

Python for data analysis

Maxime Cantié, Erwan Bringer



Explication du dataset:

Ayant fait le projet à deux, nous avons le choix entre deux datasets et c'est le dataset drug consumption que nous avons choisis.

Notre dataset comporte 1885 lignes . Chacune des lignes possèdent 12 attributs sur une personne : l'âge, le genre, le niveau d'éducation, le pays de résidence, l'ethnicité, le score de Névrosisme, d'Extraversion, d'Ouverture à l'expérience, d'agréabilité, de Conscience, d'impulsivité et de recherche de sensation.

En plus de ces valeurs, il y a 18 colonnes supplémentaires qui concernent la consommation de la personne de 18 drogues (alcohol, amphetamines, amyl nitrite, benzodiazepine, cannabis, chocolate, cocaine, caffeine, crack, ecstasy, heroin, ketamine, legal highs, LSD, methadone, mushrooms, nicotine and volatile substance abuse ainsi qu'une drogue fictive (Semeron) qui a été introduite pour repérer les personnes non fiables).

Pour toutes ces drogues, la personne devait choisir une des réponses suivantes : "Never Used", "Used over a Decade Ago", "Used in Last Decade", "Used in Last Year", "Used in Last Month", "Used in Last Week", and "Used in Last Day".

Analyse du dataset :

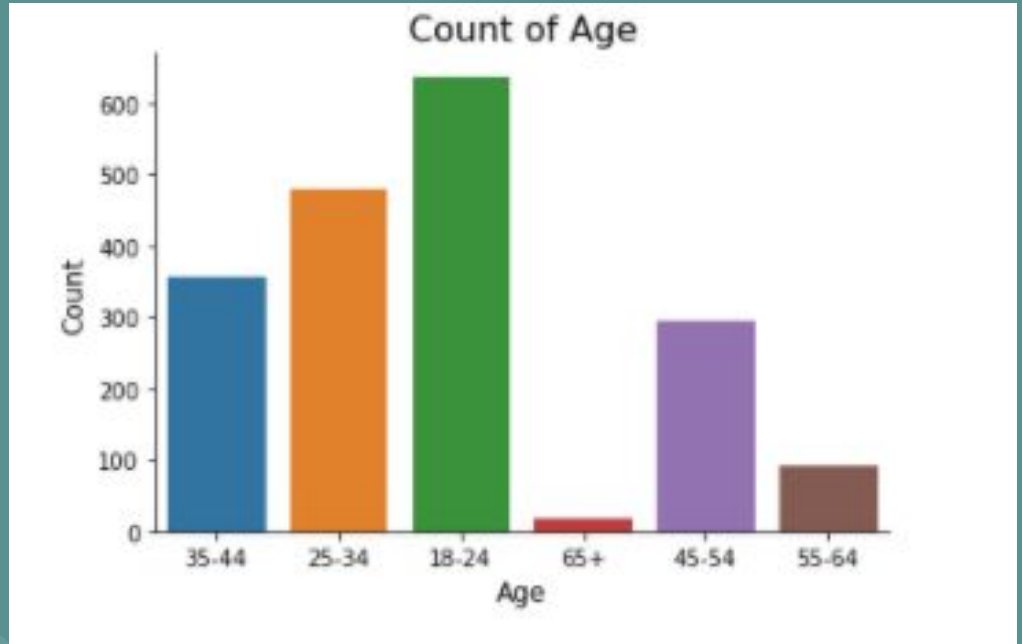
Nous avons dans un premier temps analysé le dataset en enlevant premièrement les personnes non-fiables que nous avons repéré grâce à leurs réponses sur leurs consommations de semeron.

Cette analyse s'est coupé en 3 parties avec une partie sur l'étude de la population qui constitue le dataset, puis une étude sur les consommations de drogues et enfin sur une étude de la personnalité avec l'impact de la personnalité sur la consommation de drogue.



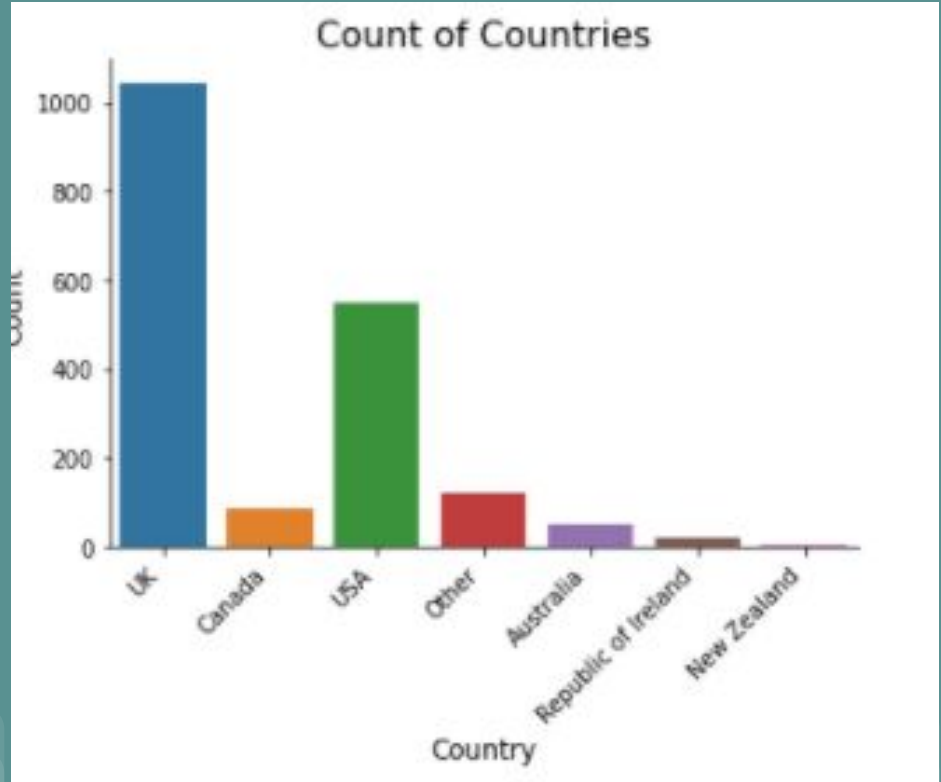
Analyse du dataset (population analysée):

