Team Members:
Alex Nguyen : 1879171- (SVM)
Abibou Mbodji : 1931937 - (SVM)
Xena Toumajian : 1662518 - (KNN)
Alan Dileep : 2132503 - (KNN)

Group Project Final Report

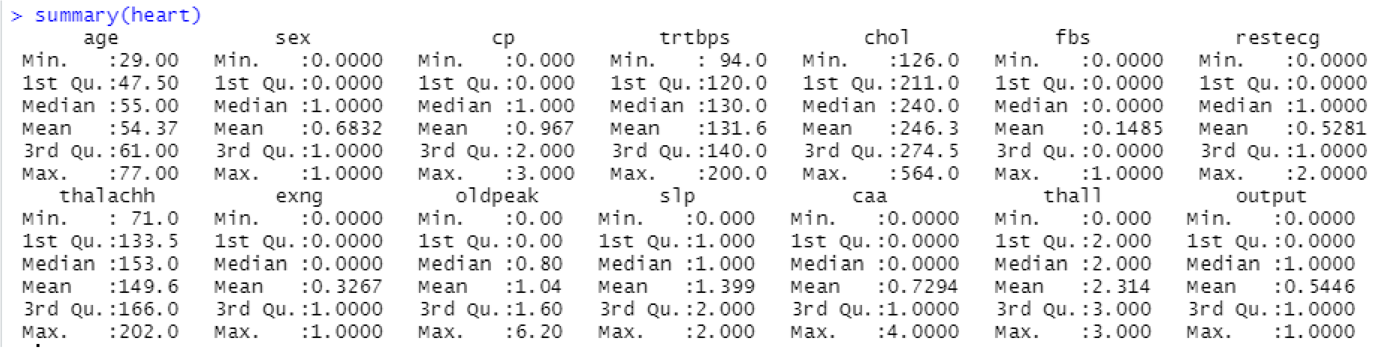
Introduction

Heart attacks are one of the leading causes of death in the United States, causing 1 in every 4 deaths. According to the CDC, one in five heart attacks are silent. Therefore it is important to be able to accurately predict whether or not a person is at risk of heart disease. One of the primary methods of finding out if a person is having a heart attack is an angiogram, which basically reveals blockages due to narrowing of the blood vessels. We use the dataset "heart" which contains 303 observations and 14 variables, including "output", our response variable.

The main goal we want to achieve through our data is accurately predicting whether a patient is more or less prone to a heart attack based on various predictors.

Our response variable "output" specifies whether or not there is narrowing of blood vessels which is directly related to the susceptibility of heart disease.

The predictor variables that we use in out data are - age, sex, chess pain type(cp), resting blood pressure(trestbps), cholesterol(chol), fasting blood sugar(fbs), resting electrocardiographic results(restecg), maximum heart rate achieved(thalach), exercise induced angina(exang), ST depression induced by exercise (oldpeak), the slope of the peak exercise ST segment(slope), number of major vessels colored by fluoroscopy(ca), and thallium stress test(thal).



```
> summary(heart)
      age             sex               cp            trtbps           chol            fbs            restecg      
 Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0   Min.   :126.0   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0   1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :55.00   Median :1.0000   Median :1.000   Median :130.0   Median :240.0   Median :0.0000   Median :1.0000  
 Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6   Mean   :246.3   Mean   :0.1485   Mean   :0.5281  
 3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0   3rd Qu.:274.5   3rd Qu.:0.0000   3rd Qu.:1.0000  
 Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0   Max.   :564.0   Max.   :1.0000   Max.   :2.0000  
    thalachh          exng           oldpeak          slp             caaa           thall           output      
 Min.   : 71.0   Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000  
 1st Qu.:133.5   1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:0.0000  
 Median :153.0   Median :0.0000   Median :0.80   Median :1.000   Median :0.0000   Median :2.000   Median :1.0000  
 Mean   :149.6   Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294   Mean   :2.314   Mean   :0.5446  
 3rd Qu.:166.0   3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.0000  
 Max.   :202.0   Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000   Max.   :3.000   Max.   :1.0000  
```

Using supervised learning methods, we will use this dataset in order to fit a model which will accurately predict whether or not a person is at risk of heart disease.

Methodology

For this project we worked with supervised learning. We applied both KNN and SVM approaches to our data set in order to determine which method would provide us with the smallest test error.

