FLIGHT DATA ANALYSIS

A

Project Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

K.Saikumar(1602-17-737-034)

N.Sri Surya(1602-17-737- 49)

Under the guidance of

Mr. Rajashekar

Assistant Professor

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2019-20

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

We, **K.Saikumar,N.SriSurya** bearing hall ticket numbers, **1602- 17-737-034**,**1602-17-737-049** respectively hereby declare that the project report entitled **"Flight Data Analysis"** under the guidance of **Mr. Rajashekar,** Assistant Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement of THEME BASED PROJECT of VI Semester of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project.

K.Sai kumar
1602-17-737-034

N.Sri Surya
1602-17-737-049

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled **"Flight Data Analysis"** being submitted by **K.Saikumar , N.SriSurya** bearing **H.T.NO:1602-17-737-034 ,1602-17-737-049** in partial fulfilment of the requirements for the completion of THEME BASED PROJECT of Bachelor of Engineering, VI Semester, in Information Technology is a record of bonafide work carried out by him/her under my guidance.

Mr. Rajashekar
Assistant Professor
Internal Guide

Dr. K. Ram Mohan Rao
HOD , IT

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. We would like to take the opportunity to express our humble gratitude to **Mr Rajashekar**, our internal guide, under whose guidance we have finished this project successfully. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the work. We would like to thank all faculty members and staff of the Department of Information Technology, Vasavi College of Engineering for their generous help in various ways for the completion of this project. Finally, yet importantly, We would like to express our heartfelt thanks to our respected **Dr.K.Ram Mohan Rao(HOD-IT)** who was very helpful in providing resources for the project and also our project coordinators **Mrs.Leela Pallava(Asst.Prof**-IT)**,Mr.Sreenivasa Chakravarthy(Asst.Prof,IT)** whose efforts were constant in monitoring our project.

K.Saikumar

1602-17-737-034

N.SriSurya

1602-17-737-049

Abstract

. In the present day, travelling via flights has become one of the most popular options for passengers across the world; and hence there is a need for more efficient management of the massive amounts of data being generated by the airlines companies. This problem has found its solution through Big Data Analytics. The main objective of our project is to analyse is to implement Hadoop Mapreduce and analyse near real-time data of the various flights taking off and provide accurate computation and visualisation of the results. We also hope to help the aviation system through understanding the flight data such as delays and helping them overcome losses. This will, in turn save capital for air authorities and make it much more convenient for passengers too.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

In the contemporary world, Data analysis is a challenge in the era of varied inters- disciplines though there is a specialization in the respective disciplines. In other words, effective data analytics helps in analyzing the data of any business system. But it is the big data which helps and axial rates the process of analysis of data paving way for a success of any business intelligence system. With the expansion of the industry, the data of the industry also expands. Then, it is increasingly difficult to handle huge amount of data that gets generated no matter what's the business is like, range of fields from social media to finance, flight data, environment and health. Big Data can be used to assess risk in the insurance industry and to track reactions to products in real time. Big Data is also used to monitor things as diverse as wave movements, flight data, traffic data, financial transactions, health and crime. The challenge of Big Data is how to use it to create something that is value to the user. How can it be gathered, stored, processed and analyzed it to turn the raw data information to support decision making. In this paper Big Data is depicted in a form of case study for Airline data.

System Analysis

2.1 Existing System:

In the existing System the website is not user friendly and airlines staff have to manually look into the details of an airlines for its specifics. However, our project incorporates big data analysis by performing analysis on the past 11 years of flight data. Through this, both customers and airlines agencies can easily identify which airlines to choose and which to avoid.

2.2 Proposed System:

Our Website is user friendly and The proposed method is made by considering following scenario under consideration An Airport has huge amount of data related to number of flights, data and time of arrival and dispatch, flight routes, No. of airports operating in each country, list of active airlines in each country. The problem they faced till now it's, they have ability to analyze limited data from databases.

2.3 Features of Hadoop

- Open Source:Hadoop is an open-source project, which means its source code is available free of cost for inspection, modification, and analyses that allows enterprises to modify the code as per their requirements.

- Highly Scalable:Hadoop cluster is scalable means we can add any number of nodes (horizontal scalable) or increase the hardware capacity of nodes (vertical scalable) to achieve high computation power. This provides horizontal as well as vertical scalability to the Hadoop framework.

● Fault Tolerance:Fault tolerance is the most important feature of Hadoop.HDFS in Hadoop 2 uses a replication mechanismto provide fault tolerance.It creates a replica of each block on the different machines depending on the replication factor (by default, it is 3). So if any machine in a cluster goes down, data can be accessed from the other machines containing a replica of the same data. Hadoop 3 has replaced this replication mechanism by erasure coding. Erasure coding provides the same level of fault tolerance with less space. With Erasure coding, the storage overhead is not more than 50%.

● High Availability:This feature of Hadoop ensures the high availability of the data, even in unfavorable conditions.Due to the fault tolerance feature of Hadoop , if any of the DataNodes goes down, the data is available to the user from different DataNodes containing a copy of the same data.\Also, the high availability Hadoop cluster consists of 2 or more running NameNodes (active and passive) in a hot standby configuration. The active node is the 11 NameNode, which is active. Passive node is the standby node that reads edit logs modification of active NameNode and applies them to its own namespace.

● Cost-Effective:Since the Hadoop cluster consists of nodes of commodity hardware that are inexpensive, thus provides a cost-effective solution for storing and processing big data. Being an open-source product, Hadoop doesn’t need any license.

2.4 Architecture Used:

- **Eclipse:** Eclipse is an integrated development environment(IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment.

- **Apache Hadoop using Map reduce: Map**

  Reduce is programming paradigm that enables massive scalability

  across hundreds or thousands of servers in a Hadoop cluster.

- **HTML**

- **JavaScript**

2.5 Overview of Hadoop Ecosystem:

Apache Hadoop is a collection of open-source software utilities providing a software framework for distributed storage and processing of big data using the MapReduce programming model. Stable possible version of Hadoop (at present) for Ubuntu or Linux is used i.e., (2.9.3). It is always advisable to download not the latest version but the previous version or its predecessor version of it for ease of use for the licensed Apache website.

