DATA 601 | Project 2| NY-DC/MD/VA Flights

Chiranjib Dutta, Asad Khan
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
(Dated: December 19, 2021)

INTRODUCTION

In today's age flight delays have been a significant problem for aviation companies. This issue has cost billions of dollars to aviation companies and even deterred some customers because of losing time from delays. Some people might miss time sensitive meetings or may be unable to make it to important celebrations and events with their families due to these delays. Tackling this problem is important. We have a collection of flight data that was granted to us to produce reports to help analyze patterns and make inferences from the results. In this report, we will compare the data with the cause of cancellation/delays of flight, understand and predict the trend based on flight data from the year 2013 using data analysis and machine learning tools to help give insight as to what may cause flight delays.

1.1 Calculate the total number of seats for all the planned flights for each destination separately?

We begin with calculating the total number of seats for each planned flight that departs from their origin (EWR,JFK and LGA) to their destination airports (BWI,DCA,IAD) and we group the data by the origin to summarize flights from where they depart from. Figure 1 illustrates this step in the picture. We then list the seats for each of the flights that we have data on. Upon observation, we come to know that the flights that took off from LGA to destination (BWI,DCA,IAD) had the most seats in their respective groups. The result is illustrated in Figure 1.

		Total_seat
origin	dest	
EWR	BWI	4645
	DCA	9479
	IAD	6785
JFK	BWI	4754
	DCA	9646
	IAD	12153
LGA	BWI	213
	DCA	71497

Figure: 1

1.2 What was the day of the year with the highest number of flights?

For this information, we started by gathering data from our main data frame which has all the information we need regarding flights. We grouped by the “day” column and applied an aggregationmethod on flights to count the total number of flights that took off on each day. This allowed us to sort the total flights in ascending order. By doing this we were able to find out that on day 86 the total number of flights were 56 which had the highest number of flights in comparison to the other days. Figure 2 illustrates the data we were working with sorted in descending order. The result is illustrated in Figure 2.

DOY	Total_flights
86.0	56
10.0	55
73.0	55
38.0	54
17.0	54
...	...
333.0	18
358.0	17
359.0	16
365.0	13
366.0	5

Figure: 2

1.3 What was the day of the year with the highest number of seats?

In order for us to find out the day of the year with the highest number of seats, we again started by grouped by the “day” column and applied an aggregation method on seats to find the sum of the total number of seats that took off on each day. Next, we sorted the total seats in ascending order in order for us to be able to select and isolate the target number we were searching for. We were able to confirm that on day 268, the total number of seats were 5911 which was the highest number in comparison to the other days. The result is illustrated in Figure 3.

DOY	Tot_aval_seats
268.0	5911

Figure: 3

2.1 What day of the year do most cancellations happen?

For this day, we first decided to use the data in the “arrival time” field. The reason is because we know if the arrival time is (“Empty”, “None” or 0) this will mean that the flight never arrived or it got canceled. After that we start changing all the (“Empty”, “None” or 0) into an arbitrarily selected integer (such as -9999) to indicate that this will represent a cancellation. Since the question asked us to find the most cancellations in a single day, our next step was to use this data on arrival times where the value represents a cancellation and then group it by the “day” field. We performed a count values and sorting method to find the highest number of cancellations in a single day. By following this method, we observed that the most canceling flights in a day were on the sixty five day and that there were a total of 21 flights that were canceled on day sixty five.The result is illustrated in Figure 4.

DOY	arr_time	no_of_times_flight_cancelled
65.0	-9999.0	21
39.0	-9999.0	18
67.0	-9999.0	11
143.0	-9999.0	11
255.0	-9999.0	10
...	...	...
209.0	-9999.0	1
74.0	-9999.0	1
205.0	-9999.0	1
78.0	-9999.0	1
3.0	-9999.0	1

Figure: 4

2.2) Is there any relationship between the weather datasets and cancellations? If yes, describe it and justify with some numbers.

We are convinced that there is a relationship between the weather datasets and cancellations. We used two different data sets to investigate if there was an observable trend between the cancellation flights and daily weather in New York. We decided to compare canceled flights with each column of New York daily weather and we noticed that “Precipitation” likely had the most effect on the cancellation so we took the results from two different data frames containing weather and cancellation data and then compared them with one another. Since all the data in precipitation was less than 1 and greater than 0, we decided to change all the (string “T”) values to zeros which represented empty rows for us. After this step we used a date time module to separate the number of the day from the full date and then load the day value in a new column we labeled day. Our next step was to use precipitation data which was grouped by “day.” We performed a sum and sorting method to add all the precipitation and then compare it with the cancellation flights as it is shown below in the picture. Upon observation, we noticed that the number of times a flight gets canceled can be associated with weather conditions. It appears that snowfall and precipitation seem to be higher on the same day, or right next to the day a flight is canceled and that when there is no snowfall or precipitation, cancellations are not as high. It is within reason to conclude that there is a relationship between weather and cancellations based on the data. The result is illustrated in Figure 5.

DOY	Precipitation	Snowfall	arr_time	no_of_times_flight_cancelled
65.0	1.00	1.0	-9999.0	21
39.0	0.78	2.9	-9999.0	18
67.0	0.27	3.0	-9999.0	11
143.0	0.03	0.0	-9999.0	11
255.0	0.29	0.0	-9999.0	10
...	...	...	...	...
209.0	0.07	0.0	-9999.0	1
74.0	0.00	0.0	-9999.0	1
205.0	0.00	0.0	-9999.0	1
78.0	0.46	0.0	-9999.0	1
3.0	0.00	0.0	-9999.0	1

Figure: 5

2.3 Is there any relationship between the Federal Holiday Schedule and cancellations? If yes, describe it and justify with some numbers.

We are convinced that there is a relationship between Federal Holidays and cancellations. In order for us to make a relationship between canceled flights and federal holidays more apparent, we used two different data sets such as (df_with_fed_hol and df_cancelled_flights) and merged it based on the day of the year column “DOY.” We used a group by columns “DOY” and “Fel_hol” and counted the total number of canceled flights (using the count() aggregate function). We came to the conclusion that only Labor day and veterans day had an effect on the cancellation flight, which was surprising to us because our expectations on the holidays to have the greatest impact on cancellations was Thanksgiving and

Christmas. Nonetheless, there is a noticeable impact that Federal Holidays have on cancellations and the data suggests that there is a relationship between the Federal Holiday Schedule and cancellations. The result is illustrated in Figure 6.

			no_of_times_flight_cancelled
DOY	Fed_hol	arr_time	
245.0	Labor Day	-9999.0	2
315.0	Veterans Day	-9999.0	1

Figure:6

2.4 What is the total number of seats for the canceled flights? If we assume the average flight price of $50, what is the total economic loss?

For this, we created a new data frame from the original data frame which only had the information of all the canceled flights such as (seats, tail number, carrier etc). After that we took a sum of all the seats in (canceled flights/new data frame) and multiplied it by fifty in order for us to find total economic loss in dollars. We found out that aviation companies are losing millions of dollars every year. nThe result is illustrated in Figure 7.

```
Total cancelled seats 27690
Total loss $ 1384500
```

Figure: 7

2.5 Determine the ratio of canceled flights/planned flights for each airline company, list it, and determine the most and least reliable airline company (most reliable = the one that has the smallest ratio of canceled/planned)

ratio	carrier
0.002901	US
0.004493	9E
0.011852	B6
0.033816	WN
0.062193	EV
0.106109	YV
0.178571	MQ

Figure:8

For this approach, we created a new dataframe that had only information of canceled flights. We used a grouped by and count method to get the total number of canceled flights per airline carrier from that data frame and then add this information to our original data frame under the column name ‘sp_airline_canceled_tot’ using lambda function.

In order for us to find out the total number of planned flights that reached their destinations. We used a grouped by and count method to get the total number of planned flights for each carrier from the original data frame and then add this information to our original data frame under the column name 'planned_airline_tot_flights' using lambda function. We created a new column called 'ratio' found the ratio between ((sp_airline_canceled_tot)/ (planned_airline_tot_flights)) and sorted by most reliable carrier

We came to the conclusion that the top 3 reliable airlines (US is the 1st, 2nd is 9E and 3rd is B6) and last 3 reliable airlines are (MQ is in the bottom, then YV and then EV). The result is illustrated in Figure 8.

