Project Write Up

Heart Attack Analysis

Comparing Performances of Logistic Regression, SVM, and Neural Network Models to Predict Likelihood of Heart Attacks

Spring 2021

Prof. Sellie

JinWen Loh, Tom Liu

April 28, 2021

Overview

The objective of this project is to analyze and compare the performances of three machine learning models to predict the likelihood that someone will get a heart attack. Said models are namely: Logistic Regression, SVMs, and Neural Networks. High (or good) performance is defined as having high accuracy when ran on the verification/test set.

Different variants of these models would also be tested to see if any differences could be observed from any changes. For example, we may perform different polynomial feature transformations for logistic regression, and note any differences in performance observed.

Apart from that, L2 regularization would also be applied to these variations in order to test if any improvements in performance could be observed.

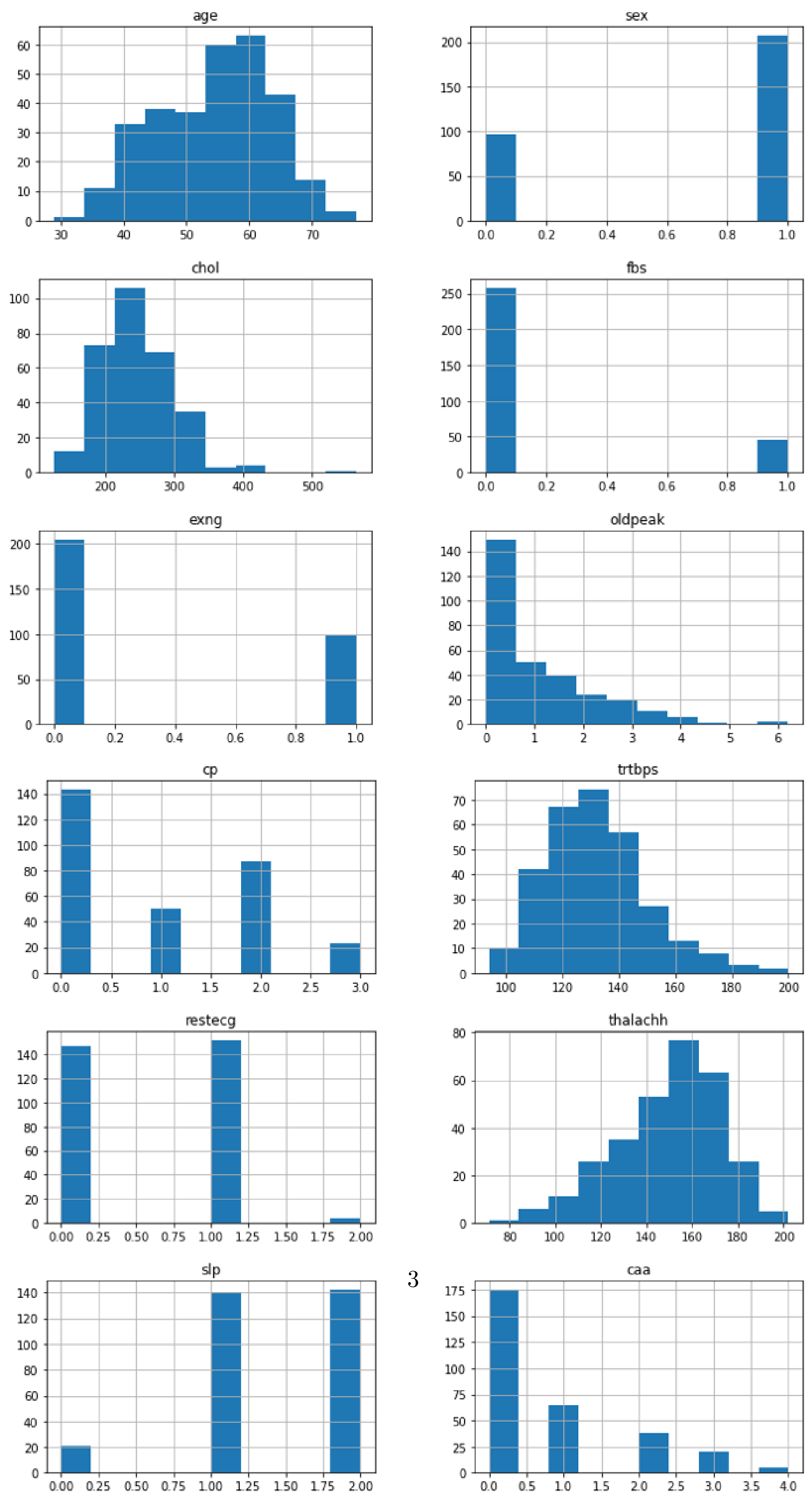
Analyzing Data

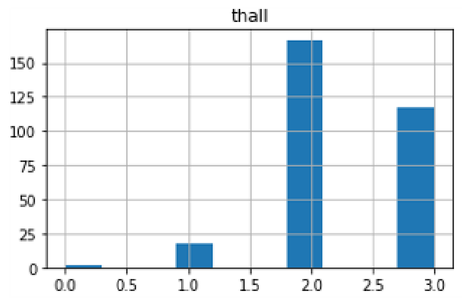
The following is the data we will be analyzing:

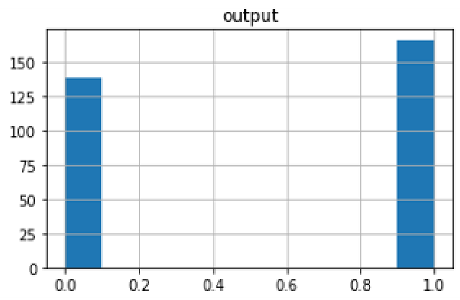
- Age : Age of the patient
- Sex : Sex of the patient
- exang: exercise induced angina (1 = yes; 0 = no)
- ca: number of major vessels (0-3)

- cp : Chest Pain type chest pain type
  - Value 1: typical angina
  - Value 2: atypical angina
  - Value 3: non-anginal pain
  - Value 4: asymptomatic
- trtbps : resting blood pressure (in mm Hg)
- chol : cholestoral in mg/dl fetched via BMI sensor
- fbs : (fasting blood sugar ¿ 120 mg/dl) (1 = true; 0 = false)
- rest_ecg : resting electrocardiographic results
  - Value 0: normal
  - Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of greater than 0.05 mV)
  - Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
- thalach : maximum heart rate achieved
- target : 0 = less chance of heart attack 1 = more chance of heart attack

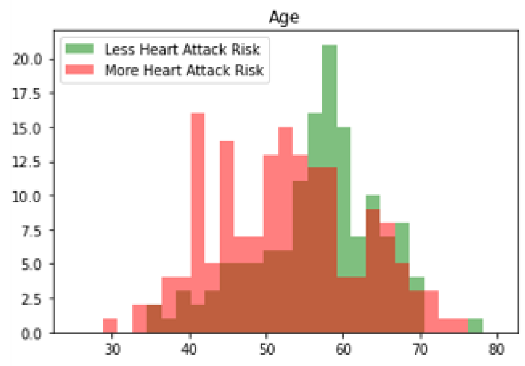
The following histograms illustrates the distributions of all the features.

