**This report is for the progress period ending on 12/03/2019.**

*Report is based on the guidelines under the "Report" section provided in the rubric.*

**Ioannis Batsios**

*Created a multiple linear regression model to check engine manifold temperature. (~5 hours)*

After seeing how ambient air temperature affected the Turbospeed, I was interested in seeing if I could use outside temperature and other engine temperatures to predict the temperature of the manifold. I thought this could be useful in determining whether or not a part was going out. My first model was created by just "doing it" and seeing what I came up with. I received a score of .734 percent accurate, which is pretty good. But I wanted to try and make a better one so I tried doing backward elimination. Unfortunately, my p-values are so low, I'm not sure it would improve my model anymore to attempt it. Need to do more research to see if I should modify my model better.

**Bill Downs**

*Continuation of evaluating the APU unit with Machine Learning (~7 hours)*

Classifier: Logistic regression

I utilized the logistic regression classification to predict if the APU unit is charged at a particular time of the day. First, I kept my linear regression scatter plot as it may be insightful while discussion my machine learning outcome. Second, I had to create a new column that calculated a binary value as a 1 if the APU is charged (>14.0 volts) else 0 it is not fully charged. I had to check electrical engineers because a lot of different values exist on the internet of "max charged voltage", which the Engineers were happy with values considered charged at 14.0 or greater.

Next is where the most time was spent, as our sensors take measurements every 10th of a second the I assumed the large amount of data would be quite insightful. I was wrong. after using the train test split method and fitting/training my data the scores were very unintuitive, so I decided to resample, as i did before at 300 seconds. This decreased the total amount of data and my training score was a little less than my testing score. But only by a very small amount. I then calculated a prediction of probability and was somewhat happy with this but I had to refer back to the time series from last project to see a visual correlation. I also calculated a confusion matrix to see where my classifier stood. The True positives and True negatives were quite high which was but it seemed about a quarter of the results were either false positive or false negative values.

Conclusion, this was a cool project and I think there are a lot of was that Volvo can further drill down on this data to come up with well-rounded assumptions of how well the APU system is performing with a particular driver and maybe if the system is over or under kill for a given application of that vehicle.

**Wahab Ehsan**

*Time to Ideal Temperature with Ambient Temperature (~7 hours)*

Worked on finding the time it takes to get to ideal temperature with outside temperature taken into consideration. The data for time wasn’t given. Had to find out the Ideal temperature for the oil. Found the average temperature of oil for each day and used it as the Ideal temperature. Then made function to find difference of seconds between two indexes. Was able to check from beginning to whenever the temperature gets to ideal temperature and found the seconds it took to get there. Made sure the temperature difference between beginning to ideal had at least 30 degrees C difference and the session the truck is been on is at least for 600 seconds. Then was able to make X equals to the outside temperature and Y as the Seconds it took to get to ideal temperature. Then plotted on graph and saw positive correlation, which was unexpected. Make least square line for predictions and was able to predict few values.

**James Polk**

*Time Series Analysis on Truck 1 (~5 hours)*

After reviewing and cleaning sensor logs, I saw that the data was spotty for most sensors. To overcome this I performed Time Series Analysis on only those sensors with no NaN values. Furthermore, the data is sampled every 100 milliseconds, causing a lot of noise when plotted.

I decided to resample my data to every 1 minute using mean. This provided at least some fix for the noise, but not for the data gaps. Unfortunately, resampling any higher resulted in a loss of accuracy, and other than making pretty graphs, did not contribute in any way to analysis. After resampling, I performed time series analysis using exponentially weighted moving averages on the CPU Load, Outside Air Temperature, Temperature of Air Entering Vehicle, Vehicle Weight, and Driver Requested Torque. When trying to run Dicky-Fuller tests on the data, I was met with errors due to gaps in the data. The same result happened when trying to run decomposition analysis.

**Christopher Thacker**

*Truck 2 GPS Speed Corrections (~4 hours)*

As we learned from Project Progress two (2), Truck 2 has some faulty data for its GPS Speed

component. This is a bigger issue for Volvo than us, but my questions from this was "how well

would this component work if it was completely functional and what would its more correct

measurements look like?" Thankfully, Truck 1's components appear to be working perfectly fine.

This means I was able to use Truck 1's data as a model to predict what Truck 2's GPS Speed values *should* be. To solve this issue, I took the difference of means for Truck 1's GPS Speed and

Wheel-Based Speeds. Using that value, I iterated through Truck 2's Wheel-Based Speed values and added that mean to each value and stored the result into a separate column. These calculated values were the "predicted" GPS Speed values for Truck 2. I created visualizations and calculated their values to better analyze the results, and those results appear to be fairly consistent with the correctly measured values for Truck 1.

*Github Repository Management (~3 hours)*

For most of Project Progress Three (3), I have been given the responsibility of managing the GitHub repository in a multitude of ways. This includes updating the home page README, creating, updating, and closing Issues, creating and assigning Labels to those Issues, managing and assigning group members to those Issues, and trying to keep the general repository structure simple and clean.