Avik Gomes and Ally Hu

Introduction to Machine Learning Final Project

COVID-19 Vaccine Side Effects

With the uncertainty surrounding the newly rolled out vaccines, Avik and I were curious as to how different people experienced side effects from the vaccine, and if there was a way to indicate whether one would have a bad experience or not. We utilized data from the Vaccine Adverse Event Reporting System(VAERS) to see if given one's sex, medictions, current illness, history, allergies, which shot you took, the dose number, and after how many days since the shot how many side effects one would likely get. VAERS provided us with 3 .csv files, in which we had to consolidate them and remove irrelevant information. While assessing the data, we noticed that there is an imbalance between males and females, 19,154 out of 79,883 were male whereas 60,729 were female. Since we will be taking a sample out of this data, it is very likely that the sample will also include this imbalance. Additionally, within the data, the majority of samples were of Moderna or Pfizer, so samples that were of Jansen were removed from the pool. While looking at symptoms faced, we counted how many symptoms individuals experienced as a feature, instead of naming the symptoms. And while keeping track of current illness, allergies, medications, we chose to process the data so that if one has such, it would become a 1 and if one did not, it would be a 0 for those features. In terms of scope and the time allotted, predicting which symptoms would be experienced was not feasible, so we decided to predict severity(how many symptoms one will experience.)

The three models we chose to work with for our project were neural network, support vector machine and logistic regression.

For our support vector machine model, we chose to try radial basis function, linear and polynomial kernels. Comparing the different kernels, the linear kernel provided the lowest accuracy scores, but the accuracy scores from RBF and polynomial kernel models were not much different. However, we can attribute this to the fact that our data is not linear. Radial basis function kernels normally provide higher accuracies, and are the default when creating a SVM model. The RBF model works best as it lifts samples onto different dimensions, where a linear decision boundary can be made to separate the classes. Accuracy scores can be found at figures 10, 11 and 12.

