Classifying Drug Abusers Using Personality Data

Leowell Bacudio, Aditya Suresh, Rui Wu

December 10, 2019

1 Introduction

Many people in the workforce are familiar with taking drug tests as part of the hiring process. These tests are included to dissuade employees from abusing drugs, as drug abuse has been clinically shown to have a negative impact on productivity. In addition, drug tests are used to prevent hiring those who use illicit drugs, identify and combat early signs of drug abuse, and provide a safe work space for employees.[1] While drug tests are a great tool for maintaining employee quality, they can also be somewhat expensive with most drug tests ranging from $10 to $30.[2] For large companies that regularly test their future and current employees, these prices can quickly add up, and with a small number of working adults who are characterized as illicit drug abusers, unnecessary resources are expended on drug testing employees who do not fall into the same category.

The goal of this project is to predict whether or not a person may be using illicit drugs through a personality test. In addition to a few basic questions, such as age and education, a respondent would be asked to take a personality test, in which the responses would be passed into our model to output the likelihood of that person being an illicit drug abuser. In the event that our model predicts a respondent to be an illicit drug abuser, a company can request that person to take a drug test, while excluding those who had an opposite prediction. By having the ability to distinguish likely illicit drug abusers from others, a company can save money by not having to expend additional resources in unnecessary drug tests.

2 Dataset Description

We utilized the Drug Consumption Data Set from the UCI Machine Learning Repository[3] to create, train, test, and validate our model. The dataset contains the demographic features of 1885 respondents, which include each respondent's age, gender, education, and ethnicity. In addition, each respondent's personality was also measured in the form of pre-quantified scalar variables. These personality tests measure a respondent's NEO-FFI-R (neuroticism, extraversion, openness to experience, agreeableness, and conscientiousness), BIS-11 (impulsivity), and ImpSS (sensation seeking). Lastly, each respondent included their frequency of usage for various recreational drugs, including alcohol, nicotine, marijuana, cocaine, heroin, and many others. This frequency of usage was then categorized into several responses: Never used or used over (a decade ago, last decade, last year, last month, last week, and last day).

For the model, we added an additional parameter called Illicit Use which represents the event in which a person responded to using any illicit drug within the past month. We also reduced our data set by re-categorizing parameters that we found to have no relationship with one another (gender, country, and ethnicity), as well as removing all illicit drug parameters. Our final data set consisted of age, education, and personality scores, along with a small number of legal drug parameters that were one-hot encoded to be machine-learning compatible.

3 Methods

To create the most accurate model, we utilized four different machine learning techniques: Logistic Regression, Naive Bayes, Support Vector Machines (SVM), and Neural Networks. Logistic Regression, Naive Bayes, and SVM were all implemented using the sci-kit-learn machine learning library while the neural network was implemented using Keras. For each technique, we modified the training parameters and documented how the technique and modification affected the results. Unspecified training parameters were set to their default values. Each method was cross-validated using k-fold cross-validation of 4 folds. Due to the original dataset's uneven distribution of recorded data, we shuffled the data during k-fold validation using random-state: 0.

3.1 Logistic Regression

We made two logistic regression models, one trained on the original dataset, and the other trained on a modified dataset in which every parameter was treated as a categorical variable and one-hot encoded. The purpose of conducting the second test was to observe performance changes by assuming independence between the values of each parameter. (Note: After one-hot encoding each parameter, the dataset contained 2,000+ columns. Due to a large number of columns, the other techniques are trained only on the original dataset.)

