# HOMEWORK 6

Jackson Hellmers
9075444662

11/15/21

## Section 1

### Solution 1.1

We are told that the diagonal entries of K are equal to 1 while any element off diagonal equals 0. This is an identity matrix so $v^T K v = v^T I v = v^T v = |v|^2$ since we are squaring the norm of v it must be greater than or equal to 0. Therefore $v^T K v \geq 0$ must be true.

### Solution 1.2

Again we know that our kernel matrix K is equal to identity. Since we are told that
$f(z) = \sum_{i=1}^{n} \alpha_i y_i k(z_i, z) + b$
If we pick $\alpha = 1$ and $b = 0$ then we can simplify to $f(z) = y_i$.
This means that for each point $y_i$ the decision boundary is $sgn(f(z)) = sgn(y_i)$.
So, each point in the set has its own decision boundary implying k creates an n dimension space.

### Solution 1.3

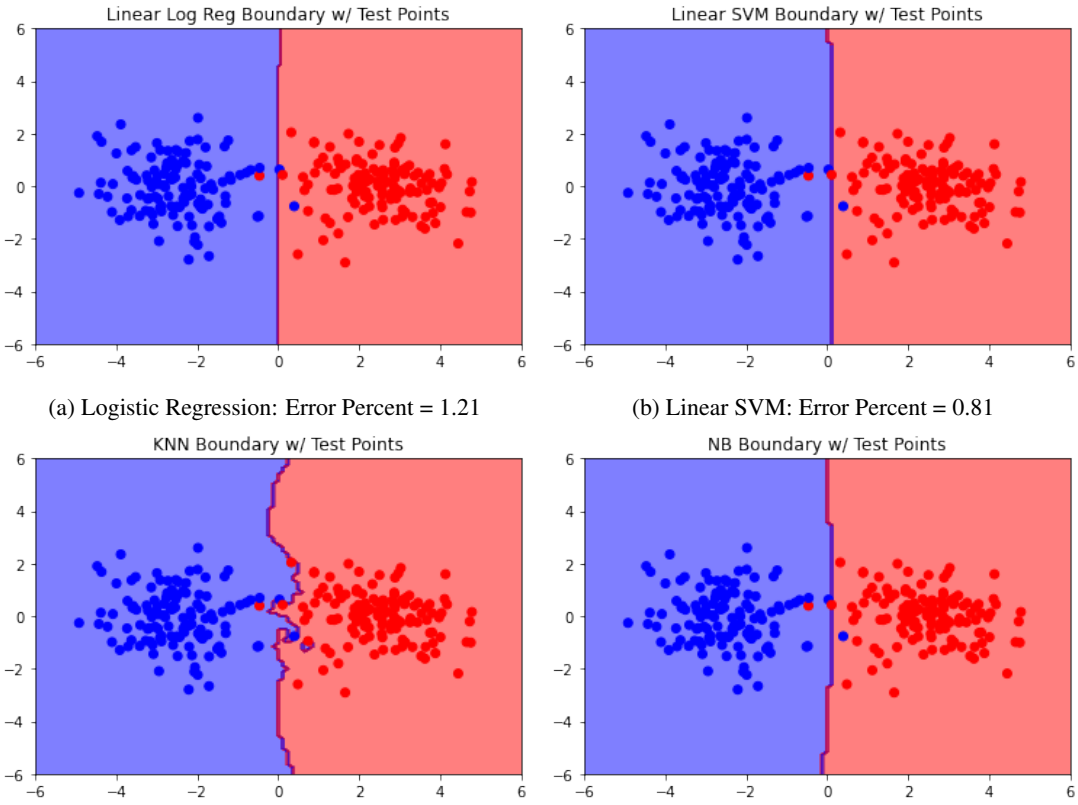
For z not in the dataset we know our kernel will take the value of 0 so $f(z) = b$.
This means that untrained test points will be classified based upon the classifier's bias term.

