

## **ABSTRACT**

The title of our project is called Flight Delay Prediction System. The purpose of this project is to find out if a flight is getting delayed during departure and arrival then what are the reasons for the delay. Therefore we intend to aid the airlines by predicting the delays by using certain data patterns from the previous information. This project explores what factors influence the occurrence of flight delays.

The procedure involves using models to predict the delay. For example, one of the models is to estimate delays according to arrival schedule. The delay is predicted for domestic flights in United States of America. The dataset for the flights obtained from Transtats - Bureau of Transportation Statistics includes data about 306 airports.

Classification algorithm is applied to classify flights into delay categories. Using OneR, a classification algorithm, models are developed to predict delay on both arrival and departure side. Discretization was applied using Weka, a data mining tool, to divide the delays on departure and arrival side into five categories viz; Negligible, Insignificant, Nominal, Significant, Indefinite. The result thus obtained from these categories was further analyzed to predict overall reason for delay. The delay predicted can be due to Weather, Security, Carrier, National Aviation System (NAS) and Late Arrival.

The models further combine this result with Meteorological Terminal Aviation Routine Weather Report (METAR) to give the report of weather conditions at origin and destination airport. METAR is a format for reporting weather information used by pilots providing a pre-flight weather briefing. The current weather report provides the weather conditions at origin and destination.

We therefore present a technique which identifies relevant attributes for the classification into flight delay categories. Our experimental evaluation demonstrates that our technique is capable of detecting relevant patterns useful for flight delay classification.

The results of data analysis will suggest that flight delays follow certain patterns that distinguish them from on-time flights. We may also discover that fairly good predictions can be made on the basis on a few attributes. Classification can be used for analyzing future data trends. It is important that the classification is appropriate so that the data prediction is accurate.

## Contents

ABSTRACT .....	i
Contents .....	iii
List of Figures.....	v
List of Tables.....	vi
Abbreviations, Notations and Nomenclature.....	vii
Chapter 1 Introduction .....	1
Chapter 2 Literature Survey .....	4
Chapter 3 Report on Present Investigation .....	14
3.1    Software Requirement Specification Document (SRS) .....	14
3.1.1    Introduction .....	14
3.1.2    The Overall Description .....	15
3.1.3    Operating environment.....	17
3.1.4    External Interfaces Requirements .....	17
3.1.5    System Features .....	18
3.1.6    Other Non-Functional requirements .....	20
3.2    Project Design .....	22
3.2.1    Block Diagram.....	22
3.2.2    Data Flow Diagram (DFD).....	24
3.2.3    Information Package Diagram.....	28
3.2.4    Star Schema .....	29
3.3    Implementation.....	30
3.3.1    Data preprocessing.....	30
3.3.2    METAR Weather Reporting.....	36
3.3.3    Data Mining Algorithm.....	38
3.3.4    User Interface .....	47
3.4    Testing .....	55
Chapter 4 Results and Discussion.....	57
Chapter 5 Conclusion & Future Scope.....	64

Appendix .....	66
References .....	69
Acknowledgements .....	71

## List of Figures

Figure 2.1: Graph for Delay .....	5
Figure 2.2: DelayCast Prediction.....	9
Figure 2.3: DelayCast Prediction table .....	10
Figure 2.4: www.metarreader.com .....	11
Figure 2.5: Heat Map .....	12
Figure 3.1: Block Diagram.....	23
Figure 3.2: Level 0 DFD .....	24
Figure 3.3: Level 1 DFD .....	26
Figure 3.4: Level 2 DFD for create Model.....	27
Figure 3.5: Information package Diagram .....	28
Figure 3.6: Star Schema .....	29
Figure 3.7: METAR Raw String Format.....	36
Figure 3.8: Weather Information Format .....	37
Figure 3.9: OneR Model .....	41
Figure 3.10: Departure Delay Model.....	44
Figure 3.11: Arrival Delay Model .....	45
Figure 3.12: Workflow of Algorithm.....	46
Figure 3.13: View Graph.....	47
Figure 3.14: Pie Chart .....	48
Figure 3.15: Bar Graph .....	49
Figure 3.16: Normalized Bar Chart .....	50
Figure 3.17: Predict Delay.....	51
Figure 3.18: IATA-ICAO Mapper.....	51
Figure 3.19: Predict Delay 1.....	52
Figure 3.20: Predict Delay 2.....	53
Figure 3.21: Delay Report.....	54
Figure 3.22: View Result .....	54
Figure 4.1: Graph for Performance of Algorithms .....	58
Figure 4.2: Delay and Delay Reasons .....	59
Figure 4.3: Departure Delay Model.....	59
Figure 4.4: Arrival Delay Model .....	60
Figure 4.5: Comparison of Algorithm Performance .....	60
Figure 4.6: Experiments Vs Accuracy .....	63

## List of Tables

Table 3.1: Month wise distribution of records .....	31
Table 3.2: Combination of types of delay .....	33
Table 3.3: Distribution of Nominal & Numeric Attributes .....	34
Table 3.4: Performance of Algorithms .....	39
Table 3.5: Range for Departure & Arrival Delay .....	42
Table 4.1: Confusion Matrix .....	57
Table 4.2: Delay and Delay Types.....	61
Table 4.3: Departure Delay .....	62
Table 4.4: Arrival Delay.....	62

## **Abbreviations, Notations and Nomenclature**

### **A**

AWOS: Automated Weather Observing System

AWSS: Automated Weather Sensor System

ASOS: Automated Surface Observing System

### **B**

BTS: Bureau of Transportation Statistics

BOS-MCO: Boston-Moscow

### **C**

CRS

### **D**

DEPT : Departure

DFD : Data Flow Diagram

### **F**

FDPS: Flight Delay Prediction System

FAA :Federal aviation administration

### **G**

GDP: Ground delay program

### **I**

IATA: International Air Transportation Association

ICAO: International Civil Aviation Organization

### **J**

J48: Classification of nominal attributes

### **M**

METAR: Meteorological Terminal Aviation Routine Weather Report

## **N**

NAS: National Aviation System

## **O**

OTP: On-Time Performance

OD: origin-destination

## **R**

REP: Reduced Error Pruning

## **S**

SRS: Software requirement Specification

## **U**

UI: User Interface

## **W**

WITI: Weather Impacted Traffic Index

WEKA: Waikato Environment for Knowledge Analysis



## Chapter 1

### Introduction

A flight delay is a when an airline flight takes off and/or lands later than its scheduled time. The **Federal Aviation Administration** (FAA) considers a flight to be delayed when it is 15 minutes later than its scheduled time. A cancellation occurs when the airline does not operate the flight at all for a certain reason. Some of the causes of flight delays are as follows:

- Maintenance problems with the aircraft.
- Fueling.
- Extreme weather, such as tornado, hurricane, or blizzard.
- Airline glitches.
- Congestion in air traffic.
- Late arrival of the aircraft to be used for the flight from a previous flight.
- Security issues.

Flight delays are an inconvenience to passengers. A delayed flight can be costly to passengers by making them late to their personal scheduled events. A passenger who is delayed on a multi-plane trip could miss a connecting flight. Anger and frustration can occur in delayed passengers.

To refine proactive scheduling, we propose classification of flights into delay categories. Our method is based on archived data at major airports in current flight information systems. Classification in this scenario is hindered by the large number of attributes, which might occlude the dominant patterns of flight delays. We therefore present a technique which identifies locally relevant attributes for the classification into flight delay categories. We give an algorithm that efficiently identifies relevant attributes. Our experimental evaluation demonstrates that our technique is capable of detection relevant patterns useful for flight delay classification.

The results of data analysis will suggest that flight delays follow certain patterns that distinguish them from on-time flights. We may also discover that fairly good predictions can be made on the basis on a few attributes.

Classification and prediction can be used for analyzing future data trends. It is important that the classification is appropriate so that the data prediction is accurate. The regression model will estimate the probability of delay and the classification model classifies whether delay is likely to occur based on the input variables. Results of both the models perform prediction of delay

## **Chapter 2**

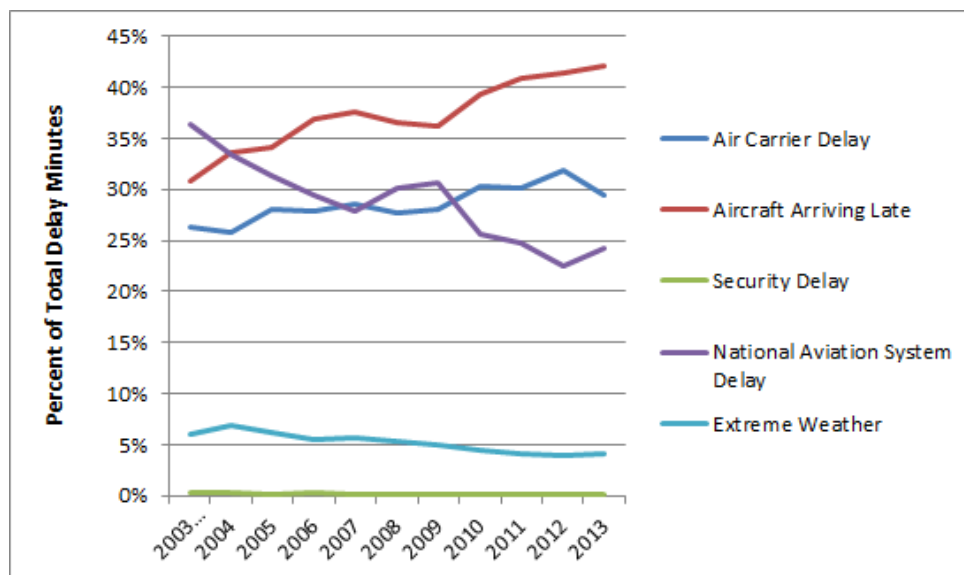
### **Literature Survey**

Flight delay is a complex phenomenon, because it can be due to problems at the origin airport, at the destination airport, or during airborne. A combination of these factors often occurs. Delays can sometimes also be attributable to airlines. Some flights are affected by reactionary delays, due to late arrival of previous flights. These reactionary delays can be aggravated by the schedule operation. Flight schedules are often subjected to irregularity. Due to the tight connection among airlines resources, delays could dramatically propagate over time and space unless the proper recovery actions are taken. Even if complex, flight delays are nowadays measurable. And there exist some pattern of flight delay due to the schedule performance and airline itself. The Bureau of Transportation Statistics (BTS) compiles delay data for the benefit of passengers. They define a delayed flight when the aircraft fails to release its parking brake less than 15 minutes after the scheduled departure time.

The FAA is more interested in delays indicating surface movement inefficiencies and will record a delay when an aircraft requires 15 minutes or longer over the standard taxi-out or taxi-in time. Generally, flight delays are the responsibility of the airline. Each airline has certain number of hourly arrivals and departures allotted per airport. If the airline is not able to get all of its scheduled flights in or out each hour, then representatives of the airline will determine which

flights to delay and which flights to cancel. These delays take one of three forms, ground delay programs, ground stops, and general airport delays. When the arrival demand of an airport is greater than the determined capacity of the airport, then a ground delay program may occur. Generally, ground delay programs are issued when inclement weather is expected to last for a significant period of time.

Second, ground stops are issued when inclement weather is expected for a short period of time or the weather at the airport is unacceptable for landing. Ground stops mean that traffic destined to the affected airport is not allowed to leave for a certain period of time. Lastly, there are general arrival and departure delays. This usually indicates that arrival traffic is doing airborne holding or departing traffic is experiencing longer than normal taxi times or holding at the gate. These could be due to a number of reasons, including thunderstorms in the area, a high departure demand, or a runway change. Our research finds that arrival and departure delays are highly correlated. Correlation between arrival and departure delays is extremely high (around 0.9). This finding is useful to prove that congestion at destination airport is to a great extent originated at the departure airport.



**Figure 2.1: Graph for Delay**

## POPULAR IMPLEMENTATION

### 1) **KnowDelay.com predicts flight problems 3 days in advance**

*By Rob Lovitt*

KnowDelay.com <sup>[10]</sup> crunches weather, airline and airport data to predict weather-related flight delays up to three days in advance. Weather delays are the most stubborn and intractable delays because oftentimes it can take travelers a day or more to get out. “But if you can say three days in advance, don’t connect over airport XYZ, change your flight and connect over airport ABC, you may be able to avoid the problem.”

KnowDelay users can view a map with colored dots — red, yellow and green — and a slider that lets them see forecasted delays over the next 72 hours. Red means there’s 60-percent chance of a 60-minute weather delay; yellow equals a 40-percent chance of a 30-minute delay, and green means a 6-percent chance of delay. During two years of beta testing, the site has accurately predicted 90 percent of weather-related delays. It now covers 37 U.S. airports, including major hubs and other destinations that are typically affected by bad weather.

### 2) **How FlightCaster Squeezes Predictions from Flight Data**

FlightCaster <sup>[11]</sup> predicts flight delays using an advanced algorithm that scours data on every domestic flight for the past 10-years and matches it to real-time conditions. They help you evaluate alternative options and help connect you to the right person to make the change. FlightCaster uses data from:

- Bureau of Transportation Statistics
- FAA Air Traffic Control System Command Center
- FlightStats
- National Weather Service

### 3) **Characterization and Prediction of Air Traffic Delays**

*Juan Jose Rebollo and Hamsa Balakrishna*

*Massachusetts Institute of Technology <sup>[31]</sup>*

This paper presents a new class of models for predicting air traffic delays. The proposed models consider both temporal and spatial (that is, network) delay states as explanatory variables, and use Random Forest algorithms to predict departure delays 2-24 hours in the future. The paper analyzes the performance of the proposed prediction models in both classifying delays as above or below a certain threshold, as well as predicting delay values. Delay prediction has been the topic of several previous efforts.

Jetzki(2009) studied the propagation of delays in Europe, with the goal of identifying the main delay sources. Tu et al. (2008) developed a model for estimating flight departure delay distributions, and used the estimated delay information in a strategic departure delay prediction model.

By contrast, Bratu and Barnhart (2005) focused on the impact of delays on passengers. Other prediction models (Klein et al. 2007, 2010, Sridhar and Chen 2009) have focused on weather-related delays, and the development of a Weather Impacted Traffic Index (WITI). By contrast, the goal of this paper is to evaluate the potential of network-scale delay dependencies in developing delay prediction models. The models presented in this paper therefore attempt to predict future departure delays on a particular origin-destination (OD) pair by considering current and/or past delays in the network.

The main objective of this paper is to predict the departure delay on a particular link or at a particular airport, sometime in the future. The departure delay of a link at time  $t$  is an estimate of the departure delay of any flight(s) taking off at time  $t$ , and flying on that link. For example, if the BOS-MCO departure delay state two hours from now is estimated to be 30 minutes, it means that the estimated departure delay for BOS-MCO flights taking off two hours from now is 30 minutes. Two types of prediction mechanisms are considered: classification, where the output is a

binary prediction of whether the departure delay is more or less than a predefined threshold, and regression, where the continuous output is an estimate of the departure delay along the link.

Ten training sets (3,000 points each) and ten test sets (1,000 points each) were sampled from the 2007-2008 dataset. The prediction models were fit and tested for each of the 10 training and test set pairs, respectively, providing measures of the variability and test error. The training and test sets were over-sampled from the 2007-2008 data. Over-sampling is the over selection of samples of the minority class in order to achieve balanced training and test datasets with sufficient representatives of both the majority and minority classes (Upton and Cook 2008). Since the majority of links do not experience delays of more than 60 minutes, a naive classification algorithm that predicts no delays of more than an hour would be correct most of the time. For this reason, the true evaluation of a classifier's performance is its ability to correctly predict delays in a balanced data set in which half the points have delays of less than 60 minutes (the so-called "majority class"), and half the points have delays of more than 60 minutes (the "minority Class"). Different classification and regression models (logistic regression, single classification trees, bagging, boosting, linear regression, neural nets and random forests) were tested, and Random Forests were chosen due to their superior performance.

