



S	
E1	
E2	
For Office Use Only	

**Masters Project Final Report**  
**(MCS)**  
**2020**

<b>Project Title</b>	Predicting Airline On-time Performance
<b>Student Name</b>	H. D. C. M. Ariyawansa
<b>Registration No. &amp; Index No.</b>	2016/MCS/008 - 16440084
<b>Supervisor's Name</b>	Dr. Ajantha Atukorale

<b>For Office Use ONLY</b>



# **Predicting Airline On-time Performance**

**A dissertation submitted for the Degree of Master of  
Computer Science**

**H. D. C. M. Ariyawansa  
University of Colombo School of Computing  
2020**



# Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: H. D. C. M. Ariyawansa

Registration Number: 2016/MCS/008

Index Number: 16440084



Signature:

Date: 17/11/2020

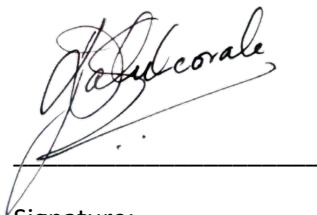
This is to certify that this thesis is based on the work of

Mr. H. D. C. M. Ariyawansa

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. Ajantha Atukorale



Signature:

Date: 17/11/2020

# Abstract

Growth of population and everyday needs made the world a busy place and transportation has now become one of the basic needs for every human being. Airlines being the easiest and fast transportation mechanism for long distance has increased its popularity over the years thus making a remarkable growth in aviation industry. Still this growth is not sufficient enough to cater air traffic congestion which causes flight delays. Therefore, the primary objective of this research is to predict and measure the flight delays so that the airlines can improve their on-time performance. The passengers have the benefit of adjusting their time schedules based on these predictions.

The research focuses on predicting the departure delay of airlines while investigating the share of weather-related delays. The research problem is addressed as classification and regression tasks. Binary classification approach is used to classify the flights into delayed and non-delayed classes while regression is used to predict the delay time of a flight.

The experiments are carried out using five years' worth of flight records. Data sampling, encoding and scaling like preprocessing techniques used to prepare the data for learning. Logistic Regression, Linear Regression, Random Forest, Decision Trees and Naïve Bayes classification are used as statistical models. A Feed Forward and Convolutional neural networks are considered for the deep learning models. Each of these models were then evaluated using their respective performance matrices.

Finally, the research came to a conclusion with the highest performing Random Forest model for the classification task. Feed forward neural network is identified as the suitable model for the regression task. Convolutional neural network seems to be the second best option for both classification and regression tasks.

# Acknowledgment

It is my humble opinion that it was through the help of many amazing individuals that I was able to complete this thesis and make it a reality. I would like to convey my gratitude to the following people.

- To my supervisor Dr. Ajantha Atukorale who guided me through the entire research with his vast experience and knowledge.
- To our project coordinator Dr. Randil Pushpananda for his guidance and support.
- To all the other academic members of UCSC for the knowledge they shared throughout all these years.
- To all my friends who kept me company in the difficult times.
- Finally my love and gratitude goes out to my parents for believing in me throughout all this work that I have put in.

# Table of Contents

<b>DECLARATION</b> .....	<b>I</b>
<b>ABSTRACT</b> .....	<b>II</b>
<b>ACKNOWLEDGMENT</b> .....	<b>III</b>
<b>TABLE OF CONTENTS</b> .....	<b>IV</b>
<b>LIST OF FIGURES</b> .....	<b>VIII</b>
<b>LIST OF TABLES</b> .....	<b>IX</b>
<b>LIST OF ABBREVIATIONS</b> .....	<b>XI</b>
<b>CHAPTER 1: INTRODUCTION</b> .....	<b>1</b>
1.1 OVERVIEW .....	1
1.2 RESEARCH BACKGROUND .....	1
1.3 MOTIVATION .....	2
1.4 GOAL.....	2
1.5 OBJECTIVE OF THE STUDY.....	2
1.6 TECHNOLOGY DOMAIN .....	3
1.6.1 Machine Learning.....	3
1.6.2 Classification and Regression.....	3
1.7 SCOPE.....	4
1.8 STRUCTURE OF THE DISSERTATION.....	4
<b>CHAPTER 2: LITERATURE REVIEW</b> .....	<b>5</b>
2.1 FLIGHT DELAYS .....	5
2.1.1 Basis for Delayed Flight Operations .....	5
2.1.2 Flight Delays Caused by Weather Conditions.....	7
2.2 RELATED RESEARCH FOR FLIGHT DELAY PREDICTION.....	8
2.2.1 Flight Delay Classification .....	8
2.2.2 Measuring Flight Delays .....	10
2.3 RESEARCH GAP .....	11
2.4 SUMMARY .....	11
<b>CHAPTER 3: METHODOLOGY</b> .....	<b>12</b>
3.1 OVERVIEW .....	12
3.2 REPRESENTATION OF THE PROBLEM .....	12
3.3. METHODOLOGY FOR BUILDING PREDICTIVE MODELS .....	13
3.4 INFRASTRUCTURE AND PLATFORM .....	13
3.4.1 PC Hardware .....	14
3.4.2 Selection of Tools and Programming Language .....	14
3.5 PREPROCESSING .....	14
3.5.1 Datasets.....	14

3.5.2 Dataset Exploration .....	15
3.5.3 Data Cleaning and Filtering.....	17
3.5.4 Sampling Dataset.....	17
3.5.5 Introducing New Attribute .....	18
3.5.6 Training, Validation and Testing Datasets .....	18
3.5.7 Handling Text and Categorical Attributes .....	19
3.5.8 Feature Scaling .....	20
3.6 LEARNING .....	21
3.6.1 Linear Regression.....	21
3.6.2 Logistic Regression .....	21
3.6.3 Decision Trees .....	21
3.6.4 Random Forest.....	22
3.6.5 Naive Bayes.....	22
3.6.6 Artificial Neural Network (ANN) .....	22
3.6.7 Hyperparameter Optimization .....	25
3.7 PREDICTIVE MODEL EVALUATION .....	25
3.7.1 Evaluation Metrics for Classification.....	25
3.7.2 Evaluation Metrics for Regression .....	29
3.8 PREDICTION.....	30
3.9 EXPERIMENTS.....	30
3.9.1 Stage One Experiments .....	30
3.9.2 Stage Two Experiments.....	31
3.9.3 Stage Three Experiments.....	31
3.9.4 Stage Four Experiments .....	31
3.9.5 Stage Five Experiments.....	31
3.9.6 Stage Six Experiments.....	32
3.10 SUMMARY .....	32
<b>CHAPTER 4: EVALUATION.....</b>	<b>33</b>
4.1 OVERVIEW .....	33
4.2 FLIGHT DELAY CLASSIFICATION .....	33
4.2.1 Stage One: One Year Worth of Data Records.....	33
4.2.2 Stage Two: Five Years' worth of Data records .....	40
4.2.3 Stage Three: Balanced Data Sample .....	44
4.2.4 Stage Four: Hyperparameter Tuning.....	49
4.2.5 Stage Five: Introducing Convolutional Neural Network (CNN) .....	51
4.2.6 Stage Five: Changing Delay Threshold.....	55
4.3 PREDICTING FLIGHT DELAYED TIME .....	57
4.3.1 Stage One: One Year worth of Data Records.....	57
4.3.2 Stage Two: Five Years' worth of Data records .....	60
4.3.3 Stage Three: Balanced Data Sample .....	62
4.3.4 Stage Four: Hyperparameter Tuning.....	64

4.3.5 Stage Five: Introducing Convolutional Neural Network (CNN) .....	65
4.4 SUMMARY .....	67
<b>CHAPTER 5: CONCLUSION.....</b>	<b>68</b>
5.1 OVERVIEW .....	68
5.2 CLASSIFICATION RESULTS .....	68
5.3 REGRESSION RESULTS.....	72
5.4 FUTURE ENHANCEMENTS.....	74
5.5 SUMMARY .....	74
<b>REFERENCES.....</b>	<b>75</b>
<b>APPENDIX .....</b>	<b>78</b>
APPENDIX A - EXTERNAL LIBRARIES FOR PROGRAMMING.....	78
APPENDIX B – DATASET EXPLORATORY ANALYSIS .....	78
APPENDIX C – FLIGHT DELAY CLASSIFICATION, STAGE ONE.....	80
C.1 Experiment One – NN Learning Curve .....	80
C.2 Experiment Two – NN Learning Curve .....	80
C.3 Experiment three - NN Learning Curve .....	80
APPENDIX D - FLIGHT DELAY CLASSIFICATION, STAGE TWO .....	81
D.1 Experiment One – NN Learning Curve.....	81
D.2 Experiment Two – NN Learning Curve .....	81
D.3 Experiment Two – Min-Max Normalization Results .....	81
APPENDIX E – FLIGHT DELAY CLASSIFICATION, STAGE THREE .....	82
E.1 Experiment One – NN Learning Curve .....	82
E.2 Experiment Two – NN Learning Curve.....	82
E.3 Experiment Two – Min-Max Normalization Results .....	82
APPENDIX F – FLIGHT DELAY CLASSIFICATION, STAGE FOUR .....	83
F.1 Experiment – Learning Curve .....	83
F.2 Experiment - ROC Curve .....	83
APPENDIX G - FLIGHT DELAY CLASSIFICATION, STAGE FIVE.....	84
G.1 Sequential NN – Learning Curve – Unbalance Dataset .....	84
G.2 Convolutional NN – Learning Curve – Unbalanced Dataset .....	84
G.3 Sequential NN – Learning Curve – Balance Dataset .....	85
G.4 Convolutional NN – Learning Curve – Balance Dataset .....	85
APPENDIX H - FLIGHT DELAY CLASSIFICATION, STAGE SIX.....	86
H.1 Sequential NN – Learning Curve .....	86
APPENDIX I – FLIGHT DELAY REGRESSION – STAGE ONE .....	86
I.1 Experiment One - Sequential NN .....	86
I.2 Experiment Two – Sequential NN .....	87
I.3 Experiment Three – Sequential NN .....	87
APPENDIX J – FLIGHT DELAY REGRESSION – STAGE TWO.....	88
J.1 Experiment One - Sequential NN .....	88



J.2 Experiment Two – Sequential NN .....	88
J.3 Experiment Two – Min-Max Normalization .....	88
APPENDIX K – FLIGHT DELAY REGRESSION – STAGE THREE .....	89
K.1 Experiment One - Sequential NN .....	89
K.2 Experiment Two – Sequential NN.....	89
K.3 Experiment Two – Min-Max Normalization.....	89
APPENDIX L – FLIGHT DELAY REGRESSION – STAGE FOUR .....	90
L.1 Experiment - Sequential NN .....	90
APPENDIX M – FLIGHT DELAY REGRESSION – STAGE FIVE.....	90
M.1 Experiment – Sequential NN .....	90
M.2 Experiment – Convolutional NN .....	91

# List of Figures

FIGURE 1 - CATEGORIES OF DELAYS BY THE YEAR .....	6
FIGURE 2 - SHARE OF WEATHER DELAY AS A PERCENTAGE OF TOTAL DELAY IN MINUTES .....	7
FIGURE 3 - PROCESS OF BUILDING PREDICTIVE MODELS .....	13
FIGURE 4 - FEATURES AND DATA TYPES .....	16
FIGURE 5 - HISTOGRAM OF DATA ATTRIBUTES .....	20
FIGURE 6 - CONFUSION MATRIX .....	26
FIGURE 7 - ROC CURVE EXPLAINED .....	28
FIGURE 8 - CLASSIFICATION, ROC CURVE FOR STAGE ONE EXPERIMENT ONE .....	35
FIGURE 9 - CLASSIFICATION, ROC FOR STEP ONE EXPERIMENT TWO.....	37
FIGURE 10 - ROC FOR STEP ONE EXPERIMENT THREE .....	39
FIGURE 11 - CLASSIFICATION, ROC CURVE FOR STAGE TWO EXPERIMENT TWO .....	42
FIGURE 12 - CLASSIFICATION, ROC CURVE FOR STAGE TWO EXPERIMENT TWO .....	44
FIGURE 13 - CLASSIFICATION, ROC CURVE FOR STAGE THREE EXPERIMENT ONE.....	46
FIGURE 14 - CLASSIFICATION, ROC CURVE FOR STAGE THREE EXPERIMENT TWO .....	48
FIGURE 15 - CLASSIFICATION, ROC CURVE FOR THE STAGE FIVE EXPERIMENT ONE .....	52
FIGURE 16 - CLASSIFICATION, ROC CURVE FOR STAGE FIVE EXPERIMENT.....	54
FIGURE 17 - CLASSIFICATION, ROC CURVE FOR THE STAGE SIX EXPERIMENTS.....	56
FIGURE 18 - AVERAGE DEPARTURE DELAY BY MONTH IN YEAR 2018.....	78
FIGURE 19 - AVERAGE DEPARTURE DELAY BY TIME OF THE DAY IN YEAR 2018.....	78
FIGURE 20 - AVERAGE DEPARTURE DELAY BY THE DAY OF THE MONTH IN YEAR 2018.....	79
FIGURE 21 - AVERAGE DEPARTURE DELAY BY THE DAY OF THE WEEK IN THE YEAR 2018 .....	79
FIGURE 22 -AVERAGE DELAY BY THE CARRIER IN THE YEAR 2018.....	79

# List of Tables

TABLE 1 - FLIGHT DATA RECORDS .....	15
TABLE 2 - CLIMATE DATA RECORDS .....	15
TABLE 3 - DATASET DESCRIPTION .....	16
TABLE 4 - FLIGHT RECORDS ORIGINATED FROM ORD .....	17
TABLE 5 - DELAYED AND NON-DELAYED FLIGHTS .....	17
TABLE 6 - BALANCED DATASET .....	18
TABLE 7 - TRAINING, TESTING AND VALIDATION DATASET DISTRIBUTION .....	19
TABLE 8 - CLASSIFICATION, DATASET DISTRIBUTION FOR STAGE ONE EXPERIMENTS .....	33
TABLE 9 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS FOR STAGE ONE EXPERIMENT ONE .....	34
TABLE 10 - CLASSIFICATION, PERFORMANCE MATRICES FOR CLASSIFICATION STAGE ONE EXPERIMENT ONE .....	34
TABLE 11 - CLASSIFICATION, CONFUSION MATRIX FOR STAGE ONE EXPERIMENT ONE .....	35
TABLE 12 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS FOR STAGE ONE EXPERIMENT TWO .....	36
TABLE 13 - CLASSIFICATION, PERFORMANCE METRICS FOR STAGE ONE EXPERIMENT TWO .....	36
TABLE 14 - CLASSIFICATION, CONFUSION MATRIX FOR STEP ONE EXPERIMENT TWO .....	37
TABLE 15 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS FOR STAGE ONE EXPERIMENT THREE .....	38
TABLE 16 - CLASSIFICATION, PERFORMANCE METRICS FOR STAGE ONE EXPERIMENT THREE .....	38
TABLE 17 - CLASSIFICATION, CONFUSION MATRIX FOR STEP ONE EXPERIMENT THREE .....	39
TABLE 18 - CLASSIFICATION, DATASET DISTRIBUTION FOR CLASSIFICATION STAGE TWO EXPERIMENTS .....	40
TABLE 19 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS FOR STAGE TWO EXPERIMENT ONE .....	40
TABLE 20 - CLASSIFICATION, PERFORMANCE METRICS FOR STAGE TWO EXPERIMENT ONE .....	41
TABLE 21 - CLASSIFICATION, CONFUSION MATRIX FOR STAGE TWO EXPERIMENT ONE .....	41
TABLE 22 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS FOR STAGE TWO EXPERIMENT TWO .....	42
TABLE 23 - CLASSIFICATION, PERFORMANCE METRICS FOR STAGE TWO EXPERIMENT TWO .....	43
TABLE 24 - CLASSIFICATION, CONFUSION MATRIX FOR STAGE TWO EXPERIMENT TWO .....	43
TABLE 25 - CLASSIFICATION, DATASET DISTRIBUTION FOR CLASSIFICATION STAGE THREE EXPERIMENTS .....	44
TABLE 26 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS FOR STAGE THREE EXPERIMENT ONE .....	45
TABLE 27 - CLASSIFICATION, PERFORMANCE METRICS FOR STAGE THREE EXPERIMENT ONE .....	45
TABLE 28 - CLASSIFICATION, CONFUSION MATRIX FOR STAGE THREE EXPERIMENT ONE .....	46
TABLE 29 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS FOR STAGE THREE EXPERIMENT TWO .....	47
TABLE 30 - CLASSIFICATION, PERFORMANCE METRICS FOR STAGE THREE EXPERIMENT TWO .....	47
TABLE 31 - CLASSIFICATION, CONFUSION MATRIX FOR STAGE THREE EXPERIMENT TWO .....	48
TABLE 32 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS BEFORE TUNING .....	49
TABLE 33 - CLASSIFICATION, PERFORMANCE BEFORE HYPERPARAMETER TUNING .....	49
TABLE 34 - CLASSIFICATION, CONFUSION MATRIX BEFORE HYPERPARAMETER TUNING .....	50
TABLE 35 - CLASSIFICATION, HYPERPARAMETERS FOR TUNING .....	50
TABLE 36 - CLASSIFICATION, PERFORMANCE AFTER HYPERPARAMETER TUNING .....	50
TABLE 37 - CLASSIFICATION, CONFUSION MATRIX AFTER HYPERPARAMETER TUNING .....	50
TABLE 38 - CLASSIFICATION, DATASET DISTRIBUTION FOR STAGE FIVE EXPERIMENT .....	51
TABLE 39 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS FOR STAGE FIVE EXPERIMENT ONE .....	51
TABLE 40 - CLASSIFICATION, PERFORMANCE METRICS FOR STAGE FIVE EXPERIMENT ONE .....	52

TABLE 41 - CLASSIFICATION, CONFUSION MATRIX FOR STAGE FIVE EXPERIMENT ONE.....	52
TABLE 42 - CLASSIFICATION, TRAINING AND TESTING PARAMETERS FOR STAGE FIVE EXPERIMENT TWO .....	53
TABLE 43 - CLASSIFICATION, PERFORMANCE METRICS FOR STAGE FIVE EXPERIMENT TWO .....	53
TABLE 44 - CLASSIFICATION, CONFUSION MATRIX FOR STAGE FIVE EXPERIMENT TWO .....	54
TABLE 45 - CLASSIFICATION, DELAYED AND NON-DELAYED FLIGHTS AFTER DELAY THRESHOLD CHANGED .....	55
TABLE 46 - CLASSIFICATION, PERFORMANCE METRICS FOR STAGE SIX EXPERIMENTS.....	55
TABLE 47 - CLASSIFICATION, CONFUSION MATRIX FOR STAGE SIX EXPERIMENTS .....	55
TABLE 48 - REGRESSION, TRAINING AND TESTING PARAMETERS FOR STAGE ONE EXPERIMENT ONE .....	57
TABLE 49 - REGRESSION, PERFORMANCE METRICS FOR STAGE ONE EXPERIMENT ONE.....	58
TABLE 50 - REGRESSION, TRAINING AND TESTING PARAMETERS FOR STAGE ONE EXPERIMENT TWO .....	58
TABLE 51 - REGRESSION, PERFORMANCE METRICS FOR STAGE ONE EXPERIMENT TWO .....	58
TABLE 52 - REGRESSION TRAINING AND TESTING PARAMETERS FOR STAGE ONE EXPERIMENT THREE .....	59
TABLE 53 - REGRESSION PERFORMANCE METRICS FOR STAGE ONE EXPERIMENT THREE .....	59
TABLE 54 - REGRESSION, TRAINING AND TESTING PARAMETERS FOR STAGE TWO EXPERIMENT THREE.....	60
TABLE 55 - REGRESSION, PERFORMANCE METRICS FOR STAGE TWO EXPERIMENT ONE .....	60
TABLE 56 - REGRESSION, TRAINING AND TESTING PARAMETERS FOR STAGE TWO EXPERIMENT TWO .....	61
TABLE 57 - REGRESSION, PERFORMANCE METRICS FOR STAGE TWO EXPERIMENT TWO .....	61
TABLE 58 - REGRESSION, TRAINING AND TESTING PARAMETERS FOR STAGE THREE EXPERIMENT ONE .....	62
TABLE 59 - REGRESSION, PERFORMANCE METRICS FOR STAGE THREE EXPERIMENT ONE .....	62
TABLE 60 - REGRESSION TRAINING AND TESTING PARAMETERS FOR STAGE THREE EXPERIMENT TWO .....	63
TABLE 61 - REGRESSION PERFORMANCE METRICS FOR STAGE THREE EXPERIMENT TWO .....	63
TABLE 62 - REGRESSION, TRAINING AND TESTING PARAMETERS BEFORE TUNING.....	64
TABLE 63 - REGRESSION, PERFORMANCE RESULT BEFORE TUNING .....	64
TABLE 64 - HYPERPARAMETERS FOR OPTIMIZATION .....	65
TABLE 65 - PERFORMANCE RESULTS AFTER TUNING .....	65
TABLE 66 - REGRESSION, DATASET DISTRIBUTION FOR STAGE FIVE EXPERIMENT .....	65
TABLE 67 - REGRESSION TRAINING AND TESTING PARAMETERS FOR STAGE 5 EXPERIMENT .....	66
TABLE 68 - REGRESSION PERFORMANCE METRICS FOR STAGE FIVE EXPERIMENTS.....	66
TABLE 69 - STAGE TWO AND STAGE THREE EXPERIMENTS.....	68
TABLE 70 - CONFUSION MATRIX FOR STAGE TWO AND STAGE THREE EXPERIMENTS.....	69
TABLE 71 - PERFORMANCE RESULTS FOR STAGE FOUR.....	69
TABLE 72 - STAGE FOUR CLASSIFICATION PARAMETER TUNING.....	69
TABLE 73 - PERFORMANCE RESULT FOR CNN FOR BALANCE AND UNBALANCED DATASETS .....	70
TABLE 74 - REDUCED DELAY THRESHOLD RESULTS .....	70
TABLE 75 - 15 MIN VS 5 MIN THRESHOLD CONFUSION MATRIX .....	71
TABLE 76 - LITERATURE REVIEW FINDINGS FOR CLASSIFICATION USING FLIGHT AND WEATHER DATA.....	71
TABLE 77 - LITERATURE REVIEW FINDINGS FOR CLASSIFICATION USING ONLY FLIGHT DATA.....	72
TABLE 78 - REGRESSION HIGHEST PERFORMANCE FROM STAGE ONE TO THREE EXPERIMENTS.....	73
TABLE 79 - REGRESSION RESULTS AFTER HYPERPARAMETER TUNING.....	73
TABLE 80 - REGRESSION CNN RESULTS .....	73
TABLE 81 - LITERATURE REVIEW FINDINGS FOR REGRESSION .....	73

# List of Abbreviations

**AI** – Artificial Intelligence

**ANN** – Artificial Neural Network

**BTS** – Bureau of Transportation Statistics

**CNN** – Convolutional Neural Network

**FAA** – Federal Aviation Administration

**FPR** – False Positive Rate

**IATA** - International Air Transport Association

**ML** – Machine Learning

**NAS** – National Aviation System

**NN** – Neural Network

**PC** – Personal Computer

**SVM** – Support Vector Machine

**TDI** – Total Delay Impact

**TPR** – True Positive Rate

# Chapter 1: Introduction

## 1.1 Overview

Airport is one of the fastest transportation portals into a country. Providing qualitative service while handling and servicing passengers is a challenging task for the stakeholders at an airport. To be successful commercially airports need to figure out their business components. Bogicevic et al [1] mentioned that real time information sharing while managing the disruptions thus creates the ideal airport experience resulting high level of passenger satisfaction.

The current airport systems should be intelligent enough to cater these business components. Such airport system consists with board research areas such as improving revenue, efficiency, passenger experience and security. Under these components, predicting delays and improving efficiency by reducing delays on airport should come as a high priority features for such intelligent solution.

## 1.2 Research Background

The world population was 1.6 billion when the Orville Wright piloted a plane hundred years ago but today there are around seven billion people in the world [2]. International Air Transport Association (IATA) noted that nearly 3.3 billion people traveled by air in the year of 2014 and they estimate by the year 2034, 7.3 billion passengers will be waiting in lines to experience the air travel worldwide [3].

Having a few minutes delay in a flight results in major consequences. In economic perspective there will be cancellation and missed connections. Airport congestion will happen causing chained effects to the airport schedules. Environmentally there will be a lot of fuel wastage and socially loss of productivity. The battle group and NEXTOR universities was sponsored by the Federal Aviation Administration (FAA) to conduct research on the flight delays occurs at United States and provide a report on the Total Delay Impact (TDI) [4]. This report includes detailed analysis of increased costs for airlines, passengers and the cost due to loss of demand. Finally, it summarizes the direct and indirect impact of delay on the US economy.

32.9 billion dollars was estimated by the TDI as the total cost of US air travel delays in the year of 2007. Nearly 8.3 billion dollars was added to the expenses of the airline companies to handle the increased expenses of crew members, fuel wastage and maintenance. The passengers had to waste around 16.7 billion dollars to the time they lost due to delays and cancellations. There was also 3.9 billion dollars loss by the passengers who lost their will to travel by air due the lasting flight delays [4].

### **1.3 Motivation**

As mentioned in the section 1.2 research background, the author has his personal experience in related to flight delays. The flight which the author supposed to take from Dubai to United Kingdom in 2016, got delayed by two hours due to a sandstorm. Therefore, the transit three hours had to extend for five hours. The author's personal experience was the motivation for this research.

### **1.4 Goal**

The goal of the current research would be to investigate, design, implement and evaluate the predictive models which can be used to classify a given flight into delayed and non-delayed classes. And also, to predict the delay time of a flight in minutes.

Further elaborating the mentioned the goal, multiple data resources which are available inside and outside of the airports will be used as the data sources for the predictive models.

### **1.5 Objective of the Study**

Objectives must be achieved to reach the author's goal. Those objectives can be further elaborated as below,

- The author needs to figure out on how to conduct a review on classification and regression methods with regard to the domain of the problem.
- Develop predictive models that can be used to forecast flight delays.
- Evaluate the predictive model using a justified evaluation criterion.
- Produce a final thesis based on the results.

## **1.6 Technology Domain**

This current research focuses on currently trending areas of computer science which are data mining and Machine Learning (ML). Under this section, a brief explanation is carried out on these fields in order for the reader to understand the background of the research.

### **1.6.1 Machine Learning**

Even though ML is not a field that stands by itself, it is a subfield of artificial intelligence (AI). The aim of ML is to understand the structure of the data by infusing them into predictive models which were generated by the ML algorithms. These predictive models can interpret the patterns in the datasets to a human understandable output. ML differs from the traditional computing approach by allowing the computers to learn. Training on input data makes the learning algorithms to identify the underlying patterns and do an analysis on them while providing a prediction as an output. This process gives advantage for robust decision making [5].

### **1.6.2 Classification and Regression**

Supervised and unsupervised ML are two of the widely used techniques in this vast field. Supervised algorithm can be used in the scenario where the target values are known for the data records. If the target values are unknown, then unsupervised algorithms can be used.

Classification and regression techniques are categorized under the same umbrella of supervised machine learning. Attempt to estimate the mapping function from the input features to discrete or categorical output can be considered as classification. If the mapping function estimates the input features into numerical or continuous output, then that particular task can be considered as a regression technique.



## **1.7 Scope**

The study is based on the airline departure delays at an airport. As stated by the Bureau of Transportation Statistics (BTS), there are nearly five thousand airports available to the public in United States [6]. Out of these airports, the author decides to choose one of the busiest airports to conduct his research. Five years' worth of data records will be considered as the data source for the research.

## **1.8 Structure of the Dissertation**

The chapters of this dissertation have been organized based on the approach taken by the author understand the problem and implement a necessary solution. The first chapter is totally focused on introducing the reader to the research problem, its background and scope of the research.

