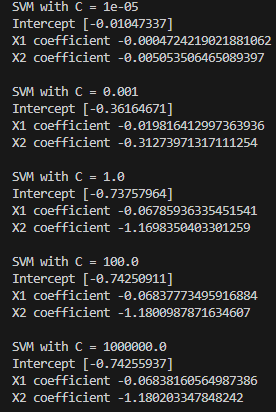
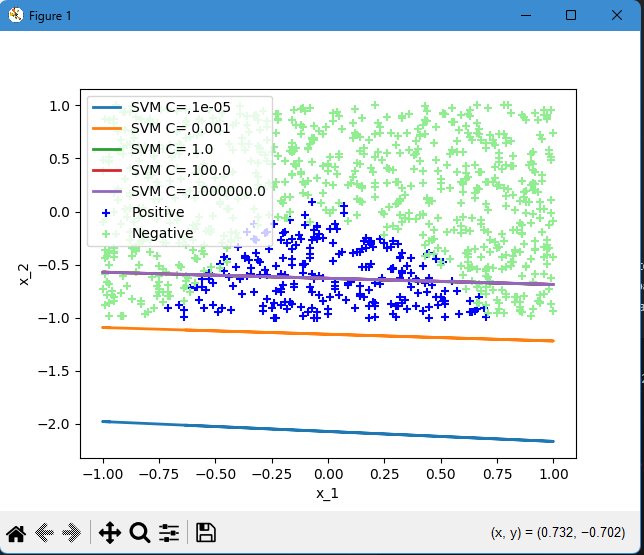
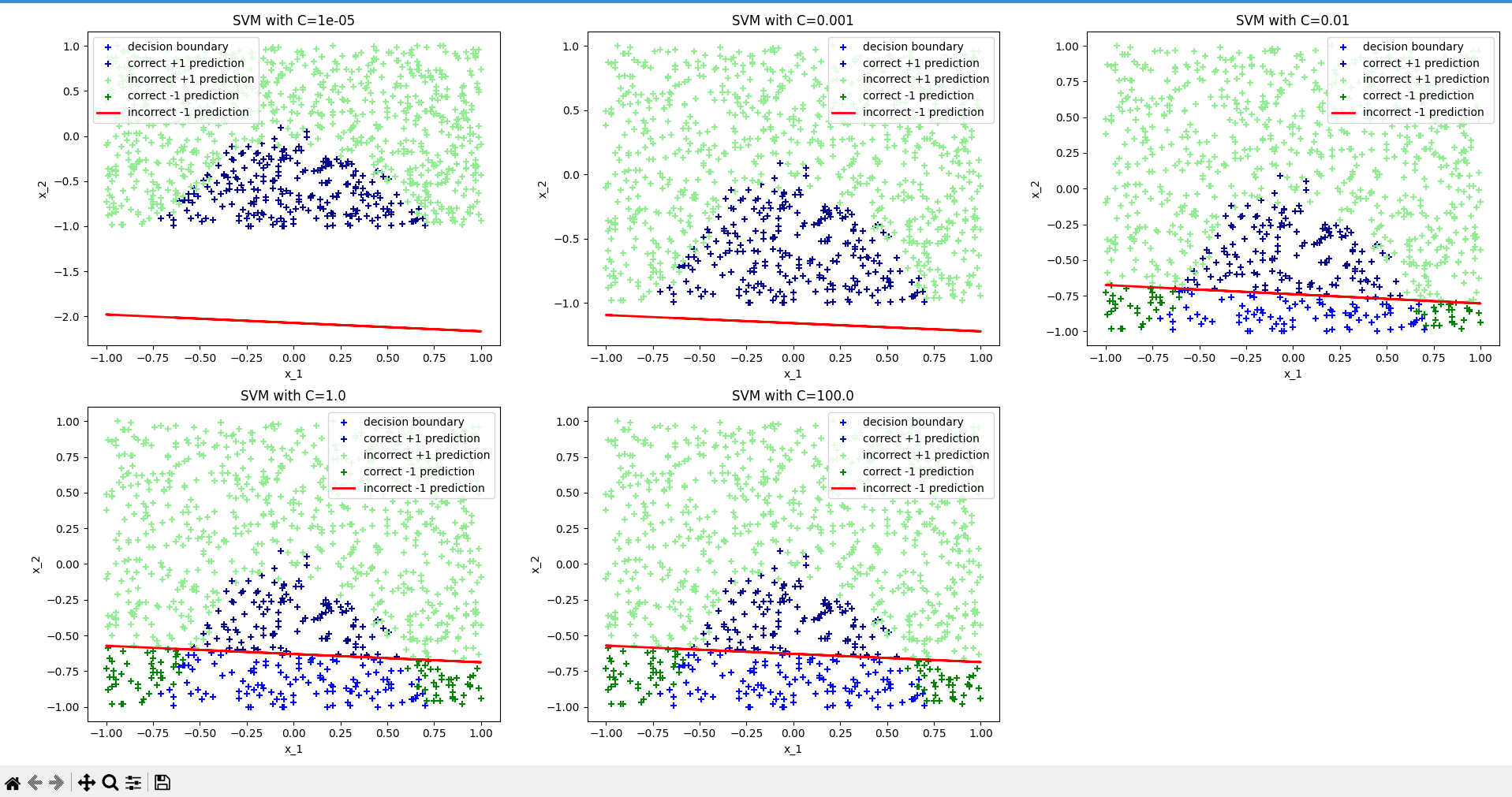
1. # id:25--50—25 : this is the data code I was told to append to the start of the report
   1. Question A
      1. I did X[ y == 1 ] and X[ y == -1 ] to separate the positive and negative values from the dataI then used matplotlib.pyplot to scatter the positive and negative values making the positive ones blue and the negative ones green. The data seems to follow a curved decision boundaryA green and blue dots