Tout d'abord nous remarquons que les personnes les plus présentes dans ce dataset sont comprises entre 18 et 24 ans et que plus l'âge augmente, moins il est présent dans cette étude.



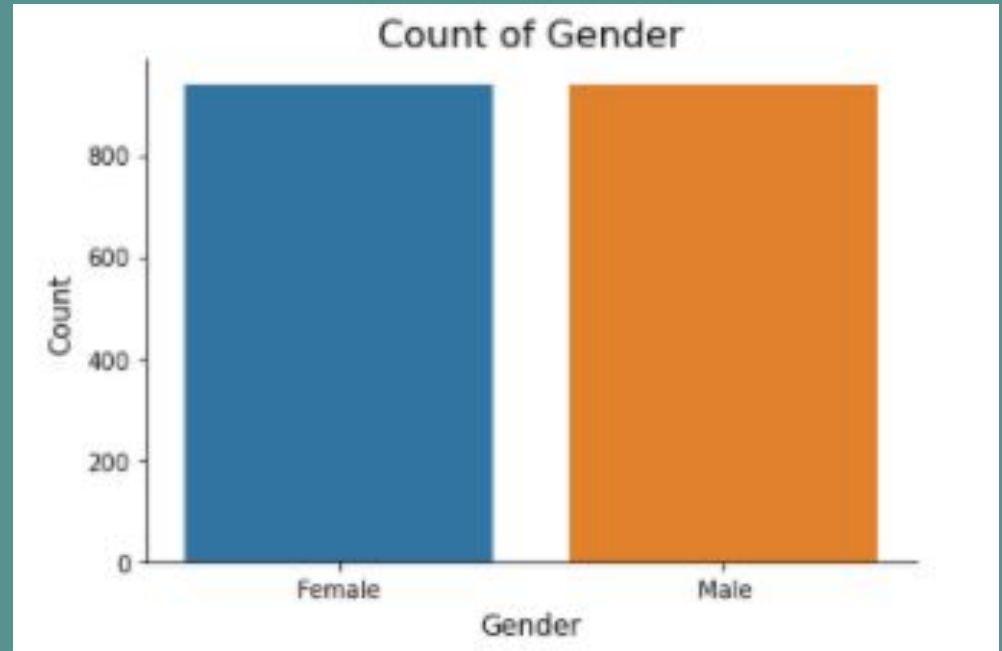
Analyse du dataset (population analysée):

Pour ce qui est des pays représentés, on rencontre surtout des pays anglophones avec une majorité de personne venant de Grande-Bretagne et des Etats-Unis.



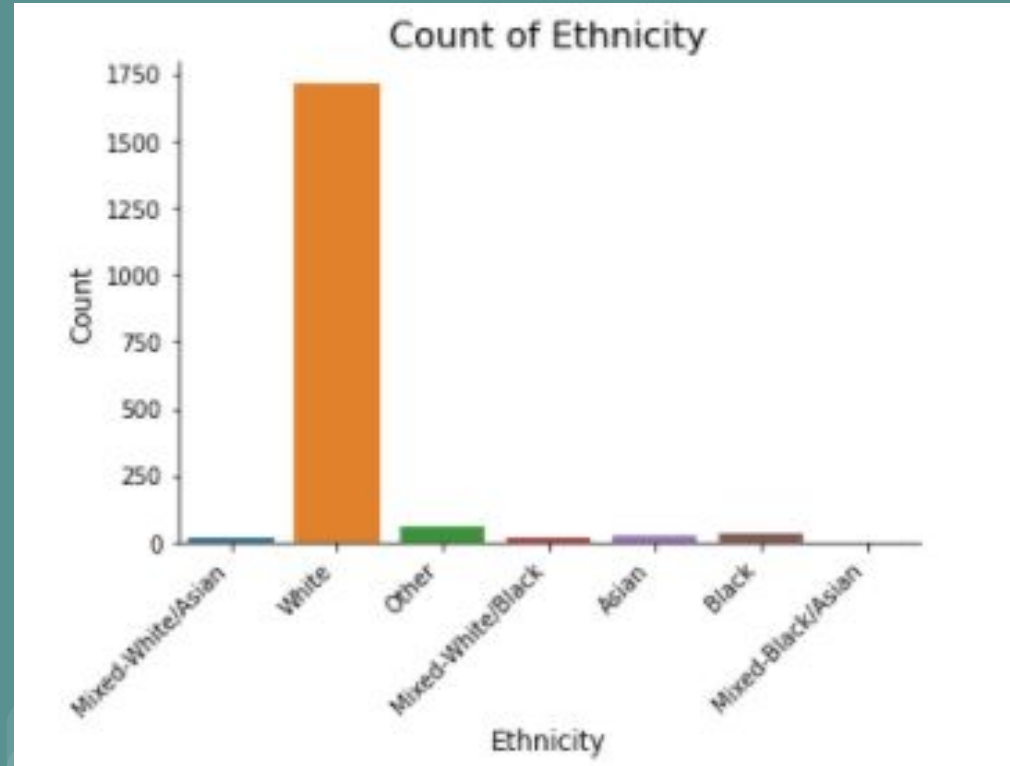
Analyse du dataset (population analysée):

Au niveau du sexe des personnes interrogées, le dataset est équilibré avec plus ou moins autant de femme que d'homme.