#### **4) Decision Support Tool for Predicting Aircraft Arrival Rates, Ground Delay Programs, and Airport Delays from Weather Forecasts**

*David A. Smith & Dr. Lance Sherry <sup>[5]</sup>*

The paper shows the possibilities of a Weather Delay Prediction Tool and what it can do to help NAS stakeholders. The algorithm is capable of classifying weather forecasts into three sets, where each set represents a specific AAR. The principle bottlenecks of the air traffic control system are major airports Atlanta, Detroit, St. Louis, Minneapolis, Newark, Philadelphia, and LaGuardia all expect to be at least 98% capacity by 2012.

The general procedure used to determine a connection between weather forecast and airport capacity was:

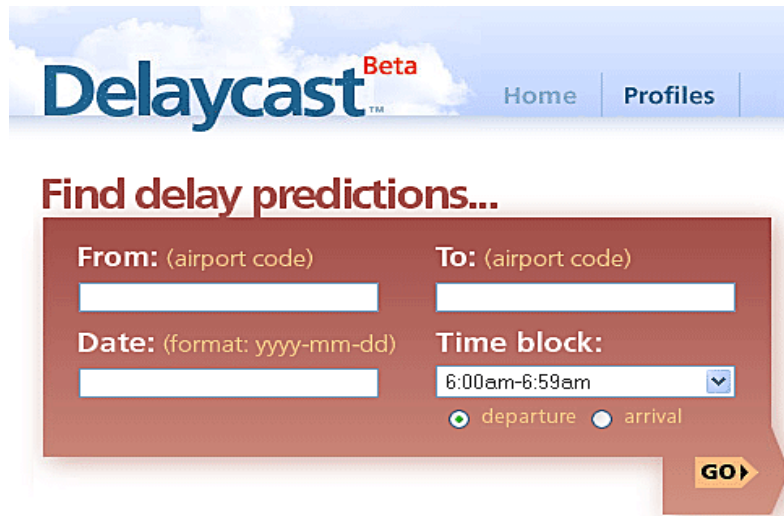
- Collect data from the various available data sources,
- Using assorted tools, format the data into a usable layout,
- use a classification tool to connect the two sets, and
- test the data to ensure there is a correlation.

Weather forecast products are uncertain and the uncertainty increases with lead-time. Useful applications of weather forecasts requires either refinement, consultation, and application of the weather forecast to estimate air traffic capacity or decision support tools that take forecasts and make predictions based on past forecasts and those forecasts connections to NAS capacity. This paper describes a methodology used to create one such decision support tool known as the Weather Delay Prediction Tool.

## **5) DelayCast.com**

DelayCast <sup>[21]</sup> is a website that helps you obtain reasonable estimates of the flight delays you may experience. These estimates are based on factors such as the airline, flight origin and destination, developing trends, holidays, date and time of the flight and so on. Apart from estimated predictions you can also use it to check more general overview of the best days, times and airlines to fly.





The screenshot shows the Delaycast Beta website interface. At the top, the logo "Delaycast" is displayed in blue with "Beta" in red to its right. Navigation links for "Home" and "Profiles" are visible. Below the header, a red banner reads "Find delay predictions...". The main search form is a red box with the following fields: "From: (airport code)" and "To: (airport code)" each with a white input box; "Date: (format: yyyy-mm-dd)" with a white input box; and "Time block:" with a dropdown menu showing "6:00am-6:59am". Below the time block, there are two radio buttons: "departure" (selected) and "arrival". A yellow "GO" button with a right-pointing arrow is located at the bottom right of the form.

**Figure 2.2: DelayCast Prediction**

**Features:**

- Check out flight delay estimates before booking a flight.
- Airports: Departure and arrival predictions made for the top 60 airports in the United States.
- Currently supported airlines: Southwest, Northwest, JetBlue, America, American, Eagle, Continental, America, West, Delta, US Airways, Alaska Airlines, United and AirTran.
- Free. No-registration required.

AIRLINE	PREDICTION	90%
<b>Chance of departing on time:</b>		
American	<b>75%</b>	<b>67-82</b>
JetBlue	<b>71%</b>	<b>63-78</b>
Delta	<b>60%</b>	<b>49-70</b>
<b>Expected departure delay:</b>		
American	<b>11 minutes</b>	<b>9-15</b>
JetBlue	<b>12 minutes</b>	<b>10-16</b>
Delta	<b>13 minutes</b>	<b>10-18</b>

**Figure 2.3: DelayCast Prediction table**

**6) METAR Reader(metarreader.com)**

METAR Reader is a website that helps the user to retrieve weather information in METAR string format by just giving the four-letter ICAO Code as the input. This fetches the current weather information of that region code. The information in the METAR string contains time, date, temperature, wind, visibility, sky and cloud conditions, weather behavior and additional remarks. Apart from this user can also convert the METAR string into a simple English language format.

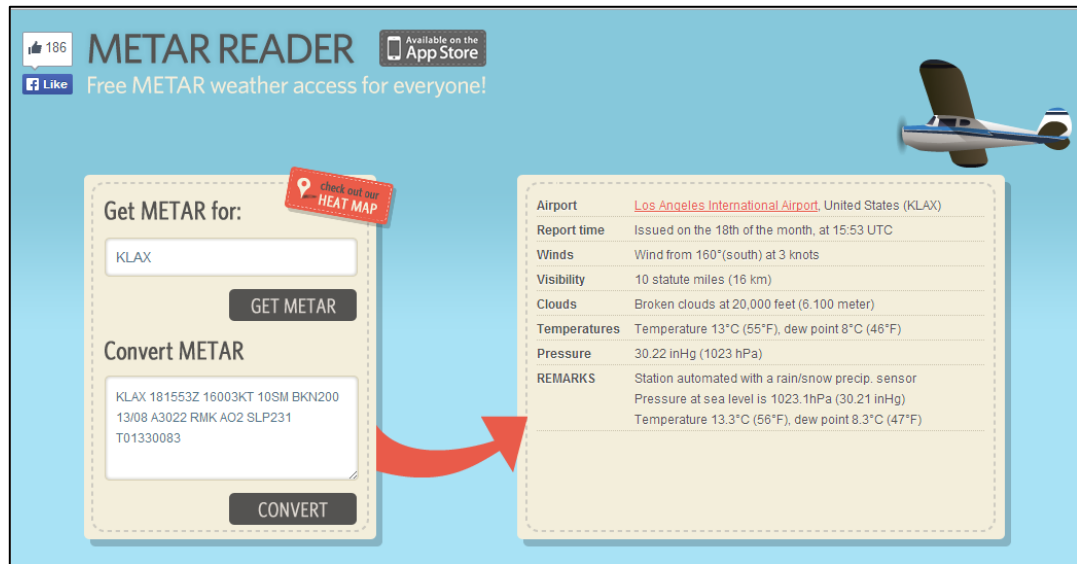
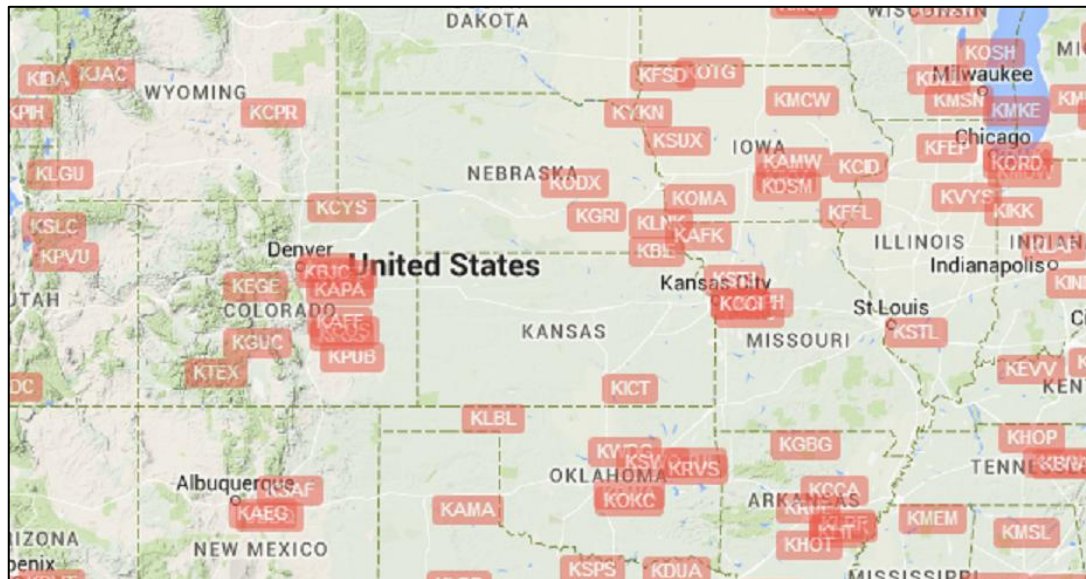


Figure 2.4: [www.metarreader.com](http://www.metarreader.com)

**Features:**

- The site has an option for viewing the map called as Heat Map, where user can have a topographical view of map along with the regions ICAO Code link. On opening any link user retrieves the current weather information for that region and weather predictions for coming week.



**Figure 2.5: Heat Map**

Statistical models and simulation method are used to analyze flight delay. But we can see that the analysis on delay is carried on data with only a few days. That is because of the huge data of flights every day. So here the flight delay is categorized into several levels, and the logistic regression models are used here to better identify the delay pattern. Studies on airport delay and delay influence on individual flight are carried out using regression model and classifiers.

## **Chapter 3**

### **Report on Present Investigation**

This section comprises of 2 parts. Firstly, Software Requirement Specification (SRS) document on the project has been elaborated (3.1). Next part includes diagrams (according to the structured approach), followed by System Interface design (3.2).

### **3.1 Software Requirement Specification Document (SRS)**

#### **3.1.1 Introduction**

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS.

##### ***3.1.1.1 Purpose***

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Flight delay prediction System (FDPS). This SRS will allow for a complete understanding of what is to be expected of the system to be constructed. The clear understanding of the system and its' functionality will allow for the correct

software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, the FDPS can be designed, constructed, and finally tested. This SRS can be used to fully understand the expectations of this FDPS to construct the appropriate software.

#### ***3.1.1.2 Scope***

The Flight delay prediction system is an desktop based application which includes user interaction. The project will help in predicting the occurrence of ground delay which can help the Aviation Administration traffic managers and airline dispatchers prepare mitigation strategies to reduce impact. The regression model will estimate the probability of delay and the classification model classifies whether delay is likely to occur based on the input variables. Results of both the models perform prediction of delay.

#### ***3.1.1.3 Intended Audience***

Administrative managers at the airport

#### ***3.1.1.4 Definitions, Acronyms and Abbreviations***

The definitions of the terms, which are used in this SRS document, are shown below

GUI	Graphical User Interfaces
SRS	Software Requirement Specification
FDPS	Flight delay prediction system
DBMS	Data Base Management System
GDP	Ground delay program
OTP	On-time Performance

#### ***3.1.1.5 References***

- [1] IEEE STD 1233-1998, IEEE Guide for Developing System Requirements Specifications.

[2] IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

### 3.1.2 The Overall Description

This section describes the general factors that affect the product and its requirements. It does not state specific requirements. Instead it provides a background for those requirements and makes them easier to understand.

#### 3.1.2.1 Product Perspective

The FDPS is an independent stand-alone system. It is totally self-contained.

#### 3.1.2.2 Product Functions

The use cases of FDPS are explained:

- **Choose Details:** The user can choose the fields from the dropdown box available in the GUI for analyzing the delay pattern.
- **Analyze delays:** The user can view the various graphs and charts generated as the output.
- **Generate Reports:** The user can generate reports, save a copy and print it as well.
- **Submit Details:** The user is required to submit the flight details for which the delay is to be predicted. Details like Origin, Destination, Carrier, Date, Time, etc of the flight needs to be selected.
- **Predict Delays:** These flight details are submitted to model and is classified as OTP or delayed. If delayed then the numeric probability prediction is made as late, very late, etc.
- **Update Flight Schedule:** The new flight schedule can be updated.

### ***3.1.2.3 User Characteristics***

The administrative staff using this software should have basic knowledge of analyzing graphs and charts. We intend to provide approximate prediction of the flight delays which will be helpful for the passengers as well as aviation management team.

## **3.1.3 Operating environment**

### ***3.1.3.1 Design and implementation constraints***

FDPS requires dataset to build a model for future trend prediction based on the past data. So, the authenticity of data is an important constraint. Also it requires real time weather information input which may or may not be perfectly accurate. These constraints directly affect the accuracy of the prediction. Thus, FDPS has reliability and accuracy constraints but we can overcome them.

### ***3.1.3.2 Assumptions and dependencies***

User interface and some functionality can change during the development process of project. New functionalities can be added which can change the dependent system requirements.

## **3.1.4 External Interfaces Requirements**

This section contains all the software requirements at a level of detail, that when combined with the system context diagram, use cases, and use case descriptions, is sufficient to enable designers to design a system to satisfy those requirements.

### ***3.1.4.1 User Interfaces***

This software is developed for administrative staff at the airport. Product will be deployed as desktop application. Each member of the airport administration team will



have unique username password. Application will have interface for selection of the flight details based on which the prediction will be made using the model.

#### ***3.1.4.2 Software Interfaces***

Flight delay intelligent prediction system consists of airport data acquisition front-end computer, database server, application server, master application, client-side equipments and network communication systems. Application server acquires real-time operating data of airport through airport data acquisition front-end computer and stores them in the database server. During the assessment of prediction models, application server accesses database server to get historical flights operation data. Then these data are converted and presented to the user in the form of chart.

#### ***3.1.4.3 Hardware Interfaces***

The system shall run on a Microsoft Windows based system.

#### ***3.1.4.4 Communication Interfaces***

The system shall use the real time weather information obtained from the web.

### **3.1.5 System Features**

#### ***3.1.5.1 Choose Details***

##### ***3.1.5.1.1 Description***

The user can select the flight details available in the dropdown box. Once the fields are selected, user can view the graph of the performance accordingly.

##### ***3.1.5.1.2 Validity Checks***

- Combo box fields should be entered.
- If any one of the fields is missing, invalid message will appear.

##### ***3.1.5.1.3 Stimulus or response sequence***

- User selects the fields of flight details.
- User clicks on the view output button.

### ***3.1.5.2 Analyze Delays***

#### ***3.1.5.2.1 Description***

The user can view the graphical output generated based on the past performance of the flights. User can analyze the delays, Flights performance and reasons for various delays.

#### ***3.1.5.2.2 Validity Checks***

- Fields for which output to be generated is entered.

#### ***3.1.5.2.3 Stimulus or response sequence***

- Flight details are submitted.
- Graphical output in the form of pie chart is generated.

### ***3.1.5.3 Generate Reports***

#### ***3.1.5.3.1 Description***

The user can view the graphical output generated, save the reports as well as print it if required.

#### ***3.1.5.3.2 Validity Checks***

- One needs to specify location for saving reports.

#### ***3.1.5.3.3 Stimulus or response sequence***

Graphical output generated can be saved in the system for future reference. The user just needs to specify location for saving report or click on the print button.

