How can drug comsuption be predicted?

Data Mining and Machine Learning Group 29

Marta Casero, Jiawei Liu, Han Xu, Jinshen Zhang, Tianjian Yang

2023-03-22

Contents

Introduction

What affects drug consumption? Given 12 attributes from over 1800 participants and using data mining techniques this is the question that will be investigated. The variables that have been recorded are divided into two categories; personality measurements, which include NEO-FFI-R (neuroticism, extraversion, openness to experience, agreeableness, and conscientiousness), BIS-11 (impulsivity), and ImpSS (sensation seeking), and demographic variables, which include level of education, age, gender, country of residence and ethnicity. All the demographic variables were recorded as categorical and have been quantified for the analysis.

The investigation will focus on the use of one drug, crack, divided into three categories; Never used, Used over a year ago and Used in the last year. *Table 1* shows how the data would be distributed for both a seven-class classification and the three-class classification used, there is very few observations on some of them so a three class classification is going to be used..

This report has been divided into parts, summaries, which include plots of the distributions of the variables, formal analysis, which include various classification techniques, results, and conclusion.

Table 1: Distribution of the response variable in a seven and three class

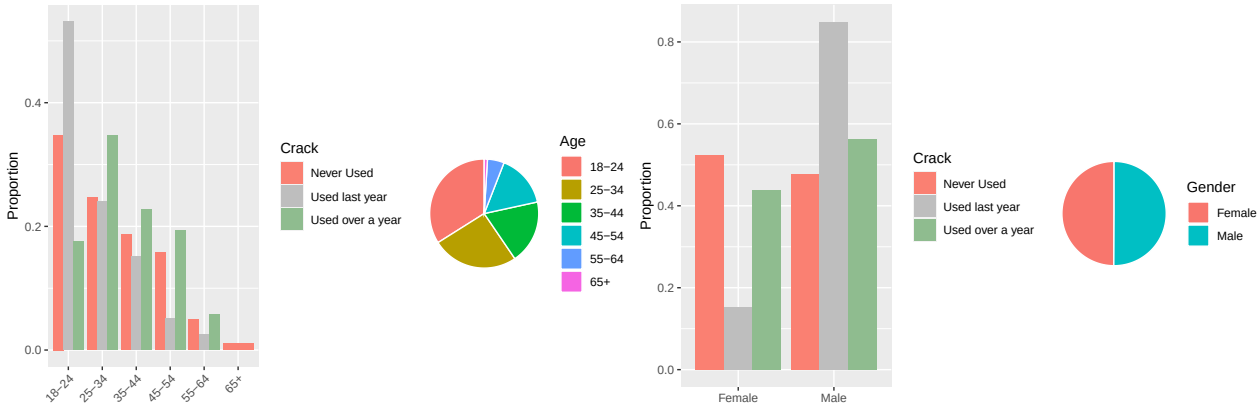
Seven-class classification		Three-class classification	
Never Used	1622	Never Used 1	1622
over a Decade	67	Used last year	79
Last Decade	109	Used over a year	176
Last Year	59		
Last Month	9		
Last Week	9		
Last Day	2		

Summaries

Figure 1 shows the distribution per drug level of all the explanatory variables. Plots *a)* to *e)* are for the categorical variables and all show barplots of the relationship between them and use of crack and a piechart of the distributions of the categories in the variable.

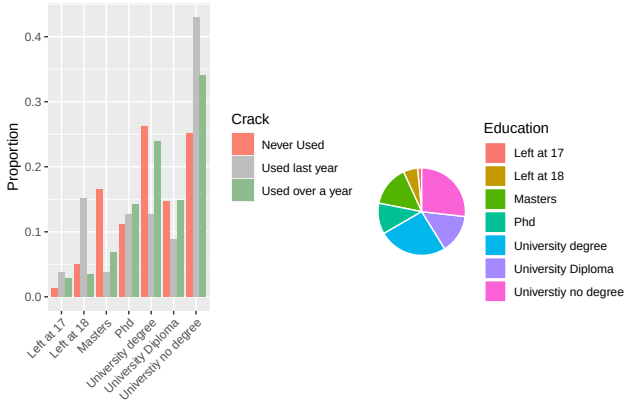
In *a)* is shown that the distributions per age are pretty similar, other that the 18-24 who have a higher proportion of used in the last year, but it also the age group with the most amount of participants. In *b)* is shown that there is a higher proportion of males who have used the drug, mainly in the last year. In plot *c)* there is a lot of changes in the different education levels, this suggests that this variable might be useful in the formal analysis. From plot *c)* is easy to see that the UK has the highest proportion of never used and that USA has the highest proportions for both the use levels, this two countries are also around 80% of the population investigated. Plot *e* shows that there is no clear relation between crack use and ethnicity.

Plot *f)* is a boxplot of the personality measurements, the data has a lot of outliers but this can be expected as there is a lot of participants in the study. Some of this measurements look to have a significant relationship with the use of crack. This suggests that even thought the dataset has a low proportion of participants that have used crack before; the formal analysis might give good results.

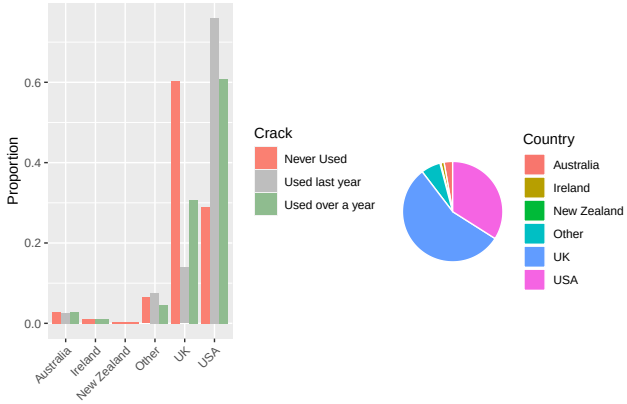


(a) Relationship between crack use and age

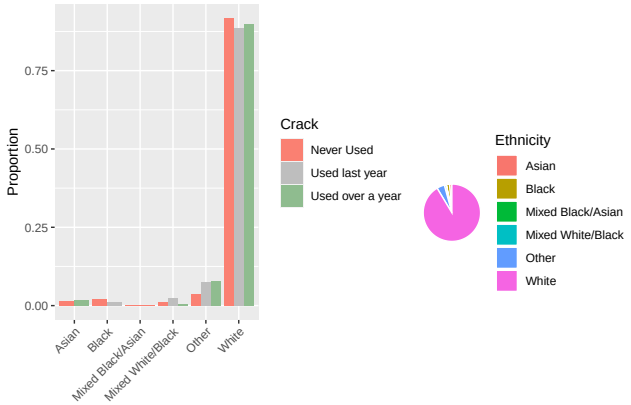
(b) Relationship between crack use and gender



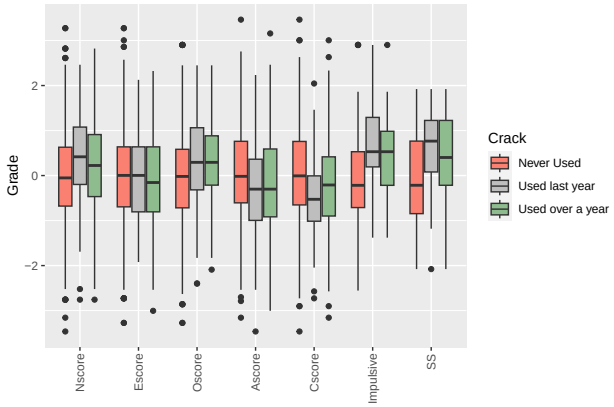
(c) Relationship between crack use and education



(d) Relationship between crack use and country



(e) Relationship between crack use and ethnicity



(f) Boxplots of personality measurements per use of drug

Figure 1: Summaries

Figure 2 shows the correlation per response level for all the explanatory variables. As a lot of this variables are quantified categorical variables, it is expected to not have a high correlation. However, the personality measurements give an interesting results in the plot, and it will be useful for the formal analysis and will be referred to in the next part.

Formal Analysis

K-nearest neighbours

The k-Nearest Neighbors (kNN) approach is a simple yet effective algorithm used for classification and regression tasks. The algorithm works by finding the k observations in the training set that are closest to a new input instance, based on a set distance metric such as Euclidean. The majority class or the mean value of the k nearest neighbors is then used as the prediction for the new instance. The value of k is a hyperparameter that can be tuned to achieve better performance. The kNN algorithm is non-parametric, meaning it doesn't make any assumptions about the underlying data distribution.

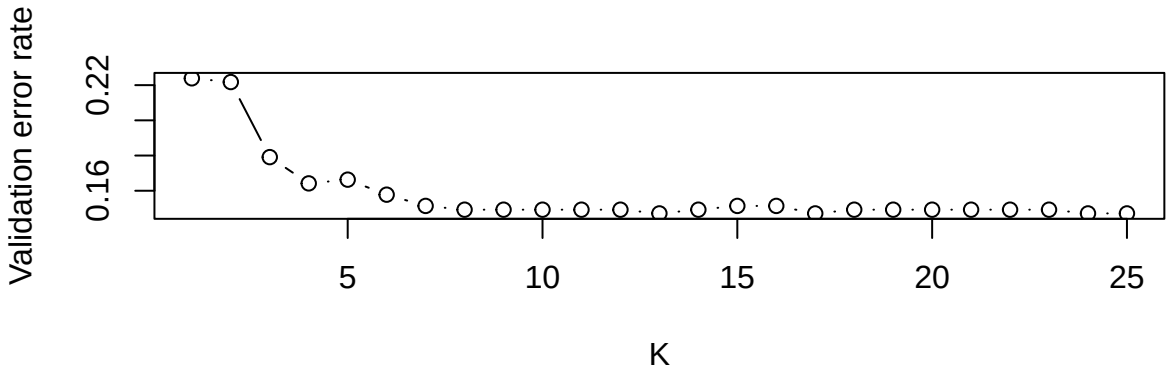


Figure 2: KNN valid error

Given both the minimum valid error and the plot in *Figure 3*, the optimal value for kNN prediction is 14.

Table 2: Prediction table form the knn method

	Never Used	Used last year	Used over a year
Never Used	410	0	3
Used last year	18	0	0
Used over a year	39	0	0

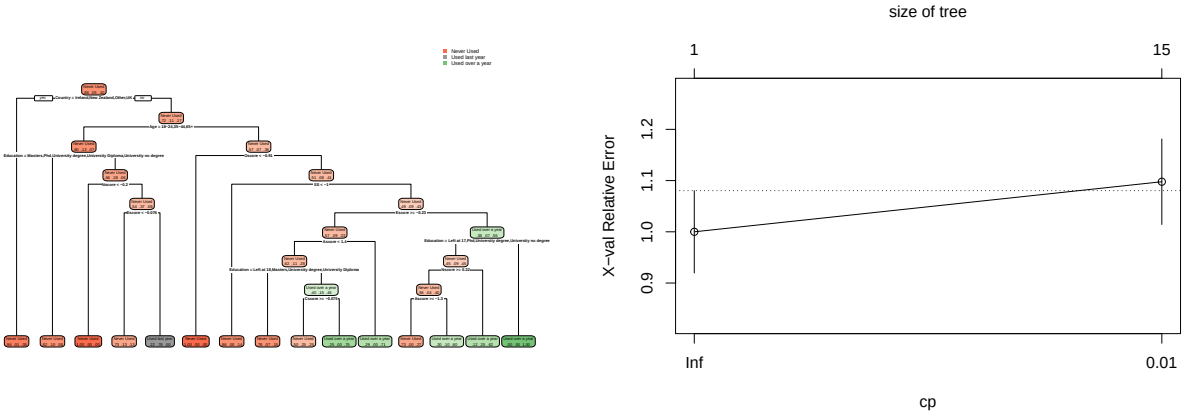
Table 2 shows that knn predicts most valuables into Never used, this probably comes from the fact that there was not many observations in the data set in the other variables and thet number gets even smaller when the dataset gets split into training and test set.

Tree method

Classification tree method is a popular algorithm used in data mining and machine learning to create a predictive model for classification tasks. It works by recursively partitioning the data into subsets based on the values of the input features, and then assigning a class label to each subset based on the majority class of the training examples in that subset.

The algorithm starts with the entire dataset as the root node, and then splits the data into two or more subsets based on the feature that provides the greatest information gain, or the most effective way to separate the classes. This process continues until a stopping criterion is met, such as a maximum tree depth or a minimum number of training examples per leaf node.

The resulting classification tree can be used to predict the class label of new examples based on their input features. This classification method is known for its simplicity and visualization, as the resulting tree can be easily interpreted and understood by the general public.



(a) Classification tree

(b) cp

Figure 3: Classification tree and cp

Figure 4 shows the classification tree and it cp, it is easy to se the larger sized tree are better to fit this data.

Table 3: Prediction table form the tree

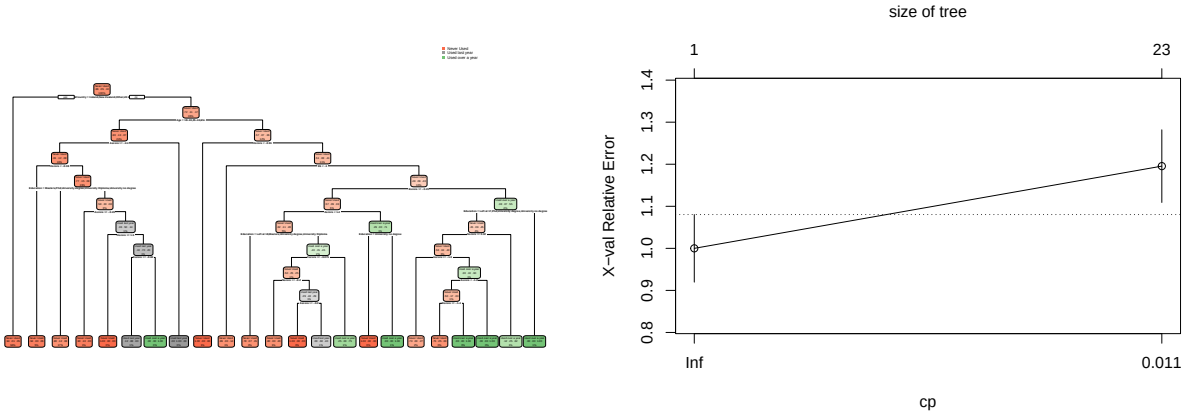
	Never Used	Used last year	Used over a year
Never Used	407	16	30
Used last year	3	0	0
Used over a year	8	1	5

Given the results on *Table 3* the accuracy of this tree is 0.8765957. This trees does not use all the possible explanatory variables.

Table 4: Table from tree

	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision	Balanced Accuracy
Class: Never Used	0.9736842	0.1153846	0.8984547	0.3529412	0.8984547	0.5445344
Class: Used last year	0.0000000	0.9933775	0.0000000	0.9635974	0.0000000	0.4966887
Class: Used over a year	0.1428571	0.9793103	0.3571429	0.9342105	0.3571429	0.5610837

A full tree can also be created and pruned, the full tree will have all the explanatory variables. This can then be compared to the previous tree to decide which one is a better fit for this dataset.



(a) Full tree

(b) cp

Figure 4: Full pruned tree and cp

Table 5: Prediction table form the full pruned tree

	Never Used	Used last year	Used over a year
Never Used	405	16	29
Used last year	4	0	2
Used over a year	9	1	4

The accuracy given by the full pruned tree on *Figure 5* can be calculated using the data on *Table 5*; the accuracy is 0.8702128

Table 6: Table form the full pruned tree

	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision	Balanced Accuracy
Class: Never Used	0.9688995	0.1346154	0.9000000	0.3500000	0.9000000	0.5517575
Class: Used last year	0.0000000	0.9867550	0.0000000	0.9633621	0.0000000	0.4933775
Class: Used over a year	0.1142857	0.9770115	0.2857143	0.9320175	0.2857143	0.5456486

Table 4 and *Table 6* have very similar values, however the first tree has a sighlty better prediction. It also has less nodes which makes it faster to run and easier to understand. Hence, the tree on *Figure 4* is a better classification method for the given data.

Linear and quadratic discriminant analysis

Linear Discriminant Analysis (LDA) is a statistical technique that seeks to find a linear combination of features or variables that best separates two or more classes or groups of data. The goal of LDA is to project the original data into a lower-dimensional space while preserving the class-discriminatory information.

In other words, LDA seeks to find a linear boundary or decision surface that best separates the classes, and accomplishes this by maximizing the between-class variance and minimizing the within-class variance. This also means that the use of LDA requires the variables to have a certain distribution and variance. To be more exact, we'll assume the group of variables follow a multivariate Gaussian distribution, and have a equal covariance matrix.

Quadratic Discriminant Analysis (QDA) is a statistical technique that is similar to LDA. However, unlike LDA, QDA allows for non-linear decision boundaries. And the variance and covariance of the predictor variables are allowed to differ between classes, resulting in a separate covariance matrix for each class.

To apply LDA (and QDA),there is an assumption that the observations in dataset follow a multivariate Gaussian distribution, which is more restrictive.

Two different cases are going to be considered, the equal covariance matrix case, and the case in which the class-covariance matrices are not assumed to be equal. The former case will apply LDA, and for later QDA will be applied.

From the chart on *Figure 3* it is easy to see that the variables Nscore to Cscore nearly follow the Gaussian distribution and Impulsive and SS roughly follow it. However, other variables are numerical variables transformed from categorical variables, and obviously do not follow Gaussian distribution. So two group of analysis will be made to see whether LDA method works well on this set of data.LDA1 will consider all the variables given and LDA2 will only consider the numerical variables; personality measurements.

	Never Used	Used last year	Used over a year
Age	0.7916112	0.5516475	0.5871009
Gender	0.2324265	0.1214834	0.2304400
Education	0.8861081	0.7309267	0.9513266
Country	0.4655506	0.3061925	0.4727129
Ethnicity	0.02923015	0.02214670	0.01466972
Nscore	0.9815622	1.0404769	1.0021583
Escore	0.9967044	0.8147390	1.0322663
Oscore	1.0027928	0.9626822	0.7488011
Ascore	0.9591215	1.0782256	1.1928083
Cscore	0.9761778	0.8317643	1.0683572
Impulsive	0.8943052	0.5818817	0.8601190
SS	0.9191740	0.5898128	0.7698302

Table 7: Variance of the explanatory variables

It seems that only Nscore and Ascore have similar variance. Although the variance and covariance show that the data might not fit well in an LDA, anyway, the analysis has been performed.

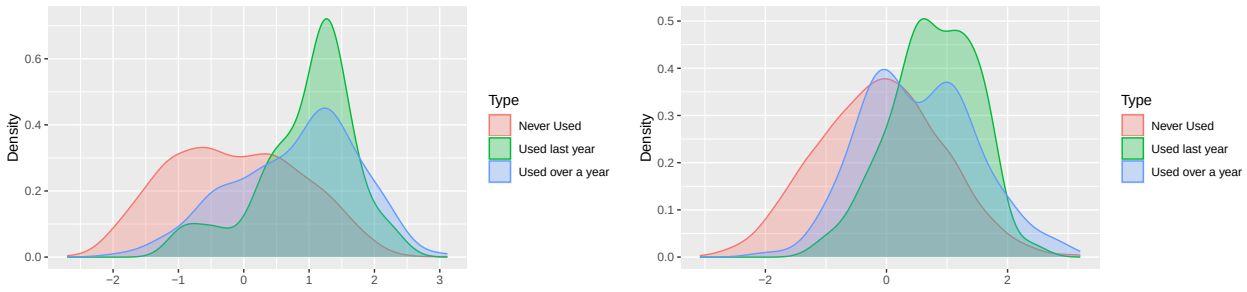
Table 8: Prediction table from LDA1

	Never Used	Used last year	Used over a year
Never Used	1287	0	4
Used last year	61	0	0
Used over a year	147	0	2

The density plot for each of the drug use factors has been plotted on the same axis to see the overlap between them.

Table 9: Prediction table from LDA2

	Never Used	Used last year	Used over a year
Never Used	1290	0	1
Used last year	61	0	0
Used over a year	149	0	0



(a) LDA1

(b) LDA2

Figure 5: Linear discriminat models

For both of the analysis performed it is shown on the plots in *Figure 5* that the LDA does not work properly as they overlap, this implies that LDA does not manage to difference between the 3 factors established.

Table 10: Prediction table from QDA1

	Never Used	Used last year	Used over a year
Never Used	1257	8	26
Used last year	55	4	2
Used over a year	121	1	27

Table 11: Prediction table from QDA2

	Never Used	Used last year	Used over a year
Never Used	1283	3	5
Used last year	58	2	1
Used over a year	143	1	5

The QDA analysis performs better than the LDA but it is still not suitable for this dataset..

As can be seen from the visualization and tables, because the distribution of variables does not meet the requirements of LDA ot QDA, the classification results are mostly shown as most observations are classified into the same class.

Support vector machines

Support vector machines (SVM) is a machine learning algorithm that can be used for classification and regression analysis. It tries to find the best hyperplane that can separate the different classes of data points with the maximum margin of separation. SVM can handle both linearly separable and non-linearly separable data by using different kernel functions(here we use radial kernel) to transform the data into a higher-dimensional space where it can be separated.

Parameter tuning of 'svm':

- sampling method: 10-fold cross validation
- best parameters: c=0.01 gamma:1
- best performance: 0.1424763

After performing the tune analysis, the parameters for a bes model fit are c=0.01 gamma:1. After implementing this paramaters in a SVM model and predictiong a model the responses in *Table 12* are obtained.

Table 12: Prediction table for OVO and OVA

	Prediction OVO	Prediction OVA	Actual
Never Used	564	564	496
Used last year	0	0	24
Used Over a year	0	0	44

The OVO (one-v-one) approach has a accuracy of 0.8953901, the same as OVA (one-v-all) approach. However this level of accuracy is not actually useful as both of the model predict all the observations into the 'Never Used' category.

In summary, based on the SVM model, the three-classification problem was trasformed into a two-classification problem by both one-to-one and one-to-all strategies, and applied the radial kernel, as well as adjusting the parameters to obtain the final two models. However, due to the poor quality of the data, the final prediction on the test set puts all observations to be in the classification of never used drugs. The model still maintains a high prediction accuracy of 86.30%. If the model continues to be used in the future, information on the drug-using population or more relevant variables should be added.

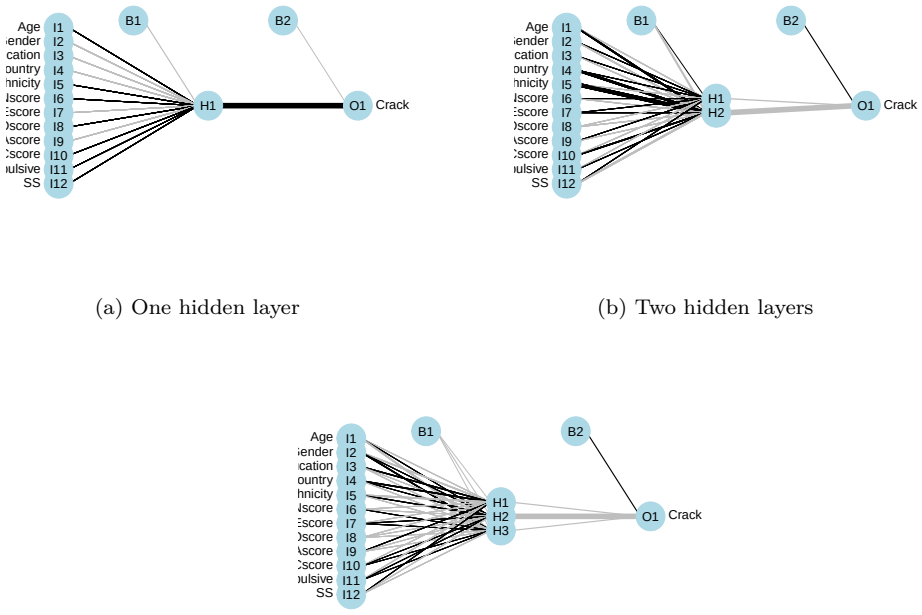
Neural network

A neural network is a computer algorithm with a structure and function similar to the biological nervous systems. It consists of a series of interconnected nodes (also known as neurons) that process input data and generate output. Neural networks are commonly used for machine learning tasks such as classification, regression, image recognition, natural language processing, etc.

A neural network can be viewed as a function combiner consisting of multiple layers, where each layer transforms input data into a new representation. The input layer receives raw data, which is then passed on to the hidden layers and finally to the output layer, which gives the neural network's predictions for the input data.

When training a neural network, the network has to provided with training data and a target output, and the network constantly adjusts its parameters to minimize the gap between its predictions and the target output. This self-tuning process is usually achieved using the gradient descent algorithm.

Neural networks play an important role in deep learning because deep learning models are usually complex models composed of multiple neural network layers.



(a) One hidden layer

(b) Two hidden layers

(c) Three hidden layers

Figure 6: Neural network plots

In *Figure 6*, the thickness of each connection is proportional to the weights' magnitude; the black colour refers to positive weights and gray refers to negative weights. The first layer includes only input variables with nodes labelled arbitrarily as I1 through I12 for 12 input variables. Hidden layers are plotted as H1, H2, H3. The output layer is plotted last with its node labeled as O1. Bias nodes connected to the hidden and output layers are also shown.

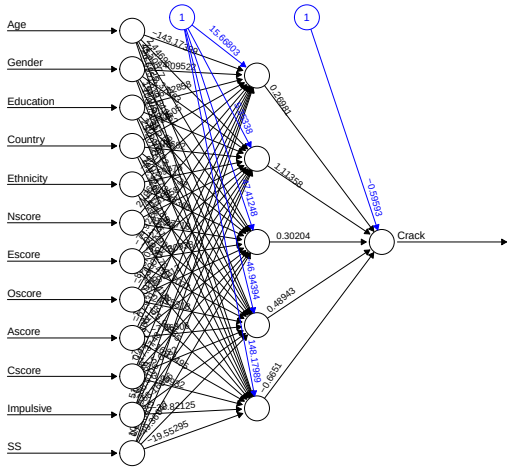
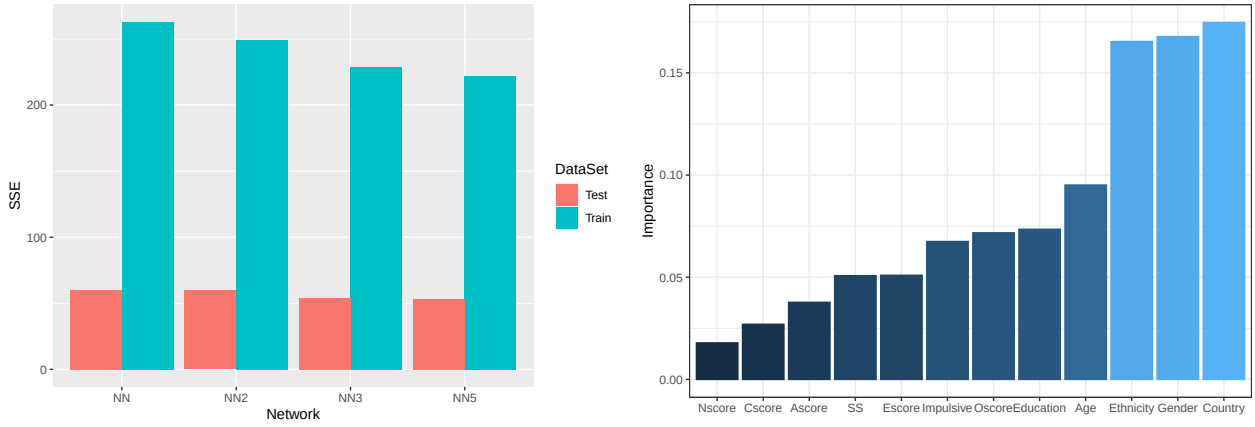


Figure 7: Neural network with 5 hidden layers

In general, training the neural network with too many steps can lead to overfitting, whereas too few may

result in an underfit model, in this case 4 models have been fitted with 1,2,3 and 5 hidden layers. Plot *a* in *Figure 8* shows that the error in the training data reduces with the addition of hidden layers but it stays about the same in the test data, hence there is need to add more layers as it could lead to overfitting and it would make the code slower.



(a) SSE

(b) Importance of NN5

Figure 8: Neural networks SSE and Importance

Plot *b* on *Figure 8* shows that the 3 explanatory variables: Ethnicity, Gender and Country have hihger importance in the analysis than the other. Hence a neural network with only them three as explanatory variables has been plotted

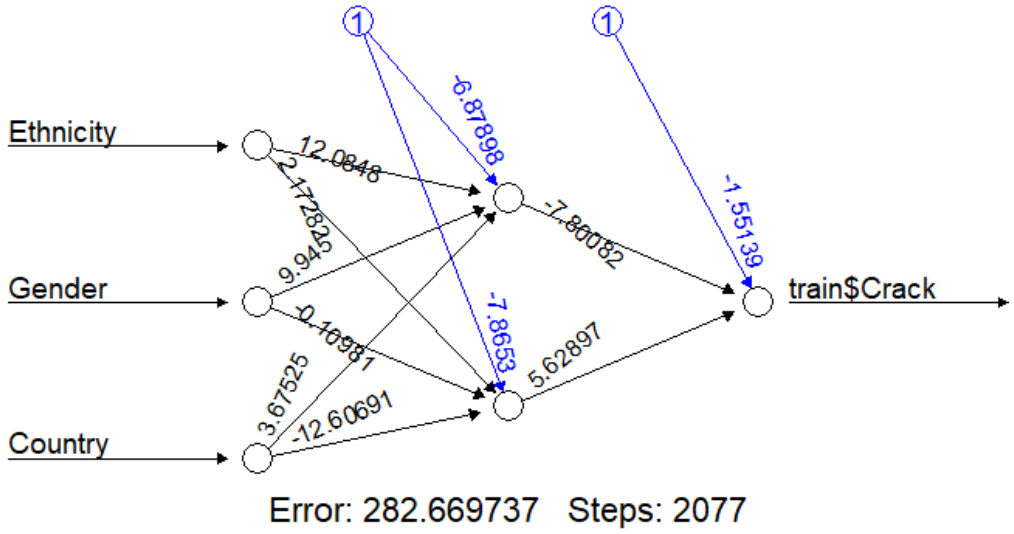


Figure 9: Neural network with explanatory variables Ethnicity, Gender and Country

Error of the neural network in *Figure 9* is 282.67.However, as shown in plot *a* in *Figure 8* this is not the lowest SSE found. Therefore the best neural network found for this data set is NN5.

T-SNE

T-SNE (t-distributed stochastic neighbor embedding) is a popular nonlinear dimensionality reduction technique used for data visualization. Its purpose is to transform high-dimensional data into a low-dimensional space (typically 2D or 3D) while preserving the local structure of the data points. In other words, t-SNE helps us to visualize complex data sets with many variables and to identify patterns or clusters that might not be apparent in the high-dimensional space.

First, all parameters have been set to default values, but $verbose = TRUE$ to see how the function works.

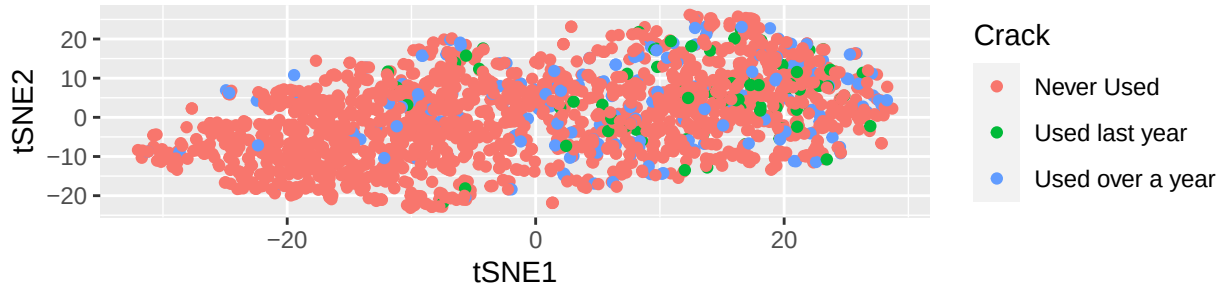


Figure 10: t-distributed stochastic neighbor embedding

In order to optimize, the initial_dims are being set to the number of variables in the input data which is 12.

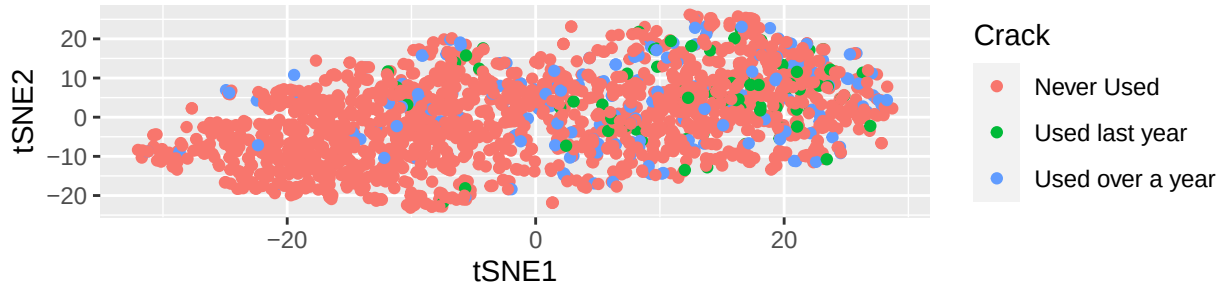


Figure 11: t-distributed stochastic neighbor embedding after dimesionalising

After the dimensionality reduction shown in the figure, the results were not satisfactory, so the model was tuned by adjusting the parameters.

In order to preserve more global features and improve accuracy (which did not significantly increase the running time of the code), we tried to increase the values of the parameters perplexity and theta, with the following results.

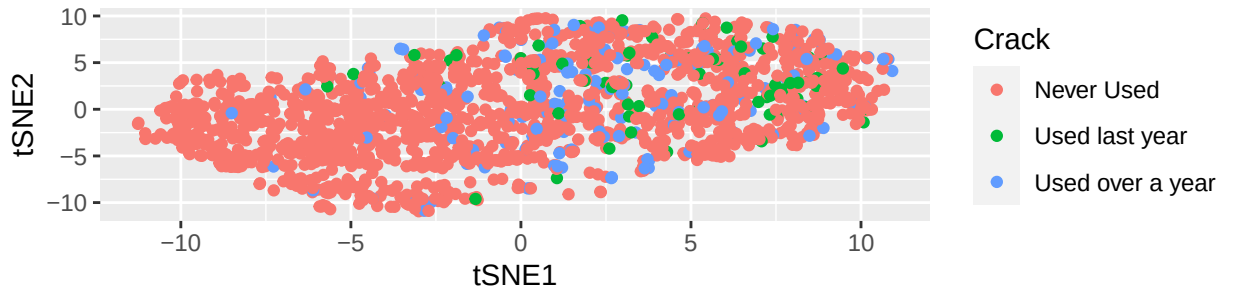


Figure 12: t-distributed stochastic neighbor embedding after tuning

Although the value of error was reduced, the images showed that the classification was still unsatisfactory, so an attempt was made to increase the maximum number of iterations max_iter.

After increasing the maximum number of iterations to 2000, the error did not show a steady decline but started to fluctuate, so the default value was set to 1000 iterations.

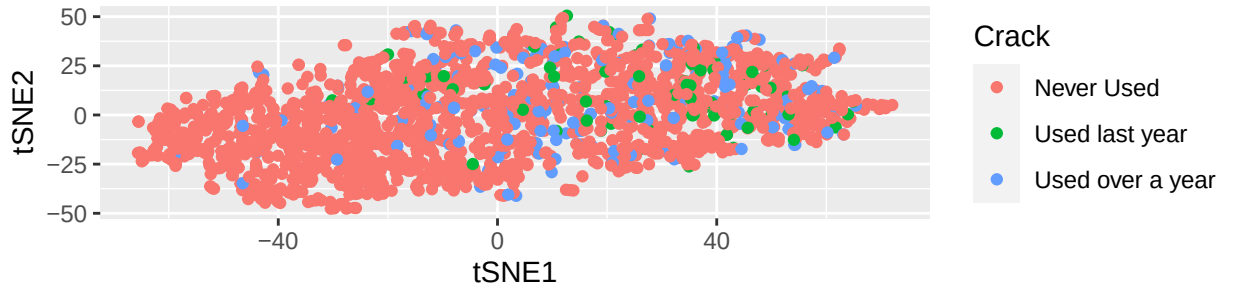


Figure 13: t-distributed stochastic neighbor embedding with 1000 iterations

Through the series of processes showed above, it was clear that the t-SNE method was not very effective in the data for this project.

Adaptive boosting

Adaptive Boosting is learning algorithm in machine learning that combines multiple weak learners to create a strong learner. It works by repetitively training a sequence of weak classifiers of the training data, with the weights updated after each iteration to give more importance to the examples that were misclassified by the previous classifiers.

In each repetition, the algorithm selects a weak classifier that performs better than random guessing on the current training data, and assigns it a weight based on its accuracy. The final classifier is a weighted combination of all the weak classifiers, with the weights determined by their individual accuracies.

Adaptive boosting is effective at handling complex classification problems and achieving high accuracy, even with noisy o data. It is also known for its ability to handle imbalanced datasets, where one class may have much fewer examples than the others. However, it can be sensitive to outliers and may overfit if the weak classifiers are too complex or the number of iterations is too large.

Given that the dataset used for this analysis has a large number of entries in one of the classes and very few in the other two adaptive boosting should give a better result than the classification methods previously used.

Table 13: Prediction table from adaptive boosting

	Never Used	Used last year	Used over a year
Never Used	1265	61	117
Used last year	4	1	2
Used over a year	31	3	17

The model error is 0.146569. This implies the accuracy is about the same as the accuracy gotten for the rest of the classification methods. However, this model does predict data into all the given categories so with a change of parameters within the function it could be a good method.

Results

After performing multiple classification techniques there is one main thing to point out. The dataset used is not very suitable for the analysis performed. Given that as shown in *Table 1* there is many more entries in the group of never used crack (CL0) than any of the other groups combined. The decision was made to use 3 classes instead of all 5 given in the data set as there would be observation in each of the classes.

All the specific results have been shown throughout the report in each of the subsetions of the formal analysis; so this will not go into much detail.

All the formal analysis performed have very similar results, with accuracies around 0.85. This number would usually be very satisfactory, but given the distribution of comsumtion of drug in the dataset used this is level of accuracy is not really useful as it's achieved by setting all the predictions into the Never Used category which is unhelpful and not realistic.

Out of the methods studied within the subject, the first classification tree in the tree method section is probably the best classification method for this data given the mix between categorical and numerical explanatory variables and the limited number of observation withing the drug use levels. Is is also the easiest method to show to the general public and even though this is not a big factor it could come useful if this was to be presented in the future.

Outside of that a more thorough analysis of adaptive boosting would probably give the best results. The main difficulty found in order to do that is the time taken by the computer programmes to run this method.

Conclusion

In summary, the classifications methods used did not give the level of precision wanted. This could be due to the distribution on the data set as most of the participants had never used the drug studied.

Given the methods studied within this course a better approach might have been to use less number of analysis techniques and do a more in depth analysis of the classification method that seemed to work, as for example the LDA and QDA was not a good fit from this data and that was easy to see from the summary plots.

More explanatory variables would be needed to get a more accurate prediction in the future, this could include the use of other drugs; people who smoke, drink or do other drugs might be more likely to have tried crack. Socio-economical variables and health variables might also come useful to perform a better model.

The analysis performed has a high level of accuracy, but given the dataset used this it does not give a useful prediction. There is improvements within both the sampling method and formal analysis that could be changed to try to get a better prediction in future works within the topic.