Introduction

Spacecraft are incredibly complex machines, and thrusters are central to any spacecraft we have today. A monopropellant thruster is an engine that uses a unique propellant, unlike a standard bipropellant thruster, which uses a combination of fuel and oxidizer. The flow of propellant into the thruster is controlled by a valve, and it is then injected into a catalyst bed, where it decomposes into a hot gas. These hot gasses are expellled through a nozzle, generating thrust. Generally monopropellant thrusters are smaller reaction control thrusters which generate 1 to 10 Newtons of thrust. These thrusters are not used for main propulsion, rather course correction, reorientation, and docking.

The performance of one of these thrusters is dictated by valve performance and degradation of the catalyst bed that the propellant decomposes on. The catalyst bed consists of many Indium granules, and it is constructed this way to optimize its lifetime. Catalyst granules experience degredation when under thermoelastic shocks, collisions with other granules, and more. They gradually break up into finer particles, reducing their efficiency, and ultimately causing voids in the catalyst bed. This causes degradation of thruster performance. Vapor lock is another phenomenon that can damage the thruster. Vaporization occurs when incoming injector pipe propellant is overheated, and some of the incoming propellant vaporizes or even creates a bubble, blocking flow completely.

To avoid degradation, it may be possible to use predictive algorithms to detect potential vapor lock and decomposition before it happens. These algorithms can then be deployed onboard spacecraft to warn the crew that repairs are required, thus avoiding catastrophic failure.

About the Data

This data consists of 2612 unique .csv files corresponding to different firing tests and one compiled metadata file. The metadata file contains the following columns:

- uid - unique identifier of the fire test run
- filename - name of the file containing the detailed test data
- test_id - test identifier
- sn - thruster serial number
- test_pressure - pressure at the inlet of the thruster
- test_mode - Different operational modes which impose different phyisical constraints on the thrusters
- vl1 - flag if the series has experienced level one propellant vaporization (minor blockage)
- vl2 - flag if the series has experienced level two propellant vaporization
- vl3 - flag if the series has experienced level three propellant vaporization.
- anomalous - flags whether or not an anomaly has occurred
- anomaly_code - provides additional information about the type of anomaly that has occurred
- cumulated_throughput - cumulative mass (in kg) of propellant that has been injected in the thruster and decomposed on the catalyst bed between its beginning of life and until the beginning of the test sequence.
- cululated_on_time - cumulative duration (in hours) that the thruster has been “on”
- cumulated_pulses - cumulative number of pulses (dimensionless) that the thruster has been commanded since its beginning of life and until the beginning of the tests - thruster aging factor

Different columns were removed for different experiments and experiment methods. Additionally, it is specified in the data’s supplementary information file that vl1,vl2,vl3, anomalous, and anomaly_code cannot be used as inputs to a predictive model.

The other 2612 files contain the following columns:

- time - timestamp of measurement

- ton - on-off command to thruster. 1=on 0=off.
- thrust - measured thrust in Newtons
- mfr - mass flow rate in mg/s with 100Hz frequency
- vl - indicator that vapor lock has occured
- anomaly_code - indicator that an anomaly has occured, and information on the type of anomaly

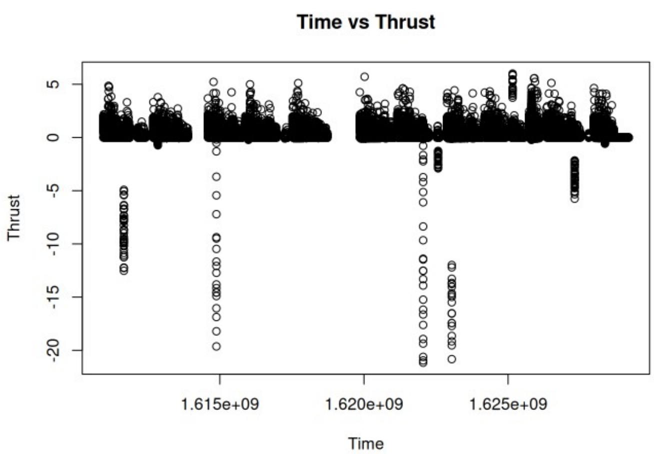
In our anomaly prediction experiments, we generalized the type of anomaly to 1(anomaly occurred) and 0(anomaly didn't occur)