#### ***3.1.5.4 Submit details***

##### ***3.1.5.4.1 Description***

The user needs to enter the flight details for which the delay prediction is to be made. The user selects the flight from the schedule and passes the data to the model for delay prediction.

##### ***3.1.5.4.2 Validity Checks***

User must specify all the fields present.

##### ***3.1.5.4.3 Stimulus or response sequence***

User must provide specific flight details for the delay prediction. Users then click on predict delay option to generate the delay prediction report.

#### ***3.1.5.5 Predict delay***

##### ***3.1.5.5.1 Description***

The user when submits the details for the flight the delay prediction report is generated. Delay prediction is made on the basis of the classification model and numeric prediction using the regression model. The weather information input is also taken into consideration while predicting delay. Using Naïve Bayesian, we plan to classify the flights as Delay or On-Time. The algorithm takes a target attribute and predicts delay based on it.

##### ***3.1.5.5.2 Validity Checks***

- User must submit all details correctly.

##### ***3.1.5.5.3 Stimulus or response sequence***

User clicks on predict delay option to generate the delay prediction report. This report includes the delay prediction in minutes as well as the reason for the delay.

### ***3.1.5.6 Update flight schedule***

#### ***3.1.5.6.1 Description***

The user can update the flight details, timings and date. This helps the user to make proper selection at the time of flight delay prediction.

#### ***3.1.5.6.2 Validity Checks***

User must specify the record to be updated or added.

#### ***3.1.5.6.3 Stimulus or response sequence***

User can add a new flight detail as well as update the timings of an existing flight.

## **3.1.6 Other Non-Functional requirements**

### ***3.1.6.1 Performance Requirements***

Performance requirements define acceptable response times for system functionality.

- The load time for user interface screens shall take less time.
- Queries shall return results fast.

### ***3.1.6.2 Safety:***

Data will not be lost in case of system breakdown

### ***3.1.6.3 Security:***

Data reports can be stored secretly as it requires password credentials

### ***3.1.6.4 Reliability:***

Specify the factors required to establish the required reliability of the software system at time of delivery.

***3.1.6.5 Availability:***

The system shall be available at all times.

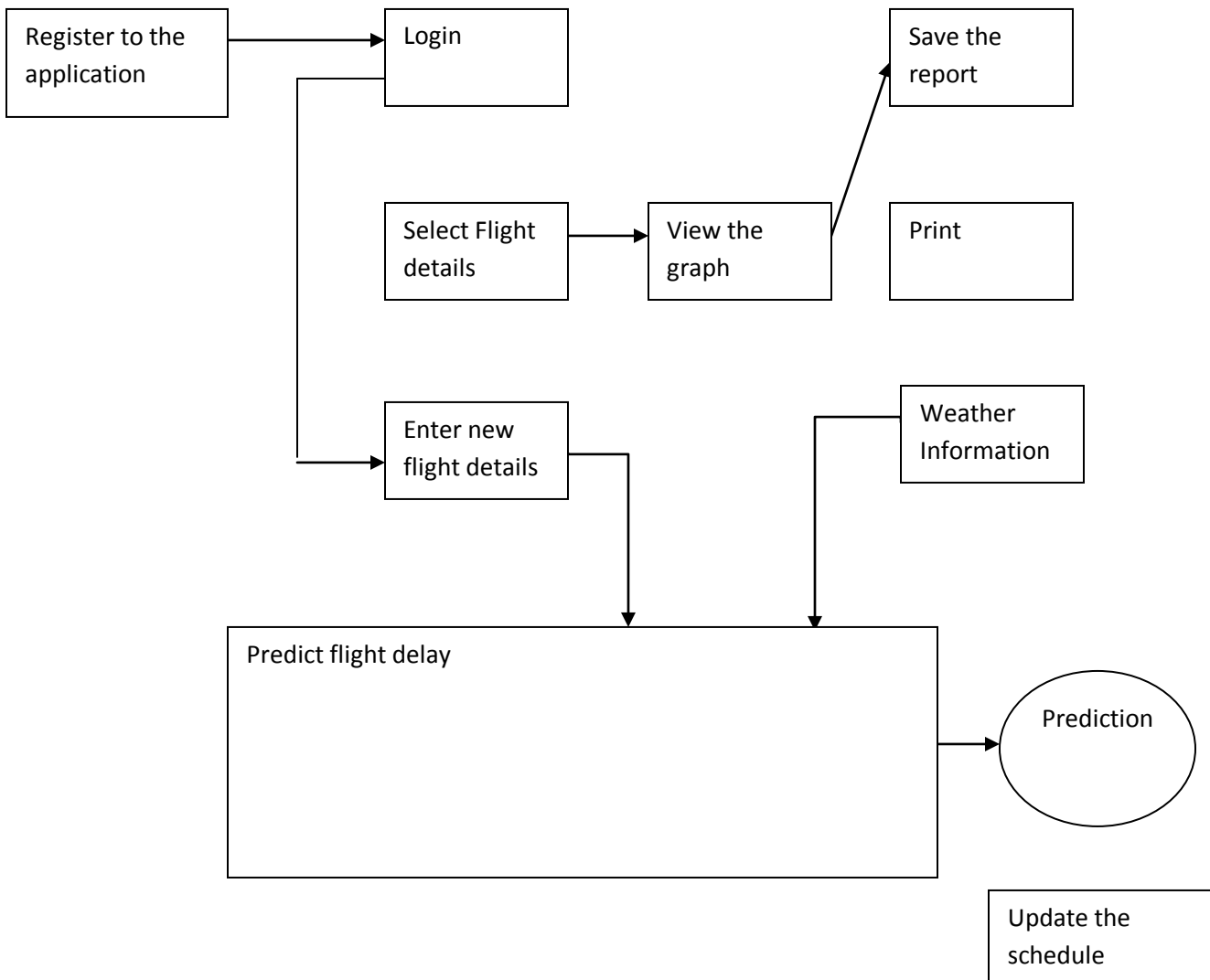
## 3.2 Project Design

### 3.2.1 Block Diagram

In order to use the system the user has to register to the application to generate his/her login credential. Once the user has logged into the system using his/her credentials, he/she can select and view the details of any particular flight and view the graph for the same. Since the user logs in using his/her credentials, he/she can save the report under his/her credential. The user enters details of flight into system through the GUI provided which is given to the Predict Flight Delay module. The Predict Flight Delay module consists of sub modules:

- Extract training data: The prime attributes are extracted from the given details.
- Classification Algorithm: Using classification algorithm,, we plan to classify the flights as Delay or On-Time. The algorithm takes a target attribute and predicts delay based on it.

Based on the prediction the user can update the schedule of the flights. The following design outline displays the work flow of the application.

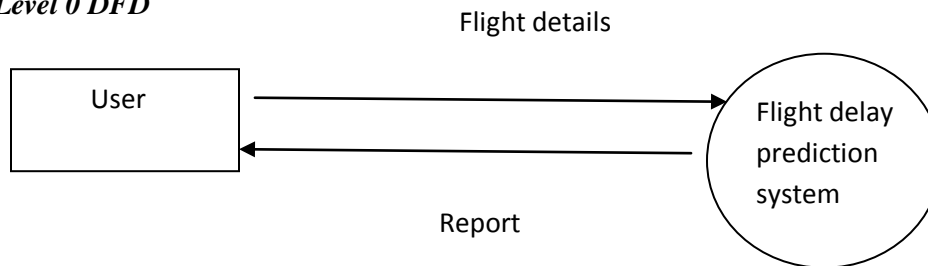


**Figure 3.1: Block Diagram**

### 3.2.2 Data Flow Diagram (DFD)

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. DFDs can also be used for the visualization of data processing (structured design). DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

1) **Level 0 DFD**



**Figure 3.2: Level 0 DFD**

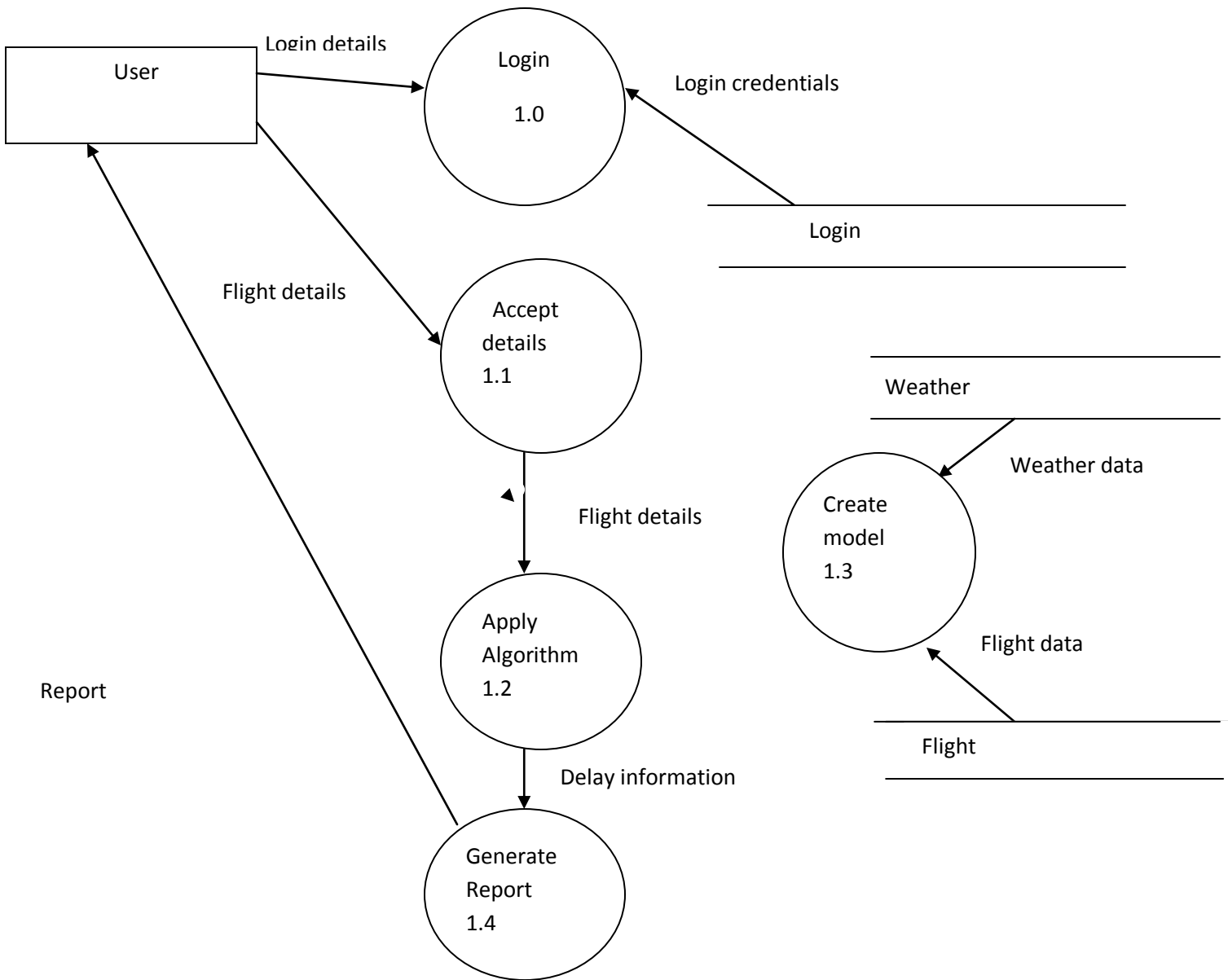
Level 0 diagram show the process that comprises the overall system. Following figure shows Flight Delay Prediction system as a process on the whole which takes the flight details from the user and gives the delay prediction in the form of report.



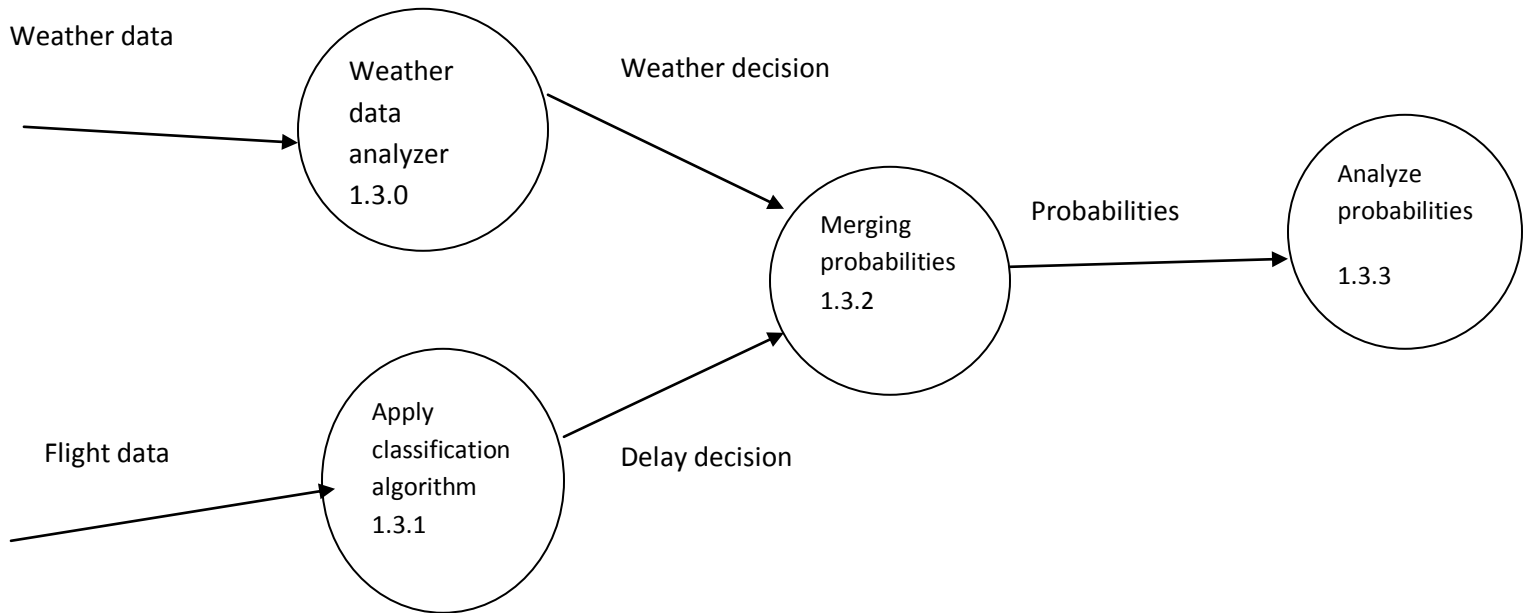
## 2) *Level 1 DFD*

The labeled arrows represent data objects or data object type hierarchies. Level 1 diagram shows all the processes that comprise a single process on the level 0 diagram. It also shows how information moves from and to each of these processes and more detail the content of higher level process.

The level 0 DFD is now expanded into a level 1 model. The User (administrative manager) login the system with username and password. System verifies the user login details with login credentials. System accepts the flight details like Origin, Destination, Carrier, Date, Time, etc. which user chooses for analyzing the delay. The Weather data (on-time) and flight data (data from flight dataset) are submitted to the model. The result as analyzed data from the model is applied to various algorithms along with flight details specified by the user to get the delay information. Report is generated based on the delay information. Report is submitted to the user.

**Figure 3.3: Level 1 DFD**

### 3) *Level 2 DFD*



**Figure 3.4: Level 2 DFD for create Model**

The processes represented at DFD level 1 can be further refined into lower levels. Weather data is applied to weather data analyzed to sort the given data and get the proper weather decision. Real-time weather data is analyzed and interpreted to get the weather decision in terms whether it is suitable for a flight or may cause delay. Flight data is extracted from the data set to get the relevant information related to delay.

