2019

Airline fleet planning and utilization hours comparison studies

Daniel Zhou
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Aerospace Engineering Commons

Recommended Citation
Zhou, Daniel, "Airline fleet planning and utilization hours comparison studies" (2019). Graduate Theses and Dissertations. 17131.
https://lib.dr.iastate.edu/etd/17131

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Airline fleet planning and utilization hours comparison studies

by

Daniel Xiaoyang Zhou

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Aerospace Engineering

Program of Study Committee:
Peng Wei, Major Professor
Leifur T. Leifsson
Lizhi Wang

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this thesis. The Graduate College will ensure this thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2019

Copyright © Daniel Xiaoyang Zhou, 2019. All rights reserved.
DEDICATION

I would like to dedicate this thesis to my Mum and Dad without whose support I would not have been able to complete this work. I would also like to thank my friends and family for their loving guidance and financial assistance during the writing of this work.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>ix</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>x</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xii</td>
</tr>
<tr>
<td>CHAPTER 1. OVERVIEW</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 2. PROBLEM STATEMENT</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER 3. AIRPLANE UTILIZATION AND TREND STUDY</td>
<td>6</td>
</tr>
<tr>
<td>3.1 Method</td>
<td>6</td>
</tr>
<tr>
<td>3.2 Results</td>
<td>11</td>
</tr>
<tr>
<td>3.2.1 Utilization Hour Comparison of Tail Numbers</td>
<td>11</td>
</tr>
<tr>
<td>3.2.2 Average Utilization Hour Comparison Between Legacy Carriers and Low-Cost Carriers</td>
<td>18</td>
</tr>
<tr>
<td>CHAPTER 4. AIRPLANE UTILIZATION PREDICTION VIA MACHINE LEARNING MODELS</td>
<td>22</td>
</tr>
<tr>
<td>4.1 Method</td>
<td>22</td>
</tr>
<tr>
<td>4.1.1 K-Fold Cross-Validation</td>
<td>24</td>
</tr>
<tr>
<td>4.1.2 Preliminary Results</td>
<td>25</td>
</tr>
<tr>
<td>4.1.3 Random Forest Model</td>
<td>27</td>
</tr>
<tr>
<td>4.1.4 Support Vector Regression Model</td>
<td>28</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>4.1.5</td>
<td>Neural Network Model</td>
</tr>
<tr>
<td>4.1.6</td>
<td>Boosting Models</td>
</tr>
<tr>
<td>4.2</td>
<td>Results</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Random Forest Algorithm Prediction Results</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Support Vector Regression Algorithm Results</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Neural Network Prediction Results</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Boosting Algorithms Prediction Results</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Summary of Machine Learning Model Predictions</td>
</tr>
</tbody>
</table>

CHAPTER 5. CONCLUSION AND FUTURE WORK .............................................. 60

BIBLIOGRAPHY ....................................................................................... 62
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3.1</td>
<td>A Summary of AOTP Database Fields</td>
<td>6</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Machine Learning Model Predictions</td>
<td>58</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Erroneous Data in Raw Data Example I</td>
<td>8</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Erroneous Data in Raw Data Example II</td>
<td>8</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Time Zone Problem in Raw Data Example</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>American Airlines A319 Single Tail Number Comparison 2014</td>
<td>11</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>American Airlines A319 Single Tail Number Comparison 2017</td>
<td>12</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Frontier Airlines A319 Single Tail Number Comparison 2014</td>
<td>12</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>Frontier Airlines A319 Single Tail Number Comparison 2017</td>
<td>13</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>American Airlines B737 Single Tail Number Comparison 2014</td>
<td>14</td>
</tr>
<tr>
<td>Figure 3.9</td>
<td>American Airlines B737 Single Tail Number Comparison 2017</td>
<td>15</td>
</tr>
<tr>
<td>Figure 3.10</td>
<td>Southwest Airlines B737 Single Tail Number Comparison 2014</td>
<td>15</td>
</tr>
<tr>
<td>Figure 3.11</td>
<td>Southwest Airlines B737 Single Tail Number Comparison 2017</td>
<td>16</td>
</tr>
<tr>
<td>Figure 3.12</td>
<td>A319 American Airlines vs Frontier Airlines 2014</td>
<td>18</td>
</tr>
<tr>
<td>Figure 3.13</td>
<td>A319 American Airlines vs Frontier Airlines 2017</td>
<td>19</td>
</tr>
<tr>
<td>Figure 3.14</td>
<td>B737 American Airlines vs Southwest Airlines 2014</td>
<td>19</td>
</tr>
<tr>
<td>Figure 3.15</td>
<td>B737 American Airlines vs Southwest Airlines 2017</td>
<td>20</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Pycharm Library Menu</td>
<td>23</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Pycharm Console</td>
<td>23</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Hyperplane On Linearly Separable Data</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>A Neural Network</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Gradient Boosting Error Fitting</td>
<td>33</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Number of Estimators Vs. Max Features</td>
<td>35</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Number of Estimators Vs. Minimum Samples at Terminal Node</td>
<td>35</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>Max Features Vs. Minimum Samples at Terminal Node</td>
<td>36</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Preliminary Results Vs. Final Results</td>
<td>36</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>Prediction Results Vs. Baseline Results</td>
<td>37</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>Cost Vs. Epsilon Linear Kernel</td>
<td>38</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>Cost Vs. Epsilon Linear Kernel Normalized</td>
<td>38</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>Cost Vs. Epsilon RBF Kernel</td>
<td>39</td>
</tr>
<tr>
<td>Figure 4.14</td>
<td>Cost Vs. Epsilon RBF Kernel Normalized</td>
<td>39</td>
</tr>
<tr>
<td>Figure 4.15</td>
<td>Cost Vs. Gamma RBF Kernel</td>
<td>40</td>
</tr>
<tr>
<td>Figure 4.16</td>
<td>Cost Vs. Gamma RBF Kernel Normalized</td>
<td>40</td>
</tr>
<tr>
<td>Figure 4.17</td>
<td>Gamma Vs. Epsilon RBF Kernel</td>
<td>41</td>
</tr>
<tr>
<td>Figure 4.18</td>
<td>Gamma Vs. Epsilon RBF Kernel Normalized</td>
<td>41</td>
</tr>
<tr>
<td>Figure 4.19</td>
<td>Preliminary Results Vs. Final Results Linear Kernel</td>
<td>42</td>
</tr>
<tr>
<td>Figure 4.20</td>
<td>Preliminary Results Vs. Final Results RBF Kernel</td>
<td>42</td>
</tr>
<tr>
<td>Figure 4.21</td>
<td>Prediction Results Vs. Baseline Results Linear Kernel</td>
<td>43</td>
</tr>
<tr>
<td>Figure 4.22</td>
<td>Prediction Results Vs. Baseline Results RBF Kernel</td>
<td>43</td>
</tr>
<tr>
<td>Figure 4.23</td>
<td>SGD Optimizer</td>
<td>45</td>
</tr>
<tr>
<td>Figure 4.24</td>
<td>RMSProp Optimizer</td>
<td>46</td>
</tr>
<tr>
<td>Figure 4.25</td>
<td>Adagrad Optimizer</td>
<td>46</td>
</tr>
<tr>
<td>Figure 4.26</td>
<td>Adadelta Optimizer</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.27</td>
<td>Adam Optimizer</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.28</td>
<td>Adamax Optimizer</td>
<td>48</td>
</tr>
<tr>
<td>Figure 4.29</td>
<td>Nadam Optimizer</td>
<td>48</td>
</tr>
<tr>
<td>Figure 4.30</td>
<td>Preliminary Results Vs. Final Results Adam Optimizer</td>
<td>49</td>
</tr>
<tr>
<td>Figure 4.31</td>
<td>Prediction Results Vs. Baseline Results Adam Optimizer</td>
<td>49</td>
</tr>
<tr>
<td>Figure 4.32</td>
<td>Number of Estimators Vs. Learning Rate Adaptive Boosting</td>
<td>51</td>
</tr>
<tr>
<td>Figure 4.33</td>
<td>Number of Estimators Vs. Max Depth Gradient Boosting</td>
<td>51</td>
</tr>
<tr>
<td>Figure 4.34</td>
<td>Number of Estimators Vs. Learning Rate Gradient Boosting</td>
<td>52</td>
</tr>
</tbody>
</table>
I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Peng Wei for his guidance, patience and support throughout this research and the writing of this thesis. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Leifur Leifsson and Dr. Lizhi Wang. I would additionally like to thank everyone in the Intelligent Air Systems Lab for their support, for which has guided me through research. Last, but certainly not least, I would like to thank those nearest and dearest to me, my friends and family, for their endless support during a strenuous time in my life. These people include my parents, Joe and Sharon, my sister, Ruby, and my friends Anshul, Xuxi, Xiaosong, George and Nicole, in no particular order whatsoever.
NOMENCLATURE