3.1 Calculate the average arrival delay for all the flights that took place on the same day and plot it (x = 1:365, y = daily average delay). On the same plot, please mark the Federal Holidays from the federal-holidays-2013.xlsx dataset.

For this, we created a new data frame called (df_New) from the old data frame and dropped all rows where arrival time equals -9999. As we mentioned in previous questions, we used -9999 to fill the empty rows in the df["arr_time"] column. Since the new data frame had a 'day' column for the day of year number, the team decided to create a new column called (day of year "DOY"). Next, we used a group by method to find an average delay per day. We plotted this information on a bar graph by setting the x value to the total number of days in a year. Since we needed to plot the holiday on the same graph as on "daily average delay". So my team decided that we use the "dayofyear method" to take the day from the date column and put it under "DOY". Doing this made it easier for us to plot and compare different data sets on the same graph as it is shown in the picture below.

We came to the conclusion that the months (May, July, Oct, Nov and Dec) had the most delays of flights due to federal holidays compared to the daily average delay. The result is illustrated in Figure 9.

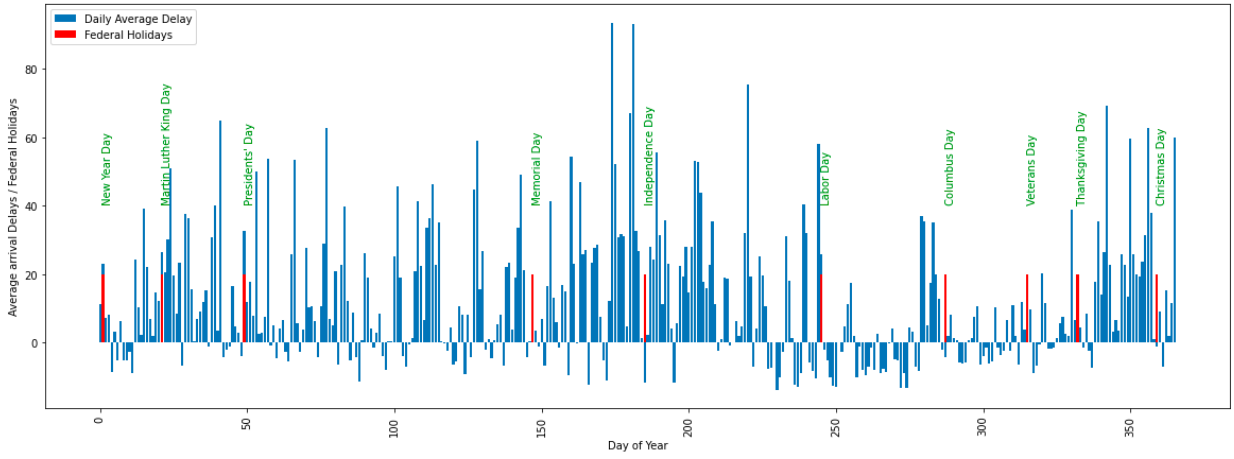


Figure: 9

3.2 Is there a correlation between the weather datasets and daily average arrival delay?

We are convinced that there is a correlation between the weather and daily average arrival delay. We are starting with a data frame containing information on the daily weather of New York, “df_wthNYdaily” and set a new data frame called df_change = df_wthNYdaily while changing the index to day of year “DOY.” In our next step, we added the column called daily_avg_delay to the df_change. In order for us to find the relationship between 2 datasets we used a correlation method and compared daily average delay with every type of weather in the weather dataset. We came to the conclusion that most of the delayed flights were caused by precipitation and snowfall. For example, Precipitation had 25% and snowfall had 16% correlation with the daily flight average delay compared to other weather. The correlation map is illustrated in Figure 10.

