A Comparison of Naïve Bayes and Random Forest on Predicting Drug Consumption

Jessica Guan

Description and Motivation

This project explores the application of Naive Bayes and Random Forest to solve a binary classification problem of classifying an individual as a user or nonuser of different types of drugs based on their personality traits. We will compare and analyse the performance of both models. This study is based on the previous research done by E. Fehrman et al (2020) [5].

Data Exploration

- Dataset: Drug Consumption from UCI Machine Learning Repository
- The dataset contains information about 1885 individuals' scores on their personality traits and consumption of different types of drugs.
- The models use the personality traits as the features, based on the study by E. Fehrnman et al (2020) [5]. The personality traits are openness (Oscore), conscientiousness (Cscore), extraversion (Escore), agreeableness (Ascore), neuroticism (Nscore), impulsiveness (Impulsive), and sensation seeking (SS), and each trait has a score in the range of -4 to 4.
- The drugs included in the original dataset are alcohol, amphetamines (Amphet), amyl nitrites (Amyl), benzodiazepines (Benzos), caffeine (Caff), cannabis, chocolate (Choc), cocaine (Coke), crack, ecstasy, heroin, ketamine, legal highs (Legalh), LSD, meth, mushrooms, nicotine, Semer (fictional drug to serve as a control), and volatile substances (VSA). In this project, we will only be classifying the highlighted drugs.
- Drug consumptions are represented in the dataset as CL0 (Never used), CL1 (Used over a decade ago), CL2 (Used in last decade), CL3 (Used in last year), CL4 (Used in last month), CL5 (Used in last week), CL6 (Used in last day).
- In Figure 1, we can observe the correlation between the 7 personality traits and the drugs. Oscore (openness), Impulsiveness, and SS (sensation seeking) seem to have a higher positive correlations with drug consumption, while Ascore (agreeableness) and Cscore (consciousness) have slightly negative correlations with drug consumption.

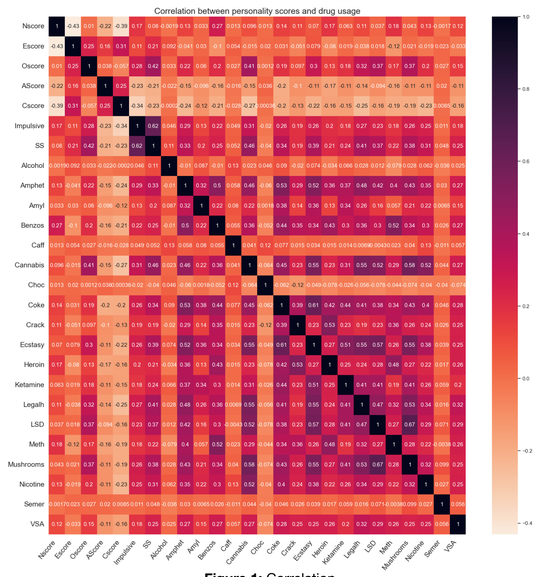


Figure 1: Correlation

- In Figure 1, we can see the the distribution of data is imbalanced. To provide more accurate results, I used Synthetic Minority Oversampling Technique (SMOTE) to balance the data.

Minority Oversampling Technique (SMOTE) to balance the data.

- Then, to make this a binary classification problem, I added a new column in the data preprocessing to classify an individual as a nonuser of a drug if they responded with CL0, CL1, or CL2, and as a user of a drug if they responded with CL3, CL4, CL5, or CL6.