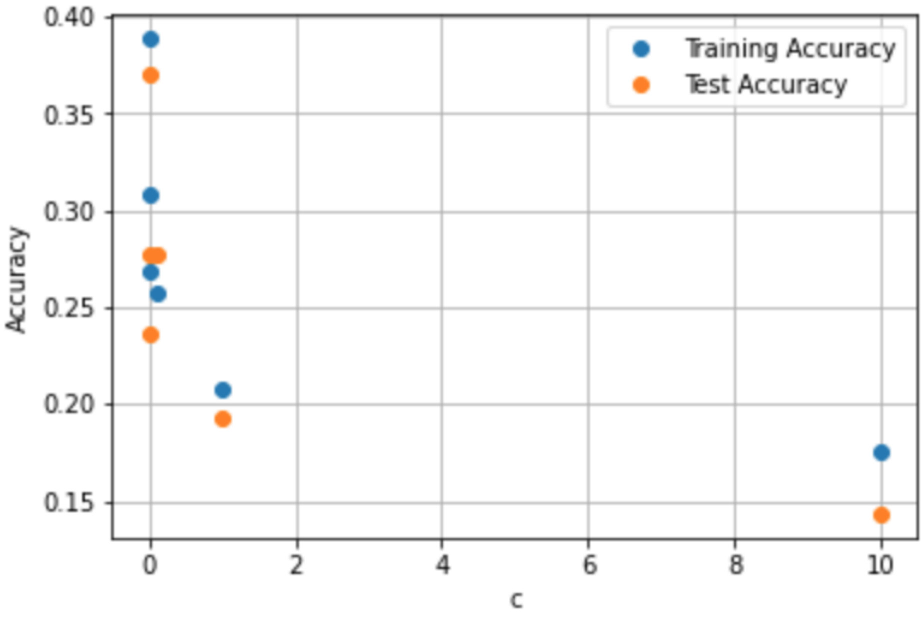


Figure 1: Graph of training and test accuracies for Linear Kernel

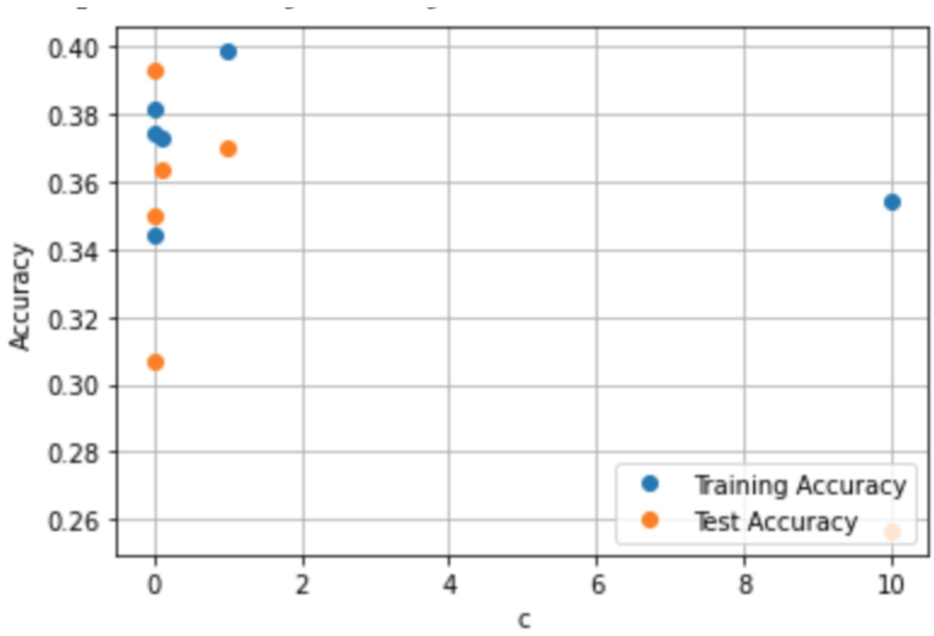


Figure 2: Graph of training and test accuracies for RBF Kernel

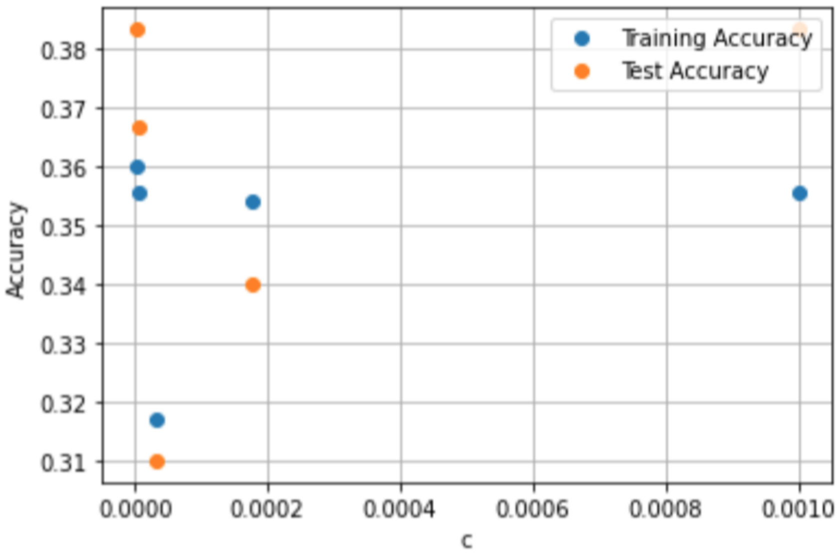


Figure 3: Graph of training and test accuracies for Polynomial Kernel

For our logistic regression model, we choose to do a Logistic Regression with a L1 penalty, L2 penalty, and a Logistic Regression with a polynomial feature with a L1 penalty. After a certain point all the data for all three of these went to a training data accuracy of about 41% and a test data accuracy of about 47%. Using a polynomial feature would get these results the fastest, but there was a "*The max_iter was reached which means the coef_ did not converge*" warning, so it is possible for the polynomial feature to go to a higher accuracy.

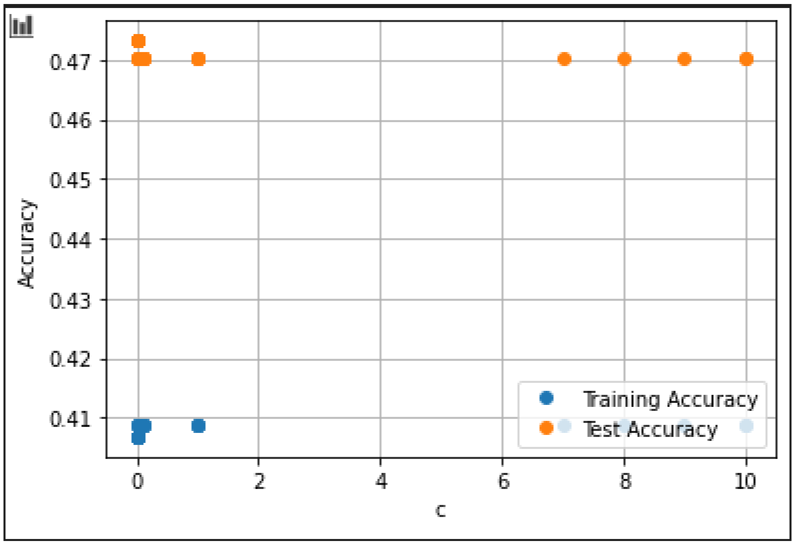


Figure 4: Graph of Training and Test Accuracies for Logistic Regression with a L1 penalty

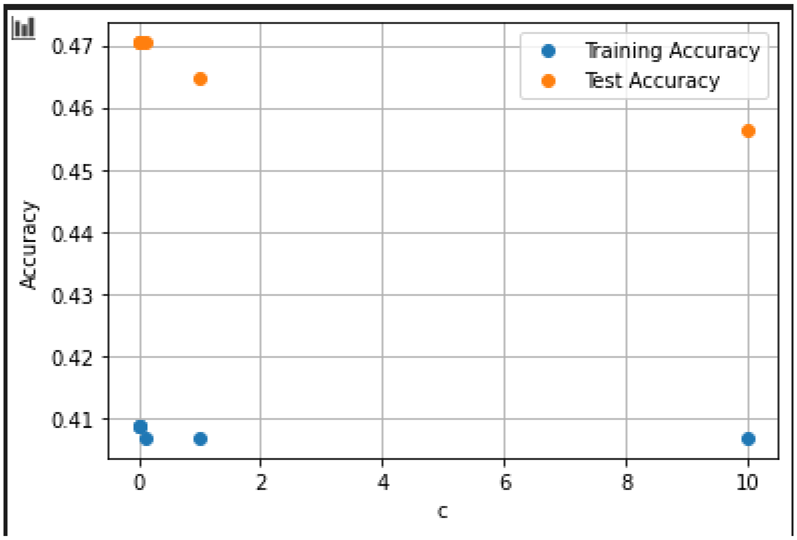


Figure 5: Graph of Training and Test Accuracies for Logistic Regression with a L2 penalty

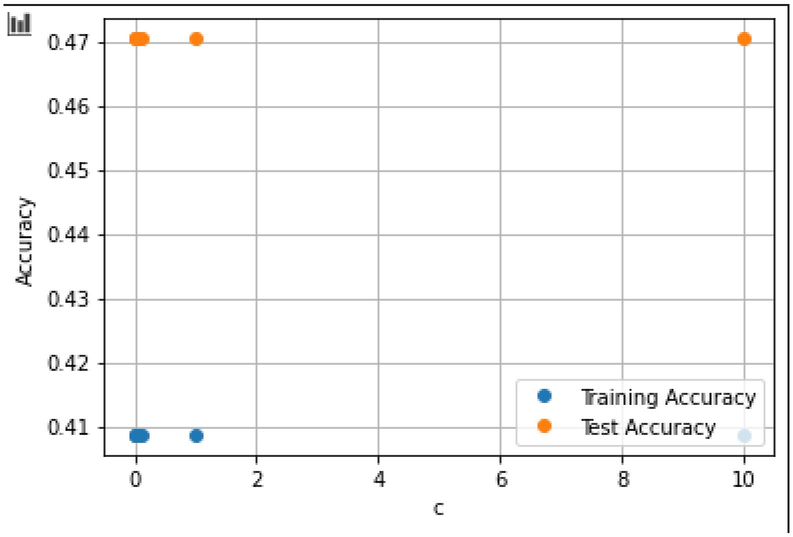


Figure 6: Graph of Training and Test Accuracies for Logistic Regression with a Polynomial Feature and a L1 penalty

Three different neural networks were made with the initial level having nine neurons, for the nine features we are using to calculate the number of symptoms someone would have, 30 hidden neurons, and 6 neurons on the final level. The two activation functions that were used were the sigmoid function and the tanh function. And another neural network used the sigmoid function with L1 regularization. Using just the sigmoid function gave the best results, but there was an overflow error when using the sigmoid function with L1 regularization so it is possible that it would yield the best results (or not). This is the reason why we got a 0.0% accuracy with the L1 regularization.

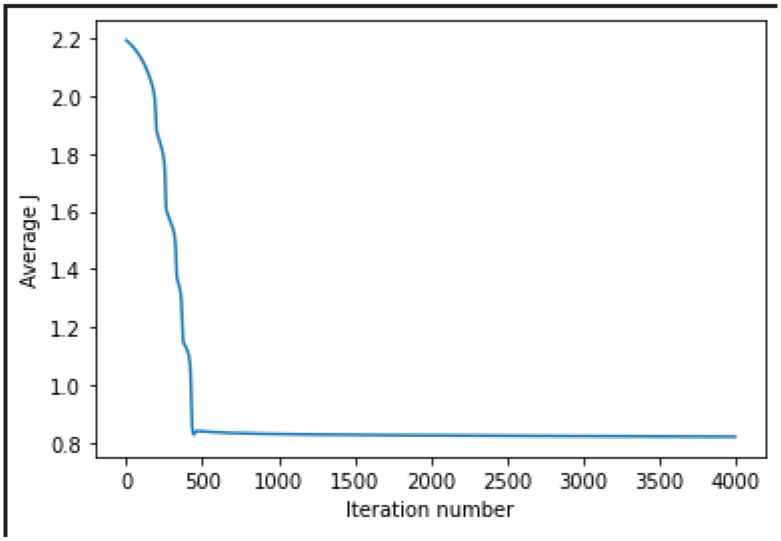


Figure 7: Graph of Average Cost vs Number of iterations using a Sigmoid as the Activation function

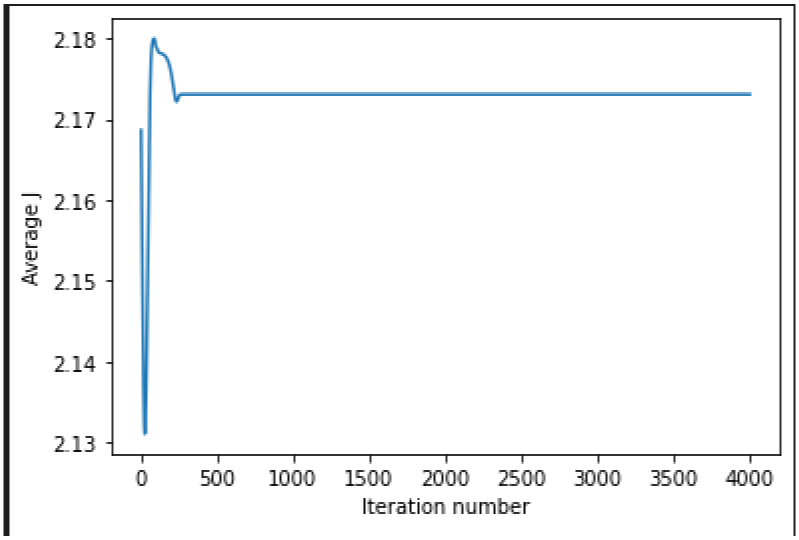


Figure 8: Graph of Average Cost vs Number of iterations using a Sigmoid as the Activation function with L1 regularization

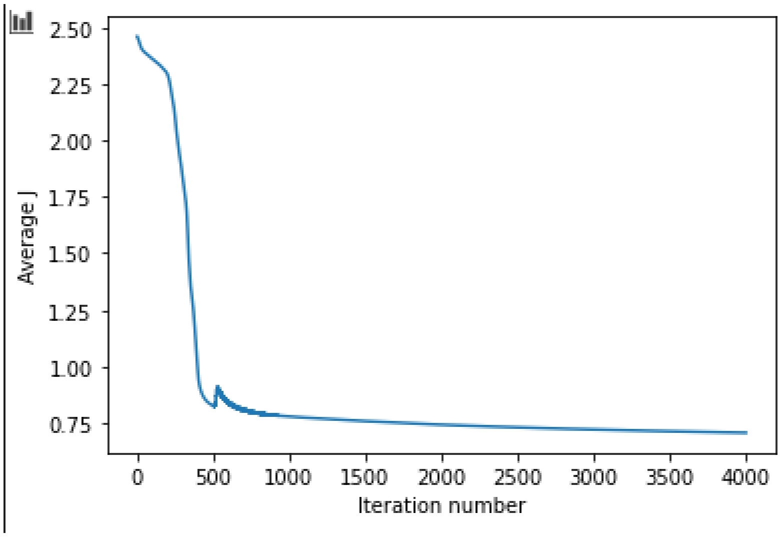
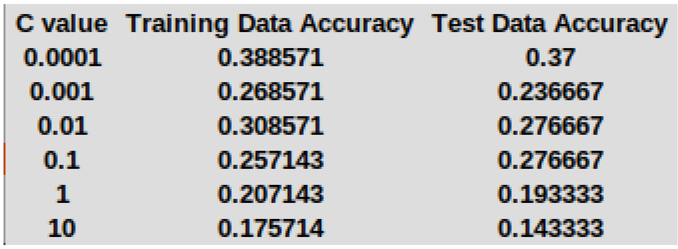