The second chapter literature review provides an extensive research finding regarding the focused problem. This chapter notes down the cases for the flight delays and the available research carried out to find the on-time performance of flights.

The third chapter provides the detailed explanation on the methodology which the author will follow to conduct the research. Starting from the datasets, preprocessing techniques, learning algorithms, evaluation techniques and different experiments which are planned to conduct will be discussed.

The fourth chapter evaluation will provide the evaluation results for the different predictive models implemented by the author based on the experiments mentioned in the third chapter methodology. Each predictive model will be analyzed using the evaluation techniques mentioned in the methodology chapter.

The final chapter draws the conclusions for the current research by providing the research contributions and findings. The author will also be noted down the future enhancements which can be done to further extend the research.

## Chapter 2: Literature Review

The Chapter One: Introduction presented about the background, problem domain, motivation of the author and the scope of the research. This chapter will provide an in depth analysis of delays happens at airport. It will also cover the current research which have been done by the research community regarding predicting and minimizing flight delays in great detail.

### 2.1 Flight Delays

Managing flight delays and its accumulated impact is a challenging task for the airlines and airport executives. Having a crystal ball forecast is not enough. A proper systematic approach should be in place to provide the necessary insights into flight delays for the executives to act and for passengers to make decisions.

#### 2.1.1 Basis for Delayed Flight Operations

The United States have a system in place for the airlines to report the causes for the delays and any valuable information regarding the particular delay. A flight which operates fifteen or more minutes later than the schedule is considered as a delayed flight according to the United States Department of transportation [7]. They presents the information regarding the reported delays in board categories which can be found in the list below.

**Air carrier delays:** The circumstances within the airline which cases the particular cancelation or delay of a flight. Examples being cleaning, fueling, maintenance or crew problems can be taken as reasons.

**Late-arriving aircraft:** This type of delay happens when the previous flight's delay causing the current flight to depart behind the schedule.

**Delays due to extreme weather conditions:** Any notable meteorological conditions which was actually happening or forecasted by the weather departments that makes the airlines prevent flying. The decision is up to the airline carrier to operate in such conditions as blizzards, hurricanes or tornadoes.

**National Aviation System (NAS):** Any cancellations or delays caused by the US NAS is represented by this category. Airport operation delays, non-extreme weather conditions and air traffic control can be taken as examples.

**Security:** This category represents the delays happens due to security concerns in an airport. Evacuations, re-boarding passengers to different terminal due to security breaches are some examples.

Figure 1 depicts the categories of delays mentioned above by year as a percent of total delay in minutes [7].

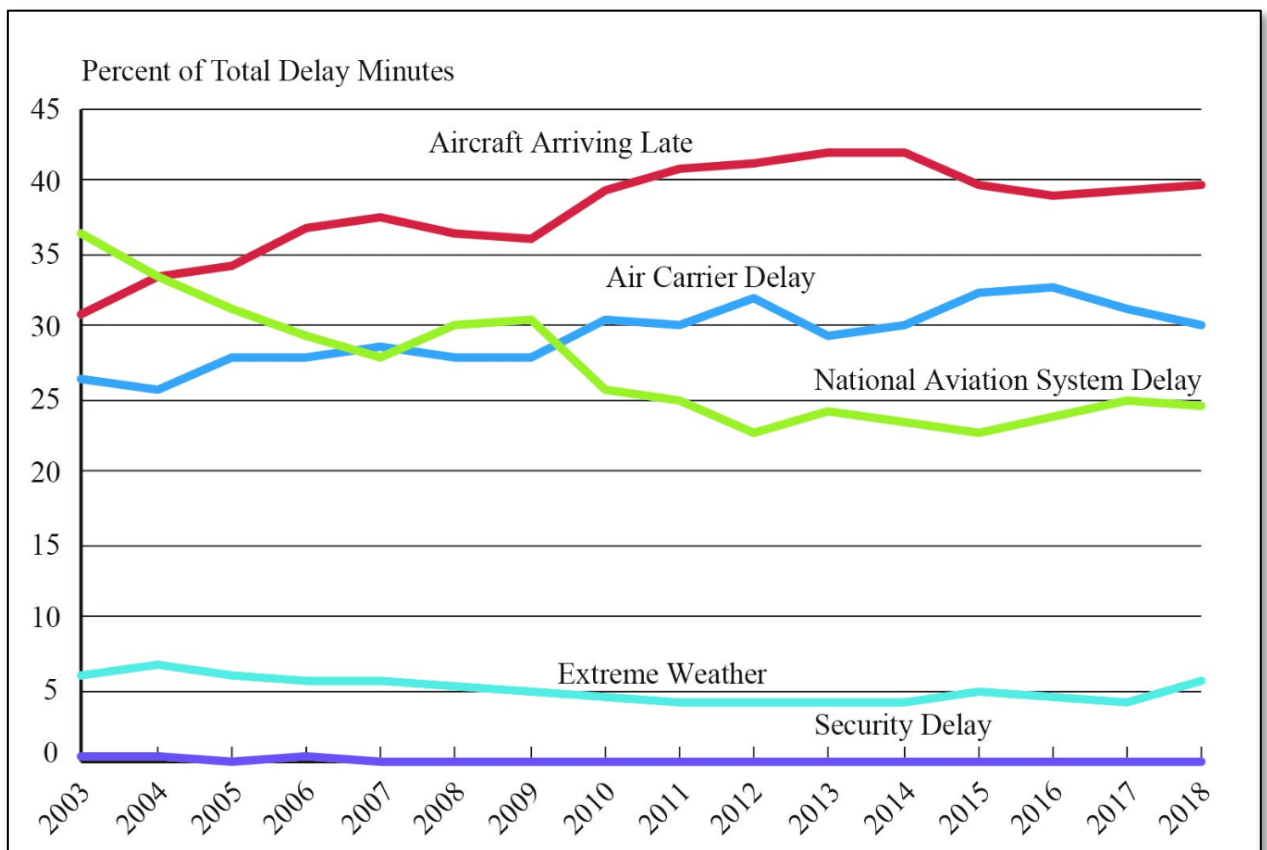


Figure 1 - Categories of delays by the year

## 2.1.2 Flight Delays Caused by Weather Conditions

Extreme weather conditions which are mentioned in Figure 1, prevent airlines from their flight operations. As mentioned in the above flight delay causes, NAS has a category of delays related to non-extreme weather conditions which does not prevent flying but slows down the operations. These types of delays can be minimized the appropriate actions and decision taken by the authorities. 55% of NAS delays in 2018 were due to non-extreme weather situations and it is 24.5% of total delays to 2018.

To understand the big picture of total flight delays which happens due to weather conditions require couple of steps. Delays due to extreme weather conditions need to be combined with the NAS weather delays mentioned above. Even though the late arriving aircrafts do not report the reasons for arrival delay, but a proportion can be identified based on the weather delays and total number of flights in the other categories. Based on this a calculation need to be done to identify the weather delays in late arriving aircraft category. Finally, all the mentioned weather-related delays combined results in total weather's share of flight delays which is shown the Figure 2 [7].

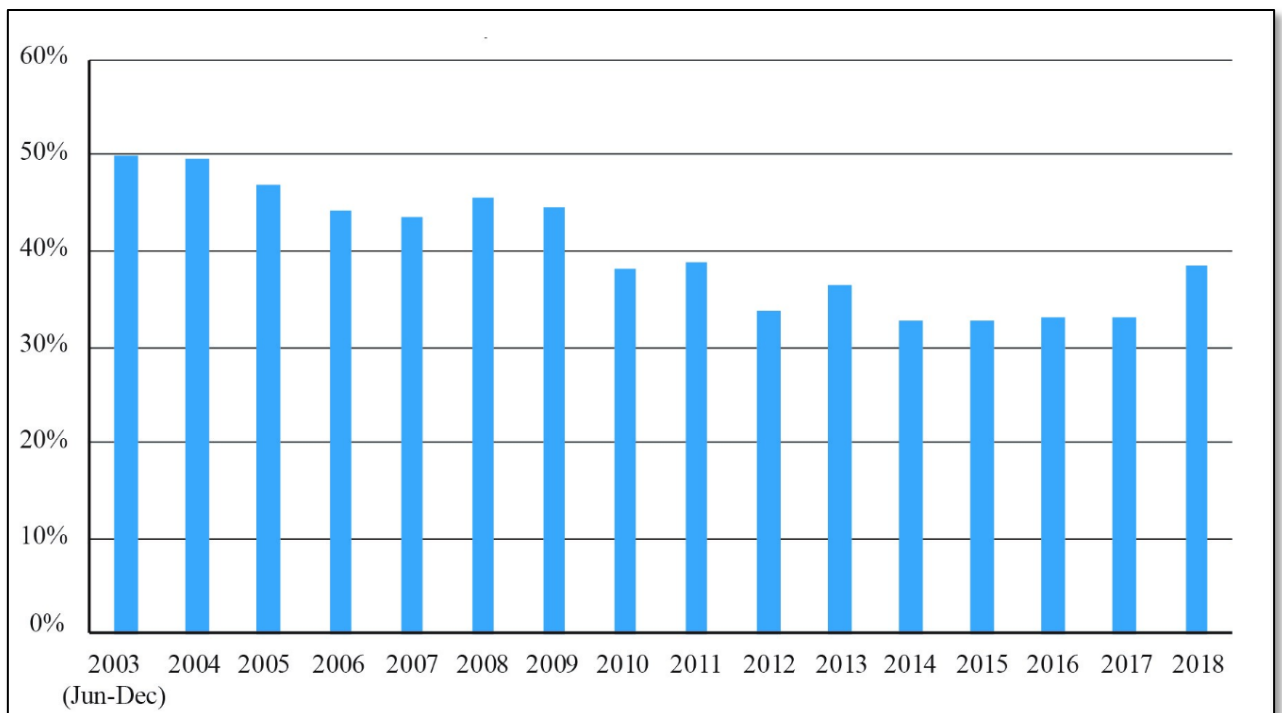


Figure 2 - Share of weather delay as a percentage of total delay in minutes

## **2.2 Related research for flight delay prediction**

The research based on predicting flight delays can be divided into two sections which are classifying the flight delays and measuring the particular delay. These approached researches are evaluated in the following sections.

### **2.2.1 Flight Delay Classification**

Cabanillas et.al [8] researched the weather conditions which has an impact on the on-time performance of a flight. The predictive models they have built were infused with both historical flight and weather data. Weather dataset was taken from the Meteorological Aerodrome Report (METAR). It contains weather information gathered from airports in every thirty minutes. They have used data from 2005 to 2008 containing 869 records, in a single route of a flight and tried to analyze the weather and flight condition in a specific point of time in the particular route. They have identified as light and heavy rain has least effect on flights while haze or fog, thunderstorms, light snow and snow have extreme effects. As for a limitation, they have not considered the influence of the wind.

Mathur et.al [9] did their research trying to predict whether a flight would get delayed or not by using the arrival and departure data related to flights. Statistical classification models like Random Forest, SVM, Logistic Regression and Naïve Bayes were taken into consideration. These models were infused with 12000 training data sample and 1200 testing samples. Weather data was in cooperated with the flight data records. Temperature, visibility, precipitation level as a binary variable and severity as a scale for rain, thunderstorms and fog were considered as weather parameters. They have tried multi class classification using three classes by categorizing the delay. Overall this research having an imbalance dataset, a smaller number of data records have achieved around 90% accuracy for all the predictive models.

Bandyopadhyay et.al [10] defines three goals for their research. First, they try to identify the factors which influence delays using Linear Regression model. Secondly, they try to do a classification to identify weather a flight gets delayed. For this task Naïve Bayes, SVM and Random Forest models were used. It was noted the research that the Naïve Bayes performs well overall and SVM models consumed a lot of time to train. The datasets were taken from the Bureau of Transportation Statistics (BTS). Weather data was taken from the weather

underground API. Temperature, humidity, wind speed, snow, hail, thunder, rain and tornado warnings were taken as the features from the weather dataset. Different Training and testing samples were used for the experiment starting from 500 to 80000 samples. The third goal is to predict the delay time using Linear Regression. This will be discussed in the next section 2.3.2.

Nathalie Kuhn and Navaneeth Jamadagni [11] has applied statistical and deep learning techniques to predict if a flight's arrival will be delayed or not. Decision trees and Logistic Regression were used as statistical models to perform this classification task. A neural network constructed with a single hidden layer with four neurons. Year 2015 data form BTS were used and a sample of hundred thousand record were constructed. Thirteen features including flight details, scheduled departure and arrival, tax-out time were used as features. Weather data was not incorporated to this research. Overall, the models were performed around 91% accuracy. As for an improvement, more training data needs to be incorporated to a to achieve more precise prediction because the dataset consists of all the airports and flight routes. Even though they have hundred thousand samples, to predict arrival delay of individual airports, the present data was not sufficient.

Sruti et.al [12] tires to explore the features which influence the flight delays along with the intensity of the delay. The developed models are being applied to predict the occurrence of flight delays at airports making the problem a multi class classification model. They classify delays into the classes specified the Figure 1. They have calibrated their model with the delay causes which is mentioned in the "Causes for flight delays section". Weather data was also incorporated to the improve the model. OneR algorithm out of naïve bayes and IBK algorithms has proven to be much more accurate.

Neural and deep belief convolutional network concepts have been used by Venkatesh et.al [13] to estimate flight delays in their research. They have used stratified sampling method to generate a sample of two hundred thousand records as the entire dataset. The neural network contained one hidden layer containing 3 neurons which results in 92% accuracy. Convolutional Neural Network (CNN) had four layers. Each layer had 6,5,4,1 neuron respectively resulting around 77% accuracy. The neural network has outperformed the CNN in this case, but the researches have mentioned that if more data was in cooperated to the deep belief network it's accuracy can be further improved. The research was not in cooperated with weather data.

Zhou et.al [14] tried to analyze the temporal patterns of the Aviation Network while creating a predictive model to do multi class classification using decision trees. Three years of data have been taken for the analysis, but the data was filtered down to a single airline flight record. Six chosen destination airports were separately analyzed and overall resulting 80% accuracy model. The delay was divided into four classes labeled 1 to 4. “1” being small delays and “4” being larger delays. Greater the number is more severe the delays. No weather data was incorporated. As a limitation they mentioned that the concentration of small delays and very large delays are high.

### **2.2.2 Measuring Flight Delays**

Sridhar et al.’s [15] research is to forecast the weather related delays at the national, regional and airport levels using FAA’s OPSNET and ASPM datasets. Linear regression and feed forward neural networks were considered for the experiments. Weather data was incorporated with the experiments. Based on the experiments following conclusions were made.

- Depending on the seasons, the use of different type of predictive models will result in better accuracy.
- Neural network models perform much better than the linear regression models.

Bandyopadhyay and Guerrero’s classification research was discussed in the previous section 2.3.1. They did experiments using Linear Regression to predict the flight delay time. Experimentations were done using a small dataset sample, 4500 training and 1500 testing data. Generalized Linear Regression model were used to conduct the experiment resulting a 37 minutes Mean Absolute Error (MAE). By performing locally weighted quadratic regression the MAE reduced to 35 Minutes [10].

Rebollo and Balakrishnan [16] presented the predictive models using Random Forest algorithm to predict the departure delays of the most influential airports. Flight data from 2007 to 2008 with weather data and runway configurations infused to the delay model. They have used the datasets from the Federal Aviation Administration’s ASPM database. Ten training data samples containing three thousand records and ten testing data samples containing thousand records for the experiments. Over sampling technique was used to generate a balance dataset. Delays more than one hour was not considered for the experiment. The results contain 19% error rate for classification and 21 minutes as medium error for predicting delay time.

## **2.3 Research Gap**

The current research is based on classifying flight delay as a binary classification problem and predicting flight delay in minutes as a regression problem. Since weather is an influential aspect for the flight delays, the adding weather data to the predictive models will be considered. As for the current investigation most of the research was carried out on the classification problem. There is less amount of research to quantify the delay that happens with an acceptable accuracy. It seems to gain better accuracy, the models need to be infused with more data, that means flight records and much more weather information. Since the percentage of weather-related delays are considerably less compared to the total delays, it's appropriate to think about a sampling technique that can capture and generate a training data sample which can provide an unbiased result to the research.

Therefore, the research will focus on a larger data datasets, with balanced and unbalanced data samples to hoping to generate predictive modals which have better performance than the investigated literature.

## **2.4 Summary**

This chapter provides the in-depth analysis of the current research carried out related to the addressed problem. Basis for delayed flight operations were discussed in great detail identifying delay categories. Then the current research problem was divided into two tasks. Classification of flights into delayed and non-delayed classes. Regression of flight delays resulting a predictive model which can output delay in minutes.



# Chapter 3: Methodology

## 3.1 Overview

The methodology is the approach on how the research will be carried out. It will use the information gathered in Chapter One: Introduction and the knowledge gained in Chapter Two: Literature Review to come up with an adequate research methodology. As mentioned in the chapter one, the domain of this research extends on multiple fields in ML. Therefore, many experiments will be carried out to meet the goals of the research.

## 3.2 Representation of the Problem

When the passenger provides his or her flight details to the system, the underlying predictive models should be able to forecast the following information.

- Whether the flight gets delayed or not.
- The predicted delay times.

Since delay can be identified in the dataset as a target variable, this problem will be considered as a supervised machine learning task. Binary classification will be used to classify the flights into delayed or non-delayed classes. Predicting the flight delay time will be considered as a regression problem since a value needs to be predicted rather than a class. More specifically predicting delay time will be considered as a multiple regression task since there are more than one independent feature to make the prediction on the dependent delay variable.

Multiple data sources and different machine learning techniques that was found in the Chapter Two: Literature Review will be considered to build the models. The methodology will be broken down to different phases, which will be discussed in the next section 3.3. To gain better accuracy over the results many experiments will be conducted. As for the final output a classification and a regression predictive model will be chosen based on the performance from the experiments to forecast the above mentioned information.

### 3.3. Methodology for Building Predictive Models

Building a predictive model involves multiple steps ranging from different techniques, therefore the process is complicated. The following workflow diagram (Figure 3 [17]), shows the different steps in building a predictive model. Each of these steps will be thoroughly discussed on how they were accommodated to tackle the current research problem.

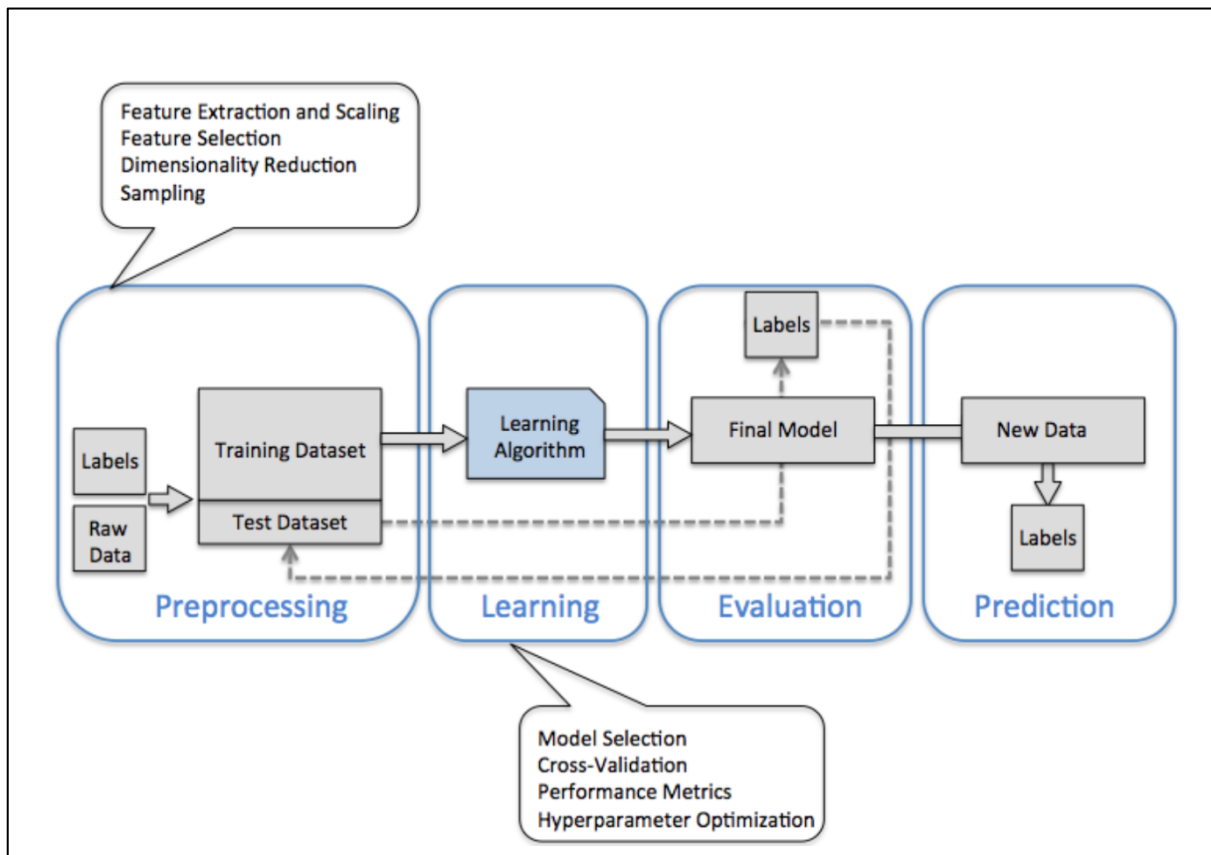


Figure 3 - Process of building predictive models

### 3.4 Infrastructure and platform

This sub section explains the reasons for selecting the particular infrastructure for the current research. It is known that in most cases the ML algorithms, libraries and tools require high performing hardware utilities for the calculation tasks.

### **3.4.1 PC Hardware**

The author will be using his personal computer for the implementation of the predictive models. The computer consists with 2.8 GHz Intel Core i7 processor with 16GB memory which is running on macOS Mojave. Even though the selected datasets are large in size, the available processing power and memory seems to be enough to handle the workload.

### **3.4.2 Selection of Tools and Programming Language**

Python was considered as the programming languages to be used for the implementation. Apart from being an easy to learn programming language, Python consists of a rich library which is dedicated for bulk data processing. It is very much suited to process Big Data and python is the most fitting programming language for high performance computing.

An open source python distribution for scientific computing called “Anaconda” was selected as the package management tool. Along with this tool “TensorFlow”, an open source machine learning platform was selected to develop the neural networks. Please check Appendix A to see the list of external libraries used for the implementation.

## **3.5 Preprocessing**

The main focus of this sub section is to give a detailed explanation about the datasets and the data pipeline which feeds the data to the learning algorithms. A data pipeline is a sequence of data processing components which manipulates and transforms the data to an expected format to be passed on to the learning phase [18]. Below sections explains the different components of the data pipeline used for the current research problem.

### **3.5.1 Datasets**

Reporting carrier on-time performance dataset, contains the commercial airline operations in the United States. The airline carriers are required to maintain on time data for the flights they operate, and these data can be downloaded from the Bureau of Transportation Statistics (BTS) [19].

Currently on time performance data are available from year 1987 to 2020 containing more than 100 million records. For the current research, complete flight records from the year 2014 to 2018 are used. At the moment, the records for the year 2019 and 2020 were not completely available, therefore the particular data were not considered. Descriptions regarding the data variables can be found in the Bureau of Transportation Statistics database profile [19]. Table 1 shows the number of flight records per year.

	2018	2017	2016	2015	2014
<b>Flight records</b>	7213446	5674621	5617658	5819079	5819811

Table 1 - Flight data records

Climate related datasets can be found from the National Oceanic and Atmospheric Administration (NOAA) [20]. This dataset contains daily climate records of temperature, precipitation, and snow records over land in United States up to the year 2020. Since the flight records were selected for the year 2014 to 2018, climate data will also be taken from the same year range. US airports have their own weather stations installed and the weather recording collected from the weather stations inside the airports are included in this dataset. Therefore, the two datasets can be mapped based on the airport code. Table 2 shows the number of daily weather summaries available per year.

	2018	2017	2016	2015	2014
<b>Climate records</b>	34588179	34853947	35326545	34899198	34420317

Table 2 - Climate data records

### 3.5.2 Dataset Exploration

Regarding the exploratory analysis, the author will first go through the dataset profile information and filter out most of the obvious unwanted data variables. As an example, the dataset contains these fields names “OriginAirportID”, “Origin” and “OriginAirportSeqId”. Since all these fields refers to the same airport “OriginAirportID” and “OriginAirportSeqId” can be dropped. “Origin” code can be used to map flight and weather datasets.

Then the dataset will be further analysed using different visualisation techniques to better understand the data. Please refer to the Appendix B for more details regarding the visualizations of the dataset features.

Based on the database profile, dataset exploration and literature review, the following data attributes shown in Table 3 will be considered for the current research.

Data Attribute	Description
<b>Flight Data</b>	
Year	Year of the flight record
Month	Month of the flight record
Day	Day of the flight record
Day of the Week	Day of the week, 1 -Monday to 7- Sunday
Hour	Flight departure hour
Distance	Distance between the origin and destination airports
Carrier Code	Unique identifier for airline
Destination	Destination airport
Target	Departure delay of a flight
<b>Weather Data</b>	
Minimum Temperature	Minimum temperature (tenths of degrees °C)
Maximum Temperature	Maximum temperature (tenths of degrees °C)
Precipitation	Precipitation (tenths of mm)
Snowfall	Snowfall (tenths of mm)
Wind speed	Average daily wind speed (tenths of meters per second)

Table 3 - Dataset Description

Figure 4 shows the data types of different data features. This particular information was generated using the year 2018 flight records. The number of entries will be different based on the datasets and they will be discussed in the below sections.