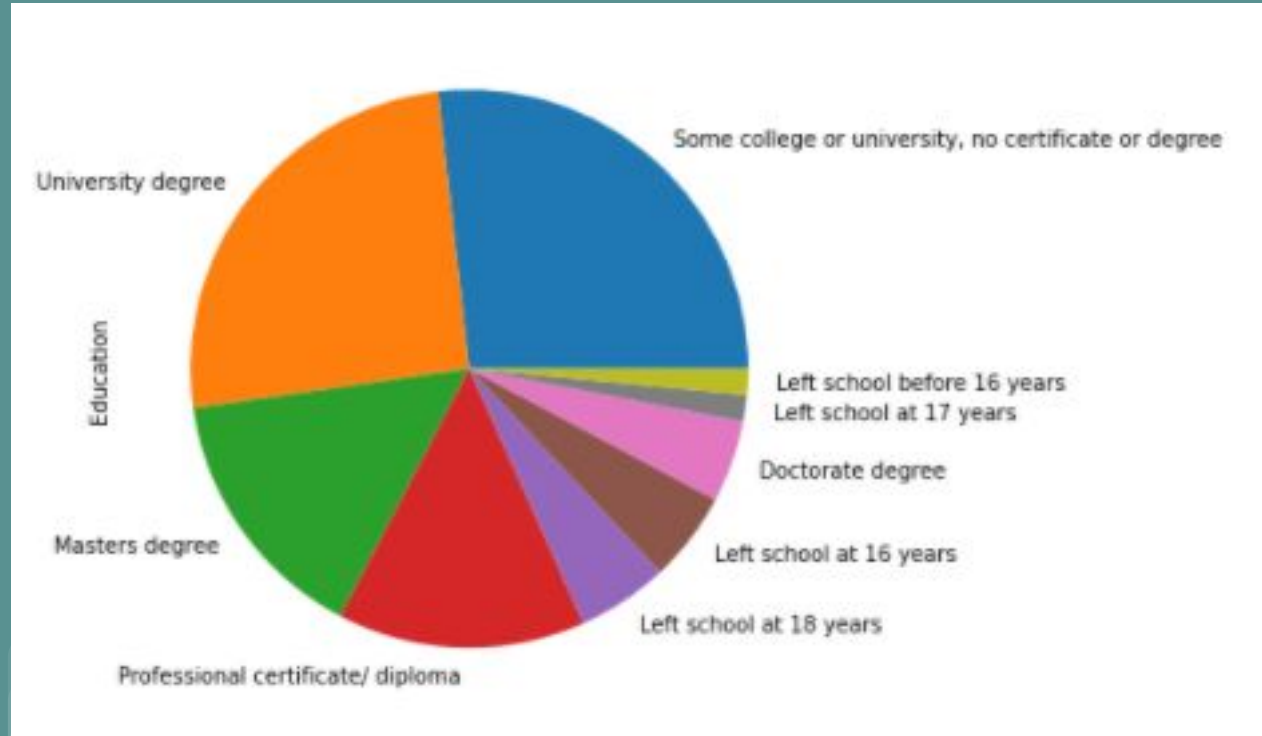
Analyse du dataset (population analysée):

Par contre, pour ce qui est de l'Ethnie des personnes interrogées, le dataset est très majoritairement constitué de personnes blanches.



Analyse du dataset (population analysée):

Pour ce qui est de l'éducation, il y a un peu de tout avec une majorité de personne ayant un diplôme universitaire ou ayant été dans une université sans avoir fini le diplôme.



Analyse du dataset (étude des conso de drogue):

On remarque que l'alcool, la caféine, le chocolat et la nicotine sont des drogues qui sont utilisées plutôt régulièrement par les personnes qui constituent le dataset tandis que les autres drogues sont majoritairement jamais consommé par ces mêmes personnes.

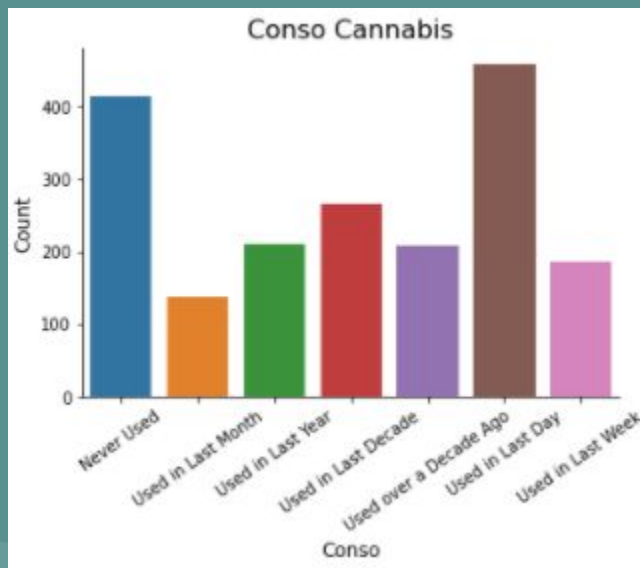
Par ailleurs, la distribution du cannabis semble très équilibré au vu du faible nombre de personnes qui constituent la plus grande catégorie pour l'utilisation de ce dernier.

```
In[94]: 1 conso_DF.describe()
```