Symbols

\[ E_{\Theta} = \text{Expectation w.r.t } \Theta_n \]
\[ \Theta_n = \text{Random Parameter} \]
\[ X_n = \text{Dataset Features} \]
\[ D_n = \text{Dataset} \]
\[ w = \text{Gradient of Separating Hyperplane} \]
\[ \varepsilon = \text{Margin of Tolerance} \]
\[ W_2 = \text{Cumulative Weights of Upstream Nodes} \]
\[ b_2 = \text{Bias of Downstream Node} \]
Abbreviations

LCC = Low-Cost Carrier
FAA = Federal Aviation Administration
NTSB = National Transportation Safety Board
RUL = Remaining Useful Life
MSA = Metropolitan Statistical Area
RMSE = Root Mean Squared Error
BTS = Bureau of Transportation Statistics
AOTP = Airline On-Time Performance
PDT = Pacific Daylight Time
MST = Mountain Standard Time
PT = Pacific Time
GBA = Gradient Boosting Algorithm
IDE = Independent Development Environment
XGBoost = Extreme Gradient Boost
RBF = Radial Basis Function
CPU = Central Processing Unit
NAG = Nesterov accelerated gradient
After the latest mechanical malfunction accidents involving Allegiant [1] and Southwest Airlines [2], a special interest was taken to investigate whether low-cost carriers (LCC) are taking an overly aggressive stance in regards to the utilization of aircraft within their respective fleets. Based on summary reports obtained from incident logs generated by the FAA (Federal Aviation Administration) and NTSB (National Transportation Safety Board), it was observed that Allegiant Airlines was almost three and a half times as likely to encounter a mid-air breakdown as legacy carriers are. On the economic front, the fallout that Southwest Airlines has faced from the Flight 1380 incident after an engine fan blade sheared may very likely have been a potential factor that led to an immediate decline in ticket reservations.

From a cost savings perspective, figures from a forecast analysis conducted by ICF International in 2015 predict a 40 percent increase in total fleet size across all airlines combined in the world between the years 2015 and 2025 [3]. With a global fleet size approaching 40,000 aircraft by the year 2025, the use of historical utilization data could play a key factor towards profit maximization in strategic forecasting for airline maintenance and fleet planning through the study and implementation of past trends; historical data could assist airlines with making more informed decisions on fleet planning and maintenance scheduling, by taking into consideration past patterns as well as seasonality effects in the planning process.

This study will look at airplane utilization of legacy carriers and LCC’s in their short-haul and medium-haul operations, and draw comparisons between each respective airline, focusing on aircraft types which are dedicated towards flying such routes. The objective is to explore the potential connection between airlines observing higher airplane utilization and higher frequency of accidents, with airlines that observe lower airplane utilization and frequency of accidents. These observations will likely serve to suggest whether LCC’s are utilizing their fleets too aggressively, and
to find common patterns and seasonality across all airlines that can be used as a general guideline towards maintenance scheduling and fleet planning; one that incorporates safe flying practices while maximizing profits simultaneously.

In addition, this study will utilize machine learning algorithms in an effort to explain the utilization patterns observed in historical data. The aim is to determine the potential of using historical utilization hours to capture the results produced by airline fleet planning models, and in addition, to test the feasibility and effectiveness of using machine learning algorithms as an accurate forecasting tool in establishing airplane utilization models. Current research in the airline industry address problems and situations ranging from forecasting future fleet demands, to forecasting passenger load factors, but with less emphasis placed on problems that could benefit from an accurate utilization model; such problems could be used in applications such as fleet planning and maintenance scheduling by the fleet planning and maintenance scheduling departments of respective airlines. This study aims to fill the void, and analyze airplane utilization via the use of historical airline data and various machine learning algorithms.
CHAPTER 1. OVERVIEW

The problem statement described will be addressed in three separate themes. The first theme will cover analyses of airplane utilization hours across three airlines based in the United States; American Airlines, Frontier Airlines, and Southwest Airlines. The second theme will cover comparisons between the legacy carrier, American Airlines, and the two low-cost carriers, Frontier and Southwest Airlines, in an attempt to find similar utilization patterns between the low-cost carriers. Furthermore, the comparisons will be analyzed for trend correlations across all three airlines, to explore the potential feasibility of using historical airplane utilization data as a general guideline for use in aircraft maintenance scheduling and fleet scheduling problems. In the final theme, a machine learning study will be conducted using the same historical airplane utilization hours data to attempt to explain the results observed from trend studies, in an effort to accurately forecast future utilization hours through the use of various prediction models, with the utilization hours predictions potentially to be used by maintenance planning and fleet planning groups within different airlines. A recommendation will be given to the model that is most accurate in the conclusion of this study. The problem statement is contained in chapter 1, and the problem statement is outlined in chapter 2. The methods used to acquire, establish, test, and analyze the results of airplane utilization trends as well as the machine learning models used will be discussed in chapters 3 and 4. Chapter 5 will present the conclusions and summarize implications drawn from the results. The various chapters are described in more detail in the following sections.

1.1 Introduction

A. Motivation

The importance of having intelligent forecasting tools for use in airplane maintenance planning and fleet planning in the global airline industry has grown steadily in the recent decades, and will
continue to grow for the foreseeable future. There are two sectors within the airline industry that especially stand out from benefiting due to the increase in global airplane demand; maintenance scheduling and fleet planning. The purpose of this study is to determine how airplane utilization may potentially impact airplane maintenance and fleet planning, and to test the viability and efficiency of using historical utilization data to build predictive models for both maintenance scheduling and fleet planning decision support.

Airplane maintenance planning and scheduling involve a complex system which constitutes the planning and co-ordination of parts availability, inspection, assembly and disassembly of parts, as well as labor [4]. Adding to an already complex system, are various levels of maintenance situations, such as A/B/C/D type checks, varying in comprehensiveness of the inspection; an ‘A’ type check refers to light checks, generally completed every 100 flight hours, a ‘B’ type check is a series of ‘A’ type checks, a ‘C’ type check is a heavy inspection generally completed every 2,000 flight hours, while a ‘D’ type check is a series of ‘C’ type checks. The public accessibility of historical flight information at the tail number level has inspired the idea of using airplane utilization hours to build a predictive model to assist with more efficient airline maintenance operations planning.

