


# ***AER850 Project 1***

Course Code / Title	AER850
Semester / Year	4
Instructor	Reza Faieghi, Ph.D.
TA	Hailey Patel
Section	03
Assignment	1
Submission Date	Oct 6 2025
Due Date	Oct 6 2025

Name	Student ID	Signature
Bosco Mak	501104446	

By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: [www.ryerson.casenate/current/pol60.pdf](http://www.ryerson.casenate/current/pol60.pdf).

## Abstract

This report provides a brief overview of the process by which a machine learning model was developed. Using coordinate data for components of an inverter for *FlightMax Fill Motion Simulator* a model was made to predict the maintenance step required given coordinates for a component. It was found the X coordinate had a 75% absolute correlation to the step required, the highest of the three variables. Three models were developed and tested to determine the most effective models. Logistic regression, Random Forest Regression, and Support Vector Machine models were developed, with the latter two proving to be most effective. Both models excelled with high values of accuracy, precision, recall, and f1 score. Stacking these models provided no further benefit as it simply repeated the outputs of RandomForest. The stacked model was finally used to test against external data, which predicted Step 9 for all presented data.

Full console outputs can be found in Appendix A, and a link to the GitHub can be found in Appendix B.

[Link to GitHub](#) (or copy-paste below)

<https://github.com/bmakTMU/AER850/tree/main/Project%201>

# Table of Contents

<b>Abstract</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
<b>2.2 - Data Visualization</b>	<b>3</b>
<b>2.3 - Correlation Analysis</b>	<b>4</b>
<b>2.4 - Classification Model Development/Engineering</b>	<b>5</b>
<b>2.5 - Model Performance Analysis</b>	<b>6</b>
<b>2.6 - Stacked Model Performance</b>	<b>6</b>
<b>Appendix A - Code Console Output</b>	<b>7</b>
2.4 - Model Development	7
Model 1 - Logistic Regression	7
Model 2 - Random Forest	7
Model 3 - SVM	7
2.5 - Model Performance Analysis	8
LogisticRegression Metrics	8
RandomForest Metrics	8
SVM Metrics	9
2.6 - Stacked Model Performance	9
2.7 Model Evaluation	9
<b>Appendix B - GitHub Link</b>	<b>10</b>

## 2.2 - Data Visualization

Figure 1 shows the density distribution of each variable plotted on a histogram. The visualization of the data helps to understand the location of each component in relation to the assembly. The X-coordinates seem to be concentrated at the extremes with fewer components being located in the middle. The Y-coordinates seem to be distributed between four major areas. This also means there is little variation in the data. Conversely, the Z-coordinate varies greatly within its interval, appearing similarly to a normal distribution. The step histogram shows the density of components required per step. As per the figure, it seems steps seven to nine have the greatest number components and coordinate variance.

The histogram charts were optimized to fit the step variable, with `data.hist(bins=13)` to accurately represent the data. Since steps is a discrete variable, this makes visualization much easier than the continuous variables of X, Y, and Z. The tradeoff here is that the histograms for the other variables will also be divided into 13 equally-sized bins. However, with the vast number of datapoints, simplifying the data into a lower number of bins is more useful for visualization. There is too much data for accurate visualization of the coordinates to be useful.

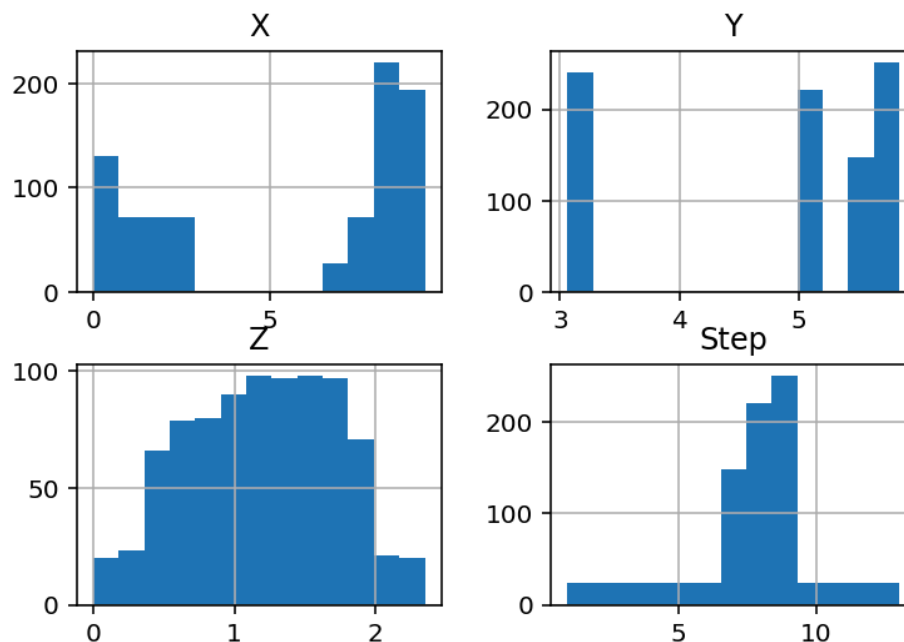


Figure 1: Histogram of raw data density (count vs approximate value).

