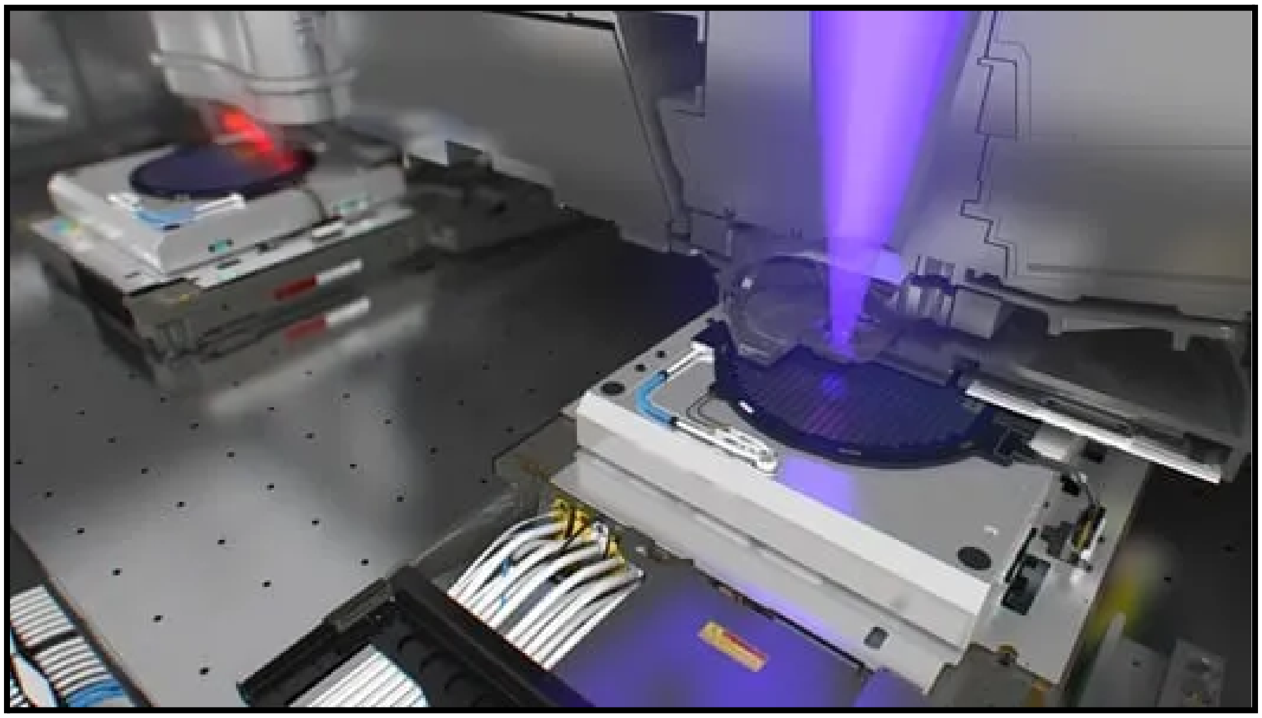
IUPUI DATATHON 2023-2024
SEMICONDUCTOR MANUFACTURING SENSOR ANALYSIS & CLASSIFICATION

—

Team: Kyle McCrocklin (kmccrock@iu.edu)



PROJECT DESCRIPTION

Prior to joining the Computational Data Science master's program at IUPUI, I worked for 5 years in the manufacturing industry and I believe there is huge potential for applying data science methodologies to manufacturing problems. For this Datathon I will be working with the UCI SECOM dataset which came from a semiconductor manufacturing process.

I am approaching this project as if I were hired by SECOM to provide insights to the engineers working on this semiconductor line. I will analyze the data in a clearly explainable way, then attempt to build a classification algorithm to predict if a produced unit will pass or fail quality control.

DATA

The dataset contains 590 features and 1567 observations. Each feature, labeled V1-V590, represents a sensor or measurement point somewhere in the semiconductor manufacturing process. No information about the measurements is provided aside from their numerical output.

Each observation represents a single produced unit and is labeled +1 for failing QC and -1 for passing along with a timestamp. There are 104 fails and 1463 passes in the dataset.

RESULTS

Part 1: Data Exploration and Cleaning

The mean, median, standard deviation, and interquartile range (IQR) were calculated for each of the 590 features and plotted on histograms. It is clear that the in-process measurements vary dramatically in scale and central tendency. The data were subsequently standardized to their Z-score.