         AI-generated content may be incorrect.
      2. Using sklearn I created a logisticRegression model for the using X(thefeatures/inputs) and Y(the target/outputs) and printed the Intercept, X1 coefficient and X2 coefficientA black background with white numbers

         AI-generated content may be incorrect. . The X1 coefficient is very small, showing that it currently plays very little impact in the predictions. The X2 coefficient is very large, showing that the second feature is much more impactful in the result. The intercept lies in the middle, showing that the X2 coefficient can
      3. By recreating y=mx+c using the parameters of the regression modelA screen shot of a computer

         AI-generated content may be incorrect.. I was able to create the decision\_boundary and plot it on the graph in redA screen shot of a computer screen

         AI-generated content may be incorrect.I then created 4 categories of points depending on the result of the training data and the prediction of the model and colored them accordingly, this completely aligned with the decision boundary, thus proving one another. From the decision boundary we can notice a few things that align with the parameters we printed earlier. The X1 coefficient being so small compared to the X2 coefficient causes a very small slope of the line.
      4. The predictors and training data are greatly misaligned, this is very obviously because the data seems to follow a nonlinear curve while the regression model is capable of a linear regression
   2. Using SVM classifiers
      1. I created code to loop through a list of c values and put them all onto the graph, these values were returned, the biggest thing to notice is how the values don’t change much with higher values of C, as C approaches 0, it causes the intercept and both coefficients to also approach 0, but as it rises far above 1 it isnt influenced much. This can also be clearly seen on the graph.  This graph demonstrates how smaller numbers of C get increasingly lower down on the graph, while the larger values of C produce decision bounddaries that are almost inseperable. The smaller C allows more entries to qualify as “negative” thus lowering the decision much lower down
      2. Using recycled code from the previous prediction vs data comparison I was able to make 5 separate charts for each value of C. The chart clearly shows the same information that was shown when toghether on the same graph, where lowering C causes the margin to lower and thus allowing more entries to be predicted as positive. On this instance I added C=0.01 as it demonstrates the progression of the bar lowering, where it is allowing many more entries than C=1 but without allowing all of them
      3. A greater C means a lower margin, which ensures that the model is as accurate as possible for the data given, and thus a lower C value would result in a larger margin, which would become less accurate. Usually a high C value would run the risk of overfitting a model to a set of data. I believe in this instance that is not a concern as overfitting a liner model is very unlikely.
      4. While the intercept and coefficients of the highC-SVM models do differ from the logistic regression they are just scaled down , the predictions are identical and the decision\_boundary is the same between the 2 once passed through the formula. On the other hand the lowC-SVM models are greatly different from the logistic regression, even outside of direct scaling.
   3. Square of each feature
      1. A screenshot of a computer program

         AI-generated content may be incorrect.By recreating the data with the squared values we can see that the X1 squared coefficient skyrockets, the Intercept sinks and the X2 coefficient increases(downward) significantly. This shows that the graph should really be based on the value of X2 and the squared value of X1 almost exclusively
      2. Again recycling much of the code, I created the same system of colors for correct/incorrect positive/negative values, and finally acquired this tableA screen shot of a computer screen

         AI-generated content may be incorrect., with only a handful of incorrect predictions(many of which visually seem to be outliers) we can see that this graph is a significantly better fit for the data than both the logistic regression and the SVM models. This demonstrates that the model always shouldve a nonlinear model in order to achieve better results.
      3. This classifier is significantly better than a baseline predictor. By comparing the length of the correct plotList vs the incorrect plotList I found that it had an accuracy of 97.3% accuracy(971 right out of 999). Compared to a baseline predictor (accuracy found by comparing the length of y==1 and y==-1) would have an accuracy of 76.6% (766 right out of 999), this shows that the model makes about 10 times less errors than a baseline predictor