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 325817 entries, 152897 to 132393
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   month                 325817 non-null  int64
1   day                   325817 non-null  int64
2   dow                   325817 non-null  int64
3   hour                  325817 non-null  int64
4   distance              325817 non-null  float64
5   carrier               325817 non-null  object
6   dest                  325817 non-null  object
7   origin_tmin          325817 non-null  int64
8   origin_tmax          325817 non-null  int64
9   origin_prcp          325817 non-null  int64
10  origin_snow           325817 non-null  int64
11  origin_wind           325817 non-null  int64
12  days_from_holiday    325817 non-null  int64
13  target                325817 non-null  int64
dtypes: float64(1), int64(11), object(2)
memory usage: 37.3+ MB

```

Figure 4 - Features and data types

### 3.5.3 Data Cleaning and Filtering

Due to the large number of flight records available in the dataset, the flights originated from O'Hare International Airport (ORD) will be considered from the datasets. It is one of the oldest and the third busiest airport in the united states covering 79.8 million passengers per year [21]. Table 4 shows the number of flight records originated from ORD per year.

	2018	2017	2016	2015	2014
<b>Flight records (ORD)</b>	325817	262741	240122	304938	273582

Table 4 - Flight records originated from ORD

With regards to the dataset undefined, null or empty values will be considered as mission values. ML algorithms cannot work with these values. Therefore, following methods will be carried out to handle those values in the dataset.

- Remove the entire feature column from the dataset.
- Remove the corresponding data rows which has the missing values.
- Set the missing values to some values based on the problem domain (E.g. zero, the mean, the median).

Weather data in the dataset are in a format represented in “tenths of”. In this particular format, the decimal point in the value is removed to convert the value to an integer. Therefore, to get the actual value each data attribute will be divided by 10. As an example, maximum temperature represented in “306 tenths of degrees °C” is actually 30.6 °C [20].

### 3.5.4 Sampling Dataset

Even though the number of flights originated from ORD are relatively high, the delayed flights per year are relatively low. Table 5 shows the delayed and non-delayed flights in their respective percentage compared to the overall flight records per year.

(ORD)	2018	2017	2016	2015	2014
<b>Non-delayed %</b>	79.22	80.82	78.12	76.66	70.65
<b>Delayed %</b>	20.76	19.18	21.88	23.34	29.35

Table 5 - Delayed and non-delayed flights

Table 5 clearly implies that this is an imbalance dataset with regards to the research problem. The models which are infused with this particular data will be biased against delayed flights. Therefore, a balance dataset needs to be constructed. This can be achieved by taking equal portions of delayed and non-delayed flight records per year and concatenating them to single dataset. All the delayed flight records and a random sample of non-delayed flight records will be considered for the balanced dataset. Table 5 show information regarding the constructed balanced dataset.

<b>(ORD)</b>	<b>2018</b>	<b>2017</b>	<b>2016</b>	<b>2015</b>	<b>2014</b>
<b>Non-delayed</b>	258114	212341	187592	233762	193277
<b>Delayed</b>	67647	50400	52530	71176	80305
<b>Sample Dataset (delayed + non-delayed)</b>	135294	100800	105060	142352	160610
<b>Total Sample Data</b>	<b>644116</b>				

Table 6 - Balanced Dataset

### 3.5.5 Introducing New Attribute

Researches have mentioned this feature attribute, the number of days from closest national holiday in US, with the assumption that the holidays will create much more delays because of the congestions happens at airports [22]. The author will introduce the same feature attribute for the current research problem. Federal holidays stated by the US government will be considered for the selected range of time period [23].

### 3.5.6 Training, Validation and Testing Datasets

The data sample which is used to fit the predictive model is called the training dataset. The model observes and learn from the data available in training dataset. Validation dataset provides impartial assessments on the training datasets while on the process of hyperparameter tuning. Test dataset provides unbiased assessments on the tuned model. Test data is used as the standard for the final evaluation of the predictive model.

For the current research problem, training datasets set will be a randomly selected data sample of 80% data records from the total data sample. And the remaining 20% data sample will be considered as the test dataset. 20% random data records will be allocated to the validation dataset from the training dataset.

Dataset	Training	Testing	Validation
Year 2018 dataset	208486	65153	52122
Five Year dataset	900572	281429	225143
Balanced Dataset	412233	128824	103059

Table 7 - Training, testing and validation dataset distribution

Table 7 shows the training, testing and validation distribution for the different data samples used in the research implementation.

### 3.5.7 Handling Text and Categorical Attributes

Categorical variable can be in the form of numerical or textual. These variables can represent data which can be organize into different groups based on a common characteristic. The characteristics can be of two types, nominal or ordinal. Nominals can be assigned a value as a number, but the particular number does not provide any numerical importance. In the current dataset, destination airport code and carrier code can be taken as nominal categorical data. Ordinal data represents values which can be ordered. Even though the dataset specifies month, day, hour and day of the week as numbers, by definition they are categorial data in nature and can be considered as ordinal categorial data.

ML algorithms favours numbers, therefore, these text type of categorial data attributes need to be convert to numbers. One of the simplest methods is to use label encoding which converts each value in the categorial column into a number. But label encoding has the disadvantage that the numerical value can be misinterpreted by the algorithm as having a sort of order in them. This ordering issue can be prevented by using one hot encoding scheme. It is one of the most widely used encoding schemes currently available. One hot encoding can transform the unique values in the categorial column into a binary representation [24].



### 3.5.8 Feature Scaling

Numerical data is expressed not by using natural language but rather in numbers. These data have a meaning as a measurement. Distance, days passed from nearest holiday, Precipitation, wind speed, snowfall, minimum and maximum daily temperature can be considered as numerical data types.

Figure 5 shows the histograms for some of the numerical features present in the dataset. The vertical axis shows the number of instances, while the horizontal axis shows the value range. Based on the vertical axis it is clear that these data attributes are in different scales. ML algorithms do not perform well when the numerical features have different scales.

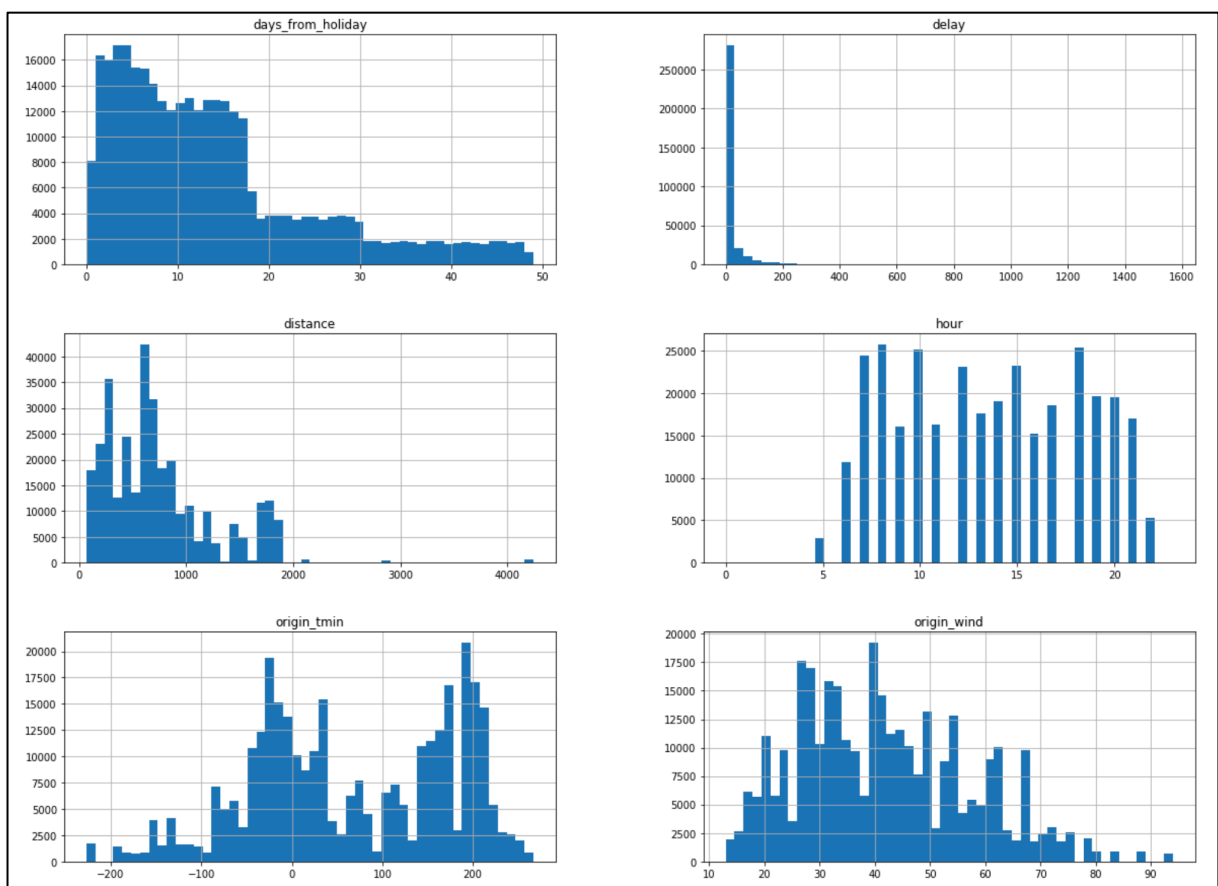


Figure 5 - Histogram of data attributes

The most common ways to get the numerical attributes to the same scale are normalization and standardization. Normalizing make the values rescale so that they end up in the range of 0 to 1. Normalization uses the min-max algorithm by subtracting the minimum value in the column and dividing it by the maximum value minus the minimum value. Standardization, first it subtracts the mean value, and then it divides by the standard deviation. Unlike min-max scaling,

standardization does not bound values to a specific range, which may be a problem for some algorithms [18].

## **3.6 Learning**

The main focus of this sub section is to mention the different learning algorithms considered for the current research problem and give a brief overview of each algorithm. A selection of algorithms mentioned in the chapter two literature review were considered. The results of these algorithms will then be compared and evaluated in the next chapter. Using the data pipeline, which was discussed in the section 3.4 preprocessing, the datasets will be transformed and fed into these learning algorithms resulting a predictive model.

### **3.6.1 Linear Regression**

Linear Regression is an approach to determine the relationships of various parameters on a target value. The linear relationships are based on the coefficients of the parameters and the results of the predictive model [25]. If there is only one parameter, then the approach is called as simple linear regression. If there are more than one parameter, then it is multiple linear regression. For the current research problem, multiple linear regression will be used.

### **3.6.2 Logistic Regression**

Even though the name “regression” appears in its name, Logistic Regression is a classification algorithm under supervised ML. It is used to obtain odds ratio, which is a measure of association between exposure and an outcome, in the presence of more than one features. The process is similar to multiple linear regression, but the only difference is that the response is binomial [26].

### **3.6.3 Decision Trees**

Decision tree is a widely used predictive modelling approach in supervised ML. The tree structure is developed incrementally by dividing the training data into small subsets. The final decision tree contains a root node, internal and leaf nodes [27]. Classification decision trees can take a discrete value as target variable while regression tree’s target variable takes continuous

values. Classification tree's leaves represent the class labels and the branches and internal nodes represents the conjunction of features which lead to the leaf node class label. Same applies for the regression trees as well [28]. Both the classification and regression trees will be considered for the current research problem.

### **3.6.4 Random Forest**

Being an ensemble ML algorithm, Random Forest consists of several decision trees. These decision trees are taken as individual trees to be trained, and predictions of these particular trees are combined through averaging. There are couple of key concepts which is needed to construct this algorithm. The first being the splitting method for the leaves. Most of the time axis aligned splits are used where the data is routed to the sub trees based on a threshold. The next concept is the predictor type used in the leaf nodes. Most of the time the average response over the training points in the particular leaf will be used as a predictor. The next concept is the method for providing the randomness to the trees. There are number of ways to introduce randomness to this algorithm, one being the choice of coefficients for random combinations of features [29]. This algorithm will be used for both classification and regression tasks.

### **3.6.5 Naive Bayes**

The Naïve Bayes classifiers apply Bayes' theorem for their learning with the assumption that the features are independent given a class. It is one of the simplest classifiers, which often performs well in many real-world applications. These classifiers are easy to build and particularly useful for very large data sets as it is highly scalable [30]. Current research implementation uses a large dataset. Therefore, this classifier seems to be useful.

### **3.6.6 Artificial Neural Network (ANN)**

ANN is modelled based on human biological network which consist of interconnected multiple neurons. The biological neurons pass signals to the intermediate neurons based on chemical reactions. This biological process influenced the ANN, and it can be used for both classification and regression tasks.

A neural network consists of neurons which can be called as units. These units are grouped and arranged into layers which converts an input vector to some kind of output. Basic NN architecture consists 3 types of layers. The first being input layer which is designed to take information from the outside world which the network will attempt to learn. Then hidden layers take a weighted input and produces an output based on an activation function. Output layer being the last layer in the NN signals how the NN responds to the information learned [18].

### **3.6.6.1 Feed Forward Neural Network**

In this type of NN, the information travels only forward. Starting from the first layer which is the input layer to the one or multiple hidden layers and finally through the last layer which is the output layer. There are no feedback connections such as output of the NN is feedback into itself.

### **3.6.6.2 Convolutional Neural Network (CNN)**

CNN are a powerful ANN technique. These networks can preserve the spatial structure of the problem by learning internal feature representations. CNN were invented for object recognition and researchers are achieving major results on computer vision using this type of NN.

There are three types of layers in a CNN. The first is Convolutional Layers (CL), which consists of filters and feature maps. The filters can be considered as the neurons of the layer. They have weights and outputs a value. The output of a filter being granted to the previous layer is called feature mapping. The distance that a filter moves across the input space in each activation is referred as the stride [31].

The second layer type is called pooling layers, which down samples the previous layers feature map. Pooling layers consists of one or more CLs and are intended to unite the learned features in the previous layers feature map. Pooling is a technique to derive features and to reduce the overfitting.

The third being a fully connected layer, which are the feedforward NN layer. These layers may have an activation function to derive the probability of a class predictions. These layers are used to construct the non-linear combination of features for the NN to make predictions.

For the current research implementation, a one-dimensional CNN will be constructed for both classification and regression tasks.

#### **3.6.6.4 Reduce Overfitting and Underfitting**

The purpose of predictive models is to achieve unbiased results on both the training and testing data. Which mean the model should be able to learn from the known sample and adapt to new data points. Less amount of training will result in an underfit model. While too much training will result in an overfitted model. In both of these cases, the model will not be able to generalize. A good fit of a model should be able to learns the training dataset and generalizes well in a newly presented dataset.

The scenario of underfitting of a deep learning model can be addressed by increasing the capacity of the model. If model shows high bias and low variance, it is likely to be underfit. Experiments can be done to the predictive model architecture to reduce the underfitting. If it is a deep learning model, then increasing the neurons of each layers and increasing the number of layers can be considered as changes to the architecture which in a increase of capacity of the model. Overfitting scenario can be identified by monitoring the performance of the model in training. A graph can be plotted against the loss of the model in training period for the training and validation data. The line which represents training will drop and may show little to no change while the line which represents the validation will drop initially but in a certain point of time it will rise again. These lines in the graph reveals the learning process of the model until begins overfitting [32].

The author will consider a weight regularization technique to reduce the overfitting of the data. Regularization will do small changes to the underlying learning algorithm, so it performs much better generalization.

Another technique that the author consider is to add Dropouts. The concept is to randomly drop the neurons in the NN during training. This process prevents the NN form adapting to the training data too much [33].

Another technique which is called “Early Stopping”, available in the TensorFlow library. Early stopping will make sure to stop the training process when the monitored metric has stopped improving [34].