Figure 2 shows a scatter plot of the data, with the coordinate value graphed against the step. This provides a more precise look at the distribution of the data per step. For clarification, it is not meant to be a map of the coordinates.

Analyzing Figure 2, initial observations of the data from the histogram still hold. The X-coordinates sweep from one side to the other, the Y-coordinates stay at nearly the same value the entire process, and the Z-coordinates are spread out the most per step. The initial observation of steps seven to nine containing the most variance and components remain true.

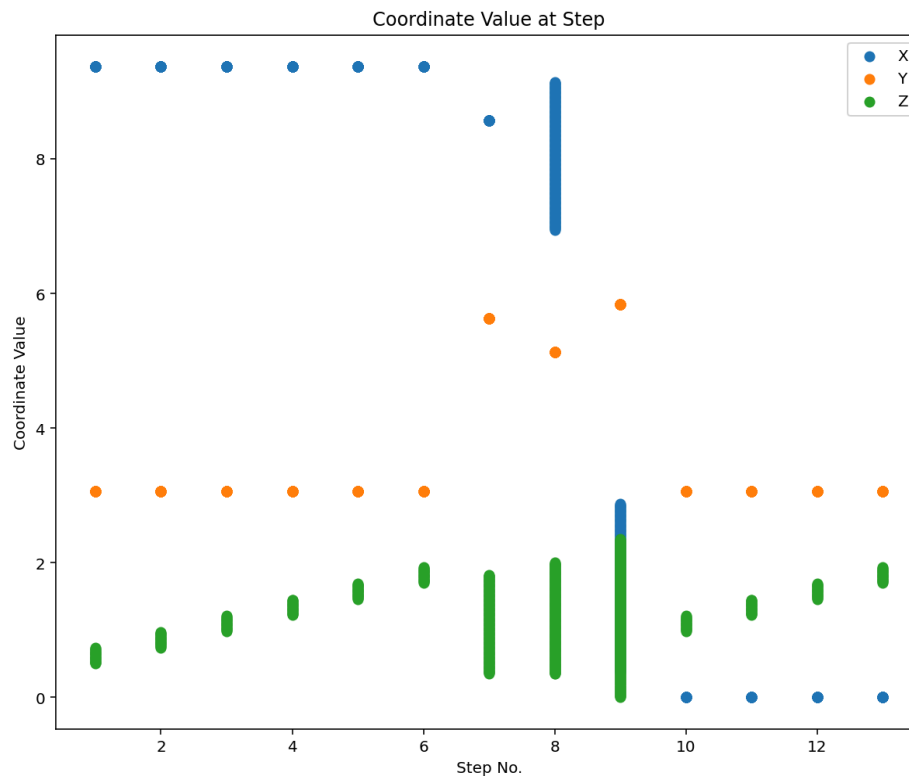


Figure 2: Scatter plot of Coordinate Value vs Step.

## 2.3 - Correlation Analysis

Figure 3 shows the correlation matrix between each variable in the dataset. The annotated matrix shows a strong, negative correlation between the X coordinate and the Step. From the sense of assembly or disassembly, the steps start from one side of the component and move across. The Y and Z coordinates have a less than 30% correlation to the step performed. These variables are two sides to an extreme. On one hand, the Y-coordinate remains stable regardless of step, whereas the Z-coordinate wildly changes at each step. The X-coordinate is the only variable

which changes with some pattern as the Steps proceed. Therefore, the 75% correlation makes sense. The negative correlation simply indicates the value of X decreases as steps increase. Thus, X is chosen to be the independent variable.