## Section 2

### Solution 2.1

Yes implementing linear SVM using a kernel is possible. In the dual problem for linear kernel SVM we find that our kernel $k(x, x') = \phi(x)\phi(x') = x^T x$ so as long our kernel is the dot product between x and itself we can implement linear SVM with a kernel.

## Solution 2.2.1




(a) Logistic Regression: Error Percent = 1.21

(b) Linear SVM: Error Percent = 0.81

(c) KNN k=3: Error Percent = 0.81

(d) Naive Bayes: Error Percent = 1.21

Figure 1: Decision Boundaries for Linearly Separable Data

These decision boundaries all use the same train data and have the test data scatter plot on top. $\mu = 2.4$

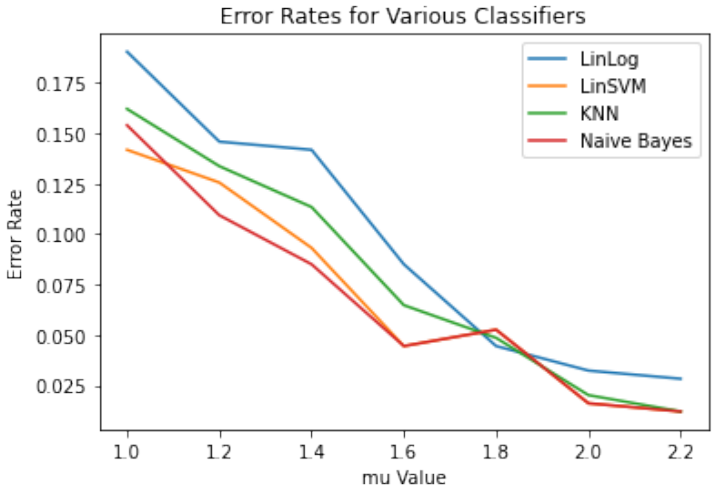
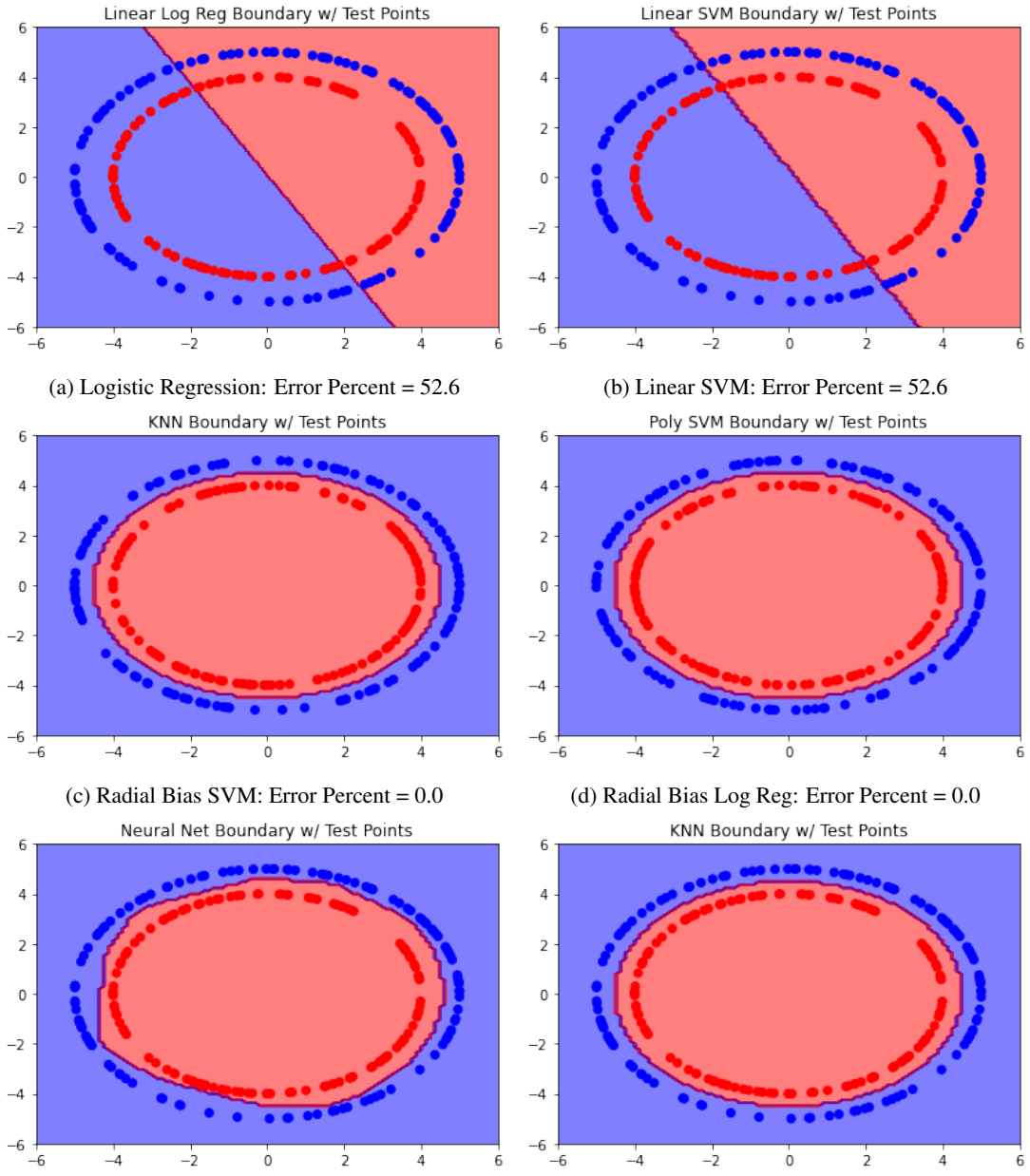