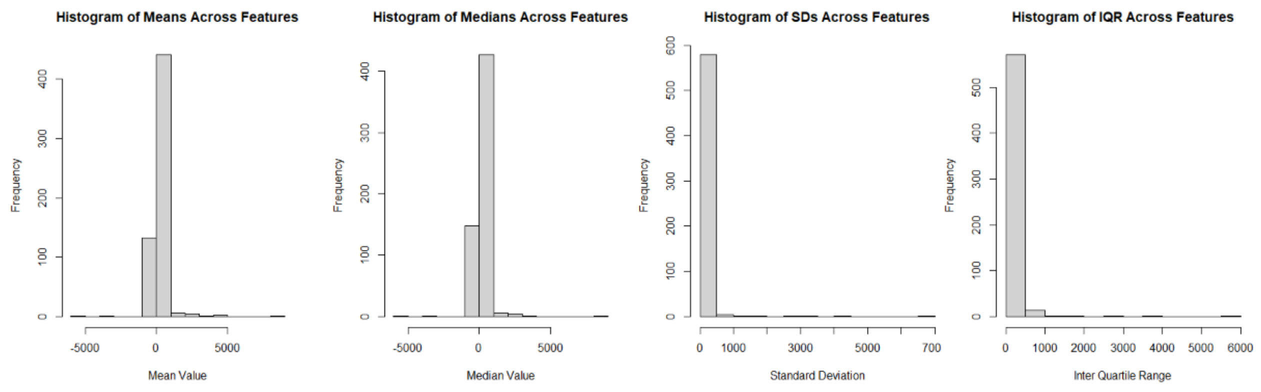


Figure 1: *Histograms of Feature Statistics*

The data were cleaned by removing features with no variation (IQR=0). Features missing more than 5% of their data were also removed. This left 412 features. Samples that are still missing data will be eliminated.

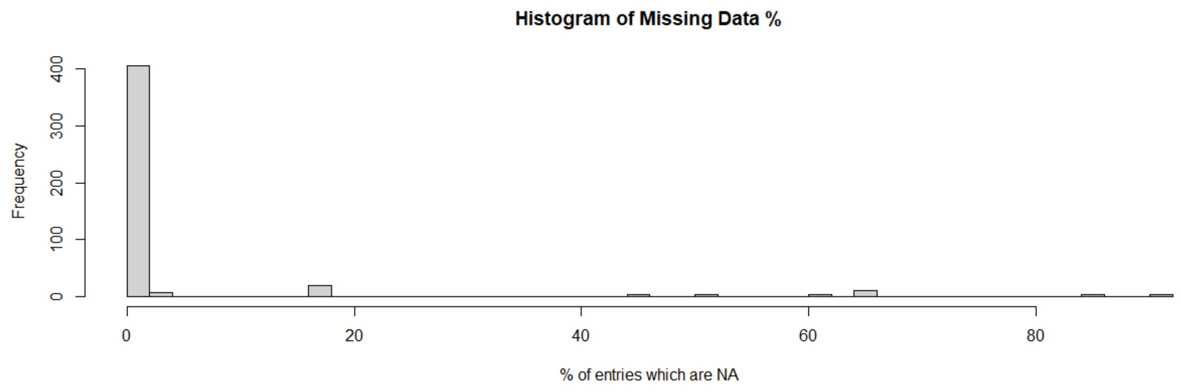


Figure 2: *Histogram of Missing Data %*

The classification labels were then added to the data and analyzed. The data are heavily imbalanced towards negative examples (passing QC). Only 104 of 1567 of the examples (about 7%) are the positive case (failing QC).

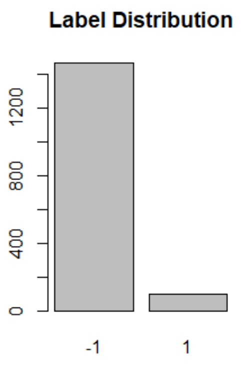


Figure 3: *Bar Plot of Imbalanced Label Distribution*

A temporal analysis did not reveal any clear patterns or trends over time (Figure 4). Therefore the timestamp was disregarded as a useful feature.