	Max Temp	Min Temp	Precipitation	Snowfall	DOY	daily_avg_delay
Max Temp	1.000000	0.964321	0.000043	-0.355048	0.286125	-0.000391
Min Temp	0.964321	1.000000	0.050843	-0.331652	0.273300	0.075351
Precipitation	0.000043	0.050843	1.000000	0.186863	-0.030841	0.248583
Snowfall	-0.355048	-0.331652	0.186863	1.000000	-0.116790	0.162302
DOY	0.286125	0.273300	-0.030841	-0.116790	1.000000	-0.092672
daily_avg_delay	-0.000391	0.075351	0.248583	0.162302	-0.092672	1.000000

Figure: 10

3.3 Is there a correlation between the Federal Holiday Schedule and daily average arrival delay?

We are convinced that there is a correlation between Federal Holidays and daily average arrival delay. We started with the Federal Holiday Schedule and set a new data frame called df_change = Fed_hol while changing the index to day of year “DOY.” We then added a column called daily_avg_delay to the dataframe df_change to store the information we were looking for. In order for us to find the relationship between two datasets we used a correlation method and compared daily average delay with the Federal Holiday Schedule. We came to the conclusion that the daily average arrival delay is around 46% which is related to the federal holidays . The correlation map is illustrated in Figure 11.

	day	month	year	DOY	daily_avg_delay
day	1.000000	0.163114	nan	0.237399	-0.466691
month	0.163114	1.000000	nan	0.997113	0.064802
year	nan	nan	nan	nan	nan
DOY	0.237399	0.997113	nan	1.000000	0.028719
daily_avg_delay	-0.466691	0.064802	nan	0.028719	1.000000

Figure: 11

3.4 Calculate the average arrival delay for all the flights for each arrival airport (e.g. IAD, DCA, and BWI) and determine most and least reliables (most reliable = the one that has the shortest average delay)

Using data that we mentioned earlier in 3.1 that we have in the dataframe called "df_New," we chose to drop all the (empty "-9999 ") rows in the arrival time for this task. We used the data frame to count the average arrival delay of each carrier using aggregate method and then grouped by the destination. We came to the conclusion that on the destination BWI, the most reliable carrier was WN with an average delay of 4.91 minutes and the least reliable carrier was EV which had an average delay of about 20 minutes. For the destination DCA, the most reliable carrier was DL which was 8 minutes earlier than the daily average and the least reliable carrier was MQ which had an average of about 28 minutes of average delay. For IDA, the most reliable carrier was UA which was 24 minutes earlier than the daily average and the least reliable YV carrier which had a delay average of 18.91 minutes. The result is illustrated in. Figure 12.

		aver_delay
dest	carrier	
BWI	9E	8.731288
	EV	20.056047
	WN	4.915000
DCA	9E	0.072206
	DL	-8.000000
	EV	21.340575
	MQ	28.195652
	UA	0.500000
	US	5.829000
IAD	9E	2.642623
	B6	12.805097
	EV	15.489133
	OO	3.000000
	UA	-24.000000
	YV	18.917266

Figure: 12

3.5 Calculate the average arrival delay for all the airlines and determine most and least reliables (most reliable = the one that has the shortest average delay)

In order for us to find arrival delay for all the airlines we used a carrier and arrival delay from df_New data frame. We then performed an aggregate method to find the average of delay and then grouped it by carriers and sorted by arrival delay. We came to the conclusion that the most reliable airline was DL which was 8 minutes earlier than the arrival time and the least reliable airline was MQ which was 28 minutes later than its arrival time. The result is illustrated in. Figure 13.

carrier	aver_delay
DL	-8.000
UA	-7.667
OO	3.000
9E	3.613
WN	4.915
US	5.829
B6	12.805
EV	17.360
YV	18.917
MQ	28.196

Figure: 13

3.6 What day of the week did we have the highest average delay?

week_day	weekday_delay
Wednesday	11.794330
Tuesday	11.176183
Thursday	13.106149
Sunday	10.094053
Saturday	11.124157
Monday	9.009183
Friday	10.903725

Figure: 14

To determine this, we used the datetime module to find a “day_name” from the date column and then set that value under the week_day column in df_New. We then perform an aggregate method to find the average delay and set the value under the new column called “weekday_avg” and then group it by week_day and sort by arrival delay. We came to the conclusion that most delays happened on wednesdays. The result is illustrated in. Figure 14.

