15.089 Analytics Capstone:

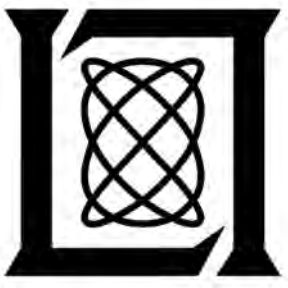
Final Report

By:

Sheamus Larkin and Michael Rieker

Faculty Advisor: Prof. Dimitris Bertsimas

MIT LL Advisors: Dr. Allison Chang, Dr. Chelsea Curran



This material is based upon work supported by the United States Transportation Command under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Transportation Command.

Table of Contents

1 INTRODUCTION 3

2 OPTIMAL ASSIGNMENT OF TANKERS TO RECEIVERS 4
2.1 Optimization Model 5
2.2 Results 7

3 TANKER FLIGHT DATA ANALYTICS 8
3.1 Datasets 9
3.1.1 Flight Data 9
3.1.2 Maintenance Data 9
3.2 Anomaly Detection 9
3.2.1 Threshold Methods 9
3.2.2 K-Nearest Neighbors 11
3.3 Classification Methods to Predict Maintenance 14
3.3.1 Data fusion 15
3.3.2 Generating the response variable 16
3.3.3 Feature engineering 17
3.3.4 Models and performance metrics 18
3.3.5 Results 19
3.3.6 Impact of predictive maintenance 23

4 CONCLUSION 24

APPENDIX A 25

1 INTRODUCTION

This report summarizes research performed for our capstone project in collaboration with MIT Lincoln Laboratory (MIT LL), which is a Federally Funded Research and Development Center (FFRDC) whose mission is to develop technology in support of national security.[1] Our project was part of the Laboratory's program sponsored by the United States Transportation Command (USTRANSCOM) – an organization within the Department of Defense (DoD) responsible for providing global transportation in service of a variety of DoD missions, including the deployment and sustainment of troops abroad, humanitarian assistance and disaster response, and presidential travel. USTRANSCOM works with its component commands – the Air Force's Air Mobility Command (AMC), the Army's Surface Deployment and Distribution Command (SDDC), and the Navy's Military Sealift Command (MSC) – to execute missions using air, land, and sea-based modes of transportation. In particular, AMC handles airlift, air refueling, and aeromedical evacuation missions, and in our project, we focused on analytics to support air refueling operations.

Figure 1. KC-135 refueling an F-15.

Air refueling refers to the in-flight transfer of fuel from one aircraft (referred to as the tanker) to another aircraft (referred to as the receiver), as shown in Figure 1. The Air Force uses air refueling to extend the range of the receiver aircraft, such as fighters and bombers, so that they can accomplish their missions. AMC is generally interested in developing capabilities for analyzing and improving tanker missions. Towards this goal, our contributions were two-fold:

1. First, we developed a **mixed-integer optimization** model to assign tankers to receivers. Current practice for planning air refueling operations is to schedule the receivers first to be able to accomplish their missions, and then to schedule the tankers according to the receiver schedule. One of MIT LL's research areas is to experiment with different algorithms to characterize the optimality that could be gained in certain scenarios by scheduling the receivers and tankers simultaneously. We demonstrated that it is possible to capture fuel transfer dynamics using mixed-integer optimization, which could be a module in a larger scheduling algorithm.

[1] https://www.ll.mit.edu/

2. Second, we applied **flight data analytics** towards the problem of predictive maintenance, which is currently an important strategic area for the Air Force. Improving the ability to foresee aircraft maintenance issues would help prevent unscheduled maintenance and thereby increase availability and readiness of the fleet. MIT LL is developing an approach to enhance maintenance predictions by accounting for how each aircraft was flown with data captured by flight data recorders. In particular, we had access to data for thousands of KC-135 tanker missions. Our tasks included exploring the flight data using anomaly detection methods, and fusing the data with maintenance data to discover features that are predictive of maintenance issues.

We explain the mixed-integer optimization and flight data analytics tasks in Sections 2 and 3 respectively. We summarize and conclude in Section 4.

2 OPTIMAL ASSIGNMENT OF TANKERS TO RECEIVERS

As stated in the introduction, air refueling operations are currently scheduled in a sequential manner. The missions of the receivers (e.g., fighters, bombers) take priority, so they are scheduled first. Then the receiver schedules are used as input, and the tankers are scheduled to meet up with the receivers at appropriate times and locations in order to ensure they have sufficient fuel to conduct their missions and return home. Intuitively, the sequential method may in some cases result in suboptimal solutions with respect to total fuel burn.

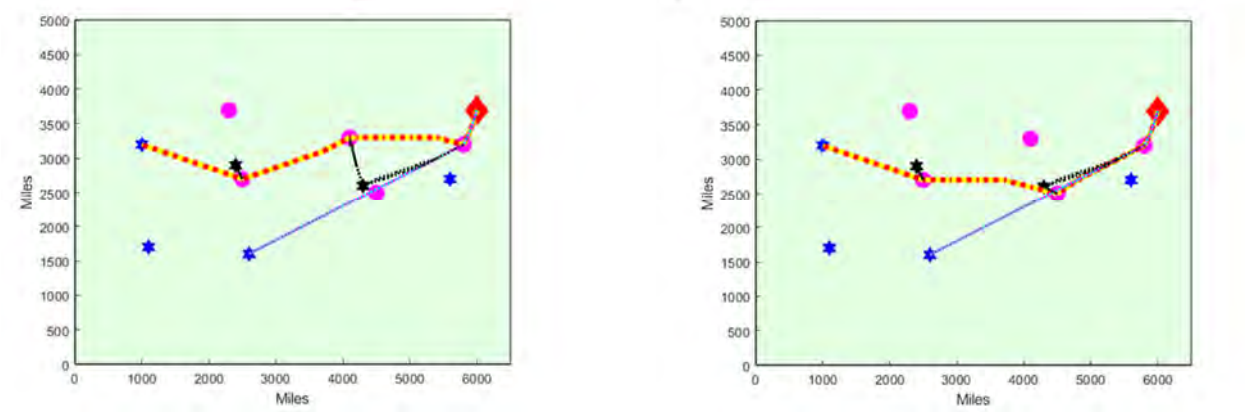


Figure 2. Notional comparison of scheduling receiver and tanker missions separately (left) rather than jointly (right).

Figure 2 illustrates a simple example in which a fighter and bomber need to reach the target represented by the red diamond. It's important to note that this is a notional set-up and not an actual operation. The fighter and bomber paths are represented by the dotted red and blue lines respectively. Meanwhile they also need to be refueled by tankers coming from the bases represented by black stars; the tanker paths are represented by black lines. Potential air refueling locations are represented by the magenta circles. The results from scheduling receivers and tankers separately versus jointly are shown in the left and right figures respectively. By scheduling jointly, the fighter follows a slightly longer path, but the tankers follow a significantly shorter path, so the overall fuel burn is reduced.

MIT LL is interested in experimenting with different methods for solving the joint receiver and tanker scheduling problem. There are tradeoffs between methods with respect to computational tractability and fidelity of the models in terms of capturing real-world constraints. The example shown in Figure 2 was generated using a dynamic programming approach on a notional problem. Another potential approach under consideration is mixed-integer optimization, and our task was to show the feasibility of using this technique for at least part of the air refueling problem. We developed a model that assigns tankers to receivers and captures the "physics" of fuel burn. The remainder of this section summarizes our model and results.

