressor Model

```
KNeighborsRegressor()  
adjusted R2 score for training data----- 0.7045886545797582  
adjusted R2 score for testing data----- 0.5021785482603923  
mean absolute error----- 1960.0590959206174  
mean squared error----- 8474932.731642779  
root mean squared error----- 2911.1737721480627
```



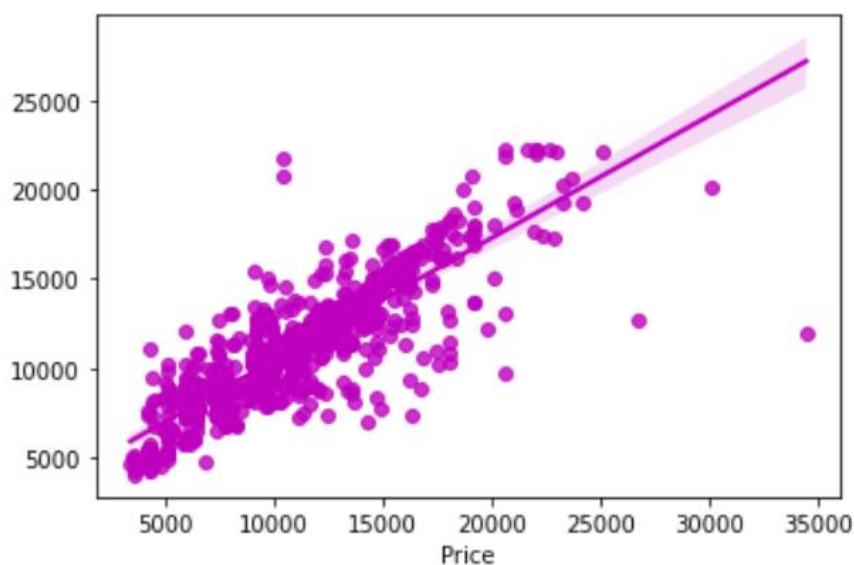
## SVR Model

```
SVR()  
adjusted R2 score for training data----- 0.027650540091973208  
adjusted R2 score for testing data----- 0.027242696932282096  
mean absolute error----- 3188.1800116842146  
mean squared error----- 16560260.066947298  
root mean squared error----- 4069.4299437325735
```



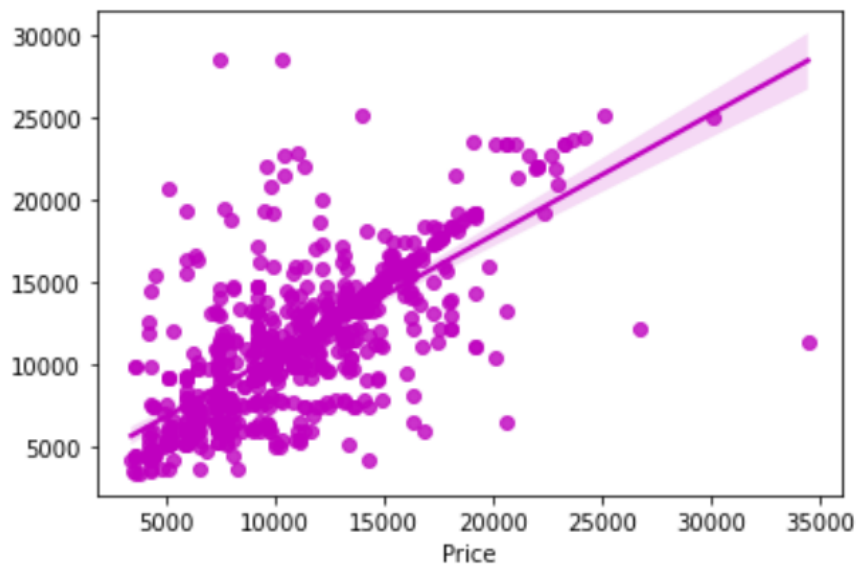
## RandomForestRegressor Model

```
RandomForestRegressor()  
adjusted R2 score for training data----- 0.9604741505119984  
adjusted R2 score for testing data----- 0.7116346249206799  
mean absolute error----- 1335.5702362953107  
mean squared error----- 4909143.925783416  
root mean squared error----- 2215.658801752521
```



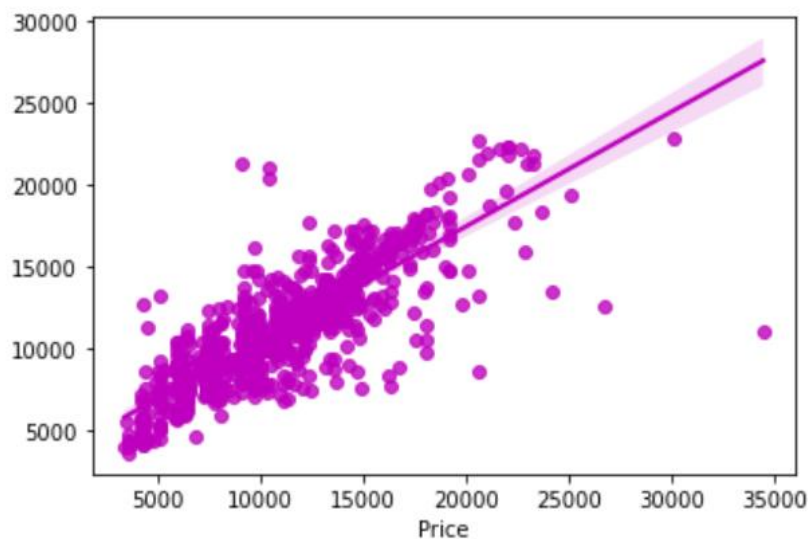
## Decision Tree Regressor

```
DecisionTreeRegressor()  
adjusted R2 score for training data----- 0.9982255033707857  
adjusted R2 score for testing data----- 0.4281651933242069  
mean absolute error----- 1615.7371922087468  
mean squared error----- 9734939.109703539  
root mean squared error----- 3120.086394589666
```



## Bagging Regressor

```
BaggingRegressor()  
adjusted R2 score for training data----- 0.942808019577655  
adjusted R2 score for testing data----- 0.6848018679039403  
mean absolute error----- 1385.877077277739  
mean squared error----- 5365945.877420349  
root mean squared error----- 2316.451138578224
```





## Finding cross validation scores for all the regression models

```
1 # Cross validation scores for all models
2 from sklearn.model_selection import cross_val_score
3 for m in models:
4     c_v= cross_val_score(m, x, y, cv = 10)
5     print ('Cross Validation Score for ',m, ' is :', c_v.mean())
6     print (' ')
```

Cross Validation Score for LinearRegression() is : 0.40953958363652304

Cross Validation Score for KNeighborsRegressor() is : -0.12498143165341899

Cross Validation Score for SVR() is : -0.05970187532778348

Cross Validation Score for RandomForestRegressor() is : 0.45619197633273406

Cross Validation Score for DecisionTreeRegressor() is : -0.06448642476716263

Cross Validation Score for BaggingRegressor() is : 0.4103893103086138

## Model Selection:

After analysing all the models we have concluded that RandomForestRegressor() model gives the best R2 score and cross validation score. And based on the R2 score we have chosen RandomForestRegressor() as the best model. We will use RandomForestRegressor() model for further analysis.

