Coursework Report in R and Python.

I have used a subset of 3 consecutive years – 2003, 2004 and 2005 with 2000 rows from each year, aggregating to a total of 6000rows and called the dataset – delays. The sample in R and Python are different. I have supplemented this data with the plane-data and airports files from the Harvard Dataverse.

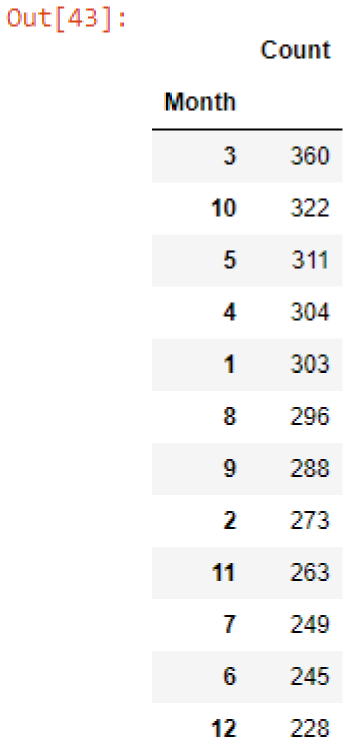
To measure Delays, I created a column called TotalDelays by adding the DepDelay and ArrDelay columns. I also created a column called Delayed, that has a binary value of 1 if the flight had a TotalDelays of greater 0 mins and value 0 if the TotalDelays was less than or equal to 0 mins.

Python

Question 1- When is the best time of the day, day of week and time of year to fly to minimise delays?

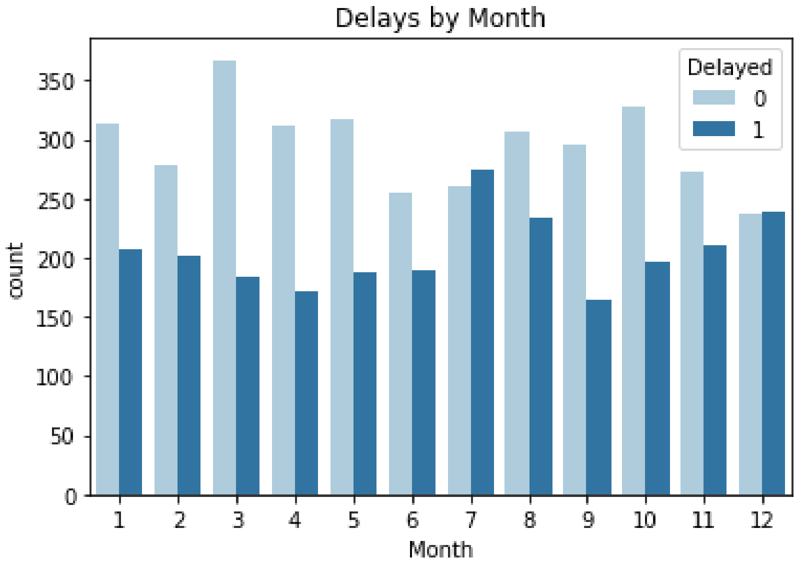
To find the answer I counted the Month, Day of Week and Time of Day which had the maximum no of flights that were not delayed or cancelled.

1- Best time of the Year – Is the Month of March



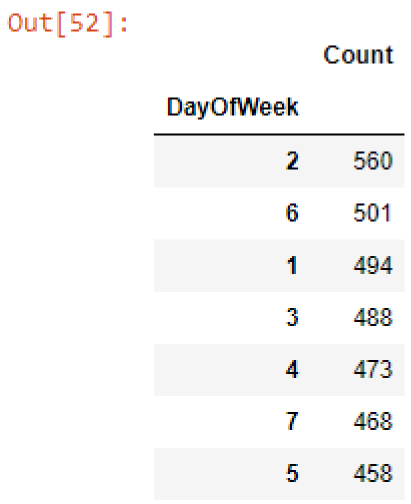
Out[43]:

	Count
Month	
3	360
10	322
5	311
4	304
1	303
8	296
9	288
2	273
11	263
7	249
6	245
12	228



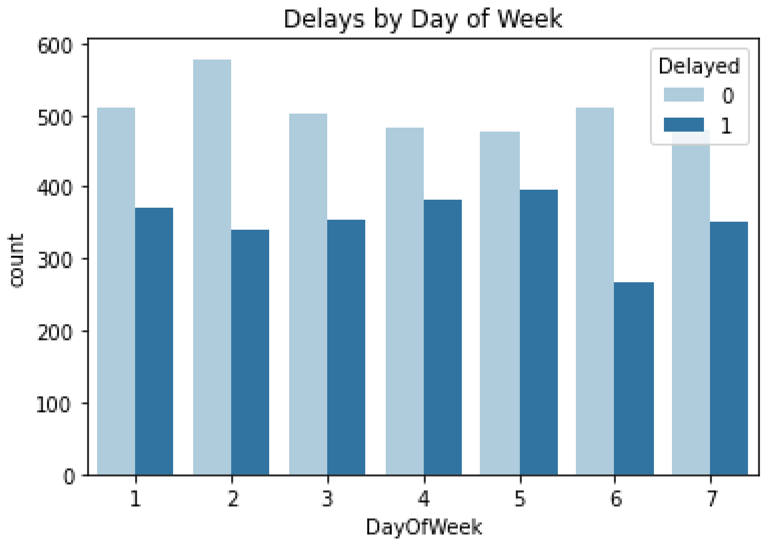
2-The best time of Week - is on Tuesday.

Monday is marked as 1 and Sunday is marked as 7.