2.1 OPTIMIZATION MODEL

We make the following assumptions for this optimization model:

- Each receiver needs at most one refueling from a single tanker
- Receivers and tankers each start with a full tank of fuel
- Receivers are always refueled back to a full tank
- The rate of fuel burn for receivers and tankers is constant
- Refueling is completed over the course of one time step

The decision variables are as follows:

$$x_{ijt} = \begin{cases} 1 & \text{if tanker } i \text{ refuels receiver } j \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if tanker } i \text{ is engaged} \\ 0 & \text{otherwise} \end{cases}$$

$c_{jt} =$ fuel quantity of receiver j at time t

$d_{it} =$ fuel quantity of tanker i at time t

$$s_{it} = \begin{cases} 1 & \text{if tanker } i \text{ returns home at time } t \\ 0 & \text{otherwise} \end{cases}$$

The parameters are as follows:

C_j = fuel capacity of receiver j

D_i = fuel capacity of tanker i

S_j = minimum allowable fuel quantity for receiver j

Q_i = minimum allowable fuel quantity for tanker i

R_j = rate of fuel burn for receiver j

P_i = rate of fuel burn for tanker i

λ_1, λ_2 = penalties in objective function

Our model is shown in Equations (1) – (11):

$$\min \quad \sum_i y_i + \lambda_1 \sum_j \sum_t c_{jt} + \lambda_2 \sum_i \sum_t d_{it} \qquad (1)$$

$$\text{s.t.} \quad x_{ijt} \le y_i \qquad \forall i,j,t \qquad (2)$$

$$\sum_i x_{ijt} \le 1 \qquad \forall j,t \qquad (3)$$

$$\left|c_{jt} - C_j\right| \le C_j\left(1 - x_{ijt}\right) \qquad \forall i,j,t \qquad (4)$$

$$\left|c_{jt} - c_{j,t-1} + R_j\right| \le (C_j + R_j)x_{ijt} \qquad \forall i,j,t \qquad (5)$$

$$d_{it} \ge d_{i,t-1} - P_i - \left(C_j - c_{j,t-1}\right) \qquad \forall i,j,t \qquad (6)$$

$$d_{it} \ge d_{i,t-1} - P_i - C_j \sum_j x_{ijt} \qquad \forall i,t \qquad (7)$$

$$R_j + S_j - c_{j,t-1} \le R_j \sum_i x_{ijt} \qquad \forall j,t \qquad (8)$$

$$d_{it} - Q_i \le (D_i - Q_i)(1 - s_{it}) \qquad \forall i,t \qquad (9)$$

$$x_{ijt}, y_i, s_{it} \in \{0,1\} \qquad \forall i,j,t \qquad (10)$$

$$c_{jt}, d_{it} \ge 0 \qquad \forall i,j,t \qquad (11)$$

Below, we describe the objective function and each constraint:

- The objective function (1) has three parts: a) minimize the number of tankers used; b) minimize the total fuel carried by the receivers; c) minimize the total fuel carried by the tankers.
- Constraint (2) is a "forcing" constraint. If a tanker is engaged, then there must be an aircraft that is being refueled. We cannot have a tanker that is not engaged, yet a receiver is being refueled.
- Constraint (3) is an assignment constraint. This ensures that each tanker, if engaged, is assigned to only one receiver. The tanker may or may not be engaged at any particular time. Put differently, each receiver can only be refueled at most once.
- Constraints (4) and (5) govern the fuel quantity for receiver *j*. When $x_{ijt} = 1$, the fuel quantity is equal to the max capacity of the receiver since we assume the receiver has a full tank after

refueling. When $x_{ijt} = 0$, the fuel quantity is equal to the fuel quantity one time step before minus the fuel being the burned by the receiver.

$$c_{jt} = \begin{cases} C_j & \text{if } x_{ijt} = 1 \\ c_{j,t-1} - R_j & \text{if } x_{ijt} = 0 \end{cases} \qquad \forall i, j, t$$

- Constraints (6) and (7) govern the fuel quantity for tanker *i*. When $x_{ijt} = 1$, the fuel quantity is equal to the fuel quantity of the tanker one time step before. However, the rate of fuel burn and amount of fuel given to the receiver must be subtracted as well. When $x_{ijt} = 0$, the fuel quantity is equal to the fuel quantity one time step before minus the fuel being the burned by the tanker.

$$d_{it} = \begin{cases} d_{i,t-1} - P_i - (C_j - c_{j,t-1}) & \text{if } x_{ijt} = 1 \\ d_{i,t-1} - P_i & \text{if } x_{ijt} = 0 \end{cases} \qquad \forall i, j, t$$

- Constraint (8) forces a refueling to occur if a receiver's fuel level drops below the minimum allowable fuel level. This constraint ensure that the fuel content of receiver *j* at any time *t* never goes below the minimum fuel level S_j.
- Constraint (9) forces a tanker to return to base to occur if the tanker's fuel level drops below the minimum allowable fuel level. This constraint ensure that a tanker *i* at any time *t* is dealt with appropriately if it reaches its minimum fuel level Q_i.
- Constraint (10) enforces the binary nature of the *x*, *y*, and *s* decision variables.
- Constraint (11) enforces non-negativity of the *c* and *d* decision variables.

Note that this model does not explicitly incorporate the network over which the tankers and receivers are traveling (e.g., with nodes representing bases and targets). The purpose of this modeling task was to demonstrate the potential of using mixed-integer optimization to capture some aspect of refueling operations, and we decided to focus the modeling on the fuel transfer portion.

2.2 RESULTS

We implemented the optimization model using Julia, and tested it on a simple example problem. For instance, the mission requires 3 fighter jets, 1 tanker, and 10 time intervals in which to refuel. The results for this example are shown in Table 1. At any given time, tankers are or are not assigned to particular fighters such that the constraints of model and mission are met. The tanker aircraft ensure that the fuel level of the fighters never falls below a minimum threshold (4,500 lbs.) and that they have enough fuel to meet mission requirements. A refueling event occurs when a fighter requests fuel, which occurs at the 5th and 9th time interval for the first and third receiver aircraft and the 5th and 7th time interval of the second receiver aircraft. The fuel quantity for both the receivers and tankers are recorded as well. The objective value is mostly arbitrary since it depends on the pre-determined fuel amount for the tanker and receiver aircraft. However, the model seeks to minimize the objective value to optimize fuel consumption. The aim of the model is such that the request is made in an optimal location in which resources are used effectively.

Table 1. Test results for optimization model.

Time Interval	Receiver #1 Refueled	Receiver #2 Refueled	Receiver #3 Refueled	Receiver #1 Fuel Quantity	Receiver #2 Fuel Quantity	Receiver #3 Fuel Quantity	Tanker Fuel Quantity	Tanker Engaged
1	0	0	0	13,500	13,500	13,500	200,000	0
2	0	0	0	10,500	10,500	10,500	188,000	0
3	0	0	0	7,500	7,500	7,500	176,000	0
4	0	0	0	4,500	4,500	4,500	164,000	0
5	1	1	1	13,500	13,500	13,500	143,000	1
6	0	0	0	10,500	10,500	10,500	131,000	0
7	0	1	0	7,500	13,500	7,500	116,000	1
8	0	0	0	4,500	10,500	4,500	104,000	0
9	1	0	1	13,500	7,500	13,500	89,000	1
10	0	0	0	10,500	4,500	10,500	77,000	0

3 TANKER FLIGHT DATA ANALYTICS

In this section, we summarize our research towards helping MIT LL and AMC develop predictive maintenance methods that utilize flight data. For our capstone project, we had access to flight and maintenance data for the KC-135. KC-135s are a large part of the Air Force's tanker fleet, along with KC-10s and the new KC-46s. The KC-135 is one of the Air Force's oldest operationally active aircraft being first used in the Air Force in the summer of 1957. This aircraft has been used by the United States for over 60 years, and still operates to this day. Most of AMC's KC-135 aircraft have over 20,000 flying hours, as shown in Figure 3.

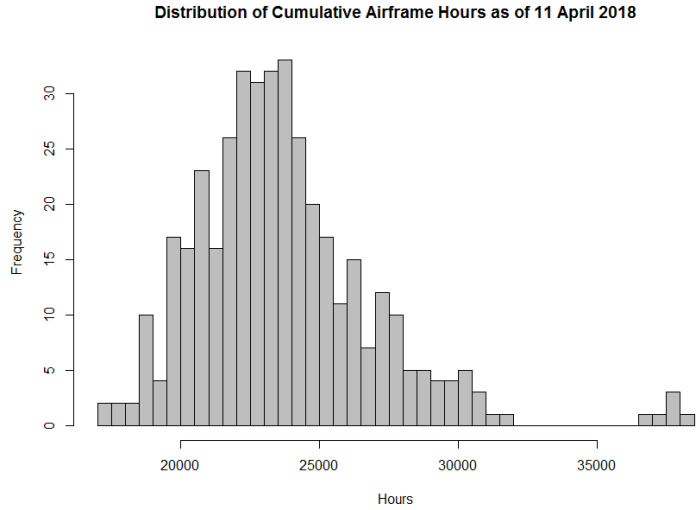


Figure 3. Distribution of KC-135 cumulative aircraft hours.

3.1 DATASETS

There were two primary datasets that we used in our analysis – flight data and maintenance data.

