Azat Dovgeldiyev

DSC 478

Final report

Drug consumption

Drug consumption and addiction are among the largest and most challenging problems facing society today. We now know that drug abuse is preventable behavior, and that drug addiction is, fundamentally, a treatable chronic.

Although we know the magnitude of the problem, what causes a person to drug in the first place? Elaine Fehrman, Vincent Egan, and Evgeny M. Mirkes viewed this question from a psychological standpoint and collected data from 1885 respondents from over seven countries to evaluate how personality traits and demographics affect drug consumption.

In this data analysis task, I use their data downloaded from Machine Learning Repository[1] to evaluate and analyze the risk of drug consumption for hard drugs[2] by utilizing :

- Data cleaning
- Data visualization
- Modeling
- Dimensionality reduction

Database contains records for 1885 respondents, and for each respondent 12 attributes are known. All input attributes are originally categorical and are quantified. After quantification values of all input features can be considered as real-valued. There are no missing values in dataset, so cleaning process involved to label variable names into attributes, deleting unnecessary columns and define hard drug variable. There seem to be a normal distribution among personality scores except sensation seeing and impulsive. The common drug types used among the respondents were Alcohol, Caffeine and Choc (Chocolate), the contribution of nicotine and cannabis are almost the same as they are legal in some states, countries. Since the hard drugs are the most dangerous and illegal, the percentage of users did not exceed 40 %. There are more male drug consumers than female, and more users in age between 18-24. The relationship among scores show that no multicollinearity in the dataset. The chance of leaving school using drugs are more likely than having a degree in college.

I started to build my model with performing Linear Regression Analysis on data using the closed form solution implementation. I split the data into training and testing sets with 80-20 partition and normalized so all the attributes are in the same scale.

$$v' = [(v - \min)/(\max - \min)] * (newmax - newmin) + newmin$$

Independent variables included demographic information of respondents and scores, while target variable included whether a person uses hard drugs. Root mean squared error on training data I got between 0.1 and 0.5.

```
rmse_train = np.sqrt(total_error/len(yHat))
print("RMSE on Training data: ",round(rmse_train,3))
```

```
RMSE on Training data:  0.393
```

Below displayed attributes are the regression coefficients and tell us whether there are positive or negative correlation between each predictor the target. Ethnicities have the highest correlation with hard drug user.

```
for i in range(len(X_train.columns)):
    print("%7s   %2.2f" % (X_train.columns[i], w[i]))
```

```
    Age    -0.16
 Gender    -0.11
Education    -0.13
Country    -0.22
Ethnicity    0.94
 Nscore    0.46
 Escore    0.08
 Oscore    0.45
 Ascore    0.07
 Cscore    -0.17
Impulsive    0.08
     SS    0.36
```

Then I performed 10-fold cross-validation and compare the cross-validation RMSE to the training RMSE, and the error for 10-fold I received almost the same values with error rate on training.

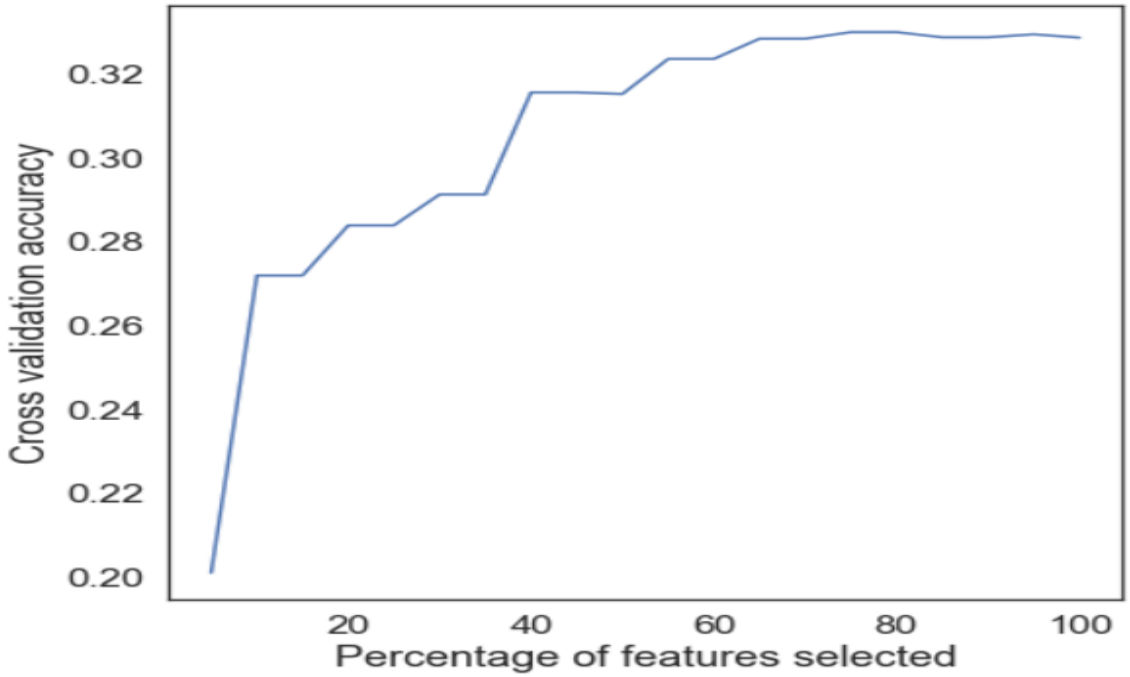
```
Method: Simple Linear Regression
RMSE on training: 0.3934
RMSE on 10-fold CV: 0.3961
```

Using the package from sckit-learn with subset of features I continued with Linear Regression. For feature selection, the script written that takes as an input the training data,

target variable, the model and other parameters to return the optimal percentage of the most informative features to use, moreover 5-fold cross validation picked on the training data.

```
Optimal percentile of features:75

Optimal number of features:9
```



The list of most informative variables and their weights shown below:

1	Country	386.328	
2	SS	302.345	
3	Age	175.251	
4	Oscore	169.004	
5	Cscore	148.373	
6	Impulsive		143.466
7	Gender	112.942	

Error we get on testing set using Linear regression is 0.386, which is slightly less than training and acceptable.

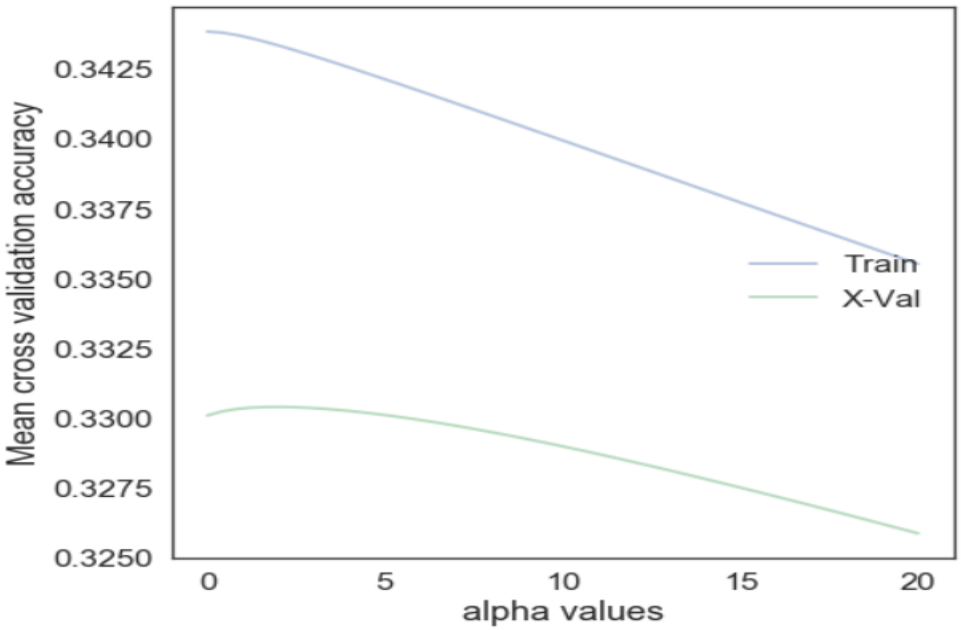
We continue our analysis with Ridge Regression and Lasso Regression using the modules from sklearn package. In each case, we perform systematic model selection to identify the optimal alpha parameter. The function that was created took as input data and target variable.

