Python for Data Analyst - Final Project

Drug Consumption

Summary

1– Explanation of the **Dataset**

2 – Data cleaning

3 – Visualizations

4 – Modeling

1 – Explanation of the Dataset

Explication of the Dataset

The dataset that we used for our Final Project is about Drug Consumption. It was made from a survey where the participants were judged on 12 attributes such are their ages, the country there are from, or more complex attributes like Personality measurements such as being open minded or aggressive.

The participants were also asked about the consumption of drugs like Cannabis, Nicotine, Cocaine, LSD or Caffeine. They had to say when was the last time they consumed it, and then 7 categories were made from "Never used" to "Used over the last day".

The dataset had 1885 respondents, so 1885 lines.

Here is a picture of how is it looking after we imported it.

You can notice that there are no columns names, and the values are not readable for a human.

	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28	29	30	31
0	1	0.49788	0.48246	-0.05921	0.96082	0.12600	0.31287	-0.57545	-0.58331	-0.91699	...	CL0	CL0	CL0	CL0	CL0	CL0	CL0	CL2	CL0	CL0
1	2	-0.07854	-0.48246	1.98437	0.96082	-0.31685	-0.67825	1.93886	1.43533	0.76096	...	CL4	CL0	CL2	CL0	CL2	CL3	CL0	CL4	CL0	CL0
2	3	0.49788	-0.48246	-0.05921	0.96082	-0.31685	-0.46725	0.80523	-0.84732	-1.62090	...	CL0	CL0	CL0	CL0	CL0	CL0	CL1	CL0	CL0	CL0
3	4	-0.95197	0.48246	1.16365	0.96082	-0.31685	-0.14882	-0.80615	-0.01928	0.59042	...	CL0	CL0	CL2	CL0	CL0	CL0	CL0	CL2	CL0	CL0
4	5	0.49788	0.48246	1.98437	0.96082	-0.31685	0.73545	-1.63340	-0.45174	-0.30172	...	CL1	CL0	CL0	CL1	CL0	CL0	CL2	CL2	CL0	CL0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1880	1884	-0.95197	0.48246	-0.61113	-0.57009	-0.31685	-1.19430	1.74091	1.88511	0.76096	...	CL0	CL0	CL0	CL3	CL3	CL0	CL0	CL0	CL0	CL5
1881	1885	-0.95197	-0.48246	-0.61113	-0.57009	-0.31685	-0.24649	1.74091	0.58331	0.76096	...	CL2	CL0	CL0	CL3	CL5	CL4	CL4	CL5	CL0	CL0
1882	1886	-0.07854	0.48246	0.45468	-0.57009	-0.31685	1.13281	-1.37639	-1.27553	-1.77200	...	CL4	CL0	CL2	CL0	CL2	CL0	CL2	CL6	CL0	CL0
1883	1887	-0.95197	0.48246	-0.61113	-0.57009	-0.31685	0.91093	-1.92173	0.29338	-1.62090	...	CL3	CL0	CL0	CL3	CL3	CL0	CL3	CL4	CL0	CL0
1884	1888	-0.95197	-0.48246	-0.61113	0.21128	-0.31685	-0.46725	2.12700	1.65653	1.11406	...	CL3	CL0	CL0	CL3	CL3	CL0	CL3	CL6	CL0	CL2

1885 rows × 32 columns

2– Data Cleaning

So, on the website , one part was the explanation of all the values.

For example, you can see that Gender can take the value 0.48246 or -0.48246 , which means Female or Male

Value	Meaning	Cases	Fraction
-0.95197	18-24	643	34.11%
-0.07854	25-34	481	25.52%
0.49788	35-44	356	18.89%
1.09449	45-54	294	15.60%
1.82213	55-64	93	4.93%
2.59171	65+	18	0.95%

Descriptive statistics

Min	Max	Mean	Std.dev.
-0.95197	2.59171	0.03461	0.87813

3. Gender (Real) is gender of participant:

Value	Meaning	Cases	Fraction
0.48246	Female	942	49.97%
-0.48246	Male	943	50.03%

Descriptive statistics

Min	Max	Mean	Std.dev.
-0.48246	0.48246	-0.00026	0.48246

4. Education (Real) is level of education of participant and has one of the values:

Value	Meaning	Cases	Fraction
-2.43591	Left school before 16 years	28	1.49%
-1.73790	Left school at 16 years	99	5.25%
-1.43719	Left school at 17 years	30	1.59%
-1.22751	Left school at 18 years	100	5.31%
-0.61113	Some college or university, no certificate or degree	506	26.84%
-0.05921	Professional certificate/ diploma	270	14.32%
0.45468	University degree	480	25.46%
1.16365	Masters degree	283	15.01%

So we used a Maping. We made a list of dictionaries for each of the 12 attributes columns to replace all the numbers by their real values.

