Python for Data Analysis

Project 2022 - Marie Betbeder - Emeric Buttin

Summary

Part 1 :
The features

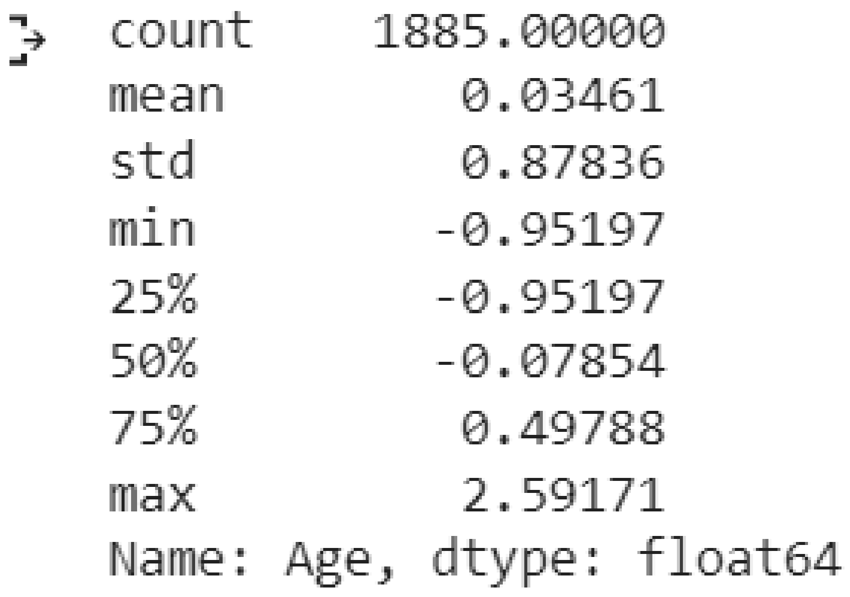
Part 3 :
The target (Binary class)

Part 2 :
The target (Non binary class)

Part 4 :
Conclusion

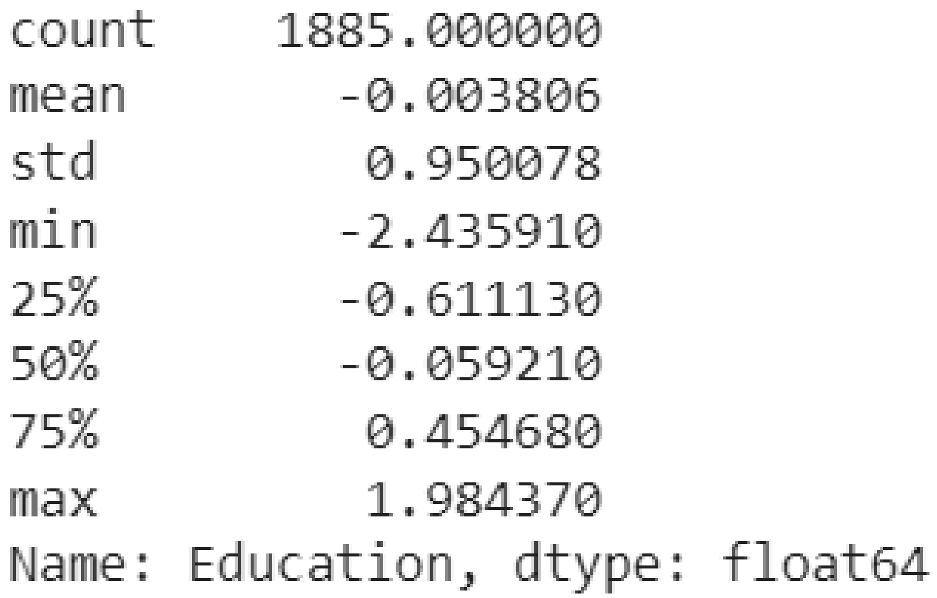
13 DIFFERENT FEATURES STANDARDIZED

Age :



```
count    1885.00000
mean        0.03461
std         0.87836
min        -0.95197
25%        -0.95197
50%        -0.07854
75%         0.49788
max         2.59171
Name: Age, dtype: float64
```

Education :

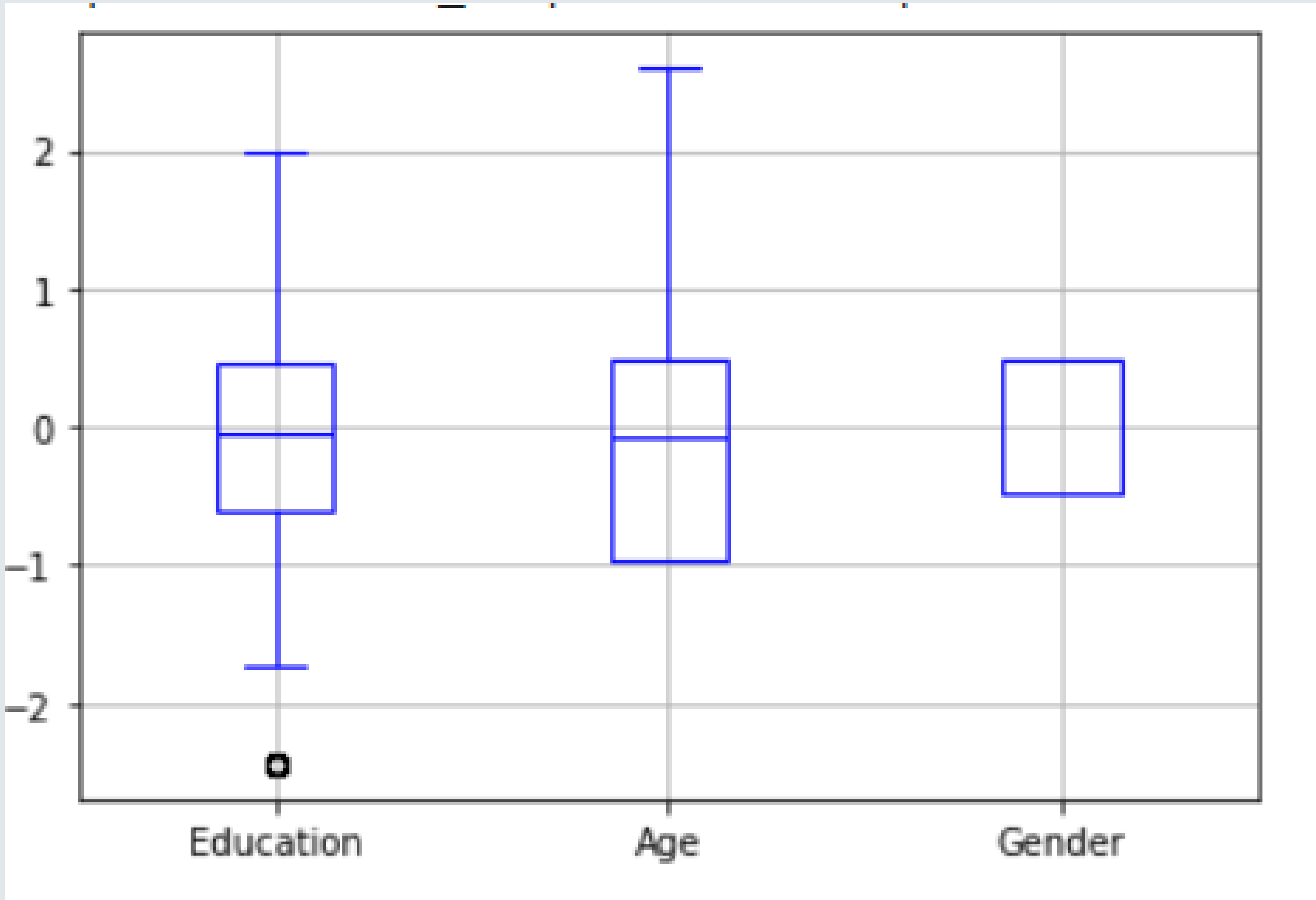


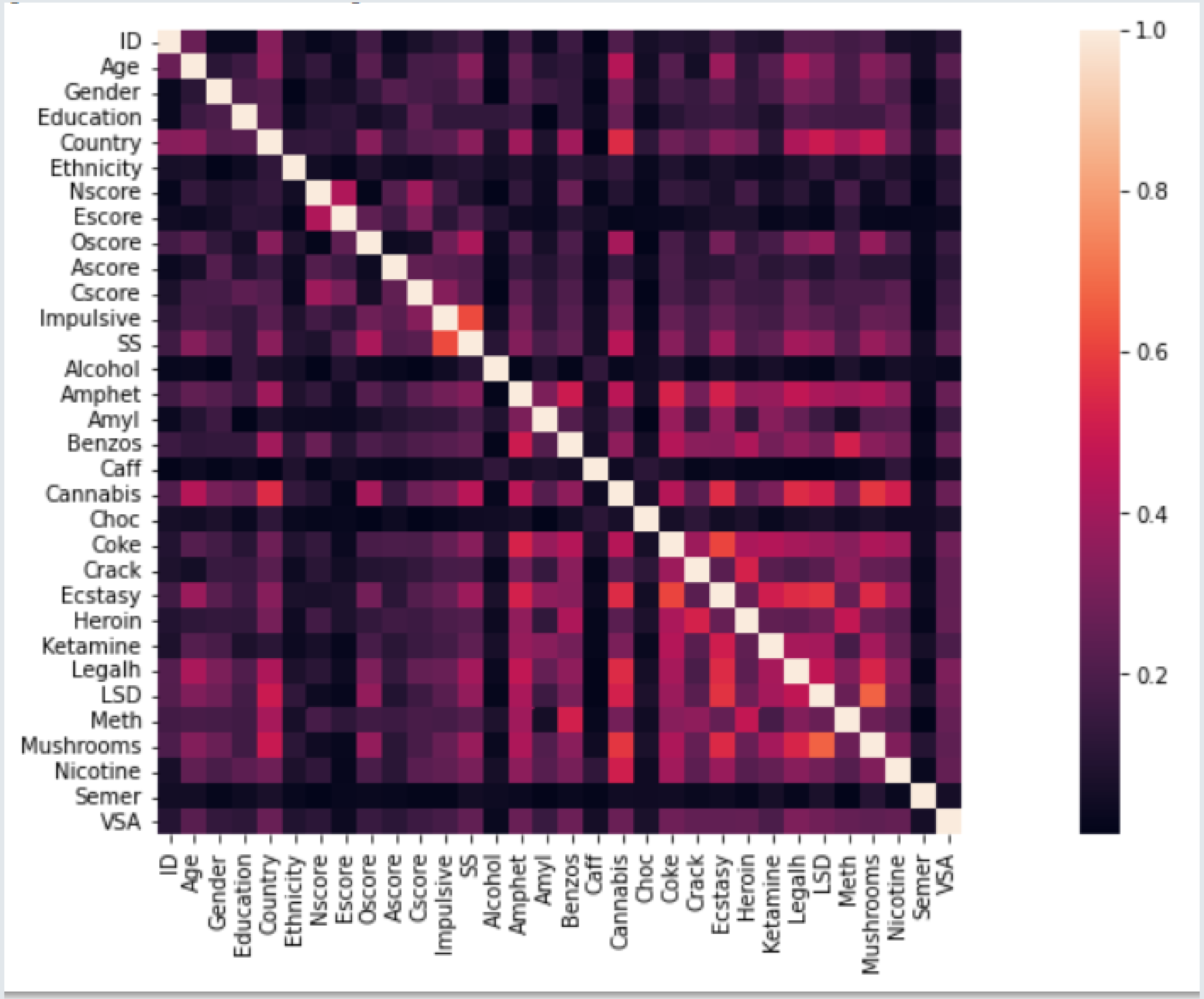
```
count    1885.000000
mean       -0.003806
std         0.950078
min        -2.435910
25%        -0.611130
50%        -0.059210
75%         0.454680
max         1.984370
Name: Education, dtype: float64
```

Part 1 :
The features

All of the features have been standardized and no value is missing, there isn't much data cleaning needed

AN INSIGHTFUL SAMPLE



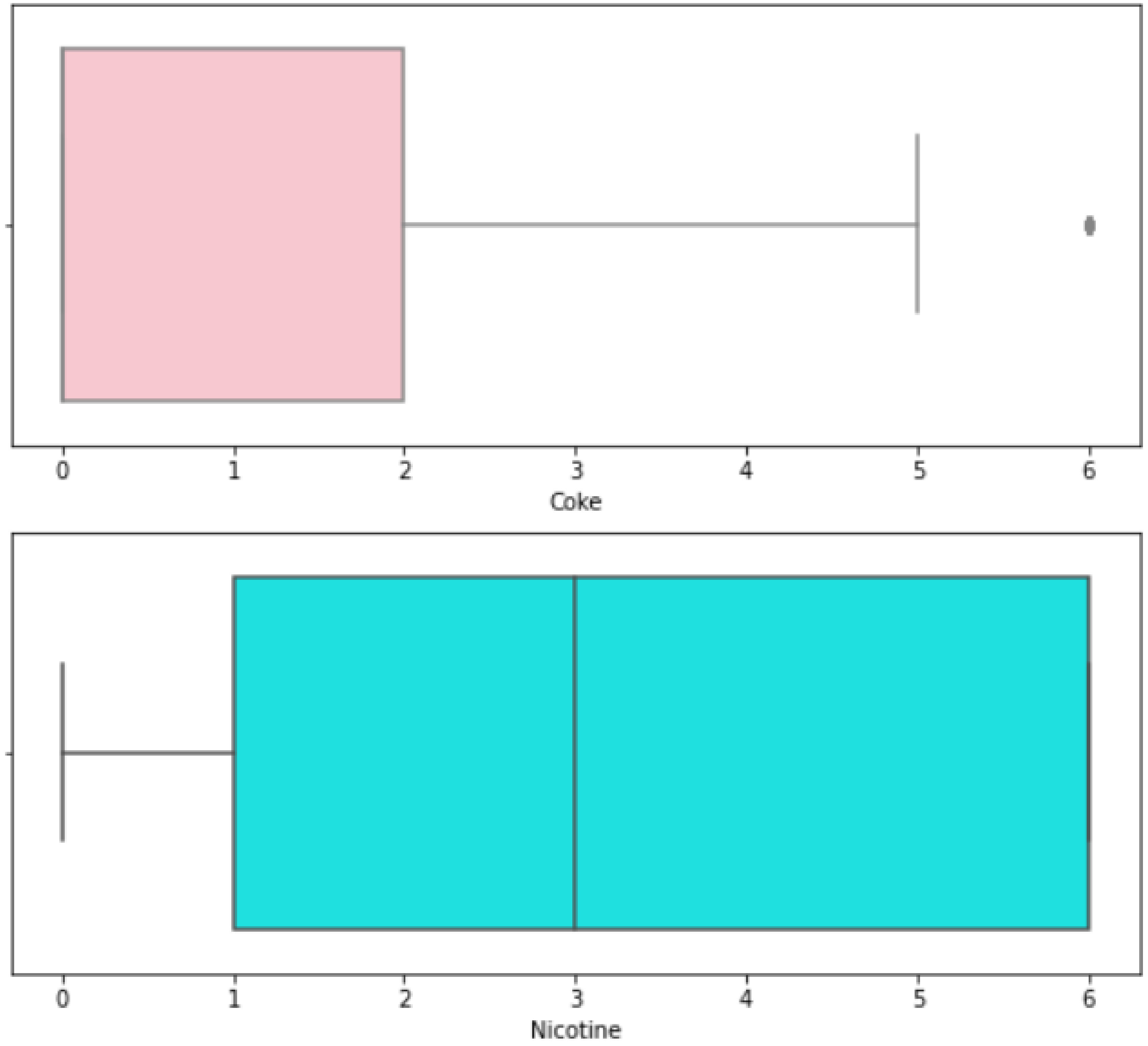


CORELATION MATRIX

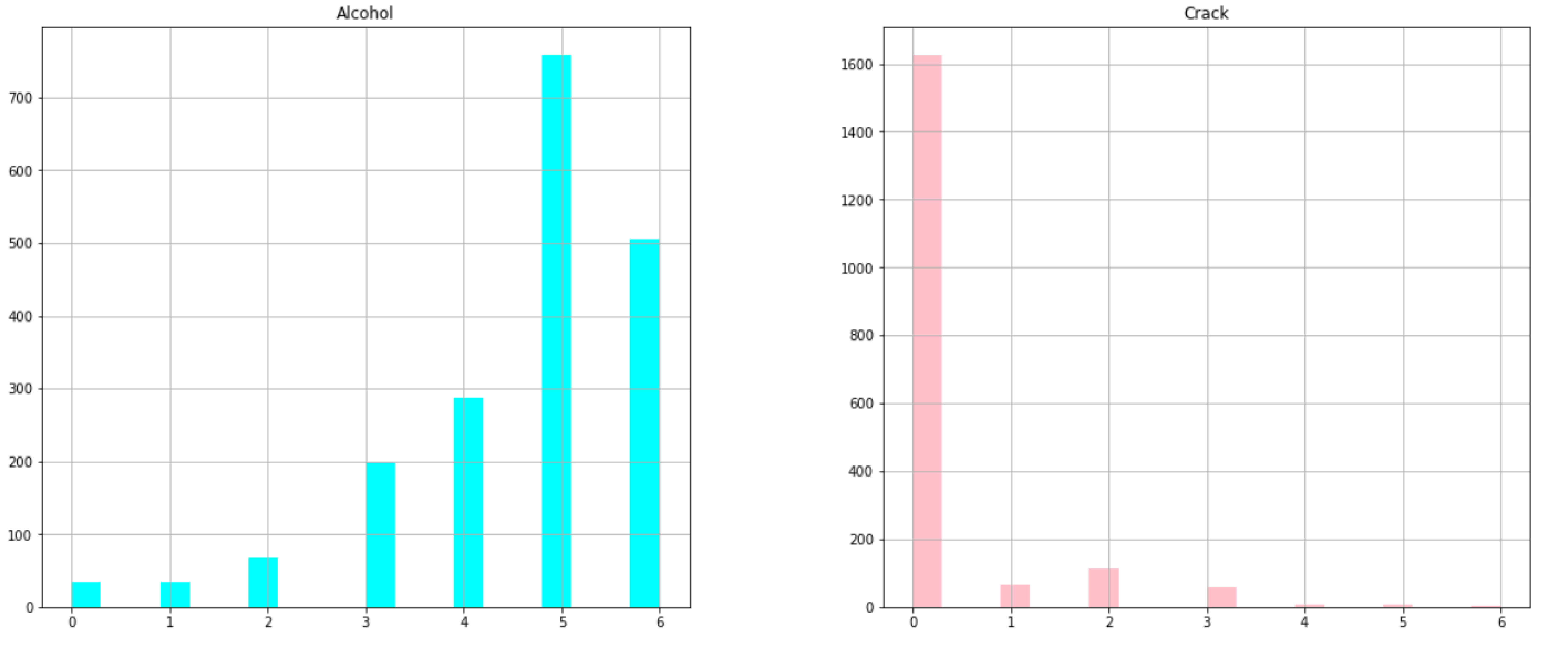
We check here the correlations between the features and targets in order to see which one are the most valuable but we decided to keep them all anyway, as our dataset is small and doesn't take much time to run

THE POSSIBLE CLASSES FOR EACH DRUGS

- 0 : Never used
- 1 : Used over a decade ago
- 2 : Used in last decade
- 3 : Used in last year
- 4 : Used in last month
- 5 : Used in last week
- 6 : Used in last day



EACH DRUG IS DIFFERENT



Classification and Regression Models

3

Regression Models :

- Linear Regression
- Logistic Regression
- Random Forest Regression

4

Classification Models :

- Decision Tree
- Random Forest Classifier
- KNN Classifier
- Boosting Classifier

Part 2 :

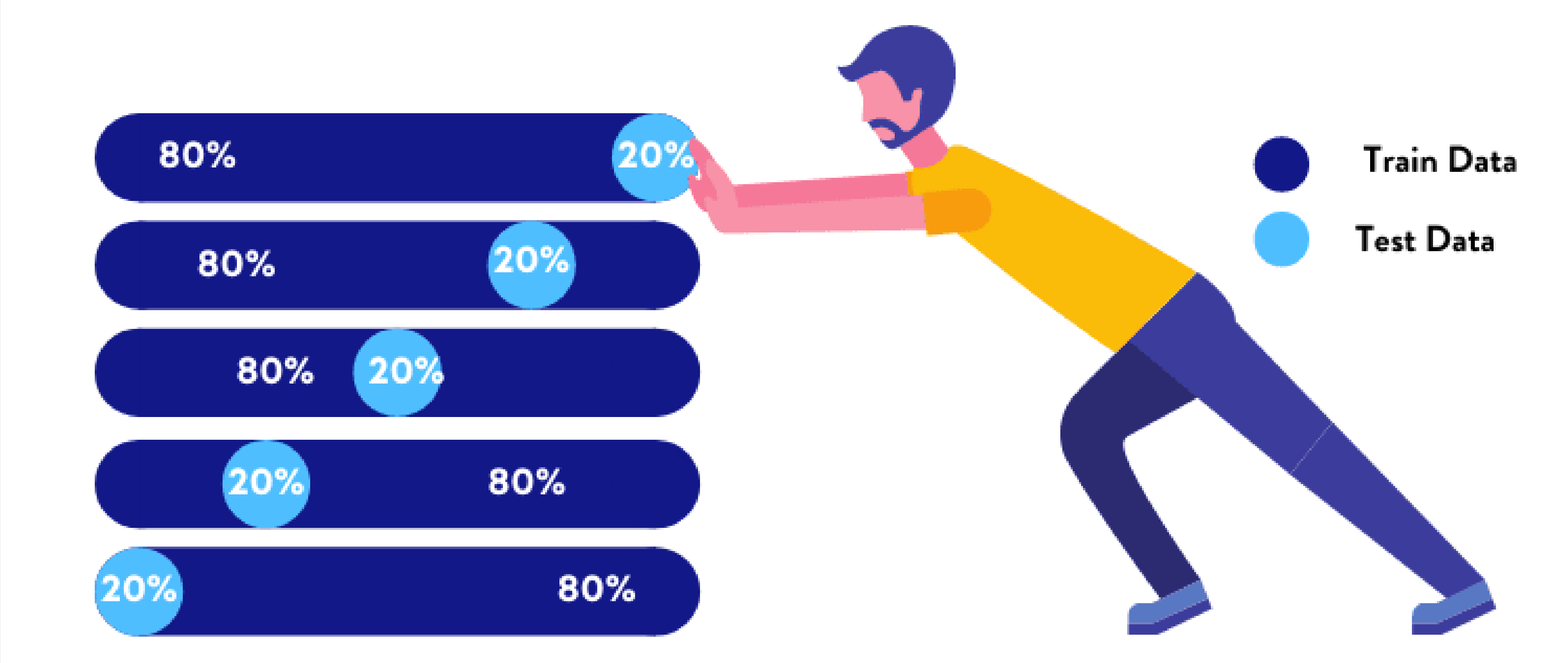
Models for Non Binary class of drug use

We tried a lot of different models from the library scikit learn and compared the results. As there was a lot of different targets, we first compared for one (cannabis) and then used the best model to predict for all the other targets.

METHODS TO IMPROVE THE MODELS

Cross validation

Randomized grid search



To improve the quality of the predictions, we did a cross validation and a grid search (depending on the model it could be a randomized grid search). It allowed us to tune the models using the best hyperparameters possible for our dataset.

Classification and Regression Models

2

Regression Models :

- Linear Regression
- Logistic Regression

3

Classification Models :

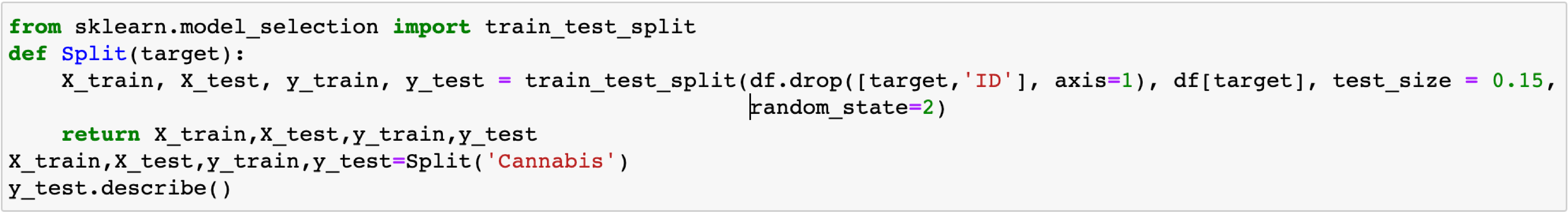
- Decision Tree
- Random Forest Classifier
- KNN Classifier

Part 3 :

Models for Binary class of drug use

When we saw the results from the models, we weren't very satisfied with the scores so we tried a different approche. We decided to use only two classes user and not a user. Not a user is a combination of 'Never used' and 'used over a decade ago'. User is a combination of all the rest. That way, our model became much better.

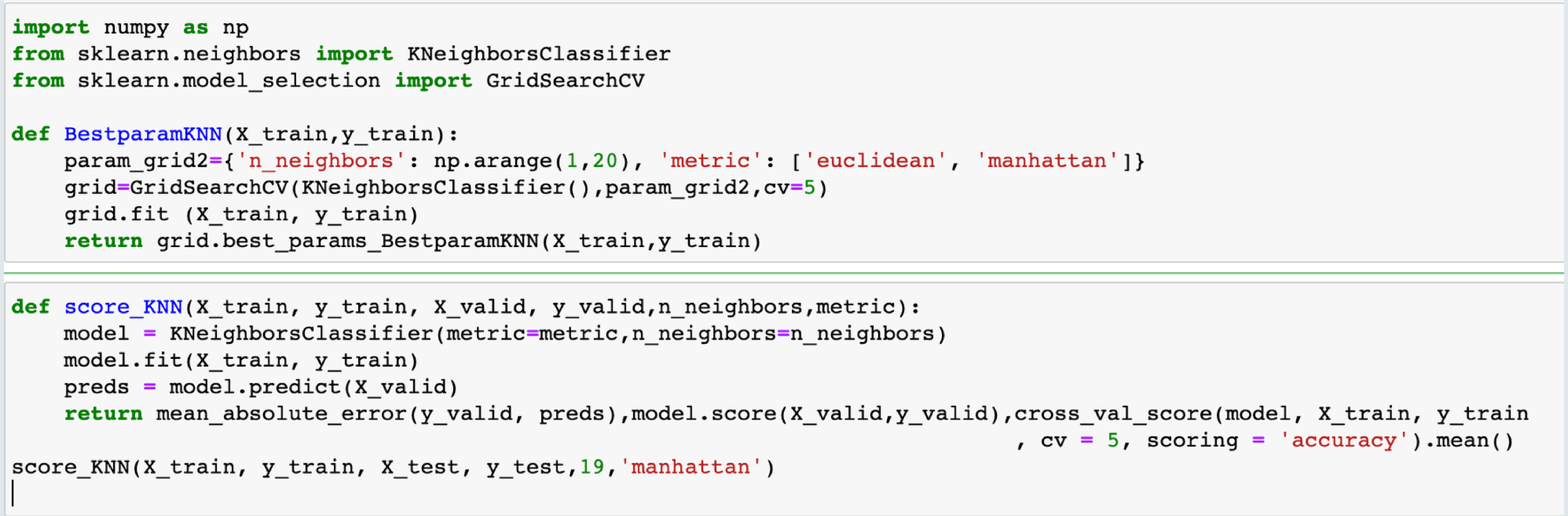
Splitting the data



```python
from sklearn.model_selection import train_test_split
def Split(target):
    X_train, X_test, y_train, y_test = train_test_split(df.drop([target,'ID'], axis=1), df[target], test_size = 0.15,
                                                        random_state=2)
    return X_train,X_test,y_train,y_test
X_train,X_test,y_train,y_test=Split('Cannabis')
y_test.describe()
```

This function takes a dataset and a target as parameters, and returns our data splitted in a test set and a training set. This is particularly usefull for us as we have a lot of different targets in our dataset.

Example of a model with cross validation



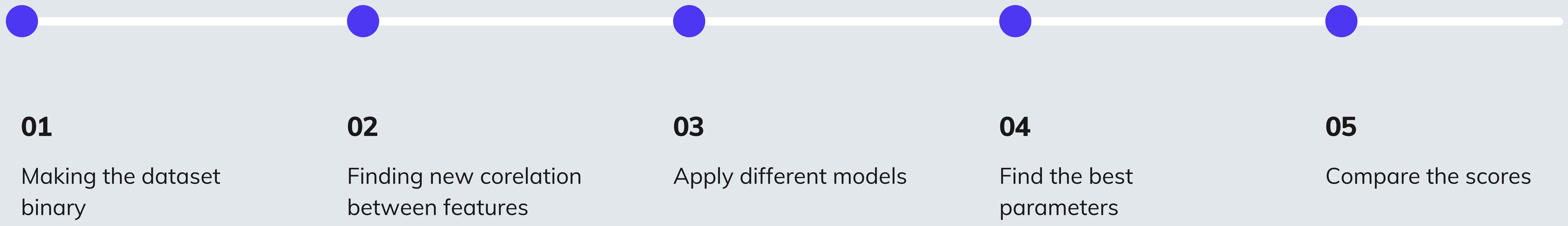
```
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV

def BestparamKNN(X_train,y_train):
    param_grid2={'n_neighbors': np.arange(1,20), 'metric': ['euclidean', 'manhattan']}
    grid=GridSearchCV(KNeighborsClassifier(),param_grid2,cv=5)
    grid.fit (X_train, y_train)
    return grid.best_params_BestparamKNN(X_train,y_train)
```

```
def score_KNN(X_train, y_train, X_valid, y_valid,n_neighbors,metric):
    model = KNeighborsClassifier(metric=metric,n_neighbors=n_neighbors)
    model.fit(X_train, y_train)
    preds = model.predict(X_valid)
    return mean_absolute_error(y_valid, preds),model.score(X_valid,y_valid),cross_val_score(model, X_train, y_train
                                                                              , cv = 5, scoring = 'accuracy').mean()
score_KNN(X_train, y_train, X_test, y_test,19,'manhattan')
```

The first function is used to find the best parameters for the KNN model. We define a grid in which each hyperparameter as different values that we want to test. The gridsearchCV of scikit learn does a cross validation fit for all the different combination of hyperparameters and return the best one. The second function takes for parameters the training and test datasets and the hyperparameters of the KNN. We then create the model with said hyperparameters and then fit, predict and return the

Binary class of drug use

We obtain better predict results than with non binary class of drug use



01

Making the dataset binary

02

Finding new corelation between features

03

Apply different models

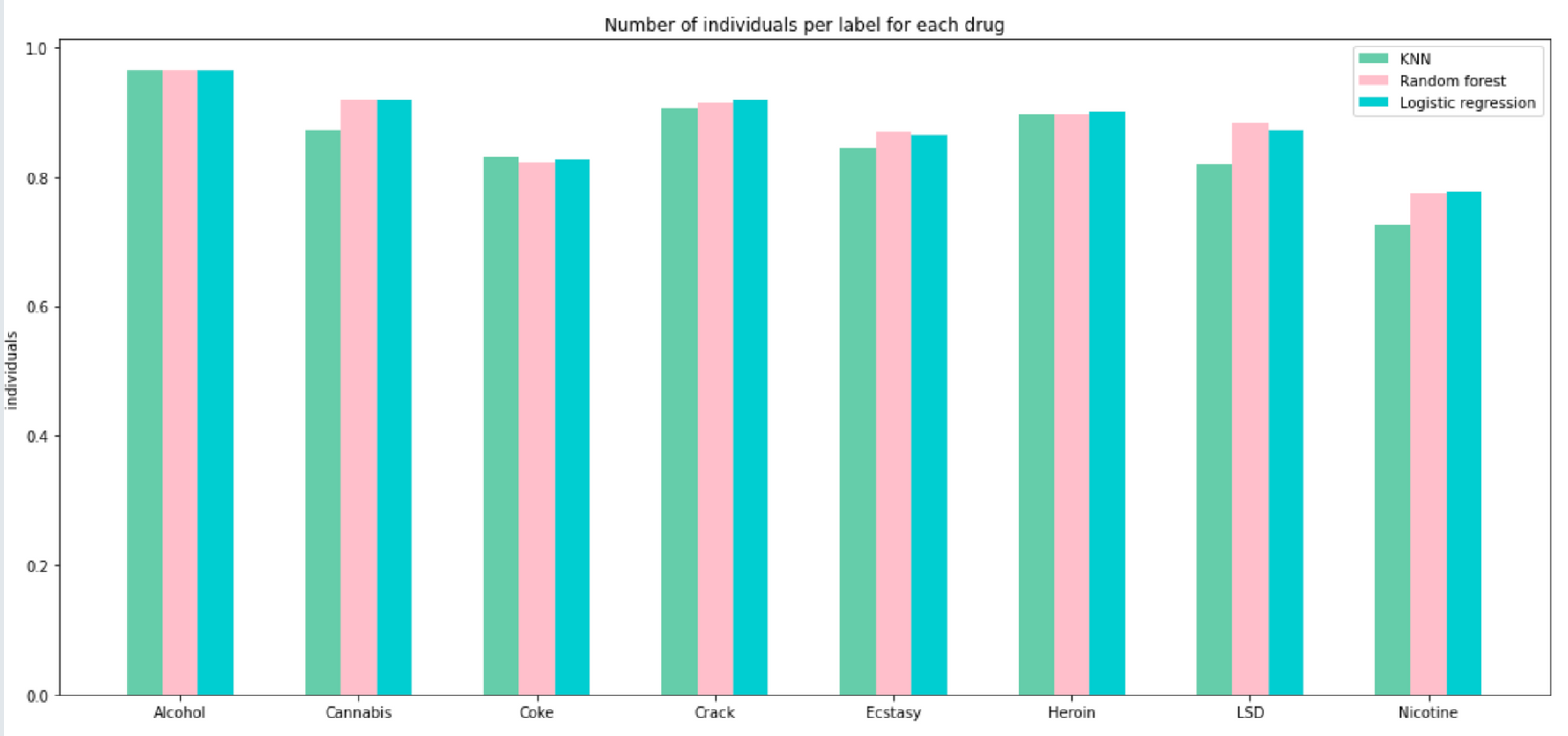
04

Find the best parameters

05

Compare the scores

We did here the same processing than with non binary classes and compared the scores as well, using visuals.



There isn't all the drugs here for readability's sake, but we can see that the models are very close in terms of scores, the logistic regression being almost always best and the KNN being usually the worst.

Flask API

The last part of our project was to create a flask API, that would ask the user for the information of one person, and would return wether or not that person is a drug user. For simplicity matter, we decided to use only cannabis as target and all the other drugs become features to help train our model. There is two pages on the API, one where the user can enter all the informations about the person, and when he submits it, the second page, where the result is displayed.

Conclusion

What are our predictions ?

- Seven class classifications for each drug separately.
- Evaluation of risk to be drug consumer for each drug.
- The best binarization of classes for each attribute.

During this project we went through different phases, always with the goal of having the best scores for our models. Thanks to scikit learn we tried a lot of models and would compare them. This library also allowed us to prefect our models thanks to cross validation and grid search for the tuning. We then tried to modify our dataset, by making the classes binary in order to have prediction over 80%, and we were successful. There are still other method we could try, like deep learning algorithms, but scikit learn isn't the best library for this. We could try tensor flow for example.

Thank you !