KNN: For our KNN models, we will be using the leave one out cross validation for our set. When using leave one out cross validation, we split the data into two subsets; a training set and a testing set of just one observation. By using nearly the whole data set, this method produces a more stable test error estimate, for a more accurate prediction than the validation set approach. We will need to find the best K value that will use similar data to our x patient in question and provide a similar diagnosis. The only disadvantage to this method is that because we are using nearly the whole data set at each step it will be computationally demanding.

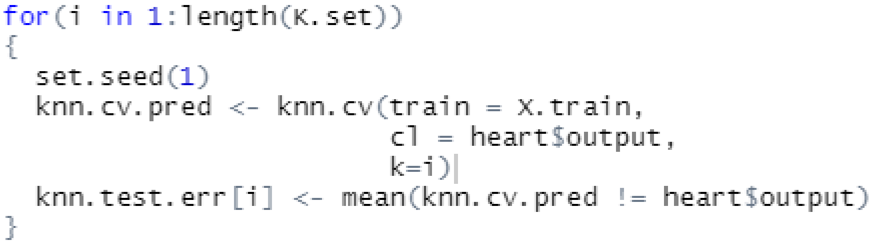
SVM: SVM will allow us to classify our patients in order to make a prediction using a hyperplane and allows for soft margins. We can try out a linear kernel but if the accuracy is not satisfactorily we will use a non−linear kernel like polynomial or radial in order to fit our SVM. The biggest advantage of using kernels rather than simply enlarging the feature space is computation. Kernels allow us to compute all distinct pairs of training observations within the original p−dimensional space. In addition, they are computed without needing to explicitly work in the enlarged feature space. If we were to work in this enlarged feature space, computations could become intractable due to how large many SVM applications are. If we were to work in a 10−dimensional space for example, we would end up with 76 features $(10 \times 3 + 10(10-1)/2 + 1))$.

The following are the hyperplane equations for each of the kernels that will be used:
Linear: $0 = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$
Polynomial: $0 = \beta_0 + \beta_1 X_1 + \beta_2 X^2_1 + \beta_3 X_2 + \beta 4 X^2_2 + \epsilon$
Radial: $K(x_i , x_j) = \exp(-\gamma \sum_{p,\, k=1} (x_{ik} - x_{jk})^2),\ \gamma > 0$