In each dictionaries, the first item is the column name and the number of the column associated with this name.

```
Age_maping = {
  'Age' : 1 ,
  -0.95197:'18-24',
  -0.07854: '25-34',
  0.49788:'35-44',
  1.09449: '45-54',
  1.82213: '55-64',
  2.59171: '65+',
}
Liste_maping.append(Age_maping)

Gender_maping = {
  'Gender' : 2 ,
  -0.48246:'Male',
  0.48246: 'Female',
}
Liste_maping.append(Gender_maping)

Education_mapping = {
    'Education': 3,
    -2.43591: 'Left school before 16 years',
    -1.73790: 'Left school at 16 years',
    -1.43719: 'Left school at 17 years',
    -1.22751: 'Left school at 18 years',
    -0.61113: 'Some college or university, no certificate or degree',
    -0.05921: 'Professional certificate/diploma',
    0.45468: 'University degree',
    1.16365: 'Masters degree',
    1.98437: 'Doctorate degree',
}
Liste_maping.append(Education_mapping)

Country_mapping = {
    'Country': 4,
    -0.09765: 'Australia',
    0.24923: 'Canada',
    -0.46841: 'New Zealand',
    -0.28519: 'Other',
```

For the drugs consumption columns, we used 2 dictionaries , one with the labels and what they meant, and one for all of the columns.

```
#Special Mapping for the Consumption of drugs
drug_mapping = {
    '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',
}
column_drug_maping = {
    13: 'Alcohol_consumption',
    14: 'Amphetamines_consumption',
    15: 'Nitrite_consumption',
    16: 'Benzodiazepine_consumption',
    17: 'Caffeine_consumption',
    18: 'Cannabis_consumption',
    19: 'Chocolate_consumption',
    20: 'Cocaine_consumption',
    21: 'Crack consumption',
    22: 'Ecstasy_consumption',
    23: 'Heroin_consumption',
    24: 'Ketamine_consumption',
    25: 'Legalhighs_consumption',
    26: 'LSD_consumption',
    27: 'Methadone_consumption',
    28: 'Mushrooms_consumption',
    29: 'Nicotine_consumption',
    30: 'Semeron_consumption',
    31: 'Volatile_substance_abuse_consumption',

}
```