## Hyper Parameter Tuning:

### Hyperparameter Tuning of RandomForestRegressor() model using GridSearchCV

```
: 1 # Hyperparameter Tuning
2 from sklearn.model_selection import GridSearchCV
3 param_grid={'min_samples_split': [2, 5, 10], 'max_features':['auto', 'sqrt'], 'min_samples_leaf':[1,2,4], 'n_estimators': [2,5,
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
253
```

```
1 gridsearch.best_params_
```

```
{'max_features': 'auto',  
'min_samples_leaf': 2,  
'min_samples_split': 5,  
'n_estimators': 17}
```

```
1 rf=RandomForestRegressor(max_features='auto', min_samples_leaf=2,min_samples_split=5,n_estimators=17)
```

```
1 rf.fit(x_train,y_train)
```

```
RandomForestRegressor(max_features='auto', min_samples_leaf=2,  
                        min_samples_split=5, n_estimators=17)
```

```
1 y_pred=rf.predict(x_test)
```

```
2
```

```
1 y_pred
```

```
array([13957.94705882,  7802.06323529, 14485.3767507 , 17353.13235294,  
       11703.48529412, 15476.49635854,  9422.37058824,  9737.99397759,  
       11599.11323529, 10667.56862745, 11695.69243697, 10415.93333333,  
       11947.75273109,  9035.57094474,  7182.72661064, 12918.17170868,  
       11990.95056022,  9820.58823529, 13569.4697479 , 17404.02254902,  
       11778.14509804, 12478.9719888 , 12727.7894958 ,  8796.1745098 ,  
       15546.04278075,  4394.35980392,  5798.51029412,  9919.88039216,  
       10837.77759104, 12573.68991597,  9437.14509804, 13692.67375859,  
       8165.86386555,  3968.49327731,  8359.83015873, 19556.80980392,  
       8177.59591291, 14070.60392157,  7212.40116713, 12049.36442577,  
       8966.29131653,  9084.34159664, 12108.03284314, 15267.03760504,  
       14539.93160598,  8579.53851541, 11697.19957983, 11506.41006069,  
       8871.37745098, 12934.06323529, 10020.98482726, 14130.53235294,  
       8527.2004902 , 10983.96176471, 10002.33984594, 10441.83188609,  
       12314.17514006, 11304.54607843,  9982.59901961,  9426.20168067,  
       12328.382493 , 10870.91736695,  7089.32941176, 12551.2219888 ,  
       8593.39136321,  8790.14769544, 10265.2627451 , 12924.85266106,  
       12099.07044818, 11850.18123249,  7469.64838936,  8793.43543417.]
```

```
1 print ('adjusted R2 score for training data-----',rf.score(x_train, y_train))  
2 print ('adjusted R2 score for testing data-----',rf.score(x_test, y_test))  
3 print ("mean absolute error-----",mean_absolute_error(y_test,y_pred))  
4 print ("mean squared error-----",mean_squared_error(y_test,y_pred))  
5 print ("root mean squared error-----",np.sqrt(mean_squared_error(y_test,y_pred)))
```

```
adjusted R2 score for training data----- 0.9108800861515229
```

```
adjusted R2 score for testing data----- 0.6933698030880564
```

```
mean absolute error----- 1409.8658388373917
```

```
mean squared error----- 5220084.998824781
```

```
root mean squared error----- 2284.750533170915
```

## Saving the final model and predicting the flight ticket price

```
1 Predicted_ticket_Price = pd.DataFrame([rf.predict(x_test)[:],y_test[:]],index=["Predicted","Original"])
2 Predicted_ticket_Price
```

	0	1	2	3	4	5	6	7	8	9	...	
Predicted	13957.947059	7802.063235	14485.376751	17353.132353	11703.485294	15476.496359	9422.370588	9737.993978	11599.113235	10667.568627	...	114
Original	13695.000000	7425.000000	15939.000000	19089.000000	11659.000000	15312.000000	11349.000000	7213.000000	16027.000000	13524.000000	...	135

2 rows × 907 columns

```
1 #saving the machine Learning model
2 import pickle
3 filename='finalized_model.pickle17'
4 pickle.dump(rf,open('finalized_model.pickle17','wb'))
```

### 3.4 Key Metrics for success in solving problem under consideration

The essential step in any machine learning model is to evaluate the accuracy and determine the metrics error of the model. I have used Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and R2 Score metrics for my model evaluation:

**Mean Absolute Error (MAE):** MAE is a popular error metric for regression problems which gives magnitude of absolute difference between actual and predicted values.

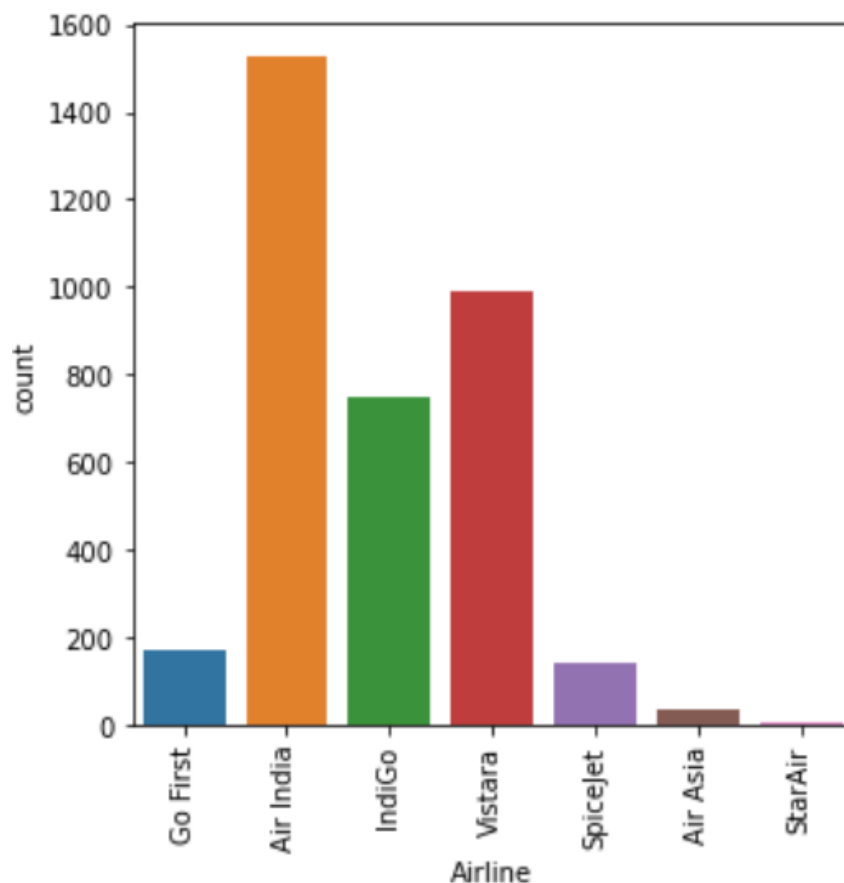
**Mean Squared Error (MSE):** MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value. We perform squared to avoid the cancellation of negative terms and it is the benefit of MSE.