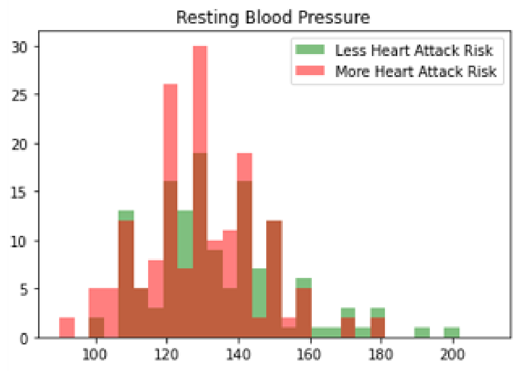

age
sex
chol
fbs
exng
oldpeak
cp
trtbps
restecg
thalachh
slp
caa

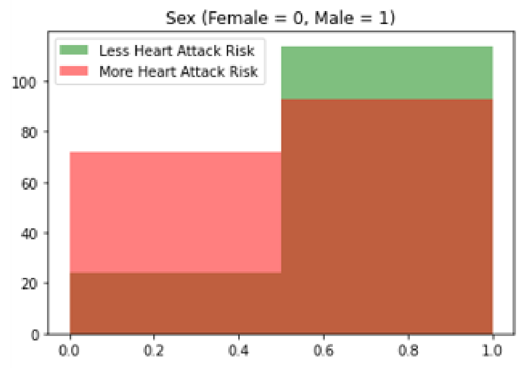


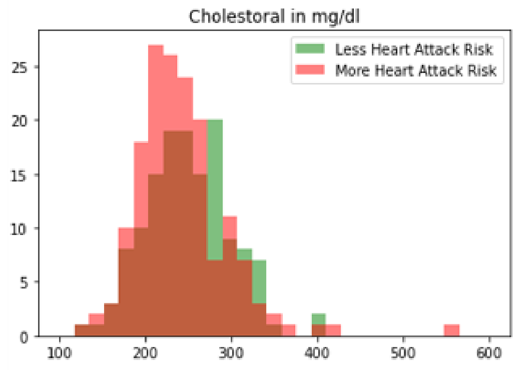


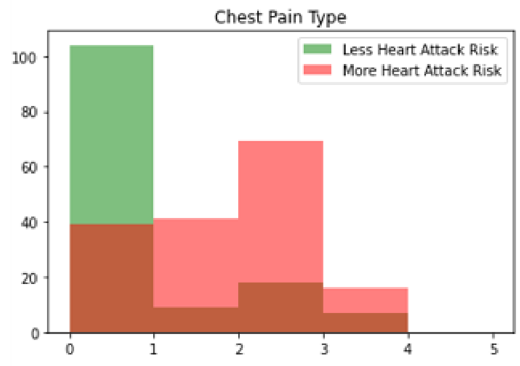
Through the histograms, most of the features demonstrates a Gaussian distribution, only those that have binary values were unable to. We would like to see the correlations between each of the features and the output we are trying to predict. The following graphs shows the relations of the distributions in each feature between people that has less chance of heart attack and people that has more chance of heart attack.

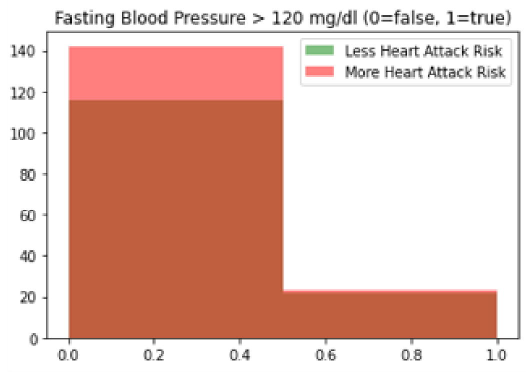

Age
Less Heart Attack Risk
More Heart Attack Risk

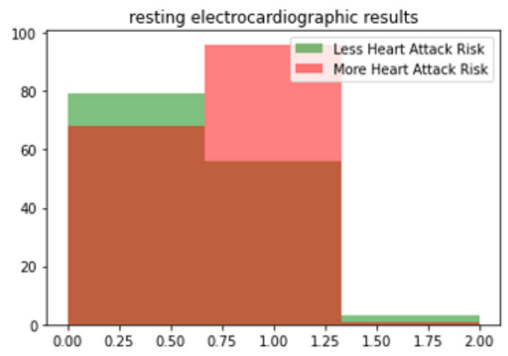

Resting Blood Pressure
Less Heart Attack Risk
More Heart Attack Risk

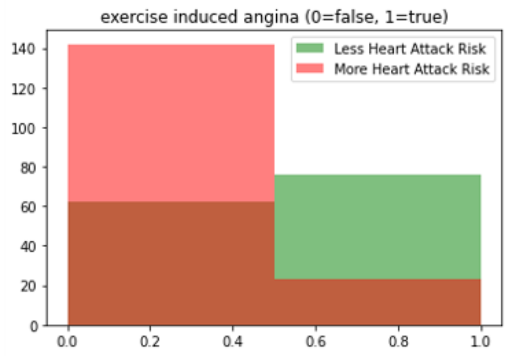

Sex (Female = 0, Male = 1)
Less Heart Attack Risk
More Heart Attack Risk

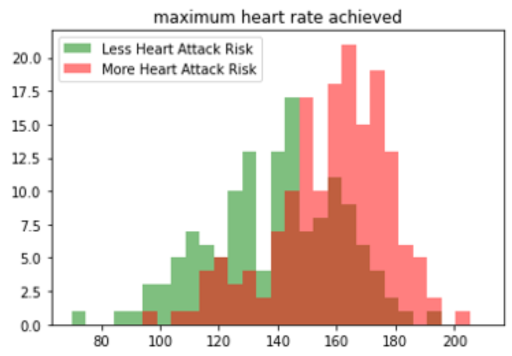

Cholestoral in mg/dl
Less Heart Attack Risk
More Heart Attack Risk

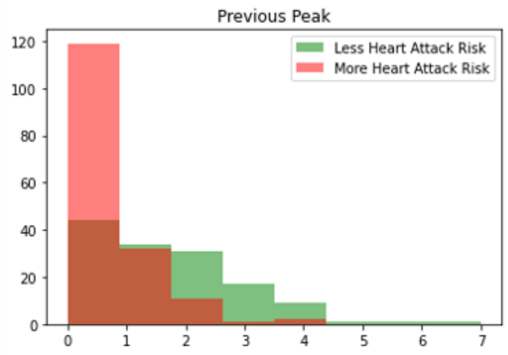

Chest Pain Type
Less Heart Attack Risk
More Heart Attack Risk

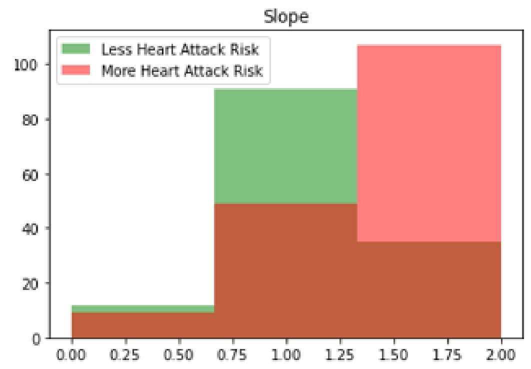

Fasting Blood Pressure > 120 mg/dl (0=false, 1=true)
Less Heart Attack Risk
More Heart Attack Risk

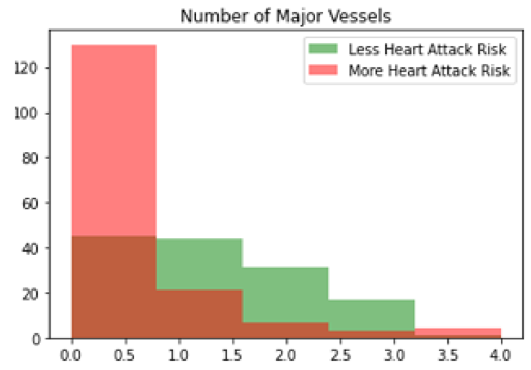

resting electrocardiographic results
Less Heart Attack Risk
More Heart Attack Risk

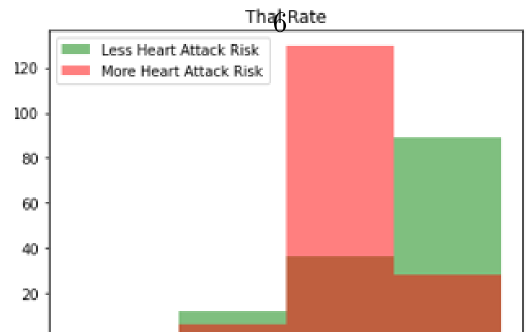

exercise induced angina (0=false, 1=true)
Less Heart Attack Risk
More Heart Attack Risk


maximum heart rate achieved
Less Heart Attack Risk
More Heart Attack Risk


Previous Peak
Less Heart Attack Risk
More Heart Attack Risk


Slope
Less Heart Attack Risk
More Heart Attack Risk


Number of Major Vessels
Less Heart Attack Risk
More Heart Attack Risk


Thal Rate
Less Heart Attack Risk
More Heart Attack Risk

In the graphs above, a green distribution shows the group that has less Heart Attack Rate, and a red distribution shows the group that has a more Heart Attack Rate. Specifically, we see a very strong separation in the age category where people that are older shows a lesser chance of having an heart attack. This may seem unintuitive at first become we have been linking old age and heart attack all along. Perhaps it is because the specific group of younger people in the data set that has other features of showing heart attack.

In the case of resting blood pressure and fasting blood pressure, we see little difference between the two groups, which implies that they have little contribution in determining the output.

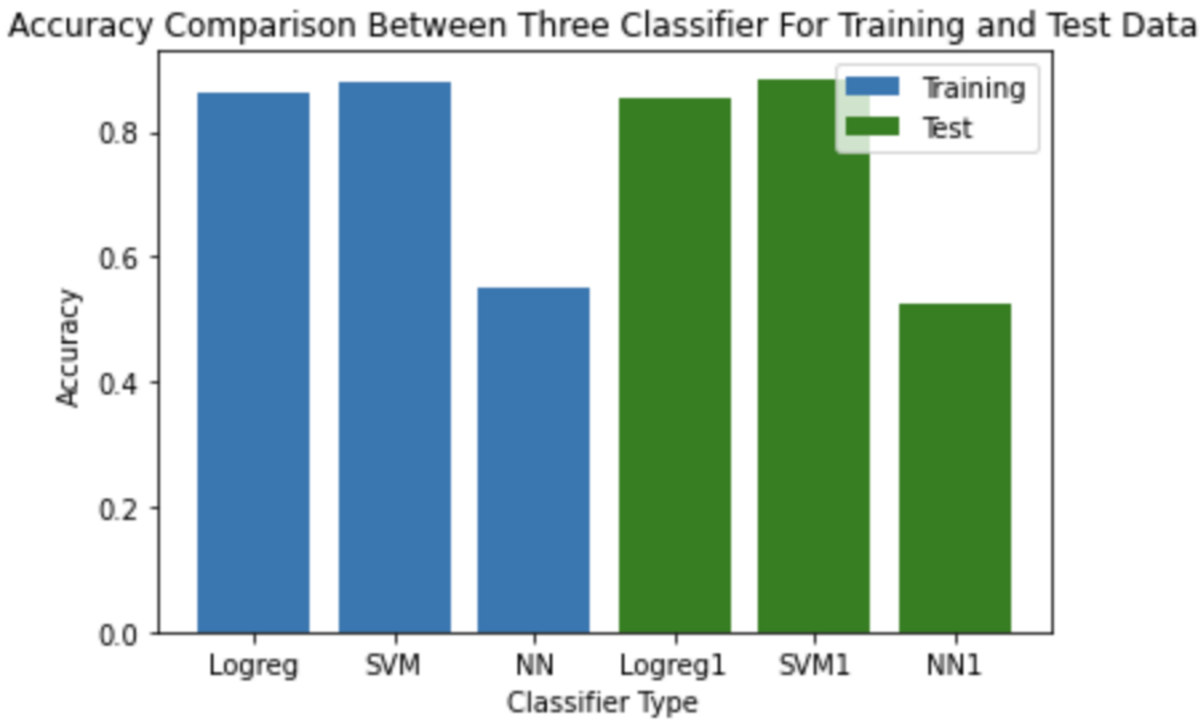
For data that shows a correlation, being female shows higher chance of heart attack; a higher Cholesterol shows a minor increased chance of having less Heart Attack Rate; having chest pain other than type 1 shows significant correlation for more Heart Attack Rate; resting electrocardiographic results of Value 1 shows more chance of Heart Attack Rate; not having exercise induced angina shows a correlation to having more Heart Attack Rate; higher maximum heart rate achieved shows significant correlation to higher Heart Attack Rate; having 0 previous peak shows strong correlation to more Heart Attack Rate; a slope value of 3 shows a correlation to higher Heart Attack rate; having less number of Vessels shows significant correlation to more Heart Attack Rate, and having less Thal Rate shows a strong correlation to more Heart Attack Rate.

Due to fact that resting/fasting blood pressure has little to do with the output, we could potentially eliminate them in the training of our models. And for other features that shows a strong or significant correlation, we could manually adjust them to have more initial weights and thus could significantly reduce the time to train.

Regular Models

Using the sklearn library, the code is run through the library provided functions for logistic regression, svms, and neural networks with essentially no regularization by giving it a high C value. The SVM is using a linear kernel, whereas for the neural network it has two hidden layers with 10 neurons each.

The dataset is split into a training set and test set. The performance of the classifiers are given by the bar graph below, which depicts the accuracy when run against the training set and the test set.



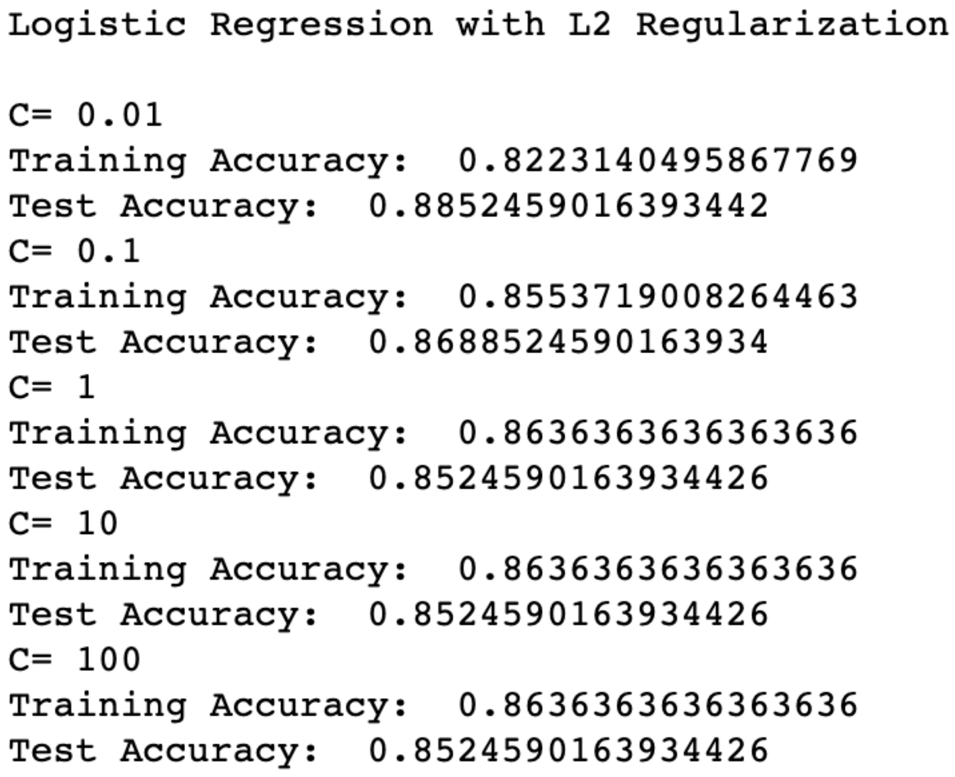
As shown by the graph, SVMs have the highest accuracy across both training and data sets. The similar accuracies between training data and test data maybe suggests that no overfitting has occurred, which is true across all three classifiers.
The logistic regression model follows closely behind, being only slightly outperformed whereas the neural network model seems to perform the worst, with barely 60% accuracy even for the training set.

Modifications to Classifiers

Modifications to Logistic Regression Model

Adding L2 Regularization

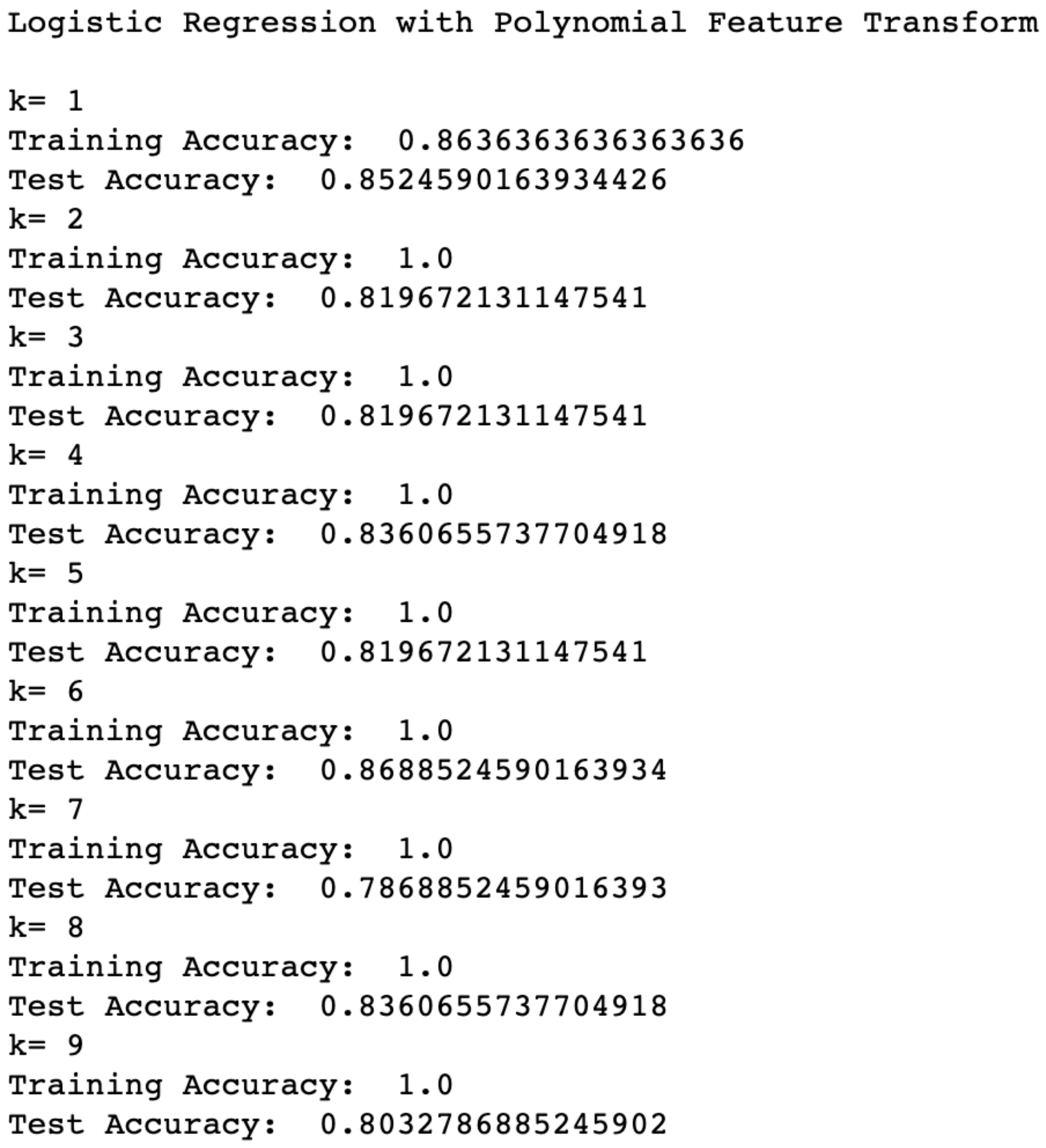
Adding L2 regularization to the logistic regression model seems to improve accuracy as c decreases. In our case when c=0.01 ($\lambda = 100$), the test data accuracy is the highest, at almost 89%. This may suggest that our unregularized model (arbitrarily high C) may be overfitting, and thus controlling the weights brings down the inaccuracy of the model.



```
Logistic Regression with L2 Regularization

C= 0.01
Training Accuracy:  0.8223140495867769
Test Accuracy:  0.8852459016393442
C= 0.1
Training Accuracy:  0.8553719008264463
Test Accuracy:  0.8688524590163934
C= 1
Training Accuracy:  0.8636363636363636
Test Accuracy:  0.8524590163934426
C= 10
Training Accuracy:  0.8636363636363636
Test Accuracy:  0.8524590163934426
C= 100
Training Accuracy:  0.8636363636363636
Test Accuracy:  0.8524590163934426
```

Polynomial Feature Transformation

Performing polynomial feature transformations also yield improvements, but not as much as using L2 regularization. Polynomial transformation up to orders of 9 were tested, and is tested with logistic regression (without regularization). Test data accuracy fluctuates as k increases, but peaks at $k = 6$ when test accuracy equals 86.9%, which is an improvement over the untransformed features.

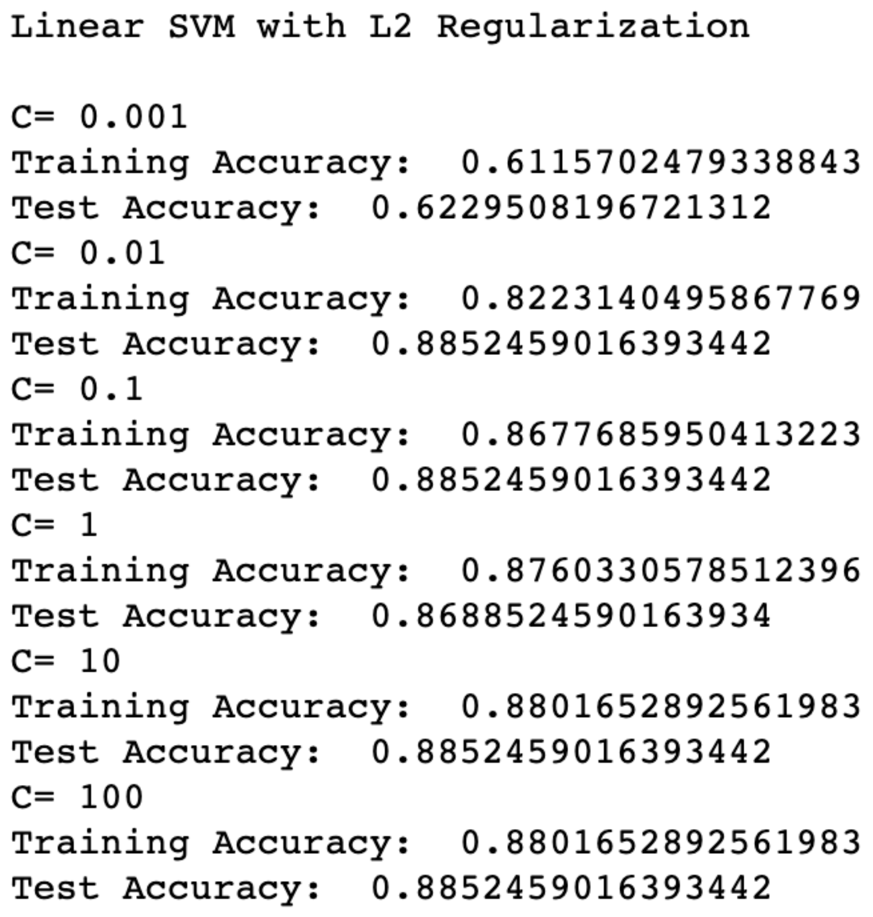


```
Logistic Regression with Polynomial Feature Transform

k= 1
Training Accuracy:  0.8636363636363636
Test Accuracy:  0.8524590163934426
k= 2
Training Accuracy:  1.0
Test Accuracy:  0.819672131147541
k= 3
Training Accuracy:  1.0
Test Accuracy:  0.819672131147541
k= 4
Training Accuracy:  1.0
Test Accuracy:  0.8360655737704918
k= 5
Training Accuracy:  1.0
Test Accuracy:  0.819672131147541
k= 6
Training Accuracy:  1.0
Test Accuracy:  0.8688524590163934
k= 7
Training Accuracy:  1.0
Test Accuracy:  0.7868852459016393
k= 8
Training Accuracy:  1.0
Test Accuracy:  0.8360655737704918
k= 9
Training Accuracy:  1.0
Test Accuracy:  0.8032786885245902
```

Modifications to SVM Model

Adding L2 Regularization

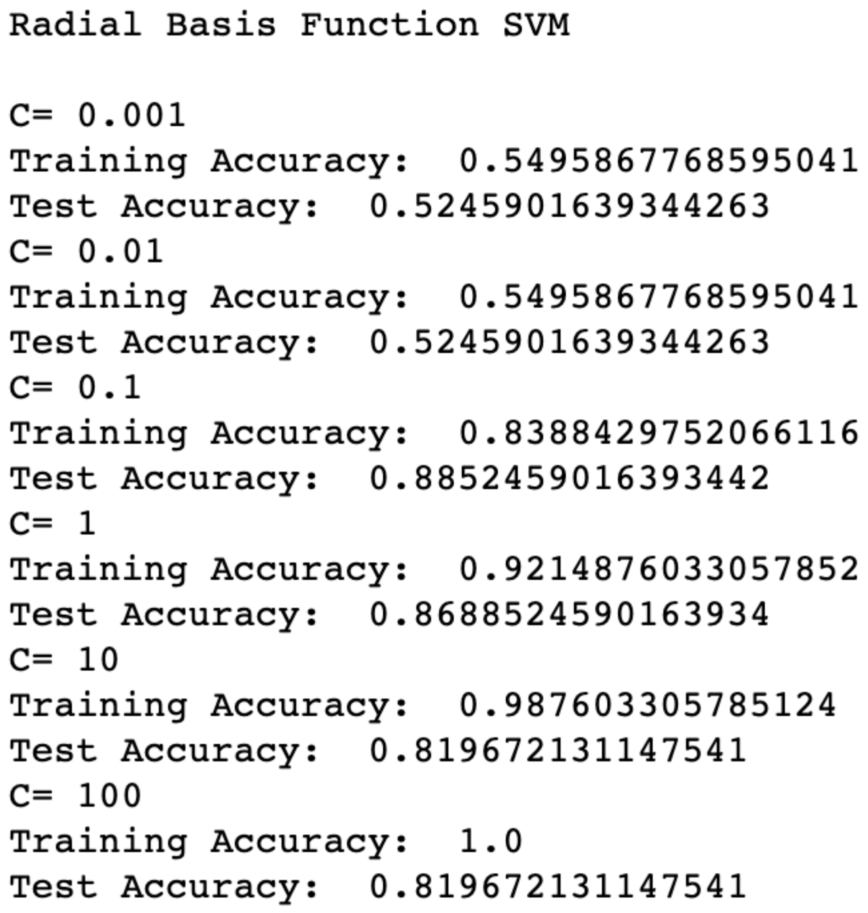
Adding L2 regularization to the Linear SVM model does not seem to change much. In fact, high regularization seems to make the model perform worse. This is expected, since having a very small margin essentially makes it a hard-margin SVM. This reveals something about the data set, in that it shows the dataset is not linearly separable. Thus, applying different kernels to the SVM may yield better results.



```
Linear SVM with L2 Regularization

C= 0.001
Training Accuracy:  0.6115702479338843
Test Accuracy:  0.6229508196721312
C= 0.01
Training Accuracy:  0.8223140495867769
Test Accuracy:  0.8852459016393442
C= 0.1
Training Accuracy:  0.8677685950413223
Test Accuracy:  0.8852459016393442
C= 1
Training Accuracy:  0.8760330578512396
Test Accuracy:  0.8688524590163934
C= 10
Training Accuracy:  0.8801652892561983
Test Accuracy:  0.8852459016393442
C= 100
Training Accuracy:  0.8801652892561983
Test Accuracy:  0.8852459016393442
```

Radial Basis SVM

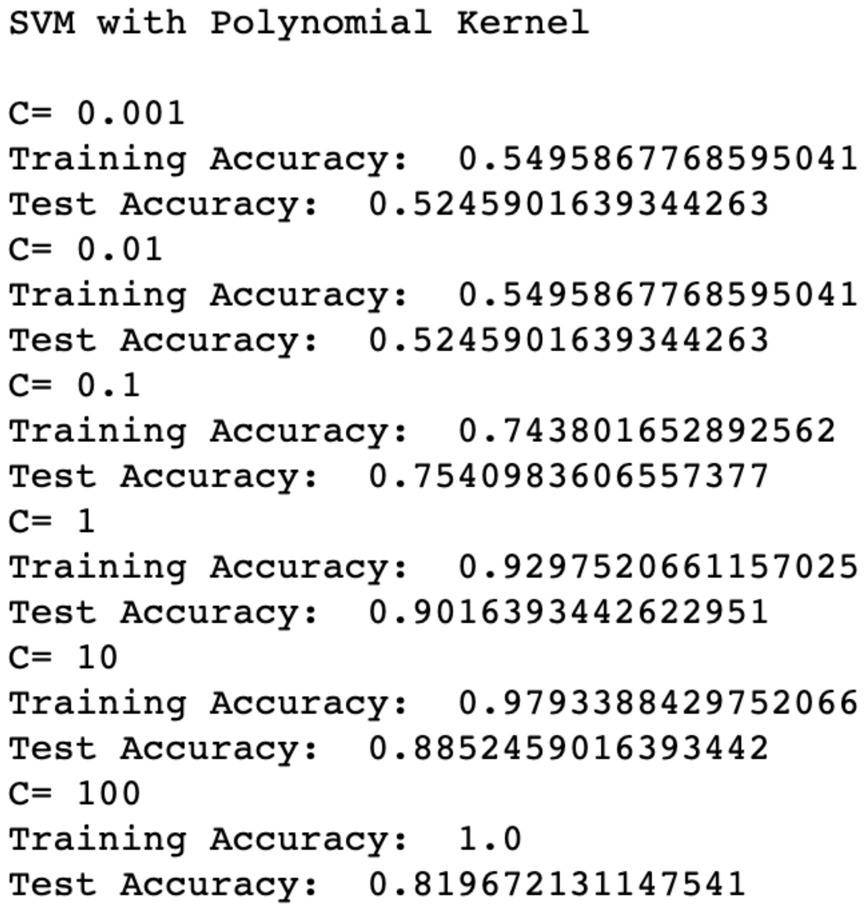
RBF SVM does not seem to yield any improvements over linear SVM. However, something of note is that RBF SVM does have very high training accuracy, whereas test accuracy is more lackluster. This may suggest that RBF SVM may actually be overfitting.



```
Radial Basis Function SVM

C= 0.001
Training Accuracy:  0.5495867768595041
Test Accuracy:  0.5245901639344263
C= 0.01
Training Accuracy:  0.5495867768595041
Test Accuracy:  0.5245901639344263
C= 0.1
Training Accuracy:  0.8388429752066116
Test Accuracy:  0.8852459016393442
C= 1
Training Accuracy:  0.9214876033057852
Test Accuracy:  0.8688524590163934
C= 10
Training Accuracy:  0.987603305785124
Test Accuracy:  0.819672131147541
C= 100
Training Accuracy:  1.0
Test Accuracy:  0.819672131147541
```

Polynomial SVM

The polynomial yields the best test accuracy result ($\tilde{9}0\%$) out of all the models tested. This may suggest that the polynomial kernel SVM may be the best fit model out of the three tested. However, note that when regularization is high (low C values) both training and test accuracy were low. This may suggest that the data is not linearly separable, and that a hard-margin SVM solution will not likely be possible.

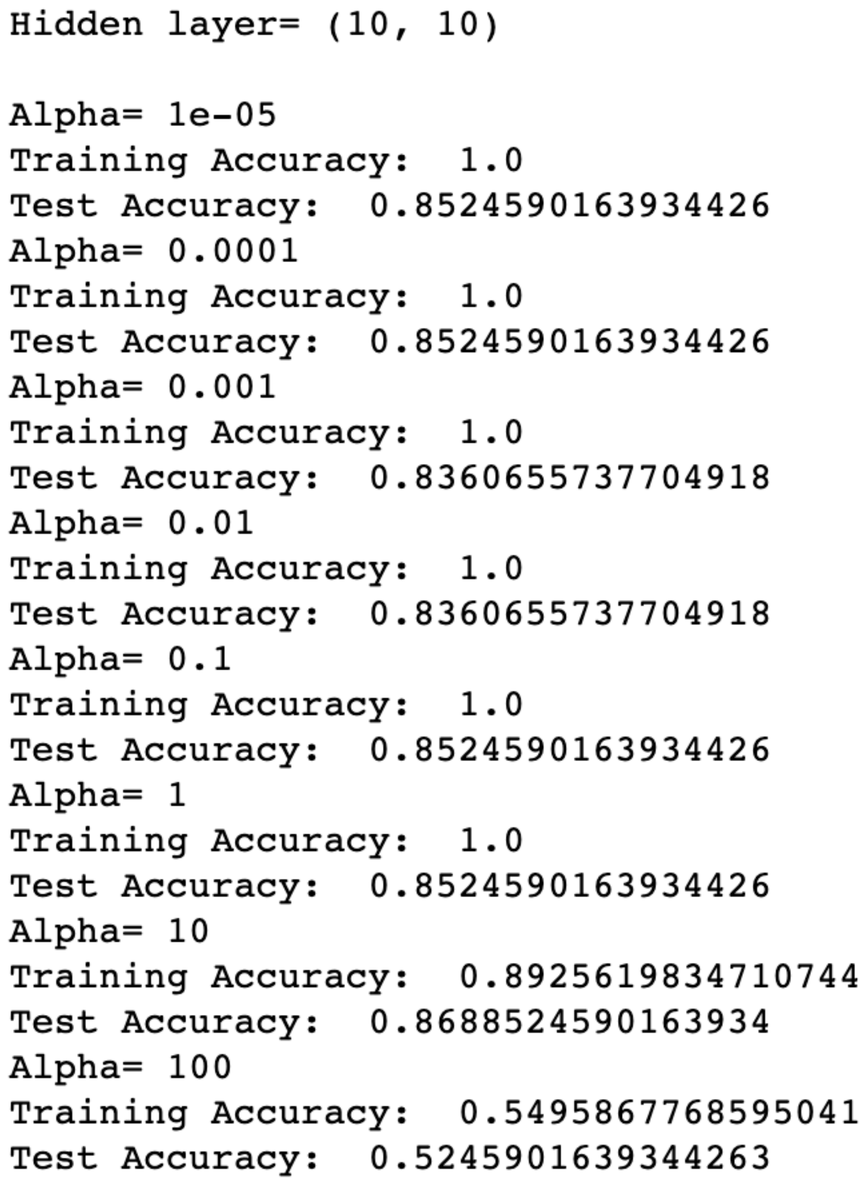


```
SVM with Polynomial Kernel

C= 0.001
Training Accuracy:  0.5495867768595041
Test Accuracy:  0.5245901639344263
C= 0.01
Training Accuracy:  0.5495867768595041
Test Accuracy:  0.5245901639344263
C= 0.1
Training Accuracy:  0.743801652892562
Test Accuracy:  0.7540983606557377
C= 1
Training Accuracy:  0.9297520661157025
Test Accuracy:  0.9016393442622951
C= 10
Training Accuracy:  0.9793388429752066
Test Accuracy:  0.8852459016393442
C= 100
Training Accuracy:  1.0
Test Accuracy:  0.819672131147541
```

Modifications to Neural Network Model

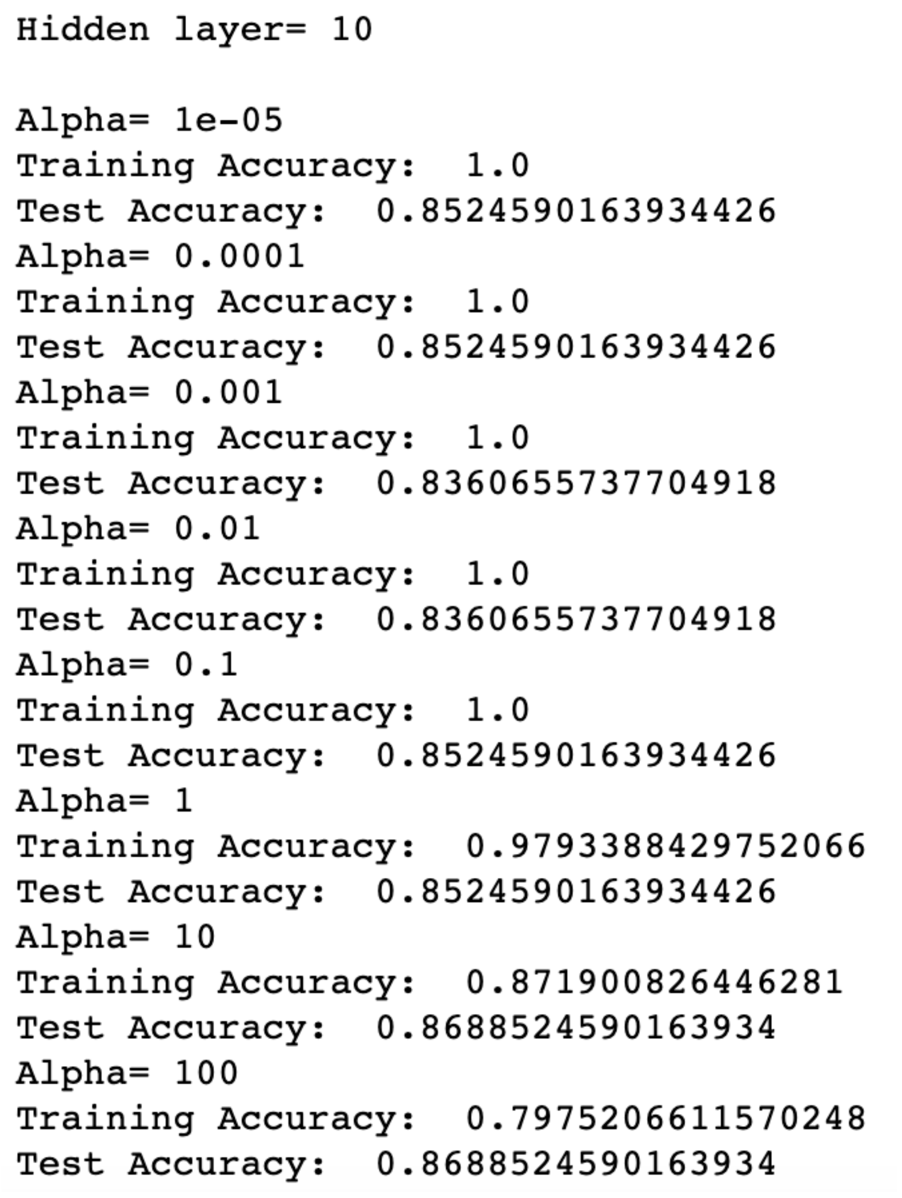
Adding L2 Regularization and Modifying Number of Layers

Recall that the original setting is two layers with ten neurons each (10, 10). Adding regularization seems to have had a big impact, which suggests that without regularization, the data set may have been overfitting.

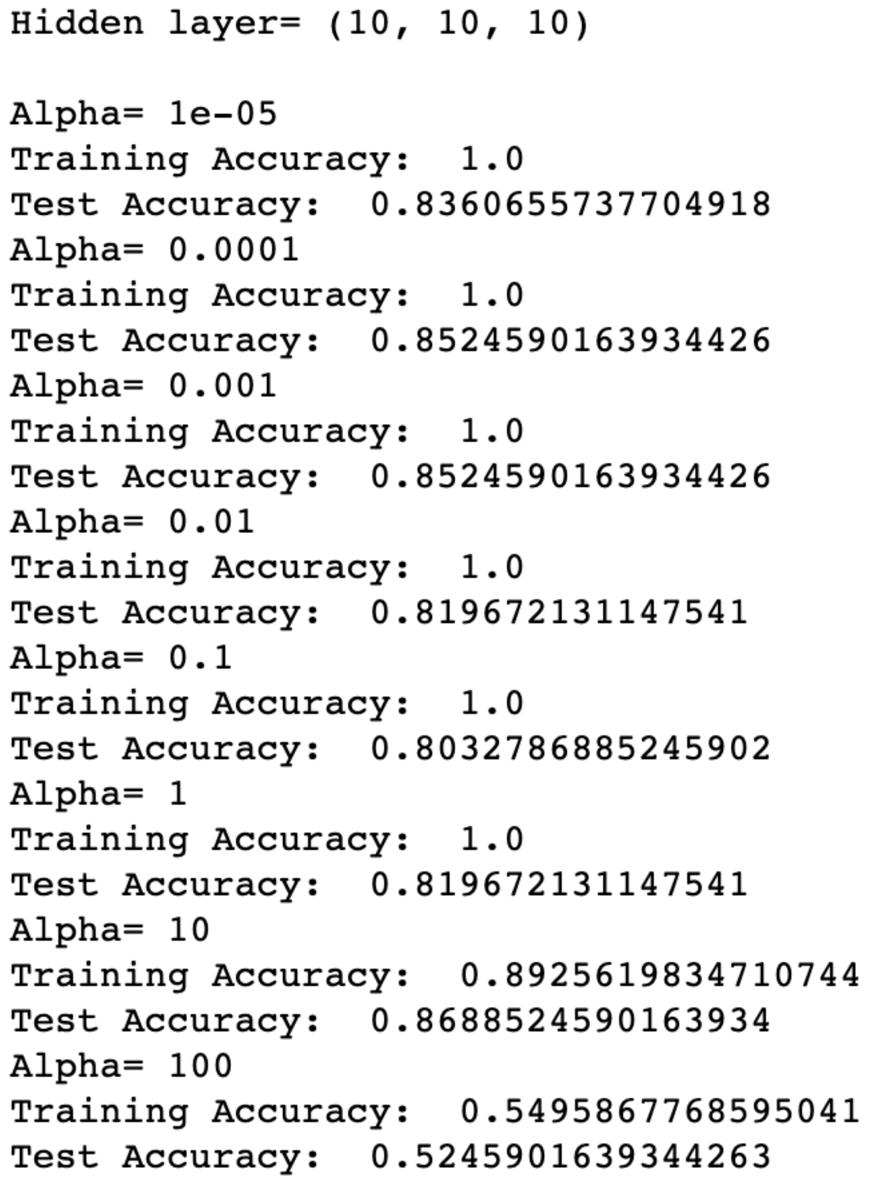


```
Hidden layer= (10, 10)

Alpha= 1e-05
Training Accuracy:  1.0
Test Accuracy:  0.8524590163934426
Alpha= 0.0001
Training Accuracy:  1.0
Test Accuracy:  0.8524590163934426
Alpha= 0.001
Training Accuracy:  1.0
Test Accuracy:  0.8360655737704918
Alpha= 0.01
Training Accuracy:  1.0
Test Accuracy:  0.8360655737704918
Alpha= 0.1
Training Accuracy:  1.0
Test Accuracy:  0.8524590163934426
Alpha= 1
Training Accuracy:  1.0
Test Accuracy:  0.8524590163934426
Alpha= 10
Training Accuracy:  0.8925619834710744
Test Accuracy:  0.8688524590163934
Alpha= 100
Training Accuracy:  0.5495867768595041
Test Accuracy:  0.5245901639344263
```

That said, adding/reducing layers does not seem to have much effect on the performance of the neural network. However, something of note is that when the neural network has only one hidden layer, the test accuracy for low regularization is significantly better when compared to neural networks with two and three layers, which may suggest that having more than one-layer may make regularization more important.



```
Hidden layer= 10

Alpha= 1e-05
Training Accuracy:  1.0
Test Accuracy:  0.8524590163934426
Alpha= 0.0001
Training Accuracy:  1.0
Test Accuracy:  0.8524590163934426
Alpha= 0.001
Training Accuracy:  1.0
Test Accuracy:  0.8360655737704918
Alpha= 0.01
Training Accuracy:  1.0
Test Accuracy:  0.8360655737704918
Alpha= 0.1
Training Accuracy:  1.0
Test Accuracy:  0.8524590163934426
Alpha= 1
Training Accuracy:  0.9793388429752066
Test Accuracy:  0.8524590163934426
Alpha= 10
Training Accuracy:  0.871900826446281
Test Accuracy:  0.8688524590163934
Alpha= 100
Training Accuracy:  0.7975206611570248
Test Accuracy:  0.8688524590163934
```



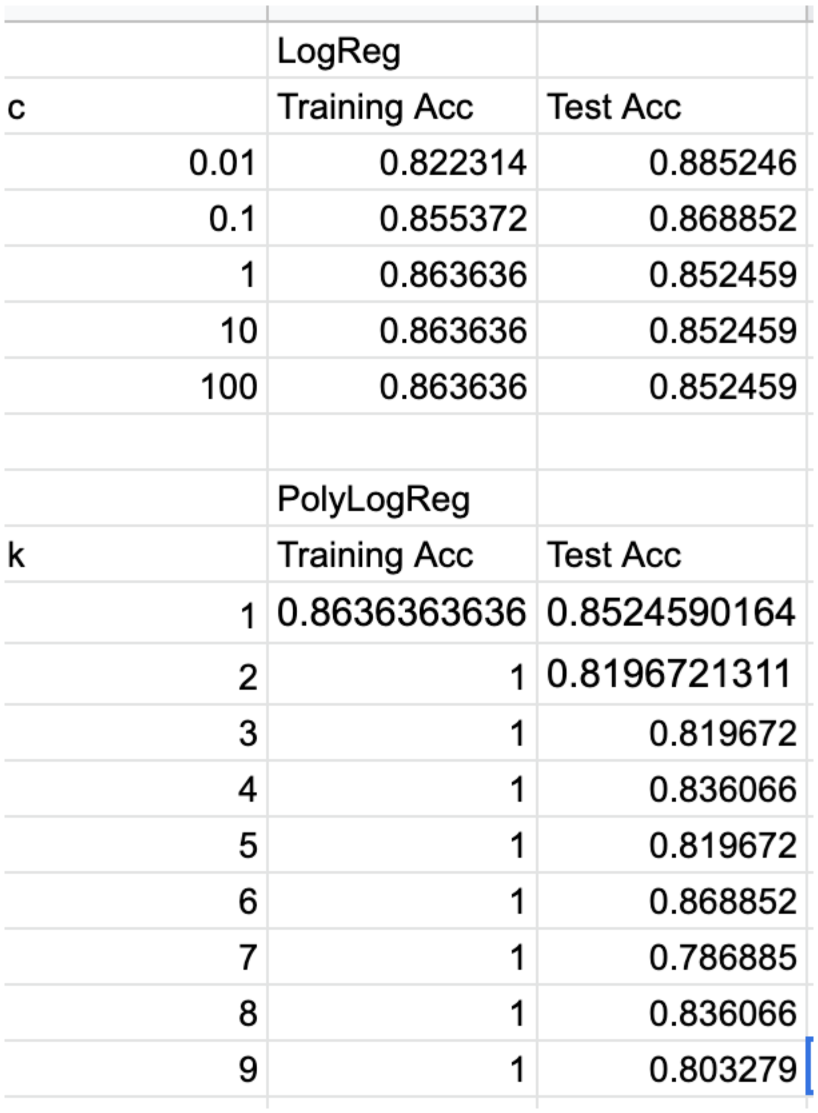
```
Hidden layer= (10, 10, 10)

Alpha= 1e-05
Training Accuracy:  1.0
Test Accuracy:  0.8360655737704918
Alpha= 0.0001
Training Accuracy:  1.0
Test Accuracy:  0.8524590163934426
Alpha= 0.001
Training Accuracy:  1.0
Test Accuracy:  0.8524590163934426
Alpha= 0.01
Training Accuracy:  1.0
Test Accuracy:  0.819672131147541
Alpha= 0.1
Training Accuracy:  1.0
Test Accuracy:  0.8032786885245902
Alpha= 1
Training Accuracy:  1.0
Test Accuracy:  0.819672131147541
Alpha= 10
Training Accuracy:  0.8925619834710744
Test Accuracy:  0.8688524590163934
Alpha= 100
Training Accuracy:  0.5495867768595041
Test Accuracy:  0.5245901639344263
```

Conclusion

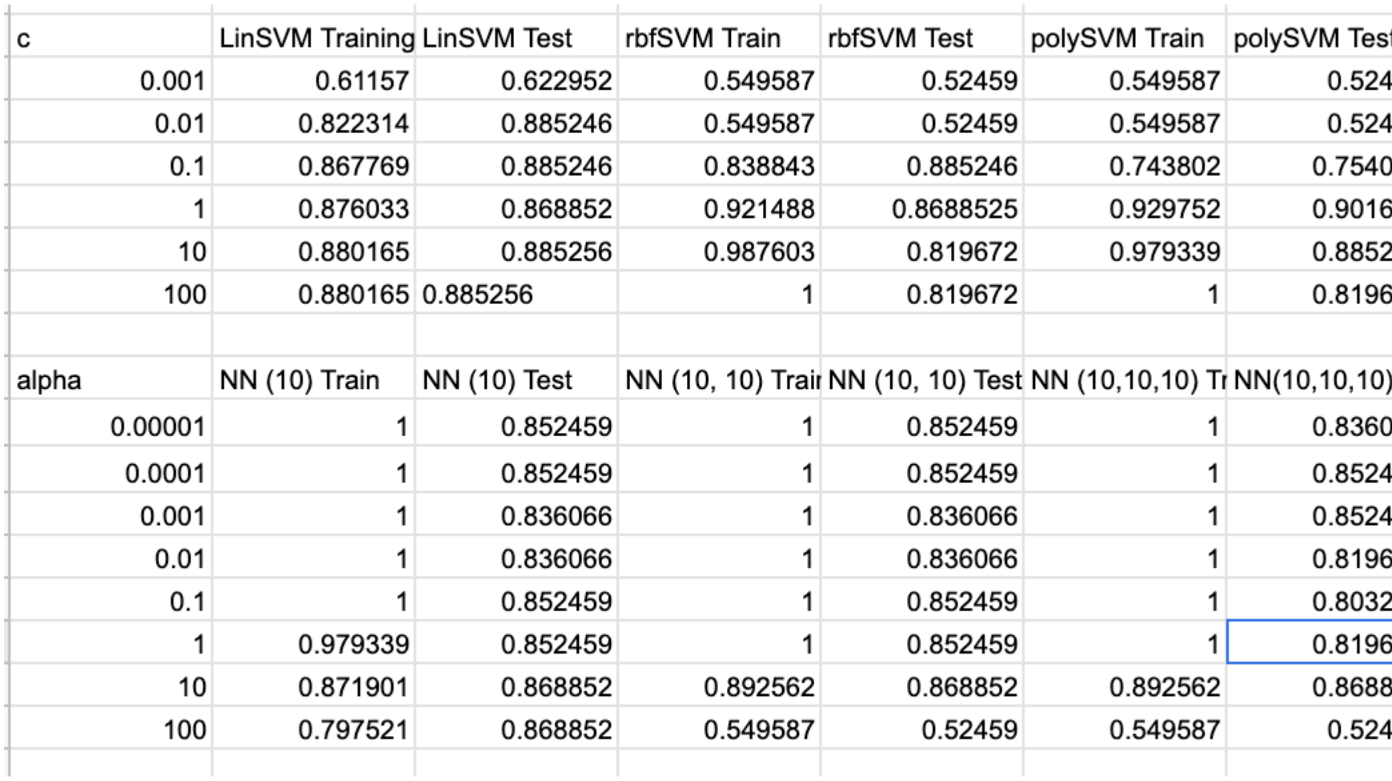
Polynomial Kernel SVMs have the best performance overall, with moderate Regularization (λ=1). In fact, even with Linear SVMs and no other special modifications to the other classifiers, SVMs still perform better overall, with Logistic Regression models following closely behind. Therefore this result may suggest that the dataset may in fact be separable. However, decreasing c (increasing λ) in SVMs yielded worse performing models, thus we conclude that although results seem to suggest that the data is separable, it is not perfectly so, and thus cannot be solved by a hard-margin SVM. Therefore some leeway should be given to through less regularization to make it into a soft-margin SVM problem.

The mediocre performance of Neural Networks when regularization is low may suggest that having two-hidden layers already overfits the model, and thus regularization may be needed to penalize high weights. Apart from that, it may seem to suggest that having only one hidden layer would be sufficient for the neural network to perform similarly well to its counterparts with more layers, further bolstering the argument that neural networks have overfitted the model.

Table of Results



	LogReg	
c	Training Acc	Test Acc
0.01	0.822314	0.885246
0.1	0.855372	0.868852
1	0.863636	0.852459
10	0.863636	0.852459
100	0.863636	0.852459
	PolyLogReg	
k	Training Acc	Test Acc
1	0.8636363636	0.8524590164
2	1	0.8196721311
3	1	0.819672
4	1	0.836066
5	1	0.819672
6	1	0.868852
7	1	0.786885
8	1	0.836066
9	1	0.803279



c	LinSVM Training	LinSVM Test	rbfSVM Train	rbfSVM Test	polySVM Train	polySVM Tes
0.001	0.61157	0.622952	0.549587	0.52459	0.549587	0.524
0.01	0.822314	0.885246	0.549587	0.52459	0.549587	0.524
0.1	0.867769	0.885246	0.838843	0.885246	0.743802	0.7540
1	0.876033	0.868852	0.921488	0.8688525	0.929752	0.9016
10	0.880165	0.885256	0.987603	0.819672	0.979339	0.8852
100	0.880165	0.885256	1	0.819672	1	0.8196

alpha	NN (10) Train	NN (10) Test	NN (10, 10) Trai	NN (10, 10) Test	NN (10,10,10) Tr	NN(10,10,10)
0.00001	1	0.852459	1	0.852459	1	0.8360
0.0001	1	0.852459	1	0.852459	1	0.8524
0.001	1	0.836066	1	0.836066	1	0.8524
0.01	1	0.836066	1	0.836066	1	0.8196
0.1	1	0.852459	1	0.852459	1	0.8032
1	0.979339	0.852459	1	0.852459	1	0.8196
10	0.871901	0.868852	0.892562	0.868852	0.892562	0.8688
100	0.797521	0.868852	0.549587	0.52459	0.549587	0.524