Classification algorithms are applied on this data to get appropriate delay decisions. It is used for creating rules and classifying the flights. The probabilities of weather data and flight data are merged to get resultant probability which is analyzed further to get right information about delays.

### 3.2.3 Information Package Diagram

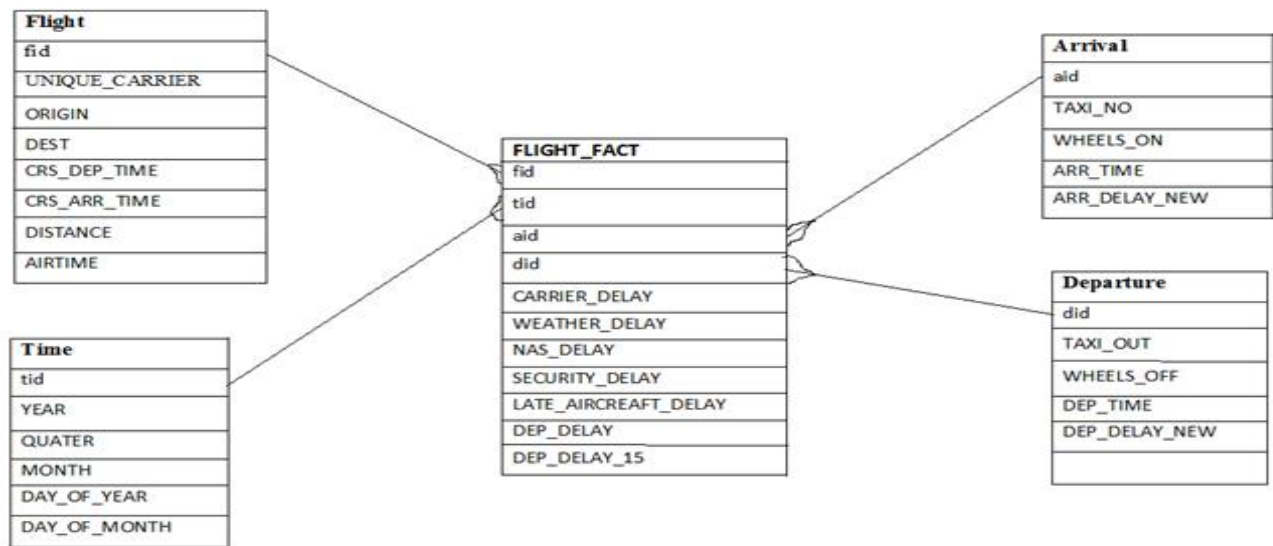
Information Package Diagram is the novel idea for determining and recording information requirements for a data warehouse. The relevant dimension and measurements in that dimension are captured and kept in a data warehouse. This creates an information package for a specific subject. Along the X axis are the dimensions and along Y axis are Hierarchies. The business measures are placed in bottom box labeled Facts these measures need to relate to all of the dimensions defined.

<b>Time</b>	<b>Flight</b>	<b>Arrival</b>	<b>Departure</b>
Tid	Fid	Aid	Did
YEAR	UNIQUE_CARRIER	TAXI_NO	TAXI_OUT
QUARTER	ORIGIN	WHEELS_ON	WHEELS_OFF
MONTH	DEST	ARR_TIME	DEP_TIME
DAY_OF_YEAR	CRS_DEP_TIME	ARR_DELAY_NEW	DEP_DELAY_NEW
DAY_OF_MONTH	CRS_ARR_TIME		
	DISTANCE		
	AIRTIME		
<b>Fact:</b> CARRIER_DELAY,WEATHER_DELAY,NAS_DELAY,SECURITY_DELAY,LATE_AIRCRAFT_DELAY,DEP_DELAY_			

**Figure 3.5: Information package Diagram**

### 3.2.4 Star Schema

In star schema each dimension is represented with only one dimension table. This dimension table contains the set of attributes. In the following diagram we have shown the data of flight with respect to the four dimensions namely, time, flight, departure and arrival. There is a fact



**Figure 3.6: Star Schema**

table at the center. This fact table contains the keys to each of four dimensions. The fact table also contain the attributes namely, CARRIER\_DELAY ,WEATHER\_DELAY, NAS\_DELAY, SECURITY\_DELAY, LATE\_AIRCRAFT\_DELAY, DEP\_DELAY\_15.

Each dimension has only one dimension table and each table holds a set of attributes. For example the Flight dimension table contains the attribute set {fid, UNIQUE\_CARRIER, ORIGIN,DEST, CRS\_DEP\_TIME, CRS\_ARR\_TIME, DISTANCE, AIRTIME}.

### 3.3 Implementation

In this project, the used data originally is from the Bureau of Transportation Statistics (BTS) to analyze and predict flight departure delays for commercial flights in the United States.

We have identified following factors:

- Factors which cause flight delay.
- Predict whether the flight will be delayed.
- By how many minutes the delay has caused.

#### 3.3.1 Data preprocessing

Data pre-processing includes data cleaning, data transformation and data reduction. The original dataset contains information for all commercial flights in the United States in the year 2014. A reasonable number of records are extracted from the original dataset reducing the size of dataset from 60 lacs to 6 lacs records approximately.

##### Data description

Originally the dataset had 38 attributes with approximately 60 lakh records. The records have been taken for the year of 2014. Roughly 5 lacs records about the airlines have been considered from each month.

The January-December 2014 records have been aggregated to give 60 lac records approximately with 38 attributes. Filtering of the entries was done by reducing instances using StratifiedRemoveFolds in WEKA. StratifiedRemoveFolds filter is applied to remove the attributes affecting the departure delay. This filter takes a dataset and outputs a specified fold for cross validation. This reduced the number of records to 6 lacs in the dataset. Also records with missing value labeled as “?” were removed using Remove Duplicates which is formatting to find, highlight and remove duplicate values and retain only unique values.

Month wise distributions of the number of records are as follows:

**Table 3.1: Month wise distribution of records**

Month	Number of records
January	509520
February	469747
March	552313
April	536394
May	542267
June	534451
July	571624
August	562922
September	193182
October	535345
November	522517
December	516740

For simplification purpose, the flights with uncommon attributes were removed. The original dataset also contained a large number of attributes and many of these were discarded because they were irrelevant or repeated information that could be found in other attributes. Attributes like diversion, cancellation have been discarded. As FDPS mainly focuses on departure delay, irrelevant attributes were removed and a new data set was formed with 20 attributes. After resampling, dataset contains around 590951 records, which is used for analysis. On removal of outliers the final dataset contains 590889 records.

List of attributes after filtering:

- Quarter
- Month
- Day\_of\_Month
- Day\_of\_week
- Unique\_Carrier
- Origin
- Dest

- Dep\_delay\_new
- Arr\_delay\_new
- CRS\_Dep\_time
- Dep\_Time
- CRS\_Arr\_time
- Arr\_Time
- Air\_time
- Distance
- Reason\_delay

The various types of delay have been converted to format of 0s and 1s. The attributes which do not cause delay have been labelled “0” and ones causing delay have been labelled “1”. The original format consisted delay in minutes now replaced by “1”. A combination of these types formed a new attribute “Reason\_delay”. An abbreviated form of delay has been taken for Reason\_delay attribute.

The delays have been labeled as follows:

C: CARRIER DELAY

W: WEATHER DELAY

N: NAS DELAY

S: SECURITY DELAY

L: LATE AIRCRAFT DELAY

If CWNSL=”00000” then Reason\_delay=”0” stating “No delay has taken place”.

If CWNSL=”10011” then Reason\_delay=”CSL” stating “Delay has taken place due to Carrier, Security and Late Aircraft”.



**Table 3.2: Combination of types of delay**

<b>C</b>	<b>W</b>	<b>N</b>	<b>S</b>	<b>L</b>	<b>Reason Delay</b>
0	0	0	0	0	O
0	0	0	0	1	L
0	0	0	1	0	S
0	0	0	1	1	SL
0	0	1	0	0	N
0	0	1	0	1	NL
0	0	1	1	0	NS
0	0	1	1	1	NSL
0	1	0	0	0	W
0	1	0	0	1	WL
0	1	1	0	0	WN
0	1	1	0	1	WNL
0	1	1	1	0	WNS
1	0	0	0	0	C
1	0	0	0	1	CL
1	0	0	1	0	CS
1	0	0	1	1	CSL
1	0	1	0	0	CN
1	0	1	0	1	CNL
1	0	1	1	0	CNS
1	0	1	1	1	CNSL
1	1	0	0	0	CW
1	1	0	0	1	CWL
1	1	1	0	0	CWN
1	1	1	0	1	CWNL

This sums up to give 25 combinations of delays.

Attributes are classified as Numeric and Nominal as follows:

**Table 3.3: Distribution of Nominal & Numeric Attributes**

Numeric Attributes	Nominal Attributes
CRS_Dep_time	Quarter
CRS_Arr_time	Day_of_Month
Dep_Time	Month
Arr_Time	Day_of_week
Air_time	Unique_Carrier
Distance	Origin
	Dest
	Reason_delay
	Arr_delay_new
	Dep_delay_new

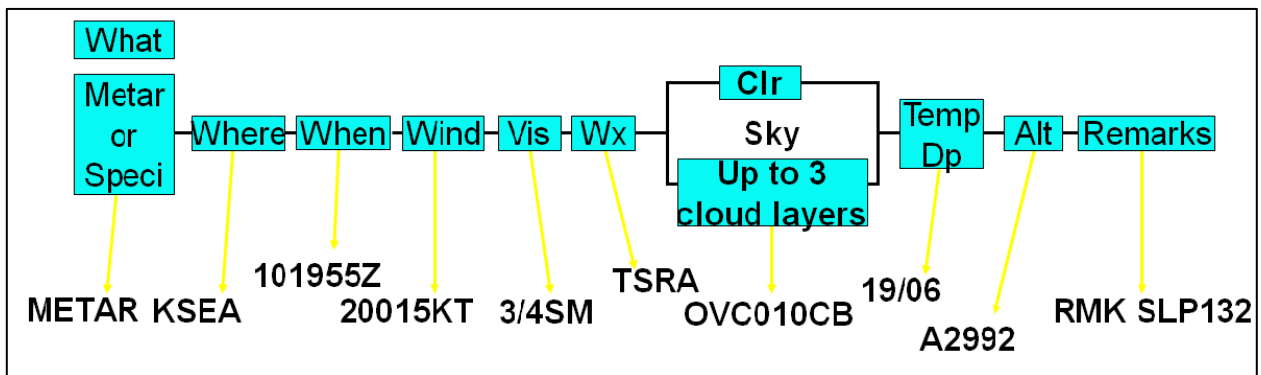
At the end of data preprocessing our dataset has the following features:

1. Flight Features: These include the departure and arrival time, source and destination airport and distance covered by the flight, types of delay viz. Carrier, Late Aircraft, NAS, Weather, and Security.
2. Weather Features: We use METAR for reporting weather information used by pilots and meteorologists to assist in weather forecasting.

### 3.3.2 METAR Weather Reporting

METAR (Meteorological Terminal Aviation Routine Weather Report) is a format for reporting weather information. A METAR weather report is used by pilots for a pre-flight weather briefing. METAR Reader provides a string by entering the ICAO (International Civil Aviation Organization) code of the airport. The ICAO code contains the airport name.

For example, the ICAO code **KLAX** stands for Los Angeles International Airport, United States. The METAR Reader takes ICAO code as input and provides string which contains various parameters like report time, winds, visibility, clouds, temperature, pressure and other remarks about the snow/rain precipitation.



**Figure 3.7: METAR Raw String Format**

The METAR reader provides the option to convert the string to raw format as well as translated format which can be understood by airport officials.

A mapping of ICAO codes and IATA (International Air Transport Association) is prepared to obtain ICAO code for corresponding airport like **LAX i.e. Los Angeles International Airport has KLAX ICAO code.**

IATA codes are usually derived from the name of the airport or the city it serves, while ICAO codes are distributed by region and country. ICAO codes are also used to identify other

aviation facilities such as weather stations or Area Control Centers, whether or not they are located at airports. The first letter is allocated by continent and represents a country or group of countries within that continent. The second letter generally represents a country within that region, and the remaining two are used to identify each airport.

To determine the weather conditions at airport the string is generated according the threshold values given below.

1. **Wind:** speed given in Knots (KT).

Wind\_speed <10KT is considered as moderate wind in air.

Wind\_speed >25KT is considered as high wind in air.

2. **Visibility:** given in Statute Miles(SM).

Visibility <4SM is considered to be low visibility

3. **Sky/Cloud conditions:** given as (OVC/FEW/CLR/SCT/CB/TCU)

OVC: Overcast

FEW: Few

CLR: No clouds below 12,000 ft. (3,700 m)

SCT: Scattered

CB: cumulonimbus cloud

TCU: towering cumulus

4. **Weather** information follows the format:

Intensity...Description... Precipitation... Obscuration... Other

(-/+)... (PR/SH/TS/FZ/DR)... (DZ/RA/SN)... (BR/FG/FU/SA)... (SS/DS/FC/SQ)

PR: partial

SH: showers

TS: thunderstorm

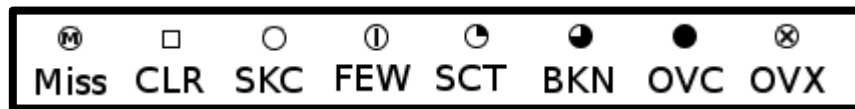
FZ: freezing

DR: drift

DZ: drizzle

RA: rain  
SN: snow  
BR: mist  
FG: fog  
FU: smoke  
SA: sand

METAR reports are also provided by Automated Weather Observing System (AWOS), Automated Surface Observing System (ASOS), and Automated Weather Sensor System (AWSS).



**Figure 3.8: Weather Information Format**

Automated stations report “CLR” when clouds may exist above 12,000 feet and “SKC” when sky is completely clear overhead. “OVX” indicates the sky is obscured which is the case when METAR reports vertical visibility and no cloud formation.

METAR information which is recorded on hourly basis is updated and replaced every 20 minutes from the previous observations if any major changes are being observed in weather during that particular hour.

The ICAO code entered in METAR Reader retrieves the weather information of that airport. A METARreport generated is combined with data mining algorithm. The combination output tells whether Delay will occur or not and if yes then reason for delay and by how many minutes has flight been delayed is displayed.

### 3.3.3 Data Mining Algorithm

The model consists of 13 attributes and Reason\_delay attribute as the class label. The various types of delay have been converted to format of 0s and 1s. The attributes which do not cause delay have been labeled “0” and ones causing delay have been labeled “1”.

A combination of these types formed a new attribute “Reason\_delay”. An abbreviated form of delay has been taken for Reason\_delay attribute.

The delays have been labeled as follows:

C: CARRIER DELAY

W: WEATHER DELAY

S: SECURITY DELAY

L: LATE AIRCRAFT DELAY

N: NAS DELAY

Reason delay consisting of 25 class labels has weights assigned to each label.

Example:

0, CL, CLW....25

$\sum \text{weight}(0) = \text{Total weight assigned to class '0'}$

$\sum \text{weight}(CL) = \text{Total weight assigned to class 'CL'}$

This gives total 25 weights for class labels. Based on the difference between the weights assigned, each class label is indexed to check whether delay has occurred or not.

Example:

If  $\sum \text{weight}(0) > (\sum \text{weight}(\text{CL}) + 100)$  then Delay has occurred else if

$\sum \text{weight}(0) < (\sum \text{weight}(\text{CL}) + 100)$  then Delay did not take place.