HDFS (Hadoop Distributed File System): HDFS has the most important job to perform in the Hadoop framework. It distributes the data and stores it on each node present in a cluster, simultaneously. This process reduces the total time to store data onto the disk.

MapReduce (Read/Write Large Datasets into/from Hadoop using MR): Hadoop MapReduce is another important part of the system that processes the huge volumes of data stored in a cluster. It allows parallel processing of all the data stored by HDFS. Moreover, it resolves the issue of high cost of processing through the massive scalability in a cluster.

Apache Pig (Pig is a kind of ETL for the Hadoop ecosystem): It is the high-level scripting language to write the data analysis programmes for huge data sets in the Hadoop cluster. Pig enables developers to generate query execution routines for analysis of large data sets. The scripting language is known as Pig Latin, which one key part of Pig, and the second key part is a compiler.

Apache HBase (OLTP/NoSQL) sources: It is a column-oriented database that supports the working of HDFS on a real-time basis. It is enabled to process large database tables, i.e. a file containing millions of rows and columns. An important use of HBase is the efficient use of master nodes for managing region servers.

Apache Hive (Hive is a SQL engine on Hadoop): With a SQL-like interace, Hive allows ofr the squaring of data from HDFS. The Hive version of SQL language is called as HiveQL.

Apache Sqoop (Data Import/Export from RDBMS [SQL sources] into Hadoop): It is an application that helps with the import and export of data from Hadoop to other relational database management systems. It can transfer the bulk of your data. Sqoop is based on connector architecture that backs the plugins for establishing connectivity to new external system.

Chapter 3

SYSTEM DESIGN

3.1 Use Cases

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases.

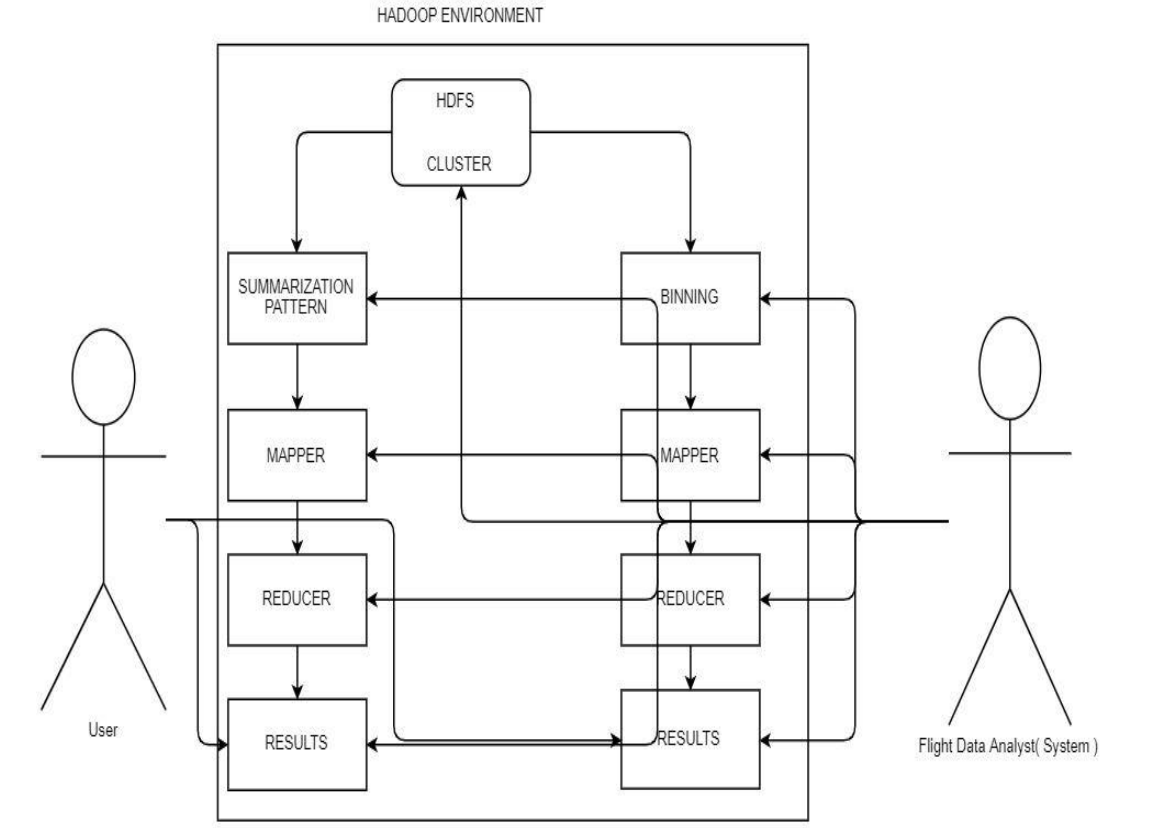

Fig 3.1 Use Case Diagram

3.2 UML Static Diagram

A **Data-Flow Diagram** is a way of representing a **flow** of **data** of a process or a system. The DFD also provides information about the outputs and the inputs of each entity and the process itself.