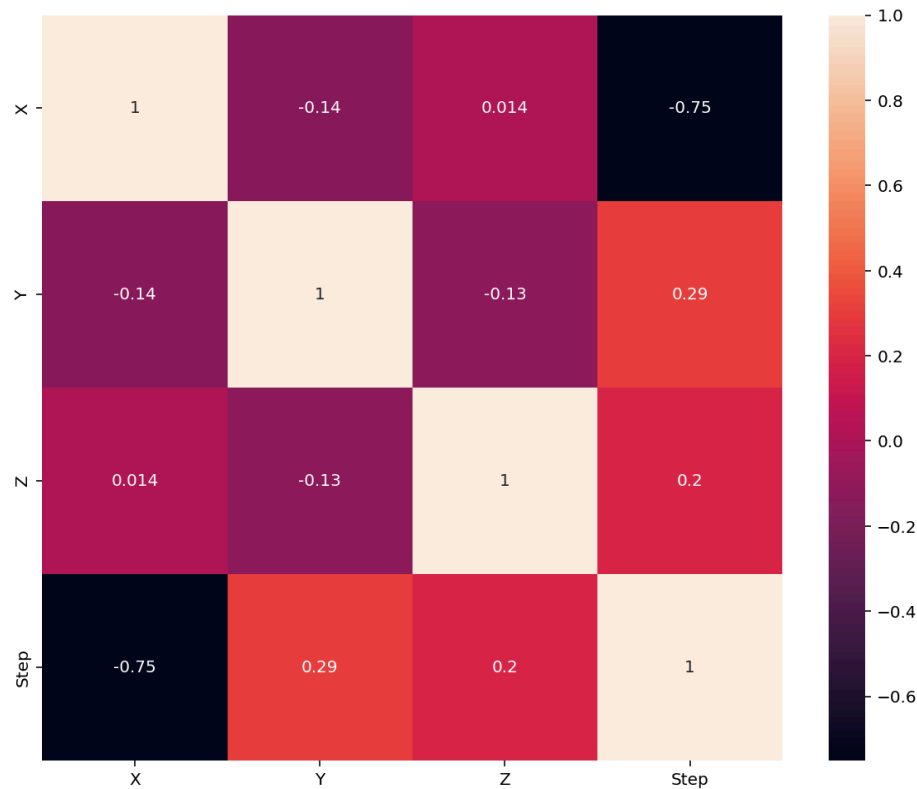


Figure 3: Correlation matrix of dataset, annotated.

## 2.4 - Classification Model Development/Engineering

The three models chosen to represent the data were:

- Logistic Regression
- Random Forest
- Support Vector Machine

Logistic regression was chosen based on the variety of data presented. There are many coordinates which can occur at one instance of X, though a decision can be made from how similar a value can match the trained data.

Random Forest was due to its ability to look at multiple paths and to not fixate on the shape of a specific function, such as a linear or sigmoid graph.

Support Vector Machine was chosen to separate out the data into steps as cleanly as possible.

Grid cross validation was used to determine the best parameters for each model. RandomSearch cross validation was used on the SVM model, which returned different but better parameters from grid CV. In Appendix A 2.4

## **2.5 - Model Performance Analysis**

Accuracy was important to determine if the model could correctly predict the step given provided data. Precision was important to see how often the model mistook data and labelled them incorrectly. Recall was important to see if the model potentially mislabeled correct data. F1 score combines precision and recall.

Based on the output, RandomForest had the highest accuracy, precision, recall, and f1 score, with a minimum of 98.8% across all metrics. SVM was next best, faltering only by a few decimal places to RandomForest. Logistic regression had the “worst” performance, with values hovering around 96%.

Analyzing RandomForest’s results, the model accurately predicted 98.8% of test data. It was extremely precise at 99%, meaning less than 1% of the data was mislabelled. The recall was 98.8%, very little correct data was mislabelled. Finally, it had an F1 score of 98.8%, meaning the precision and recall of the model was well balanced.

Confusion matrices can be found in Appendix A - 2.5.

## **2.6 - Stacked Model Performance**

Stacking the two best models, RandomForest and SVM, did not yield better performance. Results from RandomForest were repeated. Reviewing the results of RandomForest, it was already near-perfect before model stacking, therefore there was little to improve on in this new model.