Out[94]:

	Alcohol	Amphet	Amyl	Benzos	Caff	Cannabis	Choc	Coke	Crack	Ecstasy	Heroin	Ketamine	Legalh	LSD	Meth	Mushrooms	Nicotine	Sen
count	1877	1877	1877	1877	1877	1877	1877	1877	1877	1877	1877	1877	1877	1877	1877	1877	1877	1877
unique	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
top	Used in Last Week	Never Used	Never Used	Never Used	Used in Last Day	Used in Last Day	Used in Last Day	Never Used	Never Used	Never Used	Never Used	Never Used	Never Used	Never Used	Never Used	Never Used	Used in Last Day	Never Used
freq	758	973	1299	999	1380	458	805	1036	1622	1020	1600	1488	1092	1069	1424	982	607	1600

Analyse de la consommation de cannabis :



	Cannabis	Never Used	Used in Last Day	Used in Last Decade	Used in Last Month	Used in Last Week	Used in Last Year	Used over a Decade Ago
Gender	Country							
Female	Australia	10.0	20.0	15.0	15.0	15.0	15.0	10.0
	Canada	22.0	19.5	22.0	7.3	9.8	7.3	12.2
	New Zealand	0.0	0.0	100.0	0.0	0.0	0.0	0.0
	Other	25.0	22.2	16.7	2.8	13.9	8.3	11.1
	Republic of Ireland	33.3	33.3	11.1	11.1	0.0	0.0	11.1
	UK	42.1	5.1	20.5	2.4	3.0	10.3	16.5
	USA	5.3	44.9	7.2	13.5	14.5	12.6	1.9
Male	Australia	0.0	31.2	9.4	15.6	21.9	18.8	3.1
	Canada	8.7	41.3	6.5	13.0	8.7	17.4	4.3
	New Zealand	0.0	50.0	0.0	50.0	0.0	0.0	0.0
	Other	3.7	39.0	8.5	13.4	20.7	12.2	2.4
	Republic of Ireland	0.0	45.5	0.0	18.2	18.2	18.2	0.0
	UK	24.7	17.8	15.7	5.5	9.3	9.5	17.6
	USA	1.7	48.5	7.0	11.0	16.0	13.1	2.6

Analyse sur la personnalité :

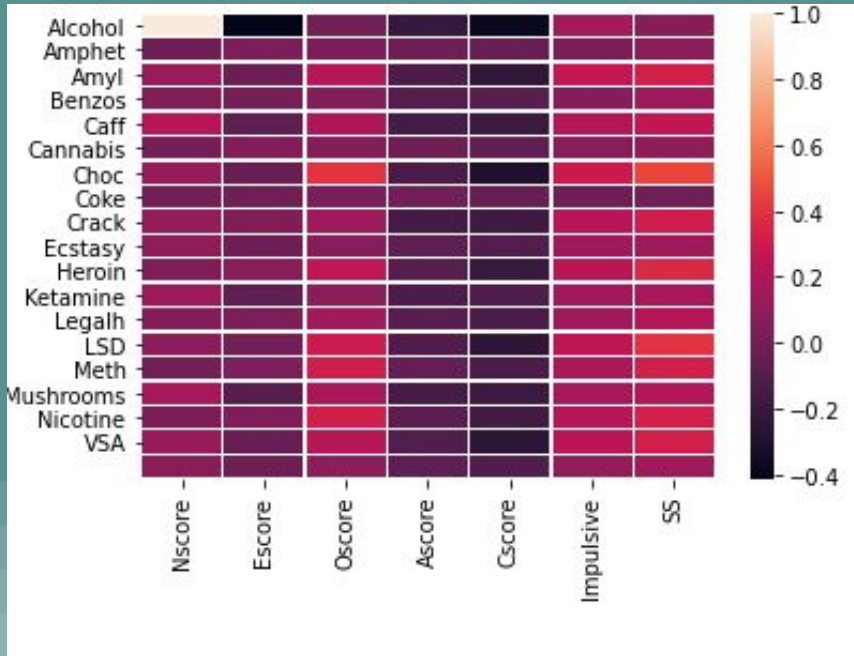
Nscore = Névrosisme, Escore = Extraversion, Oscore = Ouverture à l'expérience, Ascore = agréabilité, Cscore = Conscience, Impulsive = impulsivité, SS = rechercher de sensation.

	Nscore	Escore	Oscore	Ascore	Cscore	Impulsive	SS
count	1863.000000	1877.000000	1877.000000	1877.000000	1877.000000	1877.000000	1877.000000
mean	-0.015468	-0.001951	-0.003224	-0.000657	-0.000394	0.005293	-0.007408
std	0.987186	0.997418	0.995691	0.996689	0.997657	0.954148	0.962074
min	-3.464360	-3.273930	-3.273930	-3.464360	-3.464360	-2.555240	-2.078480
25%	-0.678250	-0.695090	-0.717270	-0.606330	-0.652530	-0.711260	-0.525930
50%	-0.051880	0.003320	-0.019280	-0.017290	-0.006650	-0.217120	0.079870
75%	0.629670	0.637790	0.723300	0.760960	0.584890	0.529750	0.765400
max	3.273930	3.273930	2.901610	3.464360	3.464360	2.901610	1.921730

Analyse sur la personnalité :

Analyse des corrélations entre les personnalités et la consommation de drogues :

Nous pouvons alors remarquer qu'il existe des corrélations entre la consommation de drogue et la personnalité comme avec le Cscore qui est toujours corrélé négativement à la consommation des différentes drogues.