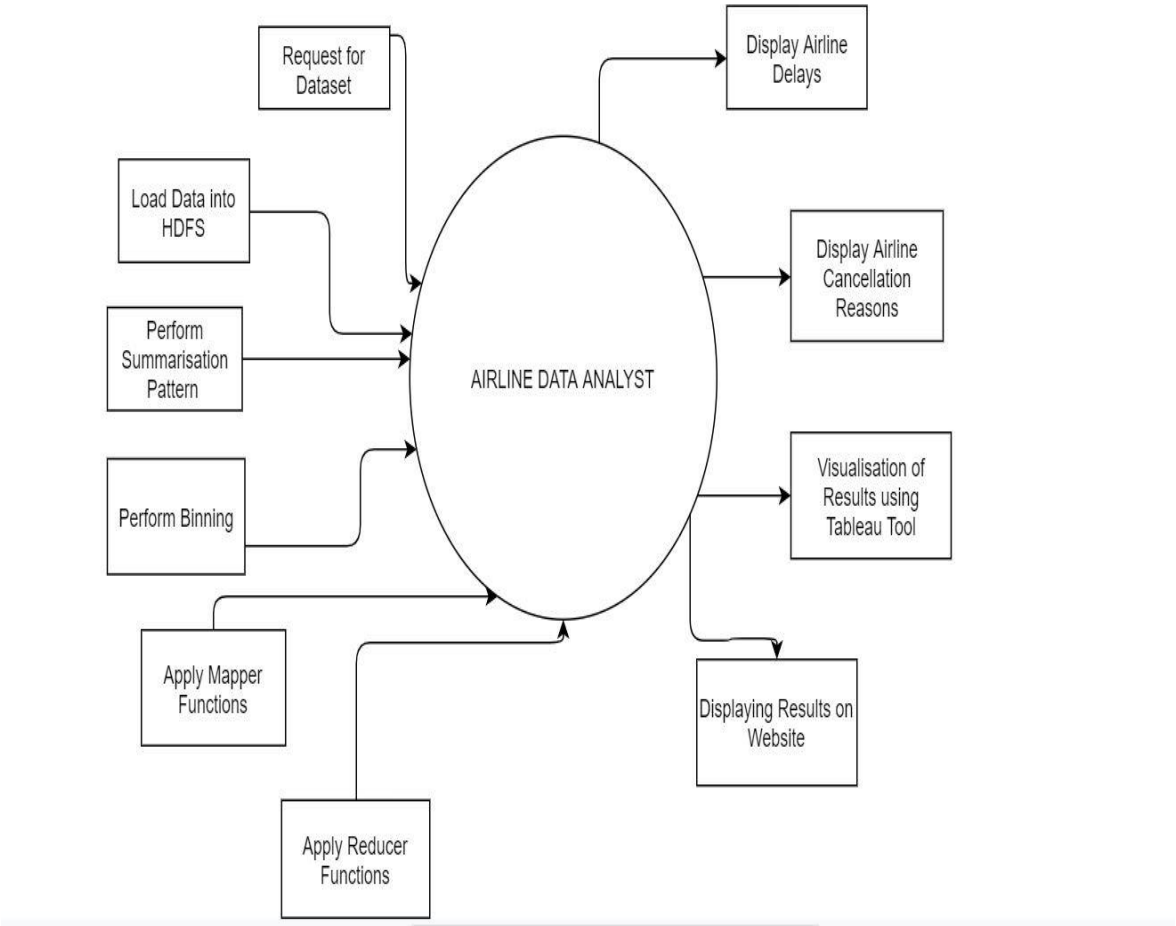

Fig 3.2 Data Flow Diagram

3.3 UML Run Time Diagram

A **sequence diagram** shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the **sequence** of messages exchanged between the objects needed to carry out the functionality of the scenario.

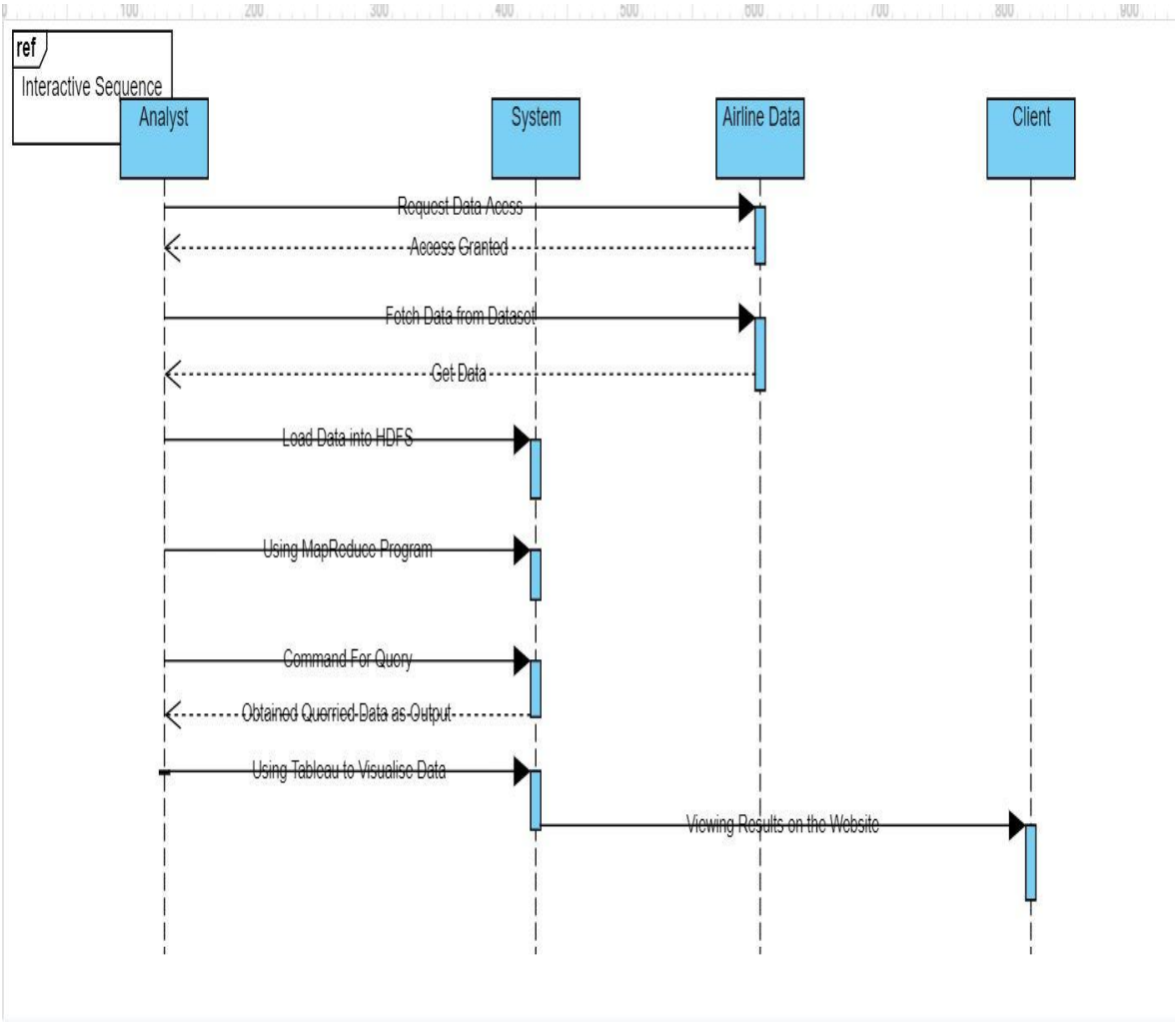

Fig 3.3 Sequence Diagram

Chapter – 4

IMPLEMENTATION

We have the following modules in our project:

4.1 Website Home Page

Through this module we are able to view the visualisations of the results we want. The website is setup to be interactive using Bootstrap and JavaScript and the UI is simple and lucid. The Website Page is a HTML File to which the two tableau visualisations are attached as hyperlinks.

4.2 Airline Carrier Delays

If a user clicks on the Delays option, then they are redirected to the graph visualisation present in tableau public. Here the Average Carrier Delay(in minutes) is displayed via summarization pattern technique for the various flights travelling in the US over a period of 11 years.

4.3 Flight Cancellation Trends

If a user clicks on this option, then they are redirected to the graph visualisation present in tableau public. First, the cancellation reason count is displayed via a pie graph, through which a user gets a clear understanding of what the major reasons are for flights to be cancelled. Then they can also view a bar graph showing which airlines company has had the most cancellations over the years, thus equipping them with the knowledge needed while booking their next flight.

4.4 Source Code:

1) Summarization.java

```
package project.summarization;


import java.io.IOException;


import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;
```

```
import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class Summarization {


	public static class Summarization_Mapper extends Mapper<Object, Text, Text,
CompositeKeyWritable>{


		private Text UniqueCarrier = new Text();


		@Override

		protected void map(Object key, Text value, Mapper<Object, Text, Text,
CompositeKeyWritable>.Context context)

				throws IOException, InterruptedException {


			if (!value.toString().contains("UniqueCarrier")) {

	String[] input = value.toString().split(",");



	if(!input[14].equalsIgnoreCase("NA") && !input[15].equalsIgnoreCase("NA")){


		double carrierDelay = 0;

		if(input[15].equalsIgnoreCase("NA")) carrierDelay=0.0;

		else carrierDelay=Double.parseDouble(input[15]);
```

```
            CompositeKeyWritable outTuple= new
CompositeKeyWritable(Integer.parseInt(input[14]),Integer.parseInt(input[14]),

        Integer.parseInt(input[15]),Integer.parseInt(input[15]),carrierDelay,1);

            UniqueCarrier.set(input[8]);

            context.write(UniqueCarrier, outTuple);

          }

        }

            }



        }




        public static class Summarization_Reducer extends Reducer<Text,
CompositeKeyWritable, Text, CompositeKeyWritable> {



            @Override

            protected void reduce(Text key, Iterable<CompositeKeyWritable>
values,Reducer<Text, CompositeKeyWritable, Text, CompositeKeyWritable>.Context
context)

                                                throws IOException, InterruptedException {

                    int minArrDel = Integer.MAX_VALUE;

                int maxArrDel = Integer.MIN_VALUE;

                    int minDepDel = Integer.MAX_VALUE;

                int maxDepDel = Integer.MIN_VALUE;

                int count = 0;

                        double sum = 0;
```

```
                for (CompositeKeyWritable val : values) {

                    if (val.getArrMinDelay() < minArrDel)
                            minArrDel = val.getArrMinDelay();

                    if (val.getArrMaxDelay() > maxArrDel)
                            maxArrDel = val.getArrMaxDelay();

                    if (val.getDepMinDelay() < minDepDel)
                            minDepDel = val.getDepMinDelay();

                    if (val.getDepMaxDelay() > maxDepDel)
                            maxDepDel = val.getDepMaxDelay();

                    sum += val.getAverage()*val.getCount();
                                count +=val.getCount();

                }

                context.write(key,new CompositeKeyWritable(minArrDel,
maxArrDel, minDepDel, maxDepDel,(sum/count),count));
            }

        }
```

```
public static void main( String[] args ){
    System.out.println( "Hello World!" );

    Configuration conf = new Configuration();

            try {
                    Job job = Job.getInstance(conf, "stock price high");
                    job.setJarByClass(Summarization.class);
                    job.setMapperClass(Summarization_Mapper.class);
                    job.setCombinerClass(Summarization_Reducer.class);
                    job.setReducerClass(Summarization_Reducer.class);
                    job.setOutputKeyClass(Text.class);
                    job.setOutputValueClass(CompositeKeyWritable.class);

                    FileInputFormat.addInputPath(job, new Path(args[0]));
                    FileOutputFormat.setOutputPath(job, new Path(args[1]));

                    System.exit(job.waitForCompletion(true) ? 0 : 1);
            } catch (Exception e) {
                    e.printStackTrace();
            }
```

```
    }
}
```

2. CompositeKeyWritable.java

```
package project.summarization;

import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;

import org.apache.hadoop.io.Writable;

public class CompositeKeyWritable implements Writable{

    private int arrMinDelay;
    private int arrMaxDelay;
    private int depMinDelay;
    private int depMaxDelay;
        private double average;
        private long count;

        public CompositeKeyWritable(){}

    public CompositeKeyWritable(int arrMinDelay, int arrMaxDelay, int depMinDelay,
                int depMaxDelay, double average, int count) {
```

```
    this.arrMinDelay = arrMinDelay;
    this.arrMaxDelay = arrMaxDelay;
    this.depMinDelay = depMinDelay;
    this.depMaxDelay = depMaxDelay;
    this.average = average;
    this.count = count;
}

    public int getArrMinDelay() {
            return arrMinDelay;
    }

    public void setArrMinDelay(int arrMinDelay) {
            this.arrMinDelay = arrMinDelay;
    }

    public int getArrMaxDelay() {
            return arrMaxDelay;
    }

    public void setArrMaxDelay(int arrMaxDelay) {
            this.arrMaxDelay = arrMaxDelay;
    }

    public int getDepMinDelay() {
```

```
        return depMinDelay;
}

public void setDepMinDelay(int depMinDelay) {
        this.depMinDelay = depMinDelay;
}

public int getDepMaxDelay() {
        return depMaxDelay;
}

public void setDepMaxDelay(int depMaxDelay) {
        this.depMaxDelay = depMaxDelay;
}

public double getAverage() {
        return average;
}

public void setAverage(double average) {
        this.average = average;
}

public long getCount() {
        return count;
```

```
}

public void setCount(long count) {
        this.count = count;
}

public void readFields(DataInput in) throws IOException {
        arrMinDelay = in.readInt();
        arrMaxDelay = in.readInt();
        depMinDelay = in.readInt();
        depMaxDelay = in.readInt();
        average = in.readDouble();
        count = in.readLong();

}

public void write(DataOutput out) throws IOException {
        out.writeInt(arrMinDelay);
        out.writeInt(arrMaxDelay);
        out.writeInt(depMinDelay);
        out.writeInt(depMaxDelay);
        out.writeDouble(average);
        out.writeLong(count);
}
```

```
	@Override
	public String toString() {
		return (arrMinDelay +" " +arrMaxDelay +" "+ depMinDelay +" " +
depMaxDelay +" " + average +" "+ count );
	}
}
```