## Appendix A - Code Console Output

### 2.4 - Model Development

#### Model 1 - Logistic Regression

```
-----
Model 1 Predictions: 7 Actual Value: 7
Model 1 Predictions: 10 Actual Value: 10
Model 1 Predictions: 9 Actual Value: 9
Model 1 Predictions: 9 Actual Value: 9
Model 1 Predictions: 9 Actual Value: 9
Model 1 training MAE = 0.04
Model 1 MAE (CV): 0.05
Model 1 Pipeline CV MAE: 0.05
Model 1 Pipeline Test MAE: 0.07
Grid Search
Fitting 5 folds for each of 16 candidates, totalling 80 fits
Best CV MAE: 0.020363905638421664
Best params: {'model__C': 10, 'model__max_iter': 200, 'model__penalty': 'l2', 'model__solver':
'lbfgs'}
Test MAE: 0.040697674418604654
```

#### Model 2 - Random Forest

```
-----
Model 2 Predictions: 7.0 Actual Value: 7
Model 2 Predictions: 10.0 Actual Value: 10
Model 2 Predictions: 9.0 Actual Value: 9
Model 2 Predictions: 9.0 Actual Value: 9
Model 2 Predictions: 9.0 Actual Value: 9
Model 2 training MAE = 0.0
Model 2 Mean Absolute Error (CV): 0.02
Model 2 Pipeline CV MAE: 0.02
Model 2 Pipeline Test MAE: 0.02
Grid Search
Fitting 5 folds for each of 216 candidates, totalling 1080 fits
Best CV MAE: 0.04418999953189334
Best params: {'model__max_depth': 10, 'model__max_features': 'log2', 'model__min_samples_leaf':
1, 'model__min_samples_split': 2, 'model__n_estimators': 30}
Test MAE: 0.07110668581684307
```

#### Model 3 - SVM

```
-----
Model 3 Predictions: 7 Actual Value: 7
Model 3 Predictions: 10 Actual Value: 10
Model 3 Predictions: 9 Actual Value: 9
Model 3 Predictions: 9 Actual Value: 9
Model 3 Predictions: 9 Actual Value: 9
Model 3 training MAE = 0.01
Model 3 Mean Absolute Error (CV): 0.01
Model 3 Pipeline CV MAE: 0.01
Model 3 Pipeline Test MAE: 0.0
Grid Search
```



Fitting 5 folds for each of 56 candidates, totalling 280 fits  
 Best CV MAE: 0.0072675341161536015  
 Best params: {'model\_\_C': 1000, 'model\_\_degree': 4, 'model\_\_gamma': 'scale', 'model\_\_kernel': 'rbf'}  
 Test MAE: 0.01744186046511628

RandomizedSearchCV  
 Fitting 5 folds for each of 10 candidates, totalling 50 fits  
 Best CV MAE: 0.0072675341161536015  
 Best params: {'model\_\_kernel': 'rbf', 'model\_\_gamma': 'scale', 'model\_\_degree': 4, 'model\_\_C': 1000}  
 Test MAE: 0.01744186046511628

## 2.5 - Model Performance Analysis

### LogisticRegression Metrics

LG Training accuracy: 0.9927325581395349

LG Test accuracy: 0.9593023255813954

LG Confusion Matrix:

```
[[ 5  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  5  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  1  4  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  3  1  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  2  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  5  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  29  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  44  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  50  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  5  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  5  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  3  2]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  4]]
```

LG Precision: 0.966016057585825

LG Recall: 0.9593023255813954

LG F1 Score: 0.9578224101479915

### RandomForest Metrics

RF Training accuracy: 1.0

RF Test accuracy: 0.9883720930232558

RF Confusion Matrix:

```
[[ 5  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  5  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  5  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  4  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  4  1  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  5  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  29  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  44  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  50  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  5  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  5  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  5  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  4]]
```

RF Precision: 0.9903100775193799

RF Recall: 0.9883720930232558

RF F1 Score: 0.9882546394174301

#### SVM Metrics

SVM Training accuracy: 0.998546511627907

SVM Test accuracy: 0.9825581395348837

#### SVM Confusion Matrix:

```
[[ 5  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  5  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  5  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  4  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  4  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  5  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 29  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 44  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 50  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  5  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  5  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  4  1]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  4]]
```

SVM Precision: 0.9856589147286822

SVM Recall: 0.9825581395348837

SVM F1 Score: 0.982440685929058

## 2.6 - Stacked Model Performance

Stacked Classifier Training Accuracy: 1.0

Stacked Classifier Test Accuracy: 0.9883720930232558

#### Stacked Classifier Confusion Matrix:

```
[[ 5  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  5  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  5  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  4  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  4  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  5  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 29  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 44  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 50  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  5  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  5  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  4]]
```

Stacked Classifier Precision: 0.9903100775193799

Stacked Classifier Recall: 0.9883720930232558

Stacked Classifier F1 Score: 0.9882546394174301

## 2.7 Model Evaluation

```
[ 5  8 13  6  4]
```

## **Appendix B - GitHub Link**

[Link to GitHub](#) (or copy-paste below)

*<https://github.com/bmakTMU/AER850/tree/main/Project%201>*