Experiments

Linear Regression

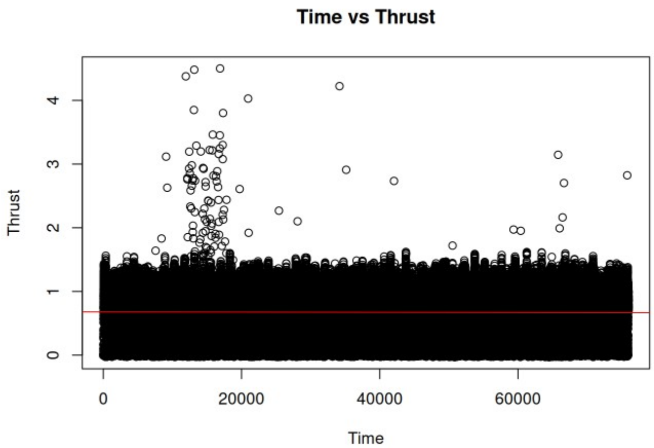
Preparing the data set for linear regression involved taking all of the csv files in the training data set and compiling them into one large file. A sample of 10% of this data set was then generated and cleaned. Cleaning simply involved unclasssing the time so that it was in second form instead of day-month-year form. Linear regression was only done with time on the x axis and thrust on the y axis so that performance could be predicted over time.

Experiment 1



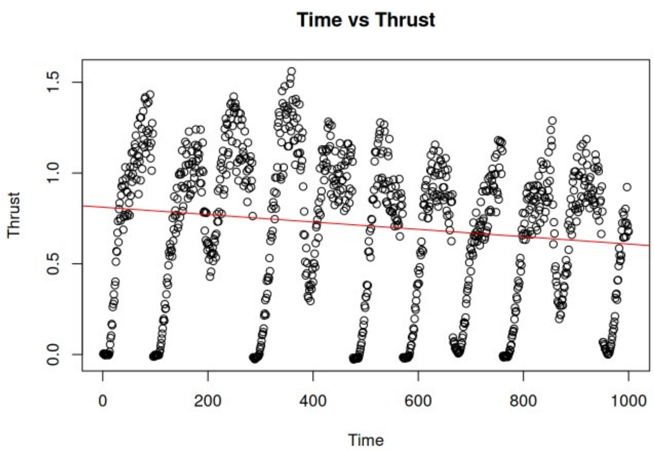
The first experiment was a failure, as it generated an abnormal graph. The first issue here is that time in this data set is based upon the amount of seconds it has been since 1970, rather than the amount of seconds since measurements began. Second of all, multiple thrusters could run at different times, and there were different types of tests occurring, producing widely variable data. To account for this, new data was set up. This time, two different test runs would be analyzed (only two separate csv files) and cleaned. The process was much simpler this time. Data was imported, the time column was changed to start at one and continue in a numeric sequence until the end of the data set(just a note, time is in centiseconds, not seconds), and all zeros were removed from the ton column (engine on flag) to only consider data where the engine was on.

Experiment 2



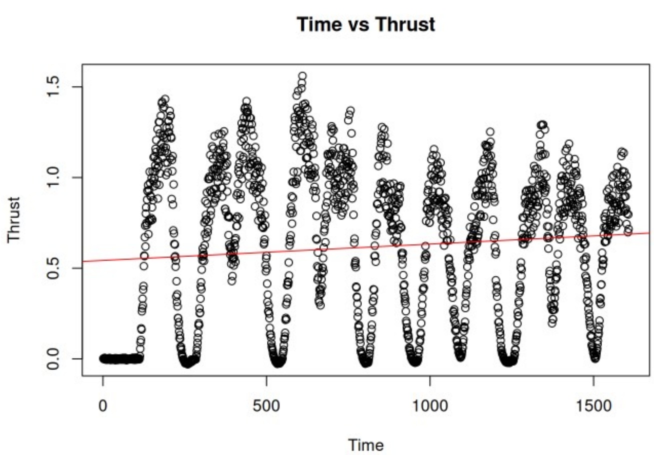
Results look more promising than experiment 2, but data is still too muddled to detect anything of value.

Experiment 3



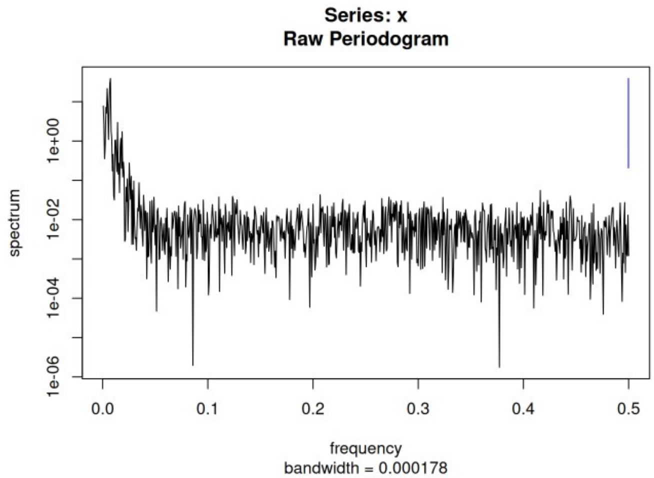
In this experiment, only the first ~1500 points were considered, and an interesting pattern can be seen. Based upon the plot, this test seems to be a sort of cycle test. Including thruster off data could give more insight.