Le problème :

Notre tâche est de déterminer dans quelle catégorie se situe la personne pour une drogue donnée en fonction des 12 premiers attributs. (cad sans utiliser ses réponses sur sa consommation de drogue).

Pour cela, nous allons rendre le problème binaire en associant "Never Used", "Used over a Decade Ago", "Used in Last Decade" comme ancien ou non consommateur et "Used in Last Year", "Used in Last Month", "Used in Last Week", and "Used in Last Day" comme utilisateur récent.

Notre travail sera donc une tâche de classification entre ancien ou non consommateur et consommateur récent grâce aux 12 premiers attributs.

Modélisation :

Nous allons alors pour chaque drogue mettre la consommation de cette drogue dans un dataframe y et les features qui serviront pour la prédiction dans un dataframe X :

Entrée [113]:

```
1 X.head()
```

Out[113]:

	Age	Gender	Education	Country	Ethnicity	Nscore	Escore	Oscore	Ascore	Cscore	Impulsive	SS
0	0.49788	0.48246	-0.05921	0.96082	0.12600	0.31287	-0.57545	-0.58331	-0.91699	-0.00665	-0.21712	-1.18084
1	-0.07854	-0.48246	1.98437	0.96082	-0.31685	-0.67825	1.93886	1.43533	0.76096	-0.14277	-0.71126	-0.21575
2	0.49788	-0.48246	-0.05921	0.96082	-0.31685	-0.46725	0.80523	-0.84732	-1.62090	-1.01450	-1.37983	0.40148
3	-0.95197	0.48246	1.16365	0.96082	-0.31685	-0.14882	-0.80615	-0.01928	0.59042	0.58489	-1.37983	-1.18084
4	0.49788	0.48246	1.98437	0.96082	-0.31685	0.73545	-1.63340	-0.45174	-0.30172	1.30612	-0.21712	-0.21575

Modélisation :

Après avoir split les données en train/test set (80/20%) nous avons remplacé les NA présents par des 0 puis nous avons testé plusieurs modèles sur la consommation d'alcool :

Modèle 1 : Logistic Regression :

Nous avons une accuracy de 92,6% pour ce test qui est un bon score, cependant nous pouvons voir que les prédictions sont quasi-uniquement des 1 et donc comme le dataset avait beaucoup de 1 il a du sur-apprendre sur les 1 ou sous-apprendre sur les 0 (100% d'erreurs pour les 0)

Pour corriger cela nous allons donc réaliser un resampling du training set pour rééquilibrer les labels.

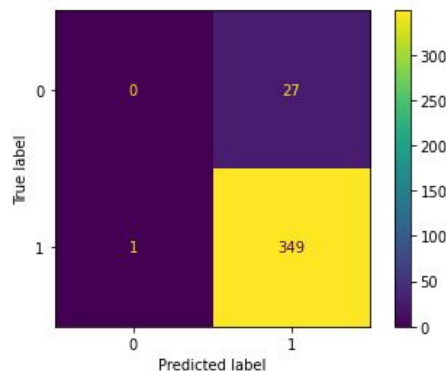
```
1 acc_lr = accuracy_score(y_true=y_test, y_pred=logreg.predict(X_test))  
2 acc_lr
```

0.9257294429708223

92,8% qui est un bon résultat regardons la matrice de confusion

```
1 plot_confusion_matrix(logreg, X=X_test, y_true=y_test)
```

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2c64a3f1f40
>



Resampling :

Rééquilibrage des labels pour avoir des résultats moins orientés en fonction du set donné.

```
1 X_train['Alcohol']=y_train
2
3 from sklearn.utils import resample
4 # Exemple up sampling - pour les données déséquilibrées
5 # Séparer class majoritaires et class minoritaires
6 df_minoritytrain = X_train[X_train.Alcohol==0]
7 df_majoritytrain = X_train[X_train.Alcohol==1]
8
9 # Sur-classe minoritaire
10 df_minority_upsampledtrain = resample(df_minoritytrain,
11 replace=True, # sample without replacement
12 n_samples=df_majoritytrain.shape[0], # to match minority class
13 random_state=123) # reproducible results
14
15 # Combiner la classe minoritaire avec la classe minoritaire suréchantillonnée
16 df = pd.concat([df_majoritytrain, df_minority_upsampledtrain])
17
18
19 y_train=df['Alcohol']
20 X_train=df
21 X_train=X_train.drop(['Alcohol'],axis='columns')
```