Figure 2: Data

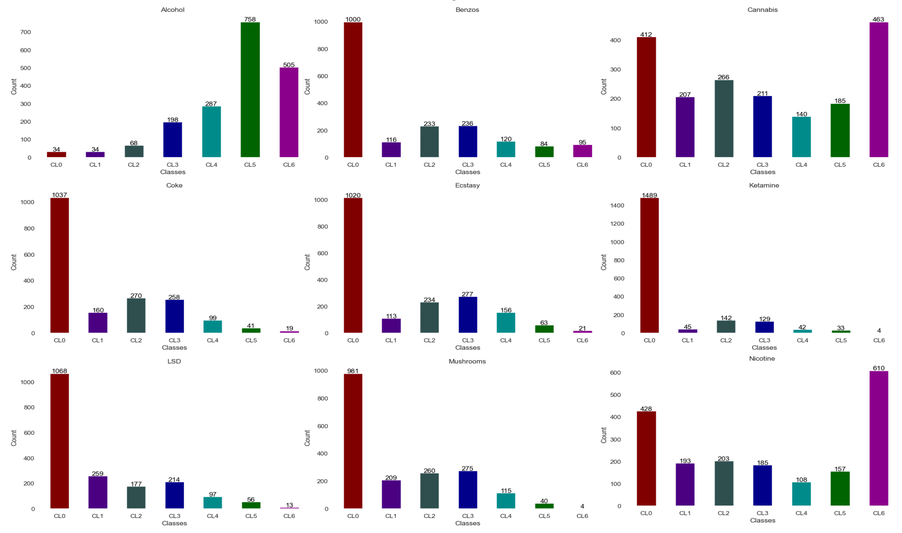


Figure 3: Statistics

Predictor	Min	Max	Mean	Median	Std
Oscore	-3.27	2.9	-0.00023	-0.019	0.996
Cscore	-3.46	3.46	-0.00038	-0.0067	0.998
Escore	-3.27	3.27	0.00014	0.0033	0.998
Ascore	-3.46	3.46	0.00024	-0.017	0.997
Nscore	-3.46	3.27	-0.00012	0.043	0.998
Impulsive	-2.56	2.9	0.0073	-0.22	0.955
SS	-2.08	1.92	-0.0027	0.0799	0.964

*Oscore = openness, Cscore = conscientiousness, Escore = extraversion, Ascore = agreeableness, Nscore = neuroticism, SS = sensation seeking

Hypothesis

- From research on random forest and naive bayes and from the research paper on this dataset (E. Fehrman et al, 2020) [5], it's likely that random forest will perform better than naive bayes.
- Random forest usually produces highly accurate classification results.
- Naive bayes assumes feature independence given the class, and I do not think this as an accurate assumption about these particular features in this dataset.

Methodology

- After the data preprocessing steps described in the "Data Exploration" section, the data was split into a training set (80%) and testing set (20%).
- Naive bayes and random forest models were built in MatLab.
- The hyperparameters were tuned and optimized (further detail in "Optimizing hyperparameters and results" section below)
- 5-fold cross validation was performed on the training models to gather their performance metrics such as accuracy, precision, k-fold loss, and confusion matrix.
- The performance metrics were examined to determine which hyperparameters had the best performance for each model and the desired target.
- After determining the hyperparameters, the model was trained on the whole training set.
- Then, the model was used on the testing set.
- Performance metrics were also gathered in the training and testing phase.

About the Algorithms

Naïve Bayes

- Naive bayes is a learning algorithm that can be used for classification in machine learning. It is called "naive" because the algorithm assumes when the class is given, the features are independent. Although in practice, we cannot assume independence, naive bayes still delivers competitively accurate classifications. (Webb, G.I. et al, 2011) [1]
- Naive bayes is based on Bayes theorem, so it uses the joint probability of p(x, y), where x are the predictors or features and y is the the target class, and calculates p(y|x) to predict the class. (Andrew Y Ng et al, 2001) [4]
- Naive Bayes produces the optimal results in binary classification problems with categorical features. (I. Rish, 2001) [2]

Advantages

- A simple and easy learning classifier to implement
- Fast and efficient runtime
- Produces accurate classifications when independence is true

Disadvantages

- Feature independence is not always true for real-life scenarios
- Cannot be used for regression

Random Forest

- Random forest is a learning algorithm that can be used for classification and regression in machine learning.
- Random forest builds upon the decision tree algorithm by creating and and using multiple decision trees.
- It is known for its high classification accuracy, and a good algorithm to be considered when dealing with unbalanced or missing data. (Kendall Lemons)
- It is an ensemble of decision trees, where each decision tree use only a subset of the data
- The final output is decided by the majority of the results from the individual decision trees in the random forest. (L. Breiman, 2001) [6]

Advantages

- High accuracy
- Can be used for classification and regression
- Lower chances of overfitting
- Can handle unbalanced or missing data