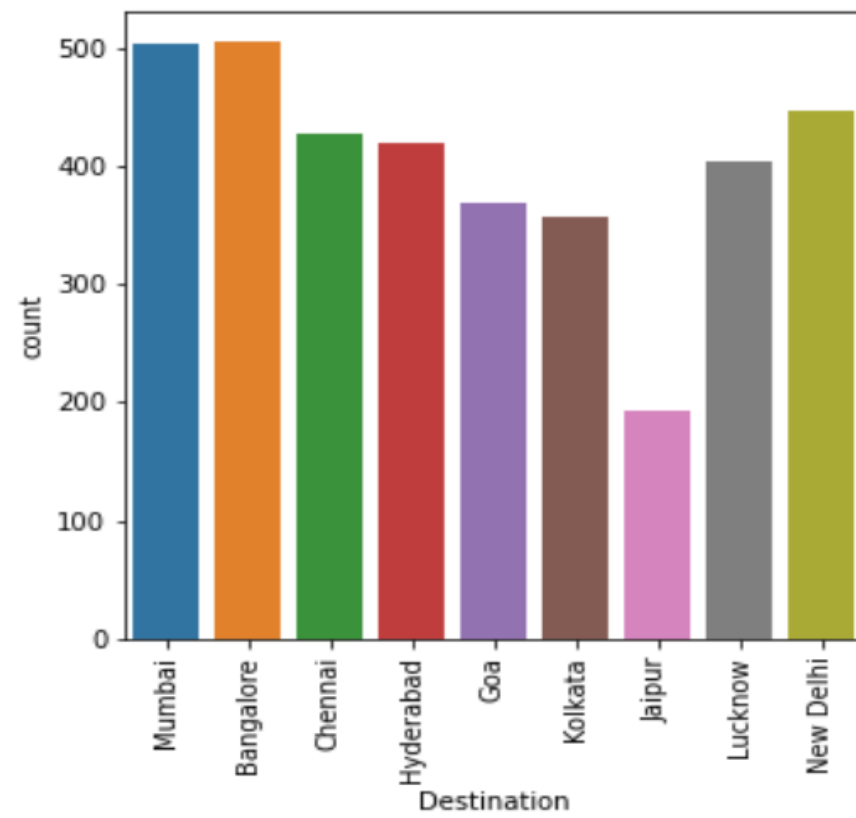
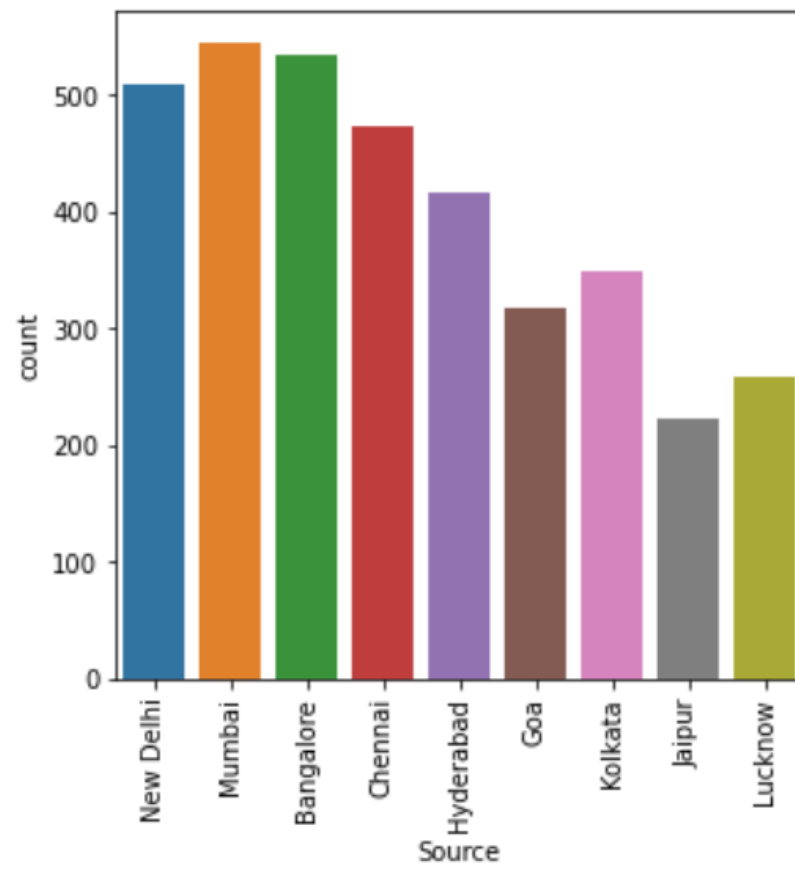
**Root Mean Squared Error (RMSE):** RMSE is an extension of the mean squared error. The square root of the error is calculated, which means that the units of the RMSE are the same as the original units of the target value that is being predicted.