3) FlightCancellationReasonBinning.java

```
package project.binning;



import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.MultipleOutputs;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;



public class FlightCancellationReasonBinning {
```

```
public static class TMapper extends Mapper<Object, Text, Text, NullWritable>{

private MultipleOutputs<Text, NullWritable> mos=null;
//private final static IntWritable one = new IntWritable(1);
//private Text hour= new Text();

@Override
protected void setup(Context context) throws IOException, InterruptedException {
      mos = new MultipleOutputs(context);
 }

 @Override
 public void map(Object key, Text value, Context context)
      throws IOException, InterruptedException {

   String row[] = value.toString().split(",");
   if (!value.toString().contains("UniqueCarrier")) {
      if(!row[21].equalsIgnoreCase("0")){
               String cancellationCode = row[22];

               if(cancellationCode.equalsIgnoreCase("A"))
                  mos.write("bins", value, NullWritable.get(),"Carrier-cancellation");
               if(cancellationCode.equalsIgnoreCase("B"))
                  mos.write("bins", value, NullWritable.get(),"Weather-cancellation");
```

```
            if(cancellationCode.equalsIgnoreCase("C"))
               mos.write("bins", value, NullWritable.get(),"NAS-cancellation");
            if(cancellationCode.equalsIgnoreCase("D"))
               mos.write("bins", value, NullWritable.get(),"Security-cancellation");
         }
   }
}

   @Override
   protected void cleanup(Context context) throws IOException, InterruptedException {
      mos.close();
   }

}
public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException {
      Configuration conf = new Configuration();
      Job job = Job.getInstance(conf, "Cancellation Binning ");
      job.setJarByClass(FlightCancellationReasonBinning.class);

      job.setMapperClass(TMapper.class);

      MultipleOutputs.addNamedOutput(job, "bins", TextOutputFormat.class, Text.class,
IntWritable.class);
      MultipleOutputs.setCountersEnabled(job, true);
```

```
        job.setNumReduceTasks(0);


        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(NullWritable.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));

        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}
```

4) index.html

```
<!DOCTYPE html>

<html>

<head>

   <meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1">

   <title>Flight Data Analysis</title>

   <!-- include bootstrap -->

   <!-- Latest compiled and minified CSS -->

   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-
BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">

   <script src="https://code.jquery.com/jquery-3.2.1.min.js" integrity="sha256-
hwg4gsxgFZhOsEEamdOYGBf13FyQuiTwlAQgxVSNgt4="
crossorigin="anonymous"></script>

   <style>

      body{
```

```
  padding: 15px;
}

.col-md-4{
  border: 0px solid black;
  height: 450px;
}

.col-md-12{
  border: 1px solid lightblue;
  height: 150px;
}

#studentname{
  color: black;
  font-family: verdana;
  font-size: 90%;
  padding-left: 400px;
}

.myForm {
    font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
    font-size: 0.8em;
    padding: 1em;
    border: 1px solid #ccc;
```

```
border-radius: 3px;
}

.myForm * {
box-sizing: border-box;
}

.myForm label {
padding: 0;
font-weight: bold;
}

.myForm input {
border: 1px solid #ccc;
border-radius: 3px;
font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
font-size: 0.9em;
padding: 0.5em;
}

.myForm input[type="email"],
.myForm input[type="password"] {
width: 12em;
}
```

```
        .myForm button {
        padding: 0.7em;
        border-radius: 0.5em;
        background: #eee;
        border: none;
        font-weight: bold;
        }

        .myForm button:hover {
        background: #ccc;
        cursor: pointer;
        }

    </style>
</head>
<body>
  <nav class="navbar navbar-inverse">
  </nav>
  <div class="jumbotron">
      <h2>Flight Arrival Data Analysis </h2>
      <p>Airline on-time performance in United States of America. </p>
  </div>

  <div class="container-fluid">
    <div class="row">
```

```
    <div class="col-md-3">

      <a
href="https://public.tableau.com/profile/sri.surya#!/vizhome/AirlineCarrierDelays/Carrier
DelayDashBoard">

      <div class="thumbnail">

        <img src="5.jfif" alt="AirlineAnalysis">

        <div class="caption">

          <h3>Airline Carrier Delays</h3>

          <p>Best airlines to avoid delays and its analysis</p>

        </a>

        </div>

      </div>

    </div>

    <div class="col-md-3">

      <a
href="https://public.tableau.com/profile/sri.surya#!/vizhome/FlightCancellationAnalysis_1
5890442234280/CancellationDashboard">

      <div class="thumbnail">

        <img src="4.jpeg" alt="AirlineAnalysis">

        <div class="caption">

          <h3>Flight Cancellation Trends</h3>

          <p>Cancellation counts on reasons, monthly basis and airlines</p>

        </a>

        </div>

      </div>

    </div>

  </div>
```

```
  <div class="jumbotron">
     <p id="studentname"> Sri Surya  |  Vasavi College of Engineering </p>
  </div>
</body>
</html>
```

4.5 GitHub Link

https://github.com/Surya707/FlightDataAnalysis

4.6 Folder Structure

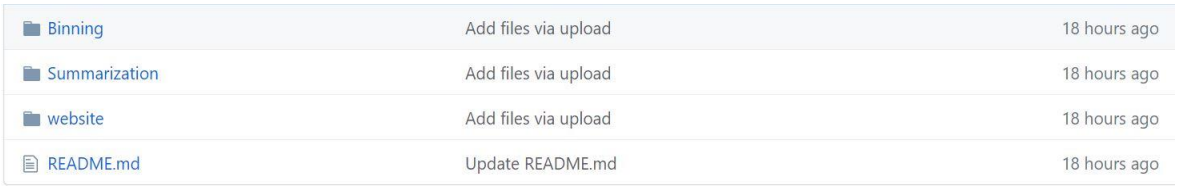

Fig 4.1 Folder Structure

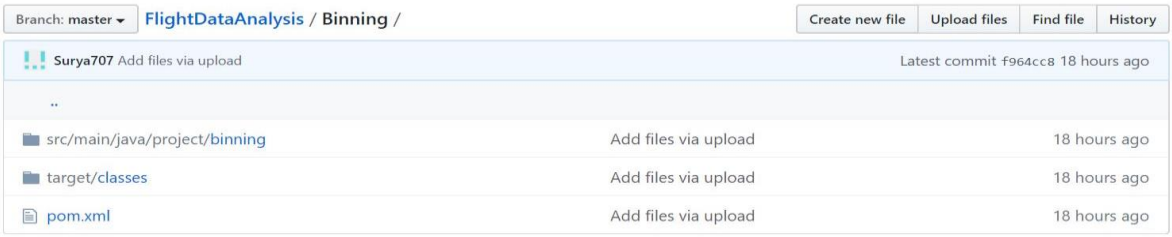

Fig 4.2 Binning File Structure

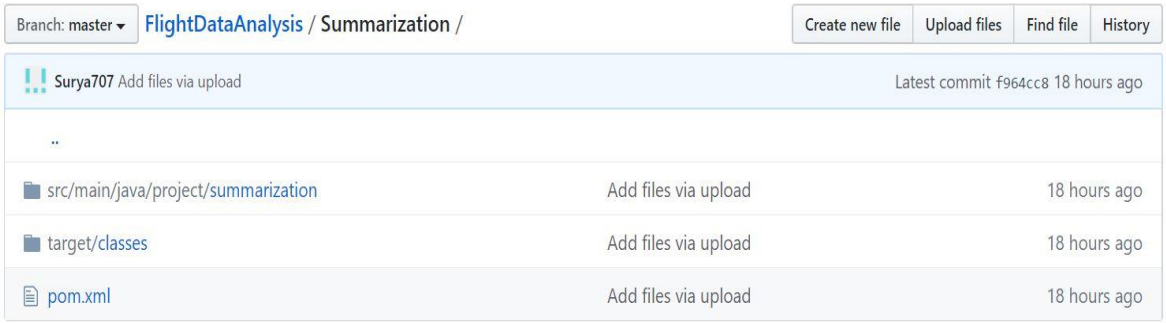

Fig 4.3 Summarization File Structure

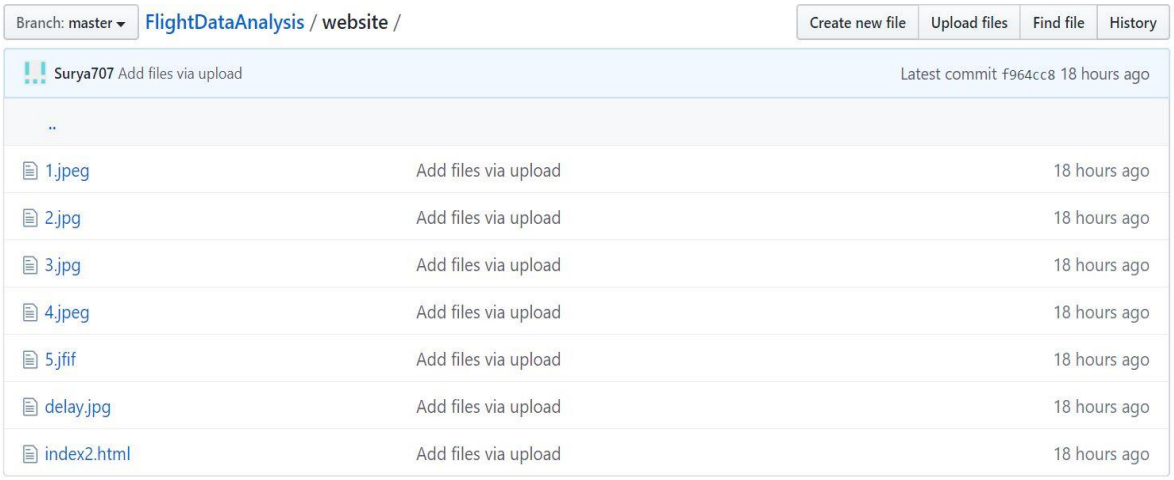

Fig 4.4 Website File Structure

Chapter – 5

Results

5.1 Raw Data:

1	C	149060
2	NA	1568795
3	D	601
4	A	317868
5	B	267000

Fig 5.1 Airlines Cancellation Reasons

1	AS	0.4536666	1275030.0	2810500.0
2	ML (1)	0.57275134	39588.0 69119.0	
3	PI	0.32256448	278118.0	862209.0
4	UA	0.51299477	6657681.0	12978068.0
5	EA	0.5185555	458443.0	884077.0
6	EV	0.5255399	864562.0	1645093.0
7	FL	0.5449398	680802.0	1249316.0
8	NW	0.5479055	5507922.0	10052686.0
9	TZ	0.58680046	120885.0	206007.0
10	B6	0.5479773	437898.0	799117.0
11	CO	0.52417624	4198218.0	8009173.0
12	DL	0.45730948	7436686.0	16261823
13	AA	0.5380722	7885584.0	14655251
14	F9	0.5189851	173778.0	334842.0
15	HA	0.7494274	204488.0	272859.0
16	OH	0.55846107	789731.0	1414120.0
17	OO	0.58839494	1777267.0	3020534.0
18	PS	0.4177755	34380.0 82293.0	
19	9E	0.60877526	307132.0	504508.0
20	HP	0.45224607	1616961.0	3575401.0
21	US	0.5034709	6924321.0	1375317.0
22	AQ	0.62106043	94095.0 151507.0	
23	DH	0.6090801	407893.0	669687.0
24	TW	0.50569713	1859941.0	3677974.0
25	WN	0.57442814	9072556.0	15794066
26	YV	0.5602707	460705.0	822290.0
27	MQ	0.55799395	2114715.0	3789853.0

Fig 5.2 Airlines Delayed Data

ACV	OUT	8
BET	OUT	6
BIL	IN	4
BQK	IN	17
BTM	IN	4
BUR	OUT	10
CLT	OUT	16
CMH	OUT	11
DBQ	IN	3
DRO	IN	4
GGG	OUT	8
GSO	OUT	13
GUC	IN	3
HPN	IN	6
ILM	IN	19
IYK	OUT	9
JAC	OUT	13
LAN	OUT	10
LBB	OUT	7
LGB	IN	5
LWS	IN	5
MAF	IN	2
MIB	OUT	6
MQT	OUT	11
MSO	OUT	10
OGG	IN	4
ORH	IN	3

Fig 5.3 Average Taxitime

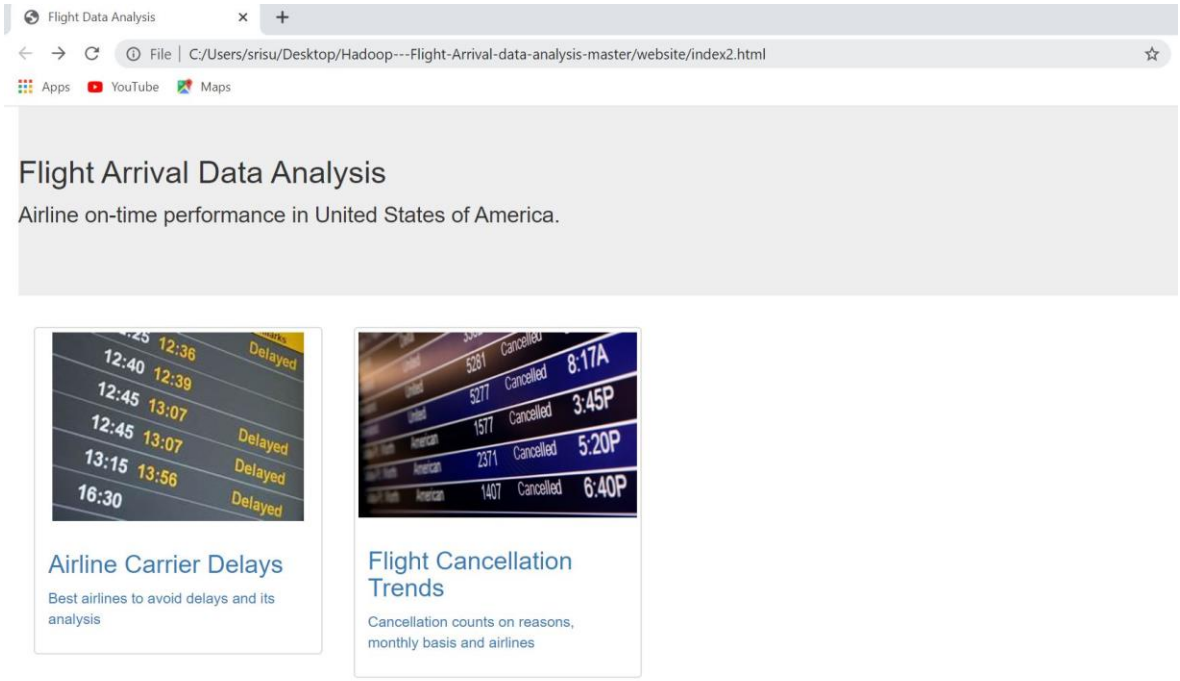

Fig 5.4 Website Home Page

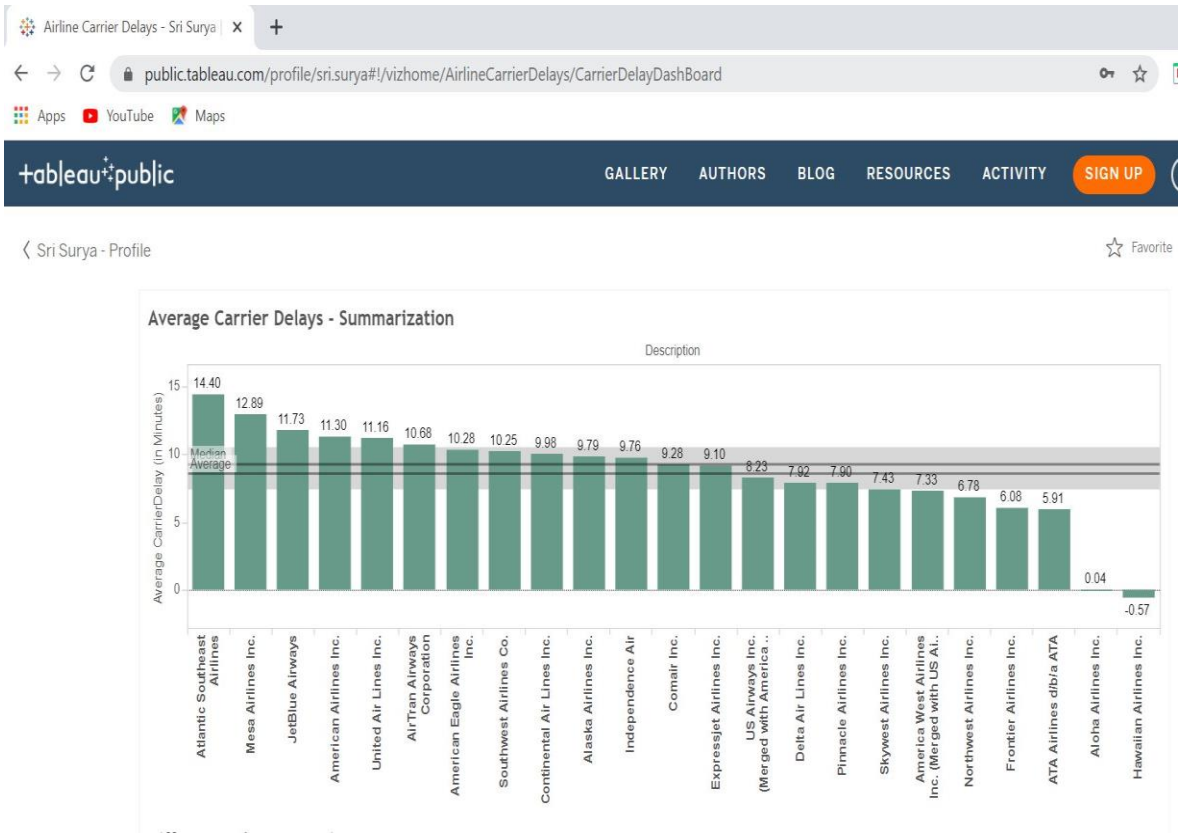

Fig 5.5 Average Carrier Delays Visualisation

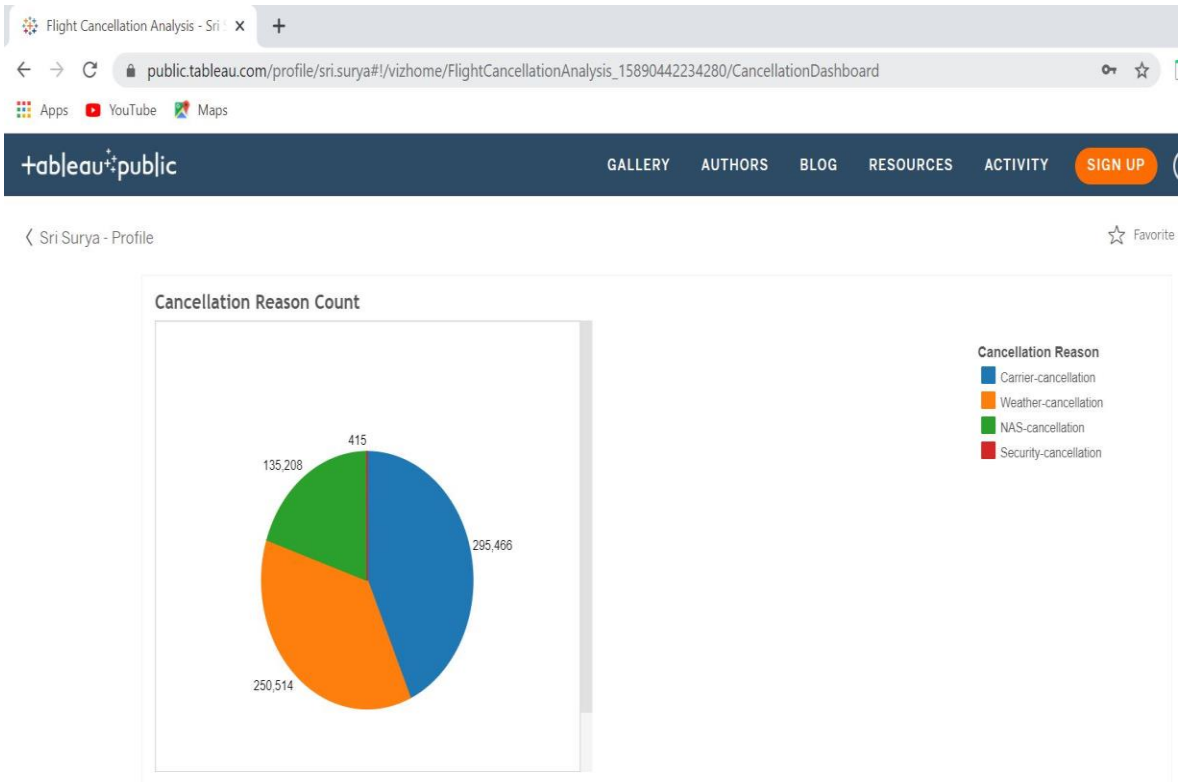

Fig 5.6 Cancellation Reason Count Visualisation

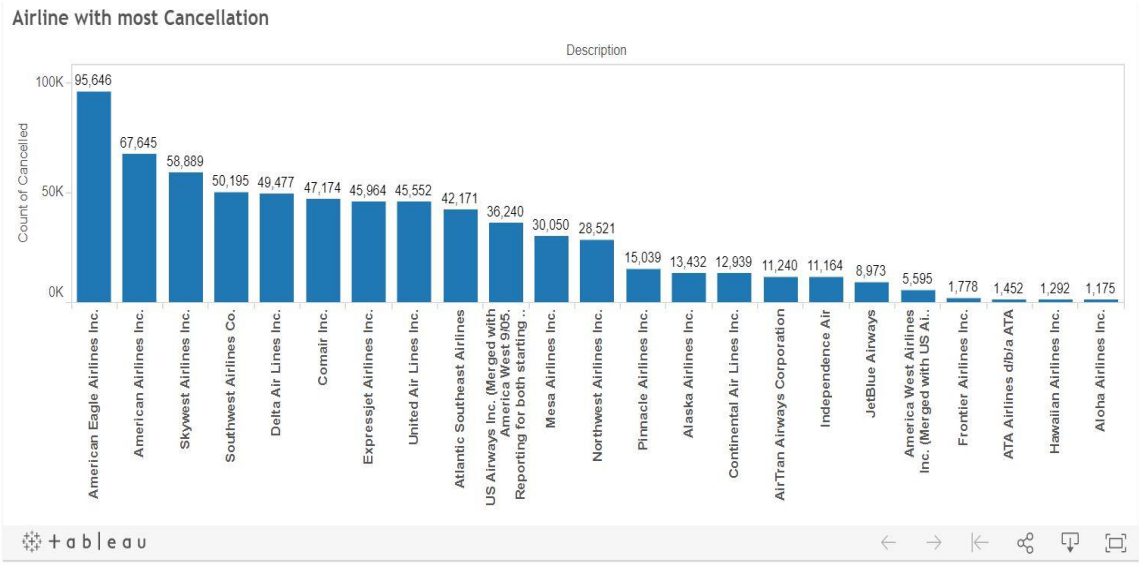

Fig 5.7 Cancellation Reason Visualisation

Chapter 6

Conclusion and Future Scope

6.1 Conclusion:

We have created a website through which we can analyse the flight data and how various delays are occurring in the field of Airlines. And our website is user friendly and user can understand without any effort.

6.2 Future Scope:

In future to develop our website and its accuracy we will add more mappers and reducers functions and we will work with large number of datasets than now.

Chapter 7

References:

Dataset: **http://stat-computing.org/dataexpo/2009/the-data.html**

UML creation: online.visual-paradigm.com

summarization patterns: www.oreilly.com