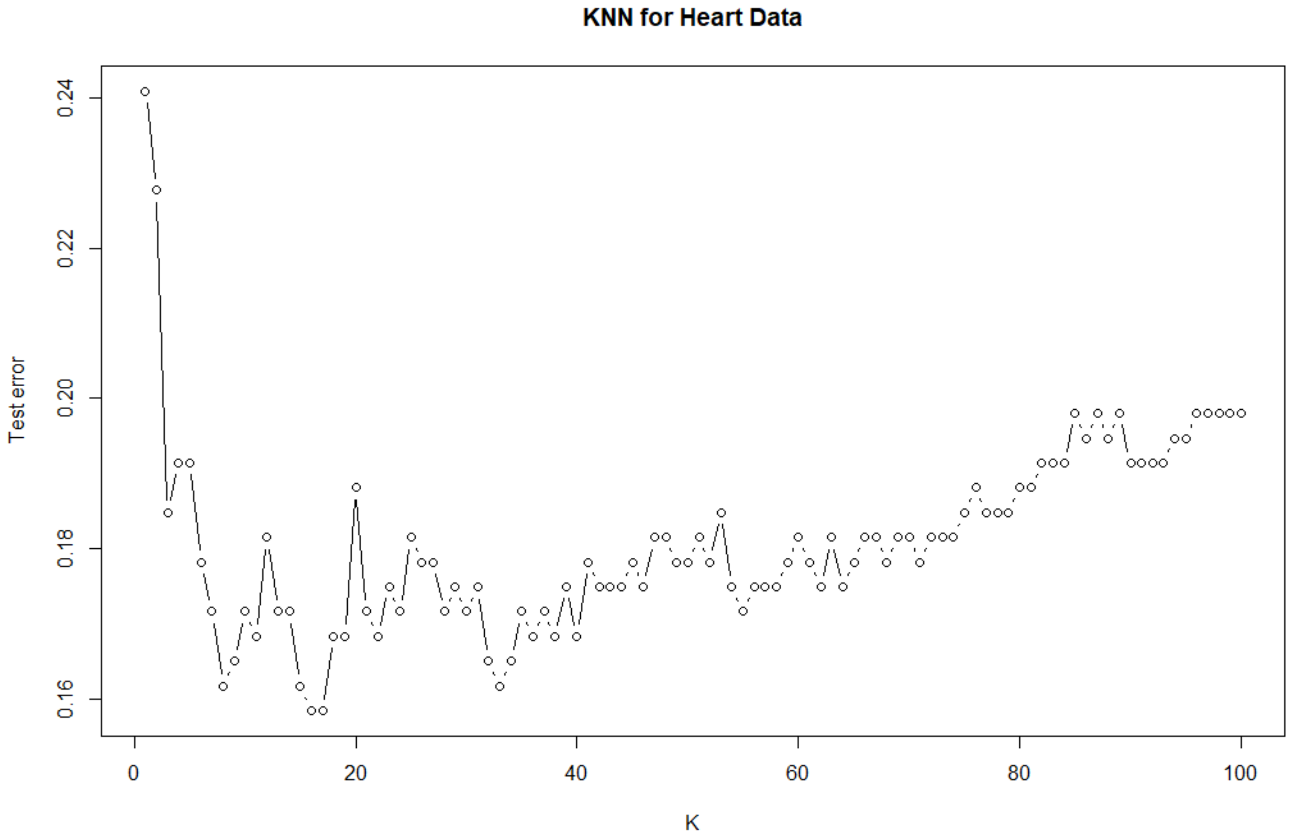
Data Analysis

A. We’ve decided not to exclude any particular predictor from our model. All of our predictors explore common symptoms of someone who is at risk of heart attack and hence it is necessary to include them in our data. We had to scale the data since some of our predictors like cholesterol and resting blood pressure were in the 100s while some other predictors like chest pain type and fasting blood sugar levels are categorical variables. KNN especially is sensitive to the scale of the data hence it is important we scale the data so we get to fit our data optimally.

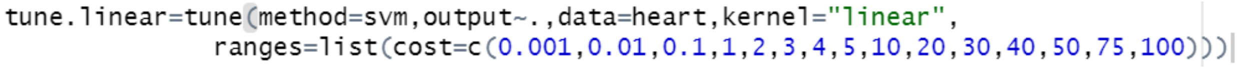
B. The optimal tuning parameters for KNN was a K value of 16. We determined this by applying the cross validation approach through K values of 1 through 100. We then determined which one would provide us with the smallest test error.



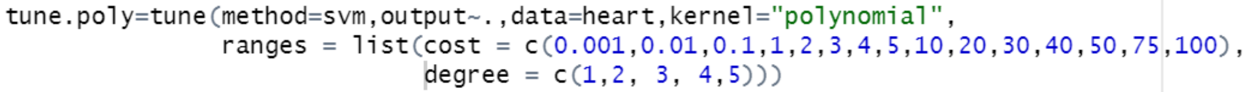
```
for(i in 1:length(K.set))
{
  set.seed(1)
  knn.cv.pred <- knn.cv(train = X.train,
                        cl = heart$output,
                        k=i)
  knn.test.err[i] <- mean(knn.cv.pred != heart$output)
}
```



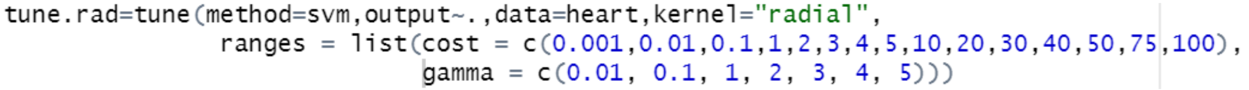
For SVM, we found the optimal parameters for linear, polynomial, and radial kernels. In the end, we found that our optimal parameter, a cost of 0.01, with a linear kernel worked best as it yielded the lowest misclassification error out of all three kernels. We used the following code to find the optimal parameters of each kernel:



```
tune.linear=tune(method=svm,output~.,data=heart,kernel="linear",
             ranges=list(cost=c(0.001,0.01,0.1,1,2,3,4,5,10,20,30,40,50,75,100)))
```



```
tune.poly=tune(method=svm,output~.,data=heart,kernel="polynomial",
               ranges = list(cost = c(0.001,0.01,0.1,1,2,3,4,5,10,20,30,40,50,75,100),
                             degree = c(1,2, 3, 4,5)))
```

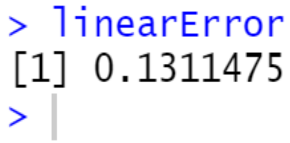


```
tune.rad=tune(method=svm,output~.,data=heart,kernel="radial",
               ranges = list(cost = c(0.001,0.01,0.1,1,2,3,4,5,10,20,30,40,50,75,100),
                             gamma = c(0.01, 0.1, 1, 2, 3, 4, 5)))
```

C.

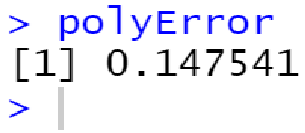
SVM Methods:

Linear: Our lowest error rate for linear came when we applied cost = 0.01 giving 0.1311475 test error



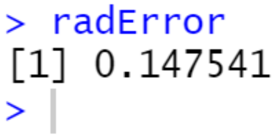
```
> linearError
[1] 0.1311475
> 
```

Polynomial: Our lowest error rate for polynomial kerne came when we applied cost = 30 and degree=1 giving 0.147541



```
> polyError
[1] 0.147541
> 
```

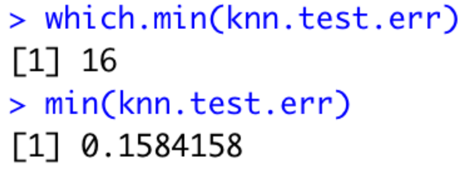
Radial: Our lowest error rate for radial kernel came when we applied cost=5 and gamma=0.01 giving 0.147541



```
> radError
[1] 0.147541
> 
```

KNN:

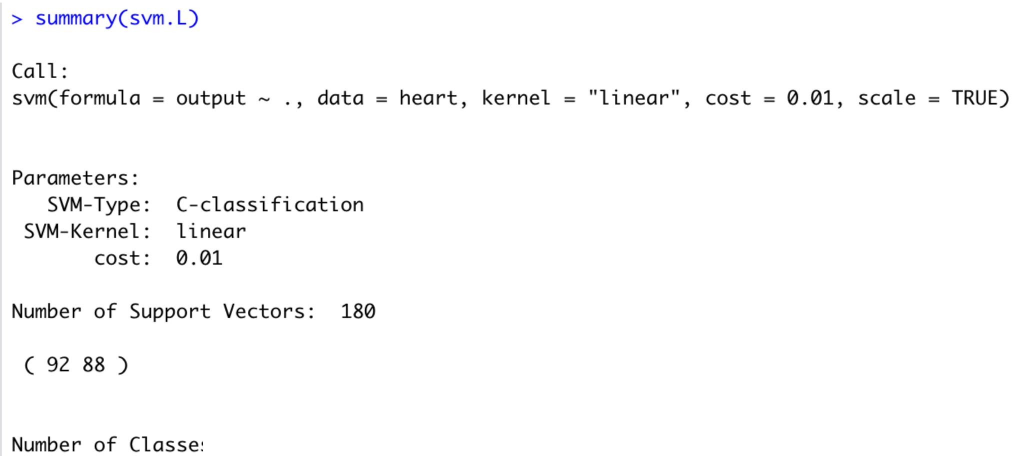
Our best test error came when we applied K=16, which provided us with a test error of 0.1584158.



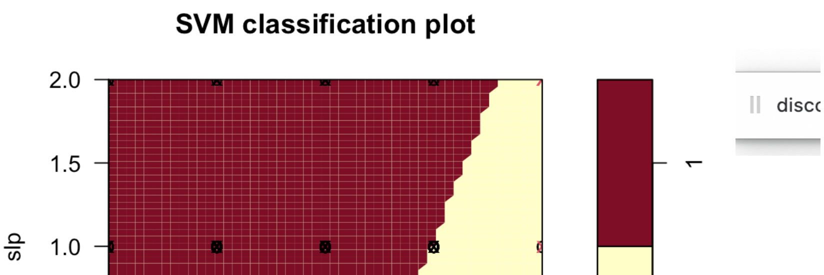
```
> which.min(knn.test.err)
[1] 16
> min(knn.test.err)
[1] 0.1584158
```

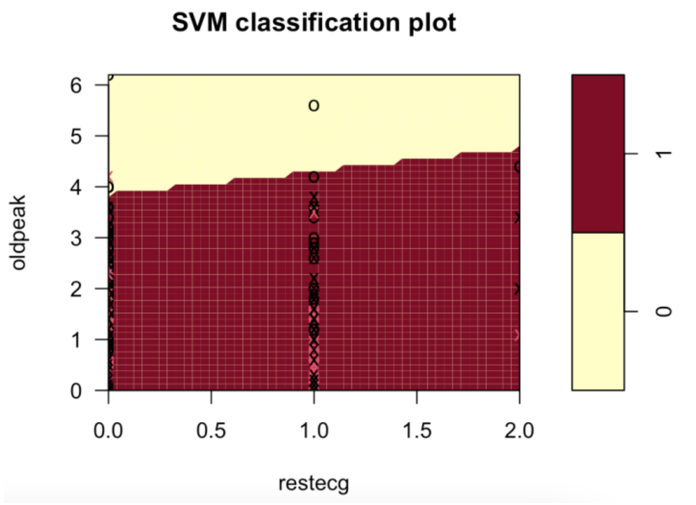
The best (lowest) error rate was produced from the Linear Kernel using the SVM method.

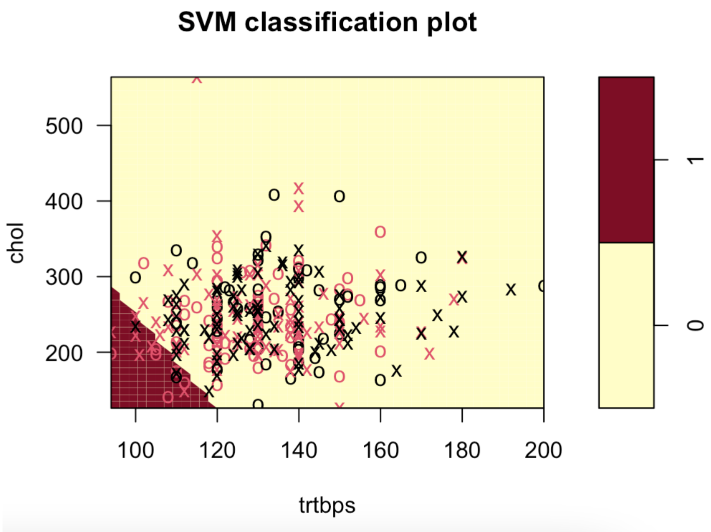
D.



```
> summary(svm.L)

Call:
svm(formula = output ~ ., data = heart, kernel = "linear", cost = 0.01, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  0.01

Number of Support Vectors:  180

 ( 92 88 )


Number of Classe:

Levels:
 0 1
```



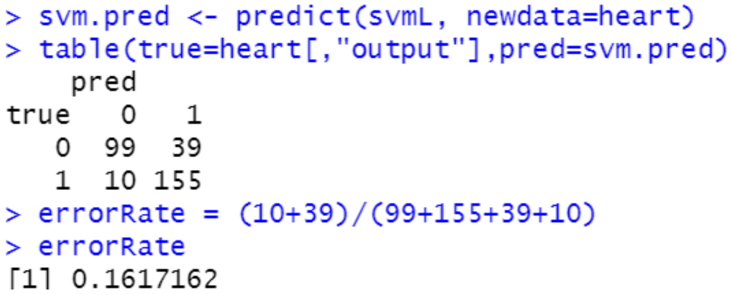




E.

Both SVM and KNN gave us models with low test error results and are reliable models which can predict the data we need. The difference in reliability of the two was minimal where the

KNN model gave us an error rate of 15.8% and the SVM model gave us an error rate of 13.1%. While these values are not 0 they are very good yields for prediction. When predicting using our best model, the SVM model with linear kernel for the full data set, the misclassification rate is 16.17%. This error rate is low, so we achieved a respectable predictive performance.



```
> svm.pred <- predict(svmL, newdata=heart)
> table(true=heart[,"output"],pred=svm.pred)
    pred
true   0   1
   0  99  39
   1  10 155
> errorRate = (10+39)/(99+155+39+10)
> errorRate
[1] 0.1617162
```

Conclusion

The goal of this project was to predict a heart attack with the highest possible accuracy. We have compared the models formed by using two supervised learning methods: KNN and SVM. For SVM, we have trained our data with linear, polynomial, and radial kernels, and the linear kernel yielded the best test error of 13.11%. We also used the LOOCV approach to make a KNN model that gave a testing error rate of 15.8%. Based on those results we selected SVM as the better approach and then we fitted a model using the entire dataset. We were able to successfully form a model with a low test error rate of 16.17% using the SVM approach.

Because of a significant amount of categorical variables in our data, we struggled with visualizing the data and interpreting it. The low amount of observations and the fact they are from a localized subset makes the model a little biased.

The easiest and most obvious way to improve our model would be to expand the dataset by adding in more observations from a more varied group. Finding more relevant predictors or fine tuning our current predictors would also help improve our model. However, simply increasing the number of observations would just make the data fitting process more computationally demanding. Some ways of combating that problem would be to use different fitting methods like K-fold approach over LOOCV for KNN.

References

Rahman, Rashik. (2021). *Heart Attack Analysis & Prediction Dataset* (Version 2) [Data set]. https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset