**Drew Allen**

**CSC375**

**Final Project**

**Predicting Remaining Useful Life of Turbofan Engines**

Accurately predicting the remaining useful life of jet aircraft turbofan engines has recently become a popular challenge amongst data scientists. Using a software from NASA, called C-MAPSS (Commercial Modular Aero-Propulsion System Simulation), the functionality and life span of a turbofan engine can be simulated. This simulation mimics a flight’s ascent to 35,000 feet and its descent back to sea level (catalog.data.gov, 2020). Data collected from these simulations is being used to build machine learning models to predict the remaining useful life of real-world turbofan engines. A single instance of a simulated flight is referred to as a cycle. While the provided data are multivariate time series, I will attempt to make accurate RUL predictions using standard Scikit Learn regression models within a Jupyter Notebook.

For this project, I will be using the first provided set titled “FD001”. This set contains 100 engines, meant to represent the aircraft of a single fleet. Some of the information provided in the data set includes the engine number, the current cycle, readings from three operational settings, and readings from 21 onboard sensors. The data is already split into training, test, and test\_rul (test targets) sets. The training set involves the full simulation cycle for each engine, while the test data stops some time before each engine fails.

Taking a look at each engine’s total life cycles we find that the mean value of total cycles for training set engines is about 205. Visual inspection of the training engines’ max cycle values shows that several high-end outliers exist, including a couple of points near or beyond 350 cycles.

I created an “RUL” column for the training data to act as targets for our models. For each row, I subtracted that row’s cycle from its engine’s final cycle value. This way, we know exactly how many RUL cycles remain for each row in the training set. I also created a new test set from the original in order to match the shape of 100 provided test labels. I did this by creating a new dataframe and appending the final provided test point for each engine.

Next, the setting & sensor values were described & visualized using pandas in python. Here, we see that several of the sensor values have near constant mean/min/max values and/or very small standard deviations. These sensors will be dropped from our train/test feature sets. Of the 21 sensors, I chose to drop sensors 1, 5, 6, 10, 16, 18, & 19.

Through trial and error, I found that using a sample size of 7,500 (of the original 20,631) for training our model provided better results. This random sampling is possible because I will not be training time series models for this project, so samples don’t have to be sequential.

I then created a couple of basic models using our newly processed train / test sets: a linear regression model and a RandomForestRegressor model. Unfortunately, neither of these models performed particularly well on several metrics. The linear regression model scored .66 on train data and .43 on test data and had a mean absolute error of about 26. The RandomForest model scored .97 on train data and .58 on test data and had an MAE of about 20. Given that this data is officially a time series set and the relative complexity of the data, an MAE of 20 might seem rather impressive. However, a key observation of the sensor readings provides an insight into adjusting our model to greatly improve scores. Line plots for both models show that our predicted values are missing the true values by a good margin.

After some research I found that other C-MAPSS model builders noted that sensor readings remained fairly constant for a significant period of time in an engine’s life. So, using four sensor readings, I created scatterplots with regression lines to visualize the change in those readings over the life cycles of all engines in our original training set. These plots show that the sensor readings really did not begin to deviate until about 150 to 100 RUL cycles remaining. This meant that a huge portion of our training data contained points from early in an engine’s life when these values were constant. If we clip our training labels’ upper bound to a value around when the sensor readings begin to truly change, we can improve our models’ performance.

With my training labels now clipped at 125 RUL on the upper bound, I trained a new linear regression model and a new RandomForest model. Now the linear regression model scored .48 on train data and .7 on test data with an MAE of 17.7. The RandomForest model now showed relatively impressive results, scoring .59 on train data and .81 on test data with an MAE of 13.86. Line plots for both of these new models show our predicted values are more closely following the true values, especially for the RandomForest model.

I believe these new models scored worse on training data because of the upper bound clipping we have performed. The training features now contain many data points with the same RUL and are likely making it difficult for the model to generalize. The test data is unclipped and inherently comes from a cycle closer to the point of failure, where sensor readings are likely more correlative.

While these standard regression models may not be the first choice by data scientists for modeling C-MAPSS data, the RandomForest model shows that with some simple preprocessing, you can prove that definite correlation exists between certain sensor readings and remaining useful life of turbofan engines. It emphasizes a focused period in an engine’s life that data scientists can watch for relevant changes in sensor readings in order to better understand failure points of these engines.