### **3.6.7 Hyperparameter Optimization**

Grid search is an exhaustive search for selecting hyperparameters for predictive models. Once the parameter grid is set up each value combination of the grid will be used to train the model and evaluate using cross validation strategy. This approach is not that inefficient when having large number of records in the training dataset. The process will consume a lot of time and processing power [35].

The random search approach will select random hyperparameters from the grid to try out with the training model. Compared to the grid search all the parameters will not be tried out. This method is useful to figure out the parameter range for grid search [36].

## **3.7 Predictive Model Evaluation**

The focus of this sub section is to mention the different performance metrics available for the predictive models generated from the algorithms discussed in the previous section 3.5.1 learning algorithms. All the models generated will be evaluated based on these performance matrices and critically analysed in the next chapter 4 Evaluation.

### **3.7.1 Evaluation Metrics for Classification**

This section will discuss the performance matrices available for the classification task.

#### **3.7.1.1 Confusion Matrix**

The confusion matrix is one of the evaluation matrix available for a classification predictive model to measure its accuracy and correctness. This metric can be used for both binary and multi class classification. Figure 6 presents the confusion matrix that can be used for a binary classification problem.

		ND	D	
		True Negative (TN)	False Positive (FP)	
Actual Class	ND	True Negative (TN)	False Positive (FP)	D - Delayed ND - Not Delayed
	D	False Negative (FN)	True Positive (TP)	
		Predicted Class		

Figure 6 - Confusion Matrix

The current research being a binary classification problem, the flights which are not delayed will be labelled as 0 and the delayed flight will be labelled as 1. Therefore, figure 6 can be interpreted as below [37].

- True Positive (TP) - Which being the case that a delayed flight getting classified as a delayed flight.
- True Negative (TN) - This is the case that a non-delayed flight being classified as a non-delayed flight.
- False Positive (FP) - Which being the case that a non-delayed flight being classified as a delayed flight.
- False Negative (FN) - This is the case that a delayed flight getting classified as a non-delayed flight.

A perfect classification model should not have any FP or FN. But in reality, predictive models will not be 100% accurate. In the context of current research, it is evident that the author needs to keep his attention on FP values because classifying an on-time flight as delayed flight will cause much more issues to the passengers.

### 3.7.1.2 Accuracy

Accuracy in classification problems can be defined as the correct number of predictions over the total predictions [38]. Equation 3.1 represents the accuracy from a confusion matrix.

$$Accuracy = (TP + TN)/(TP + FP + FN + TN) \quad (3.1)$$

Measuring the metric accuracy has its most value when the datasets contains a balanced data sample.

### 3.7.1.3 Precision

True positive predictions over the total positive predictions can be taken as the precision of a predictive model [38]. 3.2 shows the equation related to precision.

$$Precision = TP / (TP + FP) \quad (3.2)$$

Precision measure as implies from the above equation is the proportion of predicted delayed flights which were actually delayed. It is clear that recall depicts the performance in regard to false positives.

### 3.7.1.4 Recall

Recall is the portion of positive predictions that were correctly classified, which can be represent by the below equation [38]. Equation 3.3 represent recall.

$$Recall = TP / (TP + FN) \quad (3.3)$$

Recall implies what proportion of flights that were actually delayed were classified as delayed flights. It is clear that recall depicts the performance in regard to false negatives.

### 3.7.1.5 F1 Score

The F1 score is used to measure the predictions accuracy, and it balances the use of precision and recall. The F1 score can provide a more realistic measure of a model's performance by using both precision and recall [38]. Below equation 3.4 represents this measure.

$$F1\ Score = (2 * Precision * Recall) / (Precision + Recall) \quad (3.4)$$



### 3.7.1.6 ROC Curve

The receiver operating characteristics (ROC) is a two-dimensional graph. The True Positive Rate (TPR) represents in the axis y while False Positive Rate (FPR) is the axis x. The below figure shows an example for a ROC Curve.

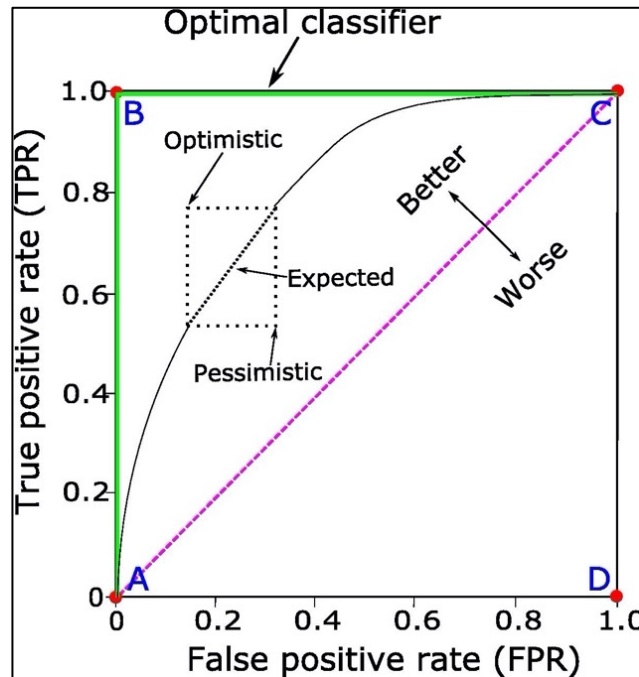


Figure 7 - ROC Curve Explained

As shown in the figure 7, there are four points marked in the ROC curve in name “A”, “B”, “C”, “D”. The point A represents a classifier when there is no positive and no negative classification which means all negatives are correctly classified. Therefore,  $TPR = 0$  and  $FPR = 0$  [37].

The point C represents a classifier where all positive samples are classified correctly, and the negative samples are misclassified. The point D represents a classifier where all positive and negative samples are misclassified. The point B represents a classifier where all positive and negative samples are correctly classified. The point B represents the perfect classification. The green curve shows the perfect classification performance. This particular curve reflects that the classifier perfectly ranked the positive samples relative to the negative samples [37].

### **3.7.1.7 Area Under the Curve (AUC)**

Evaluating classifiers is not exactly easy when they do not have any variable quantity representation of the performance. Therefore, AUC metric is calculated and it under the ROC curve. The AUC score should always be bounded between zero and one. If the score is below 0.5 then that particular classifier is worthless [37].

## **3.7.2 Evaluation Metrics for Regression**

This section will discuss the performance matrices available for the regression task.

### **3.7.2.1 Mean Squared Error (MSE)**

MSE is one of the most preferred metrics for regression problems. It represents the average of squared difference between the target and the predicted value by regression models. Smaller the MSE, the closer it is to finding a best fit [38].

### **3.7.2.2 Root Mean Squared Error (RMSE)**

RMSE measures the squared root of the average on the squared differences between the target and the predicted value by regression models. RMSE value will be larger than the MSE. Smaller the RMSE, the model performs better.

### **3.7.2.3 Mean Absolute Error**

MAE is the absolute difference between the target and the value predicted by a regression model.

### **3.7.2.4 R Squared**

R Squared represents how accurately the dependent variable can be estimated from the explanatory variables. If R Squared has a high value generally indicates a small prediction error [39].

## 3.8 Prediction

Irrespective of the predictive models, a separate function needs to be implemented to handle the prediction of a model. The particular function should take the necessary flight details of a passenger and produce a prediction as an output. The underlying weather data should be taken from a suitable weather API based on the data and time of the passenger's flight.

For the current research problem, command line functionality will be implemented so that the author can demonstrate the prediction of a respective predictive model. The weather data will be selected as a random record from the existing datasets to represent the API call. For the classification task, the output will either be 0 which means on time flight or 1 meaning a delayed flight. Regression prediction functionality will output a value in minutes as the delayed time.

## 3.9 Experiments

This subsection will explain the different experiments which have been planned to carry out through the research in a high-level manner. Each of these experiments are grouped into stages for better understanding.

### 3.9.1 Stage One Experiments

Year 2018 datasets will be used for both classification and regression tasks. Following three experiments will be carried out during this stage.

- Using only flight related features
- Using flight and weather features to check if weather features improve performance
- Standardizing and normalizing the numerical features of the dataset. Out of these two techniques, the technique which gives the highest performance will be used.
  - To reduce the overfitting of the neural network, weight regularization and dropout layer regularization will be used.
  - To get an optimum result, the training will be stopped when the loss of the network will not minimize after 10 epochs.

### **3.9.2 Stage Two Experiments**

From year 2014 to 2018, five years of data will be used for this experiment. The same type of experiment for both classification and regression will be used as 3.9.1 stage one. Depending on the stage one results, the author will decide to do continuous experiments on using flight features or both flight and weather features. This experiment will give an insight on if the performance will improve by adding more data to the predictive models.

### **3.9.3 Stage Three Experiments**

The sampling method which is mentioned in the 3.5.4 will be used to generate a balanced dataset created from the stage two experiment, which had equal portion of delayed and non-delayed flight record. Same experiment done in stage one and two will be executed in this stage as well. This experiment will indicate the performance impact by having a balanced data sample.

### **3.9.4 Stage Four Experiments**

In this stage, the author will choose the highest performing models and the dataset from stage one to three, for further fine tuning. Hyperparameter optimizations methods, which was mentioned in the 3.6.7 will be used for the tuning of selected models. This experiment will indicate the author on how much the current models can be improved using the optimization techniques.

### **3.9.5 Stage Five Experiments**

Convolutional Neural network will be introduced to the experiments. Depending on the time, which is consumed for the CNN models to train, the author might need to reduce the sample size of the dataset. The author plans to experiment with balance and unbalanced data samples to check how CNN performs in both cases.

### **3.9.6 Stage Six Experiments**

In the literature review it was found that if a flight gets delayed more than 15 minutes, that particular flight gets identified as a delayed flight. Therefore, all the experiment carried up to now have the fifteen minutes delay threshold. The author wants to see how the models will behave if the delay threshold is reduced. The author will reduce the delay threshold for 5 minutes to see the performance results. This experiment will only be carried out as a classification task.

### **3.10 Summary**

This chapter provide the overview of the methodology which will be carried out during the research. The methodology is broken down into different phases like datasets, pre-processing, learning algorithms, evaluation and experiments. Different types of datasets and pre-processing techniques have been noted down by the author. High level overview of the learning algorithms which was selected form the literature review was presented as well. Different evaluation technique which will be used to evaluate the predictive models also discussed. Finally, a plan for the experiments for the current research is presented by the author.

# Chapter 4: Evaluation

## 4.1 Overview

This chapter will discuss the predictive models presented in the Chapter 3: Methodology, section 3.6 for their performance. Each experiment that has been carried out to tackle both the classification and regression problem will be evaluated. The evaluation metrics presented in the methodology, section 3.7 will be used as the parameters for the evaluation. The experiments will be broken down to different stages and analysed for better understanding. Best performing models in each experiment are highlighted.

## 4.2 Flight Delay Classification

This sub section will be focused on the different experiments which have been carried out using the classification algorithms presented in the methodology, section 3.6. The first three stages of experiments were carried out using few statistical models and a feed forward neural network is named as “Sequential NN”. From forth stage above, the highest performing models from the first three stages will be taken into further tuning. A Convolutional Networks will be introduced for the experiments. The number of epochs for training the neural network was taken using early stopping technique motioned in the methodology section 3.6.6.4. The training will be stopped when the training loss stops decreasing for 10 epochs. Confusion matrix shows its result as a percentage by total testing sample.

### 4.2.1 Stage One: One Year Worth of Data Records

Year 2018 flight records were used for this stage of experiments. The datasets were split into training, testing and validation as mentioned in the 3.5.6 section. Table 8 shows the datasets distribution for this stage of experiments.

	Number of Records	Percentage
<b>Total Data Records</b>	325761	100%
<b>Training</b>	208486	64%
<b>Testing</b>	65153	20%
<b>Validation</b>	52122	25% from training records

Table 8 - Classification, dataset distribution for stage one experiments

#### 4.2.1.1 Experiment One – Using Only Flight Records

The first experiment for this stage was carried out using only the data features related to flights. Weather data was not in cooperated to the predictive models. Please refer to section 3.5.2 for the data attributes. Table 9 presents the different training and testing parameters used for the learning algorithms to generate the predictive models.

	<b>Training and testing parameter description</b>
<b>Logistic Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Gaussian NB</b>	Default parameters available in the library
<b>Sequential NN</b>	Input layer – 256 units First hidden layer – 128 units, activation = 'relu' Second hidden layer – 64 units, activation = 'relu' Output layer - 1 units, activation = sigmoid Number of epochs form early stopping = 45, batch size = 1000

Table 9 - Classification, training and testing parameters for stage one experiment one

Please refer to the appendix C.1 to find the learning curve plot for the created neural network.

	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Accuracy</b>
<b>Logistic Regression</b>	0.27	0.58	0.37	0.59
<b>Random Forest</b>	0.77	0.02	0.04	0.80
<b>Decision Trees</b>	0.58	0.07	0.13	0.80
<b>Gaussian NB</b>	0.24	0.45	0.31	0.59
<b>Sequential NN</b>	0.52	0.00	0.00	0.79

Table 10 - Classification, performance matrices for classification stage one experiment one

Table 10 shows the performance matrices for each predictive model that was used for this experiment. Based on accuracy Random Forest, Decision Tress, and sequential NN seems to be performing better while Gaussian NB performs the least. The unusual values for the recall and F1-Score can be explained from the below confusion matrix represented in Table 11. These results depict that the models tend to classify not delayed flights than the delayed flights. Since this is an imbalanced dataset this particular result can be accepted.

	Actual Class	Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Logistic Regression	ND	47.6	31.85
	D	8.72	11.84
Random Forest	ND	79.33	0.12
	D	20.17	0.38
Decision Trees	ND	78.34	1.10
	D	19.05	1.51
Gaussian NB	ND	49.92	29.52
	D	11.32	9.24
Sequential NN	ND	79.41	0.04
	D	20.51	0.04

Table 11 - Classification, confusion matrix for stage one experiment one

Figure 8 show the ROC curves for the stage one experiment one. Area under the curve (AUC) values can be seen in the figure as well.

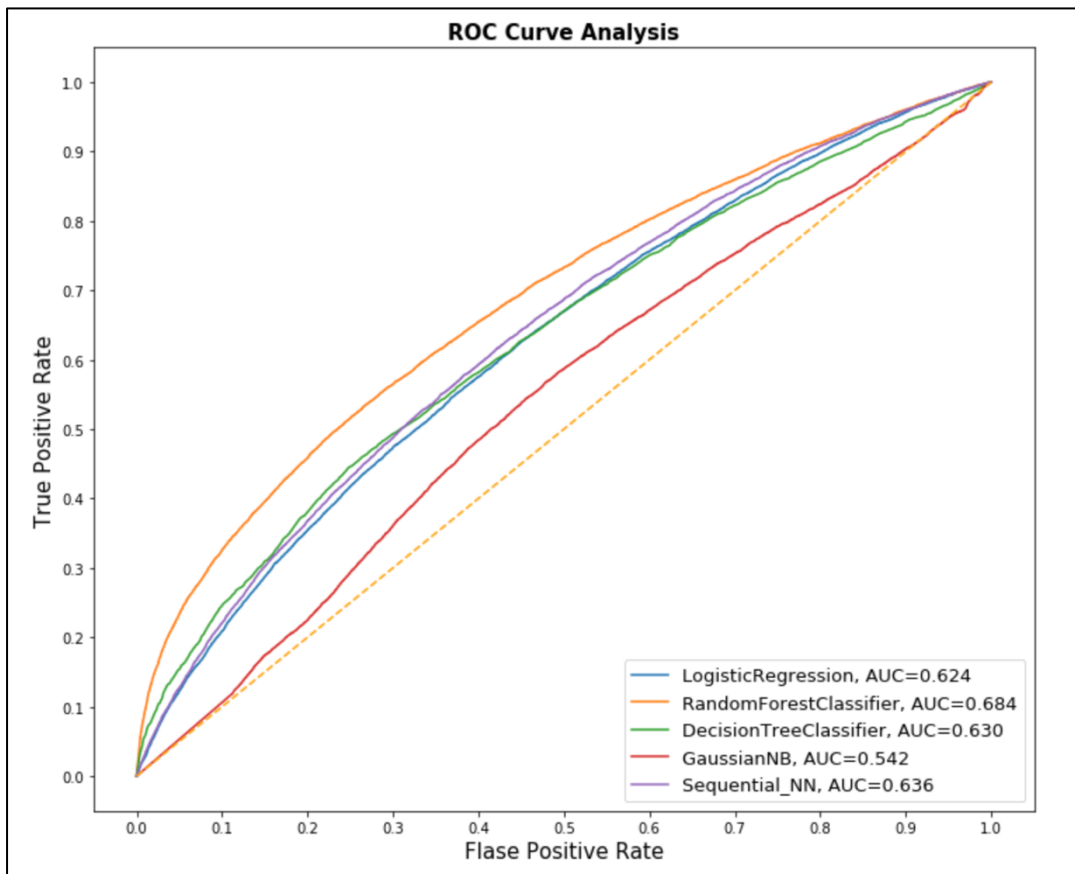


Figure 8 - Classification, ROC Curve for stage one experiment one

Based on the Figure 8 results, it seems Random Forest model performs much better than the Decision Tree and Sequential NN models for this experiment.



#### 4.2.1.2 Experiment Two – Using Flight and Weather Data Records

The second experiment for this stage is carried out using flight and weather data. As mentioned in the chapter 3 both datasets were merged based on the flight date and airport code. Table 12 shows the different training and testing parameters used for the learning algorithms.

	<b>Training and testing parameter description</b>
<b>Logistic Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Gaussian NB</b>	Default parameters available in the library
<b>Sequential NN</b>	Input layer – 256 units First hidden layer – 128 units, activation = 'relu' Second hidden layer – 64 units, activation = 'relu' Output layer - 1 units, activation = sigmoid Number of epochs form early stopping = 36, batch size = 1000

Table 12 - Classification, training and testing parameters for stage one experiment two

Please refer to the appendix C.2 to find the learning curve plot for the created neural network.

	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Accuracy</b>
<b>Logistic Regression</b>	0.30	0.61	0.40	0.62
<b>Random Forest</b>	0.75	0.09	0.16	0.80
<b>Decision Trees</b>	0.64	0.11	0.19	0.80
<b>Gaussian NB</b>	0.24	0.47	0.32	0.57
<b>Sequential NN</b>	0.65	0.13	0.21	0.80

Table 13 - Classification, performance metrics for stage one experiment two

Table 13 shows the performance matrices for each predictive model that was used for this experiment. Same as the Stage One Experiment One, Random Forest, Decision Trees and Sequential NN seems to be performing better on accuracy. But compared to Experiment One, overall performance results seem to be improved when in cooperating weather data. Even though the accuracy hasn't changed much, other metrics have a considerable improvement.

		Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Logistic Regression	ND	49.03	29.96
	D	8.27	12.74
Random Forest	ND	78.37	0.63
	D	19.09	1.92
Decision Trees	ND	77.7	1.29
	D	18.71	2.29
Gaussian NB	ND	47.17	31.83
	D	11.08	9.92
Sequential NN	ND	77.58	1.42
	D	18.36	2.65

Table 14 - Classification, confusion matrix for step one experiment two

Table 14 shows the confusion matrix for this experiment. Compared to the experiment one, classifying true positive and true negative values seems to be improved. Therefore, classifying delayed and non-delayed flights have been improved.

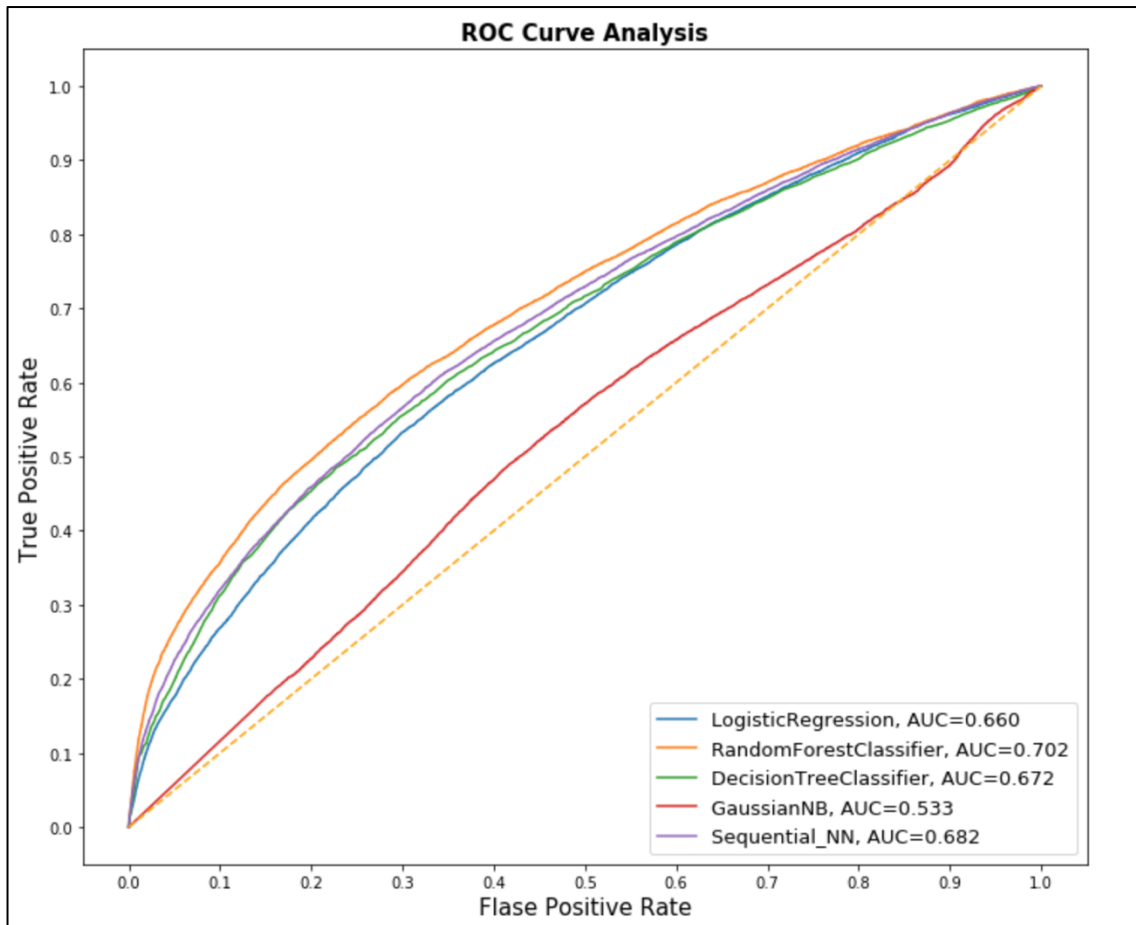


Figure 9 - Classification, ROC for step one experiment two

Figure 9 shows the ROC curve for this experiment. As experiment one, Random Forest predictive model seems to be outperforming other models. Based on the result of experiment one and two, the author decides to carry out all the other experiments using the flight and weather datasets.

#### 4.2.1.3 Experiment Three – Scaling Numerical Features

For this experiment, all the numerical features were standardized using standard scaler. For the NN, weight regularization added to the all the layers except for the output layer. Dropout layer regularization also added between each hidden layer. These regularizations were added to minimize the overfitting of the network. Table 15 shows the training and testing parameters.

	Training and testing parameter description
<b>Logistic Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Gaussian NB</b>	Default parameters available in the library
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units, activation = sigmoid Number of epochs for optimum result = 61, batch size = 1000

Table 15 - Classification, training and testing parameters for stage one experiment three

Please refer to appendix C.3.1 to find the learning curve plot for the neural network.

	Precision	Recall	F1-Score	Accuracy
<b>Logistic Regression</b>	0.29	0.57	0.39	0.63
<b>Random Forest</b>	0.71	0.08	0.15	0.80
<b>Decision Trees</b>	0.55	0.11	0.18	0.80
<b>Gaussian NB</b>	0.24	0.40	0.30	0.61
<b>Sequential NN</b>	0.66	0.11	0.19	0.80

Table 16 - Classification, performance metrics for stage one experiment three

Table 16 shows the performance matrices for each predictive model that was used for this experiment. Same as the Stage One Experiment two, Random Forest, Decision Trees and Sequential NN seems to be performing better on accuracy.

		Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Logistic Regression	ND	50.79	28.49
	D	8.93	11.79
Random Forest	ND	78.59	0.68
	D	19.02	1.71
Decision Trees	ND	77.44	1.83
	D	18.49	2.23
Gaussian NB	ND	52.36	26.92
	D	12.43	8.29
Sequential NN	ND	78.07	1.2
	D	18.41	2.31

Table 17 - Classification, confusion matrix for step one experiment three

Table 17 shows the confusion matrix for this experiment. Compared to the experiment two, results do not show much improvement.

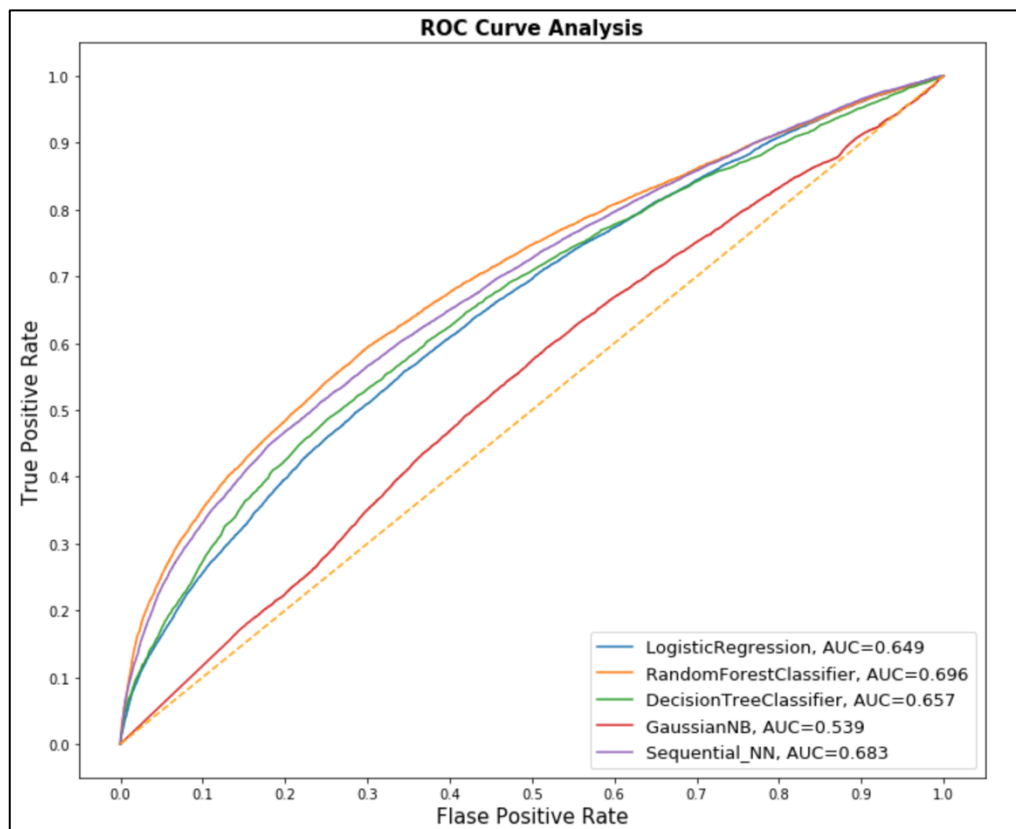


Figure 10 - ROC for step one experiment three

Figure 10 shows the ROC curve for this experiment. As the previous experiments Random Forest model performs better than the others. Sequential NN seems to be performing better than the Decision Tree model, than the experiment two.

#### 4.2.2 Stage Two: Five Years' worth of Data records

Year 2014 to year 2018 flight records were used for this stage of experiments. The datasets were split into training, testing and validation datasets as mentioned in the 3.5.6 section.

Table 18 shows the datasets distribution for this stage of experiments.

	Number of Records	Percentage
<b>Total Data Records</b>	1407144	100%
<b>Training</b>	900572	64%
<b>Testing</b>	281429	20%
<b>Validation</b>	225143	25% from training dataset

Table 18 - Classification, dataset distribution for classification stage two experiments

##### 4.2.2.1 Experiment One – Using Flight and Weather Data Records

Same as in Stage one experiment two, this experiment is carried out using flight and weather data. Table 19 shows the different training and testing parameters used for the learning algorithms.

	Training and testing parameter description
<b>Logistic Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Gaussian NB</b>	Default parameters available in the library
<b>Sequential NN</b>	Input layer – 256 units First hidden layer – 128 units, activation = 'relu' Second hidden layer – 64 units, activation = 'relu' Output layer - 1 units, activation = sigmoid Number of epochs form early stopping = 66, batch size = 1000

Table 19 - Classification, training and testing parameters for stage two experiment one

Please refer to the Appendix D.1 to find the learning curve plot for the created neural network.

	Precision	Recall	F1-Score	Accuracy
<b>Logistic Regression</b>	0.29	0.51	0.37	0.61
<b>Random Forest</b>	0.76	0.06	0.12	0.78
<b>Decision Trees</b>	0.55	0.12	0.19	0.78
<b>Gaussian NB</b>	0.27	0.42	0.33	0.61
<b>Sequential NN</b>	0.62	0.23	0.34	0.79

Table 20 - Classification, performance metrics for stage two experiment one

Table 20 shows the performance matrices for each predictive model that was used for this experiment. Same as the stage one experiments Random Forest, Decision Trees and Sequential NN seems to be performing better on accuracy. But accuracy and overall performance metrics of the models has reduced slightly compared to the stage one experiments. Sequential NN has a slight increase in the precision, recall and F1 scores.

		Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
<b>Logistic Regression</b>	<b>ND</b>	49.21	27.91
	<b>D</b>	11.32	11.56
<b>Random Forest</b>	<b>ND</b>	76.64	0.47
	<b>D</b>	21.43	1.46
<b>Decision Trees</b>	<b>ND</b>	74.93	2.19
	<b>D</b>	20.19	2.7
<b>Gaussian NB</b>	<b>ND</b>	51.71	25.4
	<b>D</b>	13.32	9.56
<b>Sequential NN</b>	<b>ND</b>	73.87	3.25
	<b>D</b>	17.52	5.36

Table 21 - Classification, confusion matrix for stage two experiment one

Table 21 shows the confusion matrix for this experiment. Compared to stage one experiment two, there are no major improvement even after adding another 4 years of data to the models. But classifying delayed flights have gained a slight increase on this experiment.

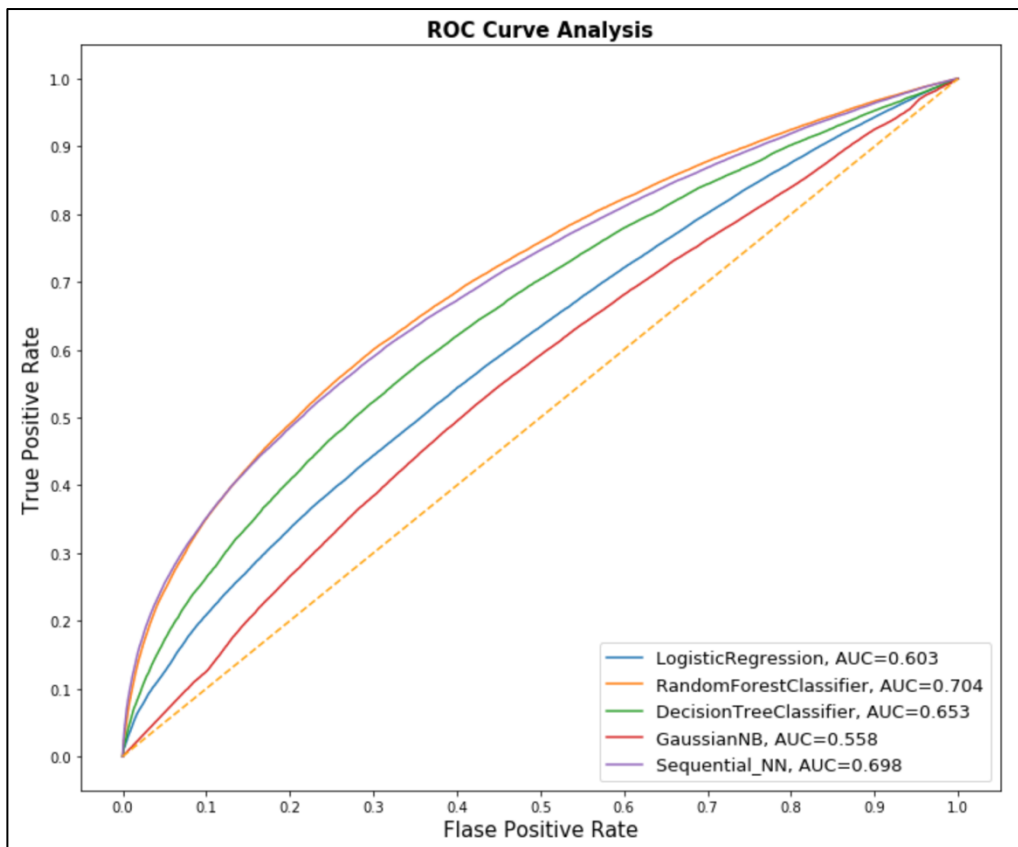


Figure 11 - Classification, ROC curve for stage two experiment two

Figure 11 shows the ROC curve for this experiment. Compared to the previous experiments Sequential NN seems to be catching up to the Random Forest model.

#### 4.2.2.2 Experiment Two – Scaling Numerical Features

	Training and testing parameter description
<b>Logistic Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Gaussian NB</b>	Default parameters available in the library
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units, activation = sigmoid Number of epochs form early stopping = 31, batch size = 1000

Table 22 - Classification, training and testing parameters for stage two experiment two

As stage one experiment three, all the numerical features have been scaled using standard scaler. Table 22 shows the training and testing parameters for the current experiment.

Please refer to the Appendix D.2 to find the learning curve plot for the created neural network. Refer Appendix D.3 for the same experiment results using Min-Max normalization. Standard Scaler and Min-Max results do not show much of difference, but Standard Scaler performs slightly better.

	Precision	Recall	F1-Score	Accuracy
<b>Logistic Regression</b>	0.33	0.56	0.41	0.64
<b>Random Forest</b>	0.76	0.06	0.12	0.78
<b>Decision Trees</b>	0.55	0.12	0.19	0.78
<b>Gaussian NB</b>	0.26	0.61	0.36	0.51
<b>Sequential NN</b>	0.61	0.10	0.17	0.78

Table 23 - Classification, performance metrics for stage two experiment two

Table 23 shows the performance matrices for each predictive model that was used for this experiment. Compared to the stage two experiment one accuracy haven't changed much. Overall other performance matrices haven't improved either.

		Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
<b>Logistic Regression</b>	<b>ND</b>	50.87	26.25
	<b>D</b>	10.16	12.72
<b>Random Forest</b>	<b>ND</b>	76.64	0.47
	<b>D</b>	21.43	1.46
<b>Decision Trees</b>	<b>ND</b>	74.93	2.19
	<b>D</b>	20.19	2.7
<b>Gaussian NB</b>	<b>ND</b>	36.69	40.42
	<b>D</b>	8.94	13.95
<b>Sequential NN</b>	<b>ND</b>	75.72	1.39
	<b>D</b>	20.68	2.2

Table 24 - Classification, confusion matrix for stage two experiment two

Table 24 shows the confusion matrix for this experiment. Compared to the previous experiment Random forest algorithm seems to be performing exactly the same while Sequential NN classification delayed flight got a negative impact. The TP rate have been reduced slightly.



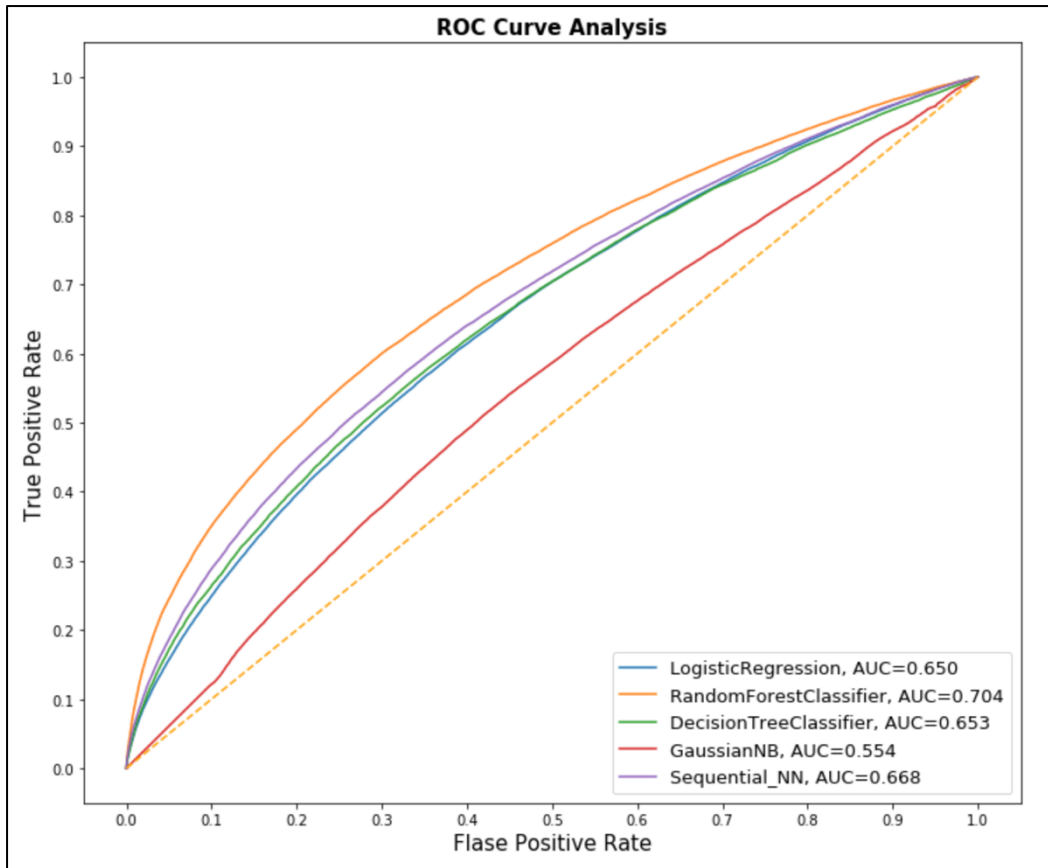


Figure 12 - Classification, ROC Curve for stage two experiment two

Figure 12 shows the ROC curve for this experiment. Compared to the previous experiment Sequential model performance have been suffered by a large margin. But Overall Random forest and Sequential NN models have been performing better than the other models though out the current experiments.

### 4.2.3 Stage Three: Balanced Data Sample

As motioned in the 3.5.4 Sampling Dataset section, a data sample form the year 2014 to 2018 dataset used in this stage three experiments. Sample was constructed to have equal portions of delayed and non-delayed records. Table 25 show the datasets distribution.

	Number of Records	Percentage
<b>Total Data Records</b>	644116	100%
<b>Training</b>	412233	64%
<b>Testing</b>	128824	20%
<b>Validation</b>	103059	25% from training dataset

Table 25 - Classification, dataset distribution for classification stage three experiments

### 4.2.3.1 Experiment One – Using Flight and Weather Data Records

Same as in stage two experiment two, this experiment is carried out using flight and weather data. Table 26 shows the different training and testing parameters used for the learning algorithms.

	Training and testing parameter description
<b>Logistic Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Gaussian NB</b>	Default parameters available in the library
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units, activation = sigmoid Number of epochs form early stopping = 50, batch size = 1000

Table 26 - Classification, training and testing parameters for stage three experiment one

Please refer to the Appendix E.1 to find the learning curve plot for the created neural network.

	Precision	Recall	F1-Score	Accuracy
<b>Logistic Regression</b>	0.58	0.49	0.53	0.57
<b>Random Forest</b>	0.65	0.58	0.61	0.63
<b>Decision Trees</b>	0.59	0.55	0.57	0.58
<b>Gaussian NB</b>	0.54	0.56	0.55	0.54
<b>Sequential NN</b>	0.61	0.59	0.61	0.61

Table 27 - Classification, performance metrics for stage three experiment one

Table 27 shows the performance matrices for each predictive model that was used for this experiment. Compared to the stage one and stage two experiments, the accuracy of the models has reduced drastically. But as usual Random Forest and Sequential NN are performing better than the other models. Decision Tree model's accuracy has reduced compared to these two models

	Actual Class	Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Logistic Regression	ND	32.63	17.48
	D	25.32	24.58
Random Forest	ND	34.58	15.53
	D	20.98	28.92
Decision Trees	ND	30.58	19.52
	D	22.29	27.61
Gaussian NB	ND	26.38	23.72
	D	22.12	27.78
Sequential NN	ND	32.33	17.77
	D	20.27	29.63

Table 28 - Classification, confusion matrix for stage three experiment one

Table 28 shows the confusion matrix for this experiment. Compared to the previous experiments, TP percentage happened to be increased. Its kind a like balance out with the TN percentage. But the results are still biased against delayed flight by a slight margin.

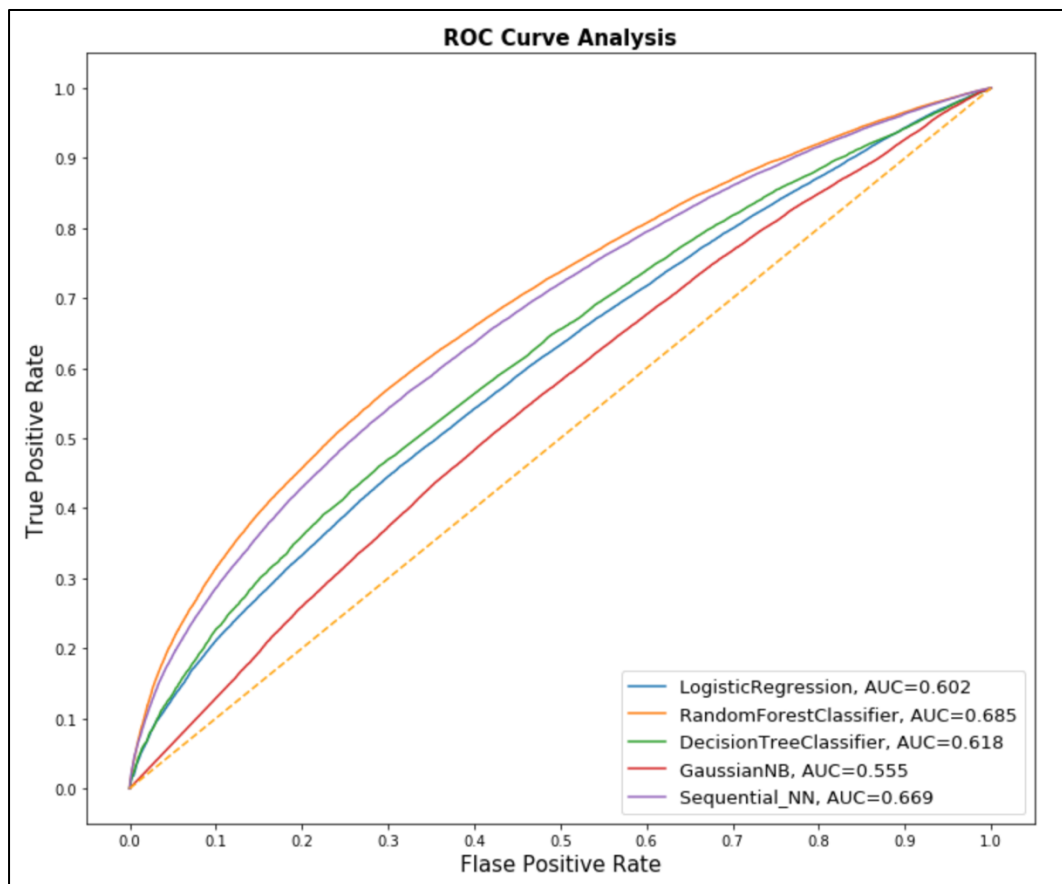


Figure 13 - Classification, ROC Curve for stage three experiment one

Figure 13 shows the ROC curve for this experiment. As previous experiments, even using the balanced dataset, the Random Forest and Sequential NN models outperform the other models. Decision Tree model's performance happens to be reduced.

#### 4.2.3.2 Experiment Two – Scaling Numerical Features

As Stage One Experiment three, all the numerical features have been scaled using standard scaler. Below table 29 shows the training and testing parameters for the current experiment.

	Training and testing parameter description
<b>Logistic Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Gaussian NB</b>	Default parameters available in the library
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units, activation = sigmoid Number of epochs for optimum result = 55, batch size = 1000

Table 29 - Classification, training and testing parameters for stage three experiment two

Please refer to the Appendix E.2 to find the learning curve plot for the created neural network. Appendix E.3 shows the results for the same experiment using Min-Max normalization.

	Precision	Recall	F1-Score	Accuracy
<b>Logistic Regression</b>	0.61	0.61	0.61	0.61
<b>Random Forest</b>	0.65	0.58	0.61	0.63
<b>Decision Trees</b>	0.59	0.55	0.57	0.58
<b>Gaussian NB</b>	0.56	0.29	0.38	0.53
<b>Sequential NN</b>	0.61	0.59	0.61	0.61

Table 30 - Classification, performance metrics for stage three experiment two

Table 30 shows the performance matrices for each predictive model that was used for this experiment. Compared to the previous experiment scaling didn't give much of an improvement.

	Actual Class	Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Logistic Regression	ND	30.41	19.69
	D	19.68	30.22
Random Forest	ND	34.58	15.53
	D	20.98	28.92
Decision Trees	ND	30.58	19.52
	D	22.29	27.61
Gaussian NB	ND	38.7	11.41
	D	35.53	14.36
Sequential NN	ND	32.48	17.62
	D	20.56	29.34

Table 31 - Classification, confusion matrix for stage three experiment two

Table 31 shows the confusion matrix for this experiment. Compared to the previous experiments, the results do not show any notable improvement. Figure 14 shows the ROC curve for this experiment.

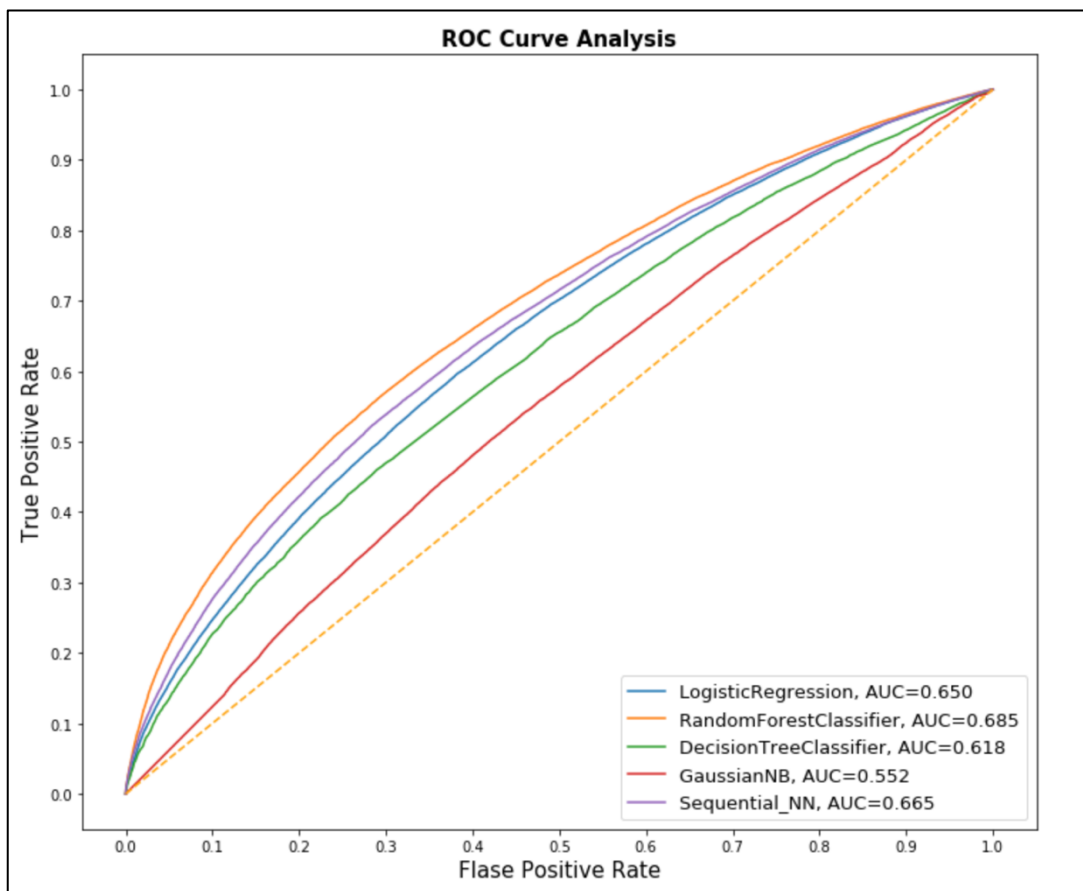


Figure 14 - Classification, ROC curve for stage three experiment two

As usual, Random Forest and Sequential NN seems to be performing better. Logistic Regression has an improvement over Decision Tree model.

## 4.2.4 Stage Four: Hyperparameter Tuning

Out of all the experiments carried out during the stage one, two and three, Random Forest and Sequential NN models outperformed the other models. Even though the stage three dataset provides the lower accuracy, considering the overall performance the author decides to use this particular dataset for the experiments which will conduct this point onwards.

Random and grid search techniques were used to narrow down the hyper parameters. Since training of models for each of these parameters consumed a large amount of time, the author had to reduce the training, testing and validation sample size for the hyperparameter optimization. The reduce sample contained equal portion of delayed and non-delayed flight records totalling hundred thousand records.

	Training and testing parameter description
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units, activation = sigmoid Number of epochs for early stopping = 55, batch size = 1000

Table 32 - Classification, training and testing parameters before tuning

Table 32 presents the training and testing parameter for both models before the hyperparameter tuning. Table 33 shows the performance result before the tuning as well. The Random forest models has performed the highest in the stage three experiment. Sequential NN slightly under performed.