If CRS time and actual time of departure is available and if the delay has occurred then result is given along with reason for Delay.

OneR algorithm is one-level decision tree that generates a set of rules that test one particular attribute assuming nominal attributes. It is a classification algorithm that generates one rule for each predictor in the data and then selects the rule with the smallest total error as its "one rule". The attribute with least error rate is the best attribute. Since the accuracy is highest and time to build the model is least i.e. the performance of OneR algorithm is better as compared to other algorithms, the models were developed using this algorithm.

**Table 3.4: Performance of Algorithms**

Classifier	Accuracy
J48	62.88%
OneR	64.08%
REPTree	62.12%
BayesNet	54.81%
Naïve Bayes	54.37%
Ibk(k=37)	62.68%

The accuracy obtained using oneR algorithm is described as follows:

Initially OneR was applied on 590951 records .After removing misclassified records, 475140 records were obtained. The classifying attribute used by oneR algorithm is Arr\_Time. Inverting the selection and again applying oneR 115811 records were obtained.

Accuracy obtained is 99.98%.

Removing Arr\_time attribute oneR algorithm is again applied on 115811 records.

After removing misclassified records, 30035 records were obtained. The classifying attribute used by oneR algorithm is Distance. Inverting the selection and again applying oneR 85776 records were obtained.

Accuracy obtained is 99.00%.

Removing Distance attribute oneR algorithm is again applied on 85776 records.

After removing misclassified records, 20382 records were obtained. The classifying attribute used by oneR algorithm is CRS\_Arr\_time. Inverting the selection and again applying oneR 65394 records were obtained.

Accuracy obtained is 97.82%.

Removing CRS\_Arr\_time attribute oneR algorithm is again applied on 65394 records. After removing misclassified records, 14372 records were obtained. The classifying attribute used by oneR algorithm is Dep\_Time. Inverting the selection and again applying oneR 51022 records were obtained.

Accuracy obtained is 98.14%.

Removing Dep\_Time attribute oneR algorithm is again applied on 51022 records.

After removing misclassified records, 10125 records were obtained. The classifying attribute used by oneR algorithm is Origin. Inverting the selection and again applying oneR 40897 records were obtained.

Accuracy obtained is 99.44%.

Removing Origin attribute oneR algorithm is again applied on 40897 records.

After removing misclassified records, 7932 records were obtained. The classifying attribute used by oneR algorithm is Destination. Inverting the selection and again applying oneR 32965 records were obtained.

Accuracy obtained is 99.47%.

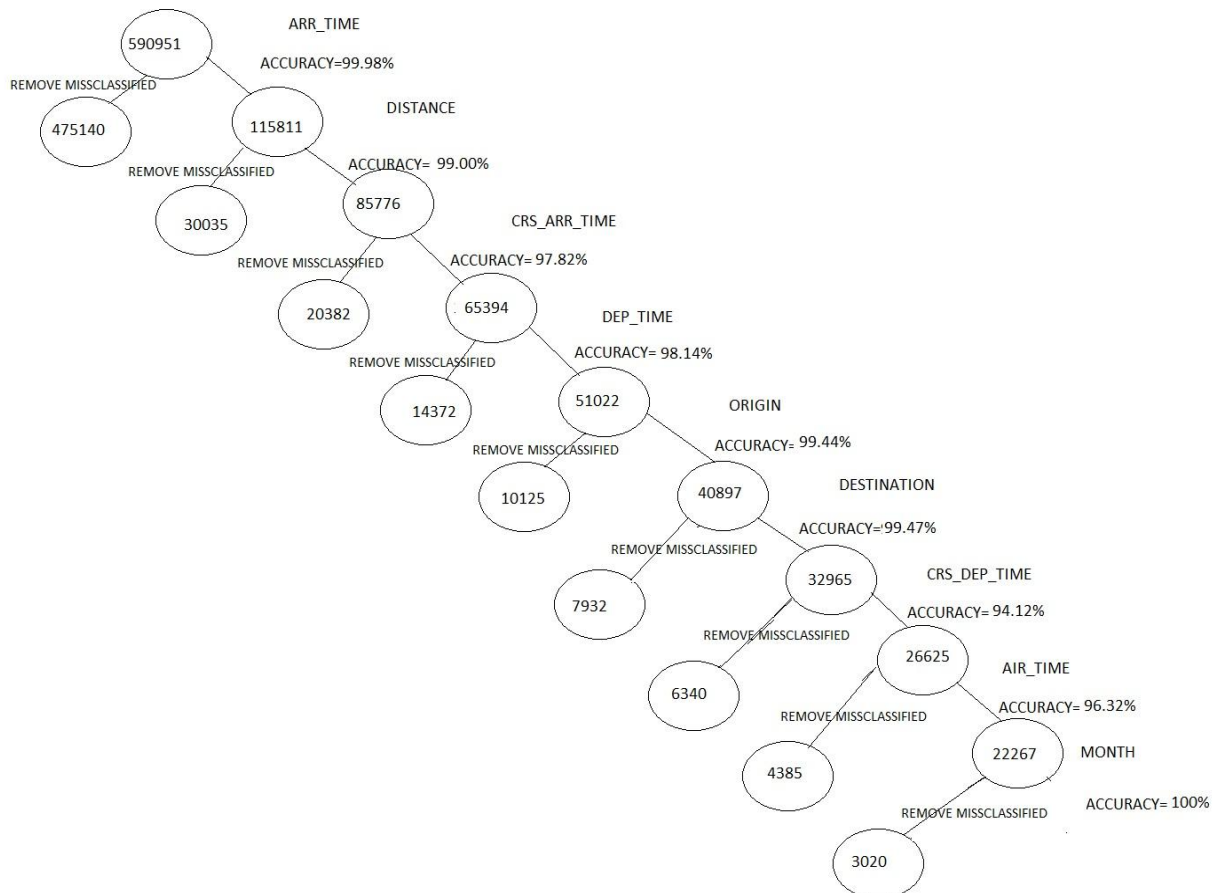


Removing Destination attribute oneR algorithm is again applied on 32965 records.

After removing misclassified records, 6340 records were obtained. The classifying attribute used by oneR algorithm is CRS\_dep\_time. Inverting the selection and again applying oneR 26625 records were obtained.

Accuracy obtained is 94.12%.

Error rate for the skipped instances is 3.76%



**Figure 3.9: OneR Model**

Removing CRS\_dep\_time attribute oneR algorithm is again applied on 26625 records.

After removing misclassified records, 4385 records were obtained. The classifying attribute used by oneR algorithm is Air\_time. Inverting the selection and again applying oneR 22267 records were obtained.

Accuracy obtained is 96.32%.

Removing Air\_time attribute oneR algorithm is again applied on 22267 records.

After removing misclassified records, 3020 records were obtained. The classifying attribute used by oneR algorithm is Month.

Accuracy obtained is 100%.

The model created by applying OneR algorithm with 10 fold cross validation is further used to predict delay on departure side and arrival side.

To find delay based on departure and arrival, dep\_delay\_new and arr\_delay\_new (nominal) is classified in 5 classes with following range:

**Table 3.5: Range for Departure & Arrival Delay**

	Departure	Arrival
Negligible	<b>0</b>	<b>0</b>
Insignificant	<b>1-15</b>	<b>1-16</b>
Nominal	<b>16 -49</b>	<b>16-49</b>
Significant	<b>50-109</b>	<b>50-109</b>
Indefinite	<b>&gt;109</b>	<b>&gt;109</b>

The range in the above the table is obtained by discretization in WEKA. Initially the range of dep\_delay\_new and arr\_delay\_new was 0-1925 minutes which was discretized to give 5 classes.

Discretization is done to use classifier to handle only nominal data. Discretization can be done with filter `weka.filters.unsupervised.attribute.Discretize` which uses simple binning. Attributes like dep\_delay\_new and arr\_delay\_new are converted from numeric to nominal by applying discretization. This means we can simply discretize by removing the keyword "numeric" as the type for the " dep\_delay\_new" and "arr\_delay\_new " attributes in the ARFF file, and replacing it with the set of discrete values values.

The WEKA discretization filter, can divide the ranges, or used various statistical techniques to automatically determine the best way of partitioning the data. In this case, we will perform simple binning. First we will load our filtered data set into WEKA. Now, we activate the Filter dialog box and select "`weka.filters.unsupervised.attribute.Discretize`".

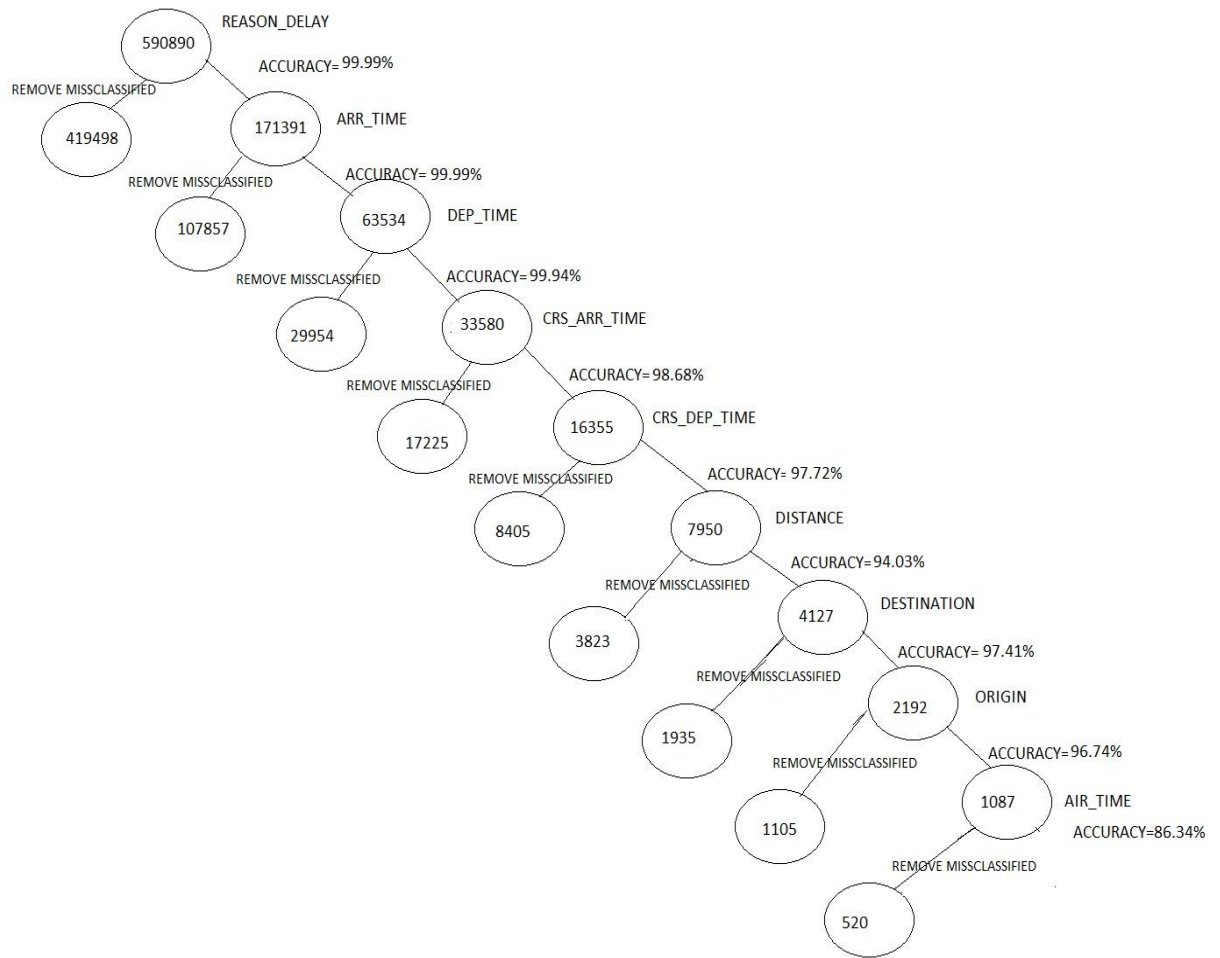
Next click on the box immediately to the right of the "Choose" button to open the Discretize Filter dialog box. We enter the index for the the attributes to be discretized. Since we are doing simple binning all other options are set to false. On clicking "Apply" the result in a new working relation with the selected attribute partitioned into 5bins.

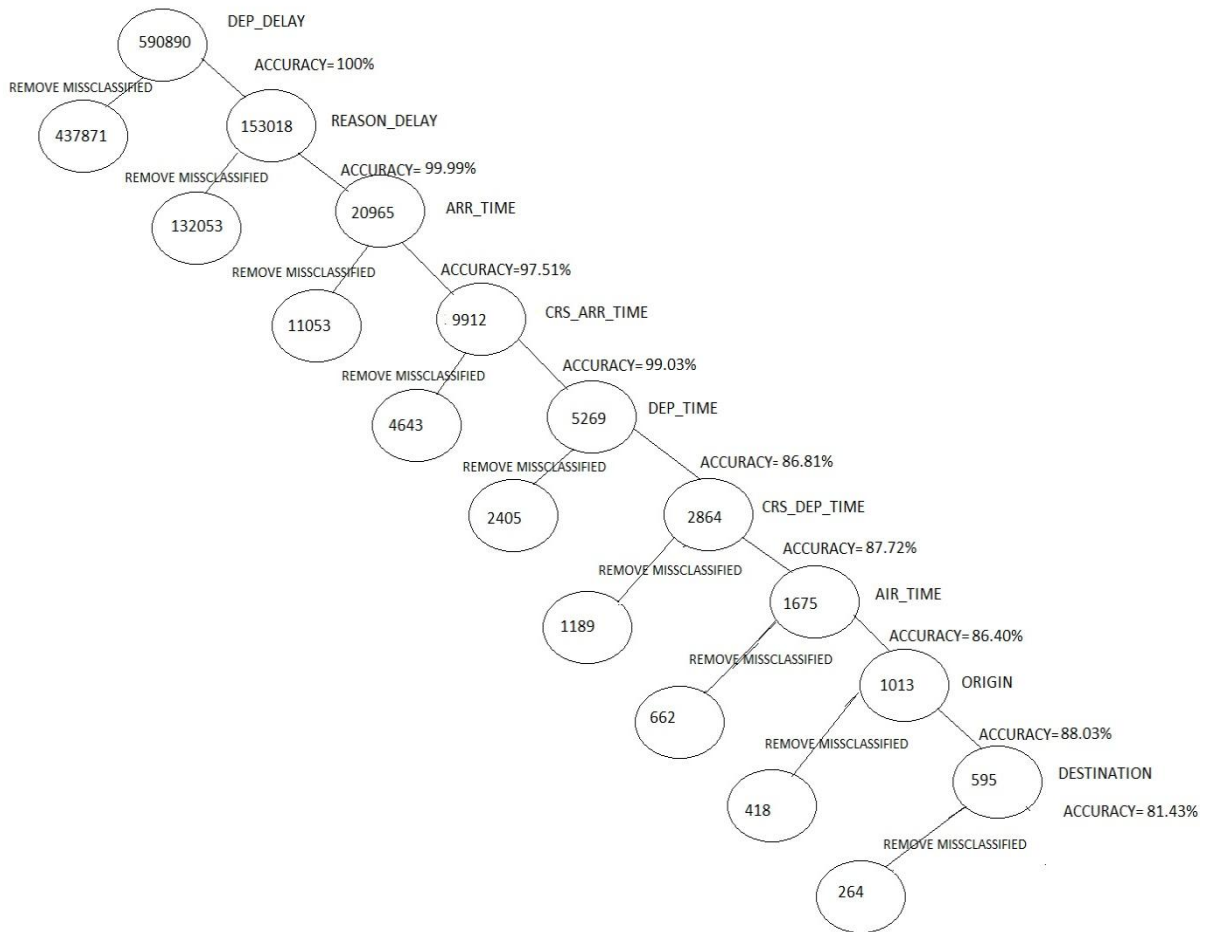
WEKA has assigned its own labels to each of the value ranges for the discretized attribute. For example, the lower range in the "dep\_delay\_new" and "arr\_delay\_new" attribute is labeled "(-inf-34.333333]" while the middle range is labeled "(34.333333-50.666667]", and so on. These labels now also appear in the data records where the original dep\_delay\_new and arr\_delay\_new value was in the corresponding range. The instances of the old patterns with the new one can be replaced. Furthermore, the outliers were identified and removed and hence the dataset for departure and arrival delay prediction model had 590889 records.

