Machine Learning & Data Mining
Project 2 - Supervised Learning: Classification and Regression

Group 72: Axel Månson Lokrantz, Isak Wilkens, Julius Ekberg Bretto

2023-11-14

Contribution

	Section 1	Section 2	Section 3	Section 4	Exam questions
s232081	30%	33%	40%	30%	33%
s232082	40%	33%	30%	30%	33%
s231401	30%	33%	30%	40%	33%

Exam Problems

Question 2

Answer: B

A matrix is created from the splits and the different classes. We will start by calculating predictions where the largest is kept for later impurity measurements. Beginning with the root the following formula is used where the denominator will be sum of observations in a class across all splits and the numerator will be total observations which is 135.

$$p_i = \frac{\sum_k R_{ki}}{N(r)}$$

The following formula is used for each split, where the denominator is the observations in each split belonging to class i and the denominator total observations in that split:

$$p_i = \frac{R_{ki}}{N(V_k)}$$

Largest proportion for root out of all classes: $\frac{33+4}{135}$
Largest proportions for each split(k) out of all the classes(i): $p_1 : \frac{33}{120}$ $p_2 : \frac{5}{14}$ $p_3 : 1$
The last split contains only one class and is therefore lead to $I(v_3)$ being 0.

Impurity gain for the split is then calculated from:

$$\Delta = I_0 - \left(\frac{N(V_1)}{N(r)} \cdot I(v_1) + \frac{N(V_2)}{N(r)} \cdot I(v_2) + \frac{N(V_3)}{N(r)} \cdot I(v_3) \right)$$

With numbers:

$$\Delta = (1 - \frac{37}{135}) - \left(\frac{120}{135} \cdot (1 - \frac{33}{120}) + \frac{14}{135} \cdot (1 - \frac{5}{14}) + \frac{1}{35} \cdot 0 \right) = 0.0148$$

Question 3

Answer: A

The number of parameters that must be trained in an artificial neural network with an input layer of 7 attributes, a hidden layer with 10 units, and an output layer with 4 outputs is equal to the total number of weights plus the total number of biases. The total number of weights is the sum of the products of each pair of adjacent layers. The total number of biases is equal to the number of output neurons plus the number of hidden neurons.

$$\text{Input Attributes} \times \text{Hidden Units} + \text{Hidden Units Biases} : 7 \times 10 + 10 = 80$$

$$\text{Hidden Units} \times \text{Outputs} + \text{Output Biases} : 10 \times 4 + 4 = 44$$

The total number of parameters that need to be trained in this artificial neural network is:

$$80 + 44 = 124$$

Question 4

Answer: D

It is evident the first splitting condition, A, needs to divide the plot where one end includes only congestion level 1 and 2, and the other part includes all levels but 2. Since the tree specifies that the condition must be false on the left node, this can only be done when $b1 \geq -0.76$, eliminating option A and C. The second splitting condition B should then consequently split congestion level 1 and 2 accordingly, which is done through the condition $b2 \geq 0.03$. This leaves only option D to be the correct answer. Furthermore, to double check the validity of option D, splitting conditions C and D, where $b1 \geq -0.16$ and $b2 \geq 0.01$ respectively, are also correct in dividing the congestion levels.

Question 5

Answer: C

To calculate the total time it takes to construct the table, we need to determine the number of loops. Given that $K = 4$ and there are 5 variables, the number of inner loops is $5 \times 4 = 20$. Since we conduct 5 outer loops, the total number of inner loops becomes $20 \times 5 = 100$.

The total time required to train the neural network and the logistic model within the inner loop is then calculated as:

$$(9\,\text{ms} \times 100) + (25\,\text{ms} \times 100) = 3400\,\text{ms}$$

However, we have not accounted for the time it takes to train and test the data in the outer loop. Therefore, we need to add another 5 loops:

$$(9\,\text{ms} \times 5) + (25\,\text{ms} \times 5) = 170\,\text{ms}$$

The total time it takes to construct the table is the sum of the total time for the inner loop and the outer loop, which becomes:

$$170\,\text{ms} + 3400\,\text{ms} = 3570\,\text{ms}$$

1 Regression, Part A

Link to Data Set

Link to the data set.

