IMPACT OF WEATHER ON FLIGHT CANCELLATION

Done by:

Charan Ravishankaran (s2331942)
Shyamprasad Sugumaran (s2289741)
Jayanth Chinthu Rukmani Kumar (s2216248)
Ravi Teja Baligudam (s2352532)
Group 7

ABSTRACT

In our lifetime, we all would have been in a situation where we were stuck in the airport due to flight delays or cancellation. Is there a way to avoid such situations? In this paper, we have focused on flight cancellations. The flights can be canceled due to aircraft issues, security issues or due to weather. One can't derive a pattern from the aircraft issue or security issue. Whereas in case of weather, we can derive insights from the flight records and avoid traveling in such scenarios. In this paper, the focus will be on weather cancellations and also determine which month is being affected the most and the least, which route is affected the most and the least and which airline is affected the most and the least. By finding these statistics, people can adapt their schedules based on that and this might also help every airline so that they can reschedule their flight such that it isn't affected by the weather.

INTRODUCTION

Flight cancellations are getting common nowadays throughout the world. Due to these issues, many passengers around the world are getting affected. There are so many reasons for these cancellations. Some of the most common reasons include weather conditions, aircraft issues, security issues, etc.

In this experiment, the cancellations due to weather will be analyzed. The research questions are structured based on the same. In particular, the primary research question is to find how weather affects the performance of airlines in the U.S. The secondary focus will be to find which month and airline will be affected the most and the least. With such statistics, airlines can be more careful about planning the time and day for the specific months or the passengers can plan accordingly. This experiment may act as a solution for many airlines on when to schedule the journey.

For every flight to reach its destination, flight routes are of prime importance. So, in this paper, the route which is affected the least and which is affected the most will also be analyzed. Thus, the airline can route it in such a way that the flight can travel without being affected by the weather.

The paper is structured as follows: Related work will be discussed in the next section; Section 3 will address the data and method which has been followed for this paper. Moving on to the next section, the results will be discussed in detail. Finally, conclusion and relevant future work will be provided.

BACKGROUND

Imagine if you are in an airport and unfortunately, the flight which you are going to catch has been canceled. The reason might be due to unpredictable weather conditions. This problem has not only been happening for you but also it is a recurring problem around the world. It would be even worse if you are traveling by flight during the winter or rainy season. Statistics [5] have shown that most of the flights have been canceled during these seasons. The advantage of this project is that it not only focuses on when? But also, there is a more in-depth analysis of which route and which airline in this project.

In the projects which have been done already, machine learning and deep learning approaches have been used to predict flight delays or cancellations. But minimal research has been done in the field of big data when it comes to predicting flight delays or cancellations. Furthermore, there

are some papers which provide some information on the flights which are canceled due to security issues. But, there is less information available for the statistics of the months affected, routes affected and the airlines affected due to weather.
So, the main aim of this project is to predict flight cancellation which occurs due to weather conditions using the big data approach of Map-reduce and also SQL functions in the data frame. Map-reduce is used for processing big data sets in parallel.

RELATED WORK

For finding relevant research papers, the search was employed in Scopus by using the keywords "Flight cancellations" AND "Weather" AND “Big Data” OR "Machine learning" OR "Neural Networks" OR "Deep Learning".
Belcastro et al. used data mining techniques for predicting flight delays [1]. They implemented a “Predictor” for predicting the arrival delay of a flight that is scheduled due to weather conditions. Their observations were done using Mapreduce programs in the cloud. They found that using these techniques, the model achieved high accuracy. They also evaluated the accuracy of their model for a given time threshold.
Varsha et al. proposed machine learning and deep learning approaches for predicting the delay of the aviation industry [2]. Their dataset was taken from kaggle. They used machine learning techniques to predict delays. They used neural network techniques like artificial neural nets to estimate the delay of the flight. They found out that using deep neural networks 77% of accuracy as achieved and using neural networks, 89% was achieved.
Nazmus et al. comprehensively described a few ways for improving the prediction of flight delays [3]. For achieving this, they used deep learning algorithms. In particular, they used deep recurrent neural networks. By employing this technique, they found that this technique gave good results and they also suggested that the methodology of data science can be employed in the aviation industry.
Sridhar et al. have used various models for predicting flight delays and cancellations which are due to weather conditions in the United States [4]. In particular, they have compared the performance of the traditional linear regression models to the neural networks. The results which they have found is that for different seasons in a country, different models can be used for predicting the flight delays. They also have concluded that neural networks perform much better than linear regression for predicting delay models and they also have found that using neural networks, there is a high correlation between model output and the metrics of the airspace.
Overall, the related work which was done did not yield the expected results, particularly in the field of big data as there was only minimal research conducted in this field. But some other papers which were found were relevant to flight cancellations but in most of the papers, machine learning, and deep learning techniques were applied.

METHODOLOGY

DATASET DESCRIPTION

The dataset describes records of the flights which operate within the United States of America. The data is taken from October 1987 until April 2008 which consists of 120 million records. However, for this experiment, the main goal is to determine whether the flight has been canceled

due to weather conditions for which the cancellation code is important. But in our analysis, we found that for the duration January 1987 to May 2003 the cancellation code column is 'NA'. Therefore, we have loaded the data from January 2000 to April 2008 of the dataset and filtered out the data for which the cancellation code is not 'NA' (i.e.) June 2003 to April 2008 has been considered.

In addition to the main dataset, there are two other sub-dataset which are 'airports' and 'carriers'. The airports dataset consists of airport name, city, state, country, latitude and longitude for the respective origin and destination code of the main dataset. The carriers dataset consists of Airlines name for the respective carrier code of the main dataset.

MAP REDUCE

MapReduce is a processing method [6]. They are two different tasks: Map and Reduce. The main job of the Map is to take a set of data and transform it into a different set where each element are split into key and value pairs. The task of the reduce is to take the output from the map which will act as the input and will combine all those key and value pairs into an, even more, smaller set of pairs of key and value. The map and reduce algorithm is given below:

1. MapReduce executes in three different stages: the map stage, the shuffle stage and the reduce stage.
2. The map will process the input and the input will be stored in the Hadoop distributed file system (HDFS). The input will be passed to the map function line by line and it will create several chunks of data.
3. This stage is a combination of shuffle and reduce stage. Reduce will process the data which is given by Map. After processing, it will produce a newer set of outputs that will be stored in HDFS.
4. When the MapReduce job is being done, Hadoop will assign tasks to each server in the cluster.
5. The computing job will take place on nodes with data on the corresponding local disk which will reduce the network traffic.
6. Finally, after completion of the entire, the cluster will collect and reduce the data to give a result and will send it back to the server of Hadoop.

METHOD

In our research, the main aim is to find if weather conditions affect airline performance in the United States. In the dataset, if a particular flight is being canceled, then it is represented as 1 and if it is not, it is specified by 0. Initially, the dataset is filtered out where the cancellation is '1' which will provide the dataset of flights being canceled. There is a column named as "Cancellation code" which gives the reason why the flight has been canceled. There are 4 reasons which are, Carrier which is denoted as 'A', Weather which is denoted as 'B', NAS which is denoted as 'C' and Security which is denoted as 'D'. The dataset is again filtered out where the cancellation code is 'B' which will provide the dataset of the flight being canceled due to weather.

FLIGHT CANCELLATION RATE

This is derived by determining the count of the dataset which has been filtered out by assigning Canceled as '1', Cancellation code as 'B' and the year as its respective year (For the year 2003, additionally, month conditions should be mentioned in order to not consider the first 5 months which have cancellation code as 'NA').

Then further specific analysis is done from the derived dataset, They are:
1. Month wise analysis
2. Airline wise analysis
3. Route wise analysis

In order to analysis overall records, a separate data frame is created which contains the flights canceled due to weather from June 2003 to April 2008.

MONTH WISE ANALYSIS

Initially from the dataset which contains the flights canceled due to weather, the month column is derived and flattened using flatmap. The next step is to convert each month into their key and value pair where key denotes the month and value denotes the number of occurrences of each month which is done using the Map function. Using Reducebykey, the months obtained in the previous step will be aggregated based on their number of occurrences. The final step will be to find the months which have the most impact and to find the months which had the least impact which is done using sorting. This method is being carried out on both year wise and overall using respective datasets.

AIRLINE WISE ANALYSIS

From the dataset containing flights canceled due to weather, the unique carrier column will be considered to find the most and least affected airlines. This column contains the airline codes for each airline. By using SQL commands like Groupby and aggregate, the airline code is grouped and aggregated with respect to its number of occurrences. Since the count is large, in order to convert it to a percentage, again groupby is used to determine how many times each carrier has occurred in the dataset containing flights canceled due to weather. Now, the Join function is used to combine both tables of counts.
The percentage for each airline is calculated by dividing the count of airline canceled due to weather to the count of airlines canceled in total and multiplying it to 100. Percentage is calculated in order to ignore large counts.
The resulting data frame consists of the unique carrier code, its respective counts and the cancellation percentage. In order to know which airline does the unique carrier code signifies, there is a separate dataset available (carriers.csv) which contains the details of the airline name and its respective unique carrier code. Here again, we use join function to combine the datasets and derive the respective airline names.
The final data frame is sorted in descending order with respect to the percentage in order to determine the top 5 most affected airlines and in ascending order to determine the least affected

airline. Drop function is used to delete the columns which are not required. This method is being carried out on both year wise and overall using respective datasets.

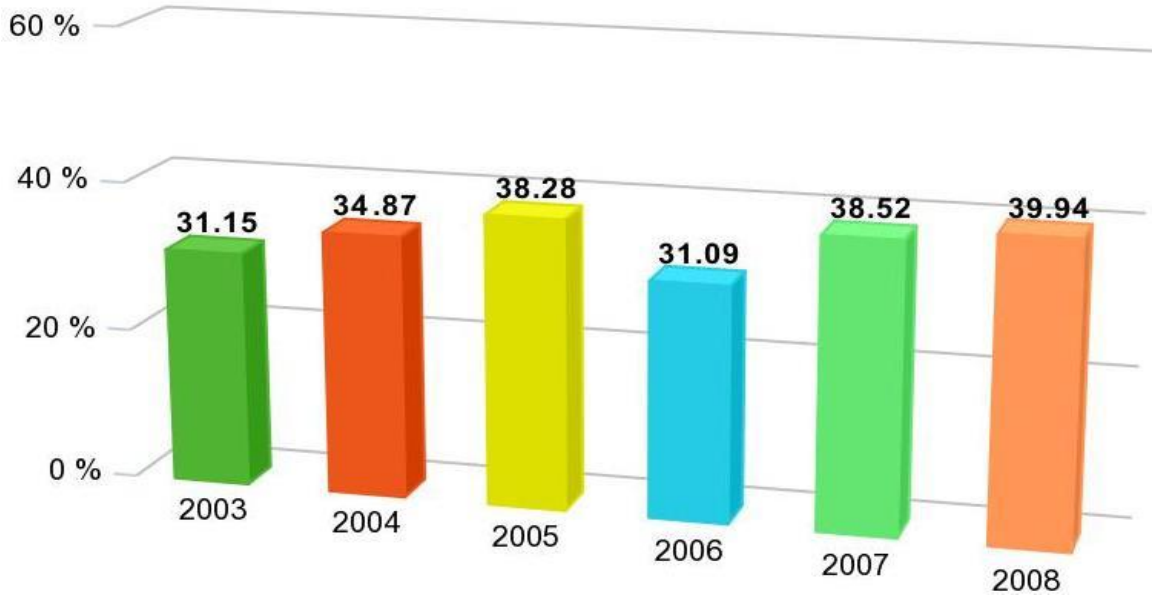
ROUTE WISE ANALYSIS

From the dataset containing flights canceled due to weather, the origin column and destination column will be considered to find the most and least affected routes. This column contains the city codes for each city. By using SQL commands like Groupby and aggregate, the city code of origin and destination are grouped and aggregated with respect to its number of occurrences of the combination.
The resulting data frame is sorted in descending order with respect to the count of occurrences in order to determine the top 5 most affected routes.
The resulting data frame consists of the origin code, destination code, and its respective counts. In order to know which city does the origin code and destination code signifies, there is a separate dataset available (airports.csv) which contains the details of city, state, country, latitude, longitude and its respective city code. Here, we use join function to combine the datasets and derive the respective city names.
Since the least affected route are in large number, instead of determining the routes, we determined the number of routes that are occurring the least using the count function. This method is being carried out on both year wise and overall using respective datasets.

RESULTS

FLIGHT CANCELLATION RATE

YEAR	2003	2004	2005	2006	2007	2008
TOTAL NO OF FLIGHT TRAVEL	3815798	7129270	7140596	7141922	7453215	2389217
CANCELLED FLIGHTS	52926	127757	133730	121934	160748	64442
CANCELLED DUE TO WEATHER	16486	44558	51204	37913	61935	25744

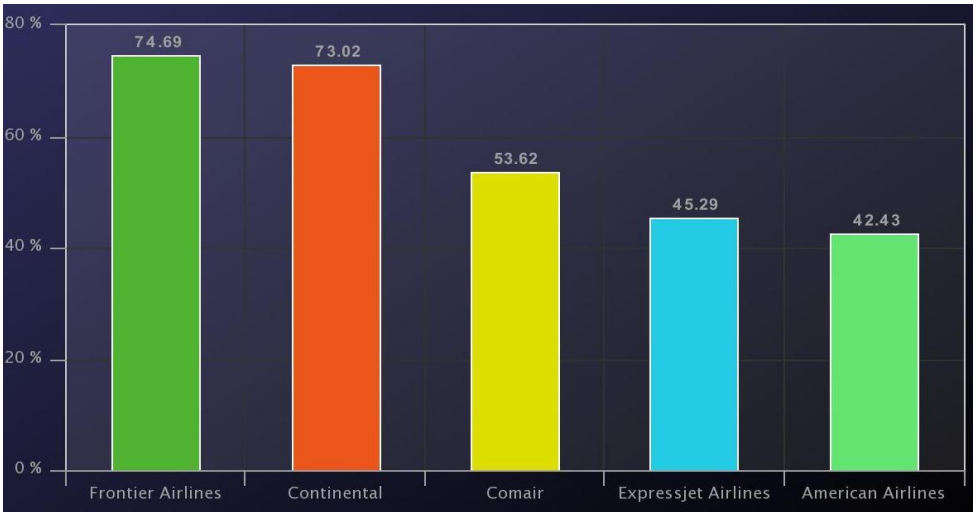


Thus, the cancellation rate occurred year wise from June 2003 to April 2008. The results show there are consistently 30%-40% flights getting canceled due to weather. Out of the 4 cancellation reasons, the weather is one-third of the reason why flights get canceled, which shows the consistent impact of weather on flight cancellations.

MONTH WISE ANALYSIS

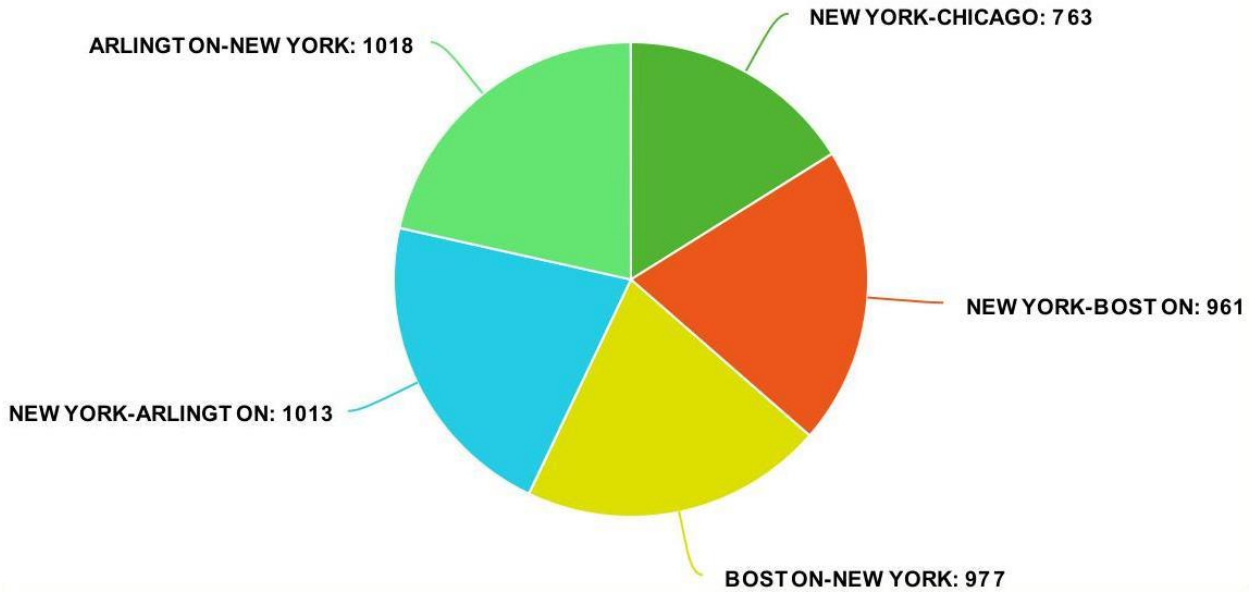
The graph which is displayed above show the flights cancellation rate for every month from June 2003 to April 2008. It can be clearly seen from the graph that during the month of January, most cancellations were made. The next 2 most affected months were December and February. This can be attributed to the fact that it is winter season during December, January and February in the U.S. During the month of May, least cancellations were made. April is the second least affected month. This can be attributed to the fact that since it is the spring season during April and May in the U.S. The month-wise analysis for each year can be found in the Appendix.

AIRLINE WISE ANALYSIS



The graph which is displayed above shows the most canceled airline due to weather from June 2004 to April 2008. From the graph, it can be observed that Frontier Airlines and Continental Airlines has its cancellation reason as the weather for almost 75% of its total flight cancellations which shows the massive impact of weather in their airline. Comair Airlines and ExpressJet Airlines also have weather cancellation rates around 50%. One of the famous airlines, American Airlines also remains as 5th most affected airline with 42%.
The least affected airline is Hawaiian Airlines which had a cancellation rate of 6.88%. The airline wise analysis for each year can be found in the Appendix.

ROUTE WISE ANALYSIS



The graph which is displayed above shows the most canceled routes due to weather from June 2004 to April 2008. From the graph, it can be observed that Arlington to New York and New York to Arlington are the 2 most-affected routes. Boston to New York to and fro remains as third and fourth most affected routes respectively. Even the fifth most affected route involves New York, which may be due to the reason the New York is one of the busiest airway routes in the U.S.

There was 374 least affected route which had a cancellation count of 1. The route wise analysis for each year can be found in the Appendix.

DISCUSSIONS AND FUTURE WORK

There were interesting patterns when the monthly analysis was done. During the month-wise analysis for each year, it was noticed the September was the most affected month in 2004, whereas the cancellation rate for September was average for other years and it was also the least affected month for the year 2007. This could be due to Hurricane Ivan which affected the United States in September 2004 and this might have had an impact on flight cancellations [10]. In contrast, February in 2004 was among the months which was affected the least but in 2008 it was the most affected month.

Moving forward, for the airlines to achieve better performance rate, they can make use of the distance metric i.e. for example, if one airport has been affected due to weather conditions, instead of canceling the flight, the airlines using distance metric can redirect the flight to the airport which is nearer to the destination. This will not only improve the performance of airline but will make sure in the future if there are ways to improve the performance of the airline. Distance column from the dataset may come into consideration for achieving this scenario.

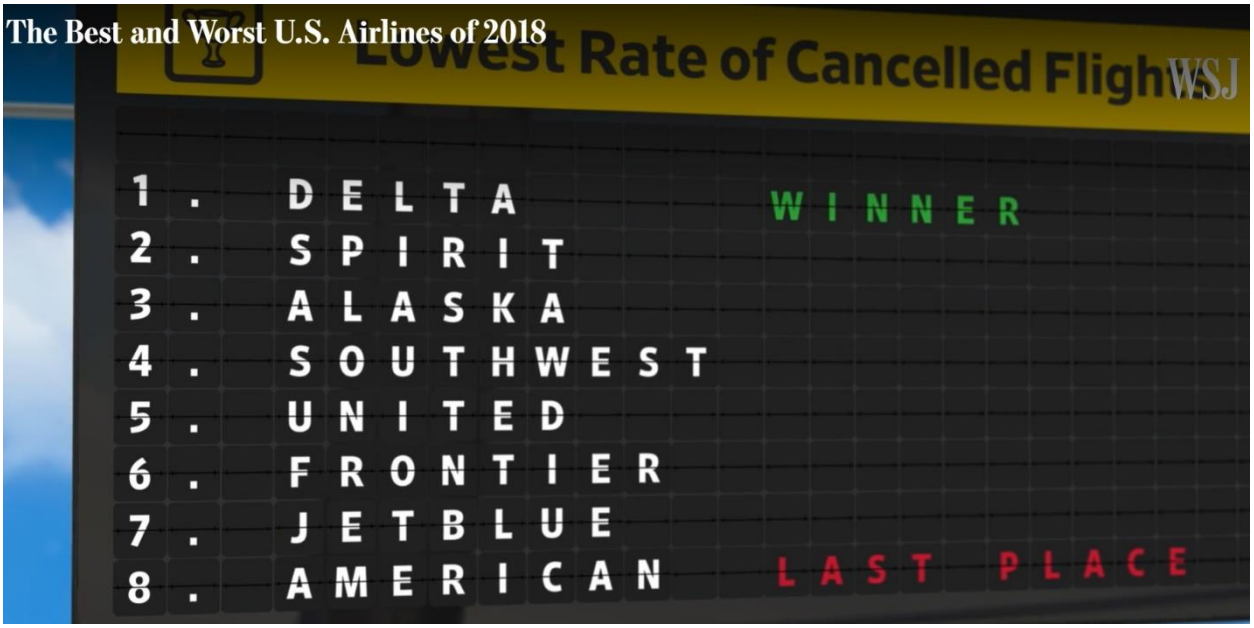
EVALUATION

The evaluation was done for finding the airlines that were affected the most. As you can observe from the image above, American Airlines is the second most affected and Frontier Airlines is the tenth most affected airline in 2013 published under “Worst airlines by cancelled flights” [8] which were present in the most affected airline of our analysis.

Here is the rest of the 10 worst U.S. airports for flight delays or cancellations. Percentages have been rounded up to the nearest 10th decimal:

Ranking	Departure airport	On-time performance
1	Newark Liberty International Airport (EWR)	64%
2	Chicago O'Hare International Airport (ORD)	65%
3	LaGuardia Airport (LGA)	66%
4	Denver International Airport (DEN)	66%
5	Dallas/Fort Worth International Airport (DFW)	68.5%
6	George Bush Intercontinental Airport (IAH)	71%
7	Charlotte Douglas International Airport (CLT)	73%
8	John F. Kennedy International Airport (JFK)	74%
9	Hartsfield-Jackson Atlanta International Airport (ATL)	77%
10	Los Angeles International Airport (LAX)	77.5%

The evaluation was also done to find the airport which was affected the most (corresponding to the route). From the statistics shown above [9], New York is the worst affected airport and it occurs thrice (Newark Liberty International Airport, LaGuardia Airport and John F. Kennedy International Airport). Furthermore, Chicago O'Hare International Airport comes second in the statistics. New York and Chicago were among the most affected route from our route wise analysis.



American Airlines and Frontier Airlines have the highest rate of Cancelled flights in 2018 as per the article [11]. Both the airlines were in the top 5 most affected airlines of our airline wise analysis.

The articles provided for the evaluation are not in the year of our analysis but still few results have been matched in the further years which shows there may be consistency in the flights getting canceled.

CONCLUSION

In this project, we have applied the Mapreduce algorithm and SQL functions in the data frame to find the months, routes, airlines which were affected the most and least due to weather. These analyses will benefit the airlines so that they may reschedule their flight timings or route and also will aid people so that they can plan their journey according to it. The results which have been shown to have some interesting trends particularly when analysis for months have been considered and that has already been discussed in the discussion section. When performing the evaluation, it was found that the results which were done in the research did match in the forthcoming years.

REFERENCES

[1] Belcastro, L., Marozzo, F., Talia, D.,Trunfio, P. (2016) Using Scalable Data Mining for Predicting Flight Delays. ACM Transactions on Intelligent Systems and Technology

[2] Venkatesh, V., Arya, A., Agarwal, A., Lakshmi, S. (2017) Iterative machine and deep learning approach for aviation delay prediction. 2017 4th IEEE Uttar Pradesh Section International conference on Electrical, Computer and Electronics (UPCON)

[3] Nazmus, S., Moniruzzaman, Md. (2019) Enhancing Airlines Delay Prediction by Implementing Classification Based Deep Learning Algorithms. Advances in Intelligent Systems and Computing book series (AISC, volume 935)

[4] Sridhar, B., Wang, Y., Klein, A., Jehlen, R.(2009) Modeling Flight Delays and Cancellations at the National, Regional and Airport Levels in the United States. Eighth USA/Europe Air Traffic Management Research and Development Seminar (ATM2009)

[5] https://www.sciencedirect.com/science/article/pii/S2212012218300753/

[6]https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

[7]http://stat-computing.org/dataexpo/2009/

[8]https://www.bestchoicereviews.org/airlines

[9]https://www.marketwatch.com/story/this-is-the-worst-airport-in-the-us-for-delayed-flights-2019-09-05

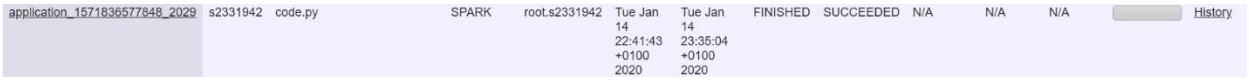
[10]https://en.m.wikipedia.org/wiki/Hurricane_Ivan

[11] https://www.wsj.com/articles/the-best-and-worst-u-s-airlines-of-2018-11547648032

APPENDIX:

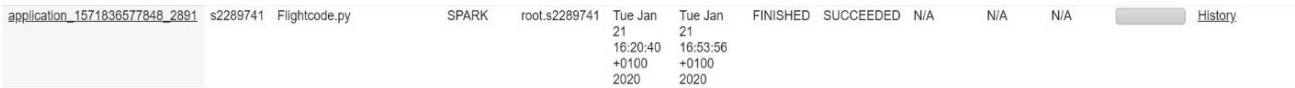
RUNTIME:

Initial run: 53 minutes 21 seconds



application_1571836577848_2029	s2331942	code.py	SPARK	root.s2331942	Tue Jan 14 22:41:43 +0100 2020	Tue Jan 14 23:35:04 +0100 2020	FINISHED	SUCCEEDED	N/A	N/A	N/A	History

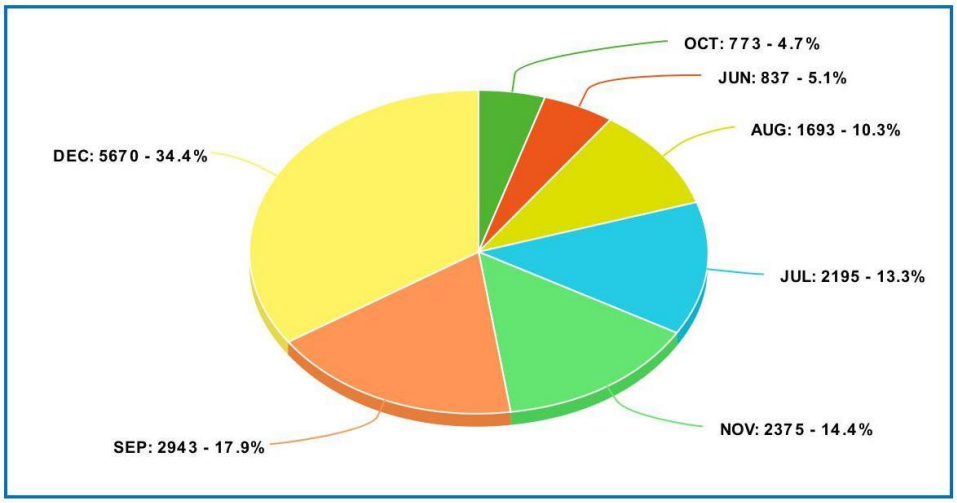
Final run: 33 minutes 16 seconds



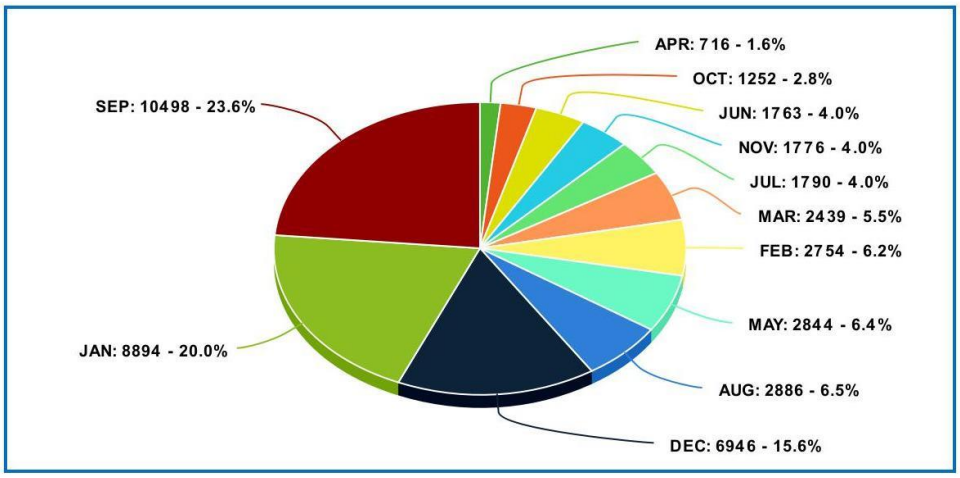
application_1571836577848_2891	s2289741	Flightcode.py	SPARK	root.s2289741	Tue Jan 21 16:20:40 +0100 2020	Tue Jan 21 16:53:56 +0100 2020	FINISHED	SUCCEEDED	N/A	N/A	N/A	History

MONTH WISE ANALYSIS: EACH YEAR

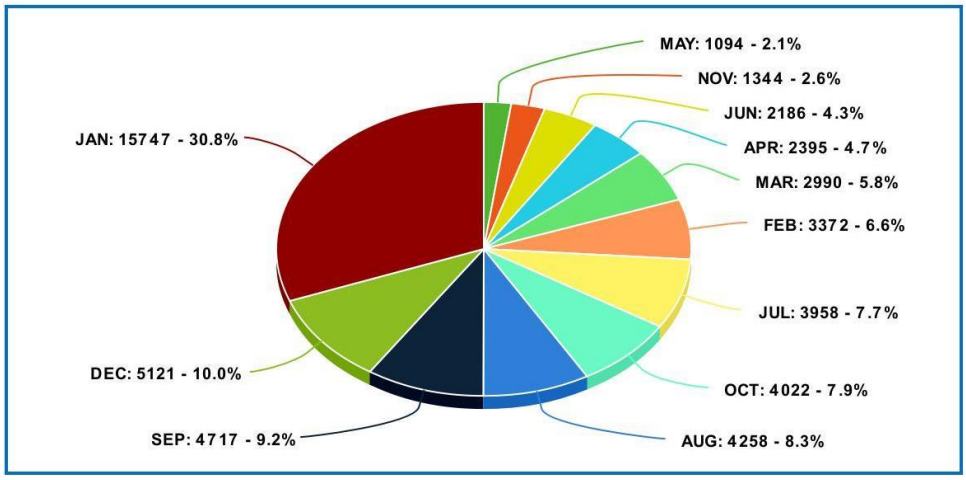
MONTH WISE ANALYSIS : 2003



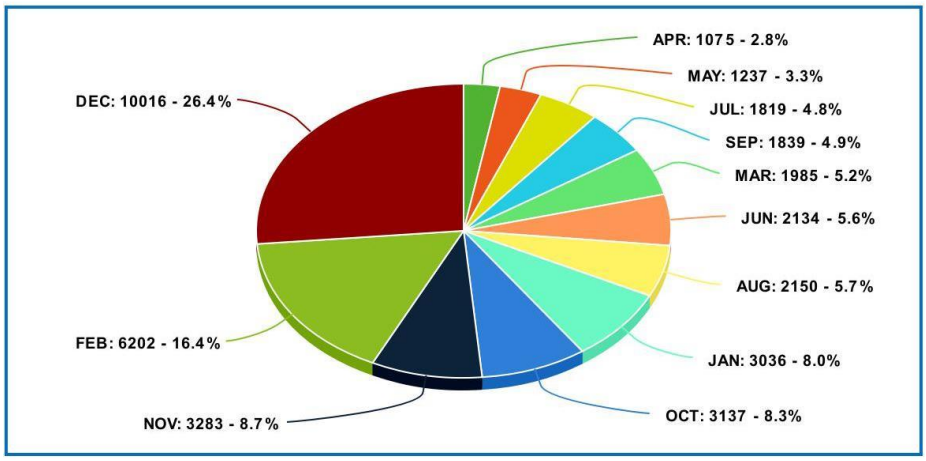
MONTH WISE ANALYSIS : 2004



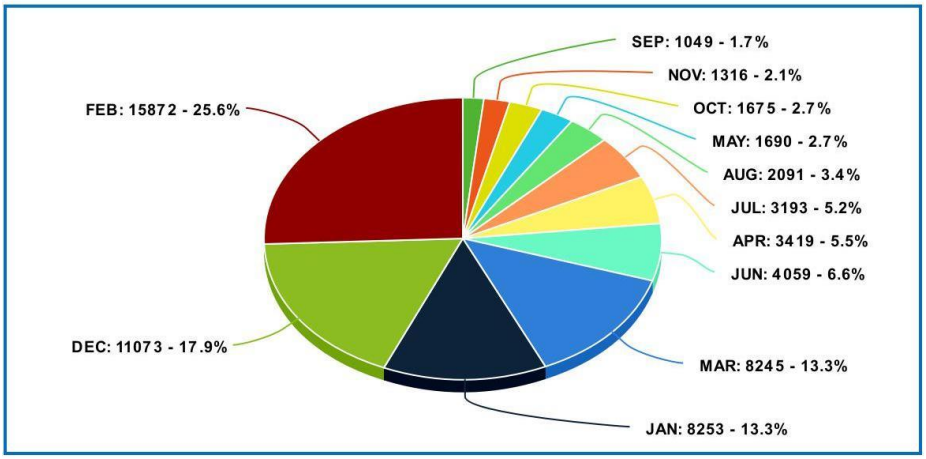
MONTH WISE ANALYSIS : 2005



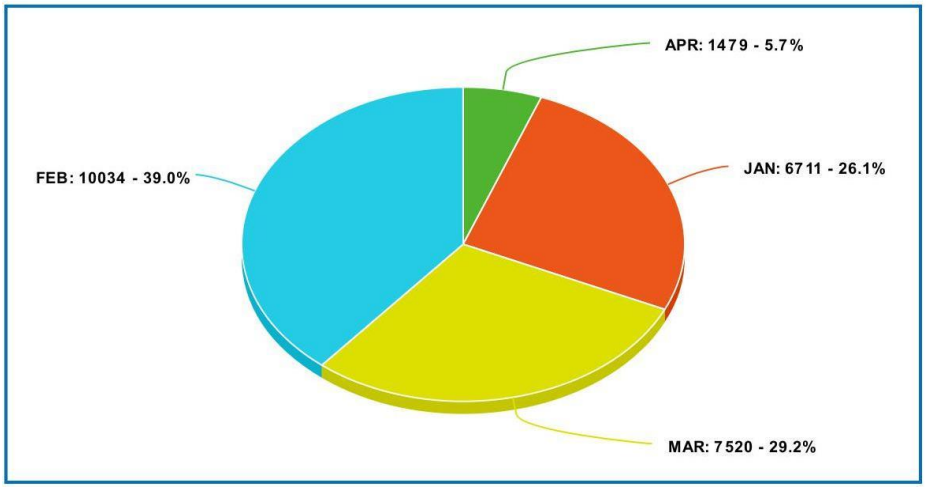
MONTH WISE ANALYSIS : 2006



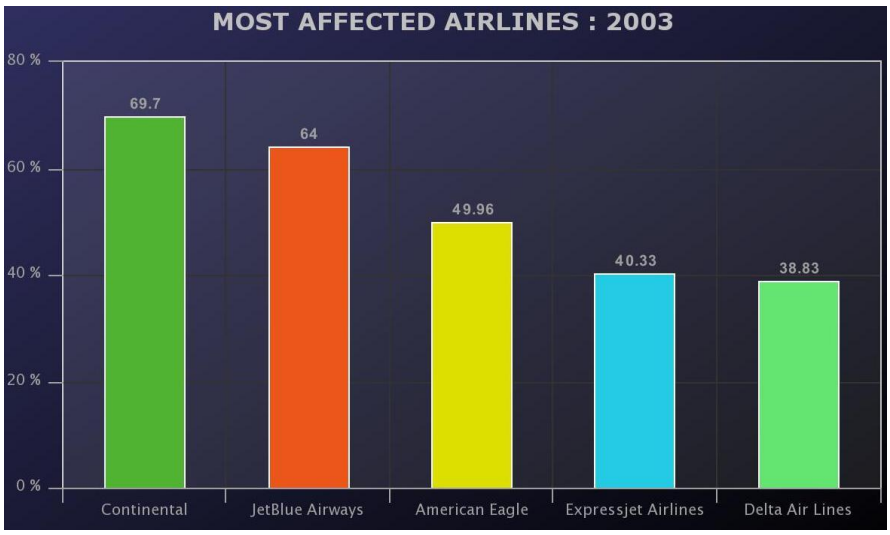
MONTH WISE ANALYSIS : 2007



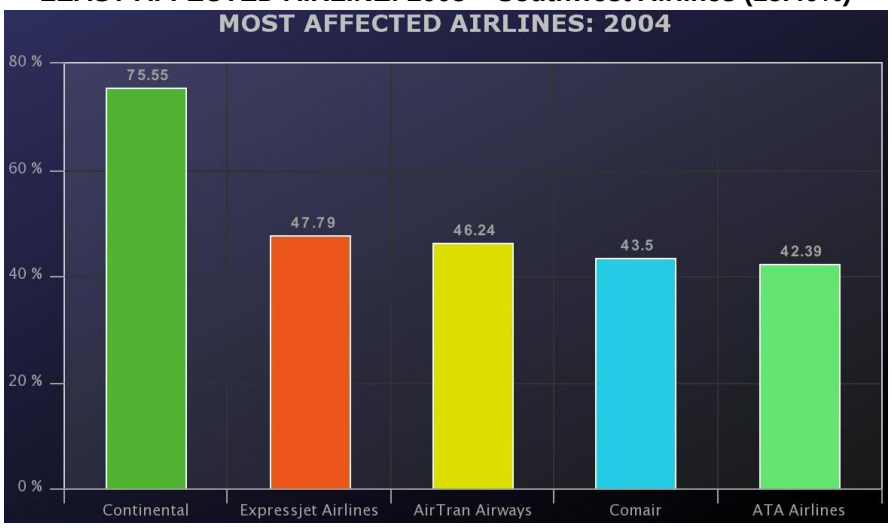
MONTH WISE ANALYSIS : 2008



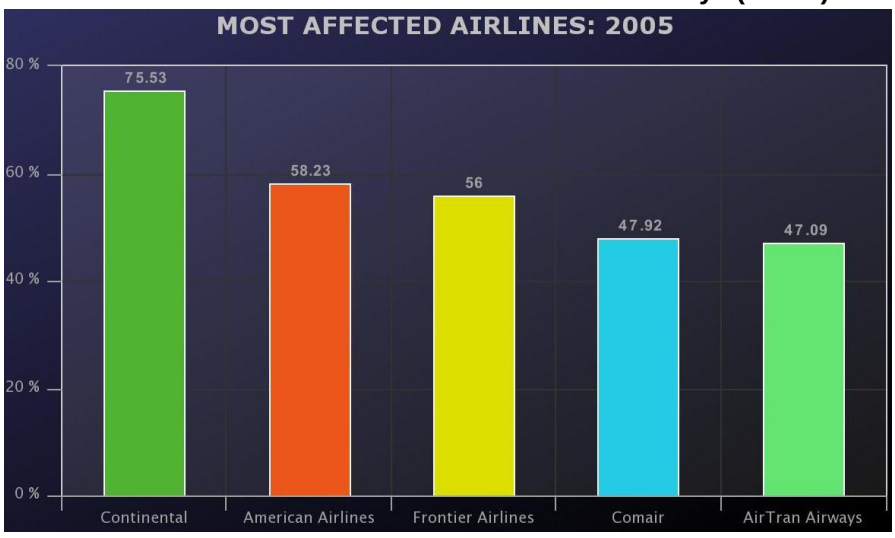
AIRLINE WISE ANALYSIS: EACH YEAR



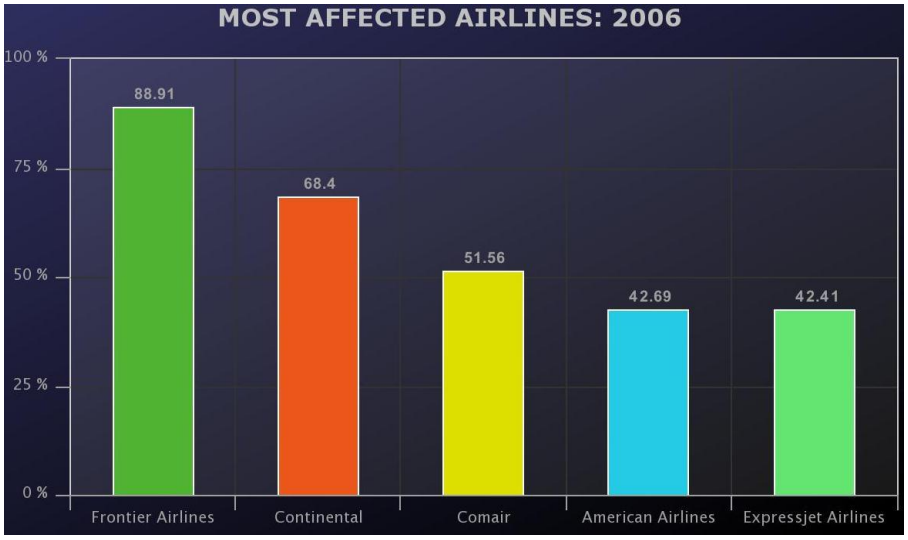
LEAST AFFECTED AIRLINE: 2003 – Southwest Airlines (15.40%)



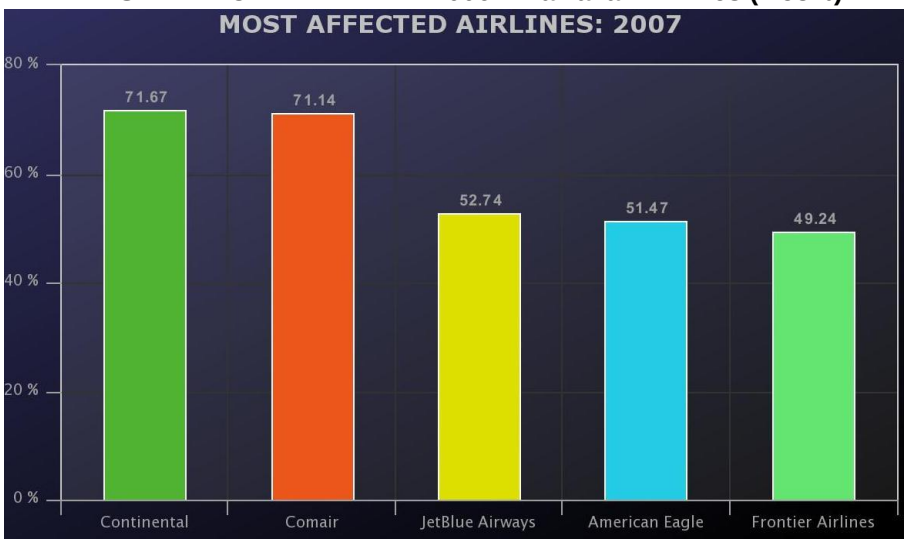
LEAST AFFECTED AIRLINE: 2004 – JetBlue Airways (5.67%)



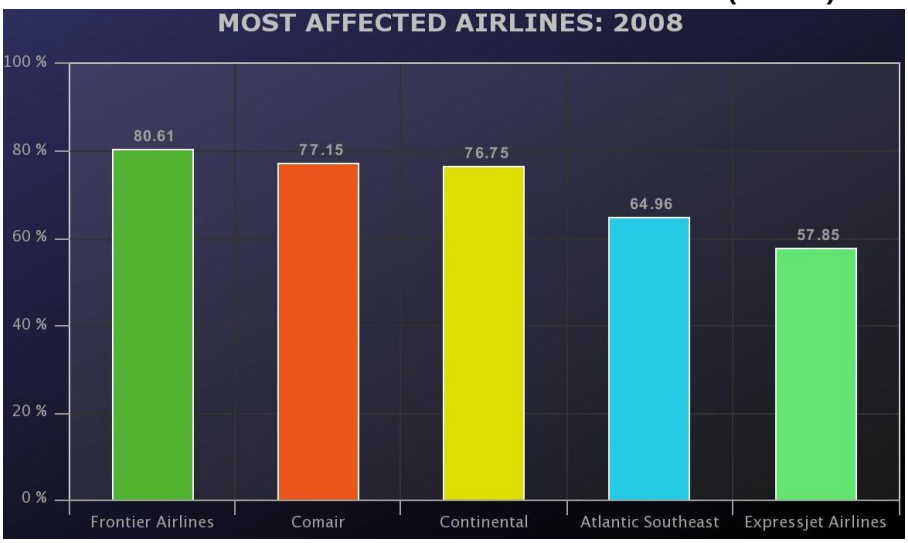
LEAST AFFECTED AIRLINE: 2005 – Hawaiian Airlines (7.24%)



LEAST AFFECTED AIRLINE: 2006 – Hawaiian Airlines (1.58%)



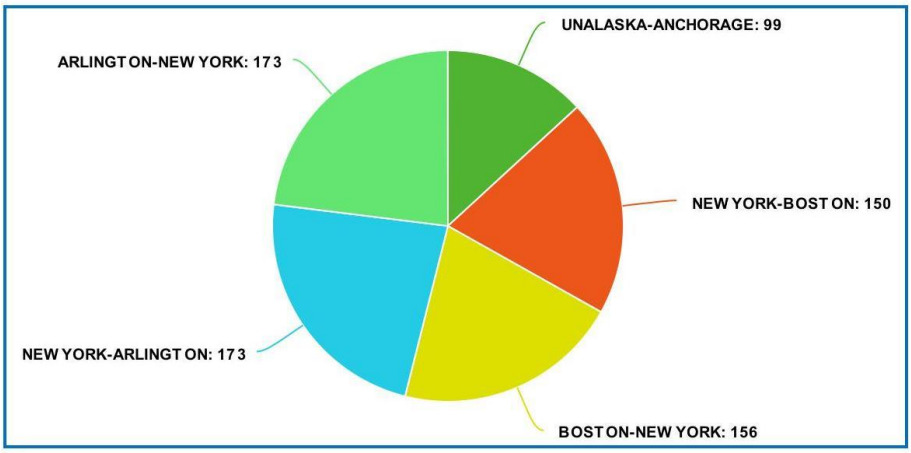
LEAST AFFECTED AIRLINE: 2007 – Alaska Airlines (13.88%)



LEAST AFFECTED AIRLINE: 2008 – Mesa Airlines (19.65%)

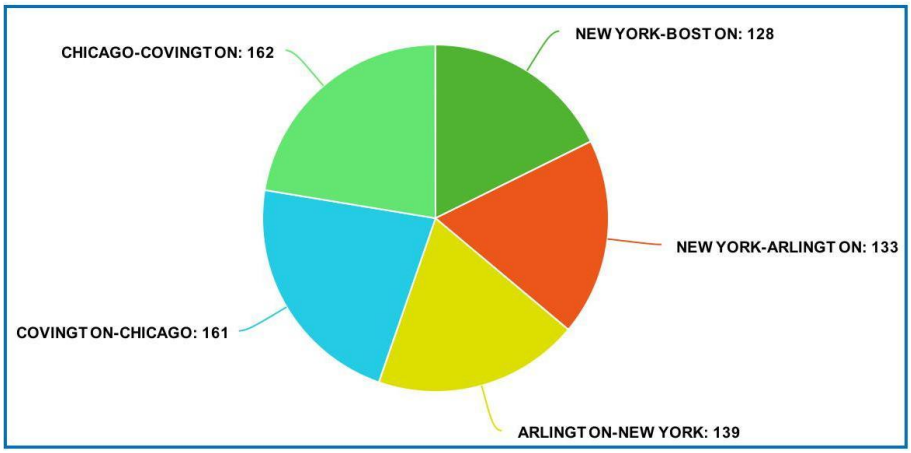
ROUTE WISE ANALYSIS: EACH YEAR

MOST AFFECTED ROUTES : 2003



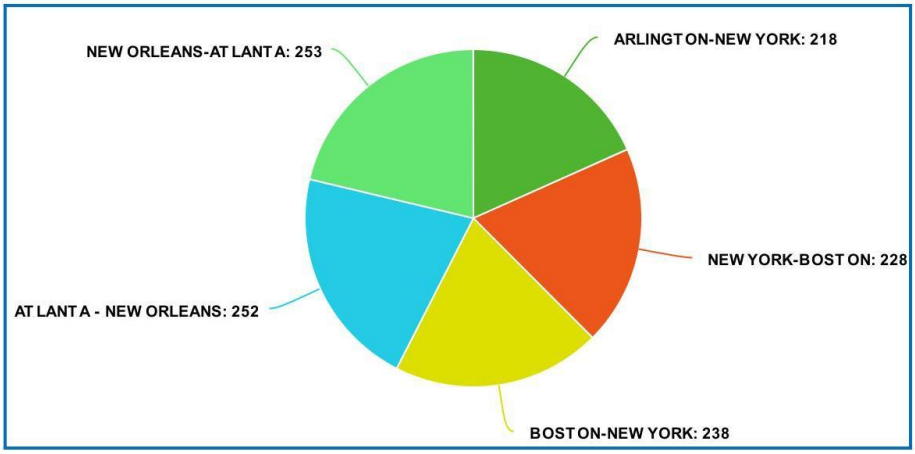
LEAST AFFECTED ROUTE: 2003 – 528 ROUTES CANCELLED ONLY ONCE

MOST AFFECTED ROUTES : 2004



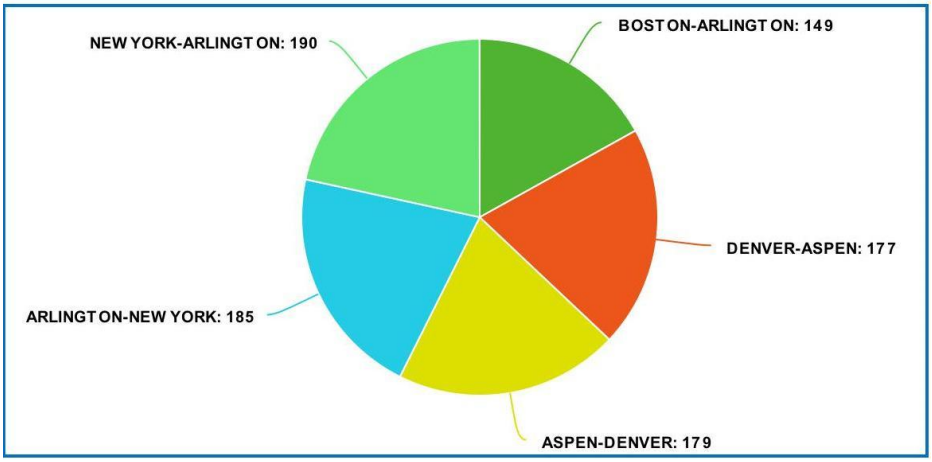
LEAST AFFECTED ROUTE: 2004 – 410 ROUTES CANCELLED ONLY ONCE

MOST AFFECTED ROUTES : 2005



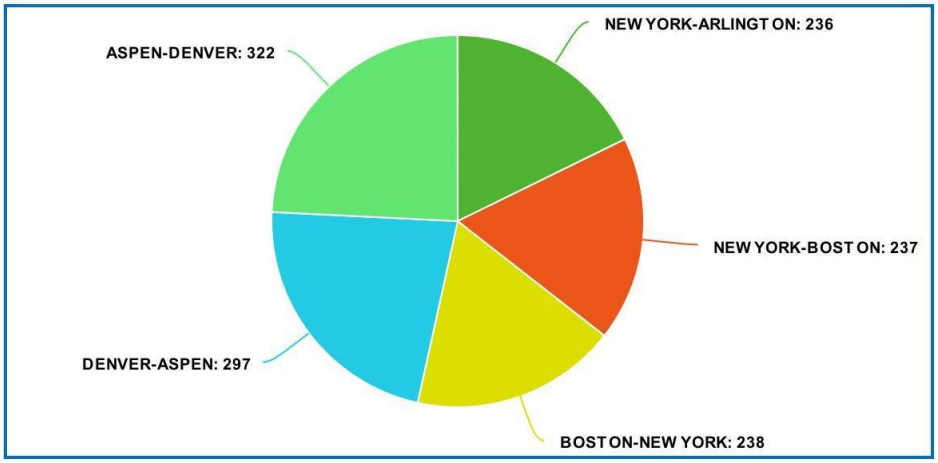
LEAST AFFECTED ROUTE: 2005 – 418 ROUTES CANCELLED ONLY ONCE

MOST AFFECTED ROUTES : 2006



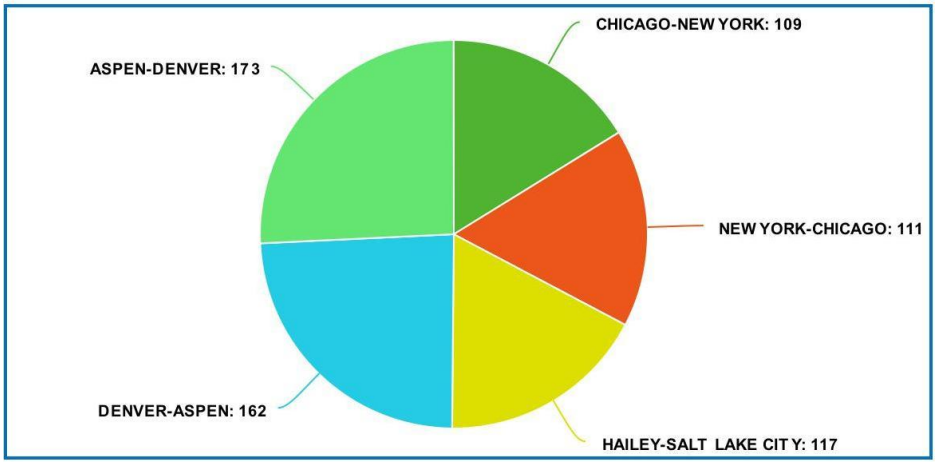
LEAST AFFECTED ROUTE: 2006 – 561 ROUTES CANCELLED ONLY ONCE

MOST AFFECTED ROUTES : 2007



LEAST AFFECTED ROUTE: 2007 – 563 ROUTES CANCELLED ONLY ONCE

MOST AFFECTED ROUTES : 2008



LEAST AFFECTED ROUTE: 2008 – 684 ROUTES CANCELLED ONLY ONCE

```
#           PYTHON SPARK CODE           #

#Importing libraries
from pyspark import SparkContext
from pyspark.sql import SparkSession
import pyspark.sql.functions as f

spark = SparkSession.builder.getOrCreate()
sc = SparkContext.getOrCreate()
sc.setLogLevel("ERROR")

#Loading main dataset and the 2 additional datasets
#2000.csv to 2008.csv are available in /user/s2289741/projectdata
#carriers.csv and airports.csv available in respective folders in /user/s2331942
df = spark.read.format("csv").option("header","true")
    .load("/user/s2289741/projectdata/*.csv")
df_airports = spark.read.format("csv").option("header","true")
    .load("/user/s2331942/airport/airports.csv")
dfairport_a = df_airports.select("iata","city")
df_carriers = spark.read.format("csv").option("header","true")
    .load("/user/s2331942/carrier/carriers.csv")
#total_records = df.count()

#To determine which year's dataset have cancellation code
checkYears = df.filter((df.CancellationCode.isNotNull()) & (df.CancellationCode !="NA"))
get = checkYears.select(checkYears.Year)
get.dropDuplicates().show()
'''
+----+
|Year|
+----+
|2005|
|2006|
|2004|
|2008|
|2007|
|2003|
+----+
'''

# To determine from which month the cancellation code is filled
# Because 2003 has lesser cancellation compared to other years.
checkMonths = df.filter((df.CancellationCode.isNotNull())
    &(df.CancellationCode !="NA") & (df.Year == 2003))
getm = checkMonths.select(checkMonths.Year, checkMonths.Month)
getm.dropDuplicates().show()
'''
+----+-----+
|Year|Month|
+----+-----+
|2003|    9|
|2003|    6|
|2003|   10|
|2003|    8|
|2003|   12|
|2003|    7|
|2003|   11|
+----+-----+
'''

#Splitting the dataset year wise
df2003 = df[(df.Year == 2003) & (df.Month != 1) & (df.Month != 2)
        & (df.Month != 3) & (df.Month != 4) & (df.Month != 5)]
#records_2003 = df2003.count()
#print(records_2003)
df2004 = df[(df.Year == 2004)]
#records_2004 = df2004.count()
```

```
#print(records_2004)
df2005 = df[(df.Year == 2005)]
#records_2005 = df2005.count()
#print(records_2005)
df2006 = df[(df.Year == 2006)]
#records_2006 = df2006.count()
#print(records_2006)
df2007 = df[(df.Year == 2007)]
#records_2007 = df2007.count()
#print(records_2007)
df2008 = df[(df.Year == 2008)]
#records_2008 = df2008.count()
#print(records_2008)

#Splitting the dataset which contains cancelled flights
df2003a = df2003[(df2003.Cancelled == 1)]
cancel_2003 = df2003a.count()
#print(cancel_2003)
df2004a = df2004[(df2004.Cancelled == 1)]
cancel_2004 = df2004a.count()
#print(cancel_2004)
df2005a = df2005[(df2005.Cancelled == 1)]
cancel_2005 = df2005a.count()
#print(cancel_2005)
df2006a = df2006[(df2006.Cancelled == 1)]
cancel_2006 = df2006a.count()
#print(cancel_2006)
df2007a = df2007[(df2007.Cancelled == 1)]
cancel_2007 = df2007a.count()
#print(cancel_2007)
df2008a = df2008[(df2008.Cancelled == 1)]
cancel_2008 = df2008a.count()
#print(cancel_2008)

#Splitting the dataset which contains cancelled flights due to weather
df2003b = df2003a[(df2003a.CancellationCode == "B")]
weather_2003 = df2003b.count()
#print(weather_2003)
df2004b = df2004a[(df2004a.CancellationCode == "B")]
weather_2004 = df2004b.count()
#print(weather_2004)
df2005b = df2005a[(df2005a.CancellationCode == "B")]
weather_2005 = df2005b.count()
#print(weather_2005)
df2006b = df2006a[(df2006a.CancellationCode == "B")]
weather_2006 = df2006b.count()
#print(weather_2006)
df2007b = df2007a[(df2007a.CancellationCode == "B")]
weather_2007 = df2007b.count()
#print(weather_2007)
df2008b = df2008a[(df2008a.CancellationCode == "B")]
weather_2008 = df2008b.count()
#print(weather_2008)

#Cancelled due to weather : year wise %

weather_perc_2003 = (float(weather_2003)/float(cancel_2003))*100
print "\nCancellation rate\n2003=",weather_perc_2003
weather_perc_2004 = (float(weather_2004)/float(cancel_2004))*100
print "\n2004=",weather_perc_2004
weather_perc_2005 = (float(weather_2005)/float(cancel_2005))*100
print "\n2005=",weather_perc_2005
weather_perc_2006 = (float(weather_2006)/float(cancel_2006))*100
print "\n2006=",weather_perc_2006
weather_perc_2007 = (float(weather_2007)/float(cancel_2007))*100
print "\n2007=",weather_perc_2007
weather_perc_2008 = (float(weather_2008)/float(cancel_2008))*100
```

```
print "\n2008=",weather_perc_2008

#Month wise analysis : each year
print "\nMonth Wise Analysis\n2003\n"
df2003c = df2003b.select(df2003b.Month)
r2003 = df2003c.rdd
r2003a = r2003 \
       .flatMap(lambda x:x) \
       .map(lambda x:(x,1)) \
       .reduceByKey(lambda a,b:a+b)
month_2003 = r2003a.top(12, key=lambda t: t[1])
for (m, c) in month_2003:
    print "Month:\t", m, "\t Count:\t", c

print "\n2004\n"
df2004c = df2004b.select(df2004b.Month)
r2004 = df2004c.rdd
r2004a = r2004 \
       .flatMap(lambda x:x) \
       .map(lambda x:(x,1)) \
       .reduceByKey(lambda a,b:a+b)
month_2004 = r2004a.top(12, key=lambda t: t[1])
for (m, c) in month_2004:
    print "Month:\t", m, "\t Count:\t", c

print "\n2005\n"
df2005c = df2005b.select(df2005b.Month)
r2005 = df2005c.rdd
r2005a = r2005 \
       .flatMap(lambda x:x) \
       .map(lambda x:(x,1)) \
       .reduceByKey(lambda a,b:a+b)
month_2005 = r2005a.top(12, key=lambda t: t[1])
for (m, c) in month_2005:
    print "Month:\t", m, "\t Count:\t", c

print "\n2006\n"
df2006c = df2006b.select(df2006b.Month)
r2006 = df2006c.rdd
r2006a = r2006 \
       .flatMap(lambda x:x) \
       .map(lambda x:(x,1)) \
       .reduceByKey(lambda a,b:a+b)
month_2006 = r2006a.top(12, key=lambda t: t[1])
for (m, c) in month_2006:
    print "Month:\t", m, "\t Count:\t", c

print "\n2007\n"
df2007c = df2007b.select(df2007b.Month)
r2007 = df2007c.rdd
r2007a = r2007 \
       .flatMap(lambda x:x) \
       .map(lambda x:(x,1)) \
       .reduceByKey(lambda a,b:a+b)
month_2007 = r2007a.top(12, key=lambda t: t[1])
for (m, c) in month_2007:
    print "Month:\t", m, "\t Count:\t", c

print "\n2008\n"
df2008c = df2008b.select(df2008b.Month)
r2008 = df2008c.rdd
r2008a = r2008 \
       .flatMap(lambda x:x) \
       .map(lambda x:(x,1)) \
       .reduceByKey(lambda a,b:a+b)
month_2008 = r2008a.top(12, key=lambda t: t[1])
for (m, c) in month_2008:
```

```
    print "Month:\t", m, "\t Count:\t", c

#Dataset containing overall flights cancelled due to weather
dfowc = df[((df.Year==2008) | (df.Year==2007) | (df.Year==2006)
        | (df.Year==2005) | (df.Year==2004) | ((df.Year == 2003)
        &(df.Month != 1)&(df.Month != 2)&(df.Month != 3)
        &(df.Month != 4)&(df.Month != 5))) & (df.Cancelled == 1)]
dfwc = df[(df.Cancelled == 1) & (df.CancellationCode == "B")]
flights_cancelled = dfowc.count()
#print(flights_cancelled)
flights_cancelled_weather = dfwc.count()
#print(flights_cancelled_weather)

#Month wise analysis: Overall
print "\nOverall\n"
dfmonth = dfwc.select(df.Month)
rall = dfmonth.rdd
ralla = rall \
      .flatMap(lambda x:x) \
      .map(lambda x:(x,1)) \
      .reduceByKey(lambda a,b:a+b)
month_all = ralla.top(12, key=lambda t: t[1])
for (m, c) in month_all:
    print "Month:\t", m, "\t Count:\t", c

#Function for route wise analysis
def routes(dataframe):
    dfR = dataframe.select(dataframe.Origin,dataframe.Dest)
    dfRa = dfR.groupby(['Origin','Dest']).agg(f.count("*").alias('count'))
    dfR_final_desc = dfRa.orderBy('count',ascending=False)
    dfRb = dfR_final_desc.join(dfairport_a, dfR_final_desc.Origin == dfairport_a.iata)
        .drop('iata').withColumnRenamed("city","Origin City")
    dfRc = dfRb.join(dfairport_a, dfRb.Dest == dfairport_a.iata).drop('iata')
        .withColumnRenamed("city","Destination city")
    dfR_final_asc = dfRa.orderBy('count',ascending=True)
    #dfR_final_asc.show(50)
    #Since all the years have 1 as least affected route
    dfR_final_asc.filter(f.col("count") == "1").count()
    dfRc.show(5)

#Calling route wise analysis function : year wise
print "\nRoute Wise Analysis\n\n2003-Most affected(5) and least affected count\n"
routes(df2003b)
print "\n2004-Most affected(5) and least affected count\n"
routes(df2004b)
print "\n2005-Most affected(5) and least affected count\n"
routes(df2005b)
print "\n2006-Most affected(5) and least affected count\n"
routes(df2006b)
print "\n2007-Most affected(5) and least affected count\n"
routes(df2007b)
print "\n2008-Most affected(5) and least affected count\n"
routes(df2008b)
#Calling route wise analysis function : Overall
print "\nOverall-Most affected(5) and least affected count\n"
routes(dfwc)

#Function for airline wise analysis
def airlines(dataframe1, dataframe2):
    dfAirway = dataframe1.select(dataframe1.UniqueCarrier)
    dfAirwaya = dfAirway.groupby(['UniqueCarrier']).agg(f.count("*").alias('Wcount'))
    #dfAirway_final_asc = dfAirwaya.orderBy('count',ascending=True)
    dfA = dataframe2.groupby(['UniqueCarrier']).agg(f.count("*").alias('Cancel_count'))
        .withColumnRenamed("UniqueCarrier","WeatherCarrier")
    #dfA1 = dfA.orderBy('UniqueCarrier',ascending=True)
    dfA2 = dfAirwaya.join(dfA, dfAirwaya.UniqueCarrier ==
    dfA.WeatherCarrier).drop('UniqueCarrier')
```

```
268         #In order to ignore large counts, percentage is calculated
269         dfA3 = dfA2.withColumn("Percentage",(f.col("Wcount")/f.col("Cancel_count")*100))
270         dfA4c = dfA3.join(df_carriers, dfA3.WeatherCarrier == df_carriers.Code).drop('Code')
271             .withColumnRenamed("Description","Airways")
272         dfA4c.orderBy("Percentage",ascending=False).show(5)
273         dfA4c.orderBy("Percentage",ascending=True).show(1)
274 
275     #Calling airline wise analysis function: year wise
276     print "\nAirline Wise Analysis\n\n2003-Most affected(5) and the least affected\n"
277     airlines(df2003b,df2003a)
278     print "\n2004-Most affected(5) and the least affected\n"
279     airlines(df2004b,df2004a)
280     print "\n2005-Most affected(5) and the least affected\n"
281     airlines(df2005b,df2005a)
282     print "\n2006-Most affected(5) and the least affected\n"
283     airlines(df2006b,df2006a)
284     print "\n2007-Most affected(5) and the least affected\n"
285     airlines(df2007b,df2007a)
286     print "\n2008-Most affected(5) and the least affected\n"
287     airlines(df2008b,df2008a)
288     #Calling airline wise analysis function: overall
289     print "\nOverall-Most affected(5) and the least affected\n"
290     airlines(dfwc,dfowc)
291 
```