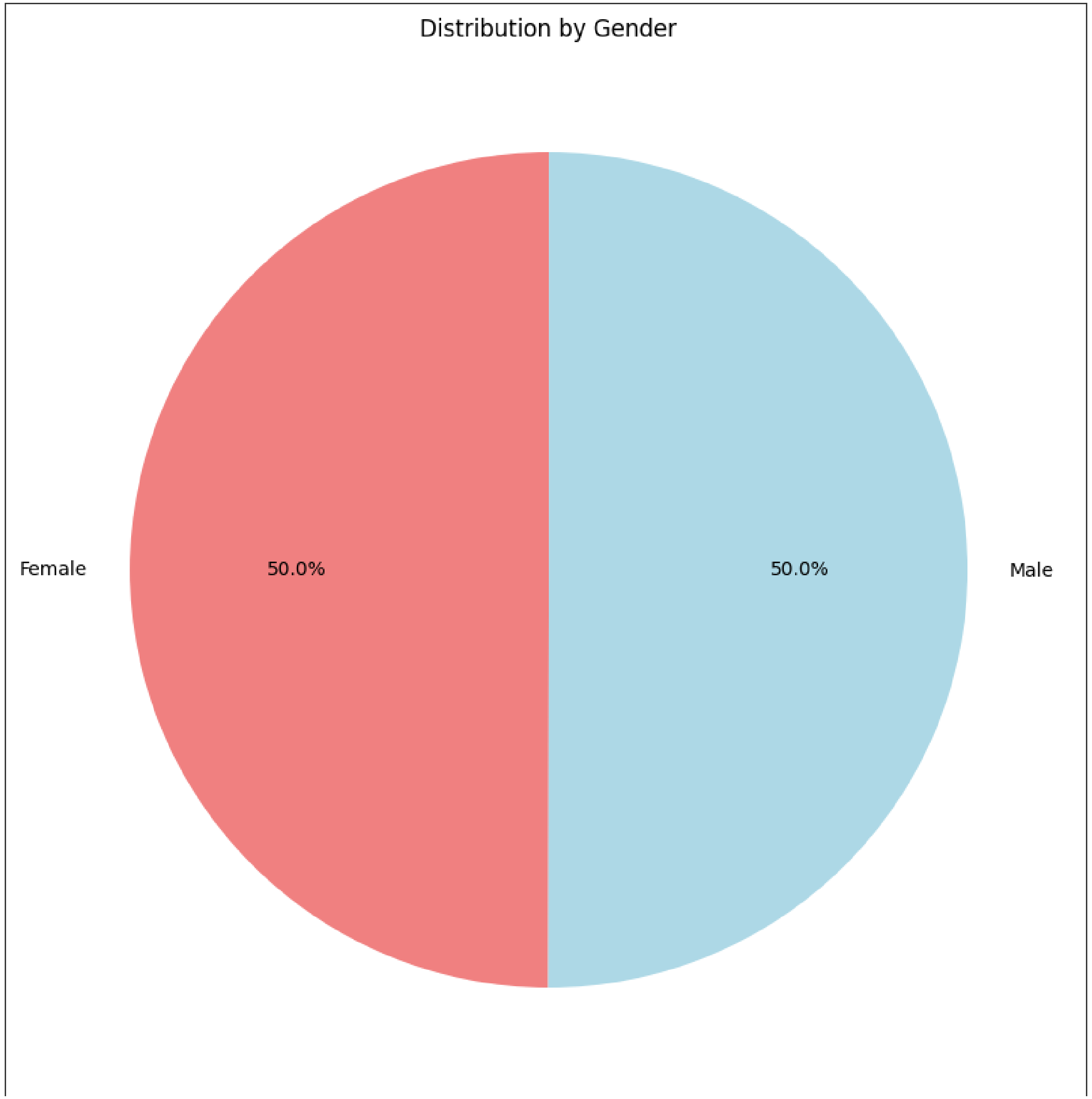
	Age	Gender	Education	Country	Ethnicity	Nscore	Escore	Oscore	Ascore	Cscore	...	Ecstasy_consumption	Her
0	35-44	Female	Professional certificate/diploma	UK	Mixed-White/Asian	39	36	42	37	42	...	Never Used	
1	25-34	Male	Doctorate degree	UK	White	29	52	55	48	41	...	Used in Last Month	
2	35-44	Male	Professional certificate/diploma	UK	White	31	45	40	32	34	...	Never Used	
3	18-24	Female	Masters degree	UK	White	34	34	46	47	46	...	Never Used	
4	35-44	Female	Doctorate degree	UK	White	43	28	43	41	50	...	Used over a Decade Ago	