The error rate for departure delay model for skipped instances is 0.18%

The error rate for arrival delay model for skipped instances is 0.1%

The Departure and Arrival Delay model are as given below:

**Figure 3.10: Departure Delay Model**

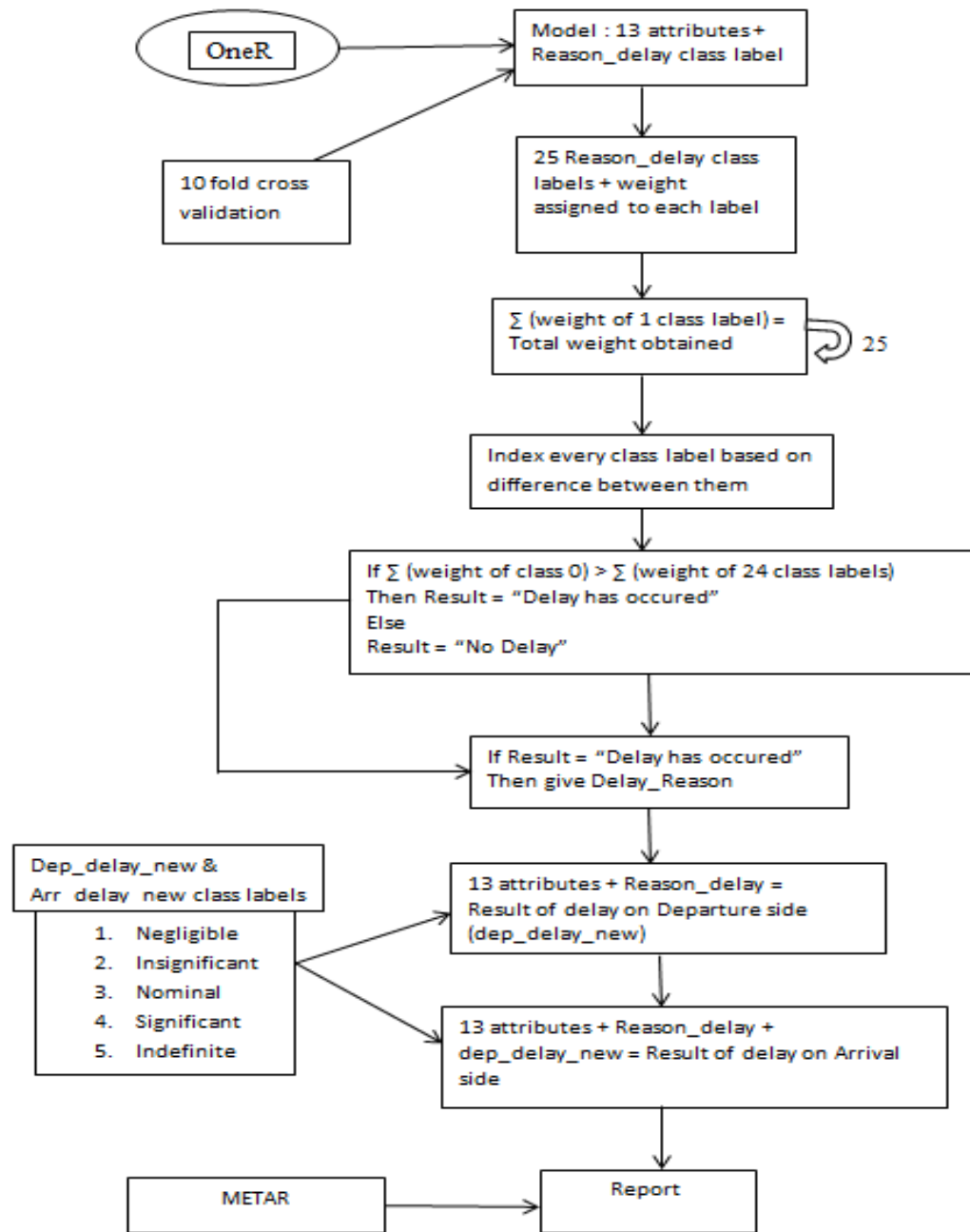


**Figure 3.11: Arrival Delay Model**

The model formed which include 13 attributes, Reason\_ delay is applied along with Dep\_delay\_new to get departure delay. On combining the class labels of Reason\_delay and Dep\_delay\_new, we get the result of delay on departure side.

The model application process further repeats to get the delay on arrival side. On combining the class labels of Reason\_delay and result of delay on departure side, we get the result of delay on arrival side.

The following figure shows the workflow of the algorithm.



**Figure 3.12: Workflow of Algorithm**

The final result obtained is combined with METAR report to give the report on overall delay predicted.

### 3.3.4 User Interface

#### MODULE 1: VIEW GRAPH

The user can select the flight details available in the dropdown box. Once the fields are selected, user can view the graph of the performance accordingly. This module will display graphs (time vs. delay). The user can view the graphical output generated based on the past performance of the flights. User can analyze the delays, Flights performance and reasons for various delays.



**Figure 3.13: View Graph**

The graphs are displayed using **JFreeChart** from following site-

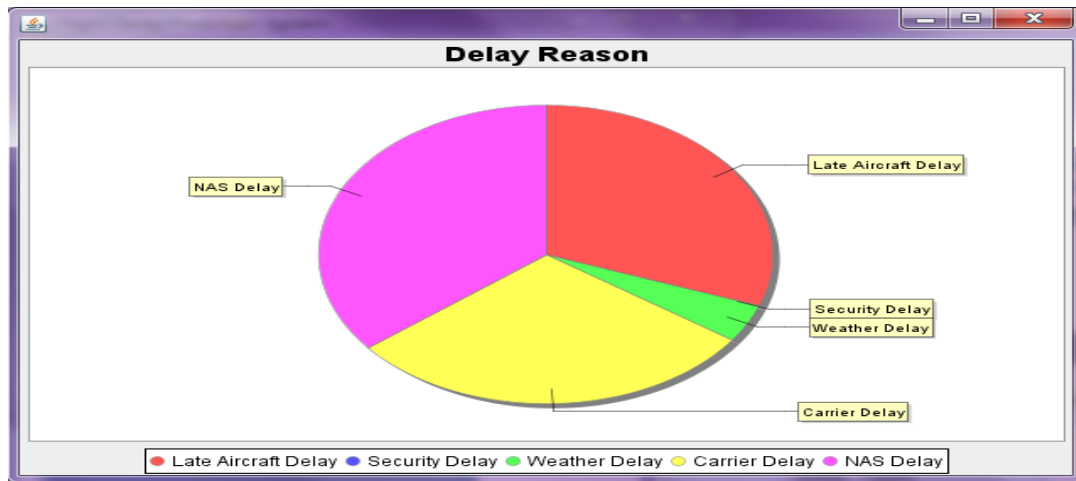
<http://www.jfree.org/jfreechart/>, which is a class library and it allows creation of interactive as well as non-interactive charts.

JFreeChart automatically draws the axis scales and legends. Charts in GUI automatically get the capability to zoom in with mouse and change some settings through local menu.

Outputs in the form of graph are displayed as follows:

- Pie Chart:** It takes the input from current dataset and generates a pie chart showing delay reasons in percentage. The chart used for displaying result is named as PieChartDemo8. This demo chart is imported from package `org.jfree.chart.demo` and is built-in in the class library `JFreeChart`. The syntax is as follows:
 

```
BarChartDemo8 demo = new BarChartDemo8("Bar Chart Demo 8");
```

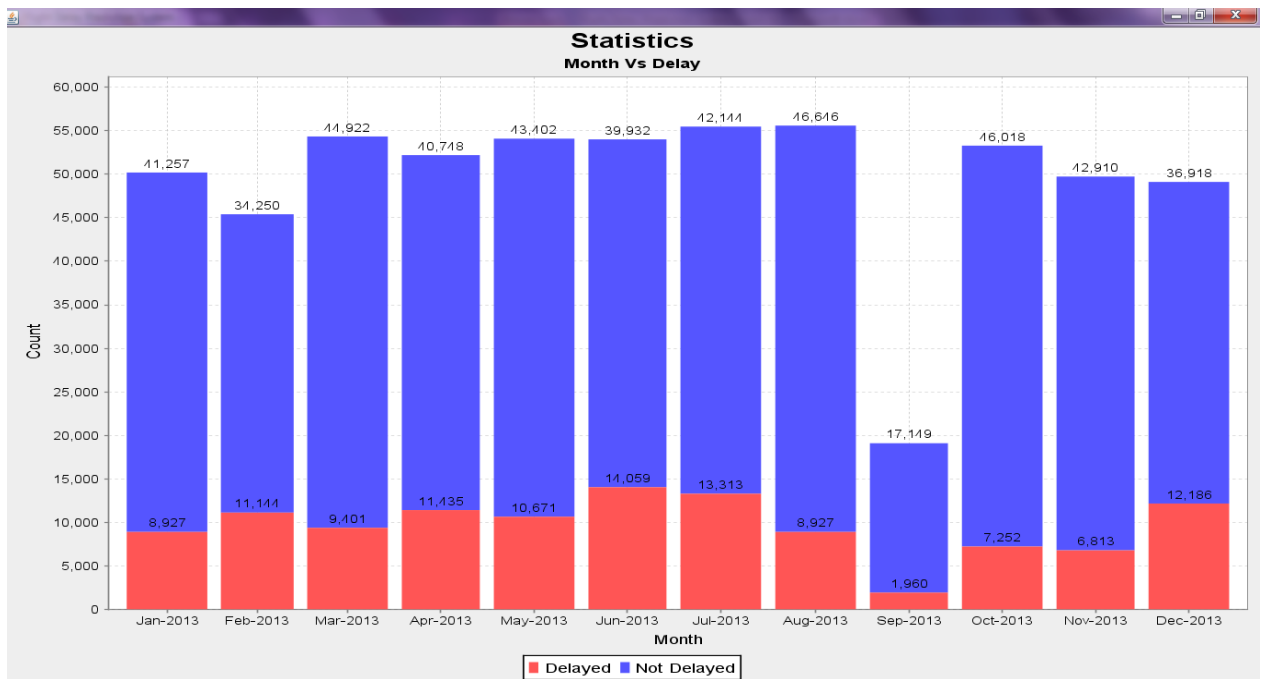


**Figure 3.14: Pie Chart**

- Bar Chart:** It shows a bar graph on quarterly, monthly or weekly basis on x-axis and its corresponding delay count on y-axis. The chart used for displaying result is named as `StackedXYBarChartDemo2`. This demo chart is imported from the package `org.jfree.chart.demo`. The syntax is as follows:
 

```
StackedXYBarChartDemo2 stackedxybarchartdemo2 = newStackedXYBarChartDemo2("StackedXYBarChartDemo2");
```





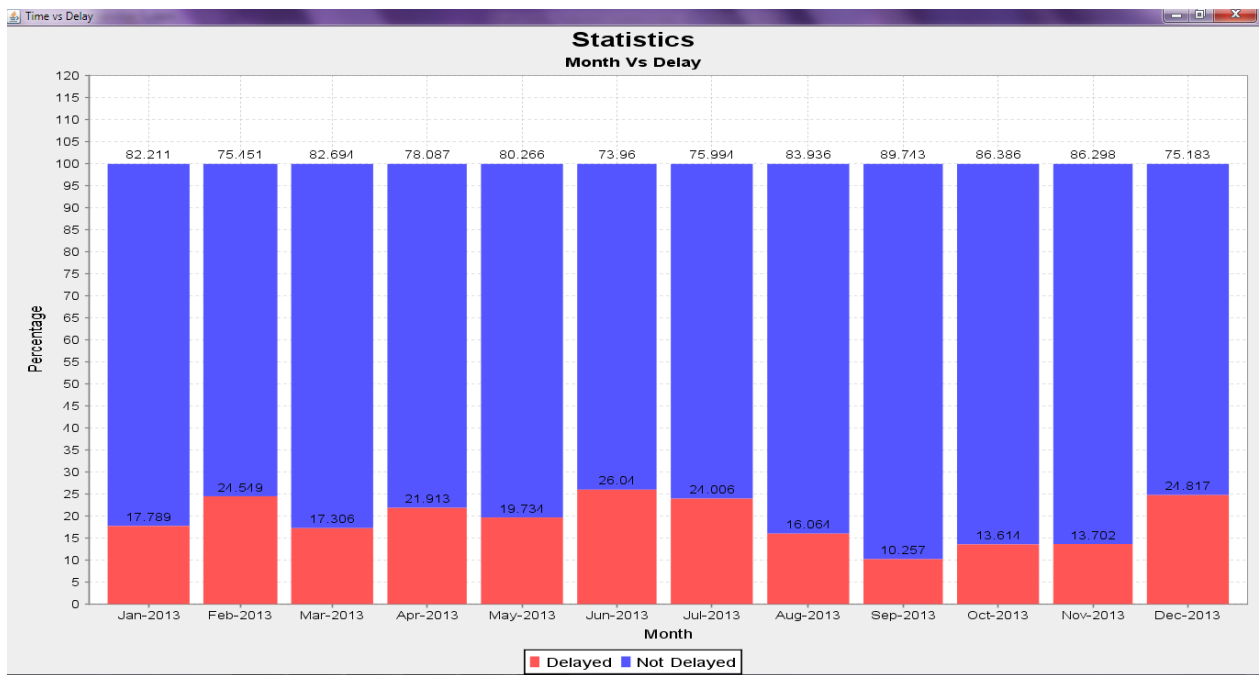
**Figure 3.15: Bar Graph**

- Normalized Bar Chart:** It shows a bar graph on quarterly, monthly or weekly basis on x-axis and its corresponding delay in percentage on y-axis. The chart used for displaying result is named as StackedXYBarChart. This demo chart is imported from the package org.jfree.chart.demo.

Percent delay and no delay are calculated as follows:

$$\text{Delay} = (\text{Delay}/\text{Sum}) * 100$$

$$\text{NoDelay} = (\text{NoDelay}/\text{Sum}) * 100$$



**Figure 3.16: Normalized Bar Chart**

## MODULE 2: PREDICT DELAY

- The user when submits the details for the flight the delay prediction report is generated. Delay prediction is made on the basis of the classification model and numeric prediction using the regression model.
- The weather information input is also taken into consideration while predicting delay. We classify the flights as Delay or On-Time. The algorithm takes a target attribute and predicts delay based on it. The module takes origin, destination and CRS departure time as input.