In a book titled ‘The Global Airline Industry’ written by Peter Belobaba [5], the book suggested building fleet planning models from cost and demand variables such as planned load factors and capacities, with added assumptions built into the model. Provided that airplane productivity, or airplane utilization is among the assumptions made, it became an inspiration to incorporate fleet planning decision making as a beneficial topic to research, in addition to airline maintenance resource planning. Given that only assumed and planned airplane utilization was mentioned, the use of historical airplane utilization hours to as a possible answer to fleet planning problems would be a novel approach.

B. Related Work

Previous studies done in the broader category of airplane maintenance primarily address the maintenance scheduling problem. The idea to use utilization hour information as an input on a
maintenance scheduling situation stems from an earlier study done by Bird, centered around heavy maintenance within the Qantas maintenance division [6]. In that study, Bird attempted to solve the problem using linear programming, taking into account the following inputs: airplane arrival time, configuration and elapsed time since last maintenance performed. In a separate study by Elkodwa, an approach based on heuristic methods was used to find a near-optimal solution for scheduling aircraft for maintenance [7]. This research will focus on predicting future airplane utilization hours through the use of historical utilization hours, to serve as a guideline for maintenance scheduling optimization models.

Another approach towards airplane maintenance type problems looks at predictive maintenance of certain vital components, such as engines, landing gear etc. Most research done in this scenario create predictive remaining useful life (RUL) models for individual components being studied. For example, the research paper ‘Degradation Modeling and Remaining Useful Life Prediction of Aircraft Engines Using Ensemble Learning’ utilizes various ensemble machine learning models, such as random forest and the boosting family algorithms, to create predictive RUL models for airplane engines [8]. This study will instead focus on machine learning problem formulations from a high-level perspective, making predictions for the utilization hours of individual tail numbers, rather than making predictions for component degradation on board airplanes.

Previous studies done on fleet planning related topics often look at the problem from an flight frequency and fleet size forecasting perspective [9]. Variables which might impact flight frequency and fleet size include the following: Route characteristics such as flight distances, level of passenger demand, and airport characteristics, such as number of runways, and hubs. An example where these variables are used can be found in a study by Pai, in which the study demonstrates a framework that looks at a three city-pair model, with either a point-to-point network or a hub-spoke network situation applied [10]. Similar to the methodology and approach described in the book titled ‘The Global Airline Industry’ by Peter Belobaba, the Pai study incorporates variables related to route characteristics and airport characteristics into cost and demand functions, which in turn can be
used to derive a profit function that is used to determine fleet size and type. By electing to use these variables however, the study highlights the difficulty research studies without access to airline data may face. For example, limitations in available Metropolitan Statistical Area (MSA) data, such as information regarding city pairs, restricted the study to the contiguous 48 states, and the year 2000.

This research aims to approach the airplane maintenance and fleet planning topics from a high-level machine learning perspective, with the goal of supporting the decision making confidence of existing maintenance scheduling and fleet planning optimization algorithms. This study will compare the results from supervised learning algorithms, focusing on utilization hours of the airplane as a common objective; it is believed that observed utilization hours in a given month may potentially provide insight towards an airplane’s maintenance outlook, where periods of low utilization activity might suggest maintenance downtime. One of the advantages of using airplane utilization data is that information is more readily available, and is generally non-proprietary information that is distributed publicly. Therefore, using utilization hours as a model input in this study allows for easier access to a greater pool of data available.
CHAPTER 2. PROBLEM STATEMENT

The objectives of this research is two-fold; the first objective is to find a positive correlation between substantially higher utilization hours and low-cost carriers. Comparisons will be made between flights which have taken place the same time during multiple years, with various short-haul airplane types considered, operated by both legacy and low-cost carriers. Utilization hour averages will be computed for each airline studied, and the trends will be compared between legacy carriers and low-cost carriers to test for aggressive over-utilization amongst low-cost carrier fleets. The utilization averages could serve as a recommendation for airlines in determining safe flying practices, one that avoids aggressive over-utilization of an airplane in a given month. In addition, the utilization averages might also serve as a reference for airline maintenance and fleet planning resource allocation.

The other objective of this research is to investigate the feasibility of applying machine learning models to predict airplane utilization, or flight time in a given month. Inputs taken from this scenario such as utilization hour, airplane type, and airline, will be formulated into a machine learning model that computes predicted values of monthly utilization hours, and used to compare to the actual utilization hour values to test for model accuracy. The historical values of utilization hour, and information pertaining to airline characteristics will be implemented into various regression model formulations, and then evaluated.

The regression model that is the least computationally expensive, whilst generating the lowest test root mean square error (RMSE) value, will be selected as the model of choice and recommended for the purposes of this study.
CHAPTER 3. AIRPLANE UTILIZATION AND TREND STUDY

In this chapter, the methods and results of the utilization study and comparisons between American Airlines, Frontier Airlines and Southwest Airlines will be discussed.

3.1 Method

Historical data in this research study is acquired from the Bureau of Transportation’s (BTS) on-time performance (AOTP) database [11]. The Bureau of Transportation collects data in numerous topics within the airline industry, such as flight delay information, fare related information, financial information, seating information among many others. In addition, The Bureau of Transportation also reports data in other transportation sectors such as rail, maritime and highways etc. This research will utilize the AOTP database found on BTS’ public domain. The AOTP database provides information pertaining to on-time data for flights that are voluntarily reported by airlines. Table 3.1 shows some examples of flight information, airline information, origin and destination information, delay information, among others that are provided by the AOTP database.

Utilization hours in this research study will be computed on a monthly basis. Monthly comparisons allow trends to be more easily observed. For example, utilization trends computed on a daily basis would become noisy, where finer seasonality effects may become more difficult to detect. Also, a hurdle was faced due to a lack of data, since there exists a physical limit to the amount of utilization data available over the life span of an airplane. As a result, computing utilization hours over a quarterly basis would be even more physically challenging. In addition, airplane maintenance

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Airline</th>
<th>Origin</th>
<th>Destination</th>
<th>Departure On-Time</th>
<th>Arrival On-Time</th>
<th>Cancellation Diversion</th>
<th>Flight Summaries</th>
<th>Cause of Delay</th>
<th>Diversion Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Airline Num</td>
<td>Origin ID</td>
<td>Destination ID</td>
<td>Dep. Time</td>
<td>Arr. Time</td>
<td>Cancelled Flights</td>
<td>Elapsed Air Time</td>
<td>Carrier</td>
<td>Airport</td>
</tr>
<tr>
<td>Quarter</td>
<td>Tail No</td>
<td>City</td>
<td>City</td>
<td>Taxi Time</td>
<td>Taxi Time</td>
<td>Diverted</td>
<td>Flights</td>
<td>Security</td>
<td>Div. Delay</td>
</tr>
<tr>
<td>Month</td>
<td></td>
<td>State</td>
<td>State</td>
<td>Delay</td>
<td>Delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day</td>
<td></td>
<td></td>
<td></td>
<td>Wheels Off</td>
<td>Wheels On</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: A Summary of AOTP Database Fields
scheduling and fleet planning make forecasts on a monthly basis. Thus, computing utilization hours monthly would be more efficient and economical for prediction purposes.