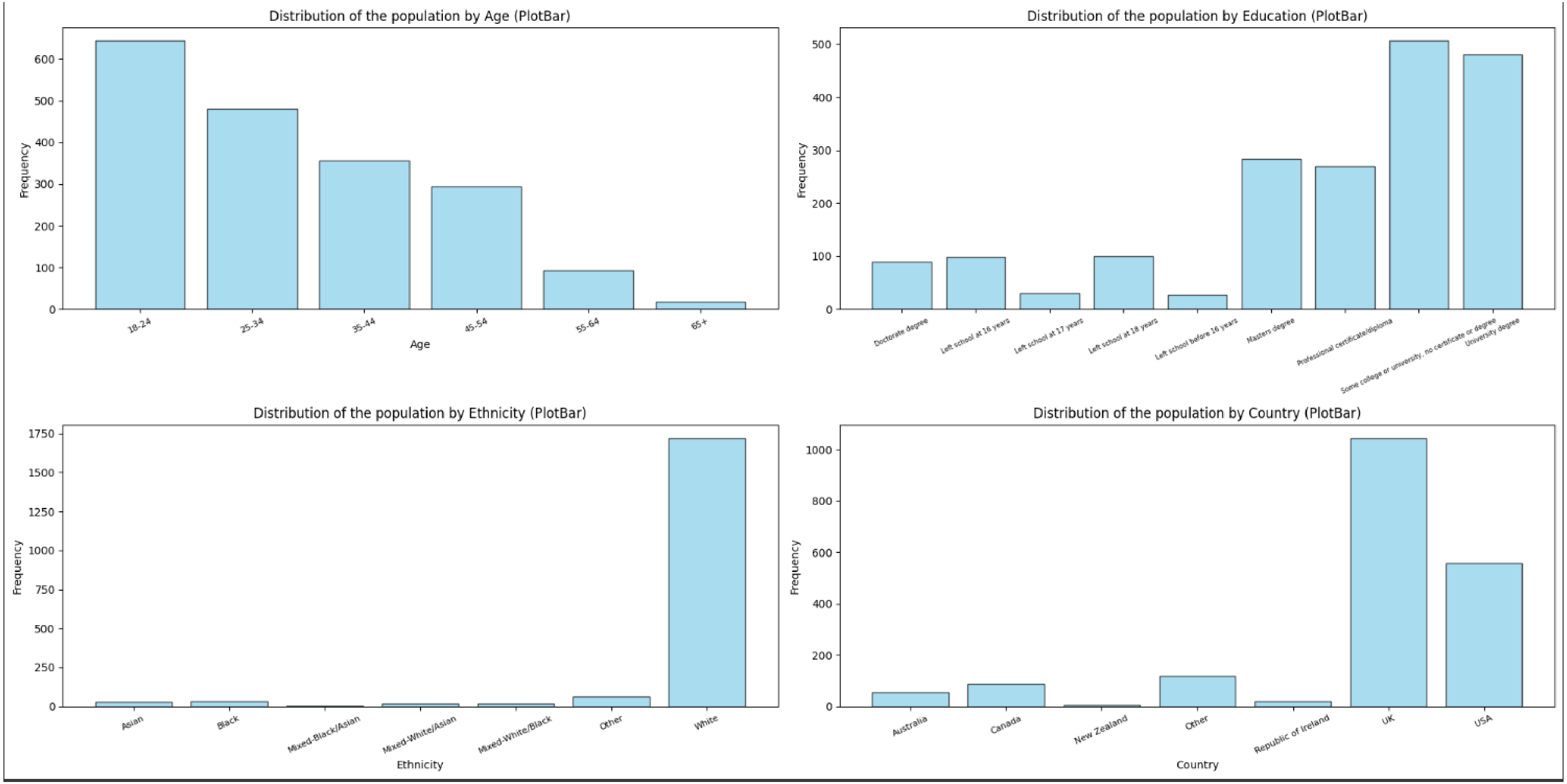
So at the end we have a clean dataset that we can use for our study.

3– Visualizations

Visualizations of the dataset

- So now that we have a readable Dataset , let's see how our population is distributed.
- For exemple we will look at the Gender first and some other attributes after

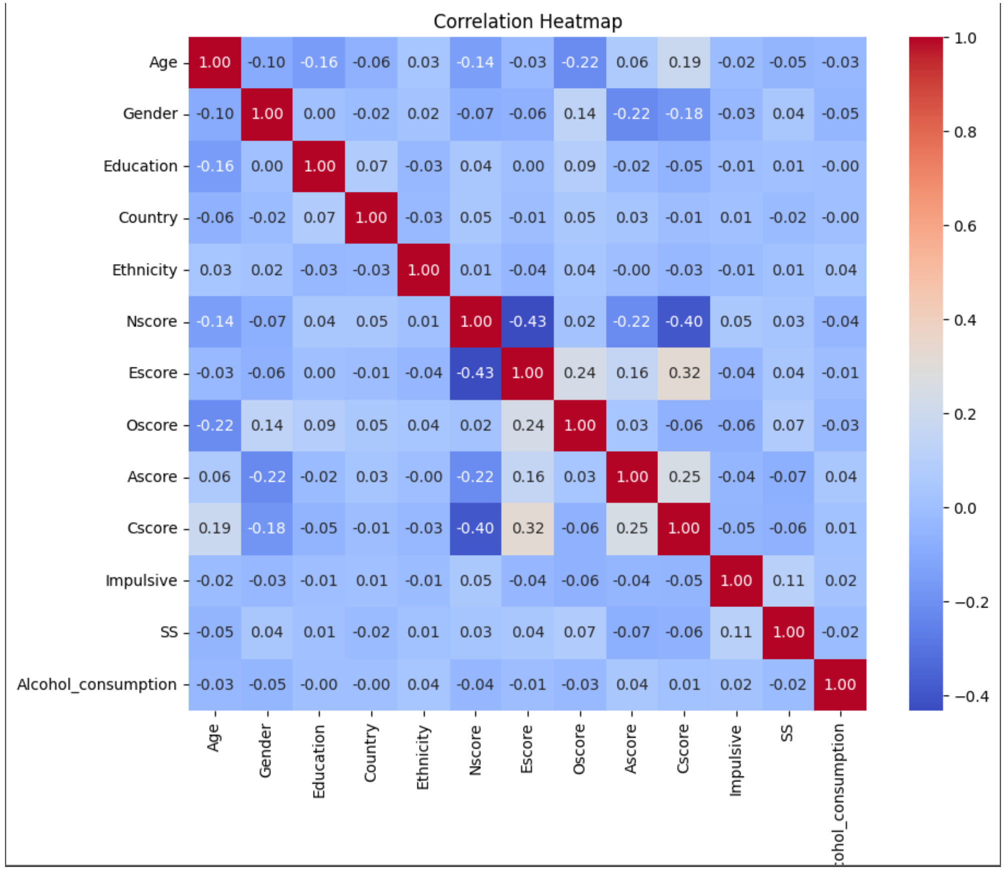

Distribution by Gender
Female
50.0%
50.0%
Male



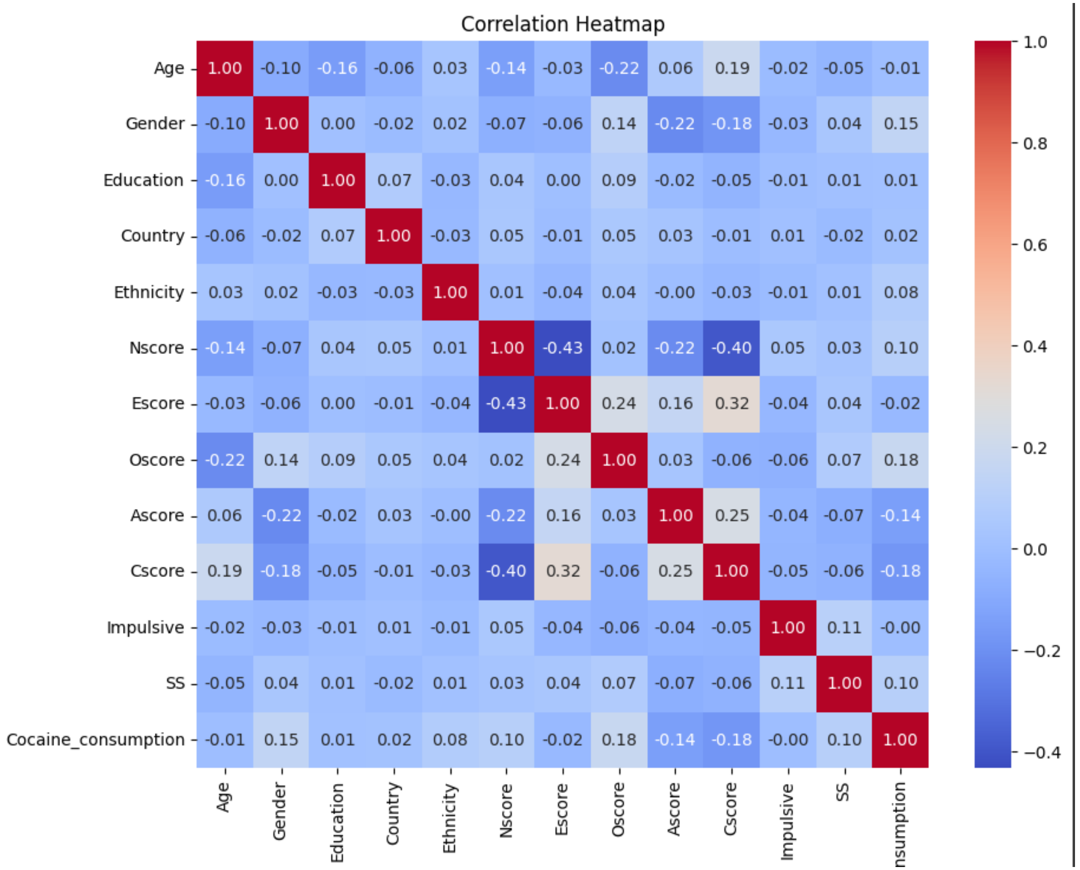
So we can see that our data is really biased, and it will make it complicated for doing predictions

Now lets see the corelation between some consumptions and our attributes

Alcohol Consumption :



Cocain Consumption :



So we can see that the Cocaine consumption will be better to predict because its more corelated to our attributes

4 – Modeling

Encoding

So we encoded our variables for making them numeric with the LabelEncoder()

```
def Label_encode(df,list_columns_encode,list_columns_to_drop):
  le = LabelEncoder()

  for column in list_columns_encode:
    df[column] = le.fit_transform(df[column])
  if list_columns_to_drop is not None:
    for column in list_columns_to_drop:
      df= df.drop(columns=column)

  return df
```