Figure 2: Classifier error rates for various $\mu$ values

We can clearly see that as the value of $\mu$ increases the accuracy of the classifiers increase. This is because there is less overlap in data points and the classes become more separated.

## Solution 2.2.2




(a) Logistic Regression: Error Percent = 52.6

(b) Linear SVM: Error Percent = 52.6

(c) Radial Bias SVM: Error Percent = 0.0

(d) Radial Bias Log Reg: Error Percent = 0.0

(e) Neural Network: Error Percent = 0.0

(f) KNN k=1: Error Percent = 0.0

Figure 3: Decision Boundaries for Nonlinear Data

In order to effectively classify data that is no longer linearly separable in 2 dimensions we must use the kernel trick to transform the data into a feature space that allows for linear separability.

This also shows the strength of radial bias function which is inherently circular as it outperformed even high dimensional polynomial kernels when using Kernel SVM and Logistic Regression.

I believe that these graphs also show how powerful a simple learning algorithm like KNN can be even when compared to something as complex as a kernel SVM.

## Solution 2.3

| Classifier | Error Rate | Parameter |
|---|---|---|
| Linear SVM | 0.0106 | C=10 |
| Linear LR | 0.0532 | N/A |
| Poly SVM | 0.0106 | Degree=2, C=10000 |
| Poly SVM | 0.0212 | Degree=3, C=1000 |
| Poly SVM | 0.0319 | Degree=4, C=1000 |
| RBF SVM | 0.0212 | Sigma=1, C=1000 |
| KNN | 0.0638 | k=9 |
| Neural Net | 0.0319 | Hidden Size=64 |

Table 1: Error Rate Table

I found that SVM classifiers deploying low dimension feature spaces (linear and 2nd degree polynomial) performed best. This would imply that as the dimensionality increases we begin to overfit the training data.

I then went and added L1 regularization to my SVM and trained the Linear SVM with various values for the regularization constant 'C'. I used the validation set to find the optimal value and then trained a final model. Looking at the weight coefficients of the trained model we know that any non-zero weights account for some portion of the variance in the data as L1 promotes sparsity. Checking these against the feature names I was able to extract that the following features are the most important in determining the tumor type.

- mean smoothness
- mean compactness
- mean concavity
- mean concave points
- mean symmetry
- worst compactness
- worst concave points
- worst symmetry