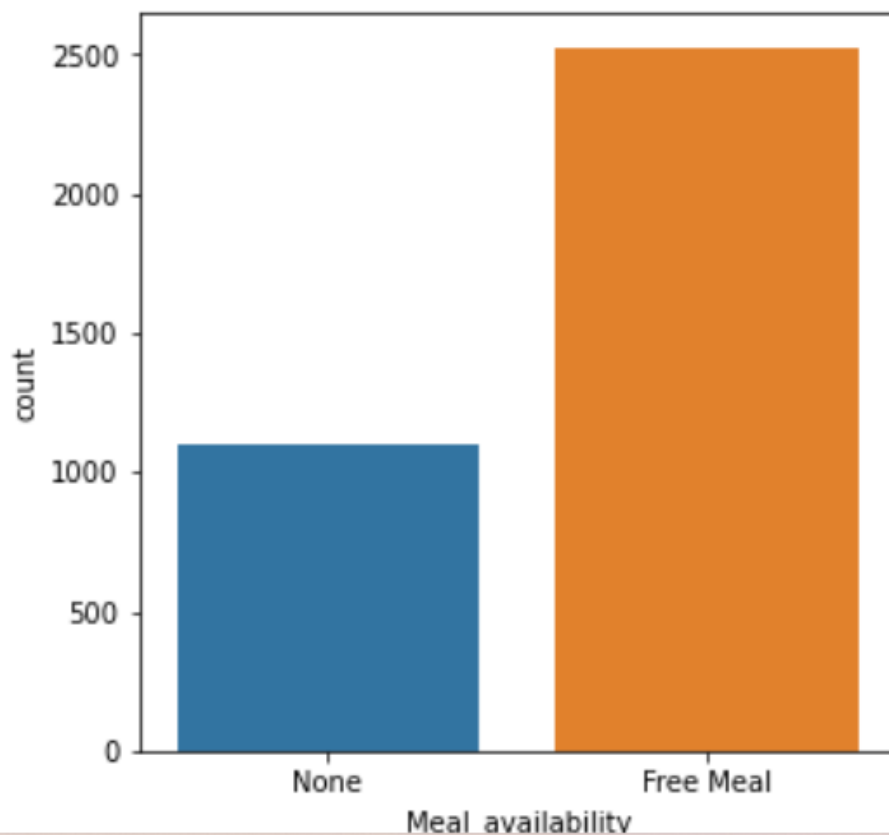
**R2 Score:** I have used R2 score which gives the accurate value for the models used. On the basis of R2 score I have created final model.

### 3.5 Visualizations

I have analysed the data by plotting the relationship between the features and labels as well as the relationship among the features. I have also used box plots to find the outliers. I have count plots and distribution plot and used reg plots, scatter plots and heatmap plot. These plots have given good knowledge about the relationships.