Split Train Test

Then we splited out dataset into Train and Test

```
def split_train_test(df,target):
  X = df.drop(target, axis=1)
  y = df[target]
  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, random_state=123)
  return X_train ,X_test ,y_train ,y_test
```

Modeling

We used 3 algorithms,
Random_Forest, SVM and KNN

```
def Modelisation(X_train ,X_test ,y_train ,y_test,type):
  #Random Forest
  rf_params = {'n_estimators': [50, 100, 200], 'max_depth': [3, 5, 7], 'min_samples_split': [2, 5, 10]}
  rf_model = RandomForestClassifier()
  rf_grid = GridSearchCV(rf_model, rf_params, cv=3)
  rf_grid.fit(X_train, y_train)
  rf_best_model = rf_grid.best_estimator_
  rf_pred = rf_best_model.predict(X_test)
  rf_accuracy = accuracy_score(y_test, rf_pred)

  #SVM
  svm_params = {'C': [1, 10], 'kernel': ['linear', 'rbf']}
  svm_model = SVC()
  svm_grid = GridSearchCV(svm_model, svm_params, cv=3)
  svm_grid.fit(X_train, y_train)
  svm_best_model = svm_grid.best_estimator_
  svm_pred = svm_best_model.predict(X_test)
  svm_accuracy = accuracy_score(y_test, svm_pred)

  #KNN
  knn_params = {'n_neighbors': [3, 5, 7, 9], 'weights': ['uniform', 'distance']}
  knn_model = KNeighborsClassifier()
  knn_grid = GridSearchCV(knn_model, knn_params, cv=3)
  knn_grid.fit(X_train, y_train)
  knn_best_model = knn_grid.best_estimator_
  knn_pred = knn_best_model.predict(X_test)
  knn_accuracy = accuracy_score(y_test, knn_pred)
```

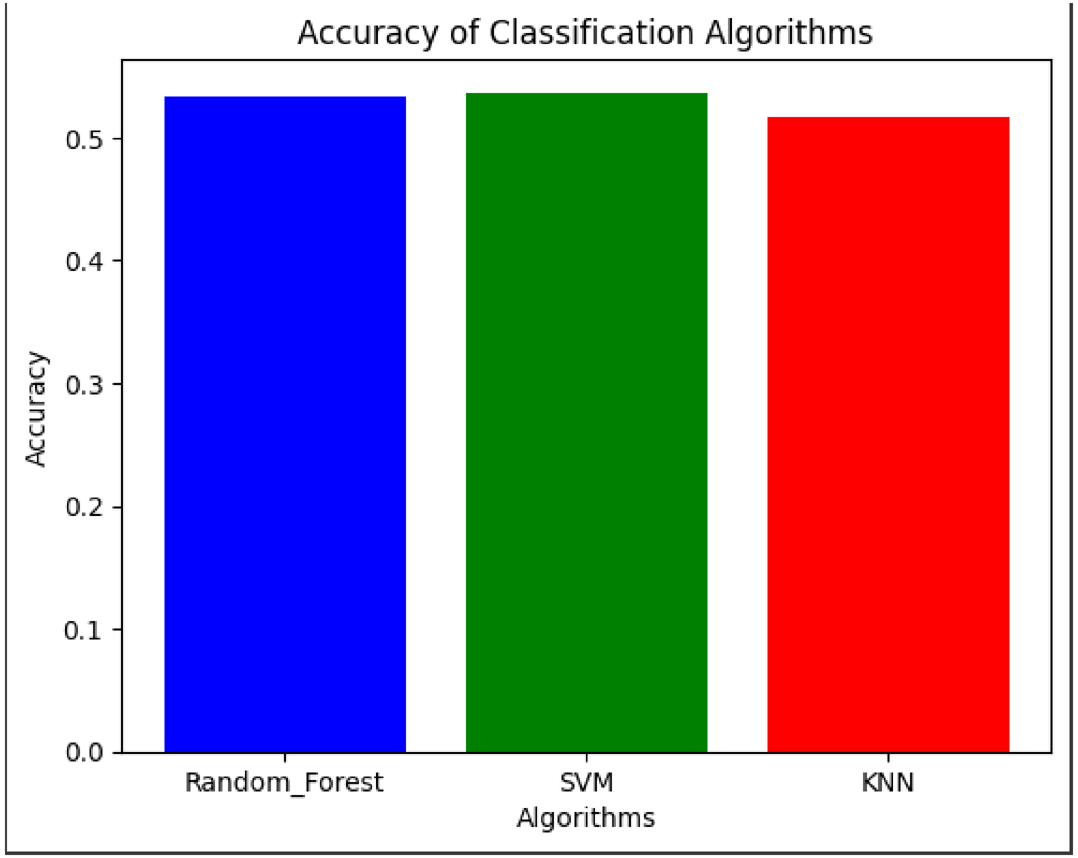
Cocaine Consumption

---> So we will only use the attributes in the first modelisation , and in the second modelisation we will use the attributes and the other drug consumption.

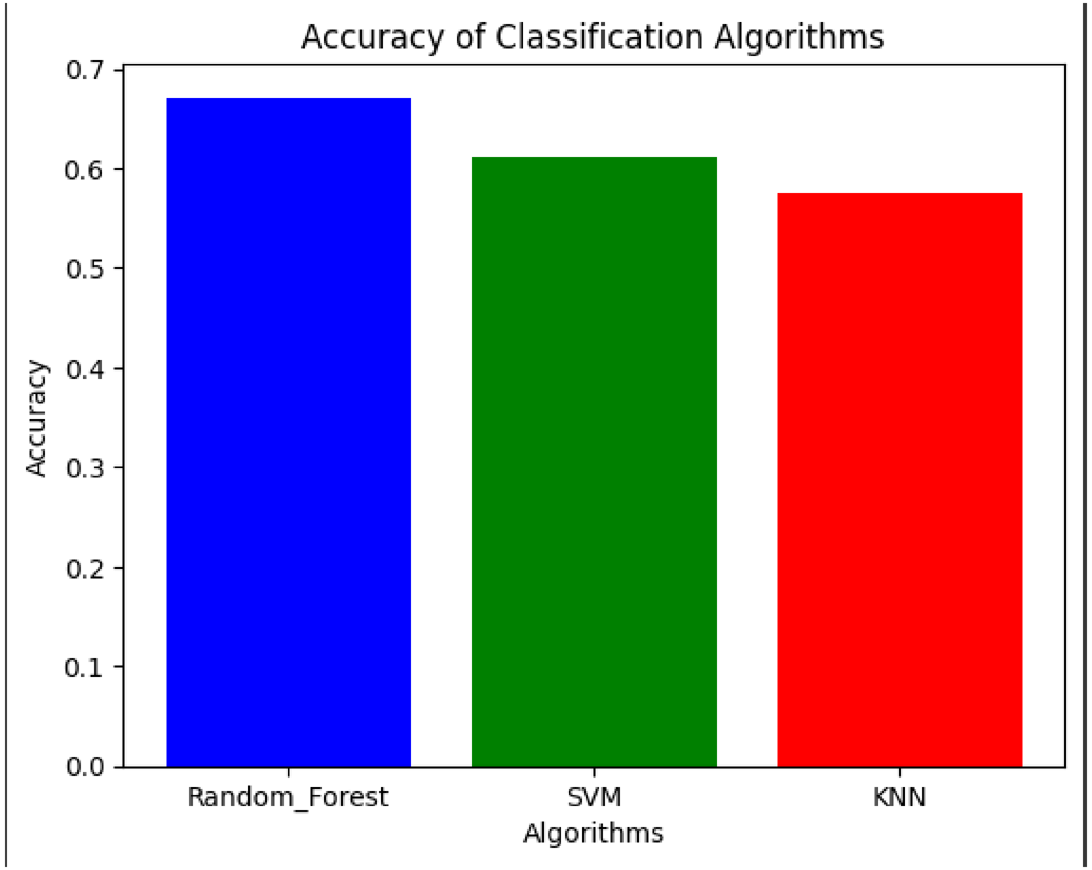
---> Our metric of comparaison between algorithms and method will be the Accuracy

---> We will try to predict all of the labels , so Last day or Last Week or Last Year.

This is the accuracy of the different algorithms using only the attributes



This is the accuracy of the different algorithms using the attributes and drug consumptions



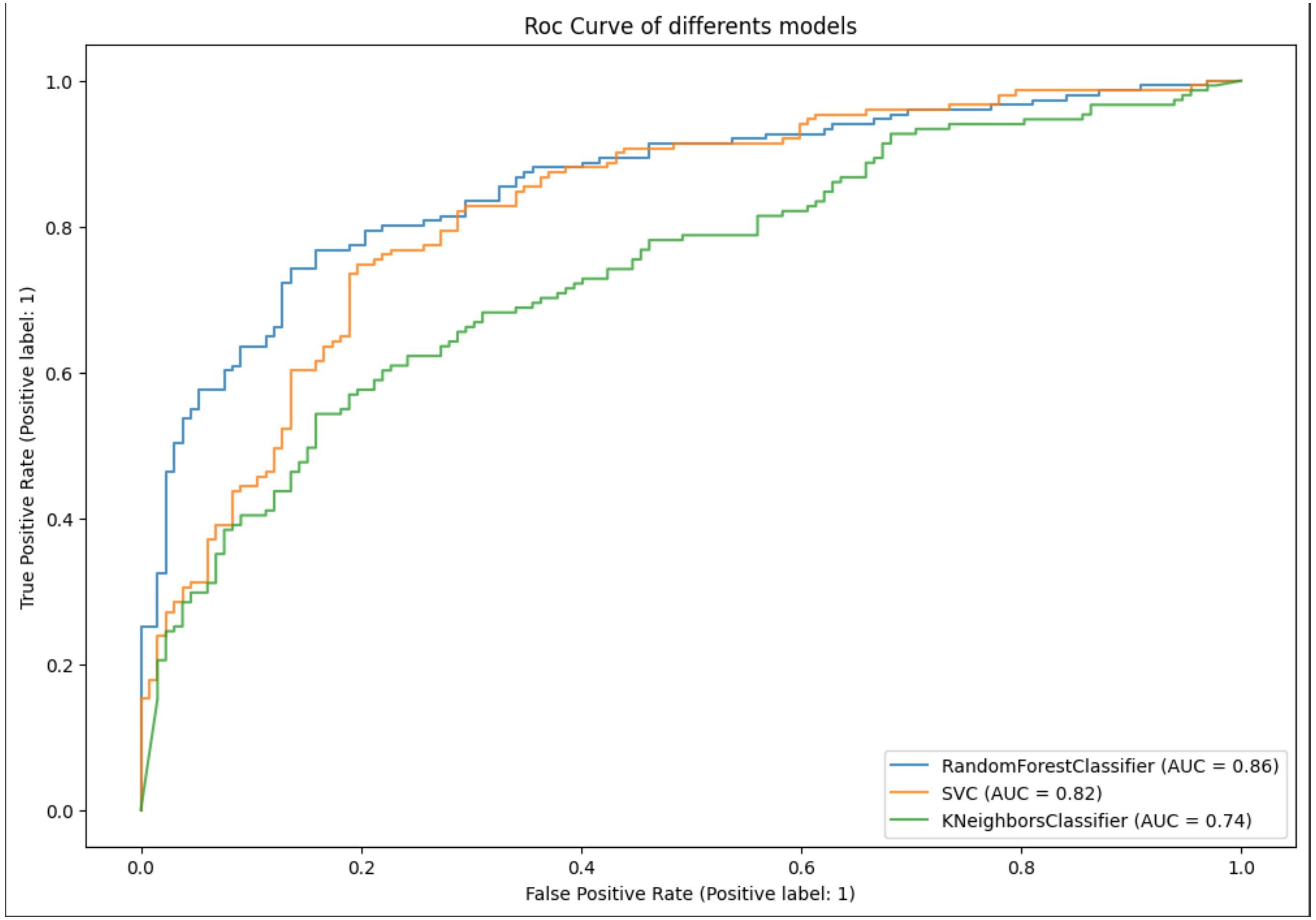
It make sense that using the other drug consumption is doing a better prediction because most of the time if someone is taking cocaine, he will take something else.

Canabis Consumption

---> So we are gonna use only the attributes

---> We will try to predict only 2 labels, people who consumed Canabis in the last month or more recently, and the people who consumed Canabis in the last year or more formerly

---> We will compare each algorithm with a Roc Curve



- So, we can see that the Roc Curve of the Random forest is better, and the precision is 0.86 which is better than the prediction for cocain.
- The key in the modeling for this dataset is to reunite some of the labels together to make it easier to predict.

Conclusion

- This project was very nice to do because we could have use the library sklearn, matplotlib, seaborn and pandas.
- The transformation of the raw data to our dataset was really interesting and maybe could have been done more optimaly.
- The biggest Problem of our predictions is that the data is really biased and not corelated to Drug consumption, so its really hard to make better prediction.
- The thing to make the prediction better is to change the problem from trying to predict 7 distinct labels as last day, last week , last month , but only 2 labels that you define as over the last year or not.