Data Science: Machine Learning

Samantha Scott[a]

[a]*Stellenbosch University, Cape Town, South Africa*

Keywords: Machine Learning, Heart Disease Prediction, Random Forests

Email address: 20945043@sun.ac.za (Samantha Scott)

Table of Contents

1 Introduction **3**

2 Research Question **3**

3 Data and Methodology **3**

4 Results and Discussion **3**

4.1 Random Forests 3

4.2 Support Vector Machine 9

5 Conclusion **10**

6 Reference List **10**

7 Appendix **10**

1. Introduction

The following paper is a comparison between two Machine Learning algorithms, namely Random Forests and Support Vector Machines, as prediction tools. Using a Linear Regression model as a baseline, the RMSE scores are compared.

2. Research Question

Problem type: supervised binomial classification

"Much like EDA, the ML process is very iterative and heurstic-based. With minimal knowledge of the problem or data at hand, it is difficult to know which ML method will perform best. This is known as the no free lunch theorem for ML (Wolpert 1996). Consequently, it is common for many ML approaches to be applied, evaluated, and modified before a final, optimal model can be determined. Performing this process correctly provides great confidence in our outcomes. If not, the results will be useless and, potentially, damaging.1"

"RMSE: Root mean squared error. This simply takes the square root of the MSE metric so that your error is in the same units as your response variable. If your response variable units are dollars, the units of MSE are dollars-squared, but the RMSE will be in dollars. Objective: minimize"

3. Data and Methodology

The data used in this investigation is heart disease data from Kaggle.

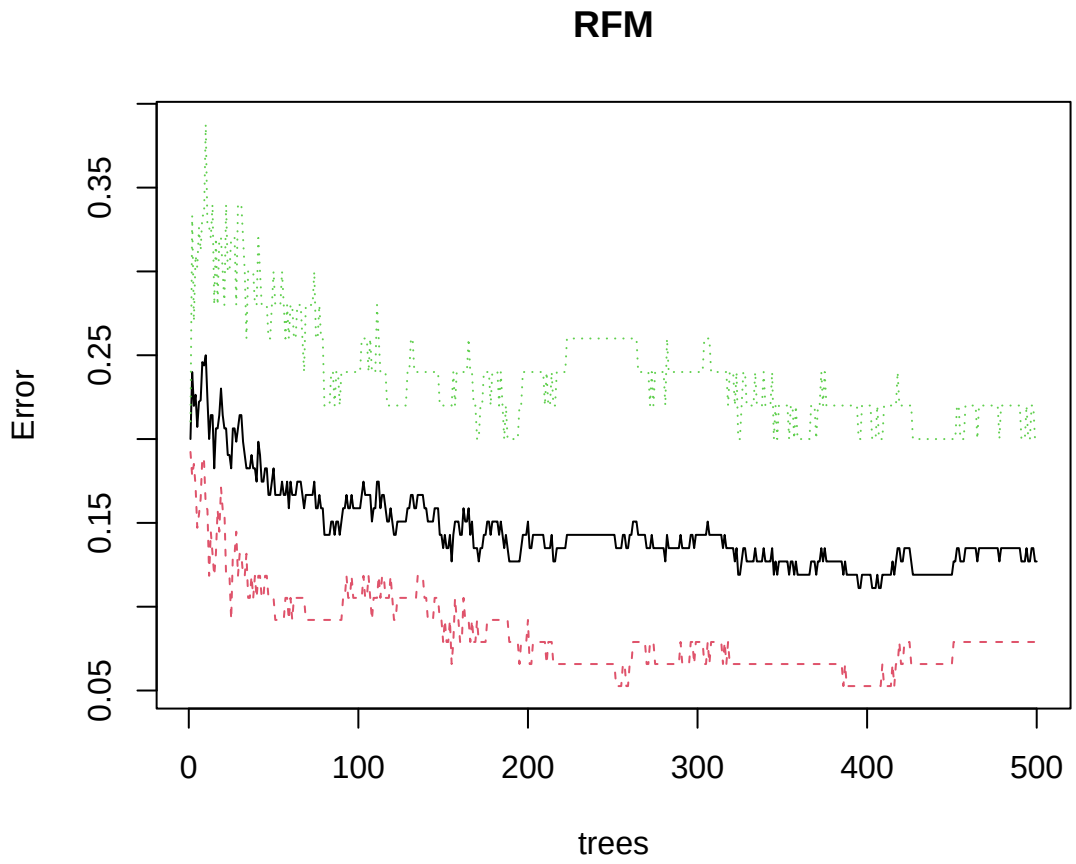
"Support vector machines (SVMs) offer a direct approach to binary classification: try to find a hyperplane in some feature space that "best" separates the two classes. In practice, however, it is difficult (if not impossible) to find a hyperplane to perfectly separate the classes using just the original features. SVMs overcome this by extending the idea of finding a separating hyperplane in two ways: (1) loosen what we mean by "perfectly separates", and (2) use the so-called kernel trick to enlarge the feature space to the point that perfect separation of classes is (more) likely."

4. Results and Discussion

4.1. Random Forests

##

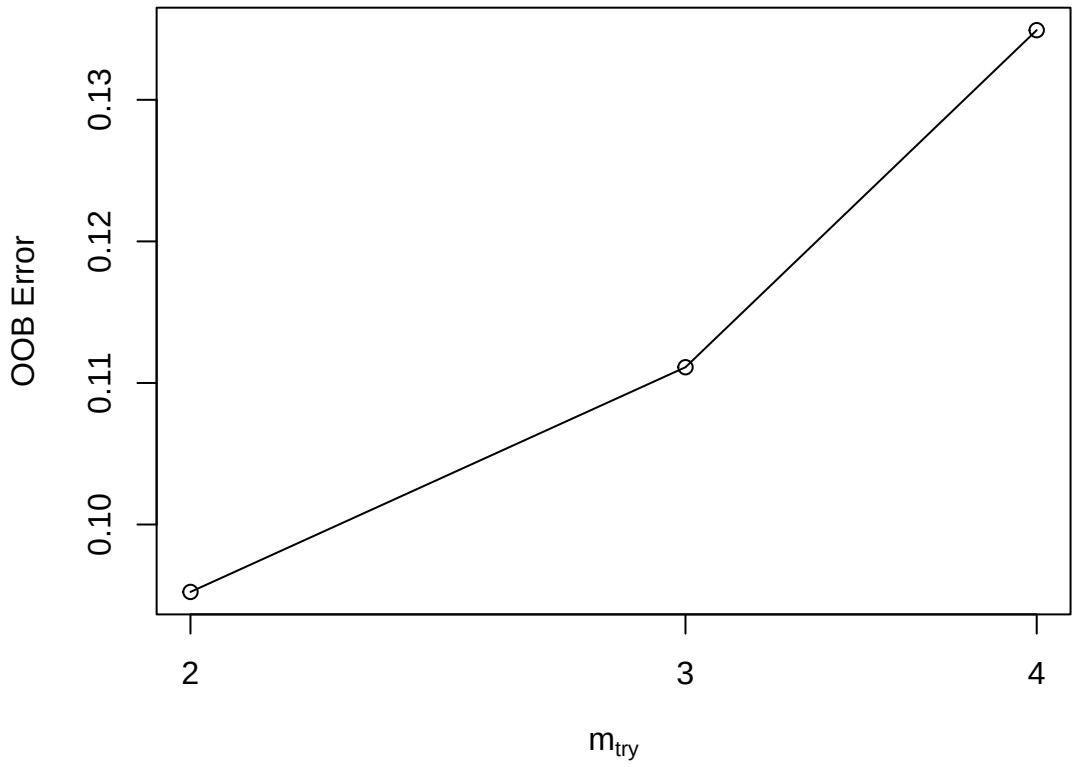
```
## Call:
##  randomForest(formula = heart_disease_present ~ ., data = train_1,      ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 12.7%
## Confusion matrix:
##    0  1 class.error
## 0 70  6  0.07894737
## 1 10 40  0.20000000
```




```
## [1] 3
```

```
## mtry = 3  OOB error = 11.11%
## Searching left ...
## mtry = 2     OOB error = 9.52%
```

```
## 0.1428571 0.01
## Searching right ...
## mtry = 4     OOB error = 13.49%
## -0.4166667 0.01
```

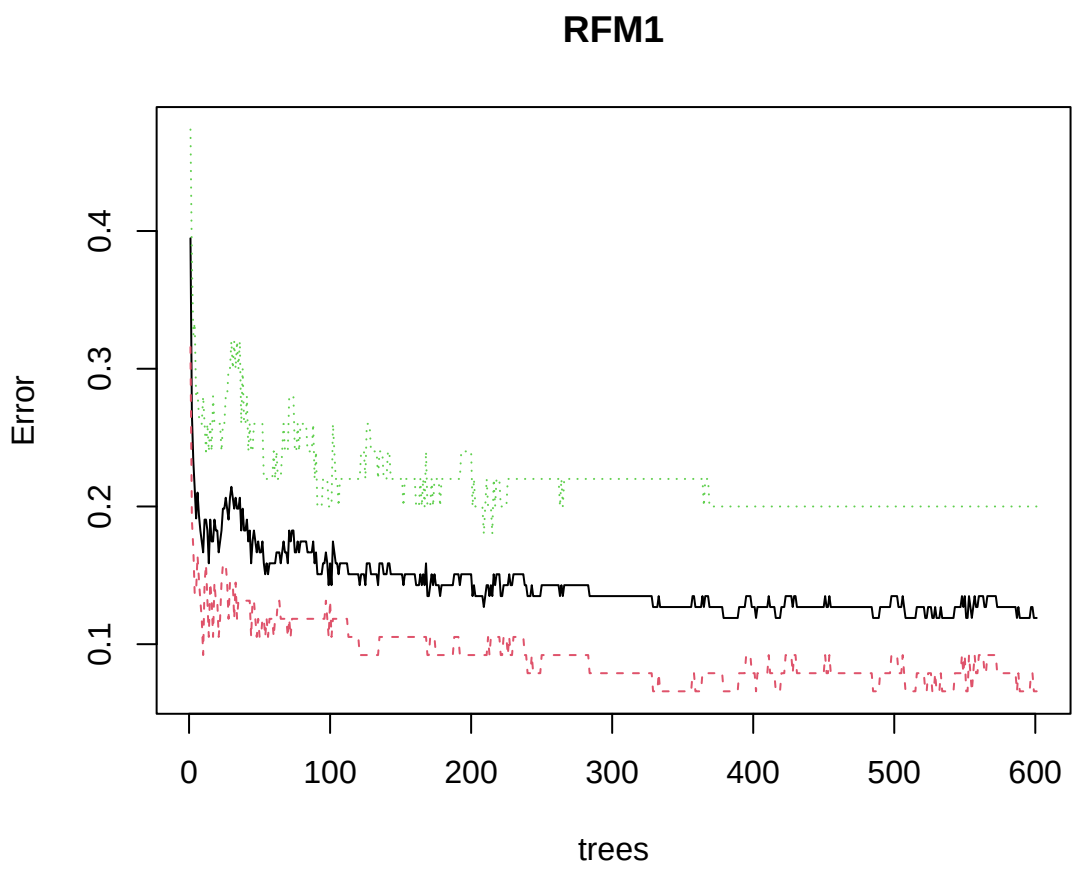


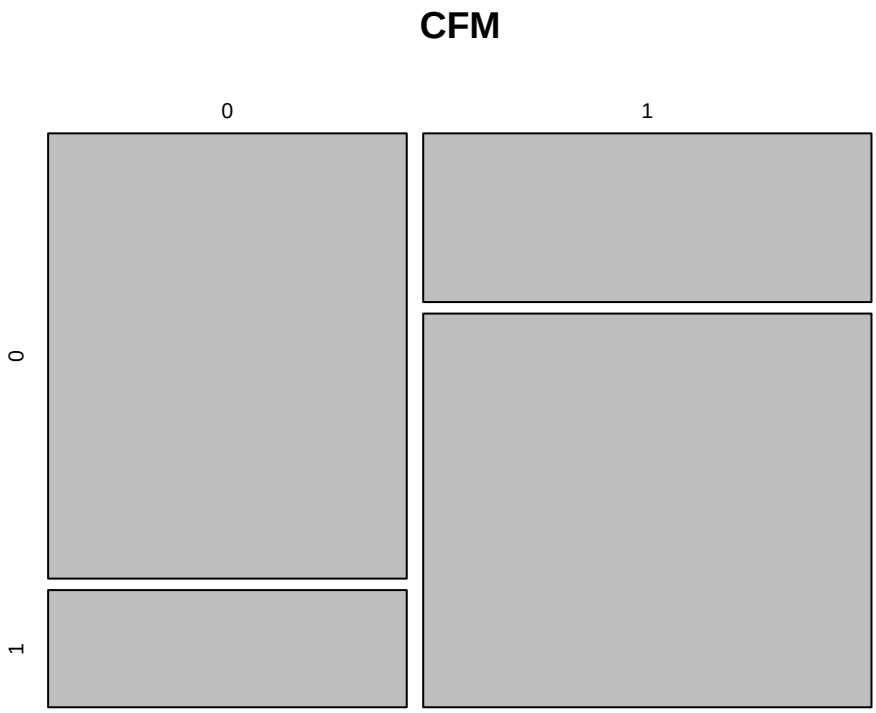

```
##       mtry  OOBError
## 2.OOB    2 0.0952381
## 3.OOB    3 0.1111111
## 4.OOB    4 0.1349206
```

```
## [1] 2
```

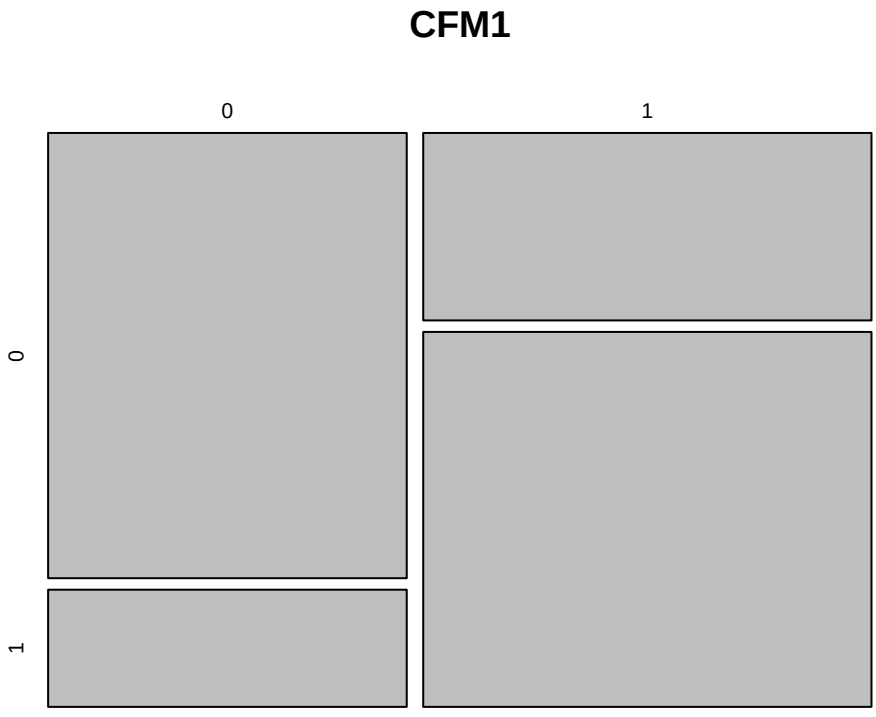
```
##
## Call:
##  randomForest(formula = heart_disease_present ~ ., data = train_1,      mtry = best.m, ntree
##                Type of random forest: classification
##                      Number of trees: 601
## No. of variables tried at each split: 2
```

```
##
##         OOB estimate of  error rate: 11.9%
## Confusion matrix:
##    0  1 class.error
## 0 71  5  0.06578947
## 1 10 40  0.20000000
```

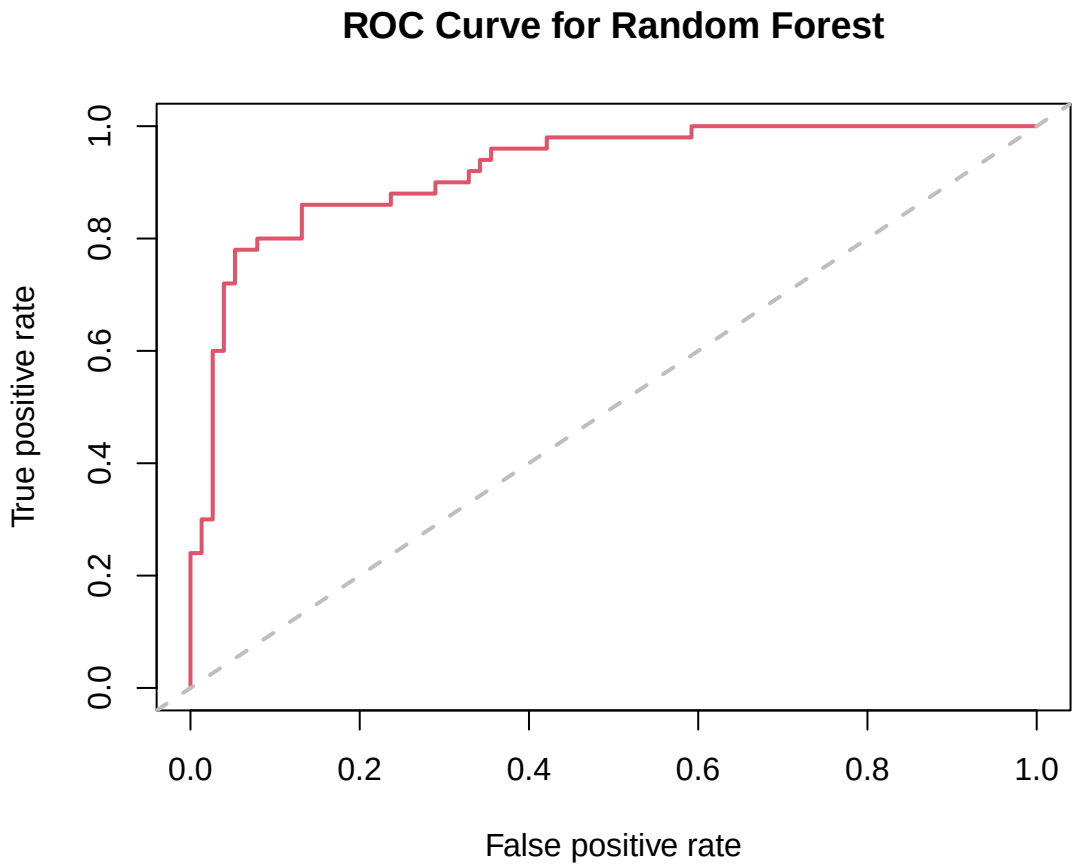




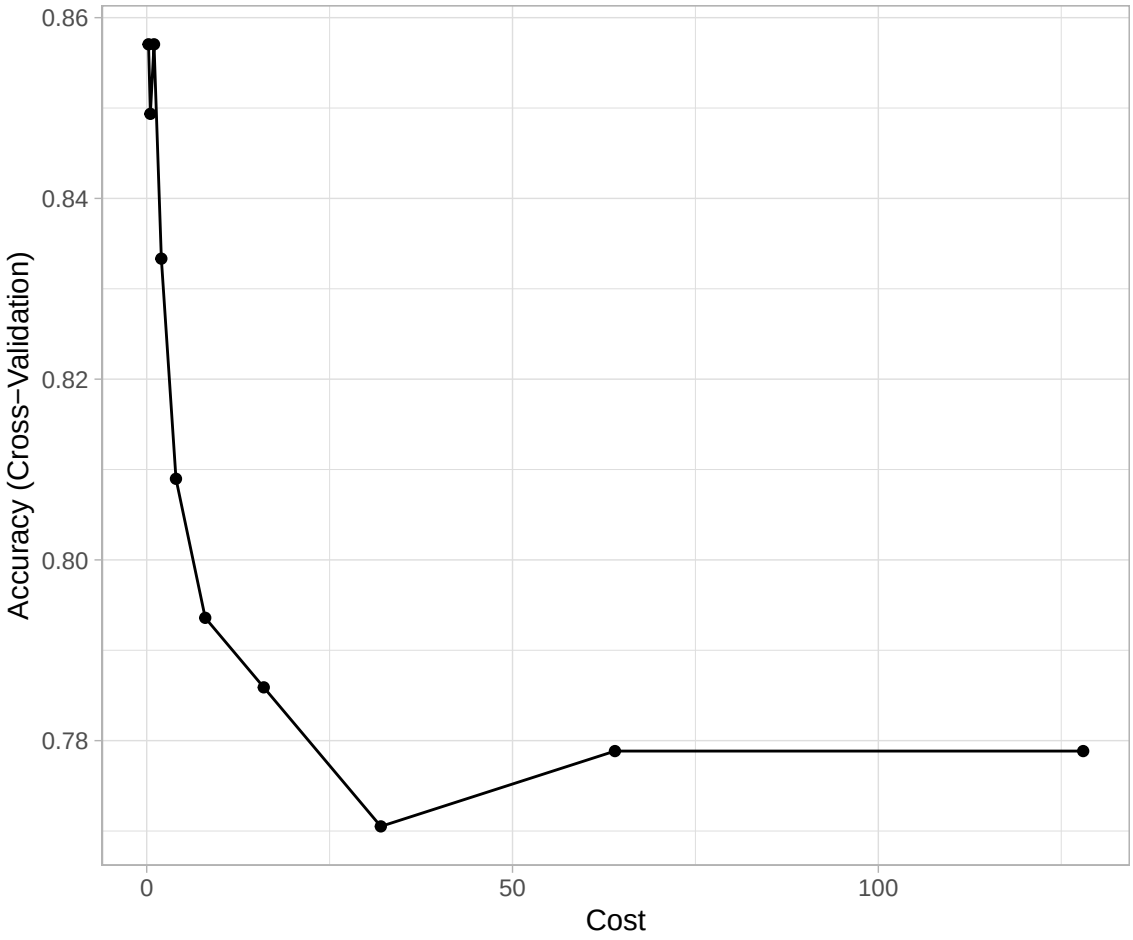

```
## [1] 0.7407407
```

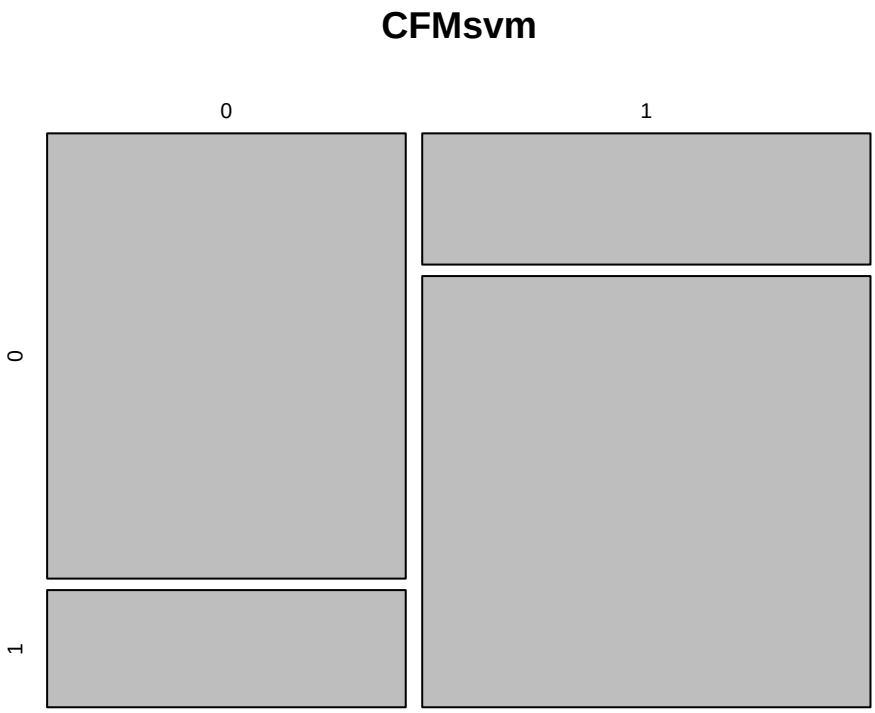


```
## [1] 0.7407407
```


ROC Curve for Random Forest
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

4.2. Support Vector Machine






```
## [1] 0.7777778
```

5. Conclusion

6. Reference List

7. Appendix

```
##
##         0         1
## 0.5555556 0.4444444
```