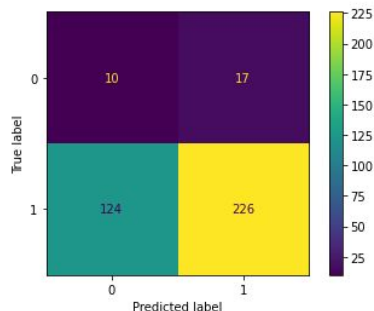
Résultat de Lr après resampling (bien moins bon (65%) mais l'ancien modèle était inutile car il ne prédisait que des 1)

```
Entrée [124]: 1 acc_lr = accuracy_score(y_true=y_test, y_pred=logreg.predict(X_test))
2 acc_lr
```

```
Out[124]: 0.6259946949602122
```

```
Entrée [125]: 1 plot_confusion_matrix(logreg, X=X_test, y_true=y_test)
```

```
Out[125]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2c64a47f5b0>
```



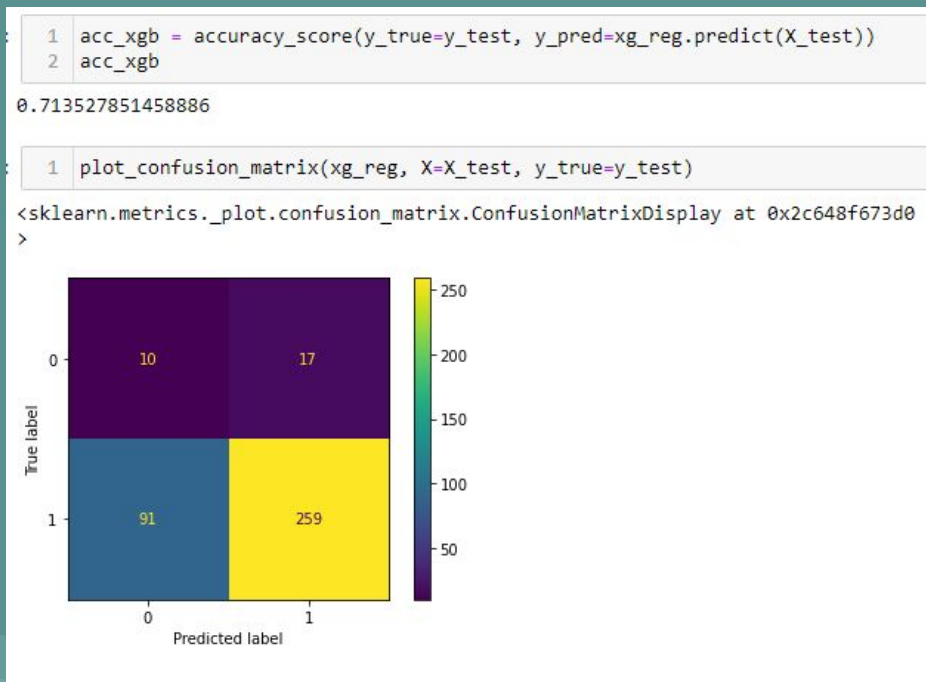
Modélisation :

Pour tous les modèles suivants, nous avons pris le training set resamplé.

Modèle 2 : XGBoost :

Nous avons une accuracy de 71,4% pour ce test qui est un meilleur score que celui que nous avons obtenue avec LR.

Les 0 sont beaucoup mieux prédit que précédemment.



Modélisation :

Modèle 3 : MLP :

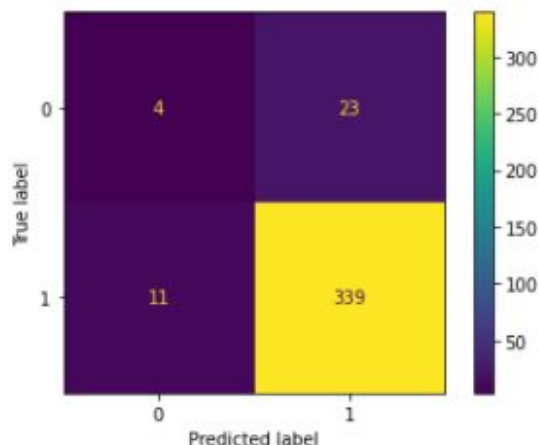
Accuracy de 91% mais qui est contrasté par le fait que les 0 sont moins bien prédit que précédemment.

```
1 acc_mlp = accuracy_score(y_true=y_test, y_pred=MLP_model.predict(X_test))  
2 acc_mlp
```

0.9098143236074271

```
1 plot_confusion_matrix(MLP_model, X=X_test, y_true=y_test)
```

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2c64a1d1370>



Modélisation :

Modèle 4 : Random Forest :

Accuracy de 92,5% mais qui prédit très peu les 0 une nouvelle fois.

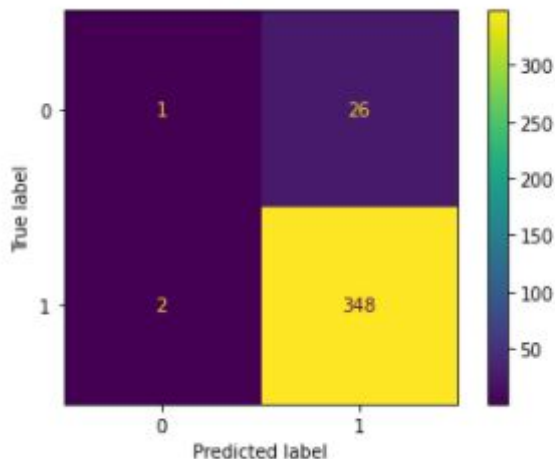
Malgré cela, nous avons décidé de garder le modèle avec le plus d'accuracy pour réaliser l'API

```
: 1 acc_rf = accuracy_score(y_true=y_test, y_pred=clf.predict(X_test))  
2 acc_rf
```

0.9257294429708223

```
: 1 plot_confusion_matrix(clf, X=X_test, y_true=y_test)
```

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2c64a1d13d0>



Modélisation des autres consos de drogues:

Nous avons alors regroupé tout ce qui est présenté précédemment pour toutes les drogues et nous récupérons le modèle avec la plus grande accuracy pour chaque drogue (exemple du code sur la prochaine slide).