3.1.1 Flight Data

All aircraft have a flight data recorder that tracks multiple parameters over the course of a flight, for instance, time, location, heading, fuel, speed, weather, etc. In the Air Force, analysts use the data for Military Flight Operations Quality Assurance (MFOQA), which involves methods for detecting and measuring unstable approaches and other safety-related events on flights. Our flight dataset included data for 31,761 KC-135 missions across 396 unique tails that occurred between November 2017 and November 2018. We focused on 44 features in this dataset, though the actual flight recorder data contains hundreds of features.

3.1.2 Maintenance Data

We also had access to a maintenance dataset for the KC-135 fleet (398 unique tails) covering a time frame from January 2017 through April 2018. The maintenance data provides a daily accounting of the mission capability status for each tail: mission capable (MC), not mission capable (NMC), in depot, or unknown. The MC and NMC categories can be further subdivided according on aircraft status – for example, fully vs. partially mission capable, or NMC due to supply, maintenance (whether scheduled or unscheduled), or both.

3.2 ANOMALY DETECTION

One of our initial tasks using the flight data was to apply unsupervised anomaly detection methods. Anomaly detection is a field of research which identifies irregularities in a dataset. In the case of flight data, anomaly detection allows us to determine which flights might be considered abnormal with respect to specific aircraft performance features, which could indicate precursors to unscheduled maintenance. There are a variety of ways to implement anomaly detection. In this section, we describe our results using simple "thresholding" methods in which we label flights as outliers if they exceed a certain threshold, and K-Nearest Neighbors (KNN).

3.2.1 Threshold Methods

At first glance, anomaly detection appears to be a very simple problem: look at the data points and find the ones that are different from the rest. In practice, this concept becomes much more difficult when dealing with high-dimensional datasets, because the problem is harder to visualize. We started with a low-dimensional anomaly detection example to motivate this approach.

We first took two variables that are related to the physics of an aircraft's fuel efficiency. The two variables are the amount of fuel burned throughout an entire flight, and the distance the plan traveled. Most people who have ever driven some type of vehicle understand that the amount of fuel that you burn is almost directly proportional to the distance you traveled. Therefore, a plot of the aircraft fuel burned vs the distance traveled should reveal a distinct linear relationship between these two variables (see Figure 4).

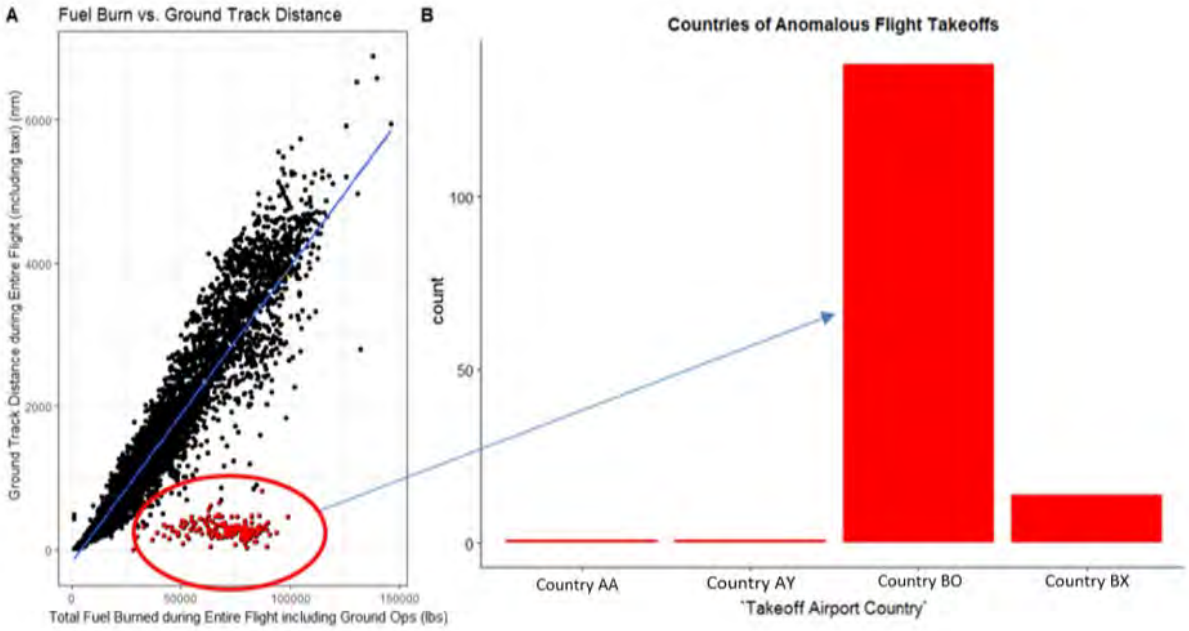


Figure 4. Anomalous flights analysis.

However, once we plotted these two variables from our dataset, we uncovered that some points do not follow this normal behavior. Looking at Figure 4, the red points appear to be in their own separate cluster; these points can be identified as having a distance to fuel burn ratio (fuel efficiency) threshold below 100. Said differently, these flights are burning significantly more fuel than expected. So we asked ourselves, why does this happen? After looking more closely at these anomalous data points, we found that the majority of these flights took off in Country BO (see Figure).

This development caused us to look further into the effect of the takeoff airport on fuel burn efficiency. Now that we detected the anomalies, we wanted to understand what factors contributed to this abnormal fuel burn effect. We broke down each fuel burn efficiency plot by country and saw that certain countries, namely Country BO and Country BX among others, had anomalous fuel burn patterns which did not represent the expected linear relationship (see Figure 5). There are several factors that can cause these abnormalities such as temperature, operational constraints or even specific flight standards within the region of Countries BO and BX.

We performed correlation analysis on temperature to determine if basic statistical methods could help explain some of these anomalous behaviors. After performing correlation analysis for fuel burn efficiency rates and latitude as a proxy for geographic temperature, we did not conclude that temperature had a statistically significant effect on fuel efficiency. From this conclusion, we concluded that just because data abnormalities occurred within areas which tend to be hot, does not mean that heat caused these abnormalities.