In order to compute utilization hours from historical data, the following inputs were selected from the AOTP database during the data mining phase:

- Year
- Month
- Day of month
- Carrier ID
- Tail number
- Flight number
- Originating airport ID
- Destination airport ID
- Departure time
- Arrival time
- Actual elapsed time

Raw data from BTS’ database is often messy; one key challenge encountered during the data mining phase was identifying erroneous data within large datasets. For example, a small number of flights have displayed arrival time and dates that precede departure time and dates. Figure 3.1 shows a case of this error in data where a flight operated by American Airlines tail number N783AA flew two trips simultaneously, both from Los Angeles, California to New York, New York; one flight arriving in New York at 8:28 A.M. with the other flight departing New York at 6:52 A.M.
Therefore, the departure time, arrival time, and day of month information were selected as inputs when acquiring data as a method to filter out data with invalid flight information. Another example of erroneous data include flights which contain only partial information; instances where airline and airplane information may be present, but flight information may be missing. Figure 3.2 shows a case of this type of error where carrier, airplane, departure and arrival information for two trips operated by Delta Airlines are omitted from their respective fields in the raw dataset.

In addition, certain instances of data pertaining to local time information present their own set of problems. As an example, the time in Arizona follows its own unique structure; when daylight saving is not active, Arizona would follow Mountain Standard Time (MST). However, when daylight saving is active, Arizona would follow Pacific Daylight Time (PDT) [12]. Therefore, in order to correctly compute the difference between departure and arrival times, it was important to correctly identify the time zones of all airports involved in the dataset beforehand. In figure 3.3 below, the confusion caused by the unorthodox time zones in Arizona is highlighted. In one case, a flight operated by American Airlines flying from Santa Ana, California to Phoenix, Arizona, took 76 minutes to complete based on elapsed flying time information. A quick computation of the amount of time that has passed between the origin and destination would also arrive at 76 minutes, if the time zone observed by Phoenix is Pacific Daylight Time. In a separate case, a flight also operated by American Airlines flying from Phoenix to Sacramento, California took 114 minutes.
to complete based on elapsed flying time information. Using the same calculation as in the case with the other American Airlines flight, the result also arrives at 114 minutes having elapsed upon arrival, but only when Phoenix is observing Mountain Standard Time. The time zones in Santa Ana and Sacramento both observe Pacific Time (PT).

![Figure 3.3: Time Zone Problem in Raw Data Example](image)

After processing data based on the inputs mentioned earlier in this paper, it is necessary to find a solution to determine the airplane type associated with each tail number. Because the AOTP data obtained from BTS does not offer information regarding the type of airplane flown, other resources would be needed to correlate flight number and time of flight with the type of airplane flown; open source databases Airfleets and Planespotters were used to address this shortcoming, containing information directly linking tail numbers with their respective airplane types [13, 14]. However, the databases are incomplete, and thus, require an additional method to identify the remaining tail numbers. Flightaware is the third resource used to locate those remaining tail numbers with a corresponding airplane type [15]. Unlike the Airfleets and Planespotters databases, Flightaware does not list airline fleets by individual aircraft, and as result, does not feature a way to directly connect a specific tail number with its airplane type. Instead, searches have to be made manually by airline, flight number, and date flown, and then traced to the matching flight number, departure and arrival times and airports of the data compiled from the AOTP database. Once the correct date of the flight has been identified, Flightaware displays the airplane information of the flight number associated with the time and date of flight provided. This process has to be repeated for each tail number found in the compiled raw data, and proved to be lengthiest portion of the data mining phase in this research.
Flight time information found in the compiled data taken from the AOTP database is given in minutes. To convert this to hours, such that utilization hours can be correlated, the following conversion is used:

\[
Utilization\ Hour = \frac{\sum Flight\ Time}{60} \tag{3.1}
\]

Where flight time refers to the total accumulated flight time of a specific tail number belonging to a certain airplane type, flown by a specific airline, between take-off and landing in a given month. The days in month refers to the number of days in the given month that the utilization hour is being computed for.

Studies will be done on three tail numbers for each of the following three airlines: American Airlines, Southwest Airlines, and Frontier Airlines. Each tail number will be compared to the average utilization hour of the same airplane type flown by the same airline. The average utilization hours will then be compared between American Airlines versus Southwest Airlines, and American Airlines versus Frontier Airlines. In order to compute the average utilization hour of a certain plane type and airline, the conversion formula had to be adapted slightly into the following:

\[
Utilization\ Hour = \frac{\sum Flight\ Time}{60 \times fleet\ size} \tag{3.2}
\]

Where in this case, flight time refers to the total accumulated flight time of every single tail number belonging to a specified family of airplane, and fleet size referring to the number of airplanes of a specified family being operated in the fleet flown by a certain airline.

In this study, the single aisle airplanes, Boeing 737-800 and Airbus A319 are chosen for evaluation purposes. The two types of airplanes generally fly on short haul trips, such as distances between Dallas, TX, and Houston, TX. Such trips generally last less than three hours. Because the low-cost carriers, Southwest Airlines and Frontier Airlines, only operate one of the two family of airplanes, it became necessary to incorporate both Boeing and Airbus single aisle offerings in order to make a legacy carrier versus low-cost carrier comparison possible.
3.2 Results

In this section, the results for the comparisons of utilization hour trends from the three airlines are presented. The first subsection will contain the results for each tail number and compared to the average trend amongst all airplanes of the same class, flown by the same airline. The second subsection will compare the average utilization hour trends between legacy and low-cost carriers.

3.2.1 Utilization Hour Comparison of Tail Numbers

![American Airlines A319 Single Tail Number Comparison 2014](image)

Figure 3.4: American Airlines A319 Single Tail Number Comparison 2014
Figure 3.5: American Airlines A319 Single Tail Number Comparison 2017

Figure 3.6: Frontier Airlines A319 Single Tail Number Comparison 2014
Figures 3.1 through 3.4 cover utilization trends amongst airplanes of the A319 type, which is a derivative of the Airbus A320 family. A319 airplanes operated by American Airlines will be compared side by side with those operated by Frontier Airlines, for the years 2014 and 2017.

The following are the observations from the airplane utilization comparisons between American Airlines and Frontier Airlines in 2014 and 2017:

1. Frontier Airlines' average A319 airplane type utilization is around 50% higher than that of American Airlines' in 2014, and close to 75% higher in 2017. The average utilization of Frontier Airlines' A319 fleet averages at around 350 utilization hours each month in 2017.

2. Tail numbers belonging to the same airline generally followed a similar trend in 2014, with a few exceptions in certain months. These trends are observed in comparisons between utilization trends of individual tail numbers and the average utilization of the airplane type. This trend is also true for Frontier Airlines in 2017. However, A319 airplanes in American Airlines' fleet observed more variation the same year; individual tail numbers were observed to fluctuate substantially month to month, while the average utilization observed little change. This behaviour might
indicate that 2017 may have been a maintenance intensive year where many airplanes purchased during a short window of time were undergoing heavy maintenance inspections due around the same time frame.

3. Results show that utilization trends show few signs of repetition when compared from one year to another. For instance, the results indicate that Frontier Airlines average utilization hours increased each month as the year progressed in 2014, while the trend was in a slight decline in 2017. In the case of American Airlines, results show that the utilization hour trend is in sharp decline in 2014, while the trend is much more flat in 2017. Perhaps the unpredictable nature of utilization trends from one year to another could be attributed to changes in external market forces, where the market share of a certain airline in a certain flight segment changes in an unpredictable pattern, leading to airplane utilization trends to follow a similar response. However, trends do generally tend to suggest that the period between June to August are the months where airlines would expect to see peak demand.

![Figure 3.8: American Airlines B737 Single Tail Number Comparison 2014](image)
Figure 3.9: American Airlines B737 Single Tail Number Comparison 2017

Figure 3.10: Southwest Airlines B737 Single Tail Number Comparison 2014
Figures 3.8 through 3.11 covers utilization trends amongst airplanes of the Boeing 737-800 type, which is a derivative of the Boeing 737 family. Boeing 737-800 airplanes operated by American Airlines will be compared side by side with those operated by Southwest Airlines, for the years 2014 and 2017.

The following are the observations from the airplane utilization comparisons between American Airlines and Southwest Airlines in 2014 and 2017:

1. Southwest Airlines’ average Boeing 737-800 airplane type utilization is around 42% higher than that of American Airlines’ in 2014, and close to 44% higher in 2017, which translates to around 100 more hours utilized each month. The average utilization of Southwest Airlines’ Boeing 737-800 fleet averages at around 340 utilization hours each month in 2014 and 325 utilization hours each month in 2017. In addition, comparisons between American Airlines’ Boeing 737-800 and A319 utilization show that American Airlines utilized their newer Boeing 737-800 fleet nearly 20% more in 2014, and nearly 12.5% more in 2017.
2. Tail numbers belonging to Southwest Airlines followed the average utilization hour trend much more closely than the tail numbers belonging to American Airlines. This is true for both the years 2014 and 2017. Similar to the average utilization trend observed in American Airlines’ A319 fleet, the average utilization of American Airlines’ Boeing 737-800 fleet also proved to be much more consistent month to month in comparison to Southwest Airlines’ fleet; fluctuations ranged as much as 50 hours on average as opposed to close to 85 hours respectively for the two airlines.

3. Similar to the A319 comparison between American Airlines and Frontier Airlines, the utilization comparisons between airplanes of the Boeing 737-800 type operated by American Airlines and Southwest Airlines also show few signs of trends of repetitive nature when compared between the years 2014 and 2017. However, not unlike the seasonality trends observed in the A319 comparisons, the period between the months of June to August also observed high levels of demand.
3.2.2 Average Utilization Hour Comparison Between Legacy Carriers and Low-Cost Carriers

Figure 3.12: A319 American Airlines vs Frontier Airlines 2014
Figure 3.13: A319 American Airlines vs Frontier Airlines 2017

Figure 3.14: B737 American Airlines vs Southwest Airlines 2014
Figures 3.12 through 3.15 show the comparisons in utilization trends between legacy carriers and low-cost carriers, between airplanes of the Boeing 737-800 type and A319 type operated by American Airlines, Frontier Airlines and Southwest Airlines, for the years 2014 and 2017.

The following are the observations from the the average airplane utilization comparisons in 2014 and 2017:

1. Utilization trends from both legacy carriers and low-cost carriers tend to share a similar pattern when compared between the same year. This observation could be used to justify the earlier hypothesis that the unpredictable nature of utilization trends from one year to another was due to changes in external market forces, as witnessed in airlines following similar utilization patterns in the same year, possibly driven by the same external market forces.

2. Average low-cost carrier utilization hours can reach 350 hours per month rather frequently, while legacy carriers average closer to 220 hours per month. In the case of an older A319 fleet operated by American Airlines, utilization hours reach as low as 140 hours per month in November and December 2014, and average at around 175 hours on a monthly basis. On the contrary, the
newer Boeing 737-800 fleet also operated by American Airlines observes average utilization hours at around 245 hours on a monthly basis. This result highlights the conservative approach taken by legacy carriers when operating a fleet composed of aging aircraft.

3. The month of February in 2014 is the only instance where utilization hours of a low-cost airline eclipsed that of an legacy carrier. Given that all other periods observed in the years 2014 and 2017 show that low-cost carriers flew no less than 250 hours in any given month, it may be a possibility that the months between January and March of 2014 was a period of heavy maintenance inspections for Frontier Airlines; a heavy inspection involving Frontier Airlines’ small fleet of airplanes would result in a significant reduction in utilization hours as indicated.
CHAPTER 4. AIRPLANE UTILIZATION PREDICTION VIA MACHINE LEARNING MODELS

In this chapter, the methods and results of the various machine learning models used to predict airplane utilization hours will be discussed.

4.1 Method

Four machine learning algorithms were compared and tested for utilization hour prediction accuracy. Of the four algorithms used, two were ensemble machine learning algorithms; random forest and the boosting family of algorithms. The other two models used were support vector regression, and neural networks. The Scikit-Learn Python library is used to import the models for random forest, support vector machines, and most of the algorithms for the boosting family [16]. Extreme gradient boosting, or ‘XGBoost,’ which is an efficient implementation of the gradient boosting algorithm (GBA) [17], is the only algorithm from the boosting family of algorithms to utilize its own library [18]. In addition, the Python Keras package was utilized for neural networks. Models were initially written in Python 2, since the data mining scripts which were first written during the onset of this research study were written with Python 2. However, with the inclusion of XGBoost used in boosting algorithm comparison tests, the machine learning models used were converted to the Python 3 syntax, since XGBoost’s library is only designed to be implemented in a Python 3 environment. This research used ‘Pycharm’ as the primary independent development environment (IDE), due to its ease of use for debugging purposes, and user friendliness with library installation and file mapping. Figure 4.1 below shows the library installation menu in Pycharm:
Figure 4.1: Pycharm Library Menu

Figure 4.2 below shows the console displaying an error when debugging in Pycharm:

Figure 4.2: Pycharm Console
Training data is organized by month as panel data; panel data is data that is derived from a number of observations over time, by month, in the case of this research. The data however, will not necessarily be in order once a random partial split is made in order to divide the dataset into training and testing portions, since there is not a sufficient amount of data to have independent training and test datasets. Training data initially utilized elapsed flight time for each tail number belonging to a common airplane type and airline as features, in addition to the elapsed flight time of the previous two months for each tail number. Furthermore, the average elapsed flight time for all tail numbers belonging to the same airplane type and airline was also included as a feature for each month of data. Once the dataset has been imported and the features and labels within the dataset have been established, an exhaustive grid search was conducted in order to fine-tune optimal hyperparameters best suited for each machine learning model. Each hyperparameter would undergo a 4-fold cross-validation process to test for data overfitting using root-mean-squre error (RMSE) as a comparison between training error and test error.

4.1.1 K-Fold Cross-Validation

Below is the algorithm for generalized k-fold cross-validation [19]:

1. Repeat the following process N (Number of features and label) times.

   (a) Divide the dataset D pseudo-randomly into V folds

   (b) For I from 1 to V:

      i. Define set L as the dataset D without the I-th fold

      ii. Define set T as the I-th fold of the dataset D

      iii. For k from 1 to K:

          A. Build a statistical model $f^k = f(L; \alpha^k)$

          B. Apply $f^k$ on T and store the predictions
(c) For each $\alpha$ value calculate the goodness of fit (loss) for all elements in $D$.

2. For each $\alpha$ value calculate the mean of the $N_{\text{exp}}$ calculations of losses.

3. Let $\alpha'$ be the $\alpha$ value for which the average loss is minimal. If there are multiple $\alpha$ values for which the average loss is minimal, then $\alpha'$ is the one with the lowest model complexity.

4. Select $\alpha'$ as the optimal cross-validation choice for tuning parameter and select statistical model $f' = f(D; \alpha')$ as the optimal cross-validation model.