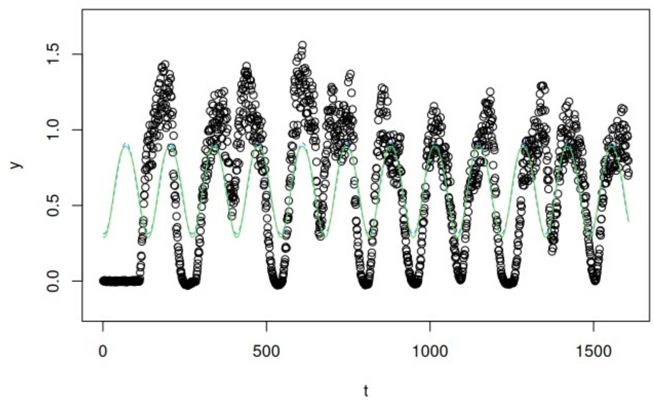
Experiment 4



Adding back the data points where the thruster was off reveals a harmonic pattern, but this linear model is not very good at predicting this model, it can just generally state that thrust is increasing relative to time in this segment.

Experiment 5





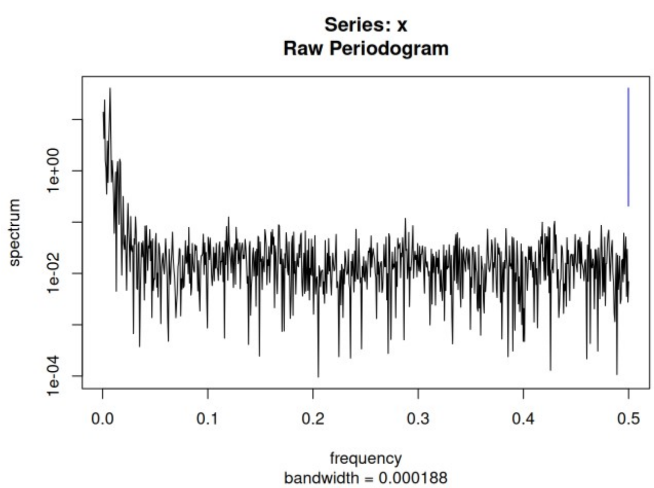
This experiment was an attempt at using a harmonic fit paired with linear regression in R. It was based upon the following trigonometric identity:

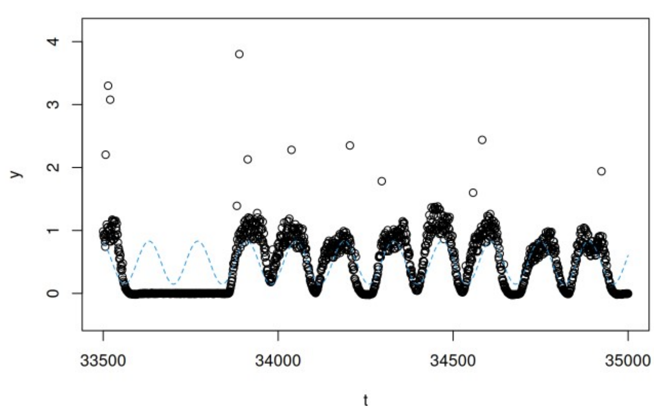
For a general sine wave with amplitude A and phase φ, Asin(x+φ) = asinx+bcosx

Optimal frequency was calculated by maximizing the graph on the left, and this frequency was inputted to the above identity, creating a harmonic fit for the graph. A few conclusions can be drawn here. First, our model does not calculate high thrust well, as the peaks of the harmonic function are cluttered and noisy, while lower-middle sections fit better. Second, our model does not fit the initial half of the data as well as it does the second half. This is likely because the thruster has more variability immediately after ignition - it needs to warm up before outputting reproducible results.

Credits to https://stats.stackexchange.com/questions/60500/how-to-find-a-good-fit-for-semi-sinusoidal-model-in-r (https://stats.stackexchange.com/questions/60500/how-to-find-a-good-fit-for-semi-sinusoidal-model-in-r) and https://stats.stackexchange.com/questions/60994/fit-a-sinusoidal-term-to-data?rq=1 (https://stats.stackexchange.com/questions/60994/fit-a-sinusoidal-term-to-data?rq=1) for information on the harmonic model

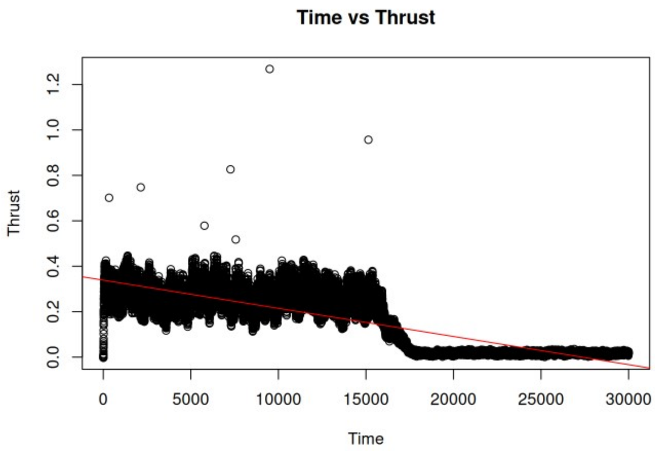
Experiment 6





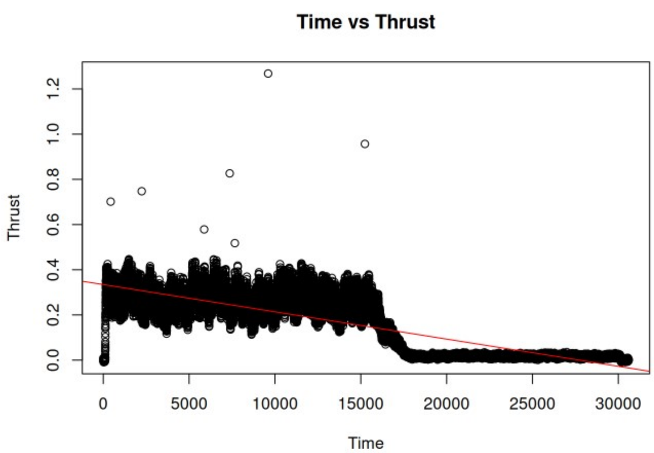
This experiment uses a 1500 data point portion of data from the middle of the data set, when the thruster has already been in use for some time. It uses the same model as experiment 5, but the fit is much much better. This proves that our harmonic model fits very well when the thruster has already had some ontime. With this model, normal thruster performance over time can now be predicted.

Experiment 7



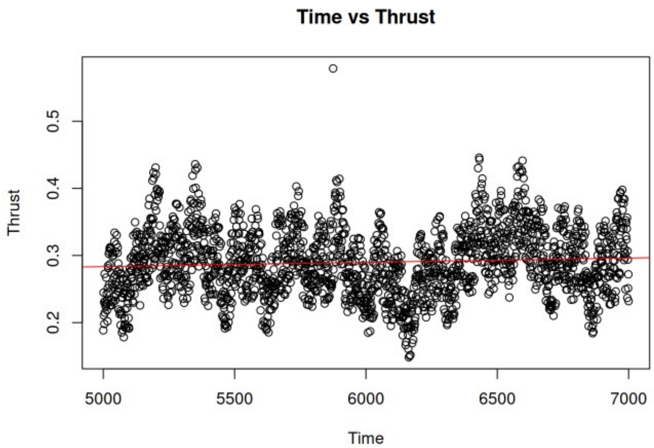
Experiment 7 takes a look at the second data set. This data set generally shows a decrease over time in thrust due to a stark decrease in overall thrust at about 15000 centiseconds.

Experiment 8



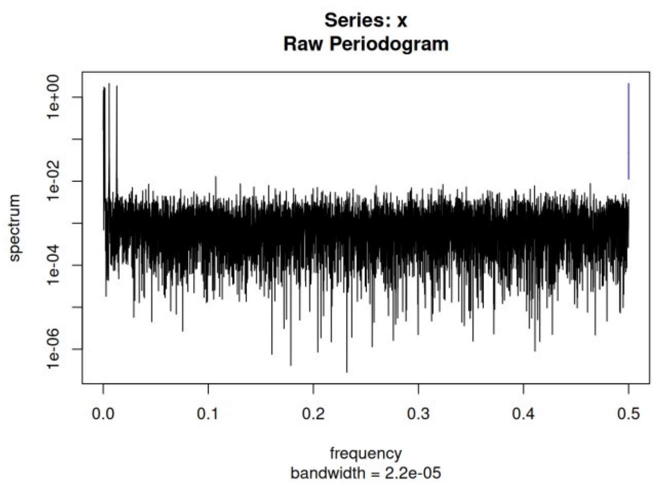
Experiment 8 adds back in engine off time, however there is no change. This indicates that this test involved the engine being set to an active state indefinitely, and some sort of failure occured at around 15000 centiseconds. This is likely due to vapor lock.

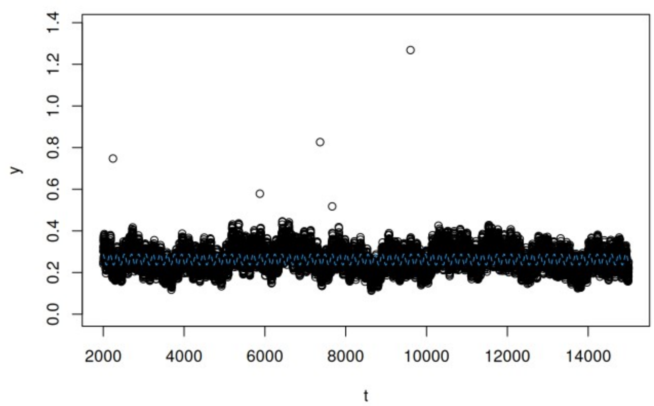
Experiment 9



Experiment 9 involved taking a closer look at a segment of the data where performance was normal, but not too much could be gathered here. The data looks vaguely sinusoidal, however.

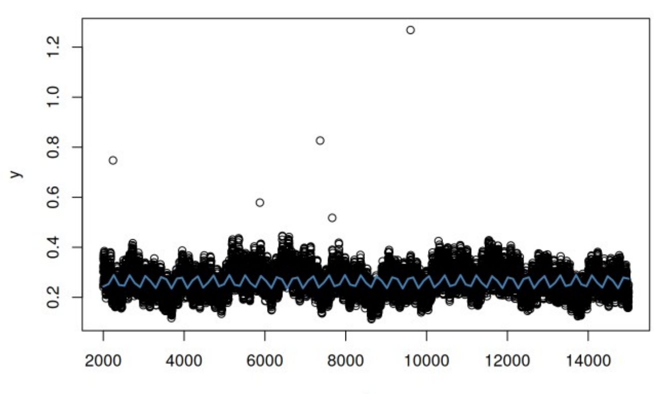
Experiment 10

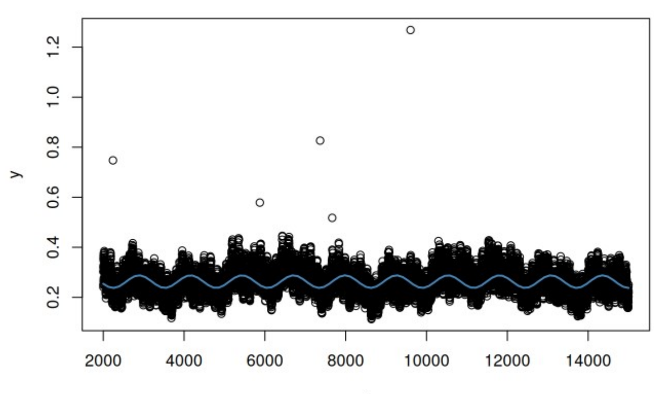




In experiment 10, the entire chunk of normally performing data was fit with the same harmonic model from earlier, but it was unsuccessful this time. However, the periodogram on the left shows multiple frequencies which fit the data well, so perhaps the largest one was not correct for this data.

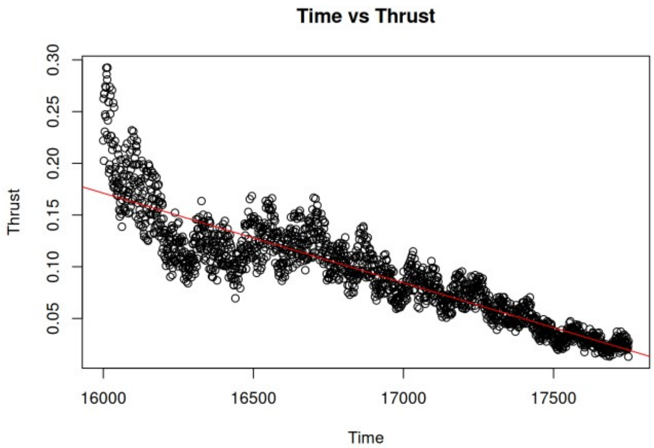
Experiment 11





Experiment 11 uses a different method for harmonic fitting, first starting with a fast fourier transform. This transform could be cleaned to detect the true best frequency, which was an omega value for the equation $y(t)=A \cdot \sin(\omega t+\phi)+C$. This model does not use the same identity from earlier, just the general form of a sine wave. Initially, this experiment outputted the result on the right, but upon reorganizing data, it outputted the result on the left. This is likely because some cleaning step was accidentally removed, but this is not an issue. Brute forcing omega values into the $y(t)=A \cdot \sin(\omega t+\phi)+C$ equation yielded a model that generally fit the curve. As a proof of concept, this model works for detecting normal prolonged performance, and the fast fourier transform can be refined to produce a more precise model.

