

Extract from the Python code of the Logit model implementation

```
[1]: import os as os
import pandas as pd
import numpy as np
import matplotlib.pyplot as mp
from sklearn.metrics import confusion_matrix, accuracy_score, \
    recall_score, precision_score, f1_score
from sklearn.model_selection import cross_val_score
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from sklearn.preprocessing import StandardScaler, normalize
from sklearn.model_selection import train_test_split
import seaborn as sb
from sklearn.svm import SVC
import scipy as sc
import warnings
warnings.filterwarnings('ignore')

[2]: os.chdir("C:/Users/John Of God/Desktop/Jobs/MBE")
df= pd.read_csv('data_mbela.csv', sep=';')

[3]: df=df.drop(['etab'], axis=1)

[4]: df['filière']=df['filière'].replace({'Chimie':'Autre', 'Production végétale':
    'Autre', 'Physique':'Autre',
    'Géologie':'Autre', 'Production animale':'Autre', 'Mathématiques':'Autre'})
df['matlycee']= df['matlycee'].replace({'Hist/Geo/Ecm':'Autre', 'Langue',
    'étrangère':'Autre', 'Comptabilité':'Autre'})
df['filpartut']=df['filpartut'].replace({'FSE':'Autre', 'Géologie':
    'Autre', 'Chimie':'Autre', 'Science biomédicale':'Autre', 'ESMV':'Autre',
    'Production animale':'Autre',
    'EGEM':'Autre', 'Science infirmière':'Autre', 'Production végétale':
    'Autre', 'RIM':'Autre'})
df['metpartut']= df['metpartut'].replace({'Etudiant':'Autre', 'Etudiant':
    'Autre', 'Cultivateur':'Autre', 'Pilote':'Autre' })
df['perfscient']= df['perfscient'].replace({'Faible':'Faible', 'Passable':
    'Faible', 'Assez-bien':'Moyen', 'Bien':'Elevé', 'Très bien':'Elevé' })
```

```

df['motiv']= df['motiv'].replace({"Moins d'efforts pour progresser":
    ↳'Autre','Encouragement des enseignants':'Autre'})

[5]: df['religion']=df['religion'].replace({'Autre religion':'Autre', 'Sans_
    ↳réligion':'Autre'})
df['serbac']=df['serbac'].replace({'Technique':'Technique ou_
    ↳autre','littéraire':'Technique ou autre'})
df['metpartut']=df['metpartut'].replace({'Retraité':'Autre', 'Autre':'Autre'})

[6]: ## Extraction des variables dont le test du Chi2 a montré une corrélation avec_
    ↳la variable dépendante
col_df= df[['filière', 'region', 'serbac','conseilor','stereo']]

[7]: ## Transformation des variables explicatives en facteur.

## Notons que les chiffres suivent l'ordre alphabétique. Ainsi la variable_
    ↳stéreo est codée en 0 pour Non et 1 pour Oui.
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
Onenco= OneHotEncoder()
Lenco= LabelEncoder()

for i in col_df:
    col_df[i]=Lenco.fit_transform(col_df[i])

[8]: ## Ajout de la variable dépendante dans la base opérationnelle

col_df['vardep']=df['vardep'].replace({'Scientifique':1, 'Technologique':0})

[9]: import statsmodels.api as sm
from statsmodels.api import Logit
from statsmodels.tools import add_constant

## Création de deux matrices X et Y avec X la matrice des variables_
    ↳explicatives et Y la colonne variable dépendante
X=col_df.drop(['vardep'], axis=1)
Y=col_df['vardep']

```



```
[10]: ## Affichage des résultats
```

```
print(res.summary())
```

```

                        Logit Regression Results
=====
Dep. Variable:          vardep      No. Observations:          203
Model:                Logit      Df Residuals:              197
Method:                MLE       Df Model:                  5
Date:                 Fri, 14 Jun 2024   Pseudo R-squ.:         0.07152
Time:                 07:40:39    Log-Likelihood:        -122.00
converged:             True      LL-Null:              -131.40
Covariance Type:      nonrobust    LLR p-value:           0.002099
=====
              coef      std err          z      P>|z|      [0.025      0.975]
-----
const          0.5784      0.479      1.207      0.227      -0.361      1.518
filière         0.0143      0.070      0.204      0.839      -0.124      0.152
region        -0.1391      0.047     -2.940      0.003      -0.232     -0.046
serbac        -0.3762      0.304     -1.239      0.215      -0.971      0.219
conseilor     -0.6525      0.338     -1.931      0.053     -1.315      0.010
stereo        -0.9176      0.498     -1.842      0.066     -1.894      0.059
=====
```

```
[11]: ## Matrice de confusion
```

```
print(res.pred_table())
```

```
[[112.  20.]
 [ 52.  19.]]
```

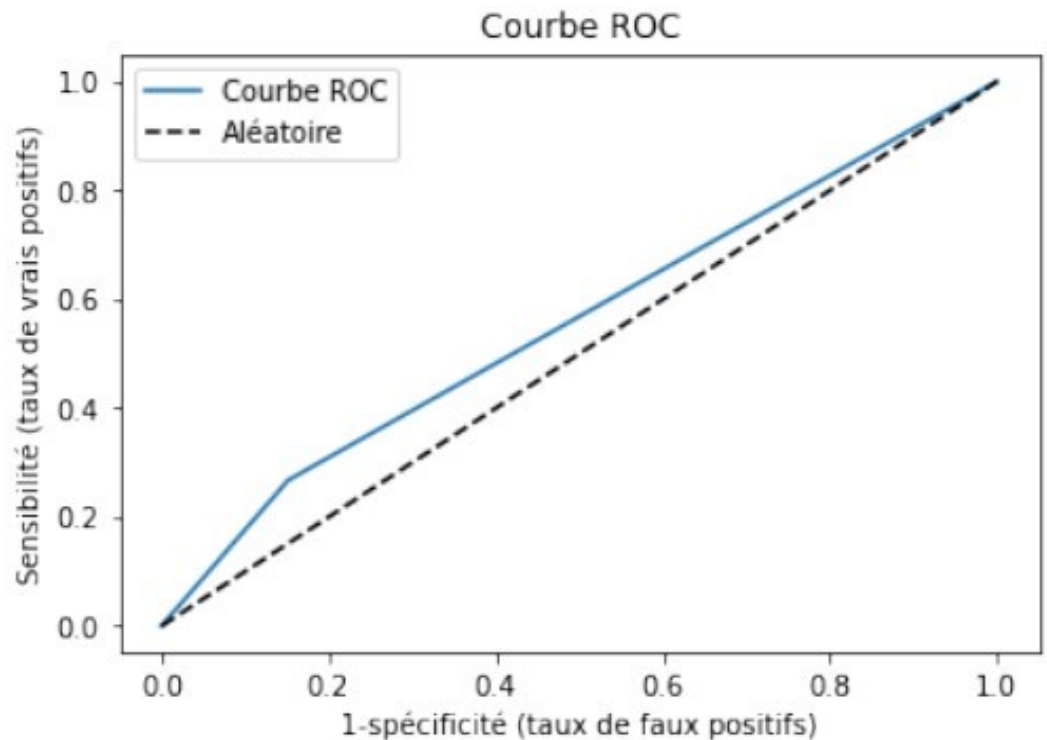
```
[28]: ## Construction de la courbe ROC
```

```
from sklearn.metrics import roc_curve, roc_auc_score
```

```
Y_pred= np.where(res.fittedvalues>0,1,0)
fpr, tpr, thresholds= roc_curve(Y, Y_pred)
```

```
## Tracé de la courbe
```

```
mp.plot(fpr, tpr, label="Courbe ROC")
mp.plot([0,1],[0,1], 'k--', label='Aléatoire')
mp.ylabel('Sensibilité (taux de vrais positifs)')
mp.xlabel('1-spécificité (taux de faux positifs)')
mp.title('Courbe ROC')
mp.legend()
mp.show()
```



[26]: `## Calcul de l'aire sous la courbe ROC`

```
auc= roc_auc_score(Y, Y_pred)
print(auc)
```

0.5580452411438327

[17]: `## Effets marginaux`

```
mareff=res.get_mareff(at='overall')
print(mareff.summary())
```

Logit Marginal Effects

```
=====
Dep. Variable:          vardep
Method:              dydx
At:                  overall
=====
```

	dy/dx	std err	z	P> z	[0.025	0.975]
filière	0.0030	0.015	0.204	0.839	-0.026	0.032
region	-0.0288	0.009	-3.193	0.001	-0.047	-0.011
serbac	-0.0779	0.062	-1.255	0.209	-0.200	0.044
conseilor	-0.1352	0.068	-1.996	0.046	-0.268	-0.002
stereo	-0.1902	0.100	-1.892	0.058	-0.387	0.007