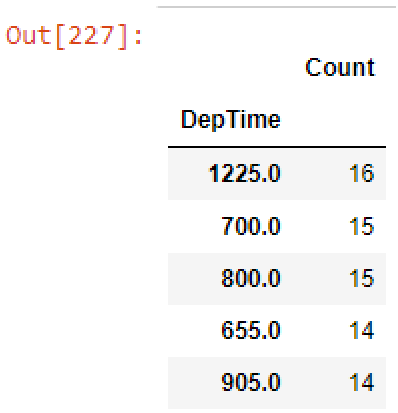


Out[52]:

DayOfWeek	Count
2	560
6	501
1	494
3	488
4	473
7	468
5	458

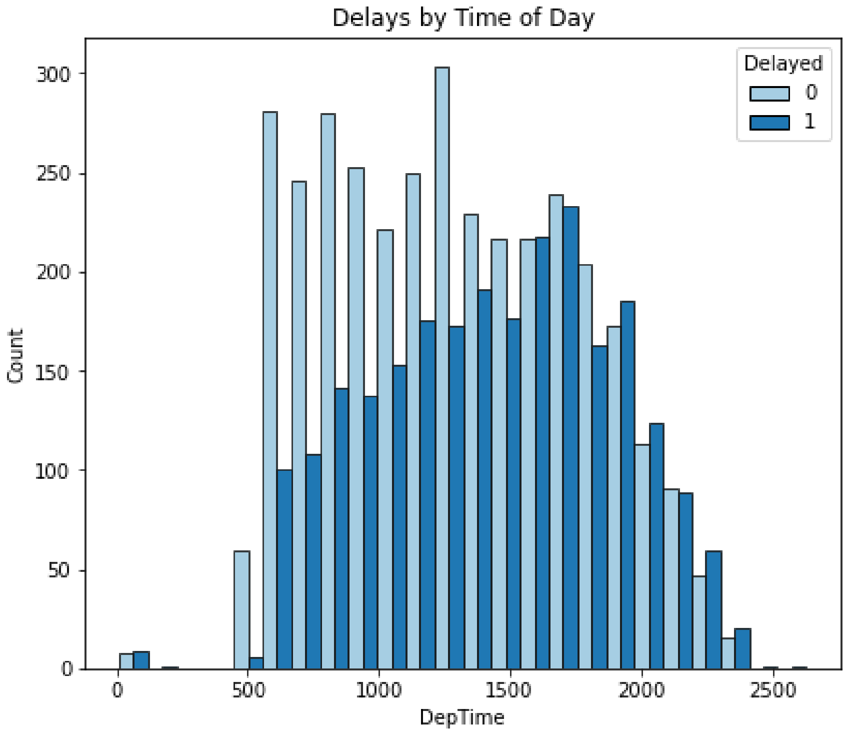


2- Best time of Day – is at 12:25 pm in the afternoon.



Out[227]:

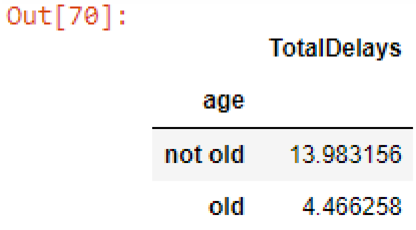
DepTime	Count
1225.0	16
700.0	15
800.0	15
655.0	14
905.0	14



Question 2 – Do older planes suffer more delays?

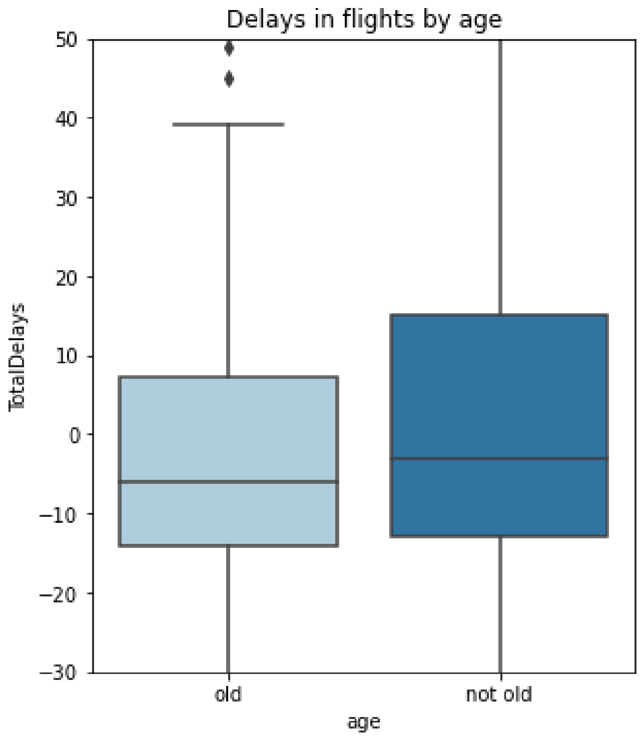
I used plane-data.csv to know the age of the plane. If a commercial plane is more than 20 years old it is considered old (reference). Hence, if the plane was built before 1984 (20 years before 2004) it is old or else it is not old. I then calculated the mean delay in mins by each group.

On average older planes are delayed for 4 mins which is much lesser than planes that are not old and have an average delay of 13 mins.