Experiment 12



This final linear regression experiment actually makes use of the linear part of linear regression. A linear model was fit to the portion of the data that experienced vapor lock, and the resultant line can be used to predict the general effects of vapor lock and the approximate decrease in performance over time. A more precise model could potentially be generated by fitting with sinx/x and rotating it.

Association Rule Mining

Association rule mining only used the metadata file, as good rules were not likely to be found in the test and train data sets. To prepare the data, the uid, filename, test_id, and sn columns were removed because they were simply identification columns, and were not necessary for ARM. The vl3 column was also removed because it was negative across all columns, so it would not generate good rules. cumulated_throughput, cumulated_on_time, and cumulated_pulses were then bucketed, and all the data was made transactional.

Full association rules are not shown, and instead interesting rules are described, so that this section does not take up too much space.

Experiment 1

Experiment 1 had the following parameters:

- support: 0.1
- confidence: 0.5
- lift: 4

Experiment 2

Experiment 2 had the following parameters:

- support: 0.1

- confidence: 0.5
- lift: 6

Lift was increased because there were too many association rules in experiment 1, so a higher lift would provide stronger rules and narrow down the data set.

Experiment 3

Experiment 3 had the following parameters:

- support: 0.05
- confidence: 1
- lift: 2
- max length: 2

Confidence was increased, lift was decreased, and support was decreased to provide better and more interesting rules, because the results from the previous experiment were fairly obvious. For example, Longer cumulative ontime was equivalent to more cumulative pulses. A max length was added to prevent redundant rules. For example, longer cumulative ontime was related to cumulative pulses 7 different times, but the lhs contained additional rules in 6 instances.

Ultimately, the only real interesting rule here that was also strong was this one found in experiment three:

{test_mode=onmod} => {vl2=False}

For whatever reason, the onmod test mode seems to have less severe vapor lock levels than other tests. This could be basis for further investigation. Implications of this are further discussed in the conclusion.

Support Vector Machines

Support vector machines could not utilize the combined data for two reasons. First of all, any number of combined data sets, even just five, were too large to process through the SVM algorithm. Large amounts of data took an unrealistic amount of time to process, so only two data sets were selected. Secondly, the large amount of variation and issues with compiling the data set were already seen in experiment 1 of the Linear Regression model, so basing a predictive algorithm off of a data set with these issues would not work well. To prepare the data, seconds and ton columns were removed, the anomaly column was changed to binary(instead of containing various anomaly flags) and the binary column was represented as a factor.

Experiment 1

Experiment 1 had the following parameters:

- C: 1
- cross: 2

and the following output:

```
##                   svmPred.1...
## svmTrainData...4.     1
##                  0 19019
##                  1   377
```

The confusion matrix above only contained one column, which indicates that the SVM algorithm only predicted one binary value. This usually means that the margin of error is too high (C is too low.) To attempt to resolve this, the next experiment increases C to 1000

Experiment 2

Experiment 2 had the following parameters:

- C: 1000
- cross: 2

and the following output:

```
##                  svmPred.1...
## svmTrainData...4.     1
##                 0 19019
##                 1   377
```

Because the same one column confusion matrix is being output, I thought that maybe decreasing cross could result in an actual predictive algorithm.

Experiment 3

Experiment 3 had the following parameters:

- C: 0.01
- cross: 2

and the following output:

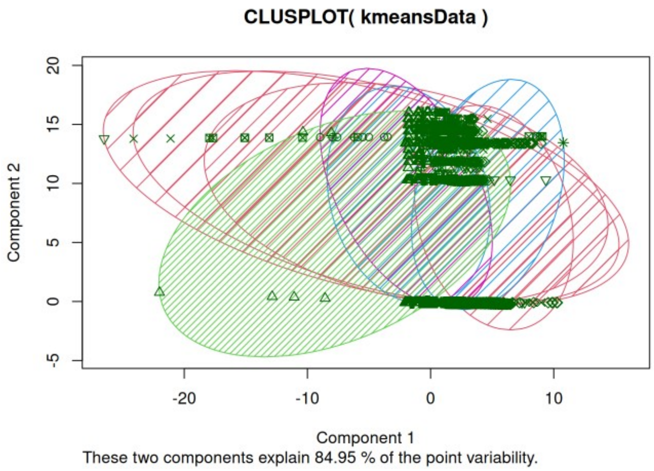
```
##                  svmPred.1...
## svmTrainData...4.     1
##                 0 19019
##                 1   377
```

It seems like SVM just isn't working. After some research, we discovered that SVM algorithms display this result when data isn't compatible, so no prediction can be made. It looks like the high variability between tests is not suitable for a predictive algorithm like support vector machines.

KMeans Clustering

KMeans benefits from larger data sets, so the combined data set was used here. The same initial process was used, and a random sample of 10% of the data set was generated. Once again, the time and ton columns were removed, as they were not helpful clustering data. Additionally, anomaly column was changed to binary again.

Experiment 1



Experiment 3 had the following parameters:

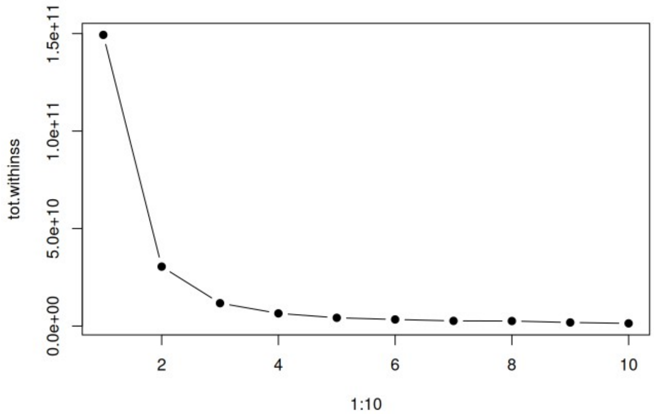
- center: 8
- nstart: 20

and the following output:

```
##
##          0     20     21     22     23     24     25     26     27     28
##   1  69780    208      3      1     20      2      1     28      9     57
##   2 483380   1068      2      3    133     18     14   1187      8    288
##   3  42789     87      2      1     21      1      4     39      2     70
##   4  38702     53      0      3      2      0      3     13      2     44
##   5  26580     45      0      2     17      0      0     56      0     74
##   6  40047    146      0      4      4      2      0     77      1      0
##   7  61715     35      3      1     29      1      6     19     11     92
##   8  43852     53      0      0     14      0      5    198      0     80
##
##         29     30     31
##   1     49      2      9
##   2    287     48     51
##   3     33      2      8
##   4      1      8      0
##   5      3      5      0
##   6      2      5      0
##   7     44      6     13
##   8      7      3      0
```

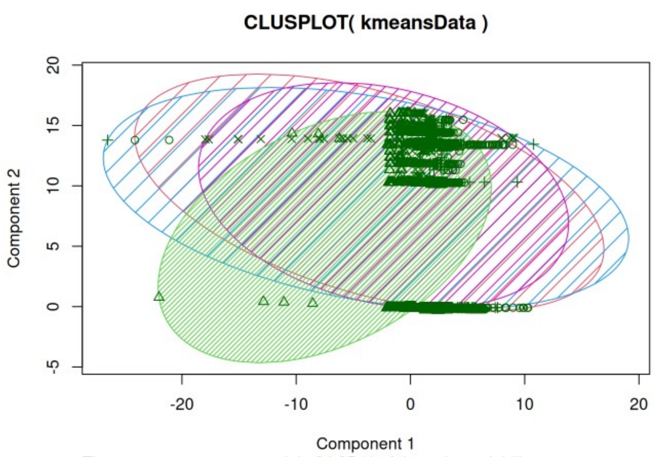
The intital clusters were not too good, but the data appeared promising for clustering, so perhaps the number of clusters were the issue.

Experiment 2



This experiment utilized a for loop and the tot.withinss function for 10 center values to determine the best one. The elbow method was used, and the best value appears to be 4

Experiment 3



These two components explain 84.95 % of the point variability.