The K-Fold cross-validation method is used for hyperparameter tuning and model selection in this study. The K-fold cross-validation algorithm splits the observations in the training dataset into $K$ sections; of the $K$ sections, $K-1$ sections will be held for the validation dataset, and one section will be held for testing against the validation dataset. This process is repeated $K$ iterations. Every machine learning model, as well as each hyperparameter is then trained on each validation dataset and the loss function, in this case being RMSE, is computed as the cross-validation error of each iteration and is then averaged between all $K$ iterations. Finally, each machine learning model and associated hyperparameters are then tested on a separate test dataset. The model and hyperparameter with the lowest test RMSE, as well the closest gap between cross-validation error and test error, will be selected as the model and hyperparameters of choice.

4.1.2 Preliminary Results

After initial runs of the original training datasets produced high RMSE test error values, the original training dataset was modified to stack data pertaining to each tail number longitudinally, instead of treating those pieces of data as features. In addition, the initial dataset was also modified to include the following as features:
• Year
• Month
• Airplane Type
• Carrier
• Cumulative Cycle of Tail Number Since New
• Cumulative Elapsed Flight Time of Tail Number Since New
• Total Elapsed Flight Time of Previous Month
• Total Elapsed Flight Time of Two Months Prior
• Total Average Elapsed Flight Time of Previous Month
• Total Average Elapsed Flight Time of Two Months Prior
• Total Distance of Previous Month
• Total Distance of Two Months Prior
• Total Average Distance of Previous Month
• Total Average Distance of Two Months Prior

Most of the information offered in BTS’ AOTP database were used as features in this study. The random forest’s feature importance impurity based ranking method also provides additional insight into the most important features to consider when conducting feature engineering with the training dataset. Any feature with an importance of more than an arbitrary value of 10% is deemed as important in this research. Features which are not continuous data types, such as airplane type, or month of year, use the one-hot encoding process to convert data into interpretable binary form.
4.1.3 Random Forest Model

A random forest model was selected as a model to test due to its robustness; random forest models do not require any scaling in data, and can perform both classification tasks as well as regression tasks. Data scaling is a part of data preprocessing. In this research, scaling refers to normalizing data from 0-1. Random forests are an ensemble of decision trees, trained using the bagging method. At each ‘split’ in the tree, a random subset of features are taken into consideration by the algorithm. The tree is continuously split until a stop criteria is met, such as the minimum samples in the leaf node being met, or the number of layers of the tree being fulfilled. Each decision tree forms a base model, and the final result is an average of all the base decision tree models. Another useful feature of random forests is the ability to perform implicit feature selection and provide indication of feature importance. For this research, the following hyperparameters will be tuned based on the paper by Probst et al. (2018) [20]:

- Number of Estimators
- Max Features for Best Split
- Minimum Number of Samples at Terminal Node

The aggregated regression estimate of a random forest model is given below [21]:

\[
\bar{r}_n(X, D_n) = E_\Theta [r_n(X, \Theta, D_n)] \tag{4.1}
\]

Where \(E_\Theta\) is the expectation with respect to \(\Theta_n\), which defines the random attributes of the model such as feature selection and data point selection, features \(X_n\) and dataset \(D_n\).
4.1.4 Support Vector Regression Model

The second machine learning algorithm implemented was the support vector regression algorithm. The support vector regression algorithm was selected mainly due to the fact that there are relatively few hyperparameter choices to choose from; this makes it less computationally expensive when executing a exhaustive grid search across all hyperparameter values. The objective of support vector regression is to identify a hyperplane of N-dimensional space that maximizes the margin distance between data of different groups as to provide some reinforcement such that future data points can be grouped more accurately. There are three main hyperparameters to tune in a support vector regression. The cost parameter defines how much penalty should be given to data points lying outside the margin of tolerance, based on their distance from the hyperplane. The gamma parameter defines how to separate datapoints when forming a hyperplane, by either giving stronger bias or stronger variance to the dataset. The epsilon parameter defines how wide to set the margin of tolerance, limiting the number of acceptable datapoints. Figure 4.3 shows an example of a hyperplane created on a few data points [22].

In this study, the following hyperparameters were tested [23]:

- Cost (C) Parameter- Denotes the penalty applied data points lying outside of margin of tolerance
- Gamma (γ) Parameter- Denotes variance and bias trade-off in model
- Epsilon (ε) Parameter- Denotes margin of tolerance where no penalty is given to errors
Where the sample data points are constrained by margin of tolerance, $\epsilon$: $y_i - wx_i - b \leq \epsilon$ and $wx_i + b - y_i \leq \epsilon$

In addition, another benefit of using the support vector regression algorithm is the fact that it can easily convert linearly separable data into a non-linear model using kernels, which measure the similarity between two data points by mapping inputs from an n-dimensional space into an m-dimensional space. Two different types of kernel functions were used in this study; linear and radial basis function (RBF). A linear kernel function computes the transformation by taking a dot product between the two inputs. An RBF kernel function computes the transformation by taking the exponent of the squared absolute difference of the two inputs.
4.1.5 Neural Network Model

The third machine learning algorithm implemented were neural networks. The name ‘neural networks’ is inspired by the architecture of the human brain, where layers of nodes are established to mimic the function of neurons. Figure 4.4 depicts an example of a neural network:

The neural network shown in figure 4.4 denotes a network with one hidden layer. Hidden layers perform computations based on the sum of the weighted inputs of nodes in the upstream layer, where each hidden node then applies a bias which feeds into an activation function, thereby deciding whether or not a node is ‘fired’ or not. This process is repeated until a predicted value is generated in the final output layer. The predicted value is compared against the actual observed value, and
the weights and biases of each node are adjusted via backpropagation as the model is being trained. Equation 4.2 below shows the output function of a node and synapse [24]:

$$Y = W_2x + b_2$$  \hspace{1cm} (4.2)

Where $Y$ refers to output of the node, $W_2$ referring to the cumulative weights of the upstream nodes feeding to the downstream node, $x$ referring to the cumulative inputs of the upstream nodes, and $b_2$, referring to the bias applied to the downstream node.

The activation functions ReLU (Rectified Linear Unit) was used because it is less computationally expensive than tanh and sigmoid functions, due to simpler mathematical operations. Equation 4.3 shows the ReLU function [25]:

$$R(z) = \begin{cases} 
z & z > 0 \\
0 & z \leq 0 
\end{cases}$$  \hspace{1cm} (4.3)

Where output is equal to input if input is larger than zero, and output is equal to 0, if input is less than zero.

In addition, the following hyperparameters were tuned:

- Optimization Function
- Number of Hidden Layers
- Learning Rate

The optimization functions used in this research study are:

- SGD (Stochastic gradient descent)-Weights are updated by computing the gradient function with respect to the error function using one data sample per iteration
- RMSProp (Root mean square propagation)-The gradient function is normalized using a moving average of squared gradients
• Adagrad (Adaptive gradient)-Learning rates are varied for each weight at each time step based on squares of past gradients which have been computed for the weight.

• Adadelta-Reduces the problem of decaying learning rates as seen in the adagrad optimizer. Instead, the adadelta optimizer keeps a running average of the squares of previous gradients.

• Adam (Adaptive moment estimation)-Adam is similar to the RMSProp and adadelta optimizers, where it stores the running average of the squares of previous gradients. In addition, the adam optimizer also stores the running average of previous gradients.

• Adamax-The current square gradient is inversely proportionally scaled against the running average of squares of previous gradients.

• Nadam (Nesterov-accelerated Adaptive Moment Estimation)-The Nadam algorithm combines the adam algorithm with the Nesterov accelerated gradient (NAG) algorithm.