1.1 Variable Prediction

In order to successfully perform a linear regression, it is favorable to use continuous variables. The continuous variables in our data set are related to personality traits, such as Escore (Extroversion), Nscore (Neuroticism) and so on. Analyzing the heat map from the first report (see appendix A), it is evident there is a relatively strong correlation between Escore and Oscore (Openness to new experiences). Therefore, the group made a decision to predict the level of Escore based on the level of Oscore. We think that it may be interesting what happens to a persons sense of extroversion the more open they are to new experiences; either they will generally be more extroverted and eager to try new things, or they might be slightly more introverted (perhaps trying new things out of discomfort, therefore the Oscore is still high).

Naturally, we have transformed the data through standardization. This is done through mean centering by subtracting the mean of each of the features of its data points, and dividing each feature with its respective standard deviation in order to scale the data with a mean of 0 and standard deviation of 1. This transformation is useful,

especially in linear regression since it involves a lot of plotting, because it ensures that features are on the same scale.

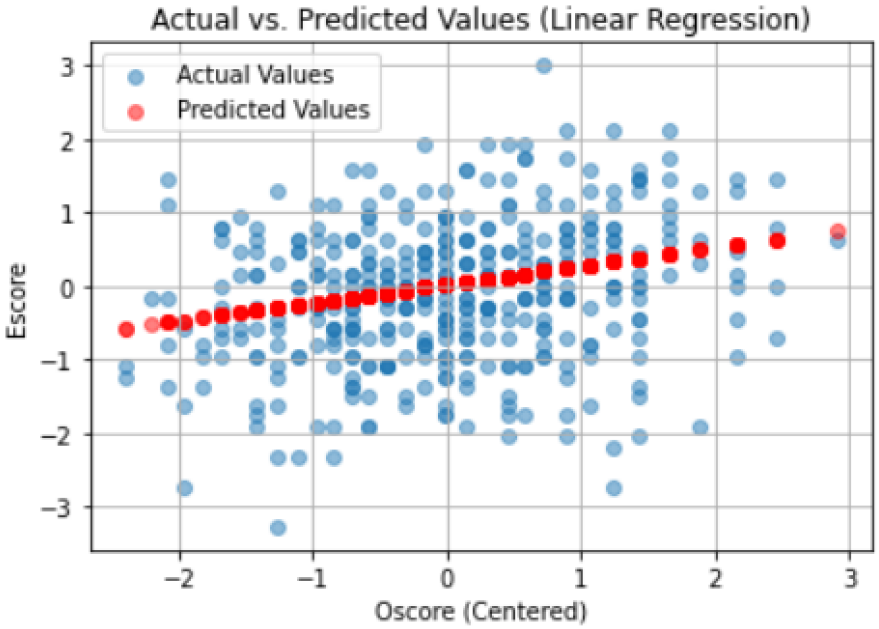

Figure 1: Linear regression of Escore based on Oscore.

As can be seen from the figure above, the Escore seems to increase along with the increase of Oscore, meaning that the respondents in this data set that are generally more open to new experiences also tend to be more extroverted.

1.2 Generalization Error

A regularization parameter lambda was introduced, that adopts values in the range from 10^{-2} to 10^{3}. For each of these values, we have used 10-fold cross-validation to estimate the lowest generalization error. The generalization error is the error of a model on new, unseen data that was not part of the training set. It represents how well the model generalizes its learning from the training data to make accurate predictions on new data, in order to avoid overfitting. The generalization error as a function of lambda can be seen in the figure below. As can be deduced, the graph in figure 2 remains relatively constantly horizontal from 10^{-2} to 10^{1}, upon which it slightly declines to reach a minimum at 10^{2} before a dramatic increase, culminating in a maximum at 10^{3}. Therefore, the best balance between bias and variance is found when lambda is equal to 10^{2}, where the model is "best fitted". So, this value is later used as a regularization term to penalize the large weights. Additionally, a similar test was conducted for an Artificial Neural Network (ANN) model to find the optimal number of hidden layers h^{*}, as can be seen in figure 3 below.

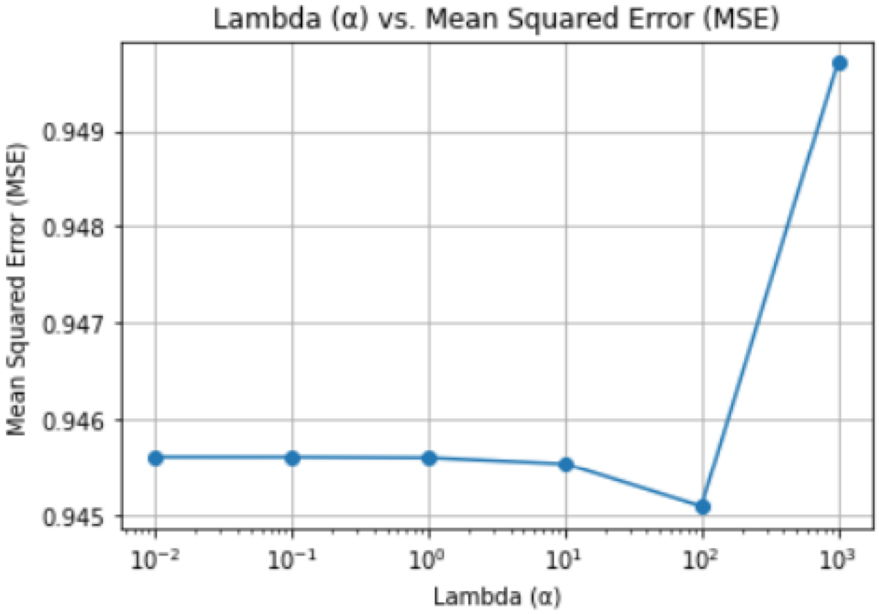

Figure 2: The generalization error as a function of lambda.

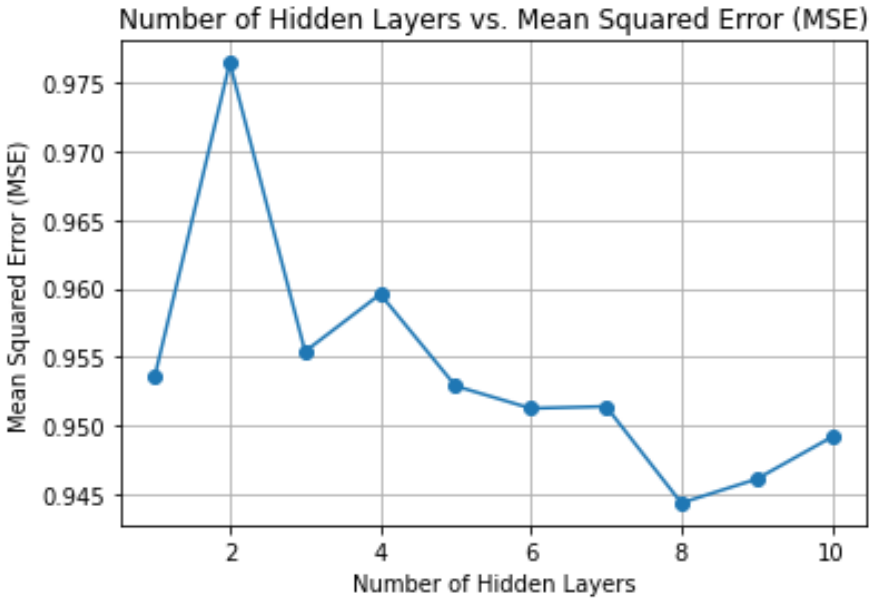

Figure 3: The generalization error as a function of h.

1.3 Individual Attributes and Output

The output y, of the linear model for a given input x can be found using the formula:

$$y = intercept + coefficient * x$$

In the case of our model the intercept is approximately 0 (-0.000163) and the coefficient for 'Oscore' is approximately 0.232. This means that for each increase of a unit in 'Oscore' the predicted 'Escore' increases by 0.232 units.

As an example, an 'Oscore' of 10 would give roughly the estimated 'Escore':

$$y = -0.000163 + 0.232x10 \approx 2.32$$

The effect of an individual attribute in x, in this case 'Oscore', on the output, y or 'Escore', is determined by its corresponding coefficient in the model. In the case of our model, it has a positive coefficient, meaning when x increases so does y. This goes in line with our understanding of an underlying correlation and it seems to make sense that a more extroverted person is more open to new experiences.

2 Regression, Part B

2.1 Two-Level Cross-Validation

Here is a quick rundown of how the two-level cross validation was performed. First we load the data set and processes it by removing unnecessary prefixes and converts all values to integers. We select 'Oscore' as the feature variable and 'Escore' as the target variable we want to predict. We standardize the predictor variable through mean centering, and standard deviation of 1.

Then, we perform a two-level cross validation, where the hyper parameters and test errors for the three different models are calculated each fold and then print the results in a table (Table 1). The program compares the performance of the ANN and ridge regression model using mean squared error (MSE) as the performance metric. The hyper parameter for the ANN is h^* which is the number of hidden layers and the penalty term λ^* for the ridge regression model. The purpose of the penalty term is to shrink the coefficients of the model to reduce the complexity and prevent over fitting.

The baseline model (DummyRegressor) is used to compute the baseline error, which is the error obtained by always predicting the mean of the target variable. Evaluating the baseline error in contrast to the error of our ANN and ridge regression model gives a benchmark against which the performance of the other models can be compared. The interval chosen for h^* was set to [1, 11] and $[10^{-2}, 10^3]$ for λ^*. In 1.2, we predicted that the optimal λ^* to be 10^2 and the optimal number of hidden layers to 8, as seen in table 1 the prediction for h^* correspond to our previous analysis, however the threshold for lambda is much lower than expected as opposed to the predicted value of 10^2 in figure 2.

2.2 Table

Table 1: Two level cross validation, K = 10 folds

Fold	h^*	λ^*	$E_{\text{test\_ANN}}$	$E_{\text{test\_Ridge}}$	Baseline Error
1	9.0	0.0100	0.885984	0.885385	1.005151
2	8.0	0.0100	0.708664	0.711209	0.786252
3	8.0	0.0001	0.727574	0.733328	0.712230
4	5.0	10.0000	0.689305	0.677560	0.764717
5	8.0	0.0001	1.018264	1.018857	1.162917
6	8.0	0.0010	1.024005	1.017696	1.127193
7	9.0	0.0100	0.956177	0.961494	1.042915
8	8.0	0.0010	0.957258	0.960400	1.037440
9	8.0	0.1000	1.366449	1.364267	1.283714
10	3.0	0.0001	1.239402	1.126202	1.042619

2.3 Performance Evaluation

To see if there is any statistically significant performance difference between the fitted ANN, linear regression and the baseline model a paired t-test was done. In this study we are using McNemera's test (setup I) meaning that our conclusions are conditional to the used dataset. The results are shown in the table below.

Table 2: Comparison of Models

Comparison	t-statistic	p-value	95% Confidence Interval
ANN vs Ridge	1.0228	0.3331	[-0.01414, 0.03748]
ANN vs Baseline	-1.1654	0.2738	[-0.11531, 0.03690]
Ridge vs Baseline	-1.9555	0.0822	[-0.10973, 0.00798]

A confidence interval of 95% with a significance value of 0.05 was chosen was chosen for these tests which is most commonly used in statistical testing. Based on the results there was no statistically significant difference between any of the tests. This means that we did not have enough evidence to conclude a difference between the models since they all had a p-value greater than 0.05. In the case of Ridge vs Baseline we see a p-value of 0.0822 which suggests weak evidence that there is a difference between the models. All of the confidence intervals also includes the zero which further supports the conclusion of no statistical difference.

3 Classification

3.1 Classification Problem

Drug use is a divisive issue that can create a wide range of opinions and attitudes among people. Some individuals are very strict and have a no-tolerance policy towards drugs, while others are more open to new experiences and ideas. This can also be seen in analyzing the respondents' personality traits in the data set. That said, the group agreed it would be interesting to further research which people have ever used a specific drug, and which have not.

For instance, we see value in the ability to forecast whether an individual has engaged in the use of a specific drug or not, using their personality traits as indicators. This predictive method serves as a screening tool that can be applied in various contexts. This could for example be used as a warning in normal health evaluations to indicate that some individuals need to be more careful. In the surveys and interviews conducted with the respondents, they were presented with seven choices to describe their level of drug use for each substance, spanning from "Never Used" to "Used in the last day." As a collective decision, the group has opted to create a classification problem that will categorize users as either "Non-users" or "Users" based on their personality traits. The categories "Never used" and "Used over a decade ago" will be combined into the "Non-user" group, while all remaining options will be classified as a "User". This definition will lead to a binary classification problem where "Non-user" is represented by 0 and "User" is represented by 1. In this classification problem we will focus on the drug cannabis. Cannabis was chosen since the use of cannabis is quite evenly spread across users and non-users. Since it is seen as a gateway drug, it can show interesting results on what personality type is more prone to try it.

3.2 Comparison

To compare the performance of our logistic regression model its performance will be compared with two other models: Artificial Neural Network (ANN), and a baseline model. In the logistic regression model, λ has been used as a complexity controller parameter to prevent overfitting. Through regularization and the addition of this penalty term the model is constrained to reduce the complexity based on learned parameters. For the ANN model the complexity controlling parameters is numbers of hidden layers where we have searched for the optimal between 1 to 10. The baseline model used is a "DummyClassifier" with the strategy set to "most frequent" which predicts the most frequent label in the training set for all instances in the test set.

3.3 Comparison Result

The test for these models calculate the test error rates for each model across 10 folds of the data along with the found optimal parameters for the logistic regression and ANN models. This will allow us to compare performance and understand how well the models are learning from the data. Table 3 shows the output results from the comparison test:

"Etest" value show the test error rates for each of the models, h* is the number of hidden layers in the ANN model, λ^* is the optimal regularization parameter for the logistic regression model. The best general performance would be the model that has the lowest average error rate across all 10 folds. The variability of the error rate across folds is also interesting since a model with low average error rate but high variability might not perform consistently on new data.

Table 3: Two level cross validation, K = 10 folds

Fold	h^*	λ^*	$E_{\text{test\_ANN}}$	$E_{\text{test\_Logistic}}$	Baseline Error
1	5	10.0	0.306878	0.285714	0.328042
2	6	0.1	0.275132	0.264550	0.328042
3	3	10.0	0.301587	0.322751	0.328042
4	7	1.0	0.216931	0.253968	0.328042
5	3	1.0	0.238095	0.227513	0.328042
6	8	1.0	0.308511	0.303191	0.329787
7	6	0.1	0.287234	0.292553	0.329787
8	4	0.1	0.244681	0.234043	0.329787
9	6	1.0	0.234043	0.281915	0.329787
10	2	0.1	0.202128	0.202128	0.329787

In the case of average error rate the ANN model has the lowest of approximately 0.2615 followed by logistic regression with approximately 0.2668 and finally baseline that gave av average error of 0.3289. By looking at the range of the error rate, we can find the variability of these results. For the Ann model the range is 0.1064, for the logistic model the range is 0.1206 and for the baseline it is 0.001745. This again show that the ANN performs best but closely followed by the logistic regression model. The baseline model is consistent but more consistently incorrect than the other models. The hyperparameters vary across folds, suggesting that the models might be sensitive to the specific split of the data.

3.4 Statistical Evaluation

Similarly to 2.3 we use 'Setup I' to statistically evaluate our three models. The null hypothesis we use is that there is no significant difference between the models.

Table 4: Comparison of Models

Comparison	t-statistic	p-value	95% Confidence Interval
Logistic vs ANN	-0.7371	0.4798	[-0.0216, 0.0109]
Logistic vs Baseline	-5.2156	0.0005	[-0.0890, -0.0352]
ANN vs Baseline	-5.4158	0.0004	[-0.0955, -0.0392]

Logistics vs ANN: The t-statistic is approximately -0.7371, and p-value 0.48. This

indicates that there is no significant difference between the ANN and the Logistic model as the p-value is greater than 0.05. The confidence interval includes zero, which further confirms the lack of significant difference.

Logistics vs Baseline: The t-statistic is approximately -5.216, the p-value is 0.0005, which indicates significant difference between the Logistic and Baseline model. The confidence interval does not include zero, which further strengthen our evidence against the null hypothesis.

ANN vs Baseline: The t-statistic is approximately -5.416, the p-value is 0.0004, which indicates significant difference between the ANN and Baseline model. The confidence interval does not include zero, which further strengthen our evidence against the null hypothesis.

In summary, both the Logistic model and the ANN perform significantly better than the Baseline model.

3.5 Logistic Regression

The Logistic Regression model has an accuracy of 0.73, which means that it correctly predicts the outcome in about 73% of the cases. The confusion matrix provides more detailed information about the model's performance. The matrix is as follows:

	Predicted: No	**Predicted: Yes**
Actual: No	54	60
Actual: Yes	42	221

Table 5: Confusion Matrix

The optimal regularization parameter λ for our model is approximately 29.76. The logistic model uses a sigmoid function to transform its output into a probability that belongs to a certain class. The function can map any number into a value between 0 and 1, which can be useful in cases such as ours (binary classification). The model calculates a weighted sum of the input feature plus a bias term and applies the logistic function to get the probability of which class it belongs. If the probability is greater than 0.5, the model predicts a positive class, otherwise it predicts a negative class. The value of the weights and λ are learned from the training data and then applied on test data. In the linnear regression model we used Oscore to predict Escore and in the logistic regression we predicted if a person use cannabis based on the personality traits Nscore, Escore, Oscore and Cscore. Potentially, the linear regression model could have performed much better as a multiple linear regression model (MLR) instead, where we used a mix of the personality features to predict someone's Escore.

4 Discussion

4.1 Key Takeaways and Learnings

We came to the conclusion that there wasn't any statistically significant performance difference between the fitted ANN model, the linear regression, and the baseline (dummy) model. Testing with a confidence interval of 95%, we got p-values greater than 0.05 for all of the three tests, indicating that we did not have enough evidence to conclude a difference between the models. In other words, it is not completely unlikely that the observed values are a product of chance.

A reason for these less-than-ideal p-values can be linked to the feature selection of the regression. Since we only used one feature, namely Oscore, to predict the target variable Escore, this can be the culprit, because if Oscore is not strongly enough correlated with Escore, the models in question might not be able to identify and learn a set pattern and therefore the machine learning models tested are rendered useless. Observing the heat map in appendix A, we can see that the correlation is not particularly weak, but not super strong either relative to for example the correlation between some demographics and drug use. The difference between this paper and the published study, which will be further discussed in 4.2, is that the latter uses several features to predict drug use (not just one), and this yields a better result.

Conclusively, the quality and relevance of the input data is therefore crucial for the performance of machine learning models, and if the input data doesn't contain enough patterns for predicting the target variable, or if the data simply consists of too few features to detect a pattern, the model will be unsuccessful in predicting output. Therefore, a better alternative would have been to use a multi-variable linear regression model instead where the Escore was predicted based on a variety of personality traits as opposed to just the Oscore.

Additionally, the models (Ridge, MLP etc) are somewhat simple models, and if the relationship between the feature and the target variable are incredibly complex, these models might not be apt in capturing it. Therefore, this could also be a cause; however, since this is a very simple problem, this is unlikely.

4.2 Previous Analysis

The data set has previously been analyzed by others, and was a part of a study called "The Five Factor Model of personality and evaluation of drug consumption risk" authored by Elaine Fehrman, Evgeny M. Mirkes, Awaz K. Muhammad, Vincent Egan and Alexander N. Gorban, also the creators of the data set. Interestingly, they conducted a similar classification problem as ours, defining users as "users" or "non-users". However, they made some different choices, such as defining a variable T-score that divides test scores into either high, neutral or low intervals, as well as not limiting themselves to just one drug (like we did in this report, choosing cannabis). They found that for almost all drugs, the Nscore and Oscore was relatively high or neutral, the Cscore relatively low, but that the Escore fluctuates depending on the drug in question (neutral for cannabis).

This differs from our report because we focused only on cannabis drug usage in the classification task, and predicted it using Escore, Nscore, Oscore and Cscore – therefore, we didn't need to divide the scores along a "T-score" interval – but judging by the heat map in appendix A, the personality traits seem to be scored similarly.

The published study mentions the use of an exhaustive search to select the most effective subset of input features, whereas in our study we unanimously selected attributes Escore and Oscore for the regression task and Escore, Oscore, Cscore and Nscore for the classification task. This is a huge difference, since the published paper has scientific backing on their choice of "predictor variables", while we made the decision purely based off of our own conceptions and the heatmap found in appendix A. Regarding the classification task, we still got similar results (around 75% accuracy) to the published study.

Furthermore, the published paper uses a variety of different classification methods such as decision trees, random forests, k-nearest neighbors, linear discriminant analysis, Gaussian mixture, probability density function estimation, logistic regression, and naive Bayes. In our report, we have only used logistic regression and ANN, which entails that our results and models might not be as accurate and complex. However, seeing as our scope is a lot more limited than the paper's scope (we limited ourselves to a singular binary classification problem), we argued that this was indeed not a problem and would not compromise the trajectory of the report.

In conclusion, both studies demonstrate the potential of using machine learning techniques to predict drug use based on personality traits. However, the abstract suggests that different drugs may have different patterns of use, and a variety of models and features may be needed to accurately predict use for each drug. Our study provides a solid foundation in examining the correlation between Escore and Oscore (as well as Escore in relation to the other personality traits) and specifically the use of cannabis depending on personality traits. We are confident our study could potentially be expanded in future work to be used in similar settings as an additional element or complement to a larger study.

5 Appendix

Appendix A: Heatmap

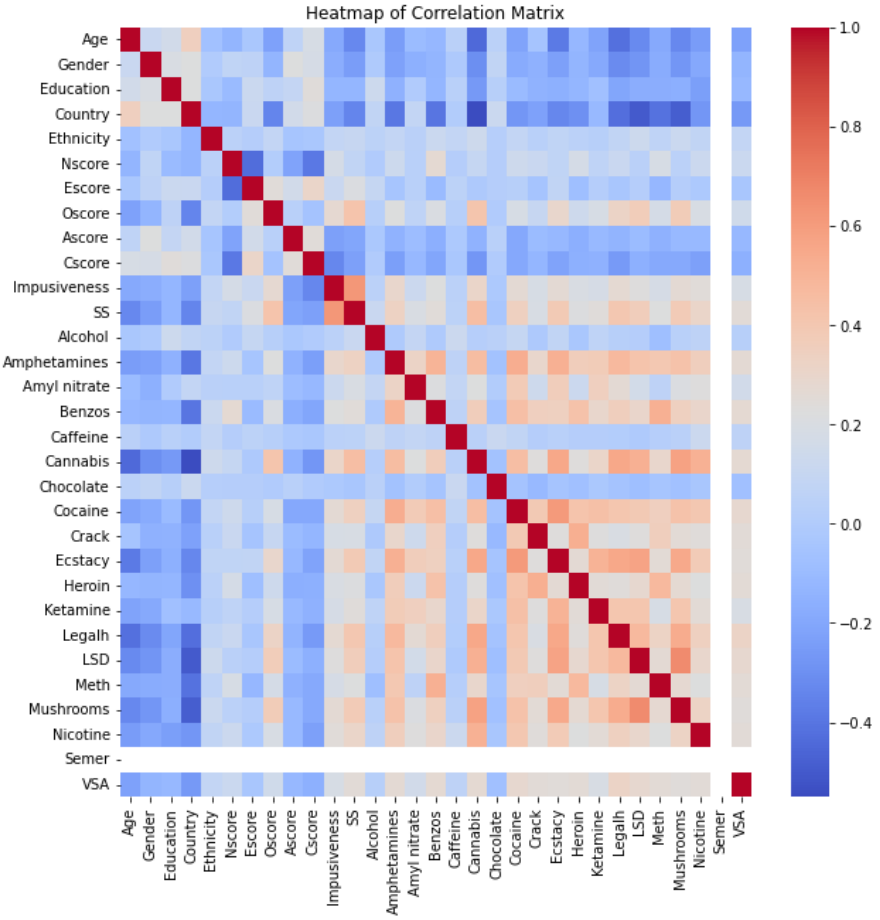

Figure 4: The heatmap from project report 1.