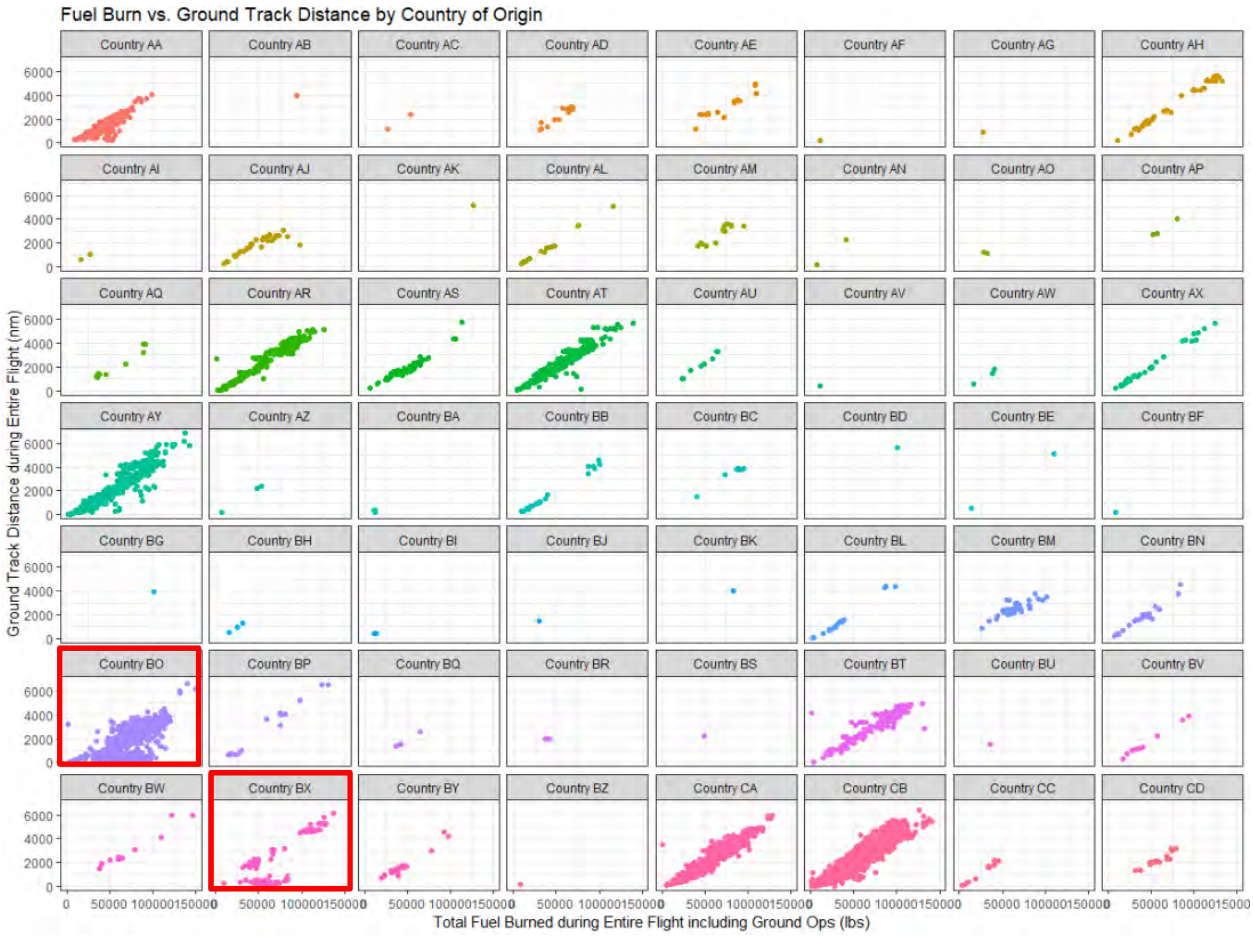


Figure 5. Fuel burn and ground track distance by country.

At this point we have uncovered that certain flights in specific countries have anomalous fuel burn patterns simply by visualizing the data points and seeing which ones do not match the expected fuel burn pattern. This approach provides insights, but does not paint the full picture of the anomaly detection problem. This approach only includes two variables of the hundreds of variables represented in our dataset. Therefore, we had to come up with different ways to detect anomalies using more of the variables.

3.2.2 K-Nearest Neighbors

In multi-dimensional datasets, the anomalies which were evident in the above visualizations are no longer as easy to detect. The problem quickly becomes more complex because the outliers are able to hide behind multiple dimensions of variables and are much more difficult to pick out because of their interactions. We, therefore, researched several distance-based anomaly detection methods that combated this dimensionality problem (represented in Appendix A). After surveying six different methods, we determined that the most interpretable and accurate method is the K-Nearest-Neighbors (KNN) detector. To explain this anomaly detection method, we first

compare the KNN method on the same two variables shown in the above “thresholding” approach (see Figure 4 for thresholding approach and Figure 6 for KNN approach).

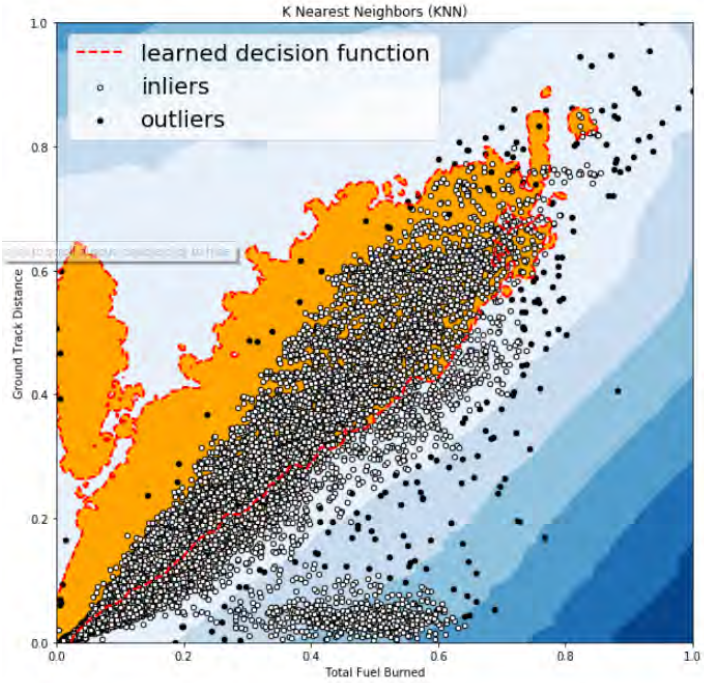


Figure 6. KNN anomaly detection on ground track distance and fuel burn.

The graph in Figure 6 shows the output of the KNN anomaly detection algorithm. This algorithm first compares a data point’s average distance to its K nearest neighbors. The average distance is then assigned as the data point’s outlier score. The idea is that outliers are points that have a large average distance from its closest K neighbors or a high outlier score. The next step in the algorithm is defining the fraction of outliers that will be detected. In our analysis we chose a value that outputs the top 1% of data points that have the highest outlier score. Once this value is defined, the algorithm learns a decision function (the red line in Figure 6) which finds the location where inliers are expected to turn into outliers. The orange contour represents areas of high expected inlier density, and the blue contours represent location of high expected outlier density. Lastly, the anomalous points, which are the points with the highest 1% of outlier scores, are represented as the black points in Figure 6.

This approach is fairly easy to understand, but it can sometimes be sensitive to the value of K- that is, how many neighboring data points you compare it to. The most important thing to note from this KNN graph is that the anomalous points described by the thresholding approach have now changed to the points in between the two clusters. This KNN anomaly detection algorithm is able to identify hidden interactions between the points that do not fit inside the two evident clusters. Because this is an unsupervised problem, there is no way to output the accuracies for either of the approaches as the definition of an anomaly is strictly dependent upon the method chosen to detect the outlier. The bottom line is that the KNN is able to detect finer intricacies

between two variables compared to thresholding, and this effect scales when the dimensionality of the datasets increase. We, therefore decided to use the KNN anomaly detection algorithm for the remaining steps of the project because of its ability to scale in higher dimensions.

The next step in the modeling process involved determining which parameters are most relevant for common maintenance issues so that we could include more variables in the anomaly detection software. After discussing with flight maintenance experts from Lincoln Laboratory's Flight Facility, we concluded that engine performance is a potential area where detecting anomalies could prove useful in predicting maintenance events. Important metrics that we had access to in our dataset were engine exhaust gas temperature (EGT), fuel flow, ground track distance, oil pressure, N2 average, and hard landings. For example, high-EGTs can lead to structural cracks in the engine. This hypothesis about high EGT led us to ask whether EGT had similar geographic anomalies as the Fuel-Burn vs Distance visualizations in Figures 4-6. We therefore created a histogram of the flight location pairs for High Maximum EGT flights to determine where high EGT flights occur the most (see Figure). A high Maximum EGT flight is determined as one that was in the top 25% of the Max EGT distribution which is a variable found in the flight record file dataset.

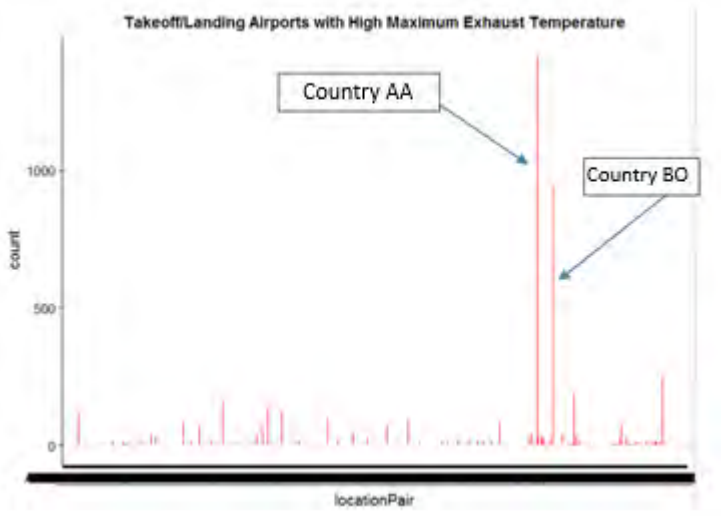


Figure 7. Histogram of high EGT flights.

These findings conclude that the majority of High Max EGT flights occur in two flight location pairs. One is Country AA and the other is Country BO, both countries were also flagged as abnormal counties in threshold methods section of this analysis. This finding reinforces the suspicion that geography has a large effect on engine performance metrics in the flight dataset and might need to be controlled for in the anomaly detection software and machine learning component of the project.

The last step in the anomaly detection process is scaling the KNN algorithm in higher dimensions. In our discussion with Lincoln Laboratory's Flight Facility, we surveyed two

maintenance experts on which parameters would have the largest effect on maintenance problems. We used their feedback to come up with three initial variables used to test the KNN anomaly detection software on. The three variables are: average exhaust gas temperature (EGT), N2 Average which is a measure of the rotational fan speed of the N2 section of an engine, and Fuel Flow Total which models when an aircraft is going through a refueling event. The results of the KNN anomaly detection software on a random flight from our dataset is shown in Figure 8.