3.2 Naive Bayes

We made two Naive Bayes models, one assuming feature likelihood is Gaussian distributed, and another assuming feature likelihood is Bernoulli distributed. Because our data is quantified, it is very easy to binarize our data set, making it a good fit for Bernoulli Naive Bayes. We compared the Bernoulli Naive Bayes to the standard Gaussian Naive Bayes to observe any differences in performance.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2})$$

Figure 1: Gaussian Distribution

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

Figure 2: Bernoulli Distribution

3.3 Support Vector Machines (SVM)

When making SVM's, we focused our attention on the way different kernels affected our model. We made 7 SVM's to test 4 different kernel methods: linear, gaussian radial basis function (rbf), sigmoid, and polynomial. The equations for these kernels are shown below:

$$k(x, y) = x^T y + c$$

Figure 3: Linear

$$k(x, y) = exp(-\gamma||x - y||^2)$$

Figure 4: Gaussian Radial Basis Function

$$k(x, y) = tanh(\alpha x^T y + c)$$

Figure 5: Sigmoid

$$k(x, y) = (\alpha x^T y + c)^d$$

Figure 6: Polynomial

For the polynomial kernels, we tested using 2, 3, 4, and 5 dimensions. All other training parameters for the SVM's are set to their default values.

3.4 Neural Networks

Given the structure of our data, our neural networks were all sequential models with a varied number of layers as well as different activation functions. We created a total of three neural networks: a basic one; one with additional layers that utilized a combination of tanh/ReLU activation functions; and one that also had additional layers, but utilized Leaky ReLU as the activation function. We kept the number of epochs and the batch size constant between all three neural networks. For all three networks, we used binary cross-entropy for loss and Adam as the optimizer.

Our basic neural network consisted of 4 dense layers with three intermediate layers containing 64, 32, and 16 nodes along with one output layer with one node. The first three used layers used ReLU as the activation function and the final layer used sigmoid.

Our second neural network consisted of 6 dense layers, with the 5 intermediate layers containing 128, 64, 32, 16, and 8 nodes and our output layer containing one node. The first three layers used ReLU, the next two used tanh, and the output layer used sigmoid.

Lastly, our third neural network used Leaky ReLU as the activation function for the intermediate layers, which had the same structure as our neural network with the exception of additional layers. This time, however, the activation function for the first 5 layers was Leaky ReLU, with our final output layer using sigmoid.

4 Analysis

The performance of the models was ranked based on two criteria: maximizing the overall accuracy while minimizing the false-negative rate. Although overall accuracy was the most important factor in determining the best model, minimizing false-negatives is also very important as we do not want to misclassify a potential illicit drug abuser as someone who was clean.

For each technique, we trained four models through 4-fold cross-validation. To find the overall accuracy, we summed the correct predictions for each of the four cross-validation models and returned the 95% confidence interval for that technique. To find the overall confusion matrix, we added each of the four confusion matrices together and reported their true negative, true positive, false negative, and false-positive rates.

4.1 Logistic Regression

Our logistic regression model shows a good baseline with an accuracy of 0.8053. The standard data set outperformed the fully categorized data set by a statistically significant margin, implying that quantifying the data does improve our models.

	Standard	Fully Categorized
False Positives	16.46%	17.85%
False Negatives	24.21%	29.55%
Accuracy	0.8053 (+/- 0.0200)	0.7761 (+/- 0.0151)

4.2 Naive Bayes

For Naive Bayes models, BernoulliNB outclassed GaussianNB in both accuracy and minimizing false negatives. Most of our dataset's scalar parameters are quantified around 0, which makes those parameters a good fit for binarization. Coupled with having the majority of parameters be one-hot encoded makes BernoulliNB a better model.

	GaussianNB	BernoulliNB
False Positives	15.42%	21.49%
False Negatives	34.33%	17.78%
Accuracy	0.7724 (+/- 0.0270)	0.7995 (+/- 0.0143)

4.3 Support Vector Machines

For the SVM models, most kernels worked pretty well, with linear, rbf, and sigmoid kernels all showing roughly the same accuracy performance. Linear kernels had a small increase in false-negative rates, but that may simply be due to variation in the data. Polynomial kernels produced notably poor fits. Out of all the polynomial dimensions tested, only the 2-D kernel's accuracy came close. 3-D, 4-D, and 5-D kernels all performed very poorly, with accuracies of 0.6223, 0.6122, and 0.6122 respectively. Interestingly, 3-D, 4-D and 5-D kernels gravitated heavily towards predicting the respondent as negative, with 4-D and 5-D kernels always predicting negative, implying that the data set is not fit for a polynomial transformation.

	Linear	Rbf
False Positives	16.64%	17.85%
False Negatives	23.26%	21.89%
Accuracy	0.8080 (+/- 0.0228)	0.8058 (+/- 0.0238)
	Sigmoid	Poly 2
False Positives	17.24%	14.30%
False Negatives	21.61%	34.06%
Accuracy	0.8106 (+/- 0.0282)	0.7804 (+/- 0.0417)

4.4 Neural Networks

Unlike the other models, our neural networks did not return the same performance statistics every time we trained the model. This is most likely due to Keras' random initiation of starting node values, resulting in a different model each time. We were unable to get our neural networks to outperform our simpler models like Logistic Regression or Naive Bayes. In conclusion, due to our dataset's compatibility with traditional machine learning models and relatively small sample size, it is difficult for the neural network to outperform traditional models without overfitting.

	Basic	More Layers (tanh/ReLU)
False Positives	20.36%	20.19%
False Negatives	29.41%	31.19%
Accuracy	0.7613 (+/- 0.0432)	0.7554 (+/- 0.0324)
	More Layers (LeakyReLU)	
False Positives	19.76%	
False Negatives	29.00%	
Accuracy	0.7666 (+/- 0.0154)	

4.5 Cumulative Analysis

Most of our models were barely able to cross the 80% accuracy threshold. This accuracy cap may be due to the dataset's limited size. Our small training set may have contained a lot of variance that is not shown within the given parameters, which led most of our models to return similar performances. Our sigmoid-kernel SVM had the best accuracy, but not by a statistically significant amount. On the other hand, our BernoulliNB had slightly poorer accuracy, but had less variance within its models and kept false negative much lower.

5 Conclusions

From all the models we created, BernoulliNB and SVM-sigmoid performed the best in the two judging criteria: maximizing accuracy and minimizing false negatives.Both models were able to accurately predict an illicit drug abuser about 80% of the time, with an approximate 20% false negative rate.Our recommendation is for companies to use the BernoulliNB model as to minimize the false negative rate.It is better to err on the safe side and select a drug-free employee for drug testing than to let a potential illicit drug abuser dodge a drug test.

Our models could save companies from spending large amounts of resources on drug testing. Considering 11.2% of people age 12+ use illicit drugs in the United States,[4] running our model to decide who receives a drug test will cut drug test expenditures by 72% (Figure 7) while only missing 17.78% of illicit drug users. For smaller companies where fund delegation is very important, saving 72% on drug tests could free up a substantial part of their budget.

$$\begin{gathered}(\%IllicitUsers) \ast (\%Accuracy) \\ + \\ (\%Non-IllicitUsers) \ast (\%FalsePositives)\end{gathered}$$

Figure 7: Cutting Drug Test Expenditures

5.1 Future Work

In the future, we would like to improve our model further by increasing the accuracy and reducing false negatives even more. One plausible method of improving our model would be to collect more data and train our models on more enriching datasets.Another aspect we would like to test is the effect of cannabis legalization on the survey results.Currently, legal cannabis usage has not been approved nationwide, thereby influencing survey takers to lie and provide fabricated answers.We would like to see if the removal of the cannabis usage category from our set of predictors would provide more meaningful results in the future.

We look forward to improving our models to provide the greatest benefit possible for potential companies.

References

[1] Drug and Alcohol Testing Industry Association. Workplace drug testing. http://www.datia.org/component/content/article/27-industry-profile/931-workplace-drug-testing.html.

[2] Connecticut Clearinghouse. Drug testing. https://www.ctclearinghouse.org/topics/drug-testing.

[3] E. Fehrman, V. Egan, and E. Mirkes. Drug consumption (quantified) data set. https://archive.ics.uci.edu/ml/datasets/Drug+consumption+\%28quantified\%29, 2016.

[4] Centers for Disease Control and Prevention. Illicit drug use. https://www.cdc.gov/nchs/fastats/drug-use-illicit.htm, 2017.