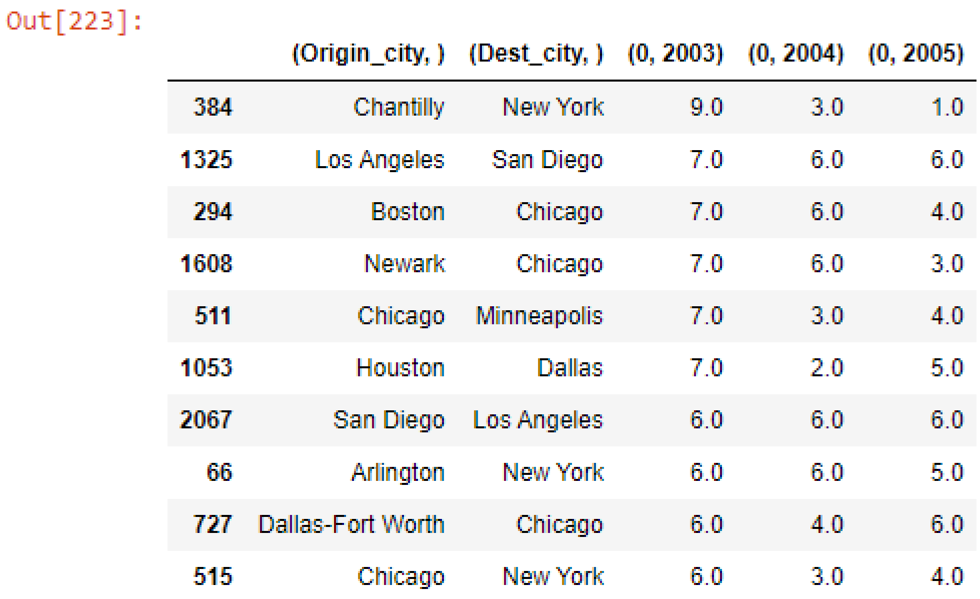
Out[70]:

age	TotalDelays
not old	13.983156
old	4.466258



Question 3 – How does the number of flights change between locations over time?

I used the airports.csv data to get information about the Origin and Destination city of the flight. I used an inner join to convert Origin to Origin_city and Dest to Dest_city. The table below shows how the number of flights between the locations change over the years. So there were 9 flights between Chantilly and New York in 2003, 3 flights in 2004 and 1 flight in 2005. I have only attached the top 10 rows due to space constraints.



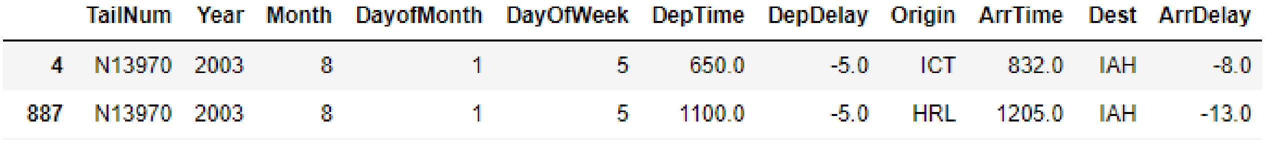
Out[223]:

	(Origin_city,)	(Dest_city,)	(0, 2003)	(0, 2004)	(0, 2005)
384	Chantilly	New York	9.0	3.0	1.0
1325	Los Angeles	San Diego	7.0	6.0	6.0
294	Boston	Chicago	7.0	6.0	4.0
1608	Newark	Chicago	7.0	6.0	3.0
511	Chicago	Minneapolis	7.0	3.0	4.0
1053	Houston	Dallas	7.0	2.0	5.0
2067	San Diego	Los Angeles	6.0	6.0	6.0
66	Arlington	New York	6.0	6.0	5.0
727	Dallas-Fort Worth	Chicago	6.0	4.0	6.0
515	Chicago	New York	6.0	3.0	4.0

Question 4 – Can you detect cascading failures as delays in one airport cause delays in others?

I have attempted this question by 3 different methods.

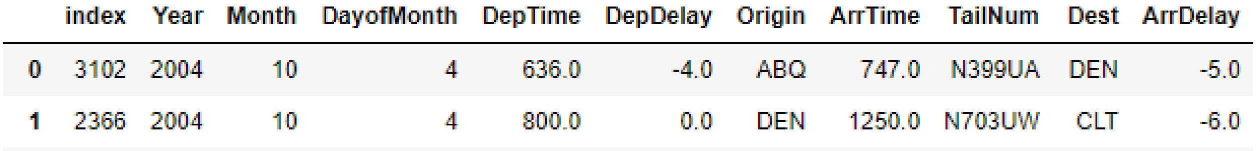
A- I have followed a flight by TailNum to see if it has a trail of delays as it flies through the day. Unfortunately, in my subset there were only 2 rows with the same TailNum in the same day.



	TailNum	Year	Month	DayofMonth	DayOfWeek	DepTime	DepDelay	Origin	ArrTime	Dest	ArrDelay
4	N13970	2003	8	1	5	650.0	-5.0	ICT	832.0	IAH	-8.0
887	N13970	2003	8	1	5	1100.0	-5.0	HRL	1205.0	IAH	-13.0

TailNum N13970 on 1st August, 2003 makes its first trip at 6:50am. It departs 5 mins early from ICT airport and lands 8 mins early at IAH airport. It makes it's second trip from IAH to HRL which is not the data, but we can guess that this trip was not delayed. It then makes its third trip from HRL airport 5 mins early and lands at IAH 13 mins early.
So, here we see that if a flight is early at departure it is early at arrival. But this is too little data to conclude cascading early flight.

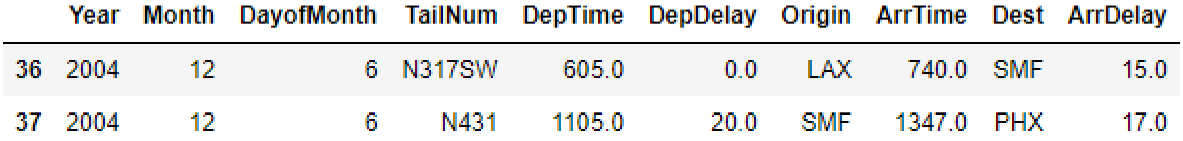
B – I tracked flights on the day that had the most number of rows in my subset. 4th October, 2004 had 14 rows in my data.



	index	Year	Month	DayofMonth	DepTime	DepDelay	Origin	ArrTime	TailNum	Dest	ArrDelay
0	3102	2004	10	4	636.0	-4.0	ABQ	747.0	N399UA	DEN	-5.0
1	2366	2004	10	4	800.0	0.0	DEN	1250.0	N703UW	CLT	-6.0

If we look at the first row the flight lands in DEN airport at 7:47am early. Another flight takes off at 8 am on time. Here too we can see if a flight departs early in one airport it lands early in another.