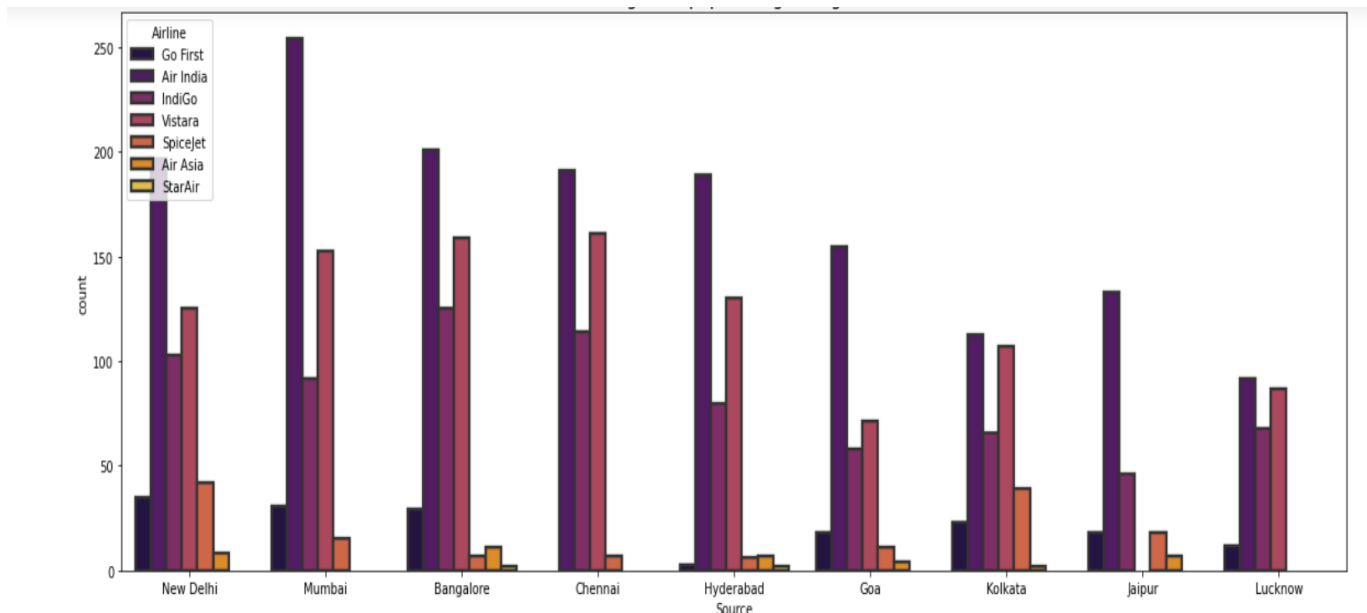




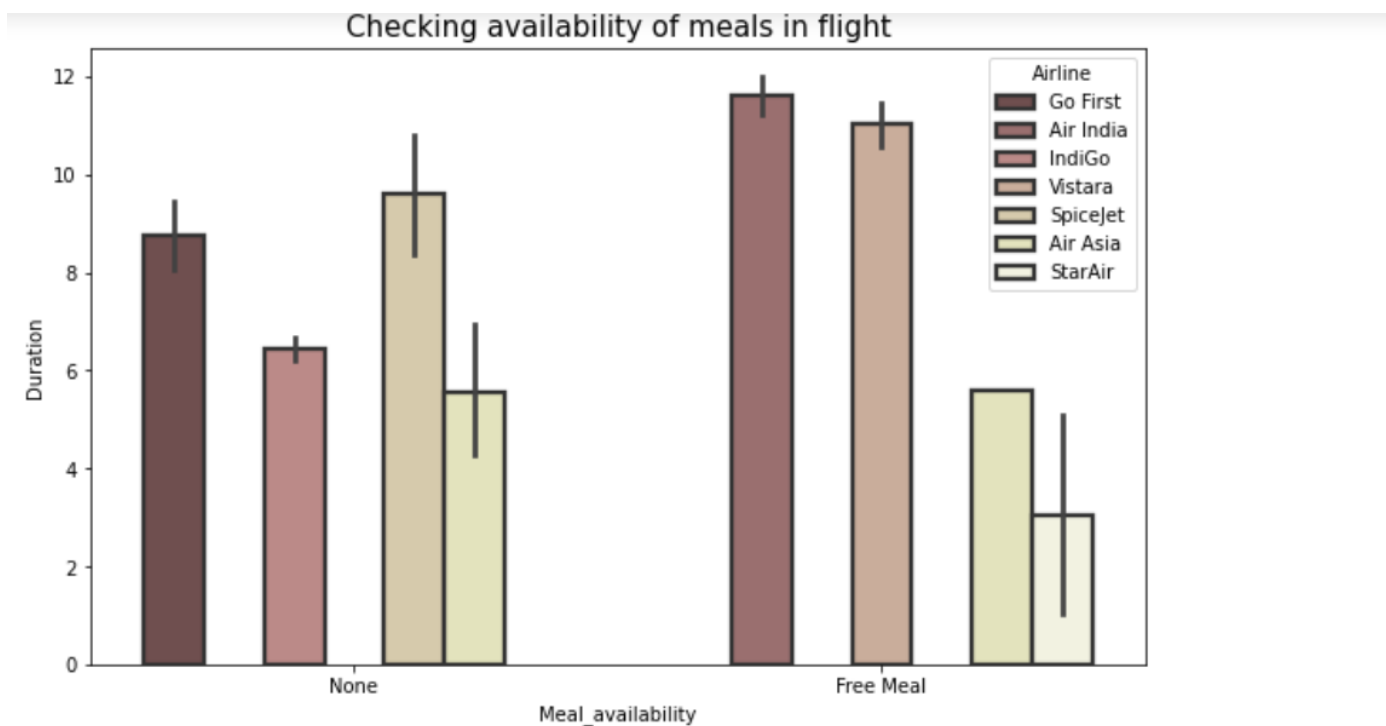


#### Observations from the count plots

- Most of the flights providing free meals and only few flights are not providing any meals.
- From the count plot we can observe more number of flights are from Mumbai, New Delhi, Jaipur, Kolkata and Bangalore. Only few flights are from Hyderabad.
- More number of flights are heading towards Lucknow, New Delhi and Kolkata. Only few flights are travelling to Hyderabad.



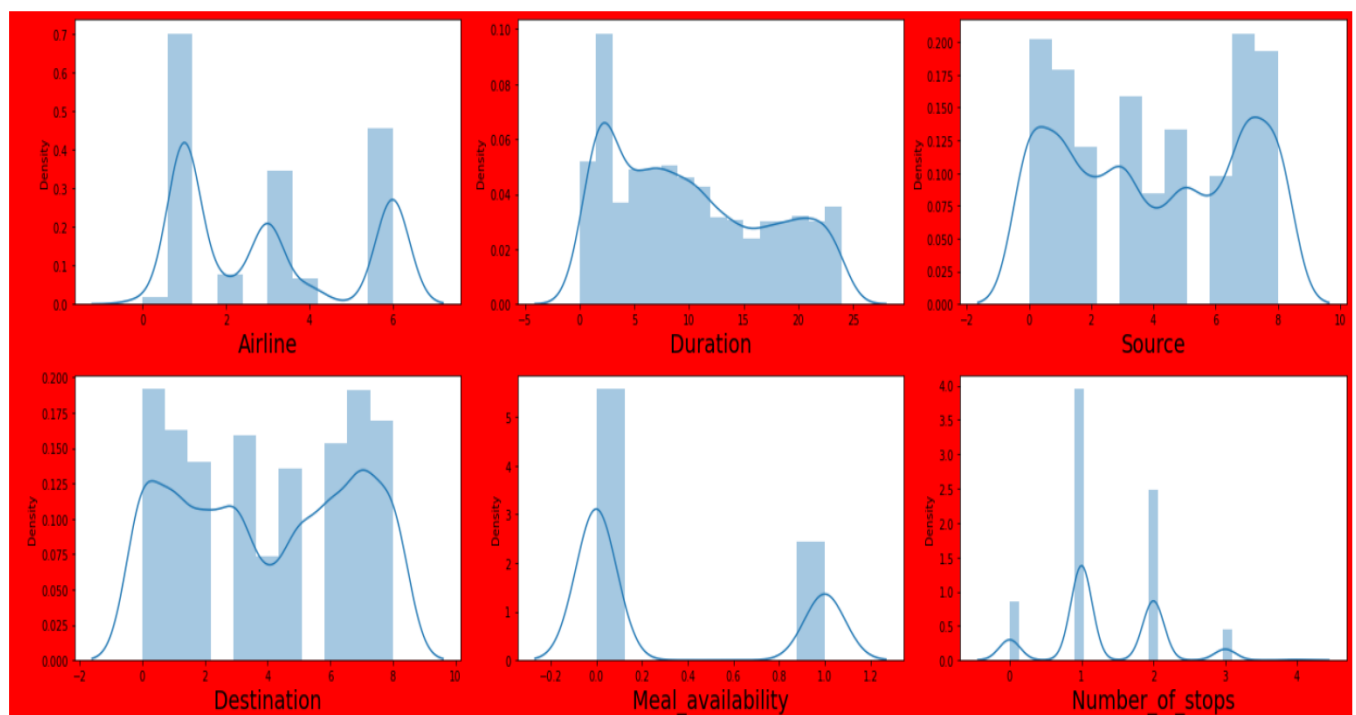
The plot shows the region wise count of airlines which tells us that Jaipur source is not having Vistara flights and it has Air India flights in higher count compared to other sources. Other sources have Air India, Vistara and Indigo flights with higher count.