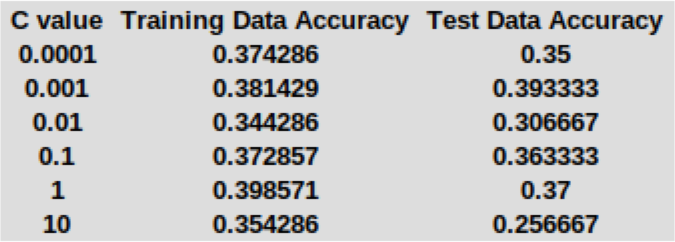
Figure 9: Graph of Average Cost vs Number of iterations using a Tanh as the Activation function

Overall, our models show high bias, low variance and general underfitting. As seen in our SVM graphs(figures 1, 2 and 3), there is not a large discrepancy between our training and test accuracies, which shows that error is low. However, in combination with our low accuracies, it shows that our model is underfitting.



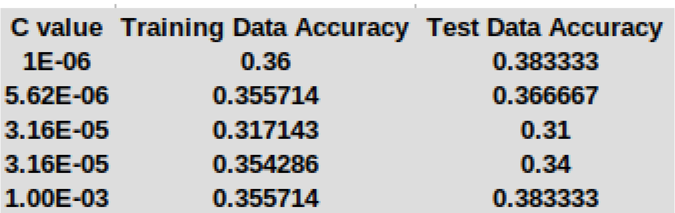
C value	Training Data Accuracy	Test Data Accuracy
0.0001	0.388571	0.37
0.001	0.268571	0.236667
0.01	0.308571	0.276667
0.1	0.257143	0.276667
1	0.207143	0.193333
10	0.175714	0.143333

Figure 10: Table for Training and Test Accuracies for Linear Kernel



C value	Training Data Accuracy	Test Data Accuracy
0.0001	0.374286	0.35
0.001	0.381429	0.393333
0.01	0.344286	0.306667
0.1	0.372857	0.363333
1	0.398571	0.37
10	0.354286	0.256667

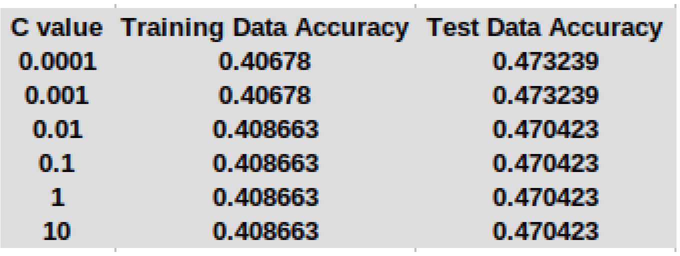
Figure 11: Table for Training and Test Accuracies for RBF Kernel



C value	Training Data Accuracy	Test Data Accuracy
1E-06	0.36	0.383333
5.62E-06	0.355714	0.366667
3.16E-05	0.317143	0.31
3.16E-05	0.354286	0.34
1.00E-03	0.355714	0.383333

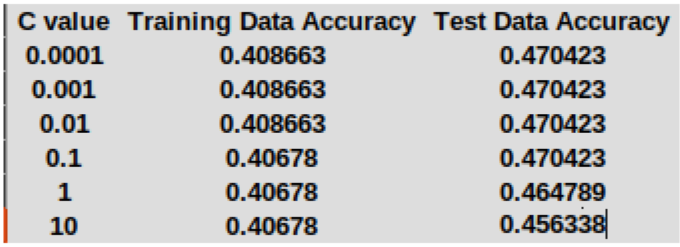
Figure 12: Table for Training and Test Accuracies for Polynomial Kernel

For Logistic regression just like SVMs the overall error between the training and test sets are somewhat low, although they are higher than the SVM. Nonetheless, we are still underfitting because the overall percentages are low.



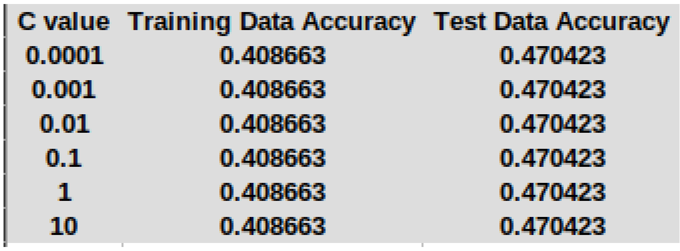
C value	Training Data Accuracy	Test Data Accuracy
0.0001	0.40678	0.473239
0.001	0.40678	0.473239
0.01	0.408663	0.470423
0.1	0.408663	0.470423
1	0.408663	0.470423
10	0.408663	0.470423

Figure 13: Table for the Training and Test Accuracies for Logistic Regression with a L1 penalty



C value	Training Data Accuracy	Test Data Accuracy
0.0001	0.408663	0.470423
0.001	0.408663	0.470423
0.01	0.408663	0.470423
0.1	0.40678	0.470423
1	0.40678	0.464789
10	0.40678	0.456338

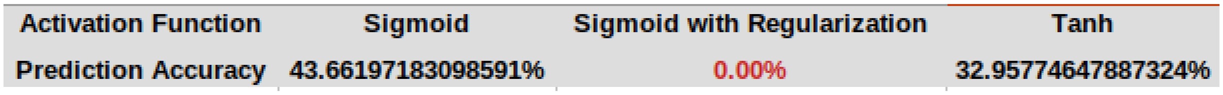
Figure 14: Table for theTraining and Test Accuracies for Logistic Regression with a L2 penalty



C value	Training Data Accuracy	Test Data Accuracy
0.0001	0.408663	0.470423
0.001	0.408663	0.470423
0.01	0.408663	0.470423
0.1	0.408663	0.470423
1	0.408663	0.470423
10	0.408663	0.470423

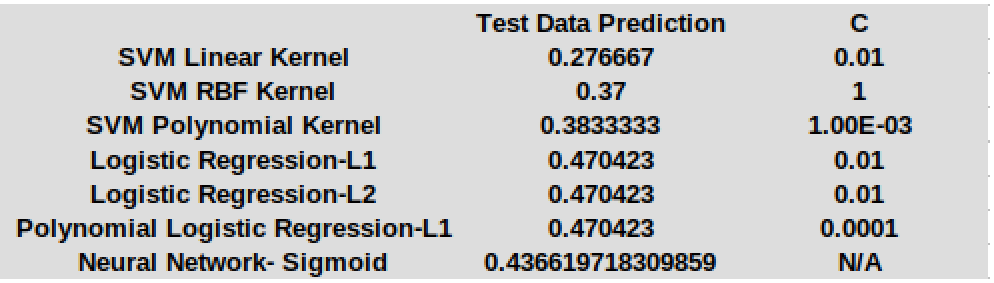
Figure 15: Table for the Training and Test Accuracies for Logistic Regression with a Polynomial Feature and a L1 penalty

Finally for neural networks, we are still underfitting, but when using a sigmoid function without regularization the prediction accuracy was about 44%.



Activation Function	Sigmoid	Sigmoid with Regularization	Tanh
Prediction Accuracy	43.66197183098591%	0.00%	32.95774647887324%

Figure 16: Table of Prediction Accuracies for each Activation Function



	Test Data Prediction	C
SVM Linear Kernel	0.276667	0.01
SVM RBF Kernel	0.37	1
SVM Polynomial Kernel	0.3833333	1.00E-03
Logistic Regression-L1	0.470423	0.01
Logistic Regression-L2	0.470423	0.01
Polynomial Logistic Regression-L1	0.470423	0.0001
Neural Network- Sigmoid	0.436619718309859	N/A

Figure 17: Best Values from all the Data

It cannot be concluded whether it is because of the small sample size or not, but the more likely reason for underfitting is the generalization of certain features. As seen in our data, our models consider allergies, history, current illnesses, medications and symptoms, but while preprocessing the data, we lost a lot of possibly valuable information. It may be important to consider what specific allergies one has, or which medications one is taking at the time and not just the binary “yes” or “no.” Other information that could have been helpful to keep would have been which specific symptoms were experienced, not just how many symptoms.

Out of all of the models created, the most accurate one was the logistic regression model. In all three logistic regression models, L1 regularization, L2 regularization and polynomial logistic regression with L1 regularization, the test prediction accuracy scores were all the same.