Amphet consommation

```
1 # On ne fait que reprendre les étapes précédemment effectuées avec l'alcool
2
3
4 X, y = drug_DF.drop(['Alcohol', 'Semer', 'ID', 'Amphet', 'Amyl', 'Benzos', 'Caff', 'Cannabis',
5                     'Ecstasy', 'Heroin', 'Ketamine', 'Legalh', 'LSD', 'Meth', 'Mushrooms',
6                     'Nicotine', 'VSA'], axis=1), drug_DF["Amphet"]
7
8
9 #Train/test split
10 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
11
12 X_train.isna().sum()
13 X_train = X_train.fillna(0)
14 X_test = X_test.fillna(0)
15
16 X_train['Amphet']=y_train
17
18
19 #Resample
20 from sklearn.utils import resample
21 # Exemple up sampling - pour les données déséquilibrées
22 # Séparer class majoritaires et class minoritaires
23 if X_train[X_train.Amphet==0].shape[0]<X_train[X_train.Amphet==1].shape[0]:
24     df_minoritytrain = X_train[X_train.Amphet==0]
25     df_majoritytrain = X_train[X_train.Amphet==1]
26 else :
27     df_minoritytrain = X_train[X_train.Amphet==1]
28     df_majoritytrain = X_train[X_train.Amphet==0]
29
30 # Sur-classe minoritaire
31 df_minority_upsampledtrain = resample(df_minoritytrain,
32                                     replace=True, # sample without replacement
33                                     n_samples=df_majoritytrain.shape[0], # to match minority class
34                                     random_state=123) # reproducible results
35
36 # Combiner la classe minoritaire avec la classe minoritaire suréchantillonnée
37 df = pd.concat([df_majoritytrain, df_minority_upsampledtrain])
38
39 y_train=df['Amphet']
40 X_train=df
41 X_train=X_train.drop(['Amphet'],axis='columns')
42 Liste_acc=[]
43
44
45 #LR
46
47 logreg = LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
48                             intercept_scaling=1, l1_ratio=None, max_iter=1000,
49                             multi_class='auto', n_jobs=None, penalty='l2',
50                             random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
51                             warm_start=False)
52
53 logreg.fit(X_train, y_train)
54 acc_lr = accuracy_score(y_true=y_test, y_pred=logreg.predict(X_test))
55 Liste_acc+= [acc_lr]
56
57 #XGB
```

```
58 #XGB
59
60 xg_reg = xgb.XGBClassifier(objective='binary:logistic', colsample_bytree = 0.3, learning_rate = 0.1,
61                           max_depth = 5, alpha = 10, n_estimators = 10)
62
63 xg_reg.fit(X_train, y_train)
64 acc_xgb = accuracy_score(y_true=y_test, y_pred=xg_reg.predict(X_test))
65 Liste_acc+= [acc_xgb]
66
67 #MLP
68
69 model_params = {
70     'alpha': 0.01,
71     'batch_size': 256,
72     'epsilon': 1e-08,
73     'hidden_layer_sizes': (300,),
74     'learning_rate': 'adaptive',
75     'max_iter': 500,
76 }
77
78 # initialize Multi Layer Perceptron classifier
79 # with best parameters ( so far )
80 MLP_model = MLPClassifier(*model_params, random_state=0)
81
82 MLP_model.fit(X_train, y_train)
83 acc_mlp = accuracy_score(y_true=y_test, y_pred=MLP_model.predict(X_test))
84 Liste_acc+= [acc_mlp]
85
86 #RF
87
88 clf=RandomForestClassifier(n_estimators=100, random_state=0)
89
90 clf.fit(X_train, y_train)
91
92 acc_rf = accuracy_score(y_true=y_test, y_pred=clf.predict(X_test))
93 Liste_acc+= [acc_rf]
94
95 if max(Liste_acc)==acc_lr :
96     pickle.dump(logreg, open('C:/Users/erwan/OneDrive/Documents/COURS/Projet_Data/final_prediction_Amphet.pickle', 'wb'))
97 if max(Liste_acc)==acc_xgb :
98     pickle.dump(xg_reg, open('C:/Users/erwan/OneDrive/Documents/COURS/Projet_Data/final_prediction_Amphet.pickle', 'wb'))
99 if max(Liste_acc)==acc_mlp :
100     pickle.dump(MLP_model, open('C:/Users/erwan/OneDrive/Documents/COURS/Projet_Data/final_prediction_Amphet.pickle', 'wb'))
101 if max(Liste_acc)==acc_rf :
102     pickle.dump(clf, open('C:/Users/erwan/OneDrive/Documents/COURS/Projet_Data/final_prediction_Amphet.pickle', 'wb'))
```

15:37:50! WARNING: C:/Users/Administrator/workspace/xgboost.win64.release.1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3

Api avec flask

← → ↻ ⓘ 127.0.0.1:5000

API pour tester si vous êtes ou non consommateur de drogue

Veuillez remplir le questionnaire ci-dessous pour prédire si vous avez consommé la drogue choisi ces 365 derniers jours

drogue :

Age :

Sexe :

Education :

Pays :