Flight Delay Prediction System

File Predict Delay View Graph Saved Results

Predict Delay Panel

\*Quarter: 2

\*Month: 4

\*Day of Month: 11

\*Day of Week: 7

\*Unique Carrier: US

\*Origin: Phoenix Sky Harbor Intl Airport

\*Destination: Ronald Reagan Washington National Airport

\*CRS Departure: Washington Dulles Intl Airport

\*CRS Arrival: Fairbanks Intl Airport

Actual Departure: Kansas City Intl Airport

Actual Arrival:

\*AirTime:

\*Distance:

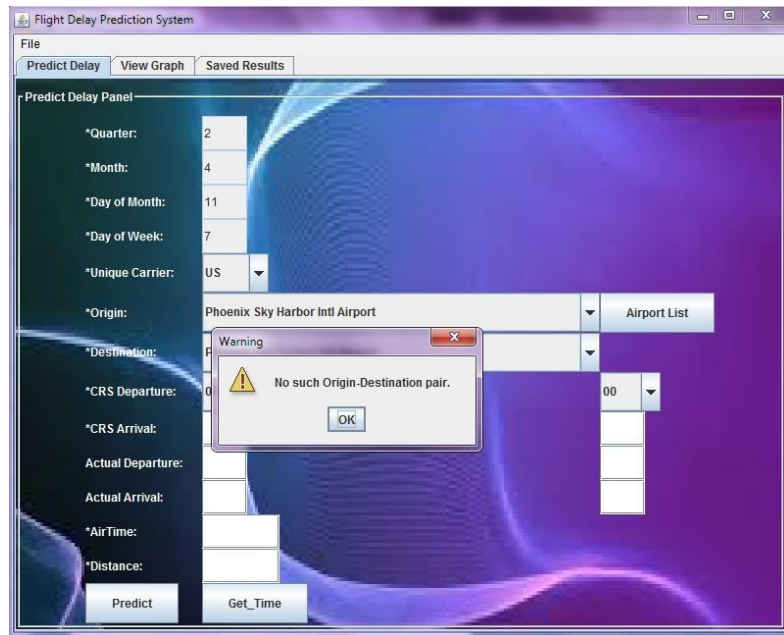
Predict Get\_Time

Figure 3.17: Predict Delay

IATA	ICAO	Location	Airport
MVY	KMVY	Vineyard Haven; Massa...	Martha's Vineyard Airport
MWA	KMWA	Marion; Illinois	Williamson County Reg...
MYR	KMYR	Myrtle Beach; South Car...	Myrtle Beach Intl Airport
OAJ	KOAJ	Jacksonville; North Car...	Albert J. Ellis Airport
OAK	KOAK	Oakland; California	Metropolitan Oakland In...
OGG	PHOG	Kahului; Hawaii	Kahului Airport
OKC	KOKC	Oklahoma City; Oklaho...	Will Rogers World Airport
OMA	KOMA	Omaha; Nebraska	Eppley Airfield
OME	PAOM	Nome; Alaska	Nome Airport
ONT	KONT	Ontario; California	Ontario Intl Airport
ORD	KORD	Chicago; Illinois	Chicago O'Hare Intl Air...
ORF	KORF	Norfolk; Virginia	Norfolk Intl Airport
ORH	KORH	Worcester; Massachus...	Worcester Regional Air...
OTH	KOTH	North Bend; Oregon	Southwest Oregon Reg...
OTZ	PAOT	Kotzebue; Alaska	Ralph Wien Memorial A...
OXR	KOXR	Oxnard; California	Oxnard Airport
PAH	KPAH	Paducah; Kentucky	Barkley Regional Airport
PBI	KPBI	West Palm Beach; Flori...	Palm Beach Intl Airport
PCW	KPCW	Port Clinton; Ohio	Erie-Ottawa Regional A...
PDT	KPDT	Pendleton; Oregon	Eastern Oregon Regio...
PDX	KPDX	Portland; Oregon	Portland Intl Airport
PFN	KPFN	Panama City; Florida	Panama City-Bay Coun...
PGA	KPGA	Page; Arizona	Page Municipal Airport
PGV	KPGV	Greenville; North Caroli...	Pitt-Greenville Airport
PHF	KPHF	Newport News; Virginia	Newport News/William...
PHL	KPHL	Philadelphia; Pennsylv...	Philadelphia Intl Airport
PHX	KPHX	Phoenix; Arizona	Phoenix Sky Harbor Intl ...
PIA	KPIA	Peoria; Illinois	Greater Peoria Region...

Figure 3.18: IATA-ICAO Mapper

- In this module on entering the CRS Departure time and selecting the corresponding two letter unique carrier code, user receives the air time (in minutes) and distance (in miles) from origin to destination. This retrieval of values is done by mapping the origin-destination IATA Code along with distance and air-time by programming in java.



**Figure 3.19: Predict Delay 1**

- Also the user needs to select the origin and destination airport code. If there is no Origin-Destination pair existing then a message box appears with message “No such Origin-Destination pair”.
- Apart from this, on entering the scheduled Departure Time of the flight in hours, user receives the CRS Arrival Time of the flight at the corresponding Destination airport on clicking “Get\_Time” button.  
A distance-airtime mapper calculates CRS Arrival time.
- The module has additional two options of actual departure and arrival time of the flight whose values will be null if the user is searching for the flight information that is yet to depart.
- The Generate button at the bottom gives user the report of the flight details along with its delay predictions and appropriate reasons for delay.

- The Generate button at the bottom gives user the weather information for the corresponding flight which is retrieved from web link:-

<ftp://tgftp.nws.noaa.gov/data/observations/metar/decoded/KLAX.TXT>

The report generated has an option of save for future use, these reports can be viewed in module view saved results.

Flight Delay Prediction System

File

Predict Delay View Graph Saved Results

Predict Delay Panel

\*Quarter: 2

\*Month: 4

\*Day of Month: 11

\*Day of Week: 7

\*Unique Carrier: US

\*Origin: Seattle-Tacoma Intl Airport

\*Destination: Denver Intl Airport

\*CRS Departure: 03

\*CRS Arrival: 5

Actual Departure:

Actual Arrival:

\*AirTime: 137

\*Distance: 1024

Predict Get\_Time

**Figure 3.20: Predict Delay 2**

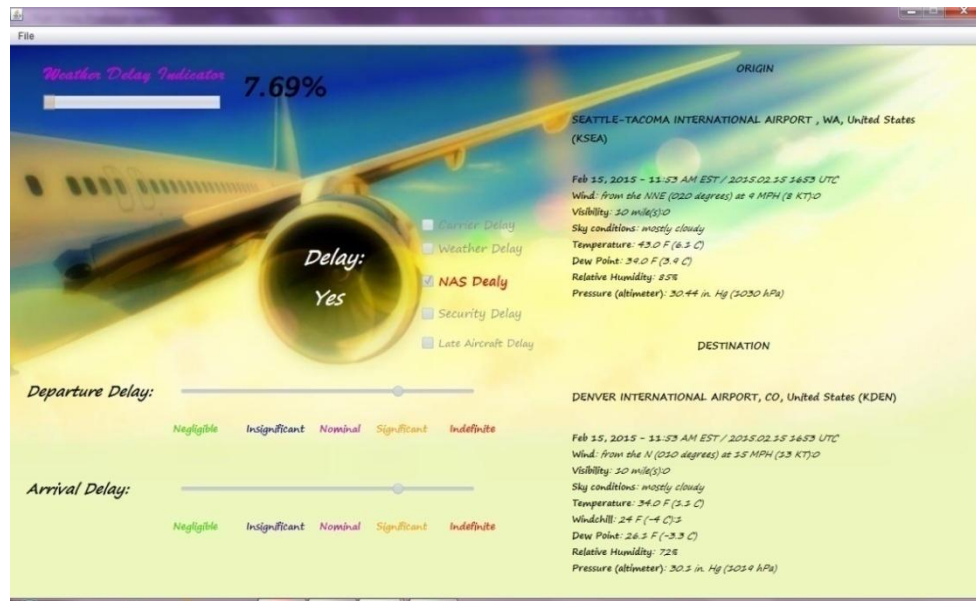


Figure 3.21: Delay Report

## MODULE 3: VIEW SAVED RESULTS

This module allows the user to view the saved reports. It has an option of saving the result for future use, which on saving the results gets stored in the result directory list. When revisited the directory list, the results are displayed in the form of image in a new window.

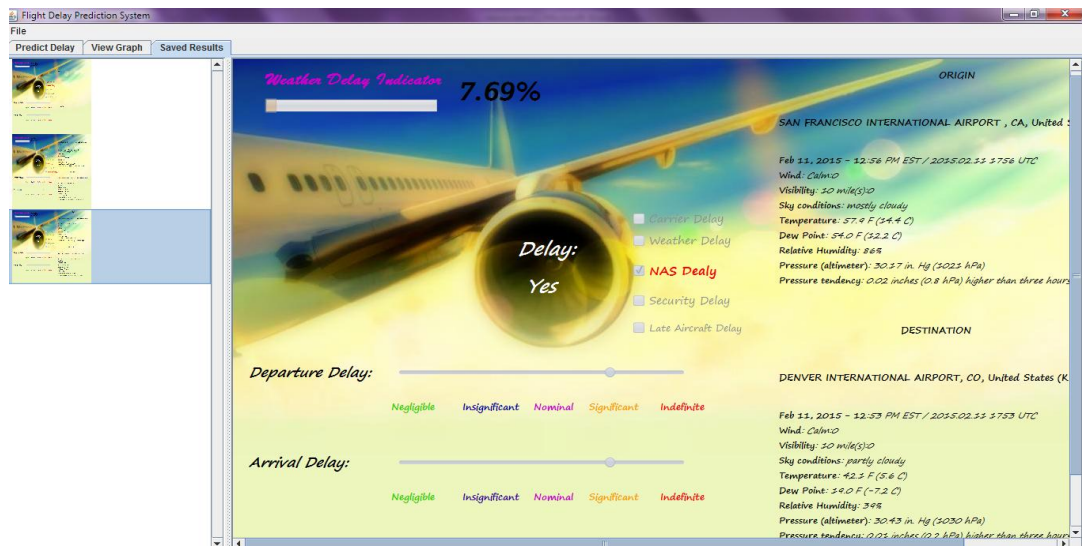


Figure 3.22: View Result

### 3.4 Testing

A test case is a set of conditions under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement. A formal written test-case is characterized by a known input and by an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post condition. A test case is usually a single step, or occasionally a sequence of steps, to test the correct behavior/functionality, features of an application. An expected result or expected outcome is usually given.

Test case ID	Objective	Test case description	Input	Expected outcome	Observed outcome	Result	Remark
1.	Origin_Destination selected from drop down menu should match	Origin-Destination pair does not exist in dataset	Select origin and destination from drop down menu	Origin-Destination pair does not exist message prompt	Exception generated	Pass	Search and remove Origin-Destination pairs which does not exist and proper error message is displayed.
2.	To generate arrival time based on departure time	Actual arrival/departure fields in GUI		Calculate Delay without error/exception on leaving it blank	Exception generated on not entering values	Pass	Improved to accept null values
3.	Correctly map the ICAO code with IATA code	ICAO mapper	Select ICAO code and IATA code	ICAO codes mapped correctly with IATA codes to retrieve data of origin and destination weather from METAR	Weather report from METAR generated correctly	Pass	

4.	Display the report	Origin-Destination Report	Enter all * marked fields on the page	Report should display Delay, Delay_Reason, Type of arrival-departure delay and METAR weather report	Results displayed as expected	Pass	
5.	Generate the weather report	Retrieval from metarreader.com	Enter the ICAO code	Generate weather report on entering ICAO code	Data not reachable	Pass	Weather information was taken FTP directory of METAR
6.	Arrival-Departure time format should be correct	Arrival-departure delay classes	Enter the arrival and departure time	Arrival-departure data should be accepted in numeric format	Regression techniques could not be applied	Pass	Data was converted to nominal format using Discretization
7.	Viewing graphs according to quarter, month, day of month and day of week	Graph generation		View graph according to quarter, month, day of month and day of week	Graph generated as expected	Pass	
8.	Creating Model to get highest accuracy	Model creation using OneR Algorithm	All the parameters for model creation with OneR algorithm	Implementation of OneR algorithm gives highest accuracy with maximum instances classified correctly	Improvement in accuracy of the prediction	Pass	
9.	Contradicting results of METAR	Result of METAR and Model	METAR weather report from FTP site and model results	Result of METAR and model can contradict	METAR and Model predict delay independently	Pass	
10.	View the saved reports	View results	Select the saved report from directory	Graph for the delay of the flight is displayed	Generation of graph done successfully	Pass	



## Chapter 4

### Results and Discussion

From this project, the results achieved are feasible and accurate enough to predict delay. At the beginning the dataset is preprocessed to identify the outliers. The preprocessed dataset is then given to the model. Models for predicting flight delay are developed using the data from The Bureau of Transportation Statistics (BTS). The model comprises of 13 attributes and class label “Reason\_Delay”. These models are then integrated in the form of a system for delay assessment. Classifier is used to detect the pattern of delay. This enables us to investigate the delay at the flight level, and the effect of a delay on the immediate flight is considered.

**Table 4.1: Confusion Matrix**

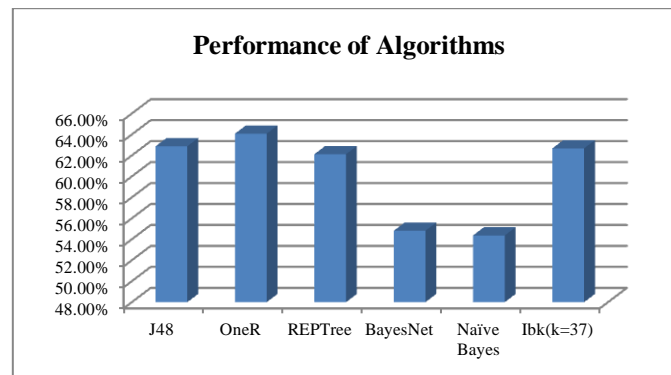
<b>Actual\Predicted</b>	<b>Delay</b>	<b>No Delay</b>
<b>Delay</b>	25978	90109
<b>No Delay</b>	12029	462835

The accuracy of the models after preprocessing was 71%. The accuracy was further improved by identifying the outliers and the pattern of outliers affecting those which were correctly classified. Discretization of these outliers helped in improving the accuracy. Improvement gave the accuracy of 82.72%. The following confusion matrix was obtained giving the number of instances correctly classified and instances which are misclassified.

The result of the delay is displayed in form of charts or graphs. The weather conditions are found to be the most significant factors that influence the arrival and departure delay. Hence current weather information is taken from website METAR Reader giving the current weather information and conditions at the airport. The algorithm and models generate stable predictions of flight plans that have small amounts of delay.

### Experiment 1:

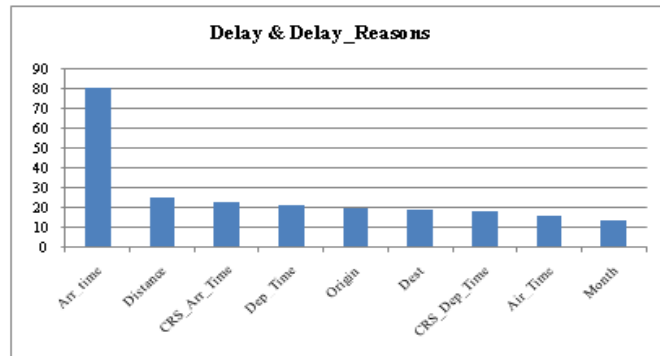
The accuracy of Naïve Bayes and Bayes obtained is 54.37% and 54.81 respectively. IBk and OneR were approximately giving the same accuracy but the model building time was more for IBk algorithm. Hence, OneR was selected to build the model.



**Figure 4.1: Graph for Performance of Algorithms**

### Experiment 2:

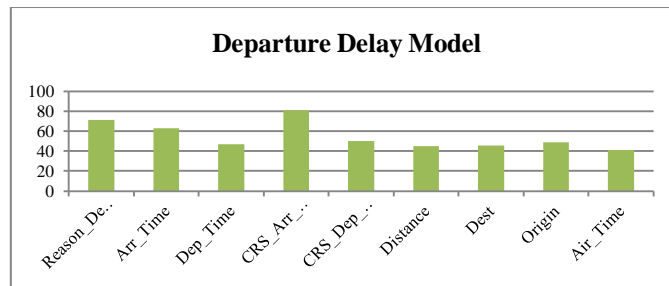
The predictor attribute were in the order as shown in the graph below. The graph below shows the weight assigned to every attribute which was determined with the help of accuracy and Relative Dataset Size (RDS).



**Figure 4.2: Delay and Delay Reasons**

### Experiment 3:

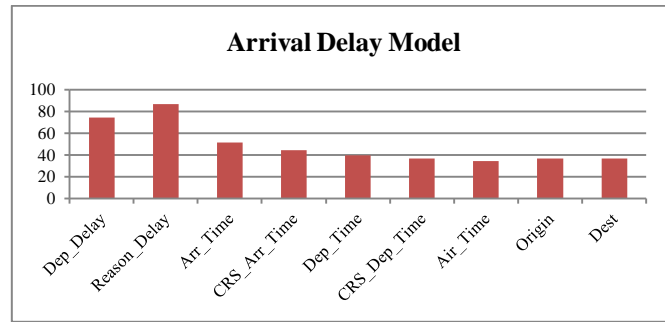
The graph below shows different weights assigned to every attribute in the Departure Delay Model. The predictor attribute were in the order as shown in the graph below. Every attribute weight was determined with the help of accuracy and Relative Dataset Size (RDS).



**Figure 4.3: Departure Delay Model**

### Experiment 4:

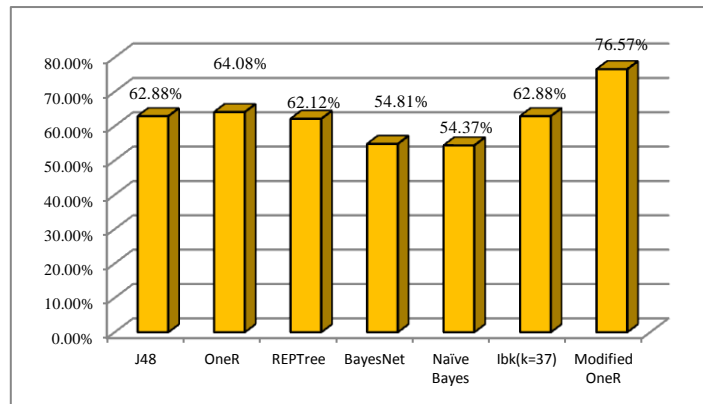
The graph below shows different weights assigned to every attribute in the Arrival Delay Model. The predictor attribute were in the order as shown in the graph below. Every attribute weight was determined with the help of accuracy and Relative Dataset Size (RDS).



**Figure 4.4: Arrival Delay Model**

### Experiment 5:

The accuracy of Naïve Bayes and Bayes obtained is 54.37% and 54.81 respectively. IBk and OneR was approximately giving the same accuracy but the model building time was more for IBk algorithm. Hence, OneR was selected to build the model.



**Figure 4.5: Comparison of Algorithm Performance**

By experimenting with the votes for each model we get the maximum accuracy of 76.57% for the modified OneR model by using the following weights:

$$\frac{9^3}{10} \quad \frac{8^3}{10} \quad \frac{7^3}{10} \quad \frac{6^3}{10} \quad \frac{5^3}{10} \quad \frac{4^3}{10} \quad \frac{3^3}{10} \quad \frac{2^3}{10} \quad \frac{1^3}{10}$$

The above weights were calculated using the following formulas.

$$1. \text{ Total weight of 1 class label} = \sum_{n=1}^{25} (\text{weight assigned to a class label})$$

$$\text{if} \left( \sum_{n=1}^{25} \text{weight assigned to class label "0"} \right) > \left( \sum_{n=2}^{25} \text{weight assigned to 24 class labels} + 100 \right)$$

Then

Result = “Delay”

Else

Result = “No Delay”

$$2. \text{ Relative Dataset Size (RDS)} = \frac{\text{Dataset size after filtering}}{\text{Original Dataset Size before filtering}}$$

$$3. \text{ Weight} = \frac{\text{Accuracy} + \text{RDS}}{2}$$

$$4. \text{ Weight} = \frac{\text{Accuracy} * \text{RDS}}{2}$$

**Table 4.2: Delay and Delay Types**

No of records	After Filter	On Attribute	Accuracy	Weight
590951	475140	Arr_time	99.98%	80.39
115811	30035	Distance	99.00%	25.68
85776	20382	CRS_Arr_time	97.82%	23.25
65394	14372	Dep_time	98.14%	21.57
51022	10125	Origin	99.44%	19.73
40897	7932	Dest	99.47%	19.29
32965	6340	CRS_Dep_time	94.12%	18.1
26625	4385	Air_time	96.32%	15.77
22267	3020	Month	100%	13.56

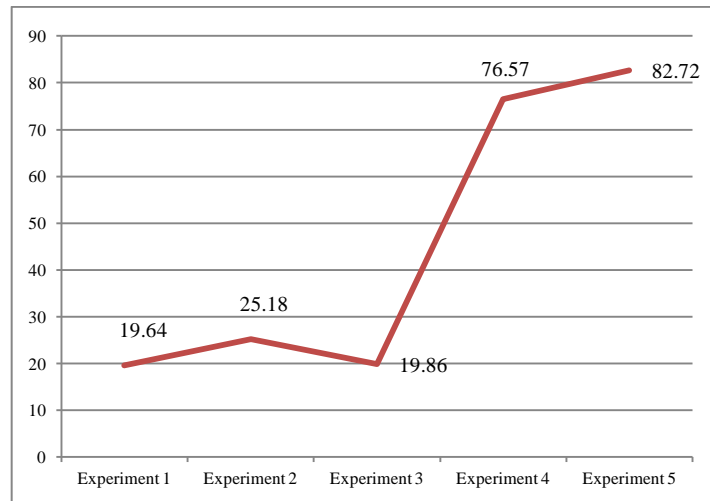
**Table 4.3: Departure Delay**

No of records	After Filter	On Attribute	Accuracy	Weight
590890	419498	Reason_delay	99.99%	70.99
171391	107857	Arr_time	99.99%	62.93
63534	29954	Dep_time	99.94%	47.12
33580	17225	CRS_Arr_time	98.68%	80.62
16355	8405	CRS_Dep_time	97.72%	50.22
7950	3823	Distance	94.03%	45.22
4127	1935	Dest	97.41%	45.67
2192	1105	Origin	96.74%	48.77
1087	520	Air_time	86.34%	41.37

**Table 4.4: Arrival Delay**

No of records	After Filter	On Attribute	Accuracy	Weight
590890	437871	Dep_DeLAY	100%	74.1
153018	132053	Reason_delay	99.99%	86.3
20965	11053	Arr_time	97.51%	51.41
9912	4643	CRS_Arr	99.03%	44.05
5269	2405	Dep_time	86.81%	39.63
2864	1189	CRS_dep	87.72%	36.42
1675	662	Airtime	86.40%	34.15
1013	418	Origin	88.03%	36.33
595	264	destination	81.43%	36.30

The tables above give the weights assigned to different models. The graph for the comparison of experiments is given as shown below.



**Figure 4.6: Experiments Vs Accuracy**

The accuracy of 82.72% is highest in Experiment 5 after applying the Modified OneR algorithm.

## **Chapter 5**

### **Conclusion & Future Scope**

After the development of modules we have come to the conclusion that the models developed can be used in predicting the delay accurately at the airports. The delay distribution of an airport can make it easier to understand the airport delay. The results of the research show that the delay is highly related to the originate delay. In response to single flight delay predictions and reason for these delays that are generated by the model, which can give indications for the appropriate recovery actions to recover/avoid these delays.

The models developed can be applied to predict occurrence of delay at airports. Such predictive capabilities can help the managers and airline dispatchers to prepare mitigation strategies for reducing traffic disruptions. The models are calibrated using historical data. Including weather forecasts as input variables is a direction of future research.

A lot of factors go into predicting a delay in a flight departure. Delays in flight departure can be subjected to various reasons. The results of the data analysis suggest that flight delays follow certain patterns that distinguish them from on-time flights. We discovered that it is possible to make fairly good predictions on the basis of a few key attributes, such as departure time, date and carrier.



By including weather information, we should be able to improve our results even further, and thus get a better picture which largely determines where and when flight delays occur. This will help to save the airport time and hassle. Several factors can be identified and data related to those can be collected and can be used to build various models to better predict the delay in a flight across all airports. A wide variety and a rich collection of data would definitely be useful in building a better model to predict the delay.

The results of the data analysis suggest that flight delays follow certain patterns that distinguish them from on-time flights. From our models and analysis, we discovered that it is possible to make fairly good predictions on the basis of a few key attributes, such as carrier, departure time, arrival time, origin, and destination. A predictive trend within our data from the models that we developed was discovered.

Regression of the data can be done to predict the delay in minutes. Further, the project can be made into an android application.

## Appendix

This section contains theory in regards to Flight Delay Prediction System using Data Mining

### A

**AirlineID:** An identification number assigned to identify a unique airline.

**ArrDelay:** Difference in minutes between scheduled and actual arrival time.

**ArrTime:** The actual arrival time recorded when the flight arrives at the destination.

### C

**Cancellation:** The attribute specifies whether flight was cancelled, 1 means yes and 0 means no.

**CancellationCode:** It specifies the reason for cancellation (A = carrier, B = weather, C = NAS, D = security)

**CarrierDelay:** The cause of the cancellation or delay was due to circumstances within the airline's control (e.g. maintenance or crew problems, aircraft cleaning, baggage loading, fueling, etc.).

**CRSArrTime:** The attribute specifies the scheduled arrival time in hhmm format.

**CRSDepTime:** The attribute specifies the scheduled departure time in hhmm format.

### D

**DayOfMonth:** The attribute takes the value from 1 to 31.

**DayOfWeek:** The attribute takes the value from 1 to 7. (1 is Monday & 7 is Sunday)

**DepDel15:** The attribute is Departure Delay indicator, 15 Minutes or more means 1=Yes

**DepDelay:** Difference in minutes between scheduled and actual departure time.

**DepTime:** Actual Departure Time in hhmm format

**Dest:** The attribute specifies the destination IATA airport code.

## **F**

**FlightNum:** The attribute specifies the unique flight number.

## **I**

**ICAO code:** It is a four-character alpha numeric code designating each airport around the world.

**IATA code:** It is a three letter code designating airports around the world.

## **L**

**LateAircraftDelay :** A previous flight with same aircraft arrived late, causing the present flight to depart late.

## **M**

**Month:** The attribute takes the value from 1 to 12.

**Metar:** A format for reporting weather information.

## **N**

**NASDelay:** Delays and cancellations attributable to the national aviation system that refer to a broad set of conditions, such as non-extreme weather conditions, airport operations, heavy traffic volume, and air traffic control.

## **O**

**Origin:** The attribute specifies the origin IATA airport code

## **Q**

**Quarter:** The attribute takes the value from 1 to 4.

## **S**

**SecurityDelay:** Delays or cancellations caused by evacuation of a terminal or concourse, re-boarding of aircraft because of security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas.

## T

**TaxiIn:** It refers to the period of time in minutes spent by the flight on the runway after WheelsOn, i.e. from WheelsOn to Gate In.

**TaxiOut:** It refers to the period of time in minutes spent by the flight on the runway before WheelsOff, i.e. from Gate Out to WheelsOff.

## U

**UniqueCarrier:** It specifies the unique carrier code of the flight.

## W

**WeatherDelay:** Significant meteorological conditions (actual or forecasted) that, in the judgment of the carrier, delays or prevents the operation of a flight such as tornado, blizzard or hurricane.

**WheelsOff:** The time the flights takes off the runway in hhmm format.

**WheelsOn:** The time the flight lands on the runway in hhmm format.

## Y

**Year:** The attribute specifies the year.

## References

- [1] J. J. R. a. H. Balakrishnan, "Characterization and Prediction of Air Traffic Delays," Massachusetts Institute of Technology Cambridge, USA, Mar 2014.
- [2] S. G. a. B. S. Avijit Mukherjee, "Predicting Ground Delay Program At An Airport Based On Meterological Conditions," AIAA Aviation, Atlanta, June 2014.
- [3] D. A. Smith, "Decision Support Tool for predicting Aircraft arrival Rates from Weather forecasts," George Mason University, 2008.
- [4] "RITA|BTS|Transtats," [Online]. Available: <http://www.transtats.bts.gov/>. [Accessed Aug 2014].
- [5] "Directorate General of Civil Aviation," [Online]. Available: <http://dgca.nic.in/>. [Accessed Sep 2014].
- [6] "FlightRadar24," [Online]. Available: <http://www.flightradar24.com/>. [Accessed Sep 2014].
- [7] "Yahoo Weather," [Online]. Available: <https://in.weather.yahoo.com/india/>. [Accessed Sep 2014].
- [8] "KnowDelay.com," [Online]. Available: <https://www.nbcnews.com/business/travel/knowdelay-com-predicts-flight-problems-3-days-advance-f1C9870958>. [Accessed Sep 2014].
- [9] "Data Wrangling," [Online]. Available: <https://www.datawrangling.com/how-flightcaster-squeezes-predictions-from-flight-data>. [Accessed Sep 2014].
- [10] "DelayCast," [Online]. Available: <http://www.delaycast.com/>. [Accessed Oct 2014].
- [11] H. B. Juan Jose Rebollo, "A Network-Based Model for Predicting Air Traffic Delays," [Online]. Available: <http://www.mit.edu/~hamsa/pubs/RebolloBalakrishnanICRAT2012.pdf>. [Accessed Sept 2013].
- [12] "Flugzeug," [Online]. Available: [http://www.flugzeuginfo.net/table\\_airportcodes\\_country-location\\_en.php#U](http://www.flugzeuginfo.net/table_airportcodes_country-location_en.php#U). [Accessed Nov 2014].
- [13] "Aeronautical Information," [Online]. Available: [http://www.faa.gov/air\\_traffic/publications/atpubs/aim/](http://www.faa.gov/air_traffic/publications/atpubs/aim/). [Accessed Jan 2015].

## Acknowledgements

We wish to express our profound gratitude to our principal Mr. Vinit Kotak for allowing us to go ahead with this project and giving us the opportunity to explore this domain. We would also like to thank our Head of Department Mrs. Swati Deshpande for her constant encouragement and support towards achieving this goal.

We would like to thank the Review Committee for their invaluable suggestions and feedback without which our work would have been very difficult.

We take this opportunity to express our profound gratitude and deep regards to our guide Mr. Vinit Kotak for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. We owe a special acknowledgment to his for giving us a lot of their time during the period of preparing this project. We could never had done it without his support, technical advice and suggestions, thorough reading of all our work. The blessing, help and guidance given by his time to time shall carry us a long way in the journey of life on which we are about to embark.

No project is ever complete without the guidelines of these experts who have already established milestones on this path before and have become masters of it. So we would like to take this opportunity to thank all those who have helped us in implementing this project.

-----

(Sruti Oza)

-----

(Rutuja Raut)

-----

(Hetal Sangoi)

-----

(Somya Sharma)

Date: