Predicting Drug Consumption Risk

Introduction

The aim of this project is to apply three types of classification algorithms - k-nearest neighbours, tree based methods and support vector machines - to a data set containing information about personality characteristics, including the Five Factor Model, and the tendency for drug consumption. The purpose is to determine which algorithm is best at predicting future drug use.

The Five Factor model is the most comprehensive system for understanding human individual differences and is comprised of:
Neuroticism - Neuroticism is the tendency to experience negative emotions such as nervousness, tension, anxiety and depression
Extraversion - A person who scores high in Extraversion is characterised as outgoing, talkative, warm, active and assertive.
Openness to experience - Openness to experience involves six dimensions, and is manifested as a preference for variety, aesthetic sensitivity, attentiveness to inner feelings and intellectual curiosity.
Agreeableness - Agreeableness is a dimension of interpersonal relations, characterised by altruism, trust, modesty, kindness and compassion.
Conscientiousness - Conscientiousness is manifested in organised, dependable, persistent, reliable and efficient characteristics.

The data set contains 600 observations and the following variables:

1. Age: denotes the age of the participant;
2. Education: the education level of the participant;
3. Country of origin: the country of origin of the participant;
4. Ethnicity: the ethnicity of the participant;
5. Nscore: the participant's score on Neuroticism;
6. Escore: the participant's score on Extraversion;
7. Oscore: the participant's score on Openness to experience;
8. Ascore: the participant's score on Agreeableness;
9. Cscore: the participant's score on Conscientiousness;
10. Impulsivity: the participant's score on Impulsivity;
11. SS (Sensation Seeking): the participant's score on Sensation-Seeking;
12. Class: the variable for drug use, with two labels - 'Never used' (taking a value of 0) and 'Used at some point' (taking a value of 1)

The data has been randomly split into three subsets:
- train.data - containing 300 observations (or 50% of the data)
- valid.data - containing 150 observations (25% of the data)
- test.data - containing 150 observations (25% of the data)

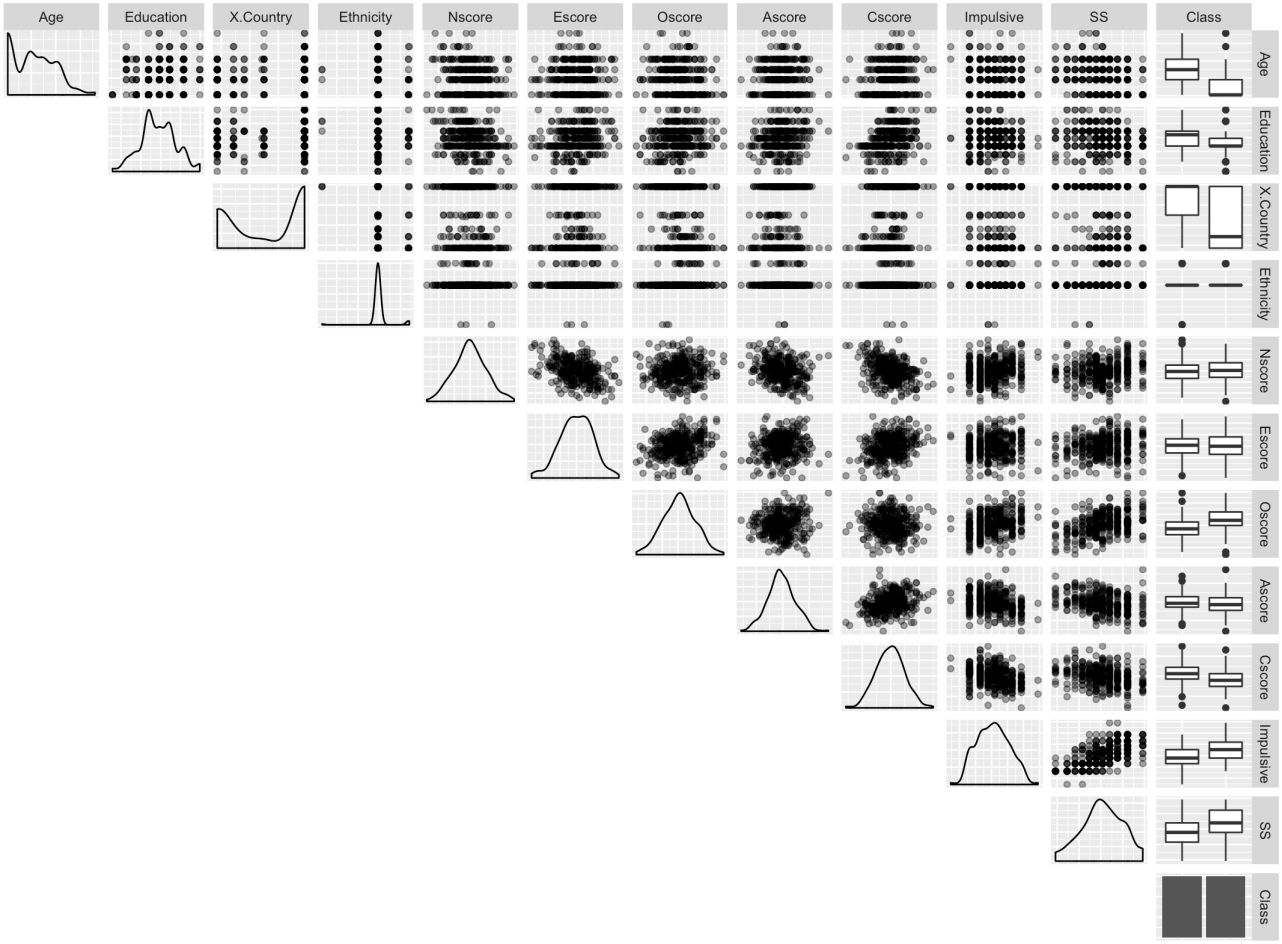
Exploratory Analysis

The exploratory analysis was performed on the train.data data set.

From the summary below, it has been noted that all the variables had been normalised.
Also, Class was coded as an integer, and it has been re-coded as a factor with two levels.
The variable X has been removed from further analysis.

```
##        X                Age              Education          X.Country
##  Min.   :  1.0   Min.   :-0.95197   Min.   :-2.43591   Min.   :-0.5701
##  1st Qu.:158.8   1st Qu.:-0.95197   1st Qu.:-0.61113   1st Qu.:-0.5701
##  Median :302.0   Median :-0.07854   Median :-0.05921   Median : 0.9608
##  Mean   :307.4   Mean   : 0.02373   Mean   :-0.11594   Mean   : 0.3262
##  3rd Qu.:460.2   3rd Qu.: 0.49788   3rd Qu.: 0.45468   3rd Qu.: 0.9608
##  Max.   :600.0   Max.   : 2.59171   Max.   : 1.98437   Max.   : 0.9608
##    Ethnicity           Nscore             Escore             Oscore
##  Min.   :-1.1070   Min.   :-2.756960   Min.   :-2.728270   Min.   :-2.6320
##  1st Qu.:-0.3169   1st Qu.:-0.678250   1st Qu.:-0.695090   1st Qu.:-0.6168
##  Median :-0.3169   Median :-0.051880   Median : 0.003320   Median : 0.1414
##  Mean   :-0.3002   Mean   : 0.006515   Mean   :-0.004133   Mean   : 0.0803
##  3rd Qu.:-0.3169   3rd Qu.: 0.629670   3rd Qu.: 0.637790   3rd Qu.: 0.7233
##  Max.   : 0.1260   Max.   : 2.821960   Max.   : 2.573090   Max.   : 2.9016
##      Ascore             Cscore            Impulsive             SS
##  Min.   :-3.1574   Min.   :-3.15735   Min.   :-2.5552   Min.   :-2.07848
##  1st Qu.:-0.7610   1st Qu.:-0.68478   1st Qu.:-0.7113   1st Qu.:-0.52593
##  Median :-0.1549   Median :-0.00665   Median : 0.1927   Median : 0.07987
##  Mean   :-0.1025   Mean   :-0.03055   Mean   : 0.1007   Mean   : 0.07888
##  3rd Qu.: 0.4385   3rd Qu.: 0.58489   3rd Qu.: 0.8811   3rd Qu.: 0.76540
##  Max.   : 3.4644   Max.   : 3.00537   Max.   : 2.9016   Max.   : 1.92173
##      Class
##  Min.   :0.0
##  1st Qu.:0.0
##  Median :0.5
##  Mean   :0.5
##  3rd Qu.:1.0
##  Max.   :1.0
```

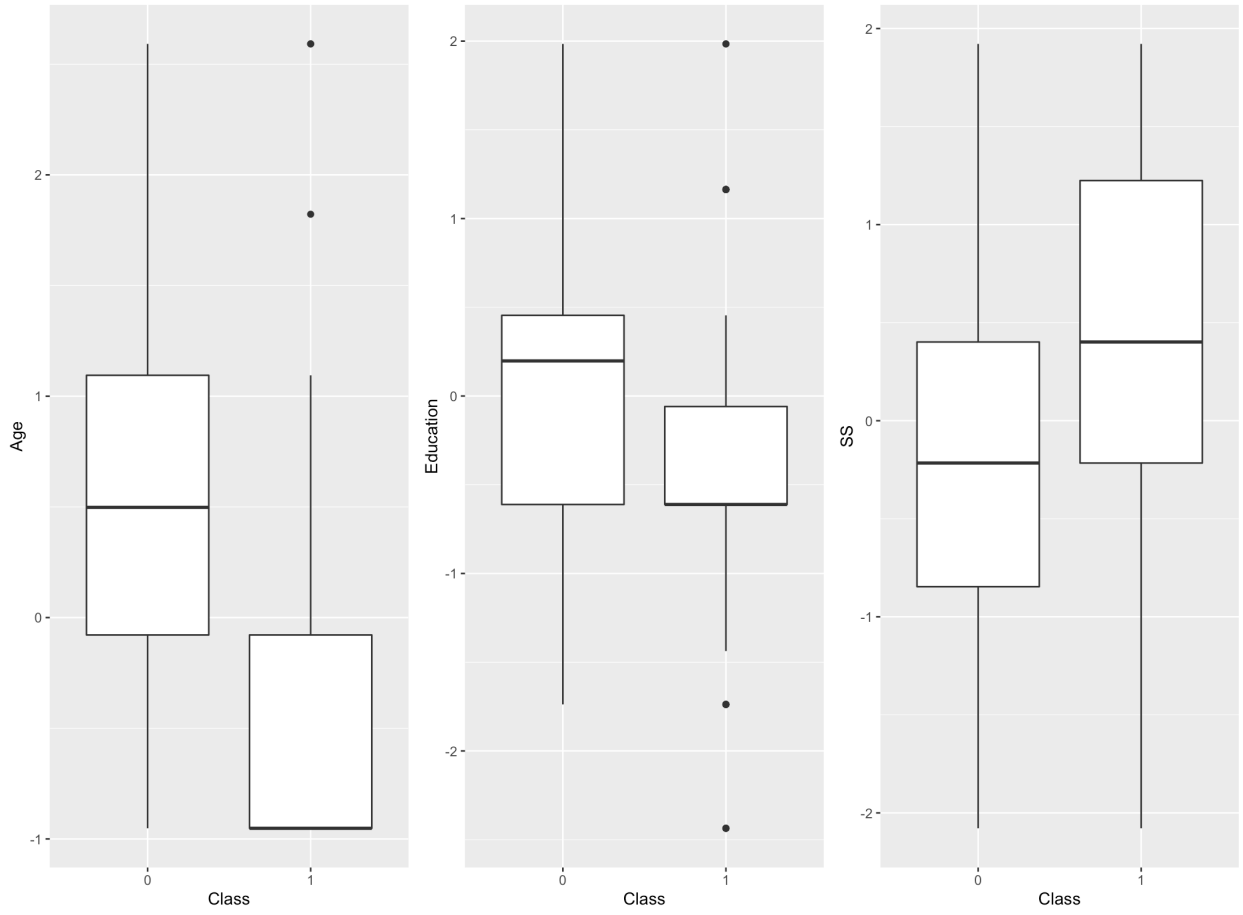
Furthermore, from the plots below, Agreeableness (Ascore) appears negatively associated with Neuroticism (Nscore). Conscientiousness seems to have a positive relationship with Education and is negatively associated with Neuroticism. A higher scores on the Impulsivity variable seem to be positively associated with drug use.
It is difficult to tell from the plots but there might be a couple of outliers present.



Below, a few pair of variables have been chosen and looked at individually.
We can see from the first plot that the median line for the age of people who have consumed drugs lies below the median line for the age of the people who haven't consumed drugs.

The second plot is indicating that the level of education of those who have never consumed drugs is higher than that of those who have consumed drugs.

In the third plot, the median line for the sensation seeking score of people who have never consumed drugs lies below the median line of those who have scored lower on this variable, indicating that higher scores on sensation seeking are associated with drug use.

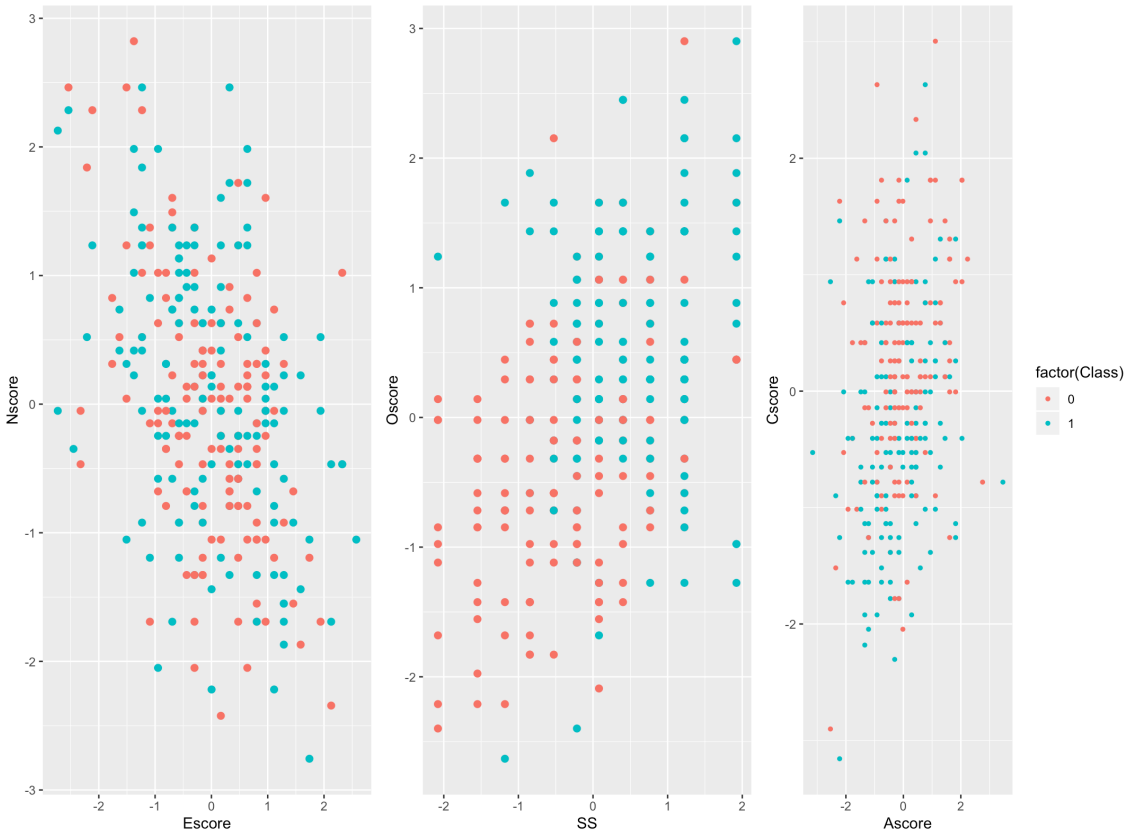


Another set of exploratory plots has been placed below.

From the first plot, we can see that there is a negative relationship between Neuroticism (Nscore) and Extraversion (Escore). Also, there is a lot of overlap between drug users / non-users.

From the second plot we can see that there is a positive association between Openness (Oscore) and sensation seeking (SS). Also, the users vs non-users are fairly divided on the plot, indicating that the probability of drug use increases with higher values scored on the Oscore and SS variables.

From the third plot we can see that lower levels of Agreeableness (Ascore) are associated with drug use, as well as lower levels of Conscientiousness. There is moderate amount of overlap on the plot.



Results

In this section, we will propose three different classification algorithms that can be used for identification of future drug use. The algorithms discussed are the following: k-nearest neighbours, support vector machines and tree based methods.

K-nearest neighbours has been fitted for values of k ranging from 1 to 50 and the performance was evaluated on the validation data in order to choose the best performing version of the model.

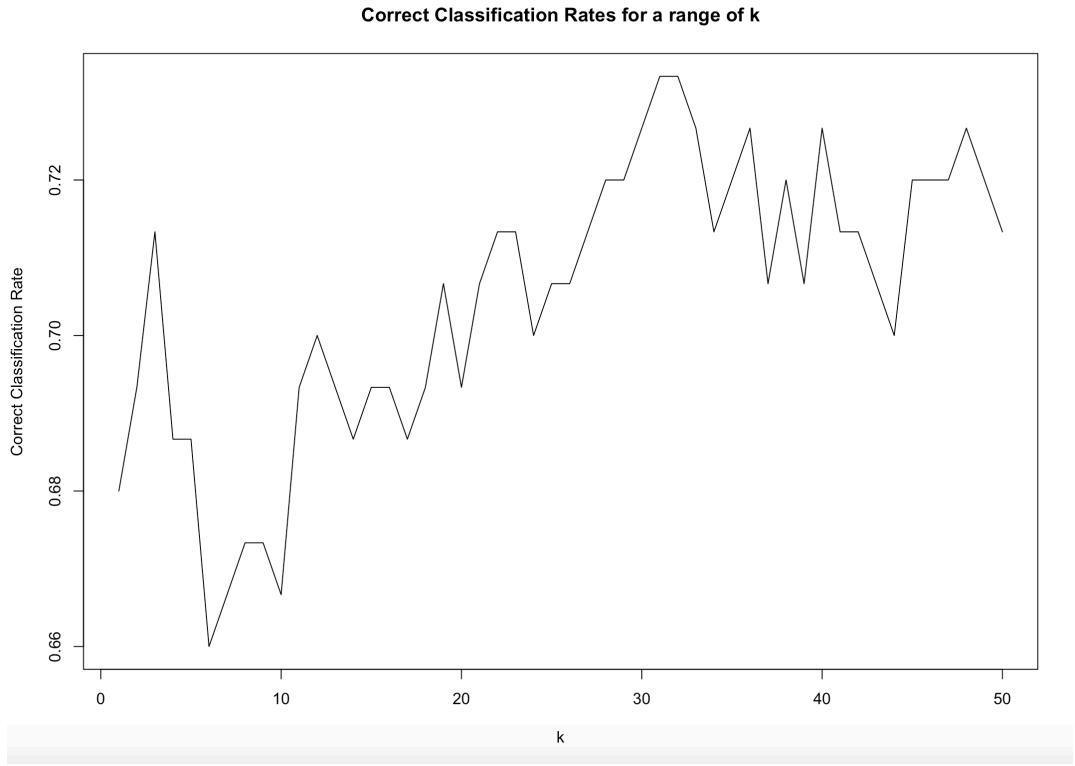
In the case of support vector machines, three different types of kernels were experimented with - linear, radial and second degree polynomial - and the best performing model and parameter were selected.

For classification trees, two different models were used for comparison - a pruned tree and a bootstrapped model - and the model with the lowest error was chosen.

Lastly, the error rates of the best models from each classification will be used for comparison the Discussion section.

K-Nearest Neighbours

The first algorithm proposed was k-nearest neighbours. The model was trained on the train.data dataset.
Afterwards, the valid.data dataset was used to select which value of k was optimal for our classification problem. The following plot was generated, showing the performance at each value of k from 1 to 50 on validation data set:



We can see that the best performance is given by k=31. This is the value that was then used to compute the error rate of the method on the validation data set.

The error rate for the k-nearest neighbours model is 0.2666667.

Support Vector Machines

The second type of algorithm proposed was the support vector machine. Once again, the models were trained on the train.data dataset.
Three types of kernels were looked at: linear, radial and second degree polynomial. The parameters *cost*, *gamma* and *coef* were chosen through cross-validation on the training set by using the *tune* function.
For the optimisation of the linear kernel SVM, a range of values from 0.001 to 10 were tested for the cost parameter. The *tune* function was applied until the cost value was no longer found at boundary. The cost value was chosen to be 0.01, as seen in the summary below. There are 228 support vectors on the training data set for this model.

```
##
## Call:
## best.tune(method = svm, train.x = Class ~ ., data = train.data, ranges = list(cost = c(0.001,
##     0.005, 0.01, 0.05)), type = "C-classification", kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.01
##
## Number of Support Vectors:  201
```

The data set valid.data was then used for validation. The trained model was used to generate predictions for this dataset, for which the error rate was computed for comparison with the radial and polynomial models.

The error rate for the linear SVM is 0.2933333.

For the optimisation of the radial SVM, a range of values from 0.01 to 10 were tested for the cost parameter. The gamma parameter was also optimised and chosen from a range of values from 0.001 to 4.
The cost value was chosen to be 0.1 and the gamma value was selected to be 0.001.

The summary of the model can be viewed below.

```
##
## Call:
## best.tune(method = svm, train.x = Class ~ ., data = train.data, ranges = list(cost = c(0.01,
##     0.05, 0.1, 0.5), gamma = c(1e-04, 5e-04, 0.001, 0.005, 0.01)),
##     type = "C-classification", kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
## Number of Support Vectors:  159
```

We can see there are 159 support vector for the radial SVM.
As with the linear model, the data set valid.data was used in order to select the best performing parameters for the model.

The error rate for the radial SVM is 0.3066667.

For the SVM with a second degree polynomial kernel, the parameters that needed optimisation were the cost, gamma and coef. The cost value options ranged from 0.01 to 5, the gamma values ranged from 0.05 to 0.1 and coef values were chosen out of a range from 0 to 3.
The cost parameter was chosen to be 0.1, the gamma value was chosen to be 0.01 and coef parameter was selected to be 2.

The summary of the polynomial SVM can be viewed below:

```
Call:
best.tune(method = svm, train.x = Class ~ ., data = train.data, ranges = list(cost = c(0.01,
    0.05, 0.1), gamma = c(0.005, 0.01, 0.1), coef0 = c(0, 1, 2, 3)), type = "C-classification",
    kernel = "polynomial", degree = 2)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  0.1
     degree:  2
     coef.0:  2

Number of Support Vectors:  239
```

We can see that there are the 239 support vectors for the second degree polynomial model.

Following the same steps as with the linear and radial SVMs, we validated the model on the valid.data dataset.
The error rate for the polynomial SVM is 0.2866667.

In order to compare the performance of the three models on the validation data set, we can look at the table below, which contains the error rates computed for each SVM:

```
      Linear Radial Polynomial.2
[1,] 0.2933 0.3067       0.2867
```

We can see the second degree polynomial performs best, as it has returned the lowest error rate, therefore this SVM will be compared to the models resulted from the other classification methods.

Classification Trees

The third algorithm applied to the data was a classification tree.

A pruned tree and a bagged model were compared against each other using the valid.data data set.

First, a fully grown tree was produced. Looking at the importance of the variables used, the ones that weighed the most in building the tree were Oscore, Nscore, SS, Age and Education. Unsurprisingly, the least important variables were Ethnicity and Country.

```
   Oscore    Nscore        SS       Age Education    Escore    Ascore    Cscore Impulsive X.Country
37.880791 37.059602 34.985445 33.099583 32.751764 28.533285 27.343553 24.013107 21.801116 16.602899
Ethnicity
 5.573198
```

In order the prune the fully grown tree, the cross validation results were produced in order to identify the correct value of the complexity parameter *cp.* For this purpose, I chose the largest value within 1 standard deviation of the minimum. The lowest cross validation error rate (xerror) was 0.52667 with a corresponding standard deviation of 0.050858. In order to prune the tree, we chose the largest subtree with the cross validation error rate lower than 0.577528. The complexity parameter was therefore chosen to be 0.0044444.

The pruned tree was then applied to the validation data set. The resulting error rate for this method is 0.3066667.

For more consistent estimates, a bootstrapped model was also applied. This model was averaged across 250 trees. We can see the out of bag estimate of error rate for the bagged model is 22%.

```
##
## Call:
##  randomForest(formula = Class ~ ., data = train.data, method = "class",      mtry = 11, ntree = 250)
##                Type of random forest: classification
##                      Number of trees: 250
## No. of variables tried at each split: 11
##
##         OOB estimate of  error rate: 22%
## Confusion matrix:
##     0   1 class.error
## 0 120  30        0.20
## 1  36 114        0.24
```

However, in order to compare this model's performance to the tree's performance, it was applied to the valid.data data set and the error rate was computed. The error rate for the bootstrapped model is 0.3, which is marginally lower than the error rate for the pruned tree. Therefore, the bagged model will be compared to the K nearest neighbours and SVM models.

Discussion

In the section, the results outlined above will be compared in order to select the best performing classification method, which will then be evaluated for future sample predictions.

The table below presents the the error rates for the best performing model on the validation data set for each classification method used.

```
         KNN  SVM Tree(Bag)
[1,] 0.2667 0.28       0.3
```

As we can see in the table, the tree method is the worst performing one with an error rate of 0.3.

The best performing model is k-nearest neighbours, which has the lowest error rate at 0.2667. Therefore, this model will be applied to the test data in order to obtain an estimation of future performance.

The correct classification rate of the model on the test data is 0.76. This is the expected future performance of the k-nearest neighbours model at k=31.

As we can see from the tables below, the algorithm does a fairly good job at predicting both classes. The performance is slightly better on Class 0 ('Never used') - it correctly predicted 60 observations out of 78. In other words, the Specificity of the model is almost 0.77.
The method performs slightly worse at making predictions for Class 1('Used at some point') for which it correctly classified 54 observations out of 72. The Sensitivity of the model is therefore 0.75.
The misclassification rate for Class 0 is 0.23 and for Class 1 it is 0.25. The false discovery rate is 0.25.

```
pred.test.knn
   0  1
0 60 18
1 18 54
```

```
pred.test.knn
          0         1
0 0.7692308 0.2307692
1 0.2500000 0.7500000
```

Summary and Conclusion

The aim of this report was to propose three types of classification algorithms and choose the one with the best performance at determining wether future patients are likely or not to have consumed drugs, based on their scores on certain personality traits and other variables such as age, and ethnicity.

For this purpose, we looked at three different classification methods: k-nearest neighbours, support vector machine and classification trees. From each classification method, the best performing model was selected and their performance was compared against each other. For comparison, the error rate of each model has been used.

The model with the lowest error rate was the k-nearest neighbours model, with k=31, therefore this was selected as the best performing model. The k-nearest neighbours model was then applied on the test data in order to estimate future performance. The correct classification rate for future observations is expected to be 0.76, with a Sensitivity of 0.75 and Specificity of 0.77. The results are therefore satisfactory, especially considering the nature of our classification problem.