Dataset Source = https://www.kaggle.com/datasets/robikscube/flight-delay-dataset-20182022?select=Combined_Flights_2022.csv

I have used a Flight Delay Dataset to do some analysis and get some insights with the help of Big Data platforms like Spark and Hadoop. I will guide the readers about the process used for this analysis and introduce you all to the Big Data systems.

INTRODUCTION

Let me introduce you all to the basic concepts of Big Data platforms.

What is Apache Spark?

Apache Spark is an open-source, distributed processing system used for big data workloads. It utilizes in-memory caching and optimized query execution for fast queries against data of any size. Simply put, Spark is a fast and general engine for large-scale data processing. (Spark.apache.org, 2022)

The fast part means that it's faster than previous approaches to work with Big Data like classical MapReduce. The secret for being faster is that Spark runs on memory (RAM), and that makes the processing much faster than on disk drives.

The general part means that it can be used for multiple things like running distributed SQL, creating data pipelines, ingesting data into a database, running Machine Learning algorithms, working with graphs or data streams, and much more.

What is Hadoop?

Apache Hadoop is an open-source framework that is used to efficiently store and process large datasets ranging in size from gigabytes to petabytes of data. Instead of using one large computer to store and process the data, Hadoop allows clustering multiple computers to analyse massive datasets in parallel more quickly. (aws.amazon.com, 2022)

Hadoop consists of four main modules:

Hadoop Distributed File System (HDFS) – A distributed file system that runs on standard or low-end hardware. HDFS provides better data throughput than traditional file systems, in addition to high fault tolerance and native support of large datasets.

Yet Another Resource Negotiator (YARN) – Manages and monitors cluster nodes and resource usage. It schedules jobs and tasks.

MapReduce – A framework that helps programs do the parallel computation on data. The map task takes input data and converts it into a dataset that can be computed in key value pairs. The output of the map task is consumed by reduce tasks to aggregate output and provide the desired result.

Hadoop Common – Provides common Java libraries that can be used across all modules.

PROJECT COMPONENTS

The first thing that we do is to import important files and also execute them

```
[1]: import import_ipynb
[2]: import HDFSLauncher
     importing Jupyter notebook from HDFSLauncher.ipynb
[3]: hdfs=HDFSLauncher.get_hdfs()
```

After initiating the HDFS file we make a directory where we store all of our data so that we can use it for further analysis.

```
[ ]: hdfs.mkdir("/data/projects/fall_2022/jbhatia1/flight_dataset")
[7]: with open('Combined_Flights_2022.csv','rb') as fid:
         hdfs.upload('/data/projects/fall_2022/jbhatia1/flight_dataset/Combined_Flights_2022.csv',fid)
```

Here I have created a directory under my username where I have stored the dataset used for the analysis.

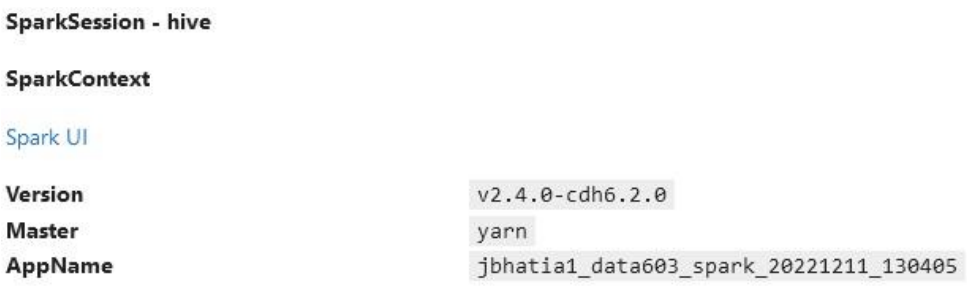
These are the two datasets that I uploaded in the HDFS but I used the recent dataset that is "Combined_Flights_2022.csv" because I wanted to do analysis for the year 2022.

```
[4]: hdfs.ls("/data/projects/fall_2022/jbhatia1/flight_dataset")
[4]: ['/data/projects/fall_2022/jbhatia1/flight_dataset/Combined_Flights_2021.csv',
      '/data/projects/fall_2022/jbhatia1/flight_dataset/Combined_Flights_2022.csv']
```

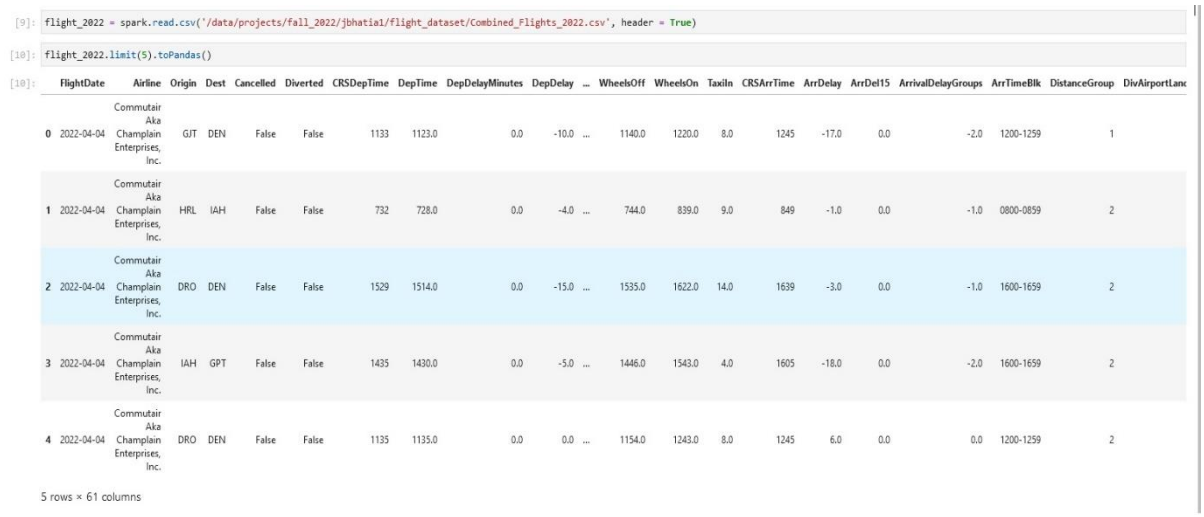
The next step is to import and initiate our spark launcher.

```
import SparkLauncher
importing Jupyter notebook from SparkLauncher.ipynb
conf= SparkLauncher.get_spark_conf()
spark=SparkLauncher.get_spark_session(conf, pack_venv=False)
Creating Spark Session: jbhatia1_data603_spark_20221211_130405
```

You can see my spark session getting activated



After initiating Spark and Hadoop now we are all set to do our analysis on the dataset that we uploaded. The first step is to read the dataset and after that create a parquet file.



```
[9]: flight_2022 = spark.read.csv('/data/projects/fall_2022/jbhatia1/flight_dataset/Combined_Flights_2022.csv', header = True)
```

```
[10]: flight_2022.limit(5).toPandas()
```

[10]:

	FlightDate	Airline	Origin	Dest	Cancelled	Diverted	CRSDepTime	DepTime	DepDelayMinutes	DepDelay	...	WheelsOff	WheelsOn	TaxiIn	CRSArrTime	ArrDelay	ArrDel15	ArrivalDelayGroups	ArrTimeBlk	DistanceGroup	DivAirportLanc
0	2022-04-04	Commutair Aka Champlain Enterprises, Inc.	GJT	DEN	False	False	1133	1123.0	0.0	-10.0	...	1140.0	1220.0	8.0	1245	-17.0	0.0	-2.0	1200-1259	1	
1	2022-04-04	Commutair Aka Champlain Enterprises, Inc.	HRL	IAH	False	False	732	728.0	0.0	-4.0	...	744.0	839.0	9.0	849	-1.0	0.0	-1.0	0800-0859	2	
2	2022-04-04	Commutair Aka Champlain Enterprises, Inc.	DRO	DEN	False	False	1529	1514.0	0.0	-15.0	...	1535.0	1622.0	14.0	1639	-3.0	0.0	-1.0	1600-1659	2	
3	2022-04-04	Commutair Aka Champlain Enterprises, Inc.	IAH	GPT	False	False	1435	1430.0	0.0	-5.0	...	1446.0	1543.0	4.0	1605	-18.0	0.0	-2.0	1600-1659	2	
4	2022-04-04	Commutair Aka Champlain Enterprises, Inc.	DRO	DEN	False	False	1135	1135.0	0.0	0.0	...	1154.0	1243.0	8.0	1245	6.0	0.0	0.0	1200-1259	2	

5 rows × 61 columns

The above picture shows the commands to read the csv file with spark and also the dataset which we are working on.

To use the parquet file we need to go through 3 process :-

1. Creating the parquet file

```
[14]: flight_2022.write.parquet('/etl/projects/fall_2022/jbhatia1/flight_dataset')
```

```
[17]: hdfs.ls('/etl/projects/fall_2022/jbhatia1/flight_dataset',detail=True)[1]
```

```
[17]: {'kind': 'file',
 'name': '/etl/projects/fall_2022/jbhatia1/flight_dataset/part-00000-582a4e13-bd39-4762-bbb0-df1c21fcecc6-c000.snappy.parquet',
 'owner': 'jbhatia1',
 'group': 'hadoop',
 'last_modified_time': 1670784252,
 'last_access_time': 1670784241,
 'size': 10942454,
 'replication': 3,
 'block_size': 268435456,
 'permissions': 420}
```

2. Reading the parquet file

```
[23]: flight_2022_df_from_parquet = spark.read.parquet('/etl/projects/fall_2022/jbhatia1/flight_dataset')
```

3. Verifying parquet file

```
[26]: flight_2022_df_from_parquet

[26]: DataFrame[FlightDate: string, Airline: string, Origin: string, Dest: string, Cancelled: string, Diverted: string, CRSDepTime: string, DepTime: string, DepDelayMinutes: string, DepDelay: string, ArrTime: string,
ArrDelayMinutes: string, AirTime: string, CRSElapsedTime: string, ActualElapsedTime: string, Distance: string, Year: string, Quarter: string, Month: string, DayofMonth: string, DayOfWeek: string, Marketing_Airli
ne_Network: string, Operated_or_Branded_Code_Share_Partners: string, DOT_ID_Marketing_Airline: string, IATA_Code_Marketing_Airline: string, Flight_Number_Marketing_Airline: string, Operating_Airline: string, DOT
_ID_Operating_Airline: string, IATA_Code_Operating_Airline: string, Tail_Number: string, Flight_Number_Operating_Airline: string, OriginAirportID: string, OriginAirportSeqID: string, OriginCityMarketID: string,
OriginCityName: string, OriginState: string, OriginStateFips: string, OriginStateName: string, OriginWac: string, DestAirportID: string, DestAirportSeqID: string, DestCityMarketID: string, DestCityName: string,
DestState: string, DestStateFips: string, DestStateName: string, DestWac: string, DepDel15: string, DepartureDelayGroups: string, DepTimeBlk: string, TaxiOut: string, WheelsOff: string, WheelsOn: string, TaxiIn:
string, CRSArrTime: string, ArrDelay: string, ArrDel15: string, ArrivalDelayGroups: string, ArrTimeBlk: string, DistanceGroup: string, DivAirportLandings: string]

[29]: flight_2022_df_from_parquet.limit(5).toPandas()
```

[29]:

	FlightDate	Airline	Origin	Dest	Cancelled	Diverted	CRSDepTime	DepTime	DepDelayMinutes	DepDelay	...	WheelsOff	WheelsOn	TaxiIn	CRSArrTime	ArrDelay	ArrDel15	ArrivalDelayGroups	ArrTimeBlk	DistanceGroup	DivAirportLandings
0	2022-04-19	Envoy Air	LEX	DFW	False	False	800	756.0	0.0	-4.0	...	808.0	901.0	13.0	931	-17.0	0.0	-2.0	0900-0959	4	0
1	2022-04-20	Envoy Air	LEX	DFW	False	False	800	753.0	0.0	-7.0	...	804.0	927.0	15.0	931	11.0	0.0	0.0	0900-0959	4	0
2	2022-04-21	Envoy Air	LEX	DFW	False	False	800	758.0	0.0	-2.0	...	828.0	933.0	13.0	931	15.0	1.0	1.0	0900-0959	4	0
3	2022-04-22	Envoy Air	LEX	DFW	False	False	800	757.0	0.0	-3.0	...	812.0	909.0	15.0	931	-7.0	0.0	-1.0	0900-0959	4	0
4	2022-04-23	Envoy Air	LEX	DFW	False	False	800	758.0	0.0	-2.0	...	815.0	915.0	21.0	931	5.0	0.0	0.0	0900-0959	4	0

5 rows × 61 columns

After executing all these we now come to our Exploratory Data Analysis part where I have used my SQL and python skills to do some analysis on the dataset and get some visualizations from the dataset.

To perform the SQL queries on the dataset we need to first the dataset into TABLE form and we have used createOrReplaceTempView.( ) to convert our CSV file to a temporary table.

```
# creating a temp table of the dataframe

flight_2022.createOrReplaceTempView('flight_2022_t')
```

NUMBER OF FLIGHTS GETTING CANCELLED

I planned to see the number of flights getting cancelled and also compare it with the number of flights that were not cancelled.

```
In [52]: cnt=spark.sql("select Cancelled, count(Cancelled) as count from flight_2022_t group by Cancelled")

In [53]: cnt.show()
```

```
+---------+-------+
|Cancelled|  count|
+---------+-------+
|    False|3955126|
|     True| 123192|
+---------+-------+
```

We see by the numbers that the flights getting cancelled is a small amount but it still causes problems to people.

NUMBER OF FLIGHTS LANDING IN MARYLAND STATE AND GETTING CANCELLED

As I live in Maryland I want to know the number of flights that reached Maryland airports (out of curiosity) and also the number of flights that were cancelled by the airlines.

```
[88]: # Number of flights landing in Maryland state

      mf= spark.sql(" select * from flight_2022_t where DestStateName='Maryland' ")

[89]: mf.count()

[89]: 49533

[90]: # Number of flights landing in Maryland state that got cancelled

      mfc= spark.sql(" select * from flight_2022_t where DestStateName='Maryland' and Cancelled = True ")

[91]: mfc.count()

[91]: 1826
```

Thus from the above data, we can see that the probability of getting your flight cancelled is around 3.6% if you are traveling to Maryland state.

I also calculated the top airlines that cancel flights with destination as Maryland: -

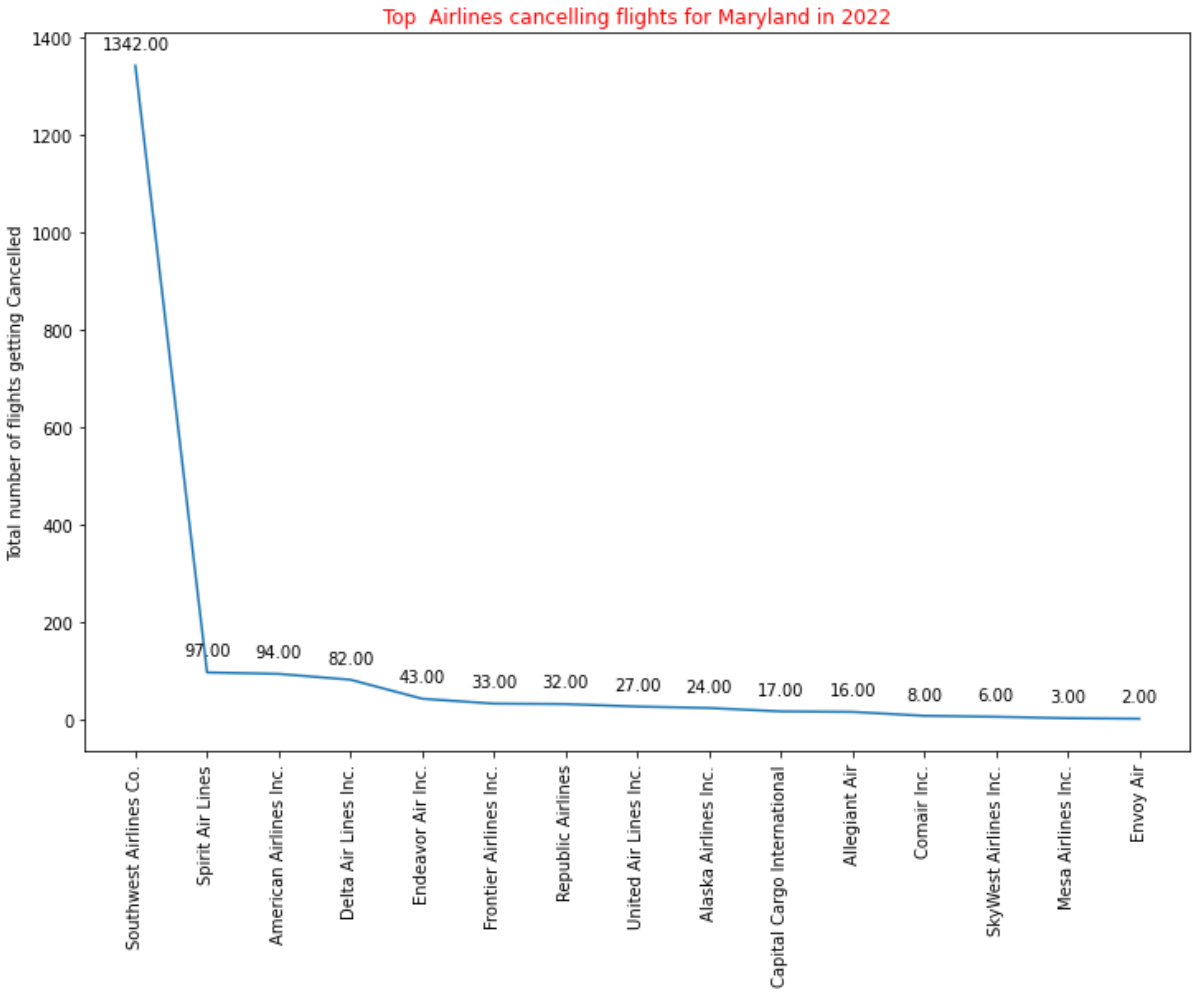
Given below is the SQL query for getting the desired output.

```
[179]: # Airlines that were cancelled the most for Maryland

cm=spark.sql(" select Airline,count(Airline) as cnt from flight_2022_t where DestStateName='Maryland' and Cancelled = True group by Airline order by cnt DESC")
```

```
[180]: cm.show()
```

```
+--------------------+----+
|             Airline| cnt|
+--------------------+----+
|Southwest Airline...|1342|
|    Spirit Air Lines|  97|
|American Airlines...|  94|
|Delta Air Lines Inc.|  82|
|   Endeavor Air Inc.|  43|
|Frontier Airlines...|  33|
|   Republic Airlines|  32|
|United Air Lines ...|  27|
|Alaska Airlines Inc.|  24|
|Capital Cargo Int...|  17|
|       Allegiant Air|  16|
|         Comair Inc.|   8|
|SkyWest Airlines ...|   6|
|  Mesa Airlines Inc.|   3|
|           Envoy Air|   2|
+--------------------+----+
```



This shows us that the top 10 airlines that cancelled flights for Maryland in 2022 and we can see that the SouthWest Airlines Co. cancelled around 1342 flights this year.

AIRLINES WITH MOST NUMBER OF FLIGHTS IN 2022

I calculated the Airlines with most number of flights in 2022.

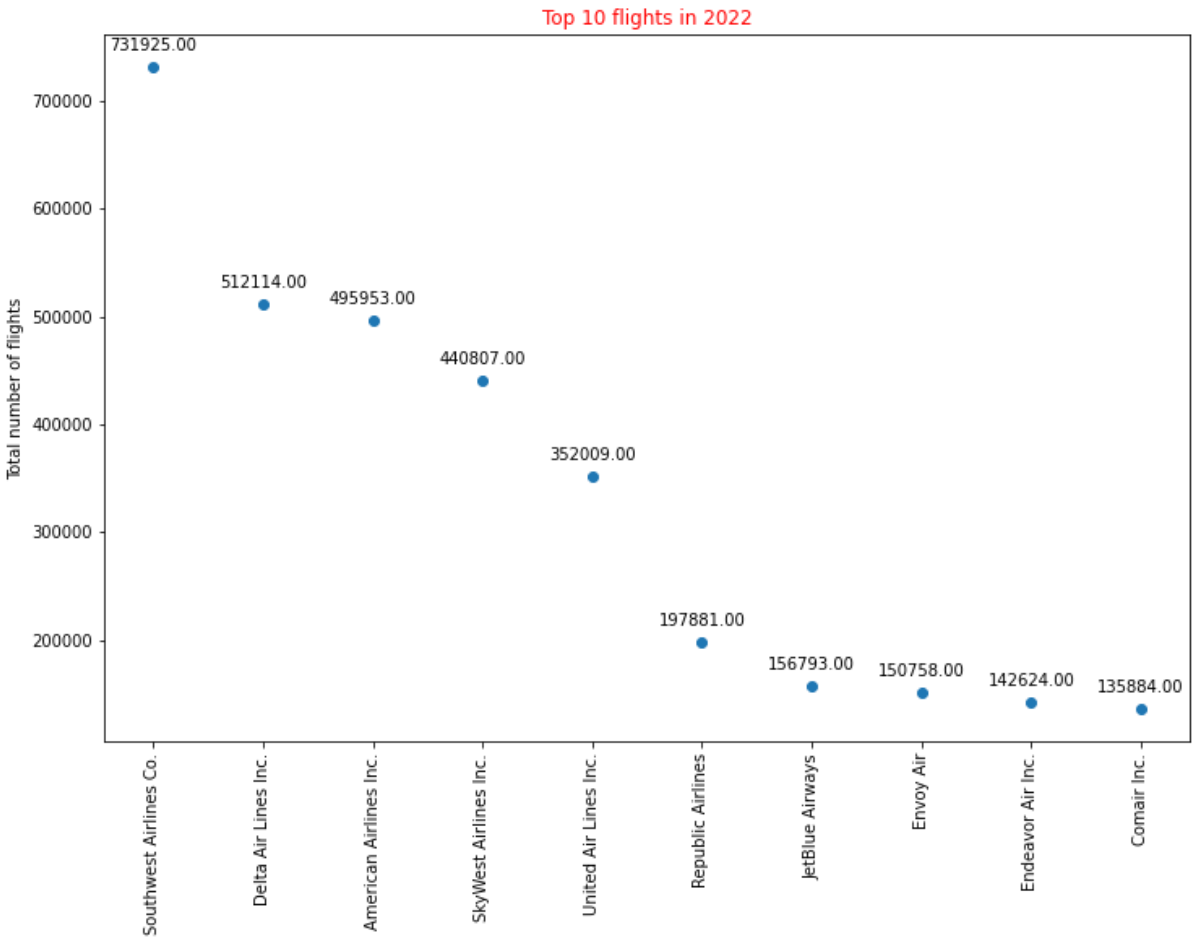
```
[161]: # Count of specific airlines

ac=spark.sql("Select Airline,count(Airline) as cnt from flight_2022_t group by Airline order by cnt DESC limit 10")
```

```
[162]: ac.show()
```

```
+--------------------+------+
|             Airline|   cnt|
+--------------------+------+
|Southwest Airline...|731925|
|Delta Air Lines Inc.|512114|
|American Airlines...|495953|
|SkyWest Airlines ...|440807|
|United Air Lines ...|352009|
|   Republic Airlines|197881|
|     JetBlue Airways|156793|
|           Envoy Air|150758|
|   Endeavor Air Inc.|142624|
|         Comair Inc.|135884|
+--------------------+------+
```

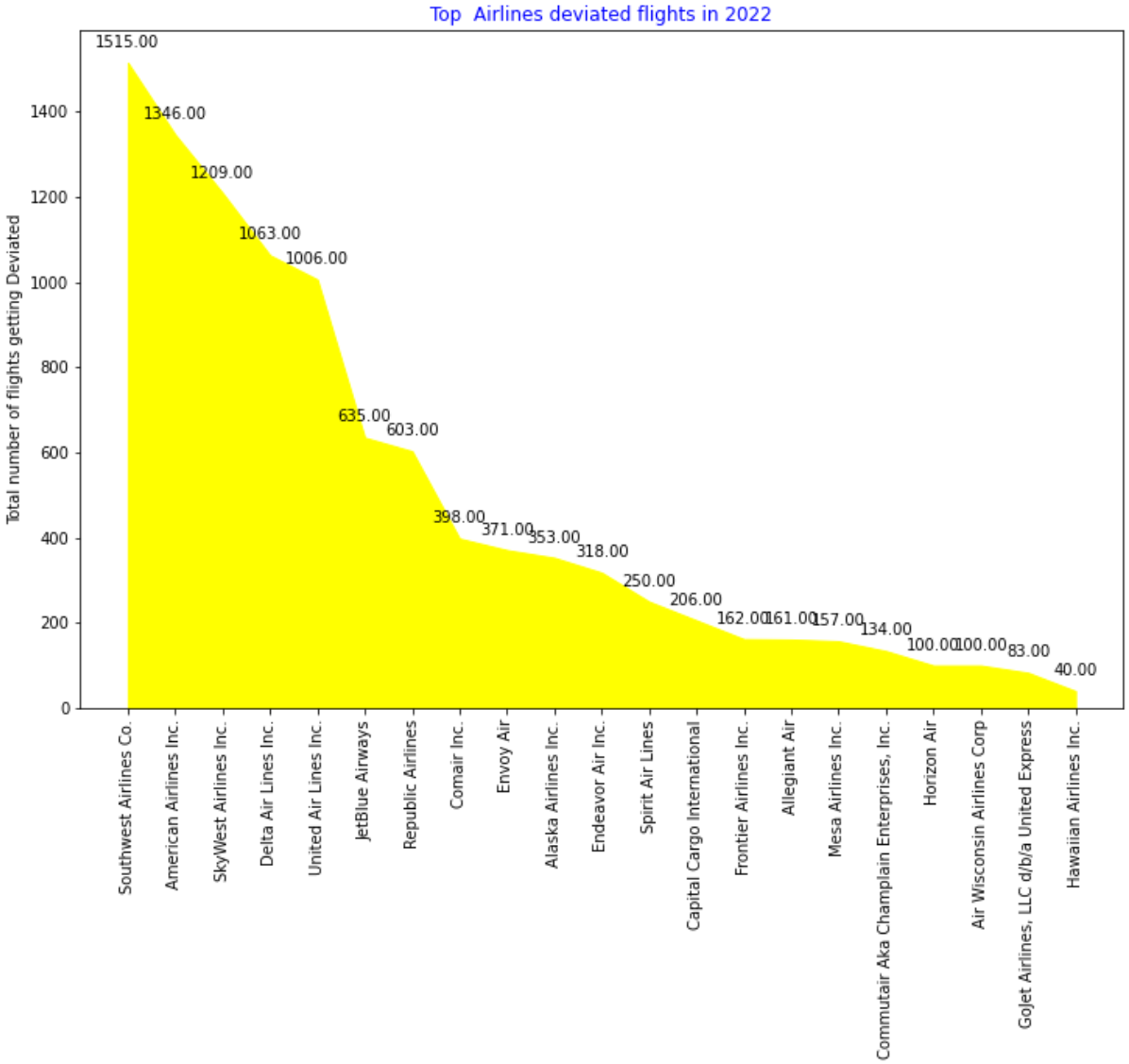
The above image shows the data of the flights arranged in Descending order of the total number of flights.