Experiment 3 had the following parameters:

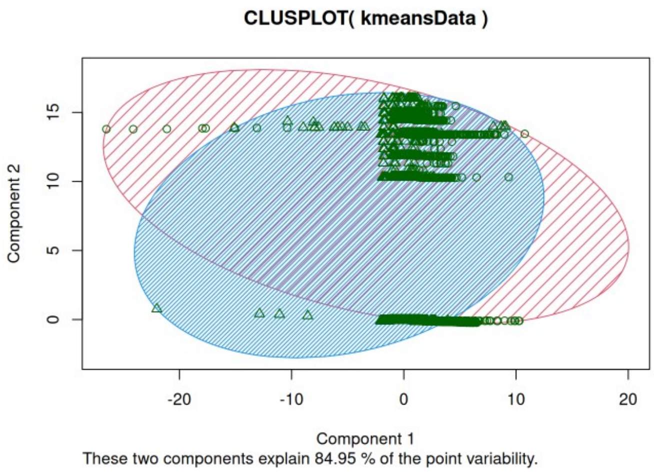
- center: 4
- nstart: 20

and the following output:

```
##
##          0     20     21     22     23     24     25     26     27     28
##    1 169762    289      6      2     60      3     12    244     18    227
##    2 529195   1161      4      4    155     19     18   1227     12    360
##    3 107888    245      0      9     25      2      3    146      3    118
##
##         29     30     31
##    1     90     11     22
##    2    329     50     59
##    3      7     18      0
```

Using the new center, the clusters still don't work well, so in the next experiment we two things. First, we change the center to the amount of obvious clusters we can see. Second, we change the nstart value to 1000. Kmeans is highly dependent on a high number of iterations to recieve good clustering data.

Experiment 4



Experiment 3 had the following parameters:

- center: 1
- nstart: 1000

and the following output:

```
##
##          0     20     21     22     23     24     25     26     27     28
##    1 186607    318      2      9     61      3     10    353      9    274
##    2 620238   1377      8      6    179     21     23   1264     24    431
##
##         29     30     31
##    1     48     22      0
##    2    378     57     81
```

Even with the adjustments, kmeans doesn't appear to be working, however this may be a problem with the algorithm rather than the data itself, because there are very clear clusters present that it does not detect. My peers also experienced similar negative results, so perhaps a different clustering method could have been used. If the data did work as expected, we believe that the upper grouping of data consists of instances without vapor lock, and the lower contains instances with vapor lock.

Conclusion

Ultimately, linear regression produced the best results out of all four algorithms. There are a few particularly important plots from the linear regression experiments. More specifically, the plots from experiments 6, 11, and 12 are very important for anomaly prediction and the effects of vapor lock. Experiments 6 and 11 demonstrate normal performance, and experiment 12 generally demonstrates the effects of vapor lock. All three of these models fit their situations very well, so they can be applied elsewhere. Inflight, using these algorithms, a computer could detect shifts in normal performance using experiments 6 and 11 and then determine if vapor lock is occuring by using experiment 12. Aditionally, thruster aging over time could be quantified by comparing the normal harmonic fits to actual thruster data and noting decreases in thrust performance.

While linear regression data was successful, association rule mining, support vector machines, and kmeans did not yield results that were too fruitful. Association rule mining did not fail, however many of its rules were predictable, and many interesting ones did not have a strong correlation. The correlation between offmod testing and decreased vapor lock seemed interesting, because one type of testing leading to better performance could potentially lead to applying the conditions of that test into actual spacecraft. However, offmod testing is just testing where the engine is perpetually given an off command. The purpose of this test is to make sure the on/off switch to the thruster valves work properly. No thrust means obviously means no heating, which in turn means no vapor lock. Therefore, the only really strong correlations in association rule mining were those that were obvious and uninsightful. Support vector machines and kmeans both ended up failing completely, but likely for different reasons. The high variability between each data set in led to the failure of SVM, but kmeans likely failed due to the algorithm itself. Very clear clusters were visible, however the algorithm could not detect them. It is possible that a different clustering program would work better. It is unfortunate that kmeans did not work, because successful kmeans data could be a more concrete marker of thruster age. It is presumed that the two visible clusters in kmeans were instances that had vapor lock and instances that didn't. By comparing the ratio of the clusters, it would be possible to get a more concrete quantification of thruster age and remaining lifetime. Currently, the number of times a thruster is activated since its creation is used to quantify its age, however this isn't always a good estimate. Sometimes thrusters are kept on for longer or commanded to thrust at variable levels. This pulse-to-pulse variation results in varying levels of degradation on different thrusters with the same number of cumulative pulses. By running health tests and comparing the ratio of vapor locked instances to non vapor locked instances, degradation can be measured better. This would avoid situations where a thruster gives out earlier than expected, and allows for maximum thruster lifetime utilization. By refining kmeans or using a different clustering algorithm, this advancement could be possible.

Although many of the models used ultimately ended in failure, the successes truly outweigh the failures. With the predictive models from linear regression, and the potential lifetime quantification applications of a perfected kmeans clustering algorithm, many advancements could be made in spacecraft thruster testing.