4.1.6 Boosting Models

From the boosting family of algorithms, three types were compared: Adaptive boosting (AdaBoost), gradient boosting, and extreme gradient boosting (XGBoost). Similar to the random forest algorithm the boosting algorithm is an ensemble method which trains a ‘weak’ or ‘base’ learning algorithm repeatedly, each time with a different subset of training examples, converting weak learners into strong learners. Each time the base learning algorithm is called, a new weak prediction rule is generated. After multiple iterations, the boosting algorithm must combine the weak rules into a single prediction rule more accurate than any one of the weak rules [26]. Unlike the random forest algorithm which calculates an average of the base decision tree models, the boosting algorithm computes a weighted average of base decision tree models.

In adaptive boosting, ‘weights’ of equal value are initialized in each observation of a decision tree. After the initial iteration, weak learners most often misrepresented have the most weights
placed on them; this places heavier emphasis on learning from the hardest examples for the base learner. Decision trees from each iteration are continuously grown and improved from the previous iteration, and the final ensemble model will become the weighted sum of the predictions made by the previous tree models. Gradient boosting, on the other hand, fits to the residual errors made in the previous iteration. Figure 4.5 shows the error fitting above [27].

The plot in green in Figure 4.5 shows the original function being approximated. Tree one is defined by the first base model, which is fitted to the original function. The subsequent trees are then fitted to each prior base model, where then final approximation would become the summation of all the individual base models.

Lastly, the XGBoost algorithm is an implementation of the gradient boosting algorithm, but is optimized for speed and performance by utilizing the following hardware implementations:

- Parallelization of tree construction using all CPU cores during training
- Cache Optimization of data structures and algorithm
- Out-of-Core Computing for very large datasets that don’t fit into memory

In this research, the following hyperparameters for the boosting algorithms will be tuned and tested for accuracy and performance:
• Max Features for Best Split

• Number of Estimators

• Learning Rate

4.2 Results

In this section, the results for the predictions made by the four machine learning algorithms are presented. The first subsection will contain the results for the random forest algorithm, the second subsection will contain the results for the support vector regression algorithm, the third neural networks, and lastly, the results for the boosting algorithms will be presented. The optimal hyperparameters for each model is chosen based on the cross-validation RMSE results as well as the gap observed between cross-validation RMSE and test RMSE values; the best result would exhibit the lowest cross-validation RMSE while the test RMSE result would be as close as possible to the cross-validation RMSE value.
4.2.1 Random Forest Algorithm Prediction Results

Figure 4.6: Number of Estimators Vs. Max Features

Figure 4.7: Number of Estimators Vs. Minimum Samples at Terminal Node
Figure 4.8: Max Features Vs. Minimum Samples at Terminal Node

Figure 4.9: Preliminary Results Vs. Final Results
1. From the cross-validation results on the validation dataset, it is shown that the best results are obtained when the number of estimators is set at 1,000, with the maximum features at each split set at the square root of the number of features in the dataset, and the minimum samples at the leaf node set at 5. The cross-validation RMSE observed when the model is set to these hyperparameters is slightly under 49 hours.

2. Results on test data show that the training dataset with revised features bested the accuracy of both the dataset with the original features used, as well as the accuracy from a two-months rolling average. In regards to the performance improvement between the training dataset with the original features and the training dataset with the modified features, percentage improvements of an average of 125% can be observed. In regards to the improvement in accuracy between the predictions made by the random forest model and the predictions made by the rolling average model, an average percentage improvement of 17% can be observed.
4.2.2 Support Vector Regression Algorithm Prediction Results

Figure 4.11: Cost Vs. Epsilon Linear Kernel

Figure 4.12: Cost Vs. Epsilon Linear Kernel Normalized
Figure 4.13: Cost Vs. Epsilon RBF Kernel

Figure 4.14: Cost Vs. Epsilon RBF Kernel Normalized
Figure 4.15: Cost Vs. Gamma RBF Kernel

Figure 4.16: Cost Vs. Gamma RBF Kernel Normalized
Figure 4.17: Gamma Vs. Epsilon RBF Kernel

Figure 4.18: Gamma Vs. Epsilon RBF Kernel Normalized
Figure 4.19: Preliminary Results Vs. Final Results Linear Kernel

Figure 4.20: Preliminary Results Vs. Final Results RBF Kernel
Figure 4.21: Prediction Results Vs. Baseline Results Linear Kernel

Figure 4.22: Prediction Results Vs. Baseline Results RBF Kernel
1. From the cross-validation results on the validation dataset using the linear kernel, it is shown that the best results are obtained when the cost parameter is set at 0.1, with epsilon parameter set at 0.5. However, given the slow computation times and high validation and test errors produced when using the modified training dataset, the training dataset was further improved to scale the data from 0 to 1. The results seen show substantial improvement, with each computation completing within 20 seconds, down from the highest of 1,541 seconds observed in a case using non-normalized data. In addition, validation results are more stable across the range of hyperparameters used, with the range between the highest and lowest RMSE’s observed at 88 hours when using the normalized training dataset, down from the 12,345 hours observed when using non-normalized training data. The best results are produced when hyperparameters cost and epsilon were set at 0.05 respectively for both. This normalized error translates to under 49 utilization hours on average in a given month.

2. From the cross-validation results on the validation dataset using the RBF kernel, it is shown that the best results are obtained when the cost parameter is set at 100, the epsilon parameter set at 0.5 and the gamma parameter set at 0.1, producing an RMSE value of 71 hours. In addition, a normalized validation dataset was also compared. The best results observed in the normalized validation dataset are produced when hyperparameters cost is set at 1, when the epsilon parameter is set at 0.05 and when the gamma parameter is set at 0.1. This normalized error translates to under 48 utilization hours on average in a given month.

3. Results on test data show that the training dataset with revised features bested the accuracy of both the dataset with the original features used, as well as the accuracy from a two-months rolling average in all but two cases when using the RBF kernel. In regards to the performance improvement between the training dataset with the original features and the training dataset with the modified features, percentage improvements of an average of 60% can be observed when using a linear kernel, and an improvement of just over 57% can be observed when using an RBF kernel. In regards to the improvement in accuracy between the predictions made by the linear kernel
support vector regression model and the predictions made by the rolling average model, an average percentage improvement of 15% can be observed. Lastly, an average percentage improvement of 4% is observed when comparing the prediction results from the RBF kernel model and the two-month rolling average model.

4.2.3 Neural Network Prediction Results

![Figure 4.23: SGD Optimizer](image)

Figure 4.23: SGD Optimizer
Figure 4.24: RMSProp Optimizer

Figure 4.25: Adagrad Optimizer
Figure 4.26: Adadelta Optimizer

Figure 4.27: Adam Optimizer
Figure 4.28: Adamax Optimizer

Figure 4.29: Nadam Optimizer
Figure 4.30: Preliminary Results Vs. Final Results Adam Optimizer

Figure 4.31: Prediction Results Vs. Baseline Results Adam Optimizer
1. From the cross-validation results on the validation dataset, it is shown that the best results are yielded when using an Adam optimizer, but only when the learning rate is set at 0.1 or lower. Similar to the scaling issues seen when training the support vector regression model on the modified training dataset, the neural network model also requires data to be normalized in order to converge to a result. When using an Adam optimizer with the learning rate set to 0.001, and having a model structure of 32, 64, 64, 64, 64, 1 (input, hidden, hidden, hidden, hidden, output) nodes, the resulting cross-validation RMSE is slightly over 50 hours. Only two test iterations were performed due to the numerous optimizers which had to be run, making the computations more time consuming.