	Precision	Recall	F1-Score	Accuracy
<b>Random Forest</b>	0.65	0.58	0.61	0.63
<b>Sequential NN</b>	0.61	0.59	0.61	0.61

Table 33 - Classification, performance before hyperparameter tuning

	Actual Class	Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Random Forest	ND	34.58	15.53
	D	20.98	28.92
Sequential NN	ND	32.48	17.62
	D	20.56	29.34

Table 34 - Classification, confusion matrix before hyperparameter tuning

Table 34 shows the confusion matrix used for the modes in stage three experiments. Below Table 35 shows the parameters generated from random search and grid search.

	Training and testing parameter description
Random Forest	N_Estimator = 100, minimum sample split = 15, min sample leaf=5, max features = 'sqrt', max depth = 75, bootstrap = false
Sequential NN	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units, activation = sigmoid Number of epochs for early stopping = 55, batch size = 500

Table 35 - Classification, hyperparameters for tuning

Table 36 presents the performance results after the tuning. Please refer to Appendix F for Learning Curves and ROC Curve.

	Precision	Recall	F1-Score	Accuracy
Random Forest	0.67	0.58	0.62	0.65
Sequential NN	0.63	0.61	0.61	0.62

Table 36 - Classification, performance after hyperparameter tuning

	Actual Class	Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Random Forest	ND	36.25	13.85
	D	21.19	28.71
Sequential NN	ND	32.57	17.53
	D	20.66	29.24

Table 37 - Classification, confusion matrix after hyperparameter tuning

Table 37 shows the confusion matrix after the tuning hyperparameters. Random forest accuracy was increased by 3% while Sequential NN increased only by 1%.

## 4.2.5 Stage Five: Introducing Convolutional Neural Network (CNN)

For this experiment, the author had to reduce the sample size of the dataset because the CNN took around seven hours to train in the stage three dataset. The author reduces the sample size to 80,000 records for equal portions of delayed and non-delayed flights records for a balanced dataset. An unbalanced dataset also contractures using around 80000 records. Below table show the distribution for both datasets.

	Number of Records	Percentage
<b>Total Data Records</b>	80000	100%
<b>Training</b>	51200	64%
<b>Testing</b>	16000	20%
<b>Validation</b>	12800	25% from training dataset

Table 38 - Classification, dataset distribution for stage five experiment

### 4.2.5.1 Experiment One - Using an Unbalanced Dataset

Below table shows the training and testing parameters used to the current experiment. Please refer to the Appendix G to find the learning curve plot for the created neural network. Table 39 presents the training and testing parameters used for the current experiment. Unbalance dataset container 80% on time records and 20% delayed records.

	Training and testing parameter description
<b>Random Forest</b>	N_Estimator = 100, minimum sample split = 15, min sample leaf = 5, max features = 'sqrt', max depth = 75, bootstrap = false
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units, activation = sigmoid Number of epochs form early stopping = 25, batch size = 500
<b>Convolutional NN</b>	Input layer – CONV1D (Filters = 32, kernel size = 7, activation = 'relu') First Hidden layer = CONV1D (Filters = 32, kernel size = 7, activation = 'relu') Dropout Layer = 0.5 MaxPooling1D, Pool size = 2 Second hidden layer – 50 units, activation = 'relu' Output layer - 1 units, activation = sigmoid Number of epochs form early stopping = 20, batch size = 500

Table 39 - Classification, Training and testing parameters for Stage five experiment one



	Precision	Recall	F1-Score	Accuracy
<b>Random Forest</b>	0.69	0.10	0.17	0.78
<b>Sequential NN</b>	0.60	0.06	0.11	0.78
<b>Convolutional NN</b>	0.65	0.08	0.15	0.78

Table 40 - Classification, performance metrics for stage five experiment one

Table 40 shows the performance results for the current experiment and Random Forest model performs slightly better than the other models.

	Actual Class	Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
<b>Random Forest</b>	<b>ND</b>	76.15	1.03
	<b>D</b>	20.54	2.28
<b>Sequential NN</b>	<b>ND</b>	76.24	0.94
	<b>D</b>	21.4	1.42
<b>Convolutional NN</b>	<b>ND</b>	76.18	1.00
	<b>D</b>	20.95	1.87

Table 41 - Classification, confusion matrix for stage five experiment one

Table 41 shows the confusion matrix for the experiment. Since this is an unbalanced dataset, it is evident from the results it is biased against the delayed flights. CNN seems to be providing similar results to the other models.

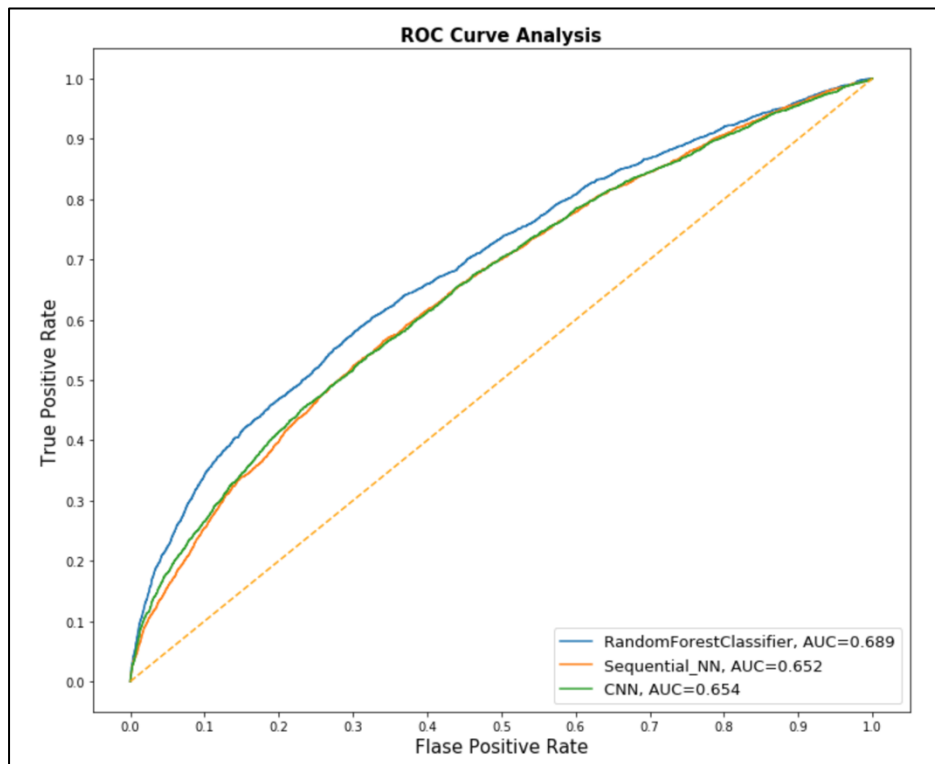


Figure 15 - Classification, ROC curve for the stage five experiment one

Figure 15 shows the ROC curve for the current experiment. CNN model seems to be performing slightly better than the Sequential NN model.

#### 4.2.5.2 Experiment Two - Using a Balanced Dataset

	Training and testing parameter description
<b>Random Forest</b>	N_Estimator = 100, minimum sample split = 15, min sample leaf=5, max features = 'sqrt', max depth = 75, bootstrap = false
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units, activation = sigmoid Number of epochs for optimum result = 23, batch size = 500
<b>Convolutional NN</b>	Input layer – CONV1D (Filters = 32, kernel size =7, activation = 'relu') First Hidden layer = CONV1D (Filters = 32, kernel size =7, activation = 'relu') Dropout Layer = 0.5 MaxPooling1D, Pool size = 2 Second hidden layer – 50 units, activation = 'relu' Output layer - 1 units, activation = sigmoid Number of epochs for optimum result = 19, batch size = 500

Table 42 - Classification, training and testing parameters for stage five experiment two

Above table 42 shows the training and testing parameters used to the current experiment. Please refer to the Appendix G to find the learning curve plot for the created neural network.

	Precision	Recall	F1-Score	Accuracy
<b>Random Forest</b>	0.67	0.56	0.61	0.64
<b>Sequential NN</b>	0.64	0.53	0.58	0.61
<b>Convolutional NN</b>	0.64	0.56	0.59	0.62

Table 43 - Classification, performance metrics for stage five experiment two

Table 43 shows the performance result for each model. Random Forest models seems to perform better than the neural networks. Table 44 shows the confusion matrix for the current experiment. The results do not have any notable difference.

	Actual Class	Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Random Forest	ND	35.55	14.00
	D	22.04	28.41
Sequential NN	ND	34.68	14.87
	D	23.66	26.79
Convolutional NN	ND	33.71	15.84
	D	22.44	28.01

Table 44 - Classification, confusion matrix for stage five experiment two

Figure 16 show the ROC curve for the current experiment.

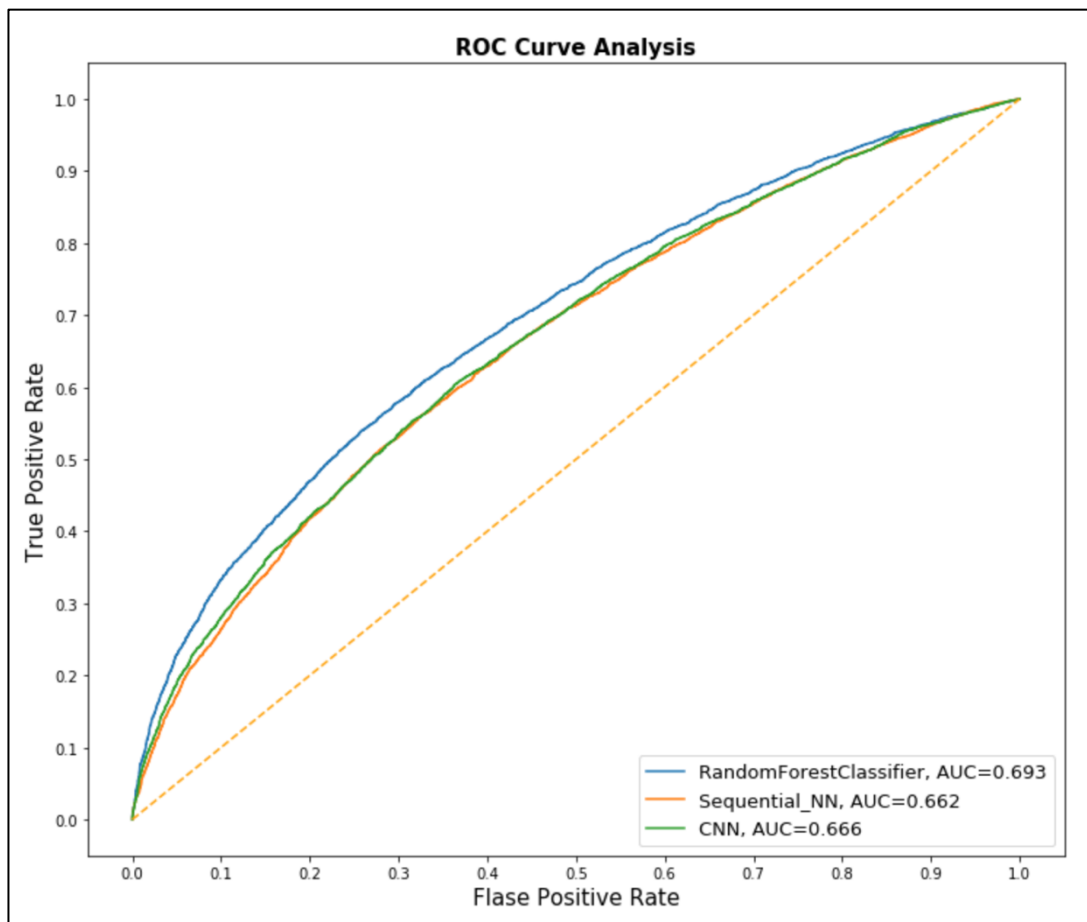


Figure 16 - Classification, ROC curve for stage five experiment

According to Figure 16, Random Forest models seems to perform better than the neural network models.

## 4.2.6 Stage Five: Changing Delay Threshold

For this experiment the ideal dataset would be the stage two, year 2014 to 2018 dataset. This particular dataset contains considerable number of records and the data are biased against the delayed flights. Same number of training, testing and validation records as the stage two experiment will be used here as well. Due to the massive number of records, CNN experiments will not be carried out in this dataset. Same number of training and testing parameter used in the stage four will be considered for the experiment. Delay threshold will be set to 5 minutes.

Total	Delayed	Non-delayed
1407144	460051	947093
100%	32.69	67.31

Table 45 - Classification, delayed and non-delayed flights after delay threshold changed

Table 45 shows the delay and on time flight record distribution for the dataset.

	Precision	Recall	F1-Score	Accuracy
Random Forest	0.71	0.29	0.41	0.73
Sequential NN	0.59	0.21	0.31	0.70

Table 46 - Classification, performance metrics for stage six experiments

Table 46 presents the performance result for the current experiment. Accuracy of the models are bit high compared to the stage two experiment. Random Forest algorithm seems to be performing better than the Sequential NN

	Actual Class	Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Random Forest	ND	63.43	3.96
	D	23.14	9.48
Sequential NN	ND	62.6	4.79
	D	25.64	6.98

Table 47 - Classification, confusion matrix for stage six experiments

Table 47 shows the confusion matrix for the current experiment. As expected, results are biased against delayed flights.

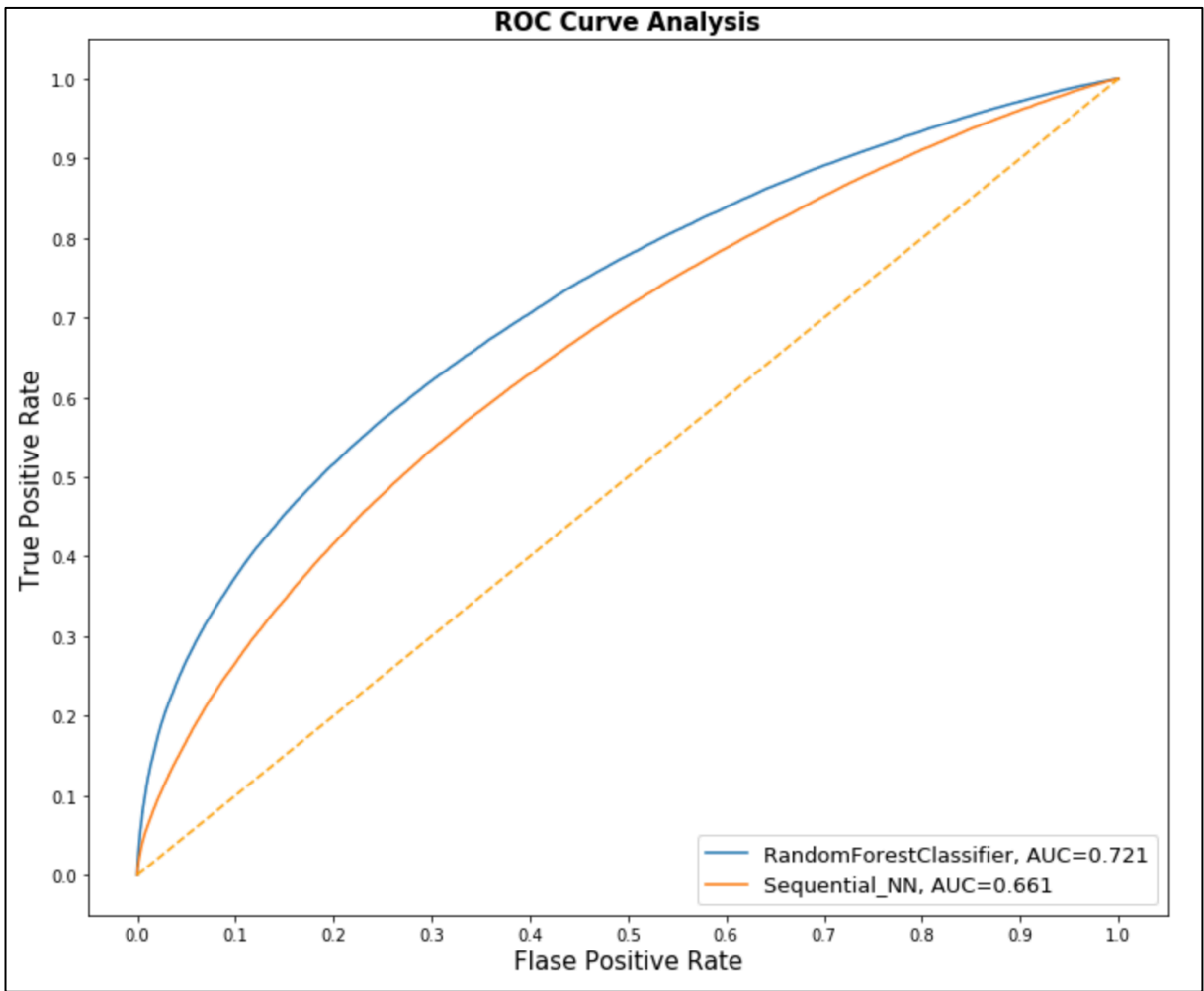


Figure 17 - Classification, ROC curve for the stage six experiments

Figure 17 shows the ROC curve for the current experiment. Random Forest algorithm seems to be performing much greater than the Sequential NN model.

## 4.3 Predicting Flight Delayed Time

This section will be focused on the different experiment which have been carried out using the regression algorithms presented in the chapter three methodology. Same as the classification experiments, the first three stages of experiments were carried out using statistical models and a feed forward neural network. From forth stage above, the highest performing models from the first three stages will be taken into further tuning. Convolutional Networks will be introduced for the experiments. The number of epochs in neural network were taken using the early stopping technique mentioned in the section 3.6.6.4. The training will be stopped when the training loss stops decreasing for 10 epochs. For the first three stages the same number of training, testing and validation parameters were used. To get an idea please refer to the experiment of 4.2 section.

### 4.3.1 Stage One: One Year worth of Data Records

Year 2018 flight records were used for this stage of experiments.

#### 4.3.1.1 Experiment One – Using only flight records

The first experiment for this stage was carried out using only the data features related to flights. Weather data was not in cooperated to the predictive models. Table 48 shows the different training and testing parameters used for the learning algorithms to generate the predictive models.

	<b>Training and testing parameter description</b>
<b>Linear Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Sequential NN</b>	Input layer – 256 units First hidden layer – 128 units, activation = 'relu' Second hidden layer – 64 units, activation = 'relu' Output layer - 1 unit, Optimizer = RMSprop = 0.001 Number of epochs form early stopping = 65

Table 48 - Regression, training and Testing parameters for stage one experiment one

	MAE	MSE	RMSE	R Squared
<b>Linear Regression</b>	20.56 Minutes	2000.28	44.72	0.02
<b>Random Forest</b>	20.27 Minutes	1913.91	43.75	0.06
<b>Decision Trees</b>	20.63 Minutes	1963.89	44.32	0.04
<b>Sequential NN</b>	19.56 Minutes	1896.76	43.33	0.08

Table 49 - Regression, performance metrics for stage one experiment one

Above table 49 show the results for the current experiment. Sequential NN seems to outperform all the predictive models. Please refer to the Appendix I.1 to find the learning curve plot for the created neural network.

#### 4.3.1.2 Experiment Two – Using Flight and Weather Data Records

The second experiment for this stage is carried out using flight and weather data. As mentioned in the chapter 3 both datasets were merged based on the flight date and airport code. Table 50 shows the different training and testing parameters used for the learning algorithms.

	Training and testing parameter description
<b>Linear Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Sequential NN</b>	Input layer – 256 units First hidden layer – 128 units, activation = 'relu' Second hidden layer – 64 units, activation = 'relu' Output layer - 1 unit, Optimizer = RMSprop = 0.001 Number of epochs form early stopping = 25

Table 50 - Regression, Training and Testing parameters for stage one experiment two

Please refer to the Appendix I.2 to find the learning curve plot for the created neural network.

	MAE	MSE	RMSE	R Squared
<b>Linear Regression</b>	21.25	1952.09	44.18	0.05
<b>Random Forest</b>	20.05	1835.23	42.84	0.10
<b>Decision Trees</b>	20.11	1905.27	43.65	0.07
<b>Sequential NN</b>	20.57	1848.34	42.99	0.10

Table 51 - Regression, performance metrics for stage one experiment two

Table 51 show the results for the current experiment. Compared to the previous experiment Sequential NN's performance have been reduced. Random Forest model have catch up to the NN model.

#### 4.3.1.3 Experiment Three – Scaling Numerical Features

For this experiment, all the numerical features were standardized using standard scaler. For the NN, weight regularization added to the all the layers except for the output layer. Dropout layer regularization also added between each hidden layer as well. These regularizations were added to minimize the overfitting of the network. Table 52 show the different training and testing parameters used for the learning algorithms.

	Training and testing parameter description
<b>Linear Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units Optimizer = RMSprop = 0.001 Number of epochs form early stopping = 30

Table 52 - Regression Training and Testing parameters for stage one experiment three

Please refer to the Appendix I.3 to find the learning curve plot for the created neural network.

	MAE	MSE	RMSE	R Squared
<b>Linear Regression</b>	22.46 Minutes	1880.48	43.36	0.05
<b>Random Forest</b>	20.71 Minutes	1779.13	42.18	0.10
<b>Decision Trees</b>	20.49 Minutes	1822.52	42.69	0.07
<b>Sequential NN</b>	19.54 Minutes	1885.30	43.42	0.11

Table 53 - Regression performance metrics for stage one experiment three

Table 53 show the results for the current experiment. Compared to the previous experiment, Sequential NN has gained performance after scaling the numerical features.



### 4.3.2 Stage Two: Five Years' worth of Data records

Year 2014 to year 2018 flight records were used for this stage of experiments.

#### 4.3.2.1 Experiment One – Using Flight and Weather Data Records

Same as in Stage one experiment two, this experiment is carried out using flight and weather data. Table 54 shows the different training and testing parameters used for the learning algorithms.

	Training and testing parameter description
<b>Linear Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Sequential NN</b>	Input layer – 256 units First hidden layer – 128 units, activation = 'relu' Second hidden layer – 64 units, activation = 'relu' Output layer - 1 unit, Optimizer = RMSprop = 0.001 Number of epochs form early stopping = 50

Table 54 - Regression, training and Testing parameters for stage two experiment three

Please refer to the Appendix J.1 to find the learning curve plot for the created neural network.

	MAE	MSE	RMSE	R Squared
<b>Linear Regression</b>	20.97 Minutes	1706.59	41.31	0.05
<b>Random Forest</b>	20.35 Minutes	1628.96	40.36	0.09
<b>Decision Trees</b>	20.71 Minutes	1702.15	41.26	0.05
<b>Sequential NN</b>	20.74 Minutes	1611.76	40.15	0.10

Table 55 - Regression, performance metrics for stage two experiment one

Table 55 presents the results for the current experiment. Same as the stage one experiment two, sequential NN performance has reduced.

### 4.3.2.2 Experiment Two – Scaling Numerical Features

Same as the stage one experiment three, all the numerical features were standardized using standard scaler. For the NN, weight regularization added to the all the layers except for the output layer. Dropout layer regularization also added between each hidden layer as well. These regularizations were added to minimize the overfitting of the network. Table 65 show the different training and testing parameters used for the learning algorithms.

	Training and testing parameter description
<b>Linear Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 unit Optimizer = RMSprop = 0.001 Number of epochs form early stopping = 26

Table 56 - Regression, training and testing parameters for stage two experiment two

Please refer to the Appendix J.2 to find the learning curve plot for the created neural network.

	MAE	MSE	RMSE	R Squared
<b>Linear Regression</b>	20.97 Minutes	1706.68	41.31	0.05
<b>Random Forest</b>	20.35 Minutes	1628.96	40.36	0.09
<b>Decision Trees</b>	20.71 Minutes	1702.15	41.26	0.05
<b>Sequential NN</b>	19.79 Minutes	1608.65	40.11	0.11

Table 57 - Regression, performance metrics for stage two experiment two

Above table 57 presents the results for the current experiment. Compared to the previous experiment, Sequential NN has gained performance after scaling the numerical features. Please refer Appendix J.3 for the same experiment result done using Min-Max normalization.