We can see from the above figure that SouthWest Airlines Co. leads with the most number of flights and also Delta airlines and American Airlines are close with their number of flights this year.

AIRLINES WITH THE MOST DEVIATED FLIGHTS

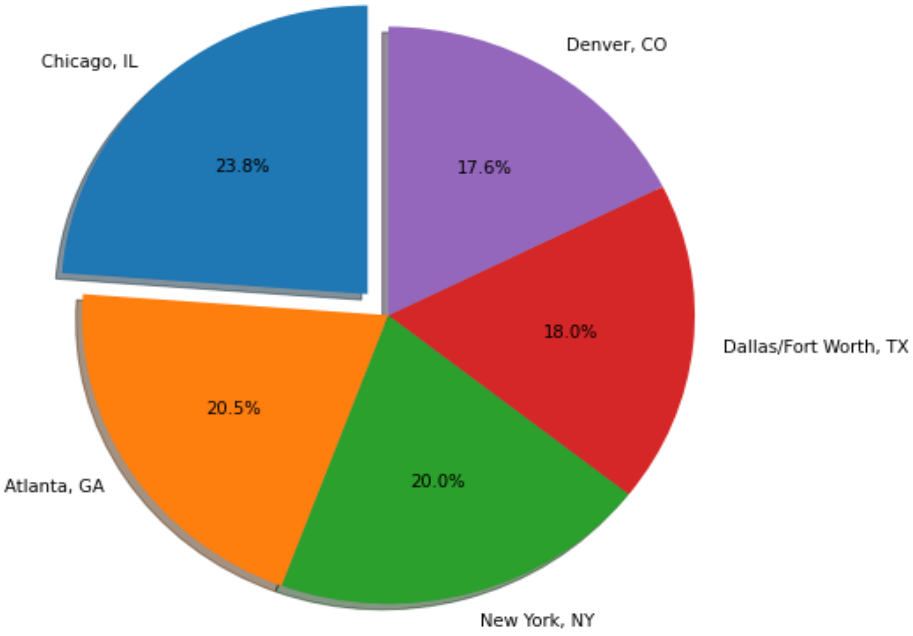
After analysing the cancelled flights I thought that it would be god to let the users know if the flights were deviated a lot this year or not.



Here we see again that SouthWest Airlines Co. leads the chart with the most deviations accompanied by American Airlines. Thus we can say that if you are booking a SouthWest Airlines Co. flight there is a 0.21% possibility and if you are booking an American Airlines flight there is a 0.28% possibility that you will deviate from your route.

MOST FAMOUS CITIES THAT PEOPLE PREFER TO TRAVEL TO IN 2022

I like travelling and I like to see the trends that are going around in this world. This helps me to explore more about places and also get to know about the things that attract the people around the world the most.

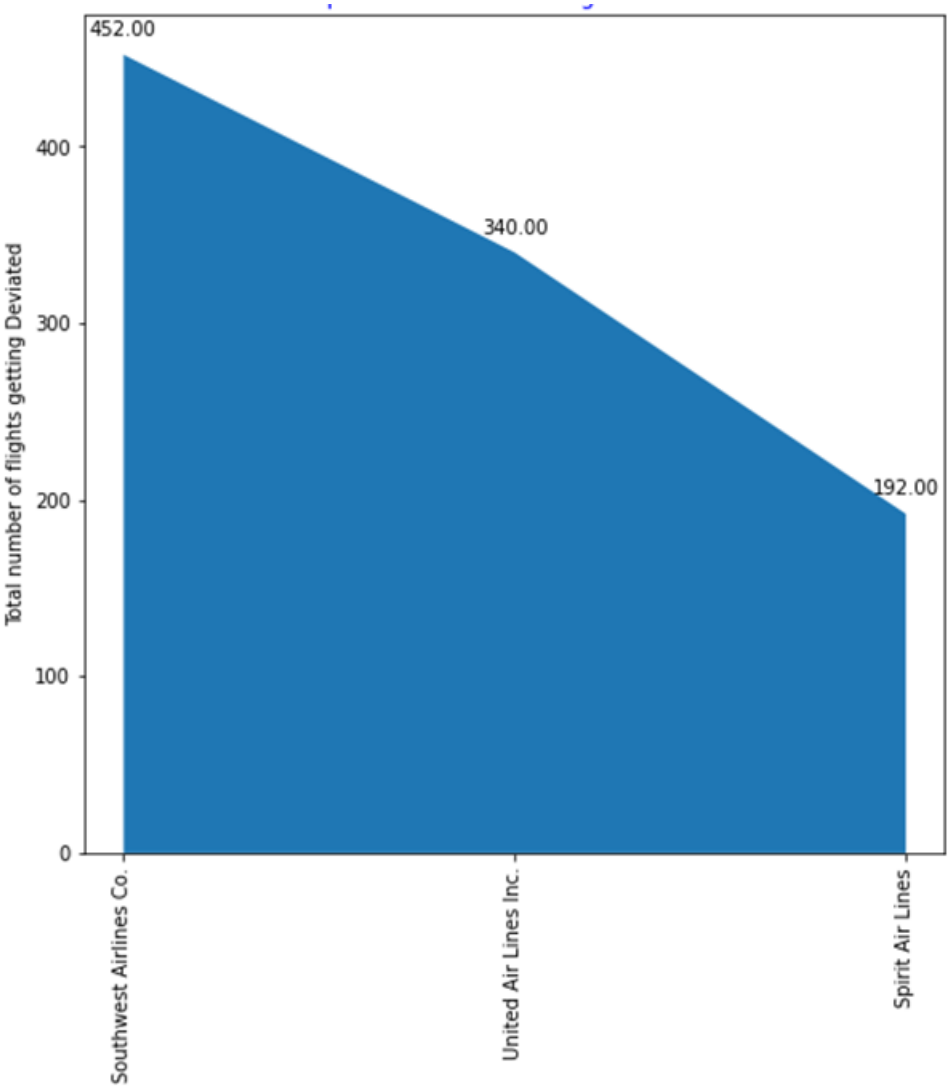


Thus, from the above diagram we see that Chicago is the most famous city among the people of USA to visit or travel followed by Atlanta and New York.

AIRLINES WITH MOST NUMBER OF FLIGHTS BETWEEN MARYLAND AND CALIFORNIA

I love California, the weather, the night life and people. Thus, having this dataset in hand I tried to explore top 3 airlines with most flights between Maryland and California.

Airline	cnt	OriginStateName	DestStateName
Southwest Airline...	452	Maryland	California
United Air Lines ...	340	Maryland	California
Spirit Air Lines	192	Maryland	California



Thus, we can say that Southwest Airlines Co. leads here too with the most number of flights between these destinations.

Citations

1. https://spark.apache.org/docs/latest/
2. https://aws.amazon.com/emr/details/hadoop/