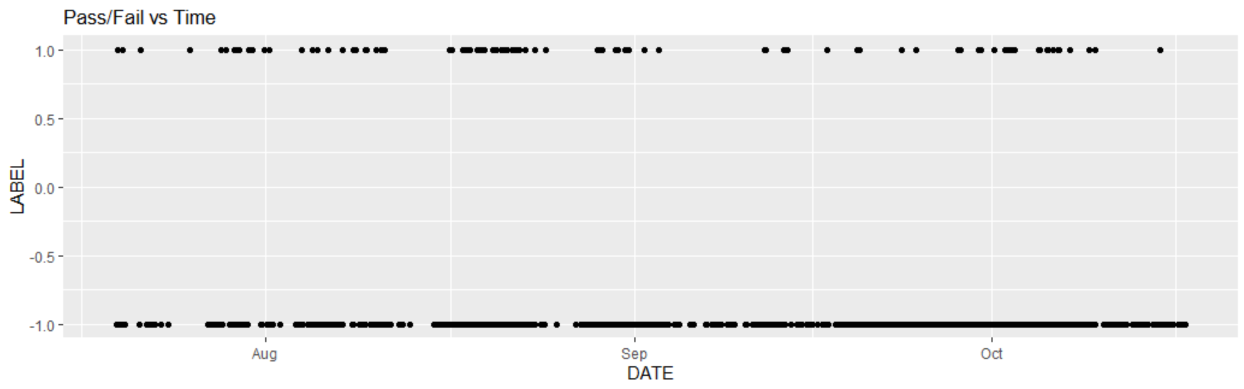


Figure 4: *Plot of Pass/Fail Over Time*

After exploring and cleaning the data, 412 features and 1393 examples remain. This dataset will be used for the classification model in Part 3.

Part 2: Feature Ranking

Feature ranking was performed using two variations of a side-by-side boxplot analysis. In this type of graphical analysis, boxplots for the passing and failing cases of a single feature are displayed next to each other. The analyst looks for trends in the first, second, and third quartiles between the two cases. Due to the large number of features, numerical algorithms were applied in place of visual analysis.

The first ranking variation was based on the magnitude of unidirectional shift in the three quartiles. This would indicate that the feature being investigated is centered around a different value for units that fail QC compared to those that pass QC. (Figure 5, next page)

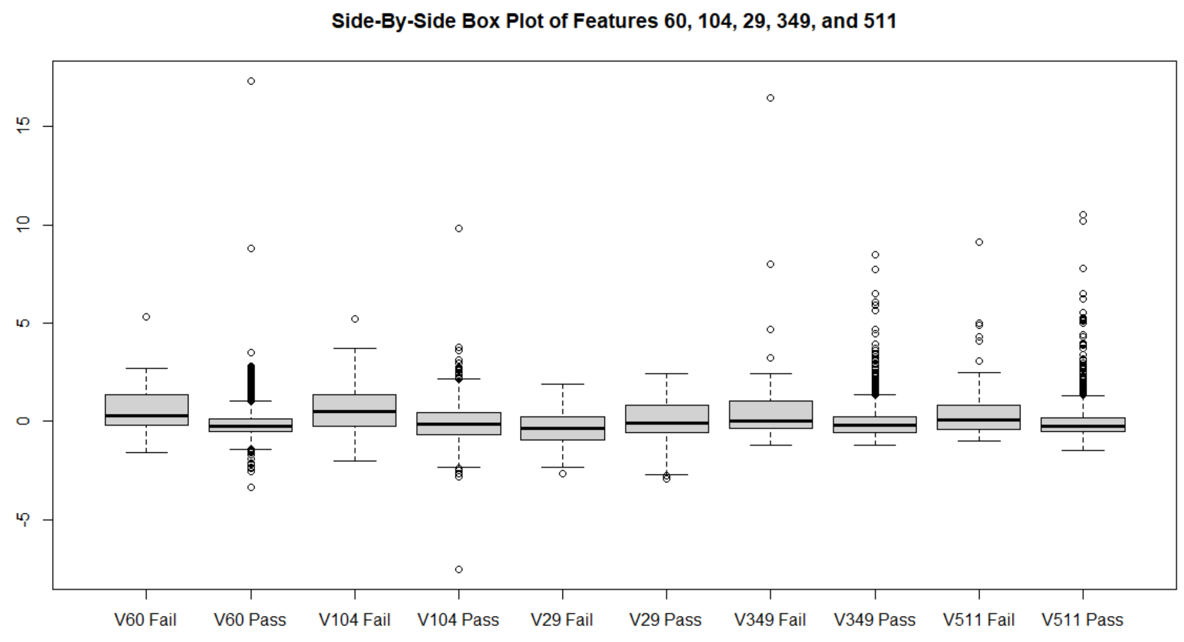


Figure 5: *Top 5 Features as Ranked by Change in Central Tendency*

The top 5 features in terms of change in central tendency do not show much separability between the pass and fail cases. Still, these sensors may be of special interest to the SECOM engineers. See “FeatureRankingCentralTendencyBased.csv” for the full feature ranking.

A second side-by-side boxplot analysis based on variance was also performed. In this case, the features are ranked based on how much the interquartile range (IQR) increases from pass case to fail case. This would indicate that the feature being investigated exhibits both higher and lower values when a unit fails QC compared to when it passes QC.

The top 5 features in terms of increased IQR in the fail case are shown in Figure 6 on the next page. Again, the distinction between passing and failing cases is not very significant. Still, these sensors with the largest change in IQR may be of special interest to the SECOM engineers. See “FeatureRankingVarianceBased.csv” for the full feature ranking.

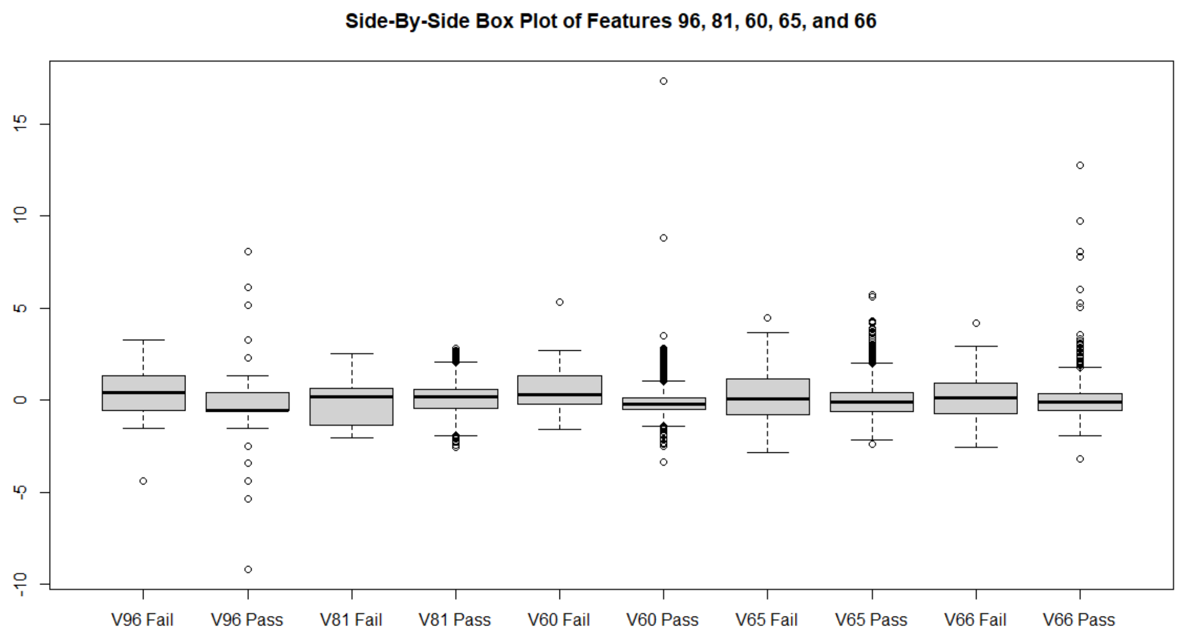


Figure 6: *Top 5 Features as Ranked by Increase in IQR*

The main insight from these two side-by-side boxplot analyses is that there is not a clear distinction between the examples of units which passed QC and those which did not pass QC. There are no individual parts of the manufacturing process being monitored which are clear indicators of whether a produced unit will pass or fail QC.

It is clear that the data are not easily separable and will be very challenging to model with a classification algorithm.

An actionable outcome of the feature ranking outlined above is that engineers may be able to reduce the number of units which do not pass QC by focusing their attention on the parts of the manufacturing process corresponding to those top ranked sensors.

Part 3: Building a Classification Model

The cleaned data were split with 70% being used to train classification models and 30% of the data reserved to test the models. The models used were logistic regression, decision tree, random forest, XGBoost, RUSBoost, and local outlier factor. To address the class imbalance, both undersampling and balanced classifiers were tested. A more detailed description can be found in the Methods section of this report.

None of the models tested were able to classify the test data with acceptable precision and recall for both the pass and fail cases. The best performing model was RUSBoost which is an ensemble method designed for data with imbalanced classes. It randomly undersamples the training data at each boosting iteration. This model achieved a recall of 81% for passing examples, 57% for failing examples, and overall accuracy of 79%.

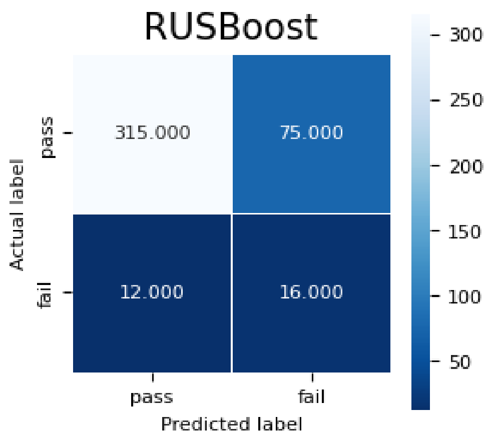


Figure 7: *Confusion Matrix for Top Performing Classifier*

It is concluded that the sensor data being collected is not adequate for a model to learn a robust classification algorithm. The data currently being collected is essentially noise as far as classification algorithms are concerned.

Some actionable recommendations from this classification analysis are to reconfigure the sensor array or improve the quality of measurements. Robust prediction of QC outcome may very well be possible with more sensitive, less noisy, and better located in-process measurements. The areas of the process to focus on are those identified in the feature ranking portion of this report. That is where at least some measurable difference between cases occurs.

METHODS

Parts 1 & 2: R

Parts 1 and 2 of this analysis were performed using R. The methods used are straightforward and make use of the R base package along with the "dplyr" and "ggplot2" libraries. The source code is provided along with this report.

Part 3: Python

The classification portion of this analysis was done in Python. The cleaned data from Part 1 was imported and split into test/train datasets. The classes were very imbalanced with 1294 examples passing QC and only 99 examples failing QC. Initially a logistic regression model, a decision tree, a random forest and XGBoost were trained on the data.

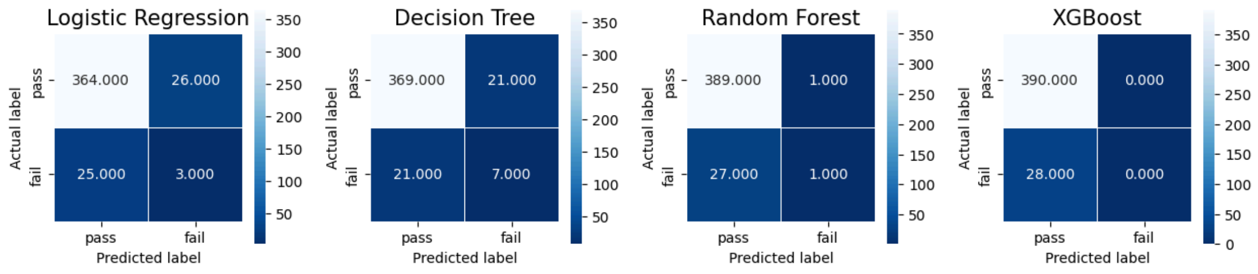


Figure 8: *Confusion Matrices for Models on Full Dataset*

All four models had great accuracy - 88%, 90%, 93%, and 93% respectively - but they were essentially only predicting the pass case which makes up 93% of the test data. This means they had terrible recall for the fail case - 11%, 25%, 4%, and 0%.

To combat the class imbalance, the pass case was randomly undersampled to contain only 99 passing examples, equal to the number of failing examples. Logistic regression, a decision tree, and XGBoost were applied to the undersampled data.

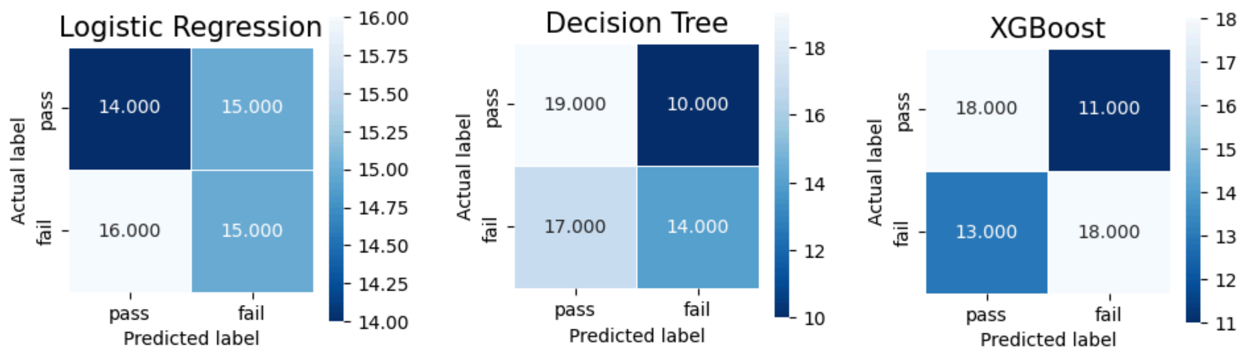


Figure 9: *Confusion Matrices for Models on Randomly Undersampled Dataset*

Accuracy is a fine metric for these models on undersampled data and it hovered around 50% with XGBoost performing slightly better than random classification at 60%. Though undersampling to balance the classes did resolve the single class prediction problem, it did not yield usable models.

Another method for dealing with imbalanced data is to use a weighted/balanced classification algorithm. These algorithms take the class imbalance into account by adjusting the weights of the loss function during training. This allows the full dataset to be used instead of throwing away valuable training examples. Weighted XGBoost, balanced random forest, RUSBoost, and local outlier factor models were trained on the full dataset.

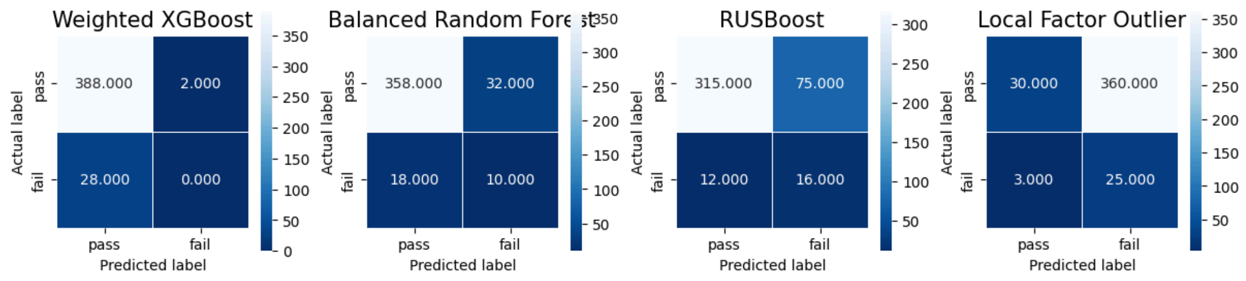


Figure 10: *Confusion Matrices for Weighted/Balanced Models on Full Dataset*

Weighted XGBoost did not perform any better than the normal XGBoost. Artificially high loss for misclassified fails was required to move it away from predicting pass almost every time. RUSBoost had the best performance of any algorithm with a recall of 81% for passing examples, 57% for failing examples, and overall accuracy of 79%.

IUPUI DATATHON 2023-2024
SEMICONDUCTOR MANUFACTURING SENSOR ANALYSIS & CLASSIFICATION

Team: Kyle McCrocklin (kmccrock@iu.edu)

CONCLUSION

The SECOM dataset turned out to be mostly noise. The feature ranking and classification modeling showed that there is little separability between the pass and fail cases. Still, I was able to uncover some insights into which areas of the manufacturing process appear to have the most correlation with whether the produced unit passes or fails QC. This is information that a client could take action. They may benefit from focusing their engineering efforts on these areas of the process. They may also have more success with a classification algorithm if better data is collected from these areas.

While it would have been nice to come up with more flashy insights, and there are many Kaggle submissions which claim >80% classification accuracy/recall or massive feature reduction on this dataset, it is better to honestly conclude that the dataset contains mostly noise, than to mistakenly claim impossible results.

Additionally, this project was good practice for future, real world situations where I may be tasked with making the most of poor quality data.

CITATIONS

McCann,Michael and Johnston,Adrian. (2008). SECOM. UCI Machine Learning Repository. https://doi.org/10.24432/C54305.