### 4.3.3 Stage Three: Balanced Data Sample

As motioned in the 3.5.4 Sampling Dataset section, a data sample form the year 2014 to 2018 dataset used in this stage three experiments. This dataset was used for the classification experiments as well. Having a balanced set of larger delays and small delays hopefully will give an advantage for regression algorithms.

#### 4.3.2.1 Experiment One – Using Flight and Weather Data Records

Same as in Stage one experiment two, this experiment is carried out using flight and weather data. Table 58 shows the different training and testing parameters used for the learning algorithms.

	<b>Training and testing parameter description</b>
<b>Linear Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf =10, max features = 'log2'
<b>Sequential NN</b>	Input layer – 256 units First hidden layer – 128 units, activation = 'relu' Second hidden layer – 64 units, activation = 'relu' Output layer - 1 unit Number of epochs form early stopping = 18

Table 58 - Regression, training and testing parameters for stage three experiment one

Please refer to the Appendix K.1 to find the learning curve plot for the created neural network.

	<b>MAE</b>	<b>MSE</b>	<b>RMSE</b>	<b>R Squared</b>
<b>Linear Regression</b>	35.18 Minutes	3242.36	56.94	0.03
<b>Random Forest</b>	33.18 Minutes	3085.24	55.54	0.08
<b>Decision Trees</b>	34.56 Minutes	3316.06	57.59	0.01
<b>Sequential NN</b>	33.51 Minutes	3137.58	56.01	0.06

Table 59 - Regression, performance metrics for stage three experiment one

Table 59 show the results for the current experiment. Compared to the previous experiments, the all the error rate happens to be increased by large margin. Still. The Random Forest and Sequential NN seems to be performing better than the other two models.

### 4.3.2.2 Experiment Two – Scaling Numerical Features

Same as the stage two experiment two, all the numerical features were standardized using standard scaler. For the NN, weight regularization added to the all the layers except for the output layer. Dropout layer regularization also added between each hidden layer as well. These regularizations were added to minimize the overfitting of the network. Below Table show the different training and testing parameters used for the learning algorithms.

	Training and testing parameter description
<b>Linear Regression</b>	Default parameters available in the library
<b>Random Forest</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Decision Trees</b>	Estimator = 10, minimum sample split = 20, min sample leaf = 10, max features = 'log2'
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units Optimizer = RMSprop = 0.001 Number of epochs form early stopping = 34

Table 60 - Regression Training and Testing parameters for stage three experiment two

Please refer to the Appendix K.2 to find the learning curve plot for the created neural network.

	MAE	MSE	RMSE	R Squared
<b>Linear Regression</b>	35.18 Minutes	3242.18	56.94	0.03
<b>Random Forest</b>	33.18 Minutes	3085.24	55.54	0.08
<b>Decision Trees</b>	34.56 Minutes	3316.01	57.58	0.01
<b>Sequential NN</b>	33.35 Minutes	3089.73	55.59	0.08

Table 61- Regression performance metrics for stage three experiment two

Table 61 presents the performance results for the current experiments. Scaling give a slight advantage for the Sequential NN, but overall, the error rates have been increased for this stage of experiments.

### 4.3.4 Stage Four: Hyperparameter Tuning

Out of all the experiments carried out during the stage one, two and three, Random Forest and Sequential NN models outperformed the other models. Stage one experiments provides the highest performance result. Therefore, author decides to use this particular dataset for the experiments which will conduct this point onwards.

Random and grid search techniques were used to narrow down the hyper parameters. Since training of models for each of these parameters consumed a large amount of time, the author had to reduce the training, testing and validation sample size for the hyperparameter optimization.

Table 62 shows the training and testing parameters used before the parameter optimization.

	<b>Training and testing parameter description</b>
<b>Random Forest</b>	N_Estimator = 100, minimum sample split = 15, min sample leaf=5, max features = 'sqrt', max depth = 75, bootstrap = false
<b>Sequential NN</b>	Input layer – 256 units, regularizer = L2=0.001 First hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units Optimizer = RMSprop = 0.001 Number of epochs form early stopping = 30

Table 62 - Regression, training and testing parameters before tuning

Table 63 show the performance results for the models before the tuning. Sequential NN has provided the lowest error rates.

	<b>MAE</b>	<b>MSE</b>	<b>RMSE</b>	<b>R Squared</b>
<b>Random Forest</b>	20.71 Minutes	1779.13	42.18	0.10
<b>Sequential NN</b>	19.54 Minutes	1885.30	43.42	0.11

Table 63 - Regression, performance result before tuning

Below table 64 show the training and testing parameters used for the hyperparameter optimization using random search and grid search.

	<b>Training and testing parameter description</b>
<b>Random Forest</b>	N_Estimator = 100, minimum sample split = 15, min sample leaf=5, max features = 'sqrt', max depth = 75, bootstrap = false
<b>Sequential NN</b>	Input layer – 512 units, regularizer = L2=0.001 First hidden layer – 256 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Third hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Fourth hidden layer – 32 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units Number of epochs form early stopping = 27, batch size = 500

Table 64 - Hyperparameters for optimization

Table 65 shows the performance results of after the tuning of hyper parameters. Based on the MAE Sequential NN seems to be outperforming the Random Forest model. Please refer Appendix L.1 for the learning curve plot.

	<b>MAE</b>	<b>MSE</b>	<b>RMSE</b>	<b>R Squared</b>
<b>Random Forest</b>	19.13 Minutes	1739.77	41.71	0.13
<b>Sequential NN</b>	18.33 Minutes	1784.96	42.25	0.11

Table 65 - Performance results after tuning

### 4.3.5 Stage Five: Introducing Convolutional Neural Network (CNN)

As mentioned in the classification stage experiments, the author had to reduce the sample size of the dataset because the CNN took around seven hours to train when using the stage three dataset. The author reduces the sample size to 80,000 from the stage two experiments.

	<b>Number of Records</b>	<b>Percentage</b>
<b>Total Data Records</b>	80000	100%
<b>Training</b>	51200	64%
<b>Testing</b>	16000	20%
<b>Validation</b>	12800	25% from training dataset

Table 66 - Regression, dataset distribution for stage five experiment

Table 66 shows the training, testing and validation dataset for the current experiments.

	<b>Training and testing parameter description</b>
<b>Random Forest</b>	N_Estimator = 100, minimum sample split = 15, min sample leaf=5, max features = 'sqrt', max depth = 75, bootstrap = false
<b>Sequential NN</b>	Input layer – 512 units, regularizer = L2=0.001 First hidden layer – 256 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Second hidden layer – 128 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Third hidden layer – 64 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Fourth hidden layer – 32 units, activation = 'relu', regularizer = L2=0.001 Dropout Layer = 0.5 Output layer - 1 units Number of epochs form early stopping = 25, batch size = 500
<b>Convolutional NN</b>	Input layer – CONV1D (Filters = 32, kernel size =7, activation = 'relu') First Hidden layer = CONV1D (Filters = 32, kernel size =7, activation = 'relu') Dropout Layer = 0.5 MaxPooling1D, Pool size = 2 Second hidden layer – 50 units, activation = 'relu' Output layer - 1 units, Number of epochs form early stopping = 19, batch size = 500

Table 67 - Regression Training and testing parameters for stage 5 experiment

Table 67 presents the training and testing parameters for the model, for the current experiment.

	<b>MAE</b>	<b>MSE</b>	<b>RMSE</b>	<b>R Squared</b>
<b>Random Forest</b>	20.78 Minutes	1731.32	41.61	0.08
<b>Sequential NN</b>	19.75 Minutes	1811.32	42.56	0.04
<b>Convolutional NN</b>	19.82 Minutes	1785.72	42.26	0.05

Table 68 - Regression performance metrics for stage five experiments

Based on the result on the table 68, it seems that the CNN is slightly outperforming the Sequential NN. Please refer to Appendix M for the learning plots of the created neural networks.

## 4.4 Summary

Each and every one of the experiments planned in the methodology chapter was able to execute and evaluated in this chapter. For the classification task, Random Forest model seems to be outperforming all the other models. For regression task Feed Forward Neural Network model seems to be the one with the minimum error rates. With the current configuration of convolutional neural network, it seems to be in the same level as the Feed Forward Neural Network for both classification and regression problems.



# Chapter 5: Conclusion

## 5.1 Overview

This chapter summarizes the current research carried out, while discussing findings and contributions. It will also point out the lessons learned and the future enhancements for the research to be extended. Summary of results from the chapter four evaluation will also be presented.

## 5.2 Classification Results

The goal of the classification task was to predict whether a flight will get delayed or not. Being a binary classification problem, the non-delayed cases were assumed as 0 and delayed cases were assumed as 1. Under five stages, number of experiments were carried out as mentioned in the section 3.9 focusing on the performance of the predictive models. From the first stage experiments it was decided that the weather data has an impact on the dataset, therefore the author used the flight and weather data for further experiments. Form the first three stages of experiments, which aims to find out how the different dataset proportions impact on the predictive models, the author chooses the third stage dataset and results as the optimum one even though the accuracy of the stage three models has reduced.

	Precision	Recall	F1-Score	Accuracy
<b>Stage Two Performance Results</b>				
<b>Logistic Regression</b>	0.33	0.56	0.41	0.64
<b>Random Forest</b>	0.76	0.06	0.12	0.78
<b>Decision Trees</b>	0.55	0.12	0.19	0.78
<b>Gaussian NB</b>	0.26	0.61	0.36	0.51
<b>Sequential NN</b>	0.61	0.10	0.17	0.78
<b>Stage Three Performance Results</b>				
<b>Logistic Regression</b>	0.61	0.61	0.61	0.61
<b>Random Forest</b>	0.65	0.58	0.61	0.63
<b>Decision Trees</b>	0.59	0.55	0.57	0.58
<b>Gaussian NB</b>	0.56	0.29	0.38	0.53
<b>Sequential NN</b>	0.61	0.59	0.61	0.61

Table 69 - Stage two and Stage three Experiments

Table 69 presents the performance result of the stage two and three experiments as a summary. Below Table 70 shows the confusion matrix for the stage two and stage three experiments.

	Actual Class	Predicted Class			
		Stage Two		Stage Three	
		Not Delayed %	Delayed(D) %	Not Delayed %	Delayed(D) %
Logistic Regression	ND	50.87	26.25	30.41	19.69
	D	10.16	12.72	19.68	30.22
Random Forest	ND	76.64	0.47	34.58	15.53
	D	21.43	1.46	20.98	28.92
Decision Trees	ND	74.93	2.19	30.58	19.52
	D	20.19	2.7	22.29	27.61
Gaussian NB	ND	36.69	40.42	38.7	11.41
	D	8.94	13.95	35.53	14.36
Sequential NN	ND	75.72	1.39	32.48	17.62
	D	20.68	2.2	20.56	29.34

Table 70 - Confusion matrix for stage two and stage three experiments

The author made the decision based on the confusion matrix. The second stage is an unbalanced dataset. The predictions are biased against the delayed flights. It is clearly visible from the TN and TP values in the confusion matrix. The Stage three experiments consist of a balanced dataset which contains equal portions of delayed and on time flights. The TP and TN values are seemed to be balanced. The increase of FP and FN values is an issue. But hyperparameter optimization made a slight difference. Below table 71 show the performance results for the stage for hyperparameter optimization.

	Precision	Recall	F1-Score	Accuracy
Random Forest	0.67	0.58	0.62	0.65
Sequential NN	0.63	0.61	0.61	0.62

Table 71 - Performance results for stage four

	Actual Class	Predicted Class	
		Not Delayed (ND) %	Delayed(D) %
Random Forest	ND	36.25	13.85
	D	21.19	28.71
Sequential NN	ND	32.57	17.53
	D	20.66	29.24

Table 72 - Stage four Classification parameter tuning

In the fourth stage author applied hyper parameter tuning and it increased the accuracy of the Random Forest model by 2%. Based on Table 72 the FN and FP rates in the confusion matrix were reduced slightly compared to the stage three results.

In stage five author introduces the convolutional NN to the experiments. Due to the massive amount of time consumed by the CNN to be trained, author had to reduce the sample size of the training, testing and validation for this experiment. The author wanted to experiment with a balanced an unbalanced data sample to see how the CNN model performs.

	Precision	Recall	F1-Score	Accuracy
<b>Performance results for unbalance dataset</b>				
<b>Random Forest</b>	0.69	0.10	0.17	0.78
<b>Sequential NN</b>	0.60	0.06	0.11	0.78
<b>Convolutional NN</b>	0.65	0.08	0.15	0.78
<b>Performance result for balance dataset</b>				
<b>Random Forest</b>	0.67	0.56	0.61	0.64
<b>Sequential NN</b>	0.64	0.53	0.58	0.61
<b>Convolutional NN</b>	0.64	0.56	0.59	0.62

Table 73 - Performance result for CNN for balance and unbalanced datasets

As presented in Table 73 CNN could not outperform Random Forest model. But it seems CNN is competing with Sequential model and wining in a slight margin. The CNN model was not optimized using random or grid search. This leaves the author with the impression that CNN model has the potential to surpass even the Random Forest model.

Stage six experiments were carried out to identify how the models perform when the delay threshold was reduced. Only the Random Forest and Sequential models were considered for this experiment due to large data size.

	Precision	Recall	F1-Score	Accuracy
<b>15 Minutes delay threshold</b>				
<b>Random Forest</b>	0.76	0.06	0.12	0.78
<b>Sequential NN</b>	0.61	0.10	0.17	0.78
<b>5 Minutes delay threshold</b>				
<b>Random Forest</b>	0.71	0.29	0.41	0.73
<b>Sequential NN</b>	0.59	0.21	0.31	0.70

Table 74 - Reduced delay threshold results

According to table 74 accuracy of the models have reduced when the delay threshold reduced. But overall performance metrics have been improved.

	Actual class	Predicted Class			
		15 Minutes delay threshold		5 Minutes delay threshold	
		Not Delayed %	Delayed(D) %	Not Delayed %	Delayed(D) %
Random Forest	ND	76.64	0.47	63.43	3.96
	D	21.43	1.46	23.14	9.48
Sequential NN	ND	75.72	1.39	62.6	4.79
	D	20.68	2.2	25.64	6.98

Table 75 - 15 min vs 5 min threshold confusion matrix

Table 75 shows the confusion matrix for the two delay thresholds. When the delay threshold is set to 5 minutes, a portion of the previous non delayed records moved to delayed state. This phenomenon is reflected in the table 75 matrix. TN rate reduced and TP rates increased in the 5 minted threshold results.

	Description	Overall Accuracy
<b>Bandyopadhyay et.al [10]</b>	Flight Data, weather data plus airplane information included Naïve Bayes, SVM and Random Forest model 80,000 data sample Unbalanced Dataset	70%
<b>Current Research</b>	Flight Data plus weather data included Random Forest, FFNN, CNN 80,000 data sample Unbalanced Dataset	78%

Table 76 - Literature Review findings for classification using flight and weather data

Table 76 shows a research identified by the author in the Chapter Two: Literature Review (LR). Bandyopadhyay et.al' experiment is somewhat similar to the stage five experiment. The weather data features are bit different since Bandyopadhyay used binary variable to represent some of the weather features while current research uses the actual values. They have included the airplane information as well. Overall, the current research model seems to be having a higher accuracy than the Bandyopadhyay et.al' experiment.

	Description	Overall Accuracy
<b>Nathalie et.al [11]</b>	Only flight data 100000 sample of flight records Imbalance data sample Random Forest, Decision Tress and Neural network with one hidden layer	90%
<b>Venkatesh et.al [13]</b>	Only flight data 100000 sample of flight records Imbalance data sample Random Forest, Decision Tress and Neural network with one hidden layer	CNN – 76% FFNN – 92%
<b>Current Research</b>	Only flight data 325761of training data sample Imbalance data sample Random Forest and NN	80%

Table 77 - Literature Review findings for classification using only flight data