C – I arranged the data in order of the time in which it occurs. Then I constructed a table with Lagged Destination values so that the next flight takes off from the same airport that it landed in.



	Year	Month	DayofMonth	TailNum	DepTime	DepDelay	Origin	ArrTime	Dest	ArrDelay
36	2004	12	6	N317SW	605.0	0.0	LAX	740.0	SMF	15.0
37	2004	12	6	N431	1105.0	20.0	SMF	1347.0	PHX	17.0

From the above data on 6th Dec, 2004 we can see that there is an ArrDelay at 7:40am at SMF airport and a corresponding Departure Delay at 11:05am from SMF airport. This also leads to Arrival Delay at PHX airport. Here we notice cascading delays.

In all three circumstances, we can see that if there is an early Arrival by the previous flight there is early departure by the following flight at the same airport and vice versa with delays. But this too little evidence to conclude that there are cascading failures from one airport to the other.

Question 5 – Use the available variables to construct a model that predicts delays.

Features used:

Categorical

- Nominal

1- Cancelled 2- Diverted

- Ordinal

1- Year 2- Month 3- DayofMonth 4- DayOfWeek 5- DepHour

Numeric

- Continuous

1- DepDelay 2- Distance 3- TaxiIn 4- TaxiOut 5- CRSElapsedTime

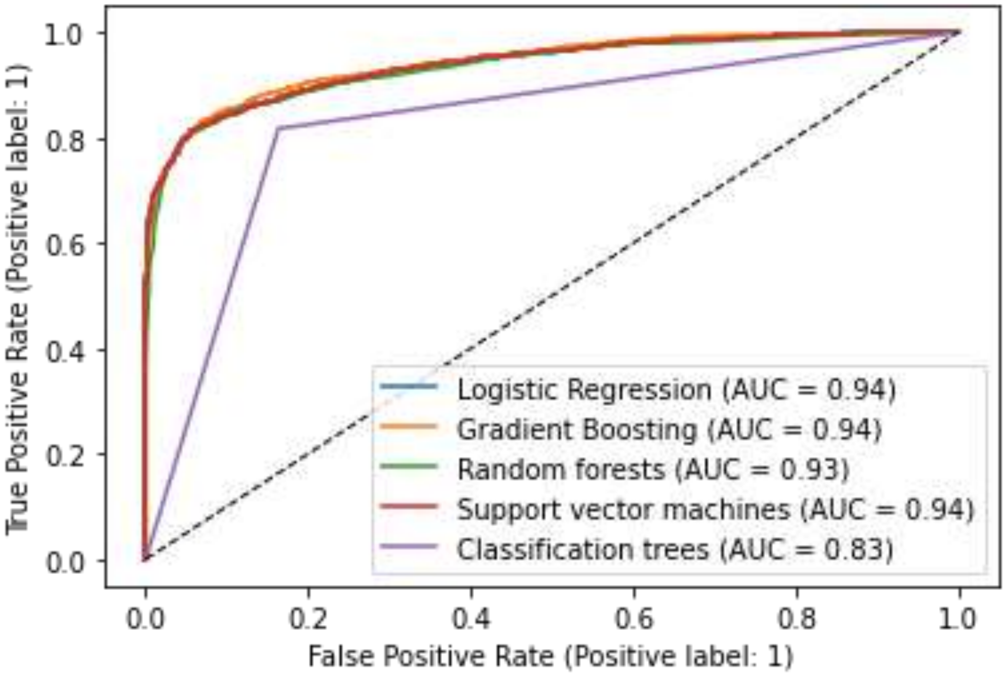
6- SecurityDelay 7- NASDelay 8- WeatherDelay 9- CarrierDelay 10 - LateAircraftDelay

Since Delayed column is made by adding the values in DepDelay and ArrDelay, giving the model both columns doesn't make any sense. Hence, I will give the model only DepDelay to see it can still predict if the flight is delayed or not**.**

Target variable is Delayed

I used 5 types of Classification models and see which one gives the best results I used ROC plot.

- Gradient Boosting
- Support Vector Machines
- Random Forests
- Logistic Regression
- Classification Trees



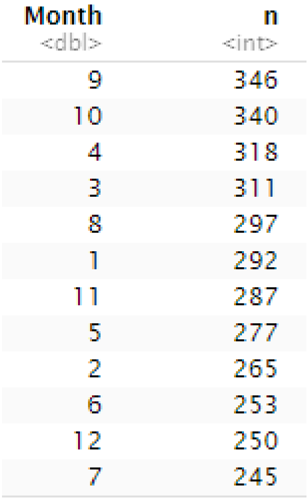
From the above diagram shows that three models - Logistic Regression, Gradient Boosting and Support Vector Machines perform well with a good score of 0.94. Random Forests gives a slightly lower score of 0.93 and Classification trees is the lowest at 0.83.