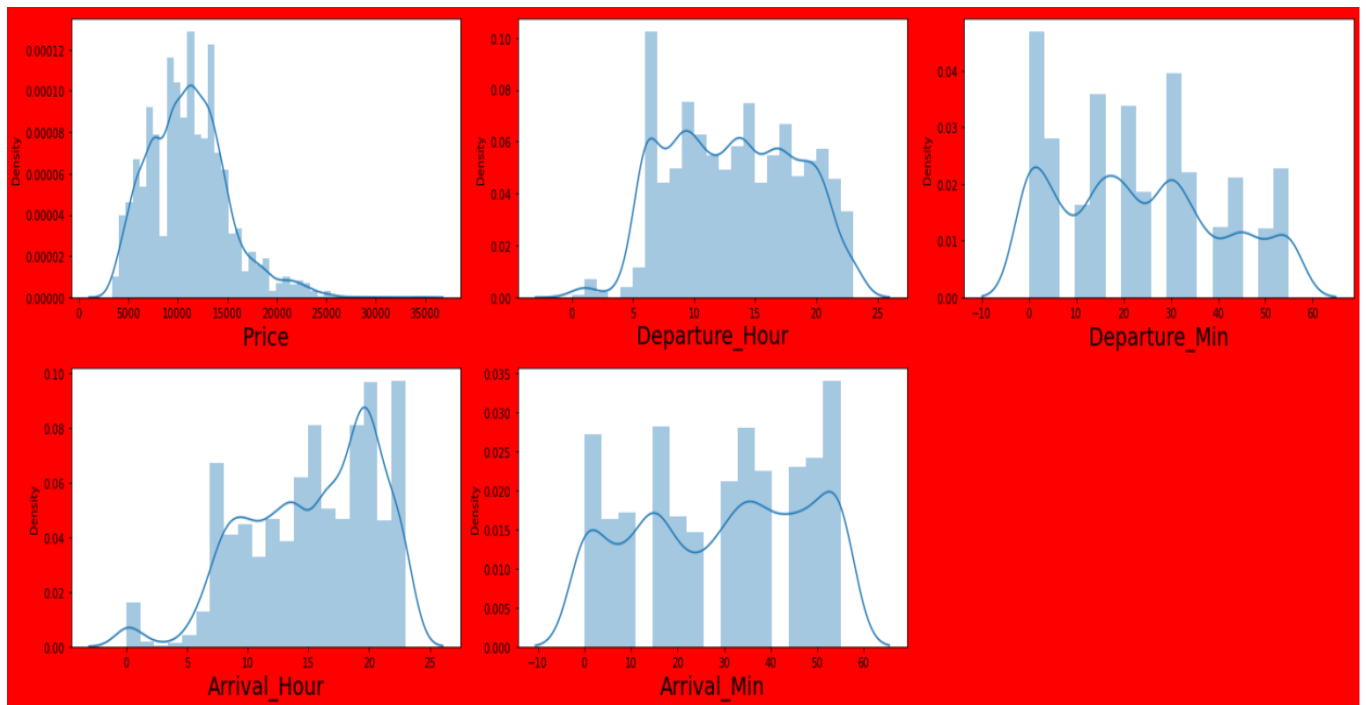
All the airlines provides free meals during the journey having the duration below 11 hours.

now the data looks good and there is no missing values and Object values so we can start visualizing the type of distribution for each feature.

```
1 # now the data looks good and there is no missing values and Object values so we can start visualizing the type of distribut
2 # we will only evaluate the type of distribution for features having continious data here
3
4 plt.figure(figsize=(20,15), facecolor='red')
5 plotnumber=1
6
7 for column in df:
8     if plotnumber<=11:
9         ax=plt.subplot(4,3,plotnumber)
10        sns.distplot(df[column])
11        plt.xlabel(column, fontsize=20)
12
13        plotnumber+=1
14 plt.tight_layout()
```







### Observations from the distplots

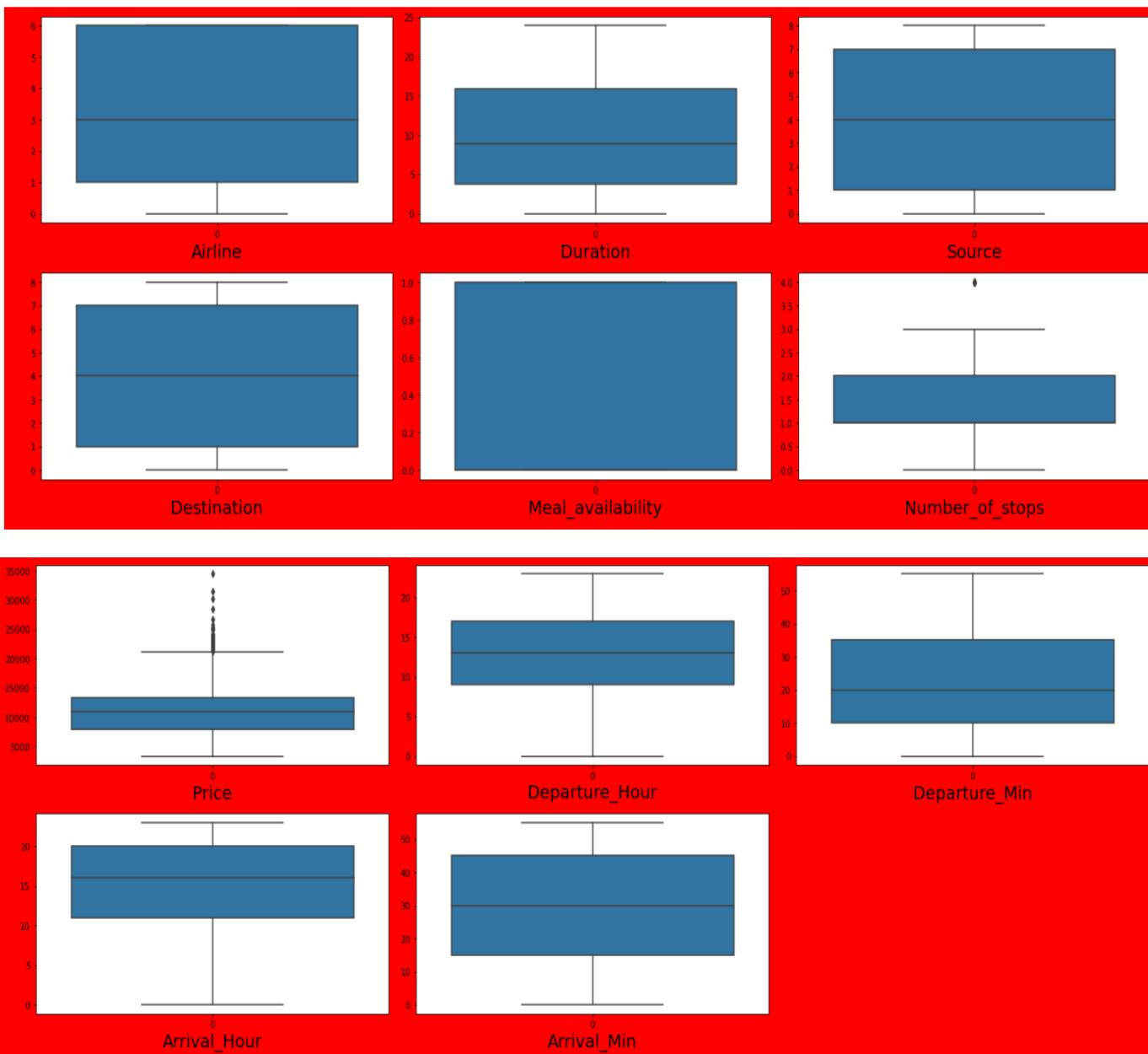
- From the distribution plot we can observe the columns are somewhat distributed normally as they have no proper bell shape curve.
- The columns like "Duration", "Number\_of\_stops" and "Price" are skewed to right as the mean value in these columns are much greater than the median(50%).
- Also the data in the column Arrival\_Hour skewed to left since the mean values is less than the median.
- Since there is presence of skewness in the data, we need to remove skewness in the numerical columns to overcome with any kind of data biasness.

