The model I created test all capacitors with all 8 total version of features. This purpose was to train the algorithm to find the best weights. The features were created per the example given in the document instructions. In the first round of iterations, 1000 per feature, 2 features were tested per capacitor. After each capacitor was tested, the new weight was updated. Upon reaching the last capacitor, the cost, J, was recorded as the best J for that specific number of features. This process was repeated up to a total of 8 features with the best J being recorded with the best number of weights.

Now that the algorithm was trained, the best weights were used on the test case to compare the predicted output to the actual output. The confusion matrix, accuracy, precision, recall, and F1 of the predicted data was calculated.

Initial Weights: 2, for all values of w in w_0x_0, … w_8x_8

Initial Learning Rate: 0.2

Initial J Value: 2.83705893

Final Weights: [2.00034251, 2.00018724, 2.00010236, 2.000167, 2.00009129, 2.00004451, 2.00008142, 2.00004991, 2.00002433] # w_0 + 8 features

Final Learning Rate: 0.2

Number of Iterations: 1000 (Number of capacitors tested)

Final J Value: 0.00171403

J Test Value: 1.73787

Confusion Matrix: FP: 17, FN: 0, TP: 16, TN: 0

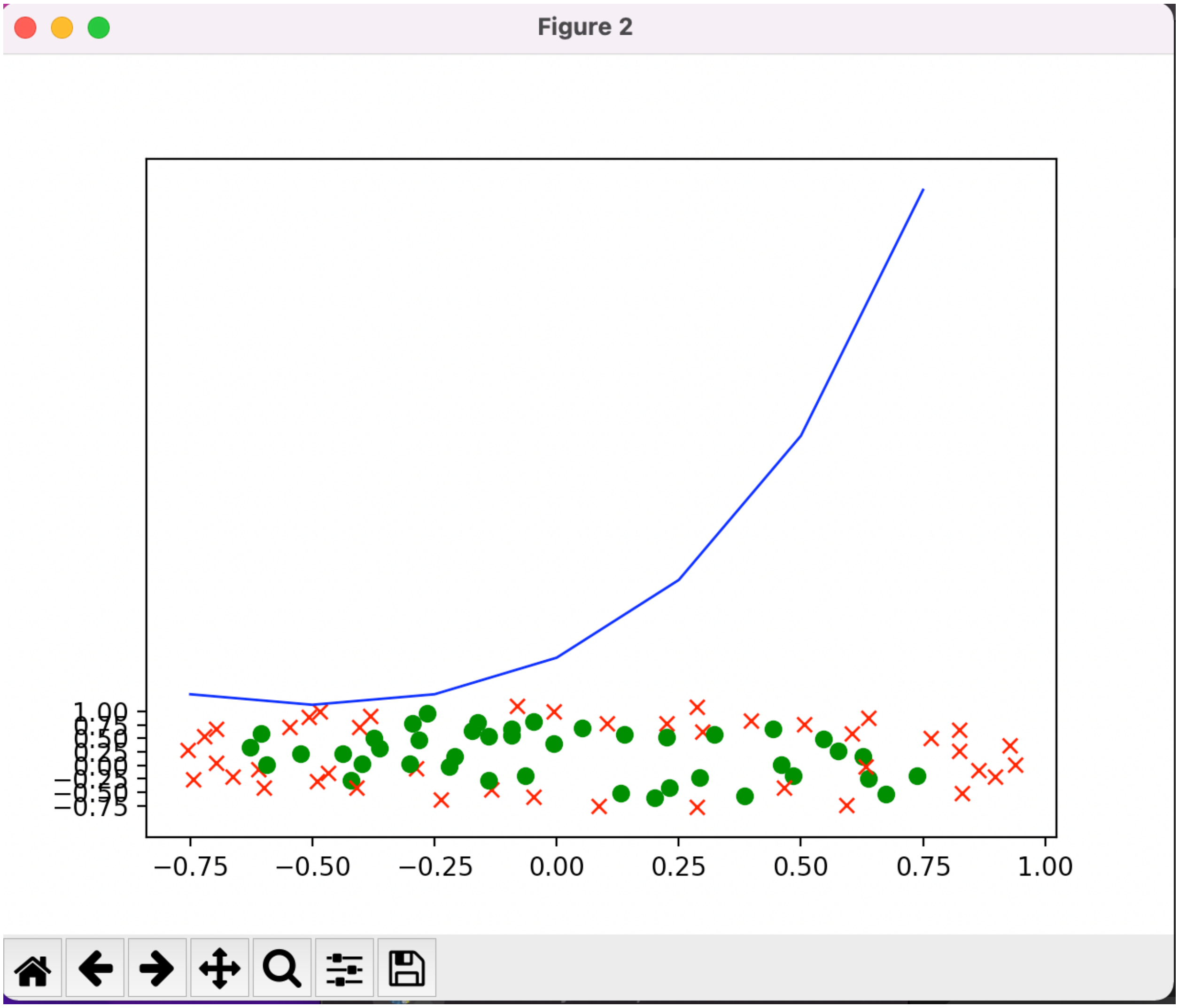
Accuracy: 48.48 %

Precision: 48.48 %

Recall: 1.0

F1: 0.65

Plot of data with Hyperplane



Plot of Number of Iterations vs J

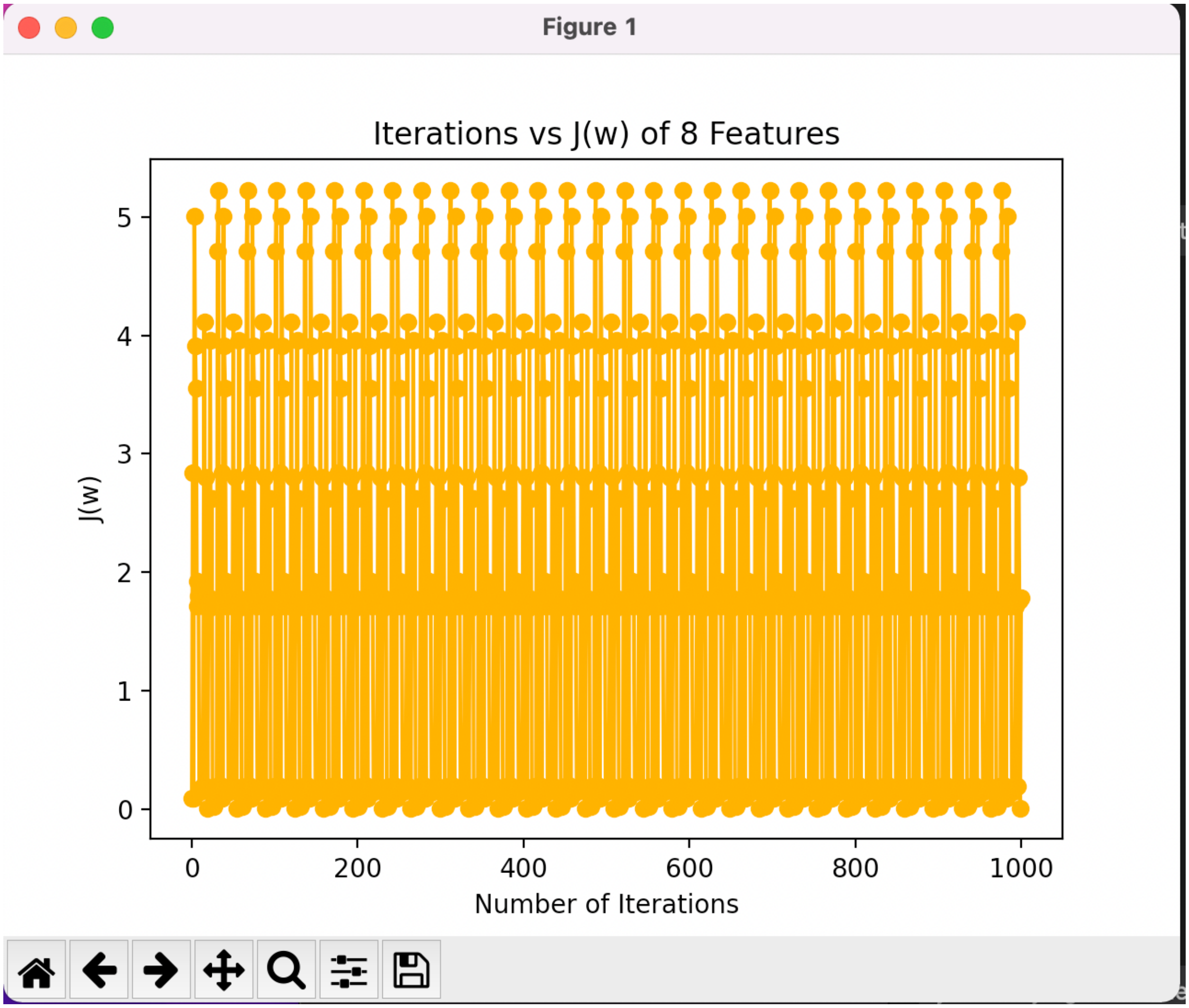


Figure 1
Iterations vs J(w) of 8 Features
J(w)
0
1
2
3
4
5
0
200
400
600
800
1000
Number of Iterations