Table 77 shows the results from the LR, where only flight data being used. Again, these experiments are also slightly different form the author’s experiments. Author’s stage one experiment one seems to be the most similar to the LR findings. Having a higher data sample may be the cause for the lesser accuracy of the current research. And the authors models are not even tuned for hyperparameters.

### 5.3 Regression Results

The goal of the regression task was to predict the delay time in minutes. Under five stages number of experiments were carried out as mentioned in the section 3.9 focusing on the performance of the predictive models. Same as the classification problem, both flight data and weather datasets were used to do the experiments. Form the first three stages it was evident that the stage one experiment provided the highest performance. The stage one dataset contains flight record of the year 2018. It seems with more data the error rate increases. Therefore, the author decides to stick with the stage one experiment.

	MAE	MSE	RMSE	R Squared
<b>Linear Regression</b>	22.46 Minutes	1880.48	43.36	0.05
<b>Random Forest</b>	20.71 Minutes	1779.13	42.18	0.10
<b>Decision Trees</b>	20.49 Minutes	1822.52	42.69	0.07
<b>Sequential NN</b>	19.54 Minutes	1885.30	43.42	0.11

Table 78 - Regression highest performance from stage one to three experiments

Table 78 shows the stage one experiment three results. Form the above result Random Forest and Sequential NN was considered for further tuning. Below table shows the tuned results.

	MAE	MSE	RMSE	R Squared
<b>Random Forest</b>	19.13 Minutes	1739.77	41.71	0.13
<b>Sequential NN</b>	18.33 Minutes	1784.96	42.25	0.11

Table 79 - Regression results after hyperparameter tuning

Both predictive modals have a considerable improvement over the results shown in table 79. Sequential Models MAE was reduced by one minute.

	MAE	MSE	RMSE	R Squared
<b>Random Forest</b>	20.78 Minutes	1731.32	41.61	0.08
<b>Sequential NN</b>	19.75 Minutes	1811.32	42.56	0.04
<b>Convolutional NN</b>	19.82 Minutes	1785.72	42.26	0.05

Table 80 - Regression CNN results

Table 80 shows the CNN result with the other regression models with a reduced data sample. The current configuration of CNN has similar results to the tuned regression models. Therefore, CNN seems to have a hidden potential left to uncover by hyperparameter tuning.

	Description	Overall Accuracy
<b>Bandyopadhyay et.al [10]</b>	Flight Data, weather data plus airplane information included Linear Regression 4500 training and 1500 testing data	35 Minutes MAE
<b>Current Research</b>	Flight and weather data Random Forest and NNN 325761of training data sample	18 Minutes MAE

Table 81 - Literature review findings for regression

As shown in the Table 81 Bandyopadhyay et.al are the only ones who performed experiments somewhat similar to authors experiments. Bandyopadhyay et.al and author both uses the same flight dataset. Only difference being in that the author does not include airplane information in his dataset. The current research is outperforming Bandyopadhyay's models by a huge margin. The large data sample played major role for the difference in the results.

## **5.4 Future Enhancements**

The improvements for the current research can be started with using an embedding column to encode the categorical data. Instead of representing the data using one-hot vector of many dimensions, this embedding column can be used to represent the data as a lower-dimensional, dense vector. In this approach each cell can contain any number, not just 0 or 1.

Dividing the flight record into different seasons of the year and conducting experiments will be interesting. May be predictive models for each season can be separately developed.

Convolution neural network need to be further fine-tuned. Unfortunately, the author's computer processing power was not enough to do a proper fine tuning of the CNN.

Dividing the delays into multiple classes, e.g.: 'short delays, medium delays, large delays' and performing multiclass classification is also possible.

## **5.5 Summary**

Flight delay prediction is an important feature to airports, airlines and passengers. With the current technology, the passenger can get the real time data of his or her flight if it is traveling from one destination to another. But predating the flight delay couple of hours earlier has its advantages. Weather is a one of the delays causes out of many for flight delays. These predictive models need to be infused with the data related to other delay causes to get the proper results. The author was able to utilize the currently available datasets to successfully generate predictive models that can solve the current research problem.

## References

- [1]. Bogicevic, V., Yang, W., Bilgihan, A., Bujisic, M., (2013). Airport service quality drivers of passenger satisfaction. *Tourism Review* 68, 3–18. Doi:10.1108/TR-09-2013-0047.
- [2]. Worldometers.info, (2020). World Population Clock: 7.3 Billion People (2020) - Worldometers. [Online] Available at: <http://www.worldometers.info/world-population/> Accessed 21 Jan. 2020].
- [3]. Iata.org, (2014). IATA - New IATA Passenger Forecast Reveals Fast-Growing Markets of the Future. [Online] Available at: <http://www.iata.org/pressroom/pr/Pages/2014-10-16-01.aspx> [Accessed 21 Jan. 2020].
- [4]. Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A., Zou, B., others, 2010. Total delay impact study. Institute of Transportation Studies, University of California, Berkeley.
- [5] O. Simeone, “A Brief Introduction to Machine Learning for Engineers,” *arXiv:1709.02840 [cs, math, stat]*, May 2018, Accessed: Jan 15, 2020. [Online]. Available: <http://arxiv.org/abs/1709.02840>.
- [6] United States. Department Of Transportation. Bureau Of Transportation Statistics, “National Transportation Statistics (series),” 2019, doi: [10.21949/1503663](https://doi.org/10.21949/1503663).
- [7]. “Understanding the Reporting of Causes of Flight Delays and Cancellations | Bureau of Transportation Statistics.” [Online]. Available: <https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations>. [Accessed: 25-Oct-2019].
- [8]. Cabanillas, C., Campara, E., Koziel, B., Di Ciccio, C., Mendling, J., Paulitschke, J., Prescher, J., 2014. Towards a Prediction Engine for Flight Delays based on Weather Delay Analysis., in: *EMoV+ MinoPro@ Modellierung*. Citeseer, pp. 49–51.
- [9]. Mathur.A, Nagao.A, Kenny.N, 2013. Predicting flight on-time performance.
- [10]. Bandyopadhyay, R.J., Guerrero, R., (2012). Predicting airline delays.
- [11]. N. Kuhn and N. Jamadagni, “Application of Machine Learning Algorithms to Predict Flight Arrival Delays,” p. 6, 2017.
- [12]. S. Oza, S. Sharma, H. Sangoi, R. Raut, and V. C. Kotak, “Flight Delay Prediction System Using Weighted Multiple Linear Regression,” vol. 4, no. 4, p. 9, 2015.
- [13]. V. Venkatesh, A. Arya, P. Agarwal, S. Lakshmi, and S. Balana, “Iterative machine and deep learning approach for aviation delay prediction,” in 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON), Mathura, 2017, pp. 562–567.
- [14] T. Zhou, Q. Gao, X. Chen, and Z. Xun, “Flight Delay Prediction Based on Characteristics of Aviation Network,” *MATEC Web Conf.*, vol. 259, p. 02006, 2019.



- [15]. Sridhar, B., Wang, Y., Klein, A., Jehlen, R., 2009. Modeling Flight Delays and Cancellations at the National, Regional and Airport Levels in the United States, in: 8th USA/Europe ATM R&D Seminar, Napa, California (USA).
- [16]. Rebollo, J.J., Balakrishnan, H., 2014. Characterization and prediction of air traffic delays. *Transportation Research Part C: Emerging Technologies* 44, 231–241.
- [17] V. Roman, “Machine Learning Introduction: A Comprehensive Guide,” Medium, 10-Mar- 2019. [Online]. Available: <https://towardsdatascience.com/machine-learning-introduction-a-comprehensive-guide-af6712cf68a3>. [Accessed: 23-Jan-2020].
- [18] A. Géron, “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems,” p. 718.
- [19] “Airline On-Time Statistics and Delay Causes.” [https://www.transtats.bts.gov/OT\\_Delay/OT\\_DelayCause1.asp](https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp) (accessed Jun. 11, 2020).
- [20] “Datasets | Climate Data Online (CDO) | National Climatic Data Center (NCDC).” [Online]. Available: <https://www.ncdc.noaa.gov/cdo-web/datasets/>. [Accessed: 23-Jan-2020].
- [21] “The Biggest and Busiest Airports in the US in 2020,” *The ClaimCompass Blog*, Mar. 26, 2020. <https://www.claimcompass.eu/blog/biggest-busiest-us-airports/> (accessed March 02, 2020).
- [22] V. Martinez, “Flight Delay Prediction,” 2012, doi: [10.3929/ethz-a-007139937](https://doi.org/10.3929/ethz-a-007139937).
- [23] “Federal Holidays in USA in 2017,” *Office Holidays*. <https://www.officeholidays.com/countries/usa/2017> (accessed Jun. 11, 2020).
- [24] K. Potdar, T. Pardawala, and C. Pai, “A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers,” *International Journal of Computer Applications*, vol. 175, pp. 7–9, Oct. 2017, doi: [10.5120/ijca2017915495](https://doi.org/10.5120/ijca2017915495).
- [25] “Linear Regression - an overview | ScienceDirect Topics.” <https://www.sciencedirect.com/topics/social-sciences/linear-regression> (accessed Jun. 11, 2020).
- [26] S. Sperandei, “Understanding logistic regression analysis,” *Biochem Med (Zagreb)*, vol. 24, no. 1, pp. 12–18, Feb. 2014, doi: [10.11613/BM.2014.003](https://doi.org/10.11613/BM.2014.003).
- [27] Y. SONG and Y. LU, “Decision tree methods: applications for classification and prediction,” *Shanghai Arch Psychiatry*, vol. 27, no. 2, pp. 130–135, Apr. 2015, doi: [10.11919/j.issn.1002-0829.215044](https://doi.org/10.11919/j.issn.1002-0829.215044).
- [28] W.-Y. Loh, “Classification and Regression Trees,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, pp. 14–23, Jan. 2011, doi: [10.1002/widm.8](https://doi.org/10.1002/widm.8).
- [29] M. Denil, D. Matheson, and N. de Freitas, “Narrowing the Gap: Random Forests In Theory and In Practice,” p. 9.

- [30] P. Kaviani and S. Dhotre, "Short Survey on Naive Bayes Algorithm," *International Journal of Advance Research in Computer Science and Management*, vol. 04, Nov. 2017.
- [31] S. Sakib, Ahmed, A. Jawad, J. Kabir, and H. Ahmed, *An Overview of Convolutional Neural Network: Its Architecture and Applications*. 2018.
- [32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," p. 30.
- [34] "tf.keras.callbacks.EarlyStopping | TensorFlow Core v2.2.0," *TensorFlow*. [https://www.tensorflow.org/api\\_docs/python/tf/keras/callbacks/EarlyStopping](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping) (accessed Jun. 16, 2020).
- [35] "sklearn.model\_selection.GridSearchCV — scikit-learn 0.23.1 documentation." [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) (accessed Jun. 21, 2020).
- [36] "sklearn.model\_selection.RandomizedSearchCV — scikit-learn 0.23.1 documentation." [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.RandomizedSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html) (accessed Jun. 21, 2020).
- [37] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, Aug. 2018, doi: [10.1016/j.aci.2018.08.003](https://doi.org/10.1016/j.aci.2018.08.003).
- [38] Hossin, Mohammad & M.N, Sulaiman. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*. 5. 01-11. 10.5121/ijdkp.2015.5201.
- [39] F. Moksony, "Small Is Beautiful: The Use and Interpretation of R2 in Social Research," *Szociologiai Szemle*, pp. 130–138, Jan. 1999.

# Appendix

## Appendix A - External Libraries for programming

Library	Version
Python	3.7
Anaconda	1.9.12
TensorFlow	2.0.0

## Appendix B – Dataset Exploratory Analysis

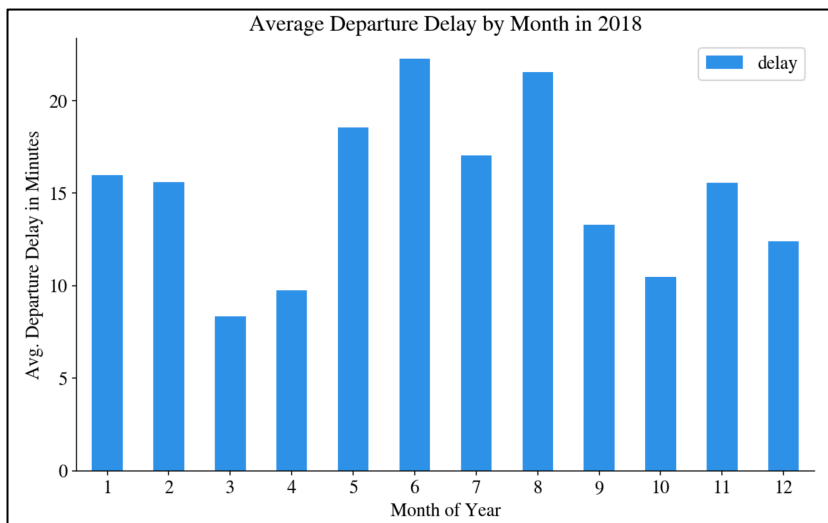


Figure 18 - Average departure delay by month in year 2018

Figure 18 shows the average departure delay in minutes for each month of the year. Seems like June has the highest average delay in 2018.

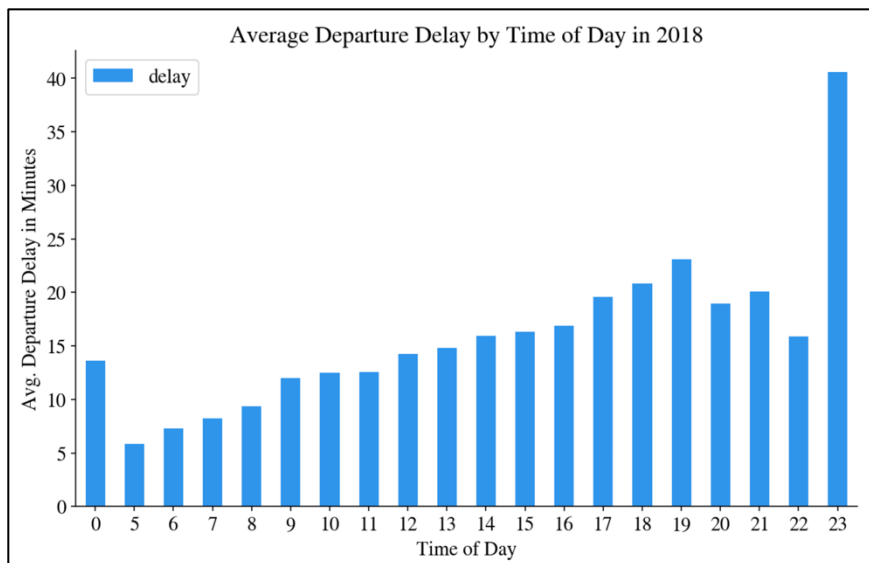


Figure 19 - Average departure delay by time of the day in year 2018

This figure 19 shows the average departure delay in minutes for the time of the day. It seems the highest delay average is around 11pm.

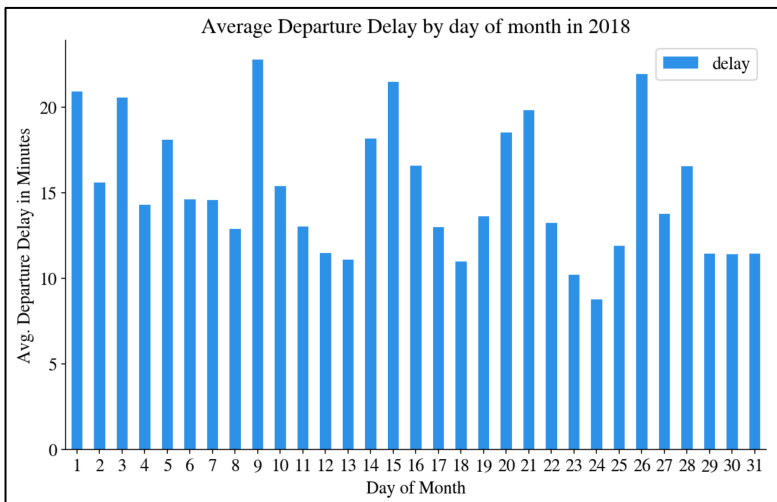


Figure 20 - Average departure delay by the day of the month in year 2018

Figure 20 shows the average departure delay in minutes for the day of the month.

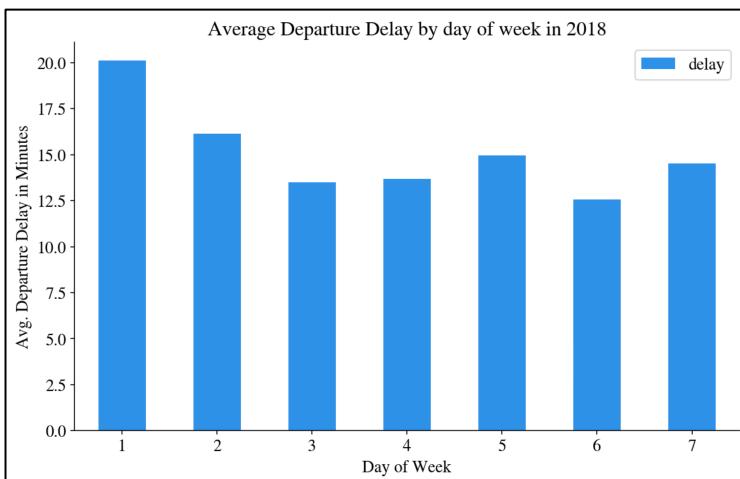


Figure 21 - Average departure delay by the day of the week in the year 2018

Figure 21 shows the average departure delay for the day of the week. Monday has the highest avg delay.

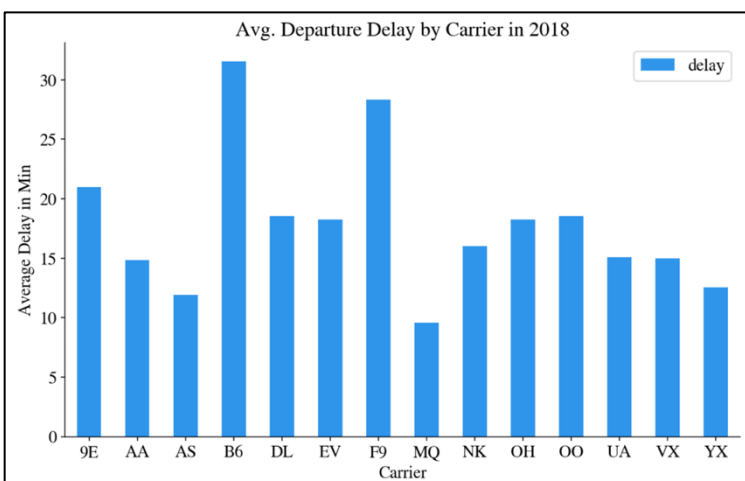
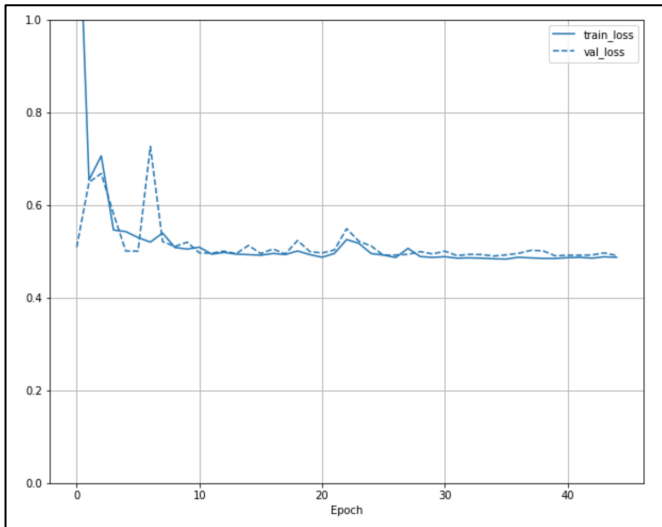


Figure 22 - Average delay by the carrier in the year 2018

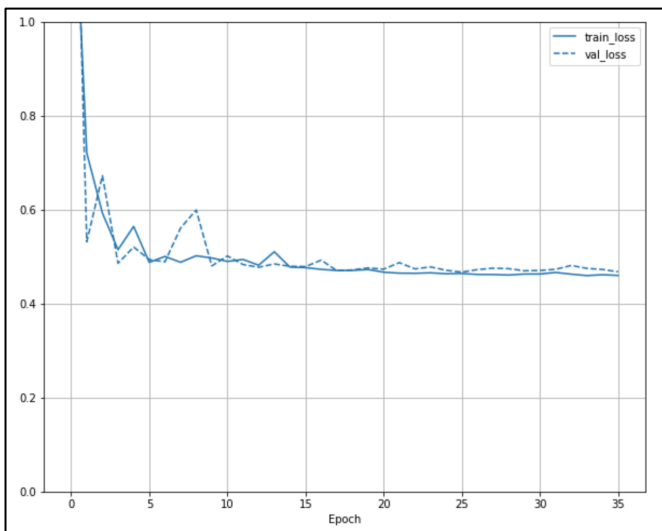
Figure 22 shows the average delay vs carrier. Seems like B6 – Jetblue Airways cooperation has the highest average delay.

## Appendix C – Flight Delay Classification, Stage One

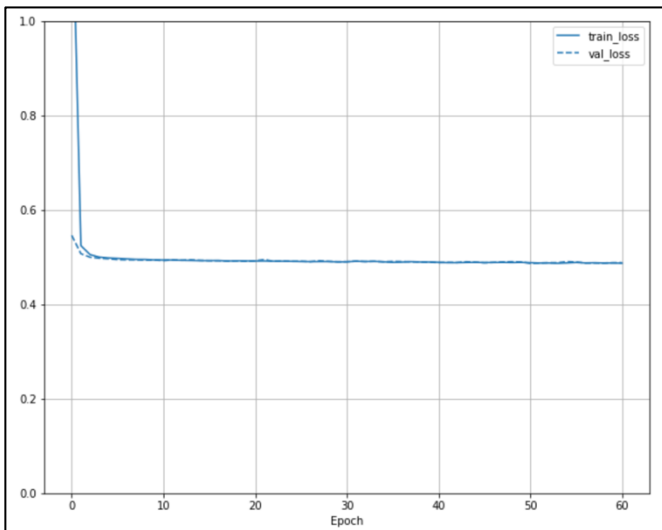
### C.1 Experiment One – NN Learning Curve



### C.2 Experiment Two – NN Learning Curve

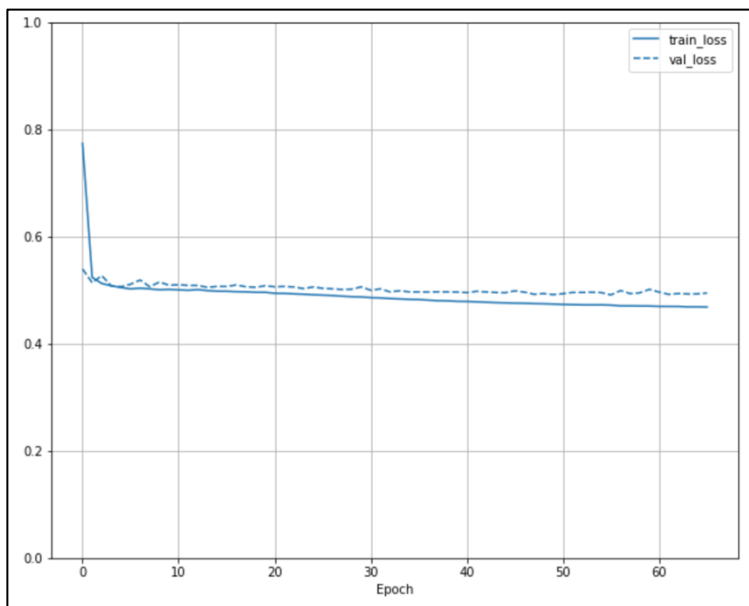


### C.3 Experiment three - NN Learning Curve

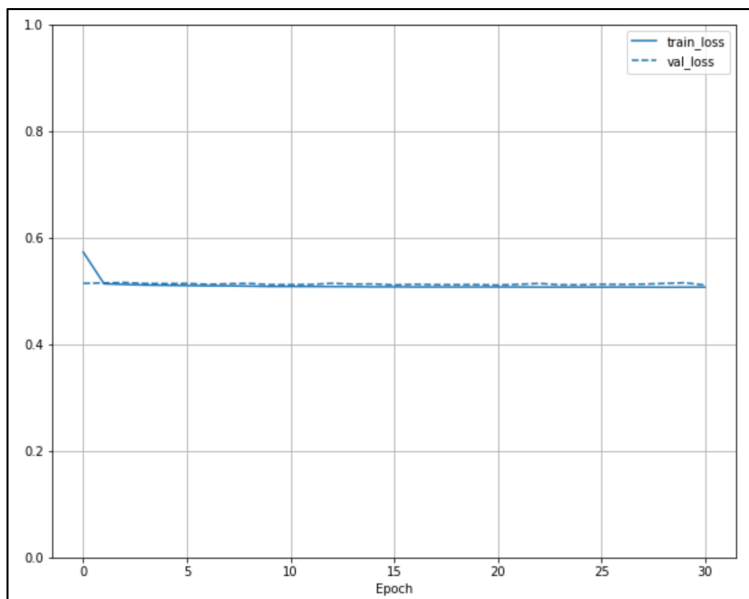


## Appendix D - Flight Delay Classification, Stage Two

### D.1 Experiment One – NN Learning Curve



### D.2 Experiment Two – NN Learning Curve

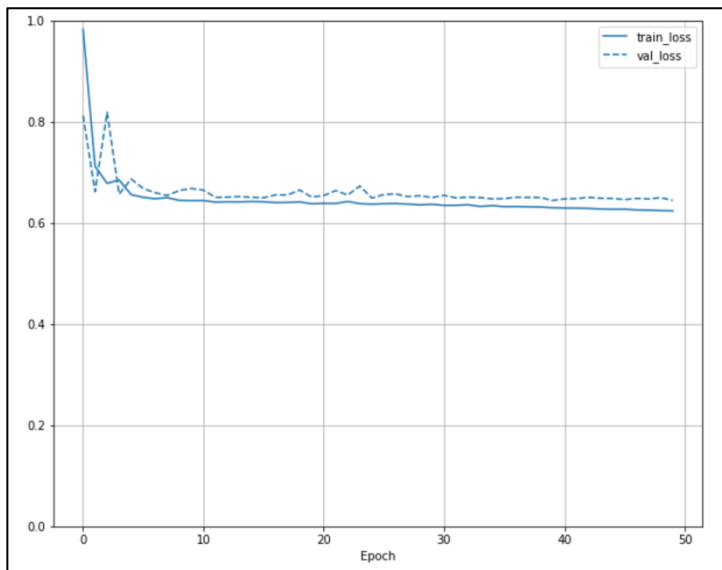


### D.3 Experiment Two – Min-Max Normalization Results

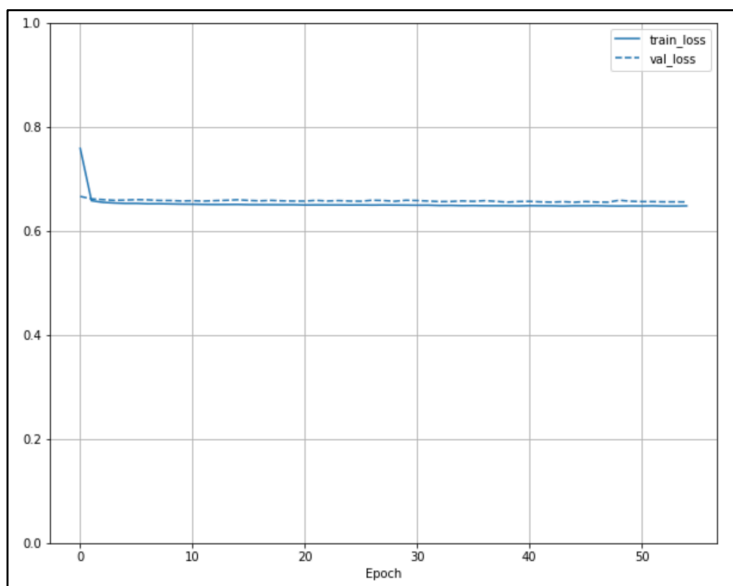
	Precision	Recall	F1-Score	Accuracy
<b>Logistic Regression</b>	0.33	0.56	0.41	0.64
<b>Random Forest</b>	0.76	0.66	0.12	0.78
<b>Decision Trees</b>	0.55	0.12	0.19	0.78
<b>Gaussian NB</b>	0.25	0.66	0.36	0.47
<b>Sequential NN</b>	0.71	0.01	0.02	0.77

## Appendix E – Flight Delay Classification, Stage Three

### E.1 Experiment One – NN Learning Curve



### E.2 Experiment Two – NN Learning Curve

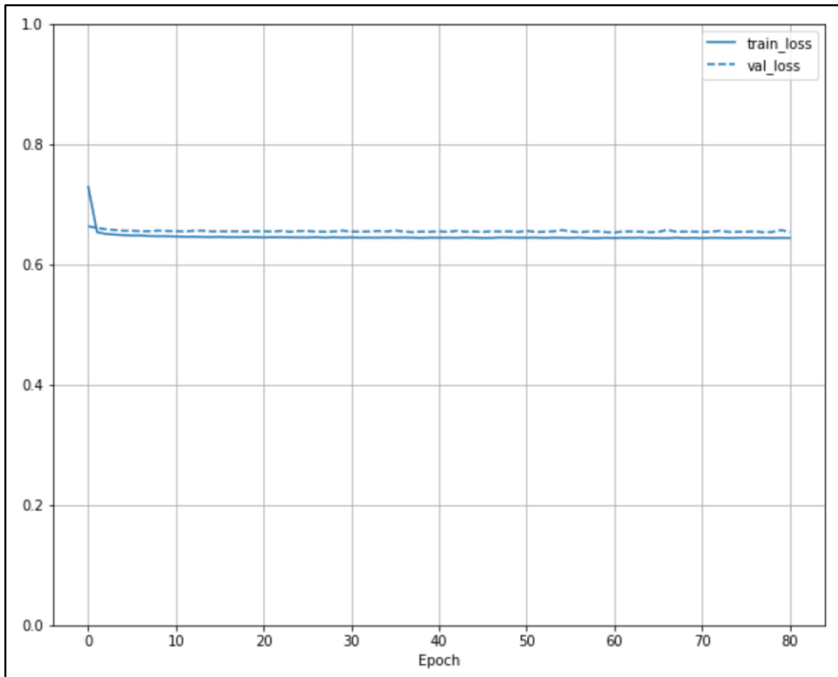


### E.3 Experiment Two – Min-Max Normalization Results

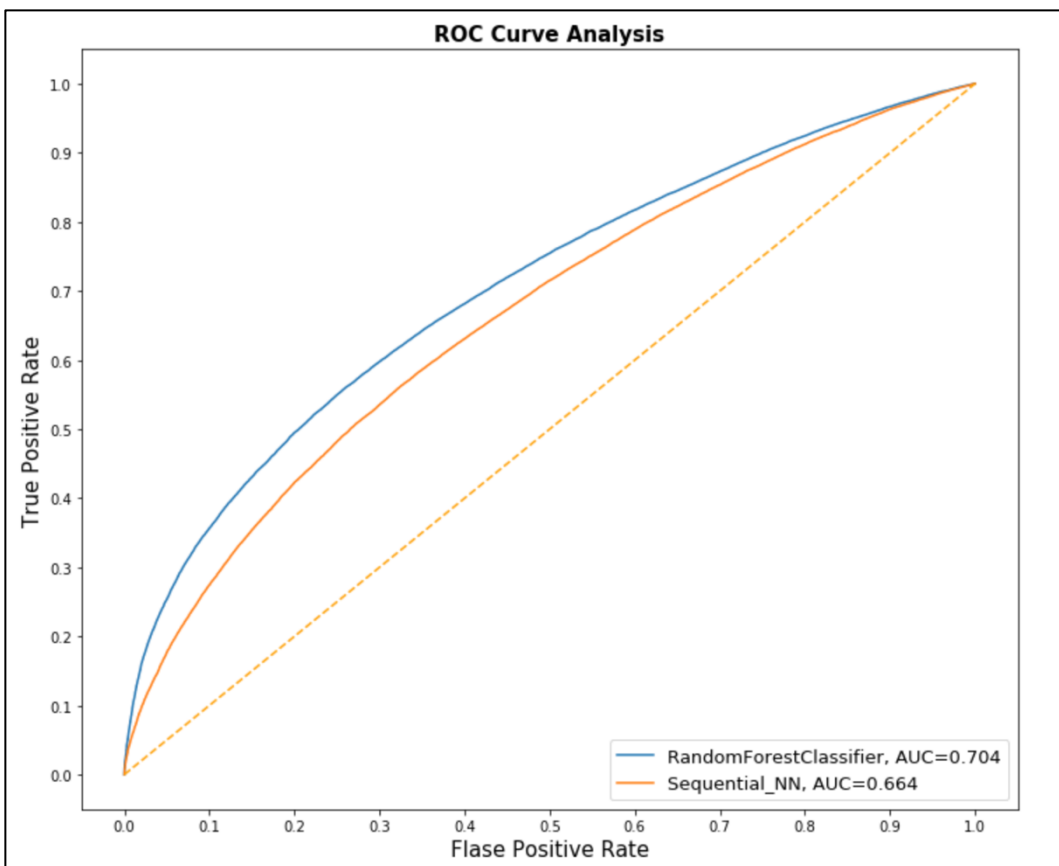
	Precision	Recall	F1-Score	Accuracy
<b>Logistic Regression</b>	0.61	0.60	0.60	0.61
<b>Random Forest</b>	0.65	0.58	0.61	0.63
<b>Decision Trees</b>	0.59	0.55	0.57	0.58
<b>Gaussian NB</b>	0.56	0.26	0.36	0.53
<b>Sequential NN</b>	0.61	0.63	0.62	0.61

# Appendix F – Flight Delay Classification, Stage Four

## F.1 Experiment – Learning Curve



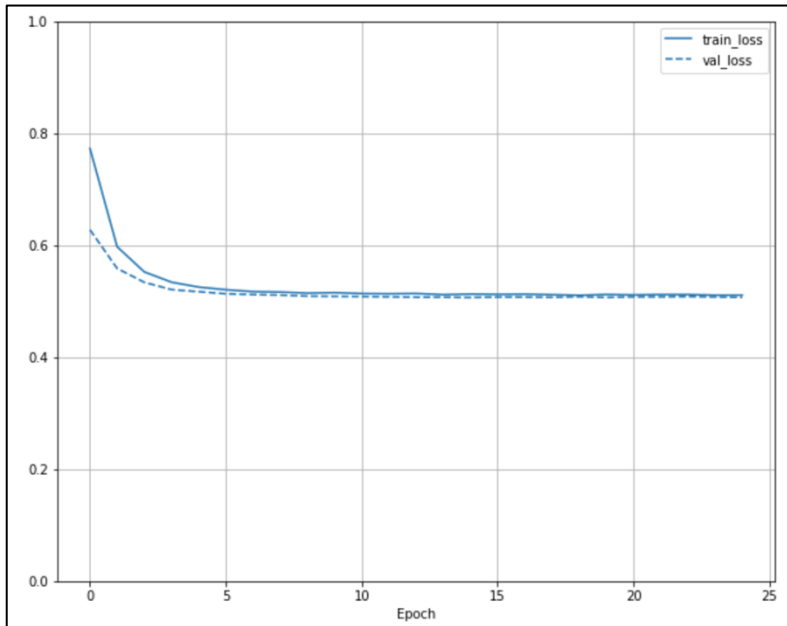
## F.2 Experiment - ROC Curve



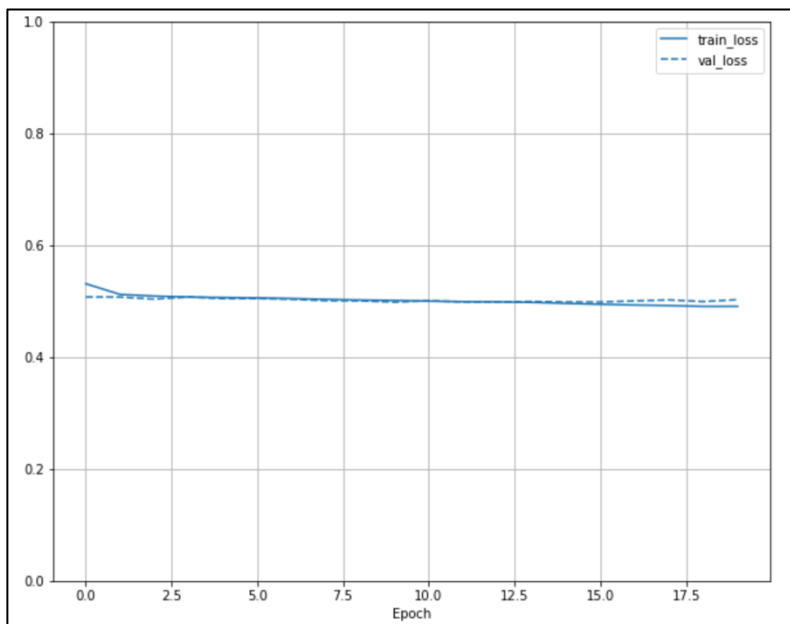


## Appendix G - Flight Delay Classification, Stage Five

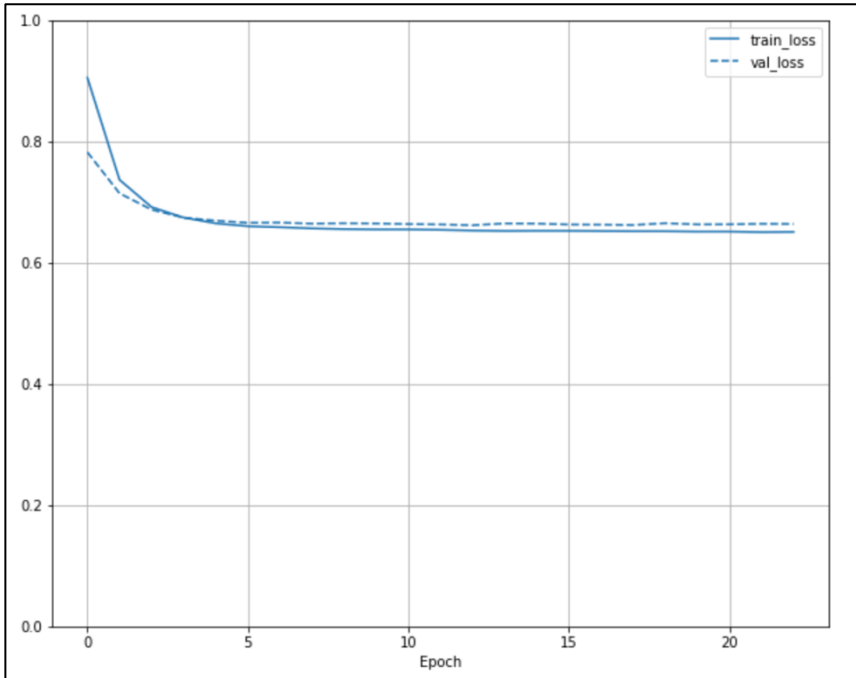
### G.1 Sequential NN – Learning Curve – Unbalance Dataset



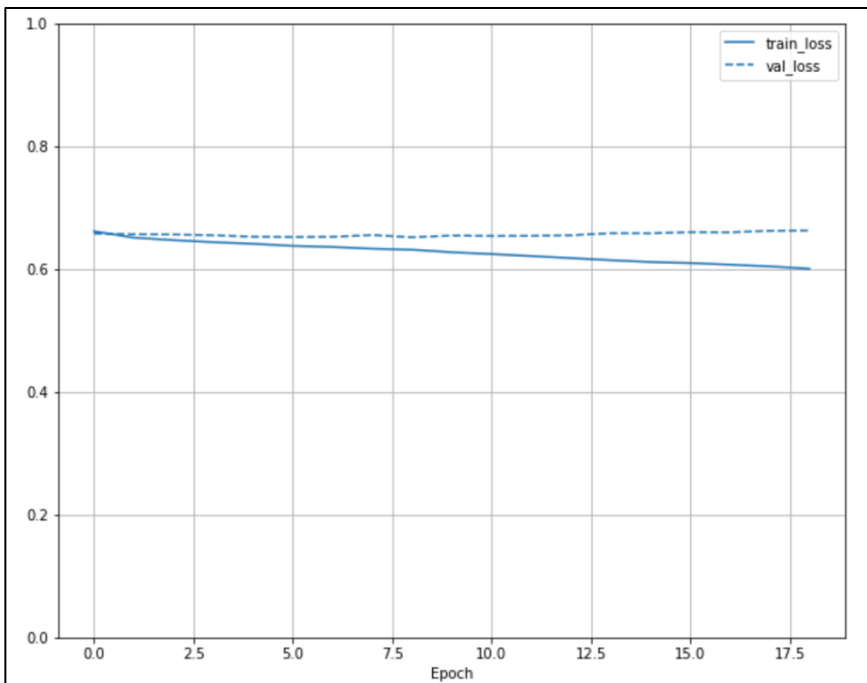
### G.2 Convolutional NN – Learning Curve – Unbalanced Dataset



### G.3 Sequential NN – Learning Curve – Balance Dataset

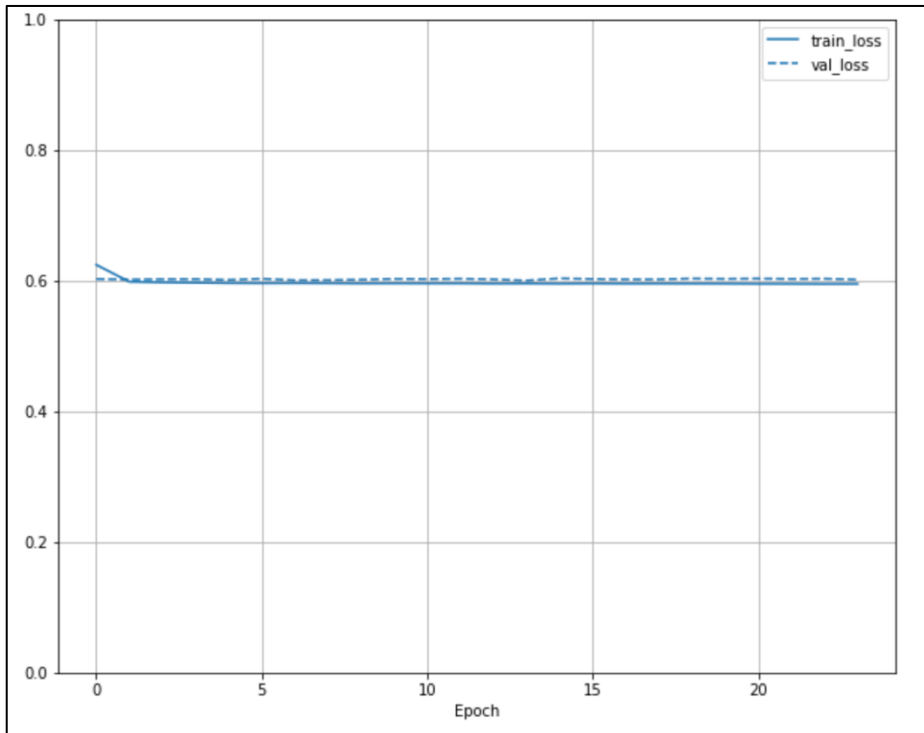


### G.4 Convolutional NN – Learning Curve – Balance Dataset



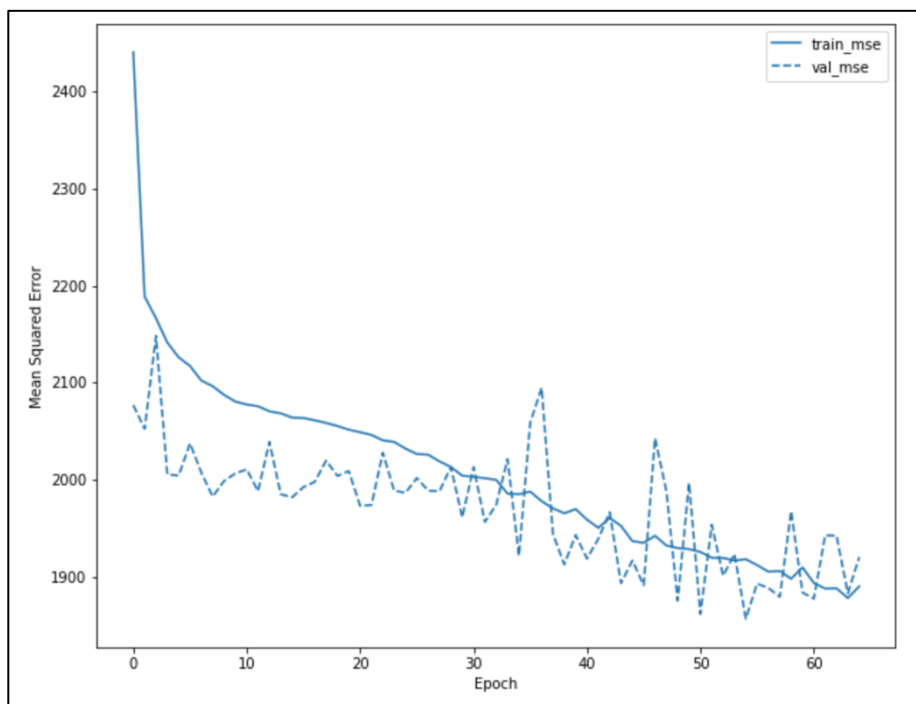
## Appendix H - Flight Delay Classification, Stage Six

### H.1 Sequential NN – Learning Curve

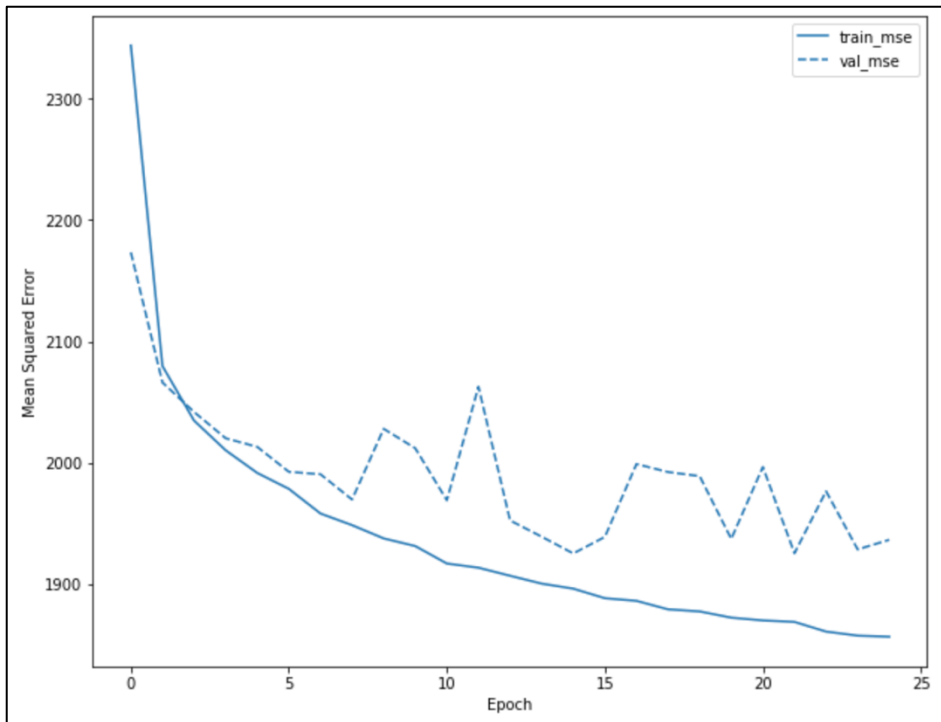


## Appendix I – Flight Delay Regression – Stage One

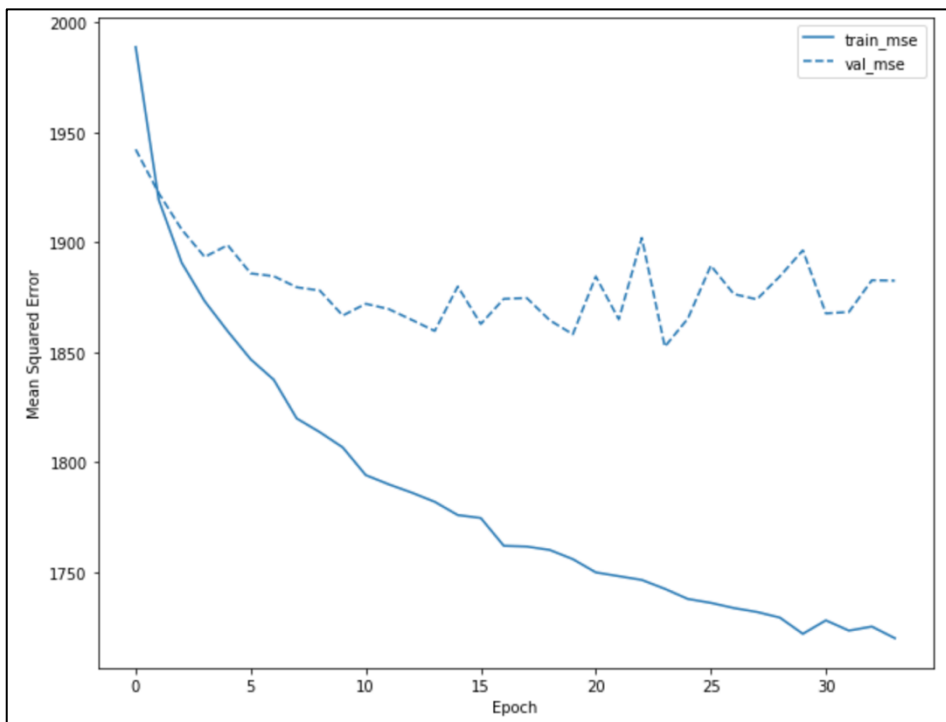
### I.1 Experiment One - Sequential NN



## I.2 Experiment Two – Sequential NN

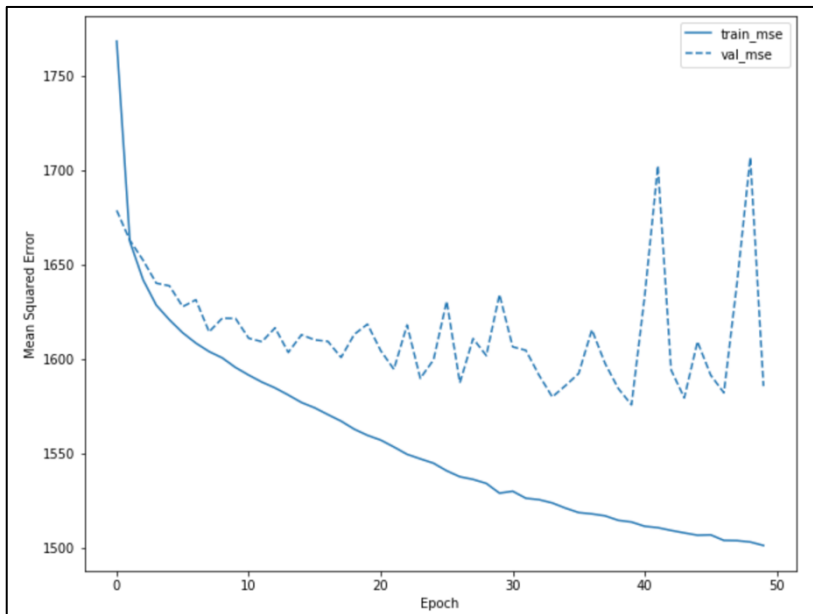


## I.3 Experiment Three – Sequential NN

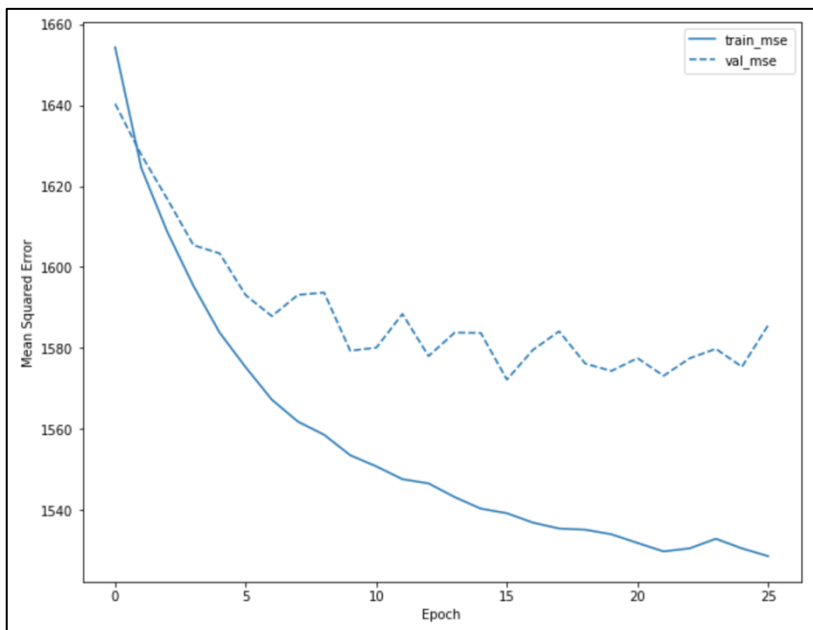


## Appendix J – Flight Delay Regression – Stage Two

### J.1 Experiment One - Sequential NN



### J.2 Experiment Two – Sequential NN

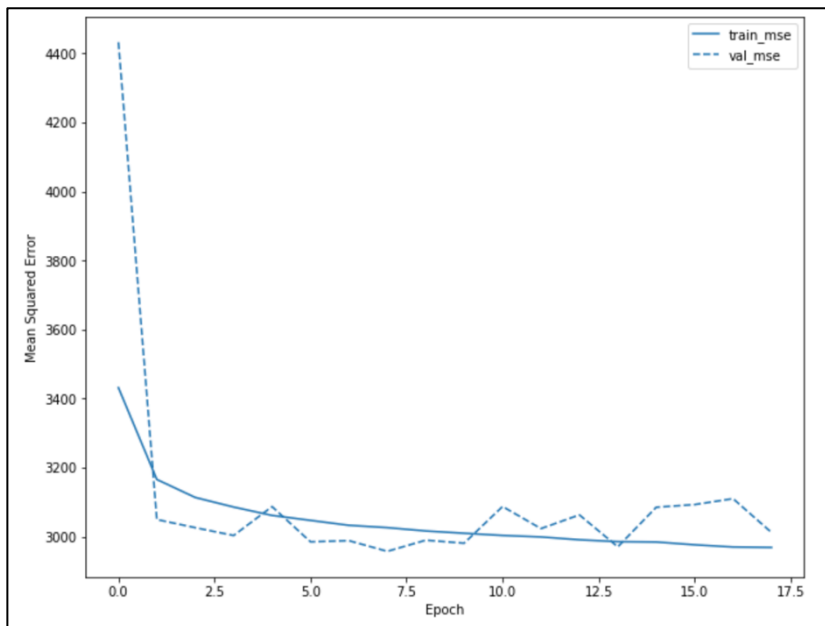


### J.3 Experiment Two – Min-Max Normalization

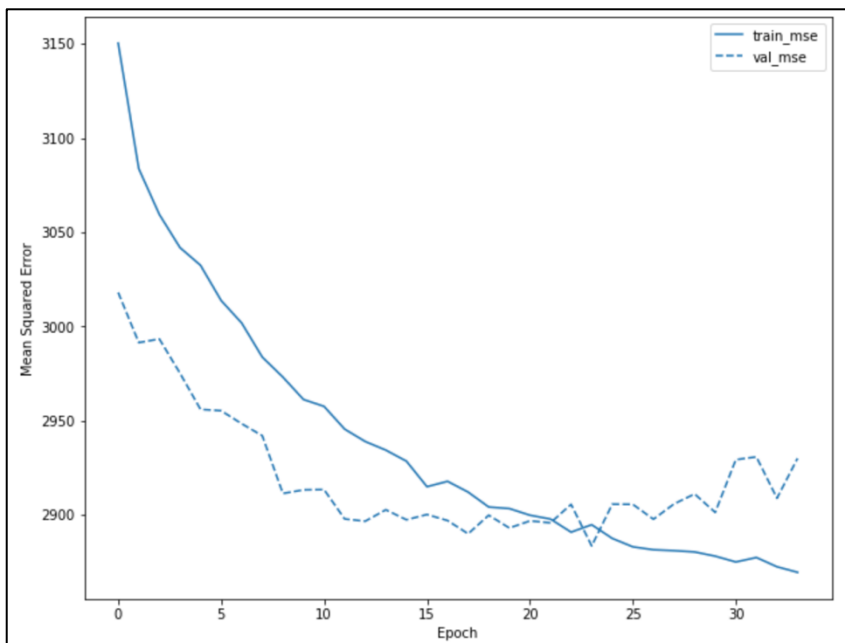
	MAE	MSE	RMSE	R Squared
<b>Linear Regression</b>	20.97 Minutes	1706.62	41.31	0.05
<b>Random Forest</b>	20.35 Minutes	1628.94	40.36	0.09
<b>Decision Trees</b>	20.71 Minutes	1702.15	41.26	0.05
<b>Sequential NN</b>	19.85 Minutes	1602.43	40.03	0.11

## Appendix K – Flight Delay Regression – Stage Three

### K.1 Experiment One - Sequential NN



### K.2 Experiment Two – Sequential NN

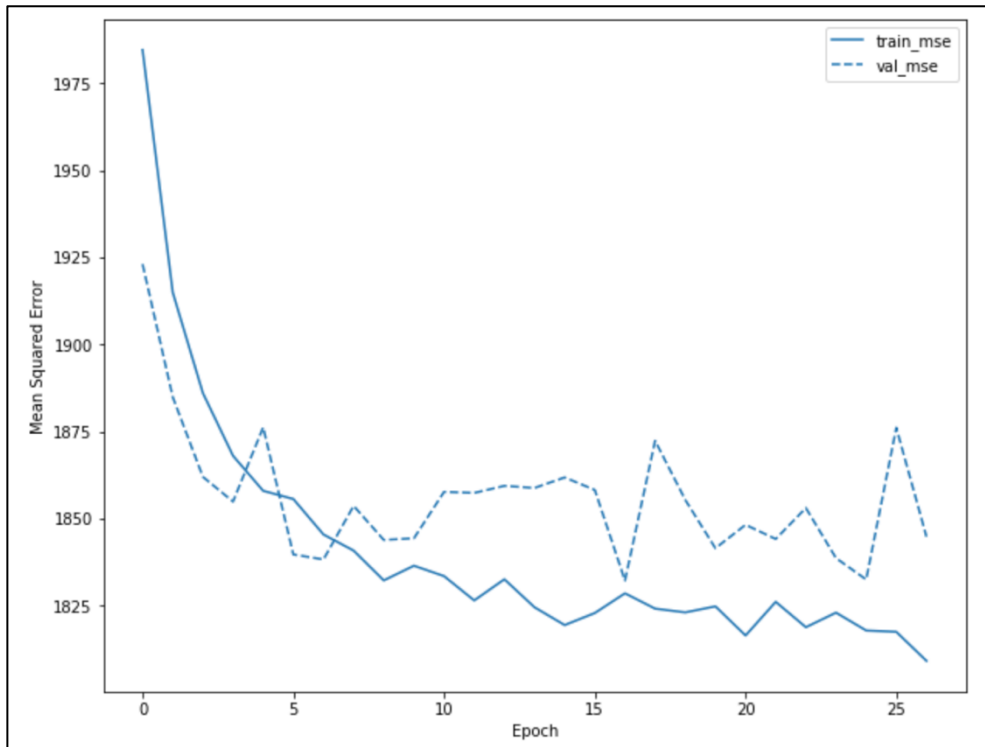


### K.3 Experiment Two – Min-Max Normalization

	MAE	MSE	RMSE	R Squared
<b>Linear Regression</b>	35.18 Minutes	3242.35	56.95	0.03
<b>Random Forest</b>	33.18 Minutes	3085.27	55.55	0.08
<b>Decision Trees</b>	34.56 Minutes	3316.06	57.59	0.01
<b>Sequential NN</b>	33.35 Minutes	3103.49	55.68	0.08

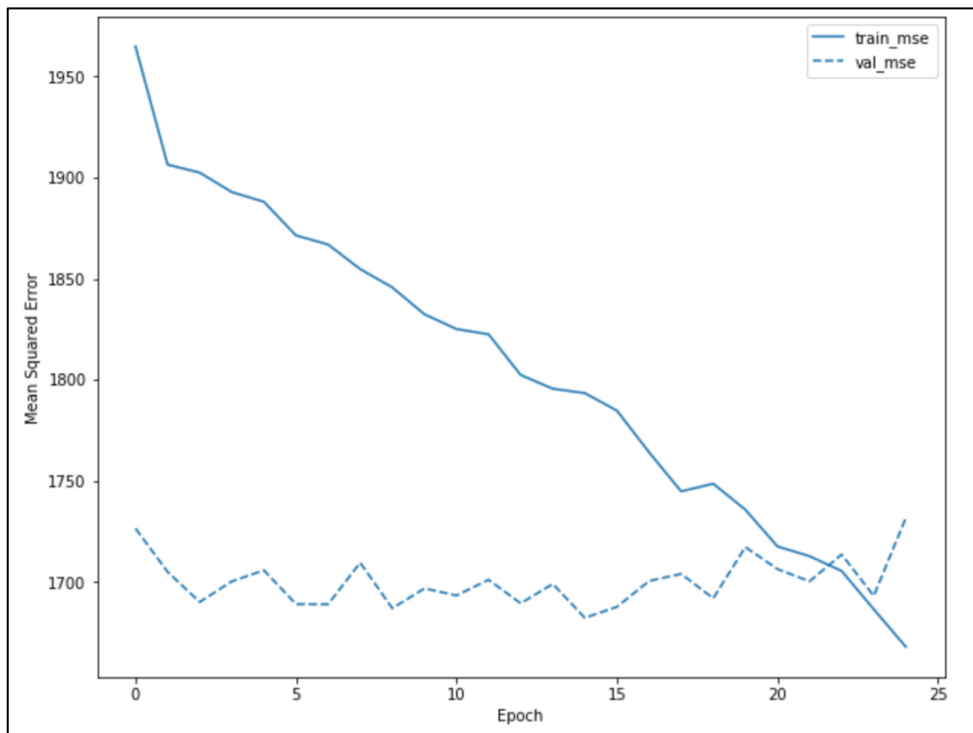
## Appendix L – Flight Delay Regression – Stage Four

### L.1 Experiment - Sequential NN



## Appendix M – Flight Delay Regression – Stage Five

### M.1 Experiment – Sequential NN



## M.2 Experiment – Convolutional NN