The model selection process performed 5-fold cross validation. Finally, using the best alpha values, we trained the model on the full training data and evaluated it on the set-aside test data (Full calculation of function can be found in Jupyter notebook).

```
alpha = np.linspace(.01, 20, 50)
ridge = Ridge(alpha = alpha)

train_scores, test_scores = calc_params(X_train_norm, y_train, ridge, alpha, 'alpha',5)
```

Average cross validation accuracy plot on each alpha value based on Ridge regression:



Parameters used for Ridge regression:

```
#Linear regression with Ridge coef
ridge = Ridge(alpha=alpha[2])

ridge.fit(X_train_norm, y_train)
```

```
Ridge(alpha=0.8259183673469387, copy_X=True, fit_intercept=True, max_iter=None,
      normalize=False, random_state=None, solver='auto', tol=0.001)
```

Root mean square error revealed 0.3861 using Ridge regression on testing data.

Parameters used for picking alpha values and finding RMSE with Lasso Regression:

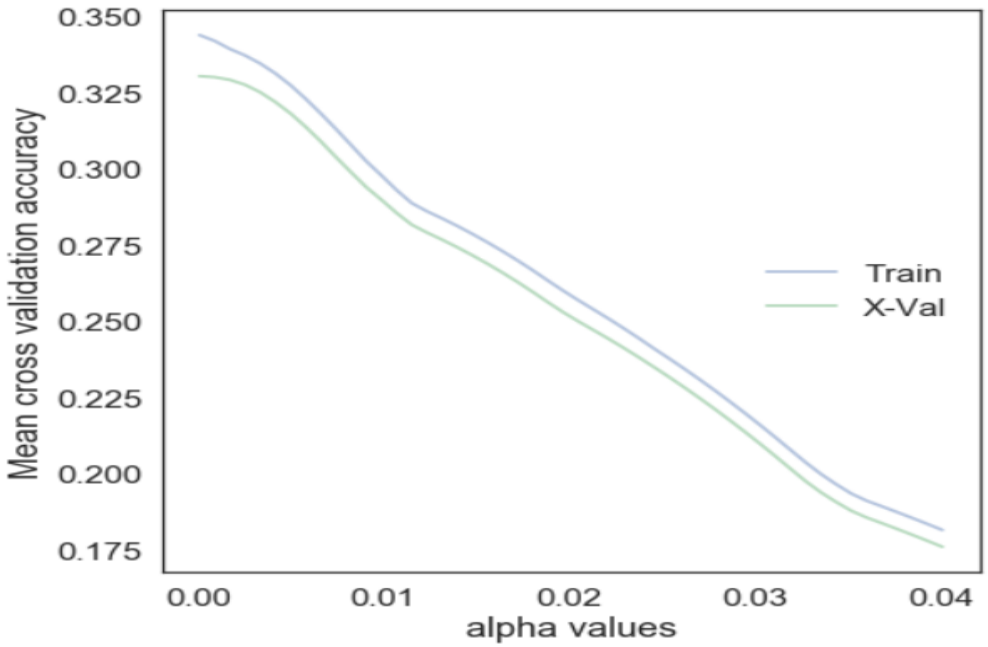
```
#perform Lasso Regression
alpha = np.linspace(.0001,.04,50)
lasso = Lasso(alpha=alpha)

train_scores, test_scores = calc_params(X_train_norm, y_train, lasso, alpha, 'alpha', 5)
```

```
lasso = Lasso(alpha = test_scores[0])
lasso.fit(X_train_norm, y_train)
```

```
Lasso(alpha=0.33033758096092986, copy_X=True, fit_intercept=True, max_iter=1000,
      normalize=False, positive=False, precompute=False, random_state=None,
      selection='cyclic', tol=0.0001, warm_start=False)
```

Average cross validation accuracy plot on each alpha value based on Ridge regression:



Root mean square error revealed 0.4817 using Lasso regression on testing data. Comparing to Ridge regression and Linear regression Lasso performed the worst in minimizing the root-mean-square-error.

Next, we perform regression using Stochastic Gradient Descent for Regression calculating the derivative from each training data instance and calculating the update immediately. For this part, we use the SGDRegressor module from sklearn package. (Full calculation of codes can be found in Notebook).

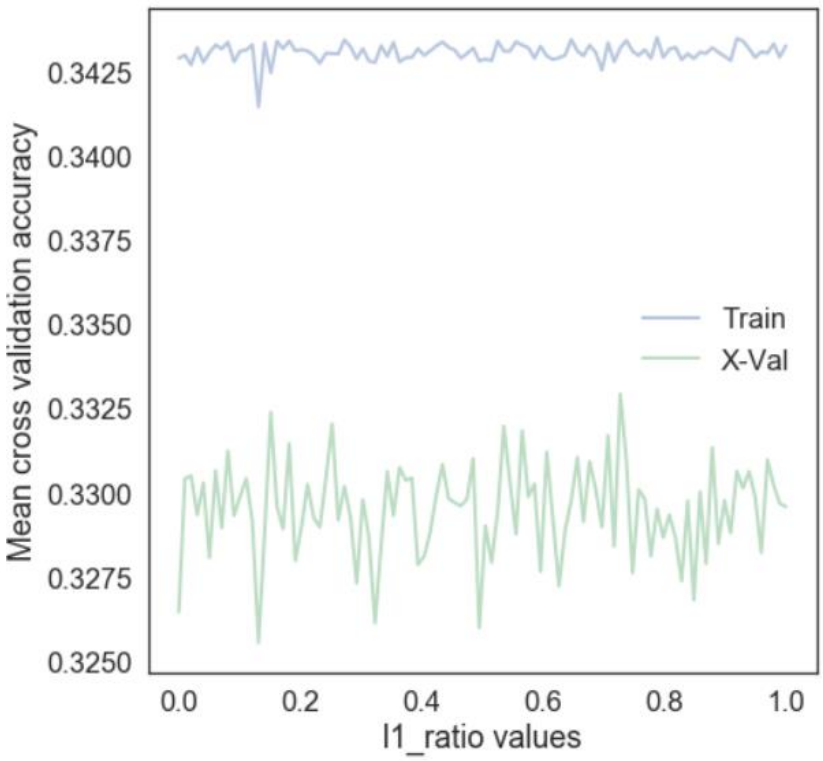
```
grid_search.fit(X_train_sgd, y_train)
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 500 out of 500 | elapsed:    3.0s finished
```

```
GridSearchCV(cv=5, error_score=nan,
             estimator=SGDRegressor(alpha=0.0001, average=False,
                                    early_stopping=False, epsilon=0.1,
                                    eta0=0.01, fit_intercept=True,
                                    l1_ratio=0.15, learning_rate='invscaling',
                                    loss='squared_loss', max_iter=1000,
                                    n_iter_no_change=5, penalty='l2',
                                    power_t=0.25, random_state=None,
                                    shuffle=True, tol=0.001,
                                    validation_fraction=0.1, verbose=0,
                                    warm_star...
       1.30646939e+01, 1.34726531e+01, 1.38806122e+01, 1.42885714e+01,
       1.46965306e+01, 1.51044898e+01, 1.55124490e+01, 1.59204082e+01,
       1.63283673e+01, 1.67363265e+01, 1.71442857e+01, 1.75522449e+01,
       1.79602041e+01, 1.83681633e+01, 1.87761224e+01, 1.91840816e+01,
       1.95920408e+01, 2.00000000e+01]),
                         'penalty': ['l2', 'l1']},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=1)
```

Average cross validation accuracy plot on each alpha value based on Elastic net with L1 parameter:

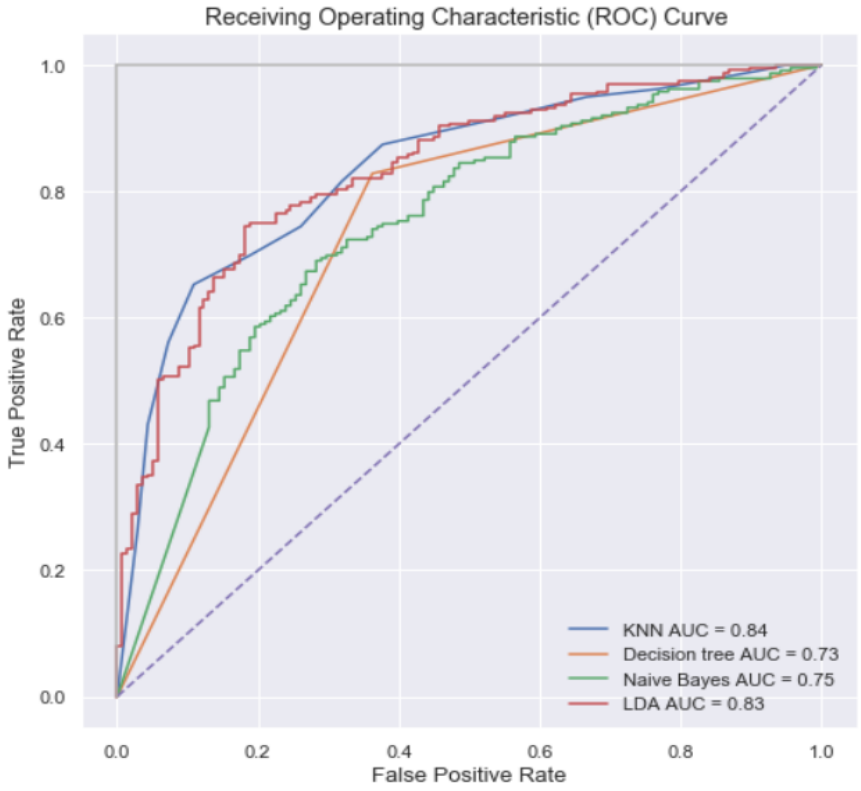


Root mean square error revealed 0.3875 using SGD regression on testing data, almost equal to Ridge and Linear Regression. Elastic Net included both L-1 and L-2 norm regularization

terms. This gave us the benefits of both Lasso and Ridge regression. It was found to have predictive power better than Lasso, while still performing feature selection.

After performing different types of Linear Regression models to minimize the errors we continued in building models using KNN, Decision trees, Naïve Bayes and Linear Discriminant Analysis. Models used normalized dataset, for KNN it was found that 13 neighbors fit better than others, accuracy on KNN test set shows 77%, still needed improvement. Accuracy with Decision trees is worse than KNN with 75.9%. In my opinion the shape of initial dataset with smaller observations is the reason that decision trees did not perform well. Naïve Bayes 78.8% and Linear Discriminant Analysis revealed 79.04% accuracy on testing set. I plotted receiver operating characteristic curve (ROC curve[3]) showing the performance of a classification model at all classification thresholds.

```
roc_auc_score for KNN: 0.84
roc_auc_score for Decision tree: 0.73
roc_auc_score for Naive Bayes: 0.75
roc_auc_score for LDA: 0.83
```



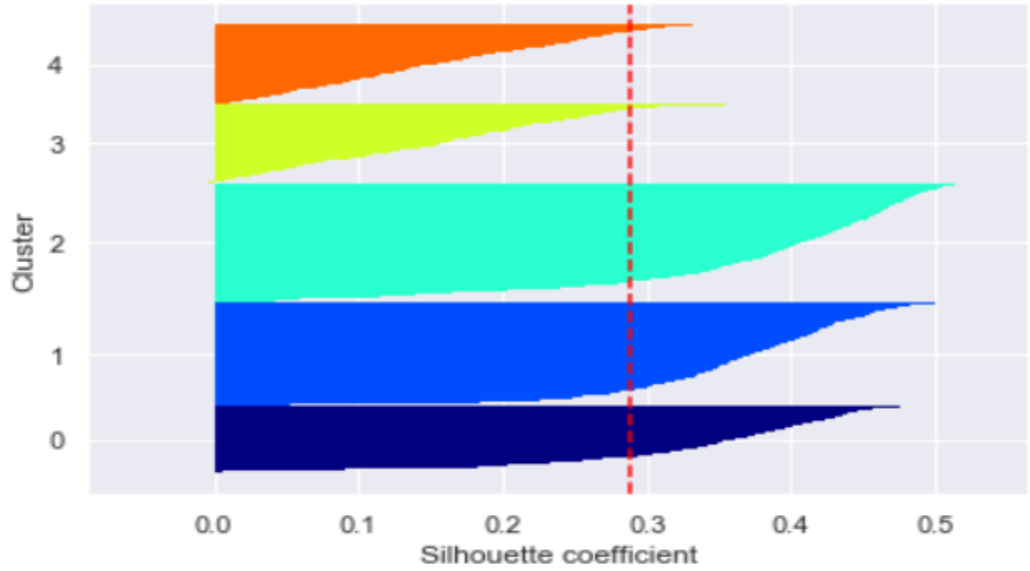
Receiving operating curves show the trend between the models we built, ROC is higher for KNN (13) and LDA than Decision trees and Naive bayes. KNN and LDA show the highest results for test set and can expect for around 84 % that one would be a hard drug user by using demographic and personal test scores.

Finally, we use dimensionality reduction implementing Principal Component Analysis (PCA[4]) to find the most important components in determining hard drug user. First, I used KMeans implementation performing clustering on the data (I changed then to K=5).

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=1000,
       n_clusters=7, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

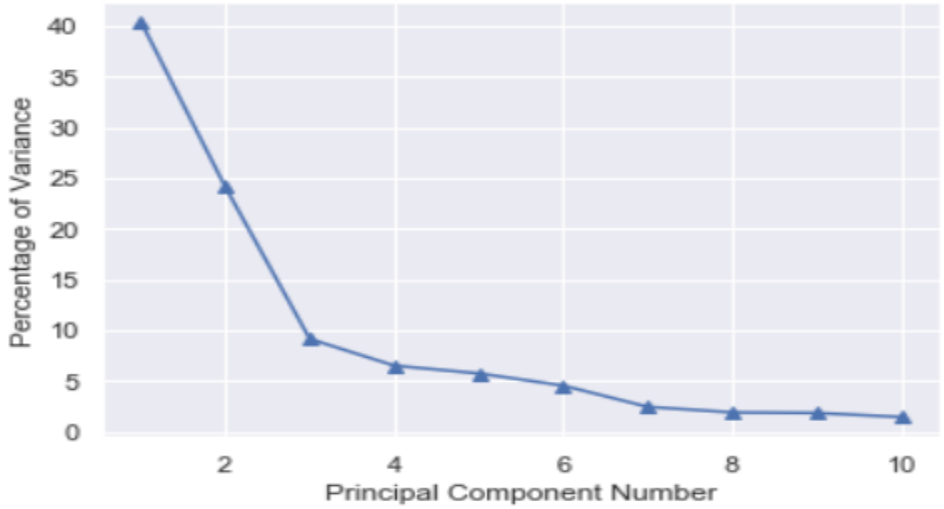
The silhouette mean for 5 clusters is 0.2887, The Homogeneity score that all the observations with the same class label are in the same cluster is 0.22, and Completeness that all members of the same class are in the same cluster is 0.94. We perform Kmeans again, but on the lower dimensional transformed data, however average scores of silhouettes are lower on original than reduced.

```
The Completeness score:  0.0941507231665934
The Homogeneity score:  0.22865824423461512
```



Next, we analyze the principal components to determine the number, r, of PCs needed to capture at least 95% of variance in the data. we provide a plot of PC variances. Then use these r components as features to transform the data into a reduced dimension space. Therefore, 9 components were selected. (Full calculation can be found in Jupyter Notebook).

PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the Scree plot below. Organizing information in principal components this way allows to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as new variables.



I used different paths in visualizing the data to show the distribution of attributes such as correlation, then I came up with different modeling approaches using supervised and unsupervised techniques with different parameters in predicting the respondents whether using hard drugs with given personality scores and demographic information. Linear Regression used to find the relationship between independent and dependent variables. Lasso Regression, Ridge Regression, and Elastic Net Regression were performed with an examination of coefficients followed by a comparison of model performance metrics RMSE. Linear Discriminant analysis, Naïve bayes gave the highest scores overall in predicting if a person uses the hard drugs with just less than 80%. We could improve our accuracy if the initial data had more observations. Lastly, we checked for dimensionality reduction, calculating important

components, and finding the scores for homogeneity and completeness scores. All mentioned techniques were learned from the Machine Learning Class.

The future works can be done with transforming binary classification by union of part of classes into one new class. For example, "Never Used", "Used over a Decade Ago" form class "Non-user" and all other classes form class "User".

Index

1. Data set downloaded from https://archive.ics.uci.edu/ml/datasets/Drug+consumption+%28quantified%29
2. Definition of hard drugs https://www.verywellmind.com/the-difference-between-soft-drugs-and-hard-drugs-22308
3. ROC curves https://medium.com/@lily_su/confusion-matrix-roc-auc-and-imbalanced-classes-in-logistic-regression-5c7ead3deefc
4. More on PCA https://builtin.com/data-science/step-step-explanation-principal-component-analysis