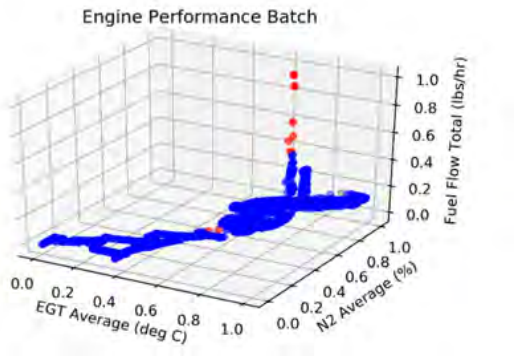


Figure 8. KNN anomaly detection in three dimensions.

The red points on this graph represent the outliers that were detected by the KNN software. These points were the points that had the highest outlier score when comparing the three variables EGT average, N2 average, and Fuel Flow Total. Overall, these outliers are difficult to interpret because of the dependencies between the variables. However, this example was created to show that the KNN algorithm can scale in higher dimensions which the thresholding approach cannot. Experts can use this algorithm to batch together various flight parameters and find the time at which these anomalous events occurred. Once the experts determine when anomalous points occur during the flight, they can begin to build a greater understanding of the abnormal time periods of that flight.

In conclusion, if we know the parameters that lead to anomalous flights, then those parameters can be monitored more closely in the future. This analysis can be directly applied to the predictive maintenance portion of the project, by assigning an anomaly score to each of the flights. The anomaly score would then be added as a feature in the dataset to represent abnormal occurrences during a flight. This added feature would allow USTRANSCOM to determine whether these in-flight anomalous refueling events have any effect on the overall health of the aircraft. This approach was never implemented because it was out of the scope of our project, but it is a good way to improve the overall performance of our machine learning algorithms and link the anomaly detection portion of the project to the machine learning portion.

3.3 CLASSIFICATION METHODS TO PREDICT MAINTENANCE

Our main motivation for this section of the report are maintenance issues and mission effectiveness. A recent article was published on the Air Force's new Condition-Based Maintenance Technique (CBM+) which uses predictive analytics to track when aircraft parts are

at risk of breaking.[2] Currently, maintenance crews must wait until an aircrew informs them that a part has broken. This approach is very reactionary and can lead to lost flight hours because the aircraft is grounded until the maintenance crew has the time and resources to fix the aircraft. By using CBM+, the Air Force can anticipate when a break is likely to happen, and use this information to turn unscheduled maintenance into scheduled maintenance. Our project is closely related to this CBM+ initiative but uses flight data instead of tracking broken aircraft parts to predict maintenance issues. Our dataset consists of variables such as total miles flown by the aircraft, number of times the aircraft has had repairs in the last 30 days, and engine performance characteristics. Although our dataset does not have detailed metrics on broken aircraft parts, there are generalizable maintenance variables which describe the overall health of the aircraft. The goal of this part of the project aims to predict when a maintenance event will occur for a specific aircraft in order to maximize aircraft mission readiness.

3.3.1 Data fusion

In order to derive predictive maintenance algorithms, we had to first fuse the two datasets described in Section 3.1.We took the portions of the datasets that overlapped in time – November 2017 through April 2018 – and merged the data using tail numbers, shown in Figure 9. The maintenance file is in the blue box on the left and the flight record file is the red box on the right. The flight record file is a summary of the actual time series flight data. The original maintenance file had 179,896 observations and 48 features. The flight record file had 26,169 observations and 44 features. The merged had 9,517 observations and 39 features, containing flight recorder information, summary statistics, and historical aircraft performance information over weeks and months, which tracked the performance of an aircraft over a given number of days in the past. **Error! Reference source not found.** shows the features of the merged dataset. Note that many of the features in Table 2 were not in the source datasets (e.g., those that are cumulative counts over the last 30 days), but were engineered to evaluate whether they are predictive of maintenance issues.

After merging the datasets, there were 3,597 missing values across various variables. We used the OptImpute software to impute these missing values.[3] The software uses a formal optimization to impute missing values based on predictions from various models such as K-Nearest Neighbors, support vector machines and tree-based methods. This step is necessary in the machine learning portion of the project and improved our out-of-sample model accuracies.

[2]Losey, Stephen. “It Ain’t Broke, but Still Fix It: How Predicting Repairs Is Transforming Maintenance.” C4ISRNET, 2019, https://www.c4isrnet.com/news/your-air-force/2019/07/20/it-aint-broke-but-still-fix-it-how-predicting-repairs-is-transforming-maintenance/?utm_source=clavis.

[3] Interpretable AI, OptImpute and Optimal Trees Software, author = "Interpretable AI, LLC", "Interpretable AI Documentation", 2019 https://www.interpretable.ai

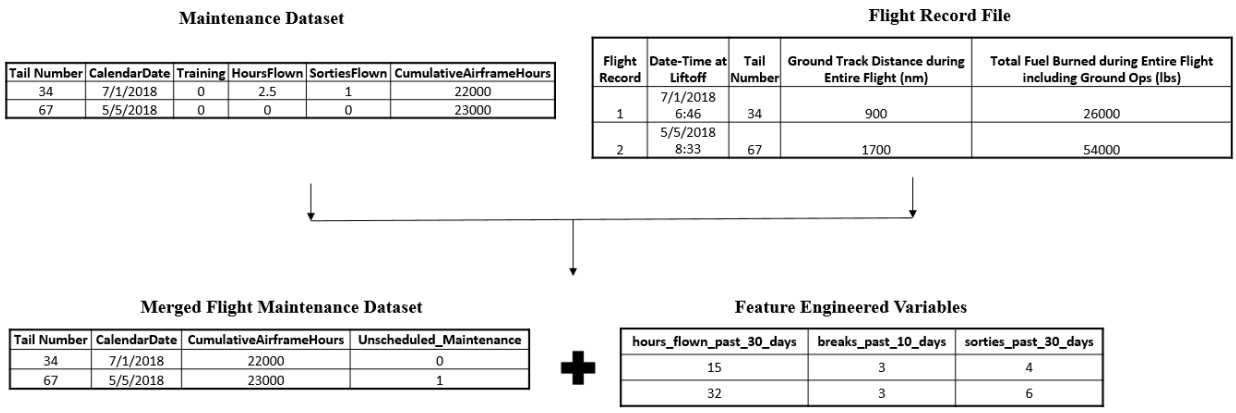


Figure 9. Illustration of merging the (notional) datasets.

Table 2. Features of merged KC-135 Dataset (excluding feature engineering).

Great Circle Distance	In-flight Between Refueling
Ground Track Distance In Air	Stable Criteria Not Met
Ground Track Distance Entire Flight	First Approach Go Around
Takeoff Longitude	Takeoff Latitude
Total Fuel Burn Entire Flight	Hard Landing
Total Fuel Burn In Taxi	In-flight Refueling
Total Fuel Burn On Ground	Tail Number
Total Fuel Burned In Air	Calendar Date
Decrease In Weight	Landing Latitude
Max EGT	Landing Longitude
Total Fuel Landing	Cumulative Airframe Hours
Total Fuel Takeoff	Go Around Count
Total Fuel Unloaded Refueling	Maintenance Event

3.3.2 Generating the response variable

In order to apply predictive methods to our dataset, we needed to decide what to actually predict. The maintenance dataset provided information about when each aircraft was in a certain maintenance state. There were also confounding factors in the data. For example, in one of our initial models, a feature that appeared highly predictive of maintenance was that the flight before would land at Tinker Air Force Base. However, Tinker is one of the primary locations used to service KC-135s. Therefore, having that feature identified by our model was a red herring. To control for this confounding variable, we removed the variables for latitude and longitude from the data.

We decided for the response variable to use the number of non-mission capable aircraft hours due to unscheduled maintenance events instead of depot events. That is, we assigned a "1" to an aircraft if it had any unscheduled maintenance hours during that day. Effectively, we are tracking the evolution of an aircraft over time, both its flight and maintenance activities.

3.3.3 Feature engineering

Another approach we explored to further improve our analysis was feature engineering. For instance, we created various historic variables such as counting the number of breaks for an aircraft within the past 10 days, and the number of sorties and hours flown in the past 30 days. These feature-engineered variables were taken from both the maintenance and flight recorder datasets to incorporate features that measure the aircraft's performance over time. Other metrics considered were the number of scheduled and unscheduled repairs. These variables provided a greater understanding of how an aircraft's previous flights affects its ability to stay mission capable. Please see Table 3 for a full list of the feature-engineered variables.