3.7 Which one had a higher average delay: flights that took off in the morning (6 am to 10 am), noon (11 am to 2 pm), afternoon (3 pm to 5pm), or evening (6 pm to 10 pm)?

For this, we created a function called “day_session” and passed the “dep_time” column to find a higher average of delay for those flights that took off each session of the day. In order for us to return the correct part of the day. After that we created a new column in df_New called “session” and used the apply method to assign function return values under the new column. Furthermore, we perform an aggregate method to find the higher average departure delay. and then group it by session, sort by departure delay and then set these 2 columns to new dataframe called “dfNewSession”. We came to the conclusion that the flights that took off in the afternoon had the highest amount of delay which was 22.59 minutes of delay. The confusion matrix is illustrated in. Figure 15.

day_session	aver_delay
Afternoon (3pm-5pm)	22.599612
Evening (6pm-10pm)	21.839815
Noon (11am-2pm)	9.709850
Morning (6am-10am)	0.126671

Figure: 15

3.8 Determine the number of airplanes used in these flights manufactured by BOEING, EMBRAER, and AIRBUS separately.

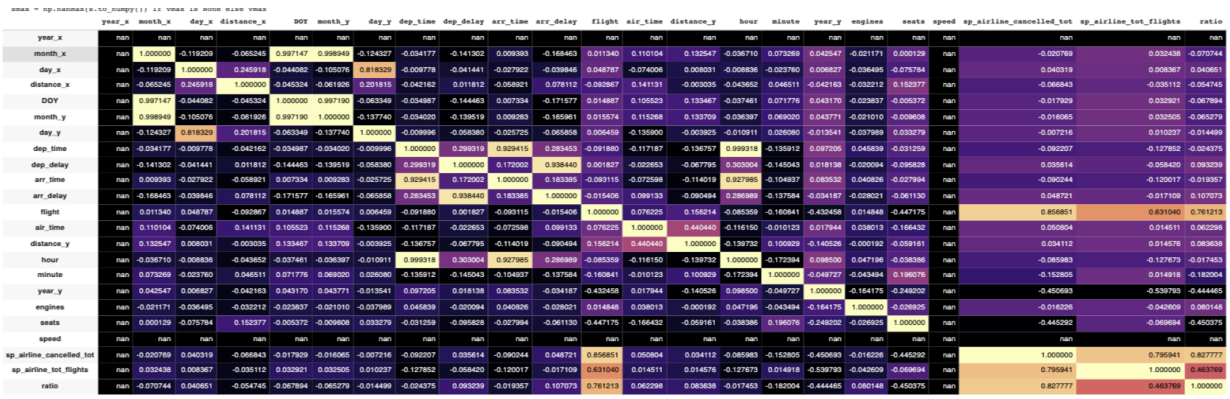
```
manufacturer
AIRBUS                    4
AIRBUS INDUSTRIE       3929
BOEING                  201
BOMBARDIER INC         3762
CANADAIR               1491
CANADAIR LTD             37
CESSNA                    8
EMBRAER                4399
GULFSTREAM AEROSPACE      1
dtype: int64
```

Figure: 16

To determine the number of airplanes used in these flights, we performed a group by method on the manufacturer column that we took from df_New and applied a size method to find the total number of planes by each manufacturer and then set those values to a new table called carrier_total_flights. We came to the conclusion that the Airbus industry had the most planes compared to other manufacturers. The result is illustrated in. Figure 16.

4. Build a linear regression model to estimate the arrival delay of the flights given in "fligts_test_data.xlsx". Note that you have the full authority to decide what columns, what datasets (among the given datasets) to work with. In your report, please explain how you build the LR model and elaborate on its accuracy.

We have created a new data frame called “df_FTD” from fligts_test_data.xlsx and inserted a column (day of year “DOY”) to take the day from datetime and change it to day of year and then set that value under the DOY column in df_FTD. In order for us to prepare the data for testing we merged df_FTD and df_New on DOY and set this merging equals to df_test. In order for us to perform linear regression on arrival delay we will need to find out the column that has the strongest correlation with arrival delay column and we used a correlation matrix to find those columns such as (departure time, departure delay, DOY, arrival delay). We set all the values equal to X-Values that had the strongest correlation with arrival delay and set Y-Value equal to arrival delay. We split 80% of the data to train and another 20%, random state 1 and use it for testing while choosing a linear regression model. The conclusion is that we had a higher RMSE value of 16.21% and R2 was around 88%. We believe we could have got much more accurate results if we had larger amounts of data than we had to work with. The correlation map is illustrated in Figure 17. The RMSE and R2 are illustrated in. Figure 18.



	year_x	month_x	day_x	distance_x	DOY	month_y	day_y	dep_time	dep_delay	arr_time	arr_delay	flight	air_time	distance_y	hour	minute	year_y	engines	seats	speed	sp_airline_cancelled_tot	sp_airline_tot_flights	ratio
year_x	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
month_x	nan	1.000000	-0.119209	-0.065245	0.997147	0.998949	-0.124327	-0.034177	-0.141302	0.009393	-0.168463	0.011340	0.110104	0.132547	-0.036710	0.073269	0.042547	-0.021171	0.000129	nan	-0.020769	0.032436	-0.070744
day_x	nan	-0.119209	1.000000	0.245918	-0.044082	-0.105076	0.818329	-0.009778	-0.041441	-0.027922	-0.039846	0.048787	-0.074006	0.008031	-0.008836	-0.023760	0.006827	-0.036495	-0.075784	nan	0.040319	0.008367	0.040651
distance_x	nan	-0.065245	0.245918	1.000000	-0.045324	-0.061926	0.201815	-0.042162	0.011812	-0.058921	0.078112	-0.092867	0.141131	-0.003035	-0.043652	0.046511	-0.042163	-0.032212	0.152377	nan	-0.066843	-0.035112	-0.054745
DOY	nan	0.997147	-0.044082	-0.045324	1.000000	0.997190	-0.063349	-0.034987	-0.144463	0.007334	-0.171577	0.014887	0.105523	0.133467	-0.037461	0.071776	0.043170	-0.023837	-0.005372	nan	-0.017929	0.032921	-0.067894
month_y	nan	0.998949	-0.105076	-0.061926	0.997190	1.000000	-0.137740	-0.034020	-0.139519	0.009283	-0.165961	0.015574	0.115268	0.133709	-0.036397	0.069020	0.043771	-0.021010	-0.009608	nan	-0.016065	0.032505	-0.065279
day_y	nan	-0.124327	0.818329	0.201815	-0.063349	-0.137740	1.000000	-0.009996	-0.058380	-0.025725	-0.065858	0.006459	-0.135900	-0.003925	-0.010911	0.026080	-0.013541	-0.037989	0.033279	nan	-0.007216	0.010237	-0.014499
dep_time	nan	-0.034177	-0.009778	-0.042162	-0.034987	-0.034020	-0.009996	1.000000	0.299319	0.929415	0.283453	-0.091880	-0.117187	-0.136757	0.999318	-0.135912	0.097205	0.045839	-0.031259	nan	-0.092207	-0.127852	-0.024375
dep_delay	nan	-0.141302	-0.041441	0.011812	-0.144463	-0.139519	-0.058380	0.299319	1.000000	0.172002	0.938440	0.001827	-0.022653	-0.067795	0.303004	-0.145043	0.018138	-0.020094	-0.095828	nan	0.035614	-0.058420	0.093239
arr_time	nan	0.009393	-0.027922	-0.058921	0.007334	0.009283	-0.025725	0.929415	0.172002	1.000000	0.183385	-0.093115	-0.072598	-0.114019	0.927985	-0.104937	0.083532	0.040826	-0.027994	nan	-0.090244	-0.120017	-0.019357
arr_delay	nan	-0.168463	-0.039846	0.078112	-0.171577	-0.165961	-0.065858	0.283453	0.938440	0.183385	1.000000	-0.015406	0.099133	-0.090494	0.286989	-0.137584	-0.034187	-0.028021	-0.061130	nan	0.048721	-0.017109	0.107073
flight	nan	0.011340	0.048787	-0.092867	0.014887	0.015574	0.006459	-0.091880	0.001827	-0.093115	-0.015406	1.000000	0.076225	0.156214	-0.085359	-0.160841	-0.432458	0.014848	-0.447175	nan	0.856851	0.631040	0.761213
air_time	nan	0.110104	-0.074006	0.141131	0.105523	0.115268	-0.135900	-0.117187	-0.022653	-0.072598	0.099133	0.076225	1.000000	0.440440	-0.116150	-0.010123	0.017944	0.038013	-0.166432	nan	0.050804	0.014511	0.062298
distance_y	nan	0.132547	0.008031	-0.003035	0.133467	0.133709	-0.003925	-0.136757	-0.067795	-0.114019	-0.090494	0.156214	0.440440	1.000000	-0.139732	0.100929	-0.140526	-0.000192	-0.059161	nan	0.034112	0.014576	0.083638
hour	nan	-0.036710	-0.008836	-0.043652	-0.037461	-0.036397	-0.010911	0.999318	0.303004	0.927985	0.286989	-0.085359	-0.116150	-0.139732	1.000000	-0.172394	0.096500	0.047196	-0.038386	nan	-0.085983	-0.127673	-0.017453
minute	nan	0.073269	-0.023760	0.046511	0.071776	0.069020	0.026080	-0.135912	-0.145043	-0.104937	-0.137584	-0.160841	-0.010123	0.100929	-0.172394	1.000000	-0.049727	-0.043494	0.196076	nan	-0.152805	0.014918	-0.182004
year_y	nan	0.042547	0.006827	-0.042163	0.043170	0.043771	-0.013541	0.097205	0.018138	0.083532	-0.034187	-0.432458	0.017944	-0.140526	0.096500	-0.049727	1.000000	-0.164175	-0.249202	nan	-0.450693	-0.539793	-0.444465
engines	nan	-0.021171	-0.036495	-0.032212	-0.023837	-0.021010	-0.037989	0.045839	-0.020094	0.040826	-0.028021	0.014848	0.038013	-0.000192	0.047196	-0.043494	-0.164175	1.000000	-0.026925	nan	-0.016226	-0.042609	0.080148
seats	nan	0.000129	-0.075784	0.152377	-0.005372	-0.009608	0.033279	-0.031259	-0.095828	-0.027994	-0.061130	-0.447175	-0.166432	-0.059161	-0.038386	0.196076	-0.249202	-0.026925	1.000000	nan	-0.445292	-0.069694	-0.450375
speed	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
sp_airline_cancelled_tot	nan	-0.020769	0.040319	-0.066843	-0.017929	-0.016065	-0.007216	-0.092207	0.035614	-0.090244	0.048721	0.856851	0.050804	0.034112	-0.085983	-0.152805	-0.450693	-0.016226	-0.445292	nan	1.000000	0.795941	0.827777
sp_airline_tot_flights	nan	0.032436	0.008367	-0.035112	0.032921	0.032505	0.010237	-0.127852	-0.058420	-0.120017	-0.017109	0.631040	0.014511	0.014576	-0.127673	0.014918	-0.539793	-0.042609	-0.069694	nan	0.795941	1.000000	0.463769
ratio	nan	-0.070744	0.040651	-0.054745	-0.067894	-0.065279	-0.014499	-0.024375	0.093239	-0.019357	0.107073	0.761213	0.062298	0.083638	-0.017453	-0.182004	-0.444465	0.080148	-0.450375	nan	0.827777	0.463769	1.000000

Figure: 17

```
(524, 4)
(132, 4)
(524,)
(132,)
The model performance for training set
--------------------------------------
RMSE is 16.21943495597122
R2 score is 0.8843628168185224


The model performance for testing set
--------------------------------------
RMSE is 16.300232818725156
R2 score is 0.8704405567476945
```

Figure : 18