2. Results on test data show that the training dataset with modified features bested the accuracy of both the dataset with the original features used, as well as the accuracy from a two-months rolling average. In regards to the performance improvement between the training dataset with the original features and the training dataset with the modified features, percentage improvements of an average of 60% can be observed. In regards to the improvement in accuracy between the predictions made by the neural network model and the predictions made by the rolling average model, an average percentage improvement of 16% can be observed.
4.2.4 Boosting Algorithms Prediction Results

Figure 4.32: Number of Estimators Vs. Learning Rate Adaptive Boosting

Figure 4.33: Number of Estimators Vs. Max Depth Gradient Boosting
Figure 4.34: Number of Estimators Vs. Learning Rate Gradient Boosting

Figure 4.35: Max Depth Vs. Learning Rate Gradient Boosting
Figure 4.36: Number of Estimators Vs. Learning Rate XGBoosting

Figure 4.37: Max Depth Vs. Learning Rate XGBoosting
Figure 4.38: Number of Estimators Vs. Max Depth XGBoosting

Figure 4.39: Preliminary Results Vs. Final Results Adaptive Boosting
Figure 4.40: Preliminary Results Vs. Final Results Gradient Boosting

Figure 4.41: Preliminary Results Vs. Final Results XGBoost
Figure 4.42: Prediction Results Vs. Baseline Results Adaptive Boosting

Figure 4.43: Prediction Results Vs. Baseline Results Gradient Boosting
1. From the cross-validation results on the validation dataset using the adaptive boosting algorithm, the best results are obtained when the number of estimators is set at 100, and the learning rate set at 0.01. This RMSE error produced when using these hyperparameters is slightly over 50 utilization hours on average in a given month.

2. From the cross-validation results on the validation dataset using the gradient boosting algorithm, the best results are obtained when the number of estimators is set at 100, the max depth of decision tree from root to leaf node at 3, and the learning rate set at 0.1. This RMSE error produced when using these hyperparameters is just under 49 utilization hours on average in a given month.

3. From the cross-validation results on the validation dataset using the extreme boosting algorithm, the best results are obtained when the number of estimators is set at 100, the max depth of decision tree from root to leaf node at 3, and the learning rate set at 0.1. This RMSE error produced when using these hyperparameters is also just under 49 utilization hours on average in a given month. The difference in RMSE results between the gradient boosting algorithm and the ex-
Table 4.1: Machine Learning Model Predictions

<table>
<thead>
<tr>
<th></th>
<th>Random Forest</th>
<th>SVR Linear</th>
<th>SVR RBF</th>
<th>Neural Network</th>
<th>AdaBoost</th>
<th>Gradient Boost</th>
<th>XGBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV RMSE</td>
<td>49</td>
<td>49</td>
<td>48</td>
<td>50</td>
<td>49</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Test RMSE</td>
<td>47.2</td>
<td>51.5</td>
<td>54.8</td>
<td>51.3</td>
<td>54.3</td>
<td>49.6</td>
<td>48.8</td>
</tr>
</tbody>
</table>

4. Results on test data show that the training dataset with revised features bested the accuracy of both the dataset with the original features used, as well as the accuracy from a two-months rolling average in all three cases. In regards to the performance improvement between the training dataset with the original features and the training dataset with the modified features, percentage improvements of an average of 102% can be observed when using the adaptive boost algorithm, an improvement of just under 149% can be observed when using the gradient boosting algorithm and an improvement of 181.5% can be observed when using the extreme gradient boosting algorithm. In regards to the improvement in accuracy between the predictions made by the adaptive boosting algorithm and the predictions made by the rolling average model, an average percentage improvement of 8.5% can be observed. In regards to the improvement in accuracy between the predictions made by the gradient boosting algorithm and the predictions made by the rolling average model, an average percentage improvement of 16.8% can be observed. Lastly, an average percentage improvement of 17.9% is observed when comparing the predictions made by the XGBoost algorithm and the two-month rolling average model.

4.2.5 Summary of Machine Learning Model Predictions

Table 4.1 above shows a summary of the cross-validation and test RMSE results for each machine learning model tested. The top row represents the cross-validation RMSE, while the bottom row represents the test RMSE:

From table 4.1, the random forest model is seen to produce the lowest test error out of the four different machine learning models used. However, the model of choice in this research is the gradient boost algorithm, given that it was the most stable with an average test RMSE standard
deviation of 0.3 hours. This result is achieved with the number of estimators set at 100, the maximum depth of the tree set at three layers, and the learning rate set at 0.01. Out of the four machine learning models used, ensemble decision tree models delivered the lowest cross-validation RMSE errors, as well as the most stable results, due to taking an average over multiple base decision tree models, filtering out some of the noise observed in the small training dataset.
CHAPTER 5. CONCLUSION AND FUTURE WORK

In conclusion, this research study has covered three main themes relative to airplane utilization. The first theme looked at flight time and utilization hour patterns of a few tail numbers within a fleet operated by the same airline. The second theme compared the average utilization hours of low-cost airlines with legacy carriers, and attempted to justify the potential use for airplane utilization data in airline maintenance planning and fleet planning problems, in addition to making observations through historical trend studies that low-cost carriers have a much more aggressive fleet utilization than their legacy carrier counterparts. The third theme implements machine learning algorithms on airline historical data in an attempt to make predictions for airplane utilization and further justify observed utilization trends seen in the second theme of this study.

The results obtained in the first and second themes of this study involving utilization comparisons between low-cost and legacy airlines show that both low-cost carriers Southwest Airlines and Frontier Airlines observed a substantially higher utilization than that of American Airlines; Southwest Airlines saw monthly utilization hours of 42% and 44% higher than American Airlines in 2014 and 2017 respectively, while Frontier Airlines saw monthly utilization hours of 50% and 75% higher in 2014 and 2017 respectively. In addition, it was observed that individual tail numbers generally followed the average pattern of all tail numbers flown by the same airline. In a few instances where a lot more variation was observed between individual tail numbers and the combined fleet average, it is hypothesized that the variation indicates airplane maintenance where airplanes not undergoing maintenance inspections have additional coverage to fulfill. Therefore, airplane utilization hours could potentially be used as an indicator for airline maintenance labor planning, as well as airline fleet planning. Furthermore, the results revealed in comparisons made between 2014 and 2017 trends show few signs of repetition. Thus, it is more likely that external market forces drive the airline industry trends for a given year, rather than historic utilization trends setting precedence.
for future airplane utilization trends. However, seasonality effects do tend to show between the June to August months where demand generally peaks.

The results obtained in the third theme of this study, which involves the implementation of machine learning algorithms on airline historical data used to predict airplane utilization, indicate that decision tree models performed best on panel data. Training datasets initially used to produce preliminary results were modified to stack training data belonging to multiple tail numbers into one training dataset. The results of the modified training dataset shows substantial improvement in each machine learning model used, with improvements ranging from 57% to 182%. Final prediction results were also compared to a baseline consisting of a two-month prior rolling average. Prediction accuracy improvements of the machine learning models on the rolling average results range from 4% to 18%.

This research has shown that certain utilization trends do carry significance, such as external market forces on an airline, and seasonality effects. Furthermore, this research has demonstrated that machine learning models have the potential to improve upon prediction accuracy of more simplistic models such as rolling averages. For future endeavors in this research topic, it is suggested that a time series study be conducted and compared with machine learning predictions, as a possibility for further improvement in prediction model accuracy.
BIBLIOGRAPHY