Table 3. Feature-engineered variables.

Engineered Feature	Description
Hours flown past 30 days	Number of Hours flown by one aircraft in last 30 days
Breaks past 10 days	Number of maintenance part breaks in last 10 days
Sorties past 30 days	Number of flights in 30 days
NMCMHrs past 30 days	Not Mission Capable, Maintenance Hours in last 30 days
NMCMUHrs past 30 days	Not Mission Capable, Maintenance Unscheduled Hours in last 30 days
NMCMSHrs past 30 days	Not Mission Capable, Maintenance Scheduled Hours in last 30 days
Training missions past 30 days	Number of training missions in the last 30 days
MCHrs past 30 days	Mission Capable Hours in last 30 days
NMCBUHrs past 30 days	Not Mission Capable, Maintenance Both Unscheduled Hours in last 30 days
Recur past 30 days	Number of recurring maintenance failures in last 30 days
OnEqMHS past 30 days	On Equipment Maintenance Hours in last 30 days
OfEqMHS past 30 days	Off Equipment Maintenance Hours in last 30 days
x8HrFix past 30 days	Number of 8 hour fixes in last 30 days
x12HrFix past 30 days	Number of 12 hour fixes in last 30 days
x24HrFix past 30 days	Number of 24 hour fixes in last 30 days
Red Ball Count past 30 days	Number of Maintenance issues that prevented on-time launch
Air Abort Count past 30 days	Number of Maintenance issues that aborted mission after takeoff
Ground Abort Count past 30 days	Number of Maintenance issues that aborted mission before takeoff
Avg MaxEGT Past 5 Flights	Average Maximum Exhaust Gas Temperature over past 5 flights
Total Fuel Burned Past 5 Flights	Total fuel burned in lbs. over past 5 flights

Total Ground Track Distance Past 5 Flights	Total distance flown over past 5 flights
In-flight RefuelingTotal Past 5 Flights	Number of refuelings in flight over past 5 flights
Hard Landings Past 5 Flights	Number of hard landings over past 5 flights

3.3.4 Models and performance metrics

The machine learning models we used were optimal classification trees (OCT), Xgboost, logistic regression, CART, support vector machines (SVM), and random forests. These models are able to perform classification tasks in which each observation or flight is classified as having or not having an unscheduled maintenance issue on any given day. The models were tuned to select the subset of parameters that resulted in the best performance when using training and testing datasets.

We split our dataset as follows. The complete dataset contained 9,517 flights. We used 7,137 randomly chosen flights for training, and used the remaining 2,380 for testing.

There were four measures of performance that were used to compare the various models: Area Under the Curve (AUC), Test Accuracy, True Negative Rate, and False Negative Rate. We illustrate the calculation of these metrics based on the example shown in Table 4.[4]

Table 4. Example data for calculating performance metrics.

	Predicted	
Actual	0	1
0	1727	81
1	453	119

- **Area Under the Curve =** Probability of differentiating when an unscheduled maintenance event does and does not occur.
- **Test Accuracy =** Percent of correct maintenance classifications
  Ex: Test Accuracy = (1727+ 119)/ (1727+ 119+453+81) = 77.6%
- **True Negative Rate =** Increase in necessary scheduled maintenance from unscheduled maintenance(higher value is good)
  Ex: TNR = (119)/ (119+453) = 20.8%
- **False Negative Rate =** Increase in excess scheduled maintenance that was not required to keep aircraft mission capable (higher value is bad)
  Ex: FNR = (81)/ (1727+81) = 4.48%

[4] Note this example uses data from the logistic regression results.

The simplest measure of performance is Test Accuracy. This column is interpreted as the number of times that the model correctly identified the flight as having an unscheduled maintenance event or not. This is a useful measure of performance, but it does not paint a complete picture of the model's overall performance because a "baseline" model that just predicts the flight will never have an unscheduled maintenance event would achieve a Test Accuracy of 75.97%. This type of baseline model would not be of use because it does not provide any meaningful insights. Therefore a better measure of performance is AUC which outputs the probability of differentiating when an unscheduled maintenance event does and does not occur. This measure of performance is more useful because it accounts for whether the model can correctly differentiate not only when an aircraft may need unscheduled maintenance but also when it does not.

The last two measures of performance are false negative rate and true negative rate. The false negative rate indicates the rate at which the model predicts that the aircraft should have scheduled maintenance performed when in reality it does not need any scheduled maintenance for that particular flight. This metric is important because the DoD does not want to waste resources. Maintenance operations are always constrained for time, and often have to prioritize which aircraft to repair. Our model cannot have a high false negative rate because this would flood maintenance operations with unnecessary work. The true negative rate indicates the rate at which the model correctly identifies unscheduled maintenance issues. This is the most important measure of performance to take into account with respect to impact for USTRANSCOM. The true negative rate is a measure for the percentage of hours that USTRANSCOM can turn from unscheduled maintenance to scheduled maintenance. This is valuable to USTRANSCOM because this information may be used to plan ahead of time the aircraft that are most likely to need unscheduled maintenance.

3.3.5 Results

The values of the performance metrics are listed in Table 5, ordered with respect to AUC. The top performing model was the Random Forest. This model achieved a 76% AUC and a true negative rate of 21%. Optimal classification trees (OCT) performed fairly well but its key strength was being the most interpretable model. The logistic regression was also a high performing model which is easy to understand with respect to individual variables, but cannot explain variable interactions as well as the OCT. We used other methods as well, including Xgboost, SVM, and CART, with the goal of comparing the performance of the models to one another. The tree based techniques allowed us to see which variables and their associated values were most important in determining how to classify each observation. The test accuracies are all very similar (near 77%) as the dataset is highly imbalanced in terms of predicting no maintenance for most of the observations.

Table 5. Performance metric values for classification methods.

Method	AUC	Test Accuracy	True Negative Rate	False Negative Rate
Random Forest	0.7580	79.08%	20.98%	2.54%
Xgboost	0.7480	78.87%	19.76%	2.43%
Logistic Regression	0.7318	77.61%	20.80%	4.48%
SVM	0.7145	78.61%	20.80%	3.10%
OCT	0.7139	76.89%	17.66%	4.26%
CART	0.7064	77.61%	17.31%	3.60%

Here we provide additional detail for two of the models: OCT and logistic regression. First, Figure 11 shows the final OCT model.

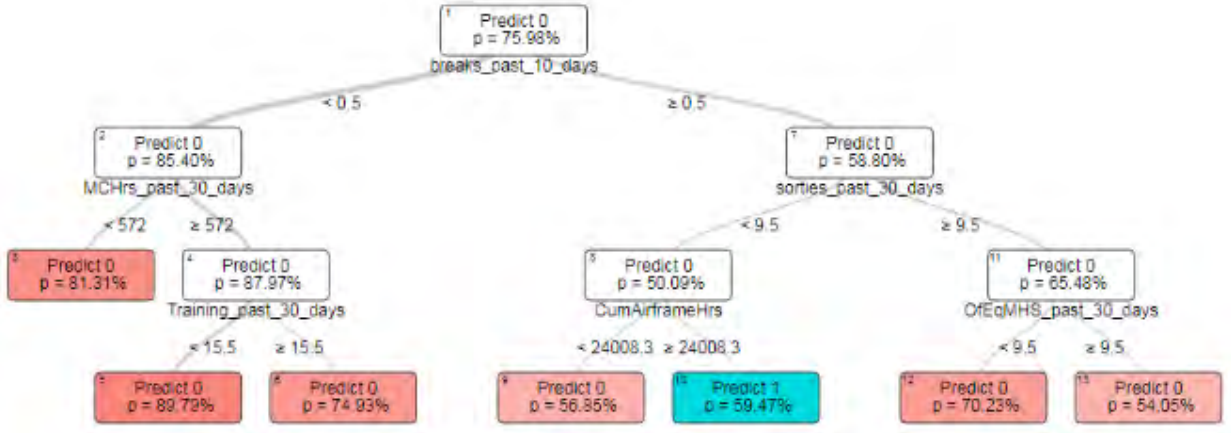


Figure 10. Final OCT model.

The tree first splits on the number of times the aircraft was found to have a break in the last 10 days. If there were no breaks for a plane in the last 10 days, then that aircraft had an 85.40% (see split #2 in Figure 10) probability that it would not encounter a break on the current flight. The most important box to look at in the tree above is the blue colored leaf. This leaf contains the flights that are most likely to have unscheduled maintenance events. Within this blue leaf, 59.47% of the 454 flight observations were correctly predicted to have an unscheduled maintenance event. Considering all the potential factors that go into an aircraft breaking down, this is a significant result which can provide insight to the AF maintenance crews. The distribution for each of these critical splits are seen below. The red bars are representative of the data points contained within the critical splits leading to the blue leaf. Each critical split occurs near the median of the data which shows that the splits are not overfitting to extreme points within the variable distributions.

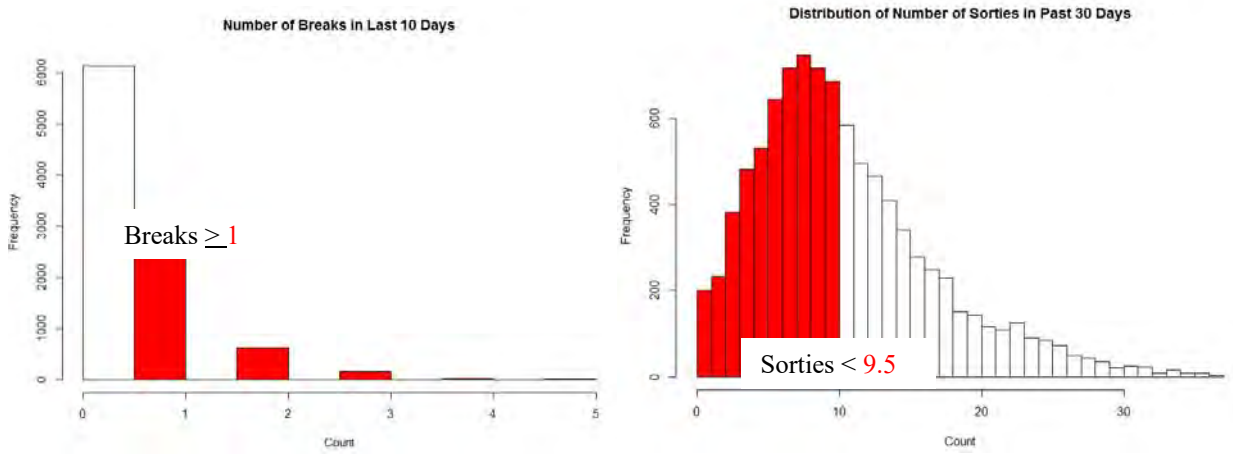


Figure 11. Histogram of Tree Split 1 (Number breaks) and 2 (Number sorties in the past month).

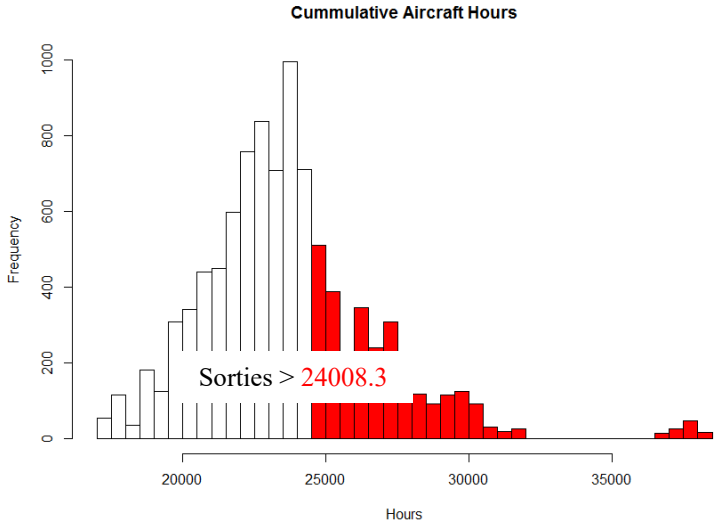


Figure 12. Histogram of Tree Split 3: cumulative aircraft hours.

A breakdown of the variable importance within the model is shown in Figure 13. The main takeaway from this analysis is: if an aircraft has had any breaks in the last 10 days, has flown less than 10 times in the last 30 days, and has more than 24,008 accumulated flying hours, it has a 60% chance of breaking down on its next flight. Therefore, these results should cause concern for aircraft crew that may be dealing with flight profiles that fit these characteristics.

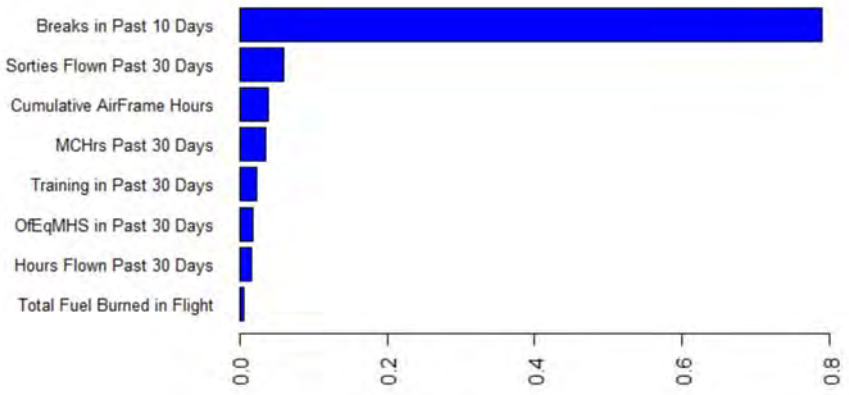


Figure 13. Relative importance of OCT variables.

One of the more simple methods we used, logistic regression, was also informative. This model is accurate and relatively easy to understand compared to the other more complicated approaches, such as Xgboost and random forest. The model is shown below.

(Intercept)	greatCircleDistance	groundtrackDistAir
-1.31E+00	-1.29E-01	-5.83E+01
groundtrackDistEntire	totFuelBurnEntire	totFuelBurnTaxi
5.83E+01	8.87E+00	-7.15E-02
totFuelBurnGround	totFuelBurnAir	decreaseWeight
-1.01E-01	-8.95E+00	-2.59E-01
maxEGT	totalFuelLanding	totalFuelTakeoff
-1.30E-01	-1.14E-01	3.34E-01
totalFuelUnloadedRefueling	inflightRefueling	inflightBetweenRefueling
-1.41E-02	1.23E-01	-2.00E-01
stableCriteriaNotMet	firstApproachGoAround	goAroundCount
3.80E-02	5.15E-02	8.17E-03
hardLanding	avg_MaxEGT_past5Flights	totalFuelBurn_past5Flights
7.52E-02	-1.99E-02	-7.66E-02
roundTrackTotal_past5Flight	inFlightRefuelingTotal_past5Flights	hardLandings_past5Flights
1.33E-01	-5.11E-02	-6.64E-02
CumAirframeHrs	hours_flown_past_30_days	breaks_past_10_days
1.67E-01	1.67E-01	7.34E-01
sorties_past_30_days	NMCMSHrs_past_30_days	Training_past_30_days
-2.24E-01	-7.81E-02	1.29E-01
MCHrs_past_30_days	OnEqMHS_past_30_days	OfEqMHS_past_30_days
-2.52E-01	5.63E-04	4.23E-02
x8HrFix_past30_days	x12HrFix_past30_days	x24HrFix_past30_days
1.64E-02	-7.32E-02	-6.05E-03
redBallCount_past30_days	airAbortCount_past30_days	groundAbortCount_past30_days
1.19E-01	4.86E-02	4.11E-02

Figure 14. Logistic Regression Model Output.

The strength of this model is its simplicity. The main weakness of this model is that it does not take into account the multitude of potential interaction variables as compared to tree based methods. The most important variable output from this model was Breaks within the Last 10

Days. This variable was significantly more important than the rest of the variables as indicated in Figure 15 which shows information gain and chi-squared variable importance metrics.

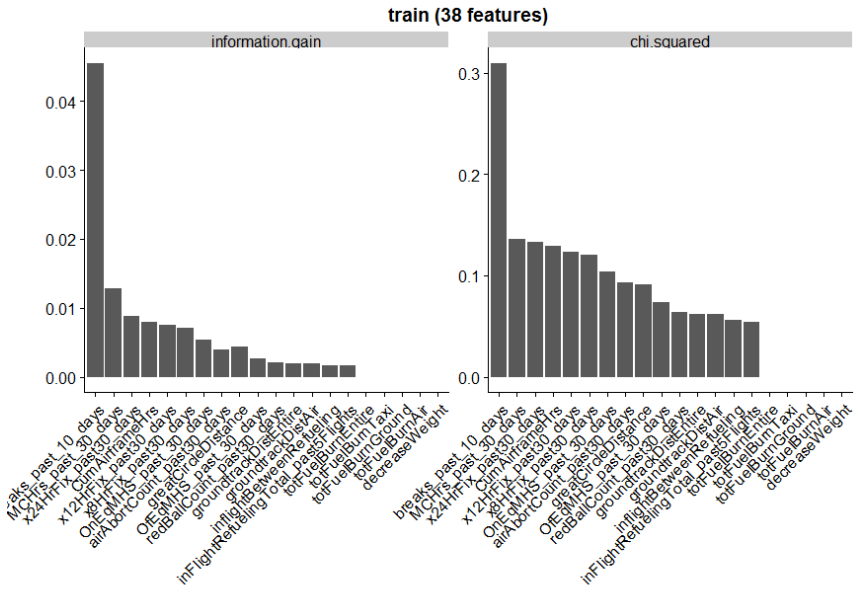


Figure 15. Logistic regression variable importance plot.

We will use this variable in an example to show how to interpret the model. For instance, an increase in the number of breaks, which has a positive coefficient value, will result in a greater probability of a maintenance issue occurring. In other words, changing the number of breaks by one unit increases the log odds ratio of a maintenance event by the value of its regression coefficient (0.734). The interpretation of this value is: if the number of breaks in the past 10 days increases by one, then we will see a 108% ($e^{0.734}$) increase in the odds of having an unscheduled maintenance problem. This logic can be followed for each of the variables included within the logistic regression equation.

3.3.6 Impact of predictive maintenance

Unlike a typical business, the top priority for the DoD is mission effectiveness, rather than cost savings or revenue generation. Nevertheless, the DoD is invested in predictive maintenance, because small gains in efficiency can lead to significant cost savings. Therefore, creating more effective mission plans will inherently reduce costs. A useful metric we can look at is the number of total unscheduled maintenance hours in our dataset over the course of 452 days between 2017 and 2018. During this time period, there were 348,818 total hours of unscheduled maintenance that were performed on the 398 KC-135 aircraft. Using the top performing machine learning models outlined above, we were able to correctly identify ~20.8% of the unscheduled maintenance occurrences. Extrapolating these results to one year's worth of data, USTRANSCOM could potentially convert up to 58,253 hours of unscheduled maintenance to scheduled maintenance.

4 CONCLUSION

The theme of our capstone project was analytics to support USTRANSCOM tanker planning. Since both capstone partners are in the Air Force, this project was an opportunity for young Air Force analysts to use new and existing analytics techniques to solve problems in the military.[5] We represent part of the military presence at MIT LL and our work has implications on how the military is viewed within the organization. The main takeaways for our sponsor, corresponding to the two parts of our capstone project, are as follows:

First, it is possible to use mixed-integer optimization for certain modules within a tanker planning algorithm. We demonstrated the technique in particular for capturing the constraints related to fuel transfer from the tanker to the receiver. Next steps would include incorporating constraints to capture the network over which the aircraft are operating, and to test the method on larger problems to evaluate scalability.

Second, it is promising to use flight data as part of a methodology for aircraft predictive maintenance. Since the flight dataset was a new resource for the MIT LL team, our role was largely exploratory and path-finding. Our work helped reveal some of the challenges of working with flight data and merging it with other data sources. Next steps would include additional feature engineering to capture factors that may influence maintenance needs, for instance, the anomaly scores described in Section 3.2.2; as well as fusing additional maintenance data that includes details about specific repairs.

In summary, tankers are essential to the DOD's ability to accomplish its missions around the world. There is significant opportunity to leverage data and advanced analytical methods to improve tanker planning and the health of the fleet. Our work has helped the MIT LL team support USTRANSCOM and AMC to achieve these goals.

[5] https://www.airforce.com/careers/detail/operations-research-analyst

APPENDIX A

Figures 16 and 17 show the results of two additional anomaly detection techniques. These figures label the outliers, inliers, and the boundary for the learned decision function.

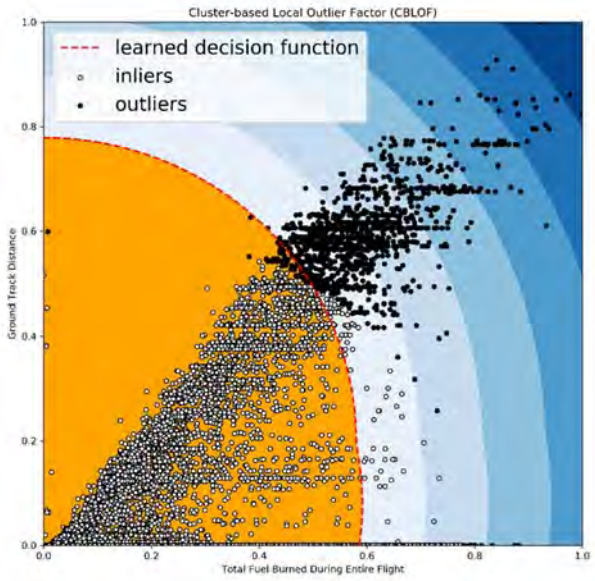


Figure 16. Cluster-based local outlier factor anomaly detection technique.

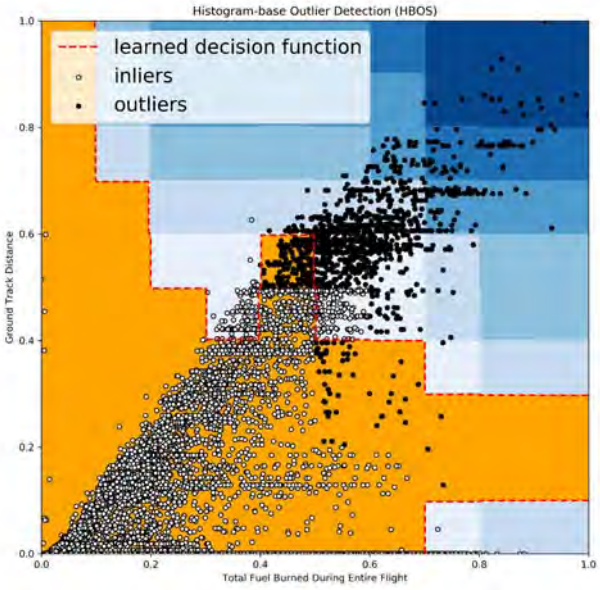


Figure 17. Histogram-based outlier anomaly detection technique.

Figure 18 is a distribution of the total number of maintenance events for each plane. More specifically, the graphic shows the frequency of unscheduled maintenance occurrences within the dataset.

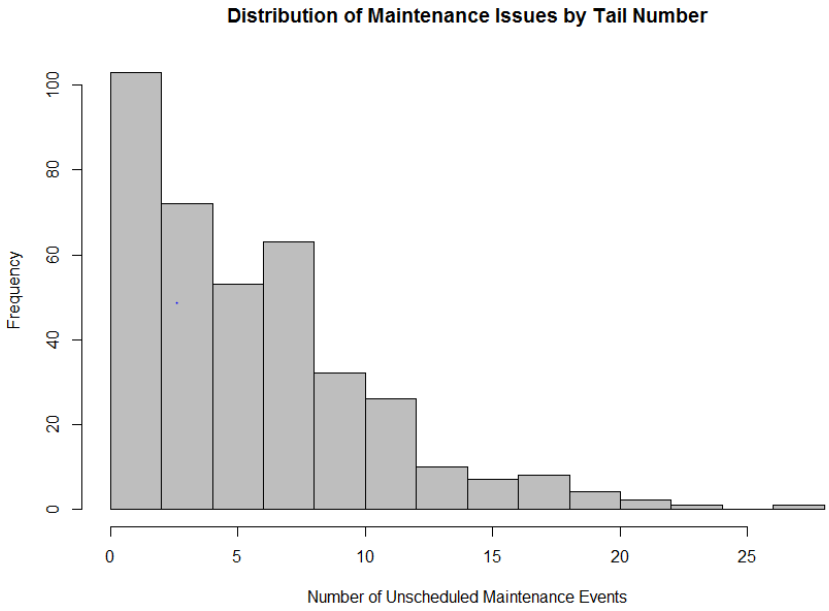


Figure 18. Count of unscheduled maintenance events by tail number.