Now lets plot box plots to look for outliers in the data.

```

1 #Now Lets find the outliers by plotting box plots
2
3 plt.figure(figsize=(20,15), facecolor='red')
4 plotnumber=1
5
6 for column in df:
7     if plotnumber<=12:
8         plt.subplot(4,3,plotnumber)
9         ax=sns.boxplot(data=df[column])
10        plt.xlabel(column, fontsize=20)
11
12        plotnumber+=1
13 plt.tight_layout()

```



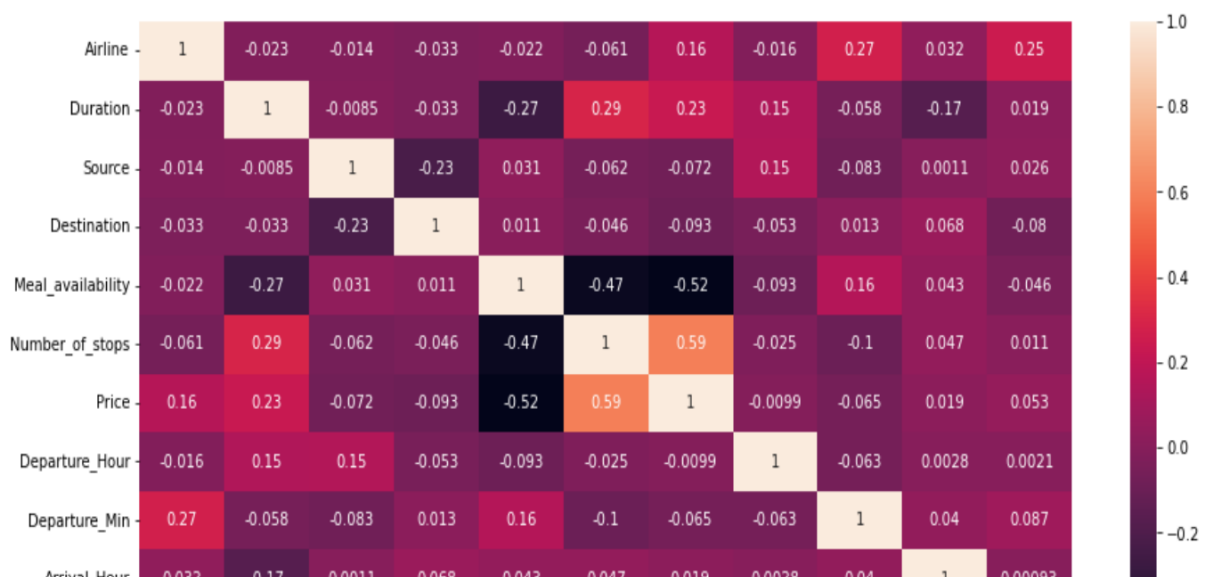
## Observations from the box plots

- The outliers present in Number\_of\_stops and "Price" columns.
- Since Price is our target column and Number\_of\_stops is our categorical variable so no need to remove outliers in this columns. Finally there is no need to remove outliers in the dataset.

Plotting heatmap to find out the correlation among the features and labels and multicollinearity.

```
1 #now lets find out the co relation using heatmap plot.  
2 plt.figure(figsize=(15,7))  
3 sns.heatmap(df.corr(),annot=True)
```

<AxesSubplot:>



## Observation from the heatmap

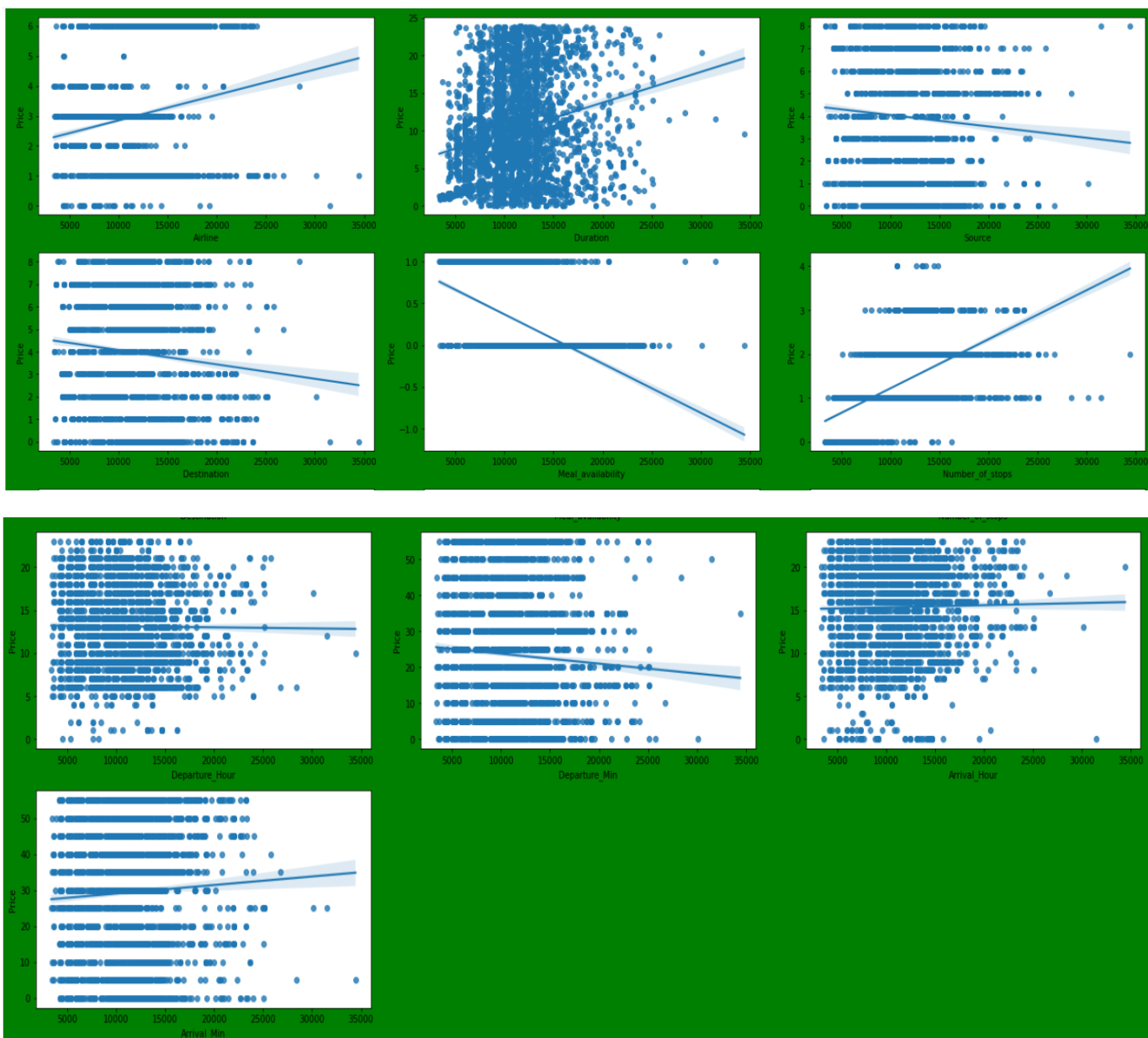
- This heat map contains both positive and negative correlation.
- The features Number\_of\_stops, Duration Arrival\_Hour and Airline are highly positively correlated with the target column compared to other features.
- The other features have very less correlation with the target column.
- From the map we can also observe there is no multicollinearity issue exists.

Plotting Regplots to find out the relationship between the features and labels.

```

1 #visualizing relationship between Labels and features
2 plt.figure(figsize=(20,15), facecolor='green')
3 plotnumber=1
4
5 for column in x:
6     if plotnumber<=11:
7         ax=plt.subplot(4,3,plotnumber)
8         sns.regplot(y,x[column])
9         plt.xlabel(column, fontsize=10)
10        plt.ylabel('Price',fontsize=10)
11
12        plotnumber+=1
13 plt.tight_layout()

```



observation from the regplot

- airlines have positive relation with ticket prices.
- we can say as the stops shows a negative relationship with ticket price decreases.
- Flight source has positive relationship with the price.
- flights having Free meal facility have high ticket prices.
- there is no significant difference between price and departure min.
- we can conclude arrival hour has some positive correlation with price.
- We can say flight ticket prices are not much dependent on the Arrival\_min.
- we can observe some positive linear relation between Duration and Price.

### 3.6 Interpretation of the Results

**Visualizations:** I have used count plots to visualize the counts in categorical variables and distribution plot to visualize the numerical variables. I have used strip plots, reg plots and box plots to check the relation between label and the features. The heat map and bar plot helped me to understand the correlation between dependent and independent features. Detected outliers and skewness with the help of box plots and distribution plots respectively. And I found some of the features skewed to right as well as to left. I got to know the count of each column using bar plots.

**Pre-processing:** The dataset should be cleaned and scaled to build the ML models to get good predictions. I have performed few processing steps which I have already mentioned in the pre-processing steps where all the important features are present in the dataset and ready for model building.

**Model building:** After cleaning and processing data, I performed train test split to build the model. I have built multiple regression models to get the accurate R2 score, and evaluation metrics like MAE, MSE and RMSE. I got RandomForestRegressor as the best model which gives 71 % R2 score. After tuning the best model, the R2 score of RandomForestRegressor has not been

increased. Finally, I saved my final model and got the good predictions results for price of flight tickets.

## **4.CONCLUSION**

### **4.1 Key Findings and Conclusions of the Study**

The case study aims to give an idea of applying Machine Learning algorithms to predict the price of the flight tickets. After the completion of this project, we got an insight of how to collect data, pre-processing the data, analyse the data, cleaning the data and building a model.

In this study, we have used multiple machine learning models to predict the flight ticket price. We have gone through the data analysis by performing feature engineering, finding the relation between features and label through visualizations. And got the important feature and we used these features to predict the car price by building ML models. Performed hyper parameter tuning on the best model and the best model's R2 score has not increased from 71%. We have also got good prediction results of ticket price.

#### **Findings:**

##### **1. Do airfares change frequently? Do they move in small increments or in large jumps?**

Flight ticket prices change during the morning and evening time of the day. From the distribution plots we came to know that the prices of the flight tickets are going up and down, they are not fixed at a time. Also, from this graph we found prices are increasing in large amounts.

##### **2. Do they tend to go up or down over time?**

Some flights are departing in the early morning 3 AM having most expensive ticket prices compared to late morning flights. As the time goes the flight ticket fares increased and midnight flight fares are very less (say after 10 PM). Also, from categorical and numerical plots we found that the prices are tending to go up as the time is approaching from morning to evening.

### **3. What is the best time to buy so that the consumer can save the most by taking the least risk?**

From the categorical plots (bar and box) we came to know that early morning and late-night flights are cheaper compared to working hours.

### **4. Does price increase as we get near to departure date?**

From the categorical plots we found that the flight ticket prices increase as the person get near to departure time. That is last minute flights are very expensive.

### **5. Is Indigo cheaper than Jet Airways?**

From the bar plot we got to know that both Indigo and Spicejet airways almost having same ticket fares.

### **6. Are morning flights expensive?**

Not all flights are expensive during morning, only few flights departing in the early morning 3 AM are expensive. Apart from this the flight ticket fares are less compared to other timing flight fares.

## **4.2 Learning Outcomes of the Study in respect of Data Science**

While working on this project I learned many things about the features of flights and about the flight ticket selling web platforms and got the idea that how the machine learning models have helped to predict the price of flight tickets. I found that the project was quite interesting as the dataset contains several types of data. I used several types of plotting to visualize the relation between target and features. This graphical representation helped me to understand which features are important and how these features describe price of tickets. Data cleaning was one of the important and crucial things in this project where I dealt with features having string values, features extraction and selection. Finally got RandomForestRegressor as best model.

The challenges I faced while working on this project was when I was scrapping the real time data from yatra website, it took so much time to gather data. Finally, our aim was achieved by predicting the flight ticket price and built flight price evaluation model that could help the buyers to understand the future flight ticket prices.

### **4.3 Limitations of this work and scope for future work**

**Limitations:** The main limitation of this study is the low number of records that have been used. In the dataset our data is not properly distributed in some of the columns many of the values in the columns are having string values which I had taken care. Due to some reasons our models may not make the right patterns and the performance of the model also reduces. So that issues need to be taken care.

**Future work:** The greatest shortcoming of this work is the shortage of data. Anyone wishing to expand upon it should seek alternative sources of historical data manually over a period of time. Additionally, a more varied set of flights should be explored, since it is entirely plausible that airlines vary their pricing strategy according to the characteristics of the flight (for example, fares for regional flights out of small airports may behave differently than the major, well flown routes we considered here). Finally, it would be interesting to compare our system's accuracy against that of the commercial systems available today (preferably over a period of time).