Ethnicity :

Nscore

Escore

Oscore

Ascore

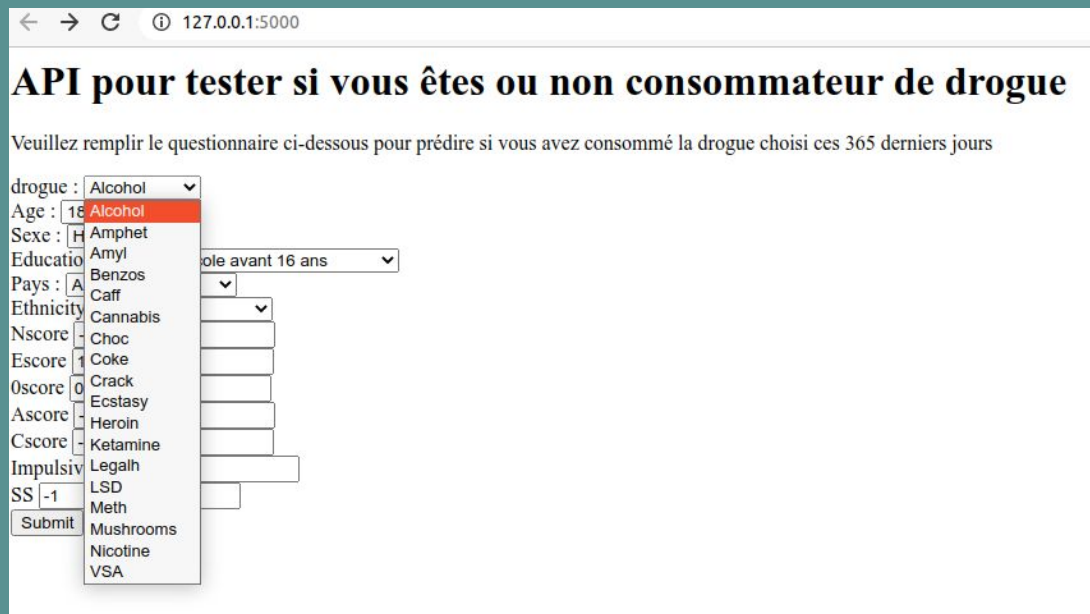
Cscore

Impulsive

SS

Pour tester avec un modèle implémenté, il suffit de remplir un formulaire qui à pour premier champs le choix du modèle à utiliser (pour quelle drogue on veut faire le test), et les 12 autres valeurs qui sont les paramètres en entré de ce modèle.

Api avec flask



A screenshot of a web browser window displaying a Flask API interface. The browser's address bar shows the URL '127.0.0.1:5000'. The page title is 'API pour tester si vous êtes ou non consommateur de drogue'. Below the title, there is a instruction: 'Veuillez remplir le questionnaire ci-dessous pour prédire si vous avez consommé la drogue choisi ces 365 derniers jours'. The form contains several input fields and a dropdown menu. The 'drogue' field is a dropdown menu with 'Alcohol' selected. The 'Age' field is a text input with '18' entered. The 'Sexe' field is a dropdown menu with 'H' selected. The 'Education' field is a dropdown menu with 'Amyl' selected. The 'Pays' field is a dropdown menu with 'A' selected. The 'Ethnicity' field is a dropdown menu with 'Cannabis' selected. The 'Nscore' field is a text input with '1' entered. The 'Escore' field is a text input with '0' entered. The 'Oscore' field is a text input with '0' entered. The 'Ascore' field is a text input with '0' entered. The 'Cscore' field is a text input with '0' entered. The 'Impulsiv' field is a text input with '0' entered. The 'SS' field is a text input with '-1' entered. There is a 'Submit' button. A dropdown menu is open next to the 'drogue' field, showing a list of drugs: Alcohol, Amphet, Amyl, Benzos, Caff, Cannabis, Choc, Coke, Crack, Ecstasy, Heroin, Ketamine, Legalh, LSD, Meth, Mushrooms, Nicotine, and VSA. The 'Alcohol' option is highlighted in red. There is also a dropdown menu for 'choisi ces 365 derniers jours' with 'avant 16 ans' selected.

API pour tester si vous êtes ou non consommateur de drogue

Veuillez remplir le questionnaire ci-dessous pour prédire si vous avez consommé la drogue choisi ces 365 derniers jours

drogue : Alcohol
Age : 18
Sexe : H
Education : Amyl
Pays : A
Ethnicity : Cannabis
Nscore : 1
Escore : 0
Oscore : 0
Ascore : 0
Cscore : 0
Impulsiv : 0
SS : -1
Submit

Alcohol
Amphet
Amyl
Benzos
Caff
Cannabis
Choc
Coke
Crack
Ecstasy
Heroin
Ketamine
Legalh
LSD
Meth
Mushrooms
Nicotine
VSA

avant 16 ans

On peut voir des barres déroulantes qui permettent de changer les différents champs tels que : Le modèle (quelle drogue), l'âge, le sexe, l'éducation, le pays et l'ethnie.

Nous avons donc 18 modèles disponibles qui correspondent à toutes les drogues présentes dans le dataset

Api avec flask

Voici les résultats possibles en sortis du formulaire, dans le premier screen on peut voir que l'usager a été considéré comme étant probablement non consommateur de la drogue qu'il a choisi de tester, tandis que sur le second screen il est probablement consommateur.

A noter qu'il y a une marge d'erreur correspondante au modèle utilisé.