Disadvantages

- Long runtime
- Not good for small datasets

Optimizing Hyperparameters and Experimental Results

Naïve Bayes

- Used grid search to iterate through 3 different distribution options: normal, kernel, and multivariate multinomial (mvmn), and 2 priors: uniform and empirical.
- Tested results for data from the original dataset and after using SMOTE. In predicting cannabis users where the data was more balanced, the results were similar. For consistency, I used the SMOTE data. For imbalanced data like in ketamine users, the results were slightly more accurate after applying SMOTE.
- Every drug had a different hyperparameters that produced the best results. For example, cannabis performed similar with normal and kernel distribution, while ketamine performed better with kernel distribution.
- The confusion matrices shown are for the hyperparameters that were chosen.
- This was done for each drug of interest

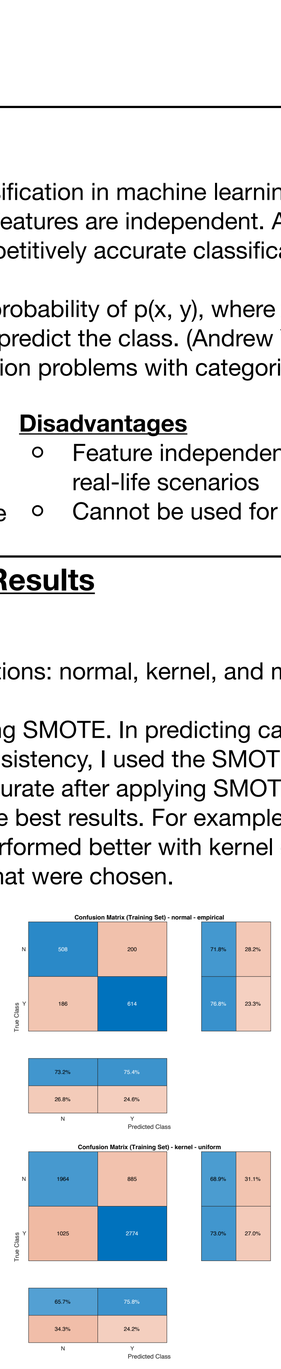
Figure 4: Naive bayes hyperparameter tuning results

Normal	Kernel	Cannabis User Data	Normal	Kernel
Accuracy: 0.746 Precision: 0.726 K-fold loss: 0.256	Accuracy: 0.744 Precision: 0.722 K-fold loss: 0.265	**Uniform**	Accuracy: 0.751 Precision: 0.704 K-fold loss: 0.25	Accuracy: 0.751 Precision: 0.711 K-fold loss: 0.254
Accuracy: 0.744 Precision: 0.732 K-fold loss: 0.255	Accuracy: 0.746 Precision: 0.729 K-fold loss: 0.261	**Empirical**	Accuracy: 0.7592 Precision: 0.732 K-fold loss: 0.241	Accuracy: 0.768 Precision: 0.748 K-fold loss: 0.24

Original dataset SMOTE dataset

Normal	Kernel	Ketamine User Data	Normal	Kernel
Accuracy: 0.684 Precision: 0.197 K-fold loss: 0.332	Accuracy: 0.696 Precision: 0.209 K-fold loss:	**Uniform**	Accuracy: 0.638 Precision: 0.686 K-fold loss: 0.363	Accuracy: 0.71 Precision: 0.768 K-fold loss:
Accuracy: 0.86 Precision: 0.25 K-fold loss: 0.145	Accuracy: 0.865 Precision: 0.3 K-fold loss: 0.145	**Empirical**	Accuracy: 0.632 Precision: 0.638 K-fold loss: 0.369	Accuracy: 0.709 Precision: 0.718 K-fold loss: 0.296

Original dataset SMOTE dataset



Random Forest

- Used 'OptimizeHyperparameters' options in 'fitcensemble' to get the optimized hyperparameters for the ensemble method, NumLearningCycles, MinLeafSize, and MaxNumSplits.
- Like for naive bayes, I tuned the hyperparameters in in the random forest model to each drug of interest. Cannabis and ketamine will be used as examples, more details in the supplementary material on other drugs.
- I tested the results on both the original dataset and the SMOTE dataset, as random forest could produce better results with a larger dataset.
- For cannabis:
  - Method: Bag, NumLearningCycles: 387, MinLeafSize: 1, MaxNumSplits: 2102
- For ketamine:
  - Method: Bag, NumLearningCycles: 487, MinLeafSize: 1, MaxNumSplits: 6351

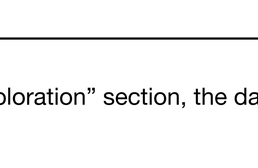


Figure 6: Cannabis

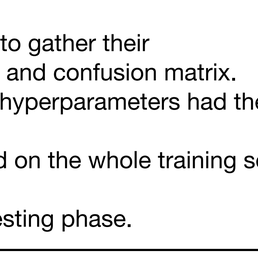
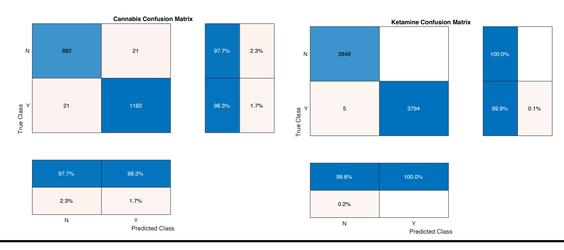


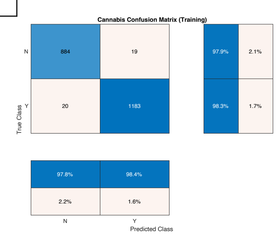
Figure 7: Ketamine

Cannabis User Data	Original dataset	Ketamine User Data
0.809	**Accuracy**	0.89
0.813	**Precision**	NaN
0.251	**K-fold loss**	0.111

Cannabis User Data	SMOTE dataset	Ketamine User Data
0.98	**Accuracy**	0.999
0.977	**Precision**	1
0.176	**K-fold loss**	0.275

Figure 5: Random forest hyperparameter tuning results





Training and Testing Results

Naïve Bayes

Cannabis (Training) NB

Accuracy	0.759
Precision	0.732
K-fold loss	0.2422

Cannabis (Testing) NB

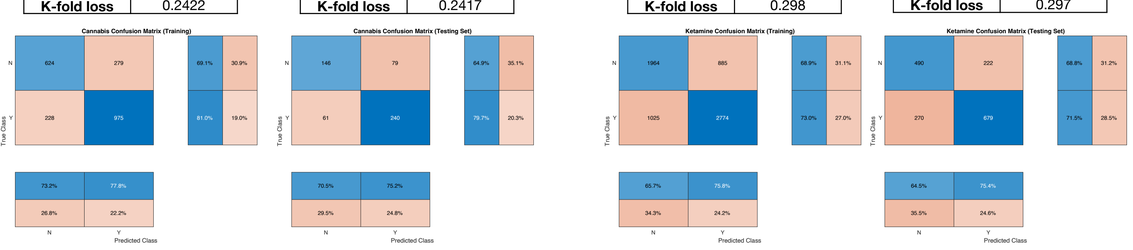
Accuracy	0.759
Precision	0.705
K-fold loss	0.2417

Ketamine (Training) NB

Accuracy	0.71
Precision	0.758
K-fold loss	0.298

Ketamine (Testing) NB

Accuracy	0.71
Precision	0.754
K-fold loss	0.297



Random Forest

Cannabis (Training)

Accuracy	0.981
Precision	0.978
K-fold loss	0.1895

Cannabis (Testing)

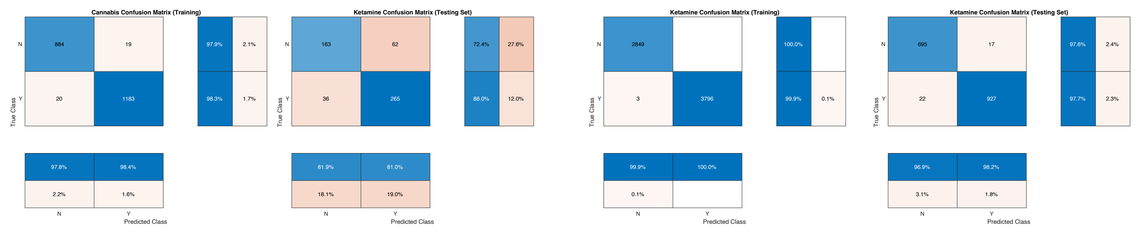
Accuracy	0.981
Precision	0.819
K-fold loss	0.183

Ketamine (Training)

Accuracy	0.9996
Precision	1
K-fold loss	0.0271

Ketamine (Testing)

Accuracy	0.9996
Precision	0.982
K-fold loss	0.0269



Analysis and Evaluation

- In the training results, random forest performed 31.85% better in accuracy, 33.61% better in precision, and had 27.09% less k-fold loss than naive bayes on the cannabis data. Random forest also performed 40.79% better in accuracy, 31.93% better in precision, and had 91.21% less k-fold loss than naive bayes on the ketamine data.
- In the testing results, random forest performed 34.77% better in accuracy, 33.69% better in precision, and had 12.78 less k-fold loss than naive bayes on the cannabis data. Random forest also performed 42.05% better in accuracy, 32.1% better in precision, and had 84.56% less k-fold loss than naive bayes on the ketamine data.
- In conclusion, random forest performed significantly better than naive bayes on the data used for this project. This could be attributed to the naive bayes' independent assumption about the feature variables. In this dataset, the features are the seven personality traits (openness, consciousness, extraversion, agreeableness, neuroticism, impulsiveness, and sensation seeking). In the correlation graph (Figure 1), we can observe that there are traits that are correlated with each other. For example, openness and impulsiveness are positively correlated with sensation seeking, while impulsiveness is negatively correlated with consciousness. In psychological personality traits of people, we cannot assume independence.
- Random forest is better for datasets that are complex, while naive bayes performs better on simpler datasets.
- Random forest is also an ensemble of decision trees, so it is robust and produces highly accurate results.
- When choosing between these two algorithms, the nature of the dataset is important. From the different choices of drugs from the dataset I could have used as my target variable, I chose to mainly focus on cannabis and ketamine. I chose cannabis because it was the most balanced, while ketamine was heavily skewed. I wanted to see the differences that this would make in both algorithms. I also tested the original dataset for both drugs as well as the dataset after applying SMOTE. For naive bayes, the original and SMOTE dataset for cannabis gave very similar performance results because the original data for cannabis was relatively balanced. However, the SMOTE data for ketamine produced better results than the original ketamine data because it was heavily skewed to nonusers.
- Although I originally thought that random forest would produce similar results for the original dataset and the SMOTE dataset because of its ability to handle imbalanced data, this was false. The results for the SMOTE data were better. This can be attributed to the fact that after applying SMOTE, data was synthesized resulting in a larger dataset, and random forest performs better with larger datasets.
- The runtime differences between the two algorithms were very different. Naive bayes took around 28 seconds to perform grid search to tune the hyper parameters, and around 9 seconds for the training and testing model to run. Random forest took around 10 minutes and 19 seconds to tune the hyperparameters, and around 1 minute and 10 seconds for the training and testing model to run.

Lessons Learned

- Pay attention to distribution of the dataset, like if it is balanced or imbalanced.
- Understand the nature of the data, like the relationships between the features themselves and the relationship between the features and target.
- Understand the data when choosing hyperparameters for tuning

Future Work

- Explore different data balancing techniques such as ADASYN.
- Use more performance metrics such as F1 and recall.
- Get ROC curve

References

[1] G. I. Webb, E. Keogh, R. Miikkulainen, R. Miikkulainen, and M. Sebag, "Naïve Bayes," *Encyclopedia of Machine Learning*, pp. 713–714, 2011, doi: https://doi.org/10.1007/978-0-387-30164-8_576
[2] I. Rish, "(PDF) An Empirical Study of the Naïve Bayes Classifier," *ResearchGate*, Jan. 2001. https://www.researchgate.net/publication/228845263_An_Empirical_Study_of_the_Naive_Bayes_Classifier
[3] K. Lemons, "A Comparison Between Naïve Bayes and Random Forest to Predict Breast Cancer," *International Journal of Undergraduate Research and Creative Activities*, vol. 12, no. 1, p. 1, Oct. 2020, doi: https://doi.org/10.7710/2168-0620.0287
[4] Andrew Y. Ng , Michael I. Jordan, "On Discriminative vs. Generative classifiers: A comparison of logistic regression and Naive Bayes ", 2001, https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf
[5] E. Fehrman, V. Egan, A. Gorban, J. Levesley, E. Mirkes, and A. Muhammad, "Personality Traits and Drug Consumption A Story Told by Data Springer," 2020. Available: https://arxiv.org/pdf/2001.06520.pdf
[6] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: https://doi.org/10.1023/a:1010933404324.