R

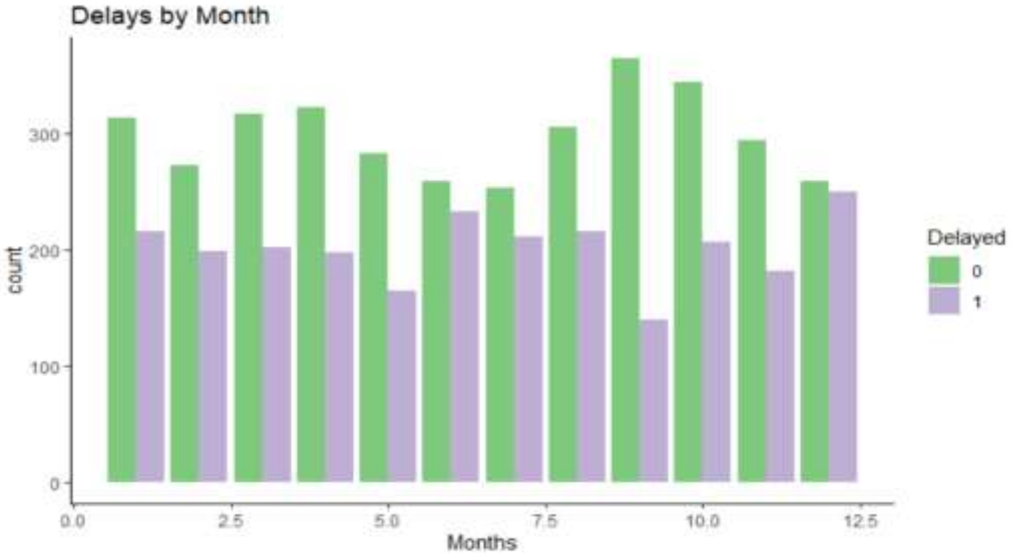
Question 1 – When is the best time of the day, day of week and time of year to fly to minimise delays?

To find the answer I counted the Month, Day of Week and Time of Day which had the maximum no of flights that were not delayed or cancelled.

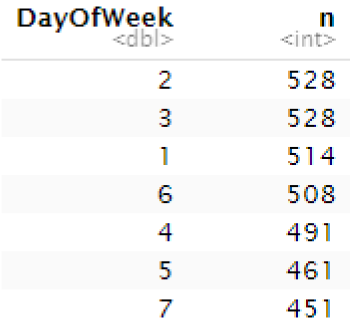
1- Best time of Year to fly to minimise delays is the month of September.



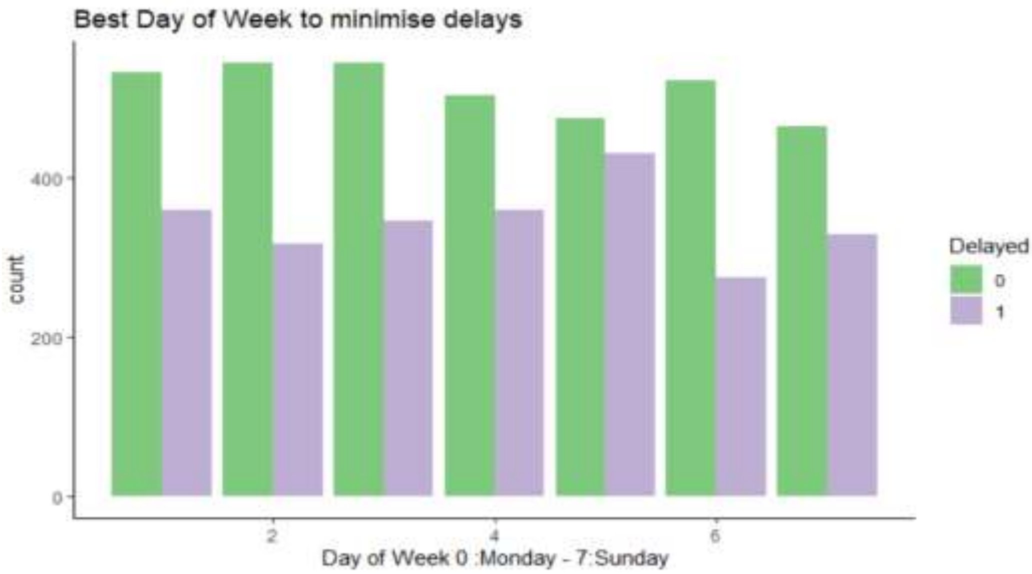
Month <dbl>	n <int>
9	346
10	340
4	318
3	311
8	297
1	292
11	287
5	277
2	265
6	253
12	250
7	245



2- Best Day of the week to fly is both on Tuesday and Wednesday.

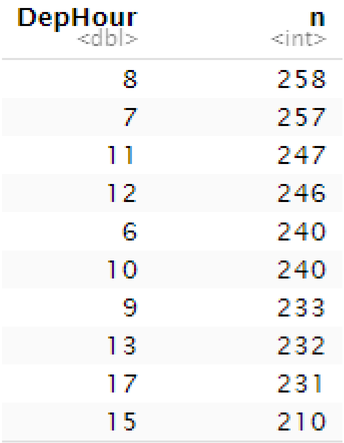


DayOfWeek <dbl>	n <int>
2	528
3	528
1	514
6	508
4	491
5	461
7	451

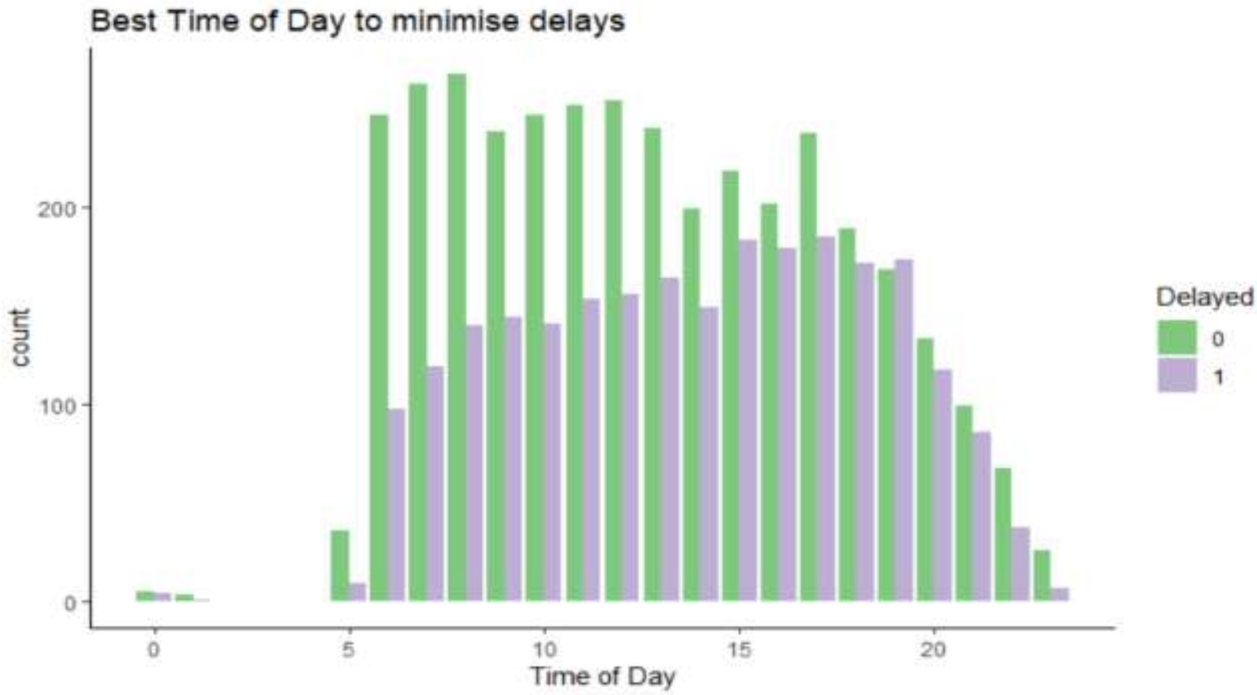


I have rounded off the time to the hour and calculated the best hour to fly in.

3- Best time of Day to fly to minimise delays is in the morning at 8am.

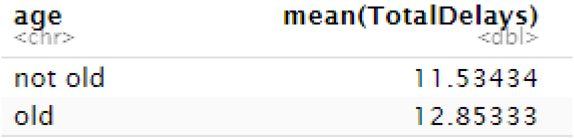


DepHour <dbl>	n <int>
8	258
7	257
11	247
12	246
6	240
10	240
9	233
13	232
17	231
15	210



Question 2 – Do older planes suffer more delays?

I did an inner join with the delays and planes dataframe and used the year column in planes to get the age of the plane. If a commerical plane is more than 20 years old it is considered old. I constructed an age column that take the value old if the plane was built before1984, else not old. I then calculated the mean delays in mins by the age.

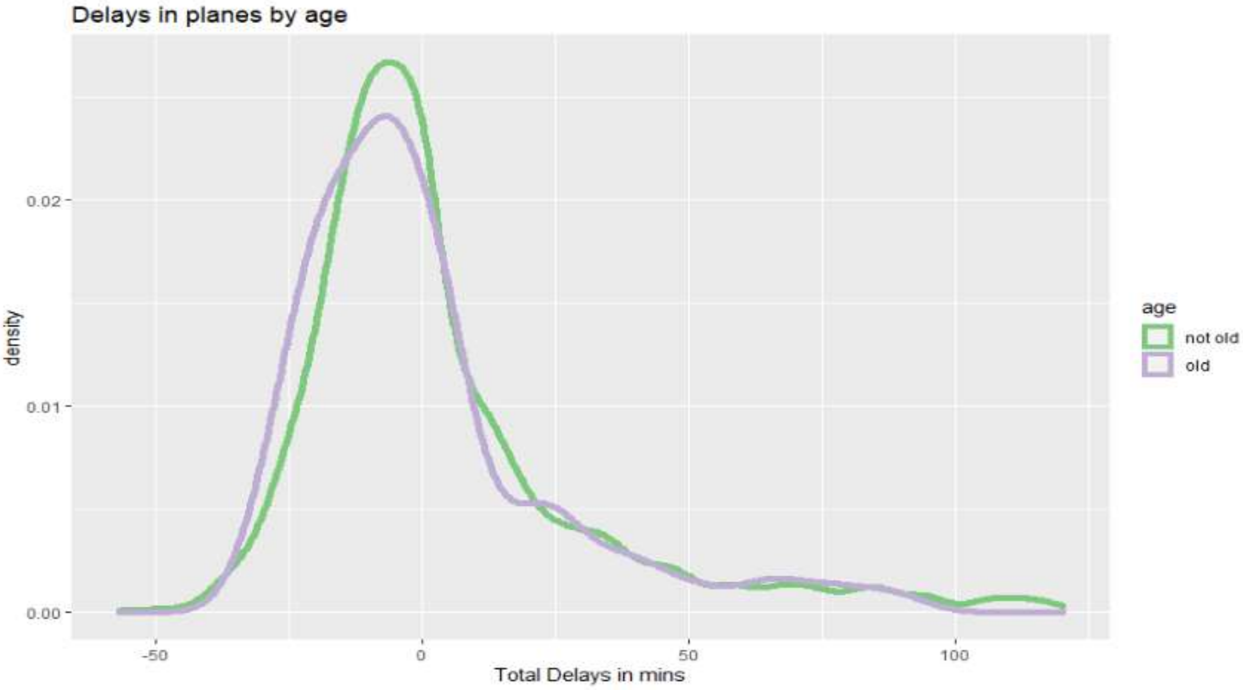


age <chr>	mean(TotalDelays) <dbl>
not old	11.53434
old	12.85333

Older planes have a mean delay of 12:85 mins.

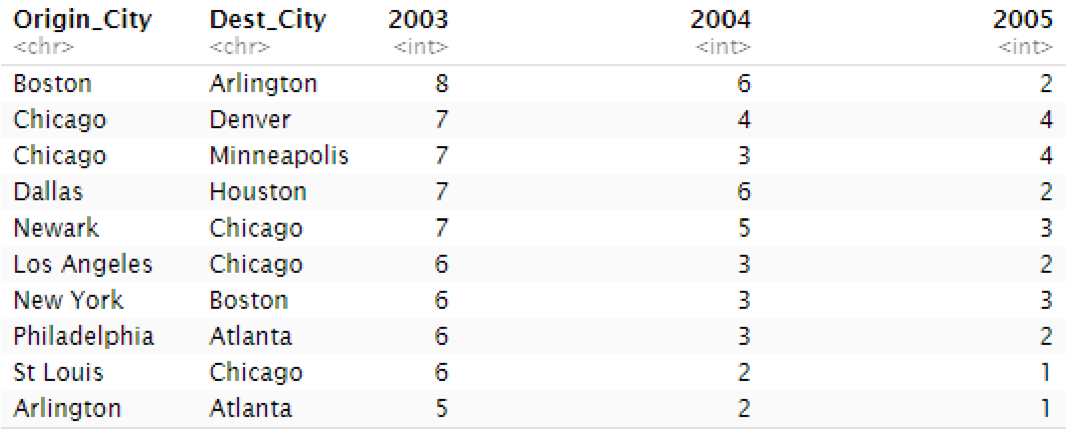
Younger planes have a mean delay of 11:53 mins.

The mean delays in mins for older planes and not old planes are very close. By looking at the distribution of the two variables, older planes have a wider spread around the mean and a shorter peak – more variance. Hence, I can conclude that older planes suffer more delays.



Question 3 – How does the number of flights change between locations over time?

To answer this question, I used the airports data to get the names of the cities by the airports. This table show the change in the number of flights between different locations over the years. It is seen that there were 8 flights between Boston and Arlington in 2003, 6 flights in 2004 and 2 flights in 2005. I have only attached the top 10 columns due to space constrictions.



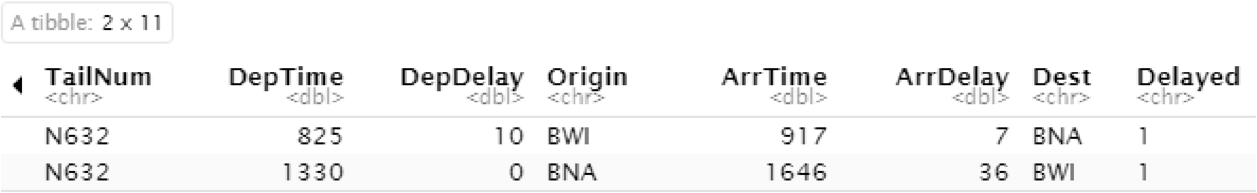
Origin_City	Dest_City	2003	2004	2005
<chr>	<chr>	<int>	<int>	<int>
Boston	Arlington	8	6	2
Chicago	Denver	7	4	4
Chicago	Minneapolis	7	3	4
Dallas	Houston	7	6	2
Newark	Chicago	7	5	3
Los Angeles	Chicago	6	3	2
New York	Boston	6	3	3
Philadelphia	Atlanta	6	3	2
St Louis	Chicago	6	2	1
Arlington	Atlanta	5	2	1

Question 4 – Can you detect cascading failures as delays in one airport cause delays in others?

To answer this question, I tried 3 methods in which cascading failures can be detected.

A – First by looking at the TailNum that occurs the most in a single day within my subset. Then I examined it to see if it displayed any cascading delays.

On the 21st March 2003, TailNum N632 made 2 trips.



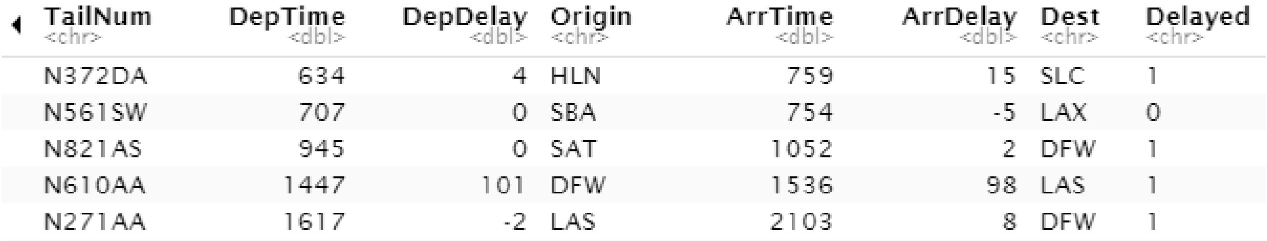
A tibble: 2 x 11

TailNum	DepTime	DepDelay	Origin	ArrTime	ArrDelay	Dest	Delayed
<chr>	<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<chr>	<chr>
N632	825	10	BWI	917	7	BNA	1
N632	1330	0	BNA	1646	36	BWI	1

On the first trip from the flight departs and arrives late. On the second trip the flight departs on time and arrives late. Both trips are delayed but we cannot conclude that a delay in the first airport lead to a delay in the second.

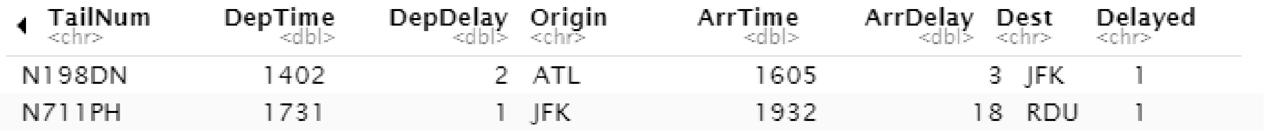
B – Another way is to look at the day that has the highest number of records in my data to investigate for cascading failures. After looking at the top 3 days in my subset I was not able to get any conclusion for this answer.

C – I first arranged the data by the sequence it which it occurs. Then I created a lagged variable for the Destination airport called Lagged. I selected data where the Previous flight's Destination airport is the same as the Origin airport on the same day. I then inspected the days that recorded this phenomenon.



TailNum	DepTime	DepDelay	Origin	ArrTime	ArrDelay	Dest	Delayed
<chr>	<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<chr>	<chr>
N372DA	634	4	HLN	759	15	SLC	1
N561SW	707	0	SBA	754	-5	LAX	0
N821AS	945	0	SAT	1052	2	DFW	1
N610AA	1447	101	DFW	1536	98	LAS	1
N271AA	1617	-2	LAS	2103	8	DFW	1

On 7th December 2003, we can see, a flight arrives late at DFW airport at 10:52pm. Then a flight departs 101 mins late from DFW airport at 14:47pm arrives late at LAS airport so it is also delayed. The next flight takes off early but lands 8 mins late at DFW. Hence all three trips are delayed.



TailNum	DepTime	DepDelay	Origin	ArrTime	ArrDelay	Dest	Delayed
<chr>	<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<chr>	<chr>
N198DN	1402	2	ATL	1605	3	JFK	1
N711PH	1731	1	JFK	1932	18	RDU	1

On 21st March a flight arriving at 16:05 at JFK is delayed as well as the flight taking off at 17:31 from JFK is delayed. This flight is delayed at arrival in the next airport for 18 mins. Here too both flights are delayed.

The above 3 methods can be used to detect cascading failures. All the above examples show cascading delays but it is too little evidence to prove it for the entire population.

Question 5 – Use the available variables to construct a model that predicts delays.

Features used -

Categorical - Year, Month, DayofMonth, DayOfWeek, DepHour, Cancelled, Diverted

Numeric - DepDelay, Distance, TaxiIn , TaxiOut, AirTime, ActualElapsedTime, SecurityDelay , NASDelay, WeatherDelay, CarrierDelay, LateAircraftDelay

I used 4 classification models

- Logistic Regression
- Gradient Boosting
- Classification Trees
- Support Vector Machines

Benchmarking visualisations

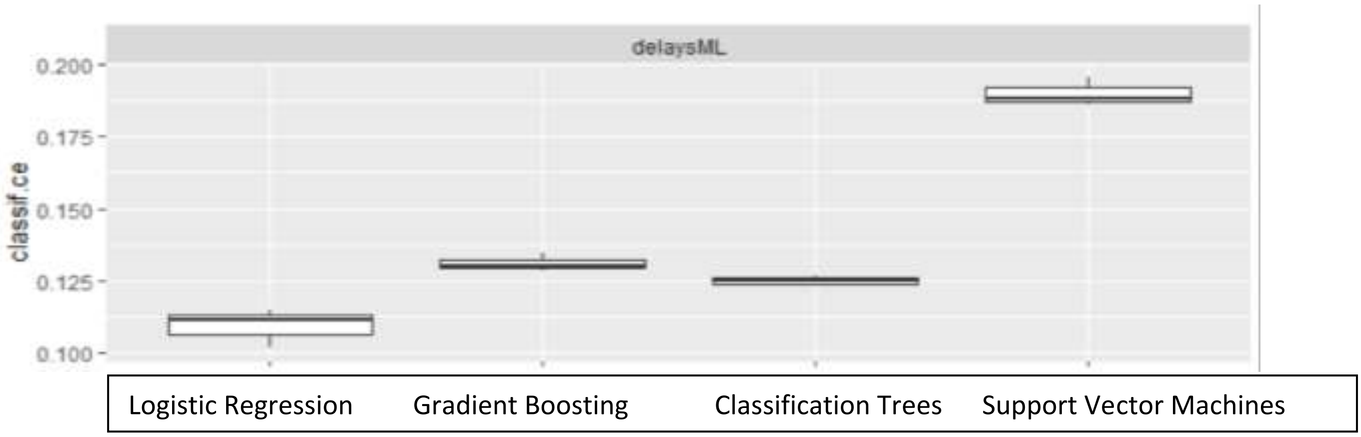
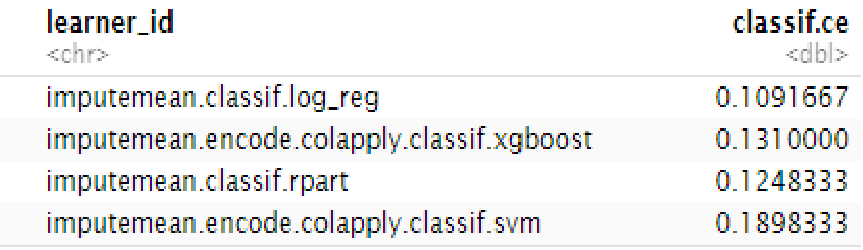


Table with all the Classification Error by models.



	learner_id <chr>	classif.ce <dbl>
Logistic Regression	imputemean.classif.log_reg	0.1091667
Gradient Boosting	imputemean.encode.colapply.classif.xgboost	0.1310000
Classification Trees	imputemean.classif.rpart	0.1248333
Support Vector Machines	imputemean.encode.colapply.classif.svm	0.1898333

The model with the lowest classification error is Logistic Regression with an error of 0.109 . Hence, the Logistic Regression model is the best.