File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ ... / S8 / projet S8 /

Name	Last Modified
data_anonymous	a year ago
projetbdtnS8	a year ago
site web	a year ago
TextPython	a year ago
yapo	a year ago
C#_Python.zip	a year ago
data_anonymous.rar	a year ago
DataMachineLearning_semaine1_2...	a year ago
imagename.png	a year ago
MLRFID_Projet_S8 (methode analyt...	a year ago
MLRFID_Projet_S8_arbre_de_decisi...	a year ago
MLRFID_Projet_S8_comparaison.ip...	a year ago
MLRFID_Projet_S8_Random_Forest....	a year ago
MLRFID_Projet_S8_regression_logis...	a year ago
MLRFID_Projet_S8_SVM.ipynb	a year ago
MLRFID_Projet_S8.ipynb	a year ago
MLRFID_Projet_S8(étude statistiqu...	a year ago
PrésentationProjetTech2022.pdf	a year ago
RFID et Machine Learning.pdf	a year ago
s8.txt	a year ago
Tahar, machine learning & rfid.pdf	a year ago
Untitled.ipynb	a year ago
Untitled1.ipynb	a year ago

MLRFID_Projet_S8(étude stat | MLRFID_Projet_S8_Random_I

Code

Python 3 (ipykernel)

[56]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import datetime
```

[57]:
```
pathfile=r'data_anonymous'

# reflist: list of epc in each box
reflist=pd.DataFrame()
#
files=os.listdir(pathfile)
for file in files:
    print(file)
    if file.startswith('reflist_'):
        temp=pd.read_csv(os.path.join(pathfile,file),sep=',').reset_index(drop=True)[['Epc']]
        temp['refListId']=file.split('.')[0]
        reflist=reflist.append(temp)
reflist=reflist.rename(columns={'refListId':'refListId_actual'})
reflist['refListId_actual']=reflist['refListId_actual'].apply(lambda x:int(x[8:]))
Q_refListId_actual=reflist.groupby('refListId_actual')['Epc'].nunique().rename('Q refListId_actual').reset_index(drop=False)
reflist=pd.merge(reflist,Q_refListId_actual,on='refListId_actual',how='left')
reflist.head()
```

```
ano_APTags-TagsAutomatedProcess-2019-11-07-16_29_46.csv
ano_APTags-TagsAutomatedProcess-2019-11-08-10_55_25.csv
ano_supply-process.2019-11-07-CUT.csv
reflist_0.olpn
reflist_1.olpn
reflist_2.olpn
reflist_3.olpn
reflist_4.olpn
reflist_5.olpn
reflist_6.olpn
reflist_7.olpn
reflist_8.olpn
reflist_9.olpn
```

[57]:

	Epc	refListId_actual	Q refListId_actual
0	epc_90	0	16
1	epc_91	0	16
2	epc_92	0	16

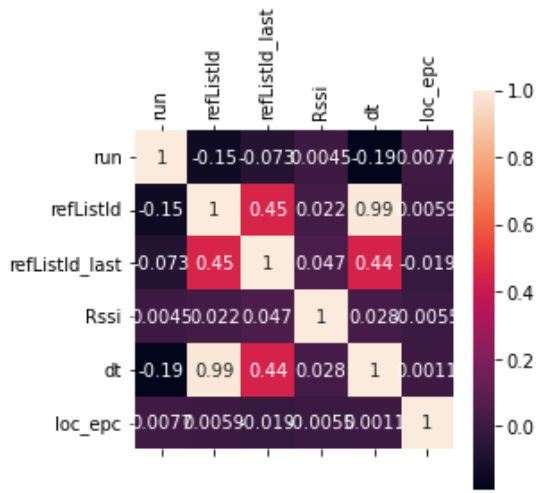
Simple 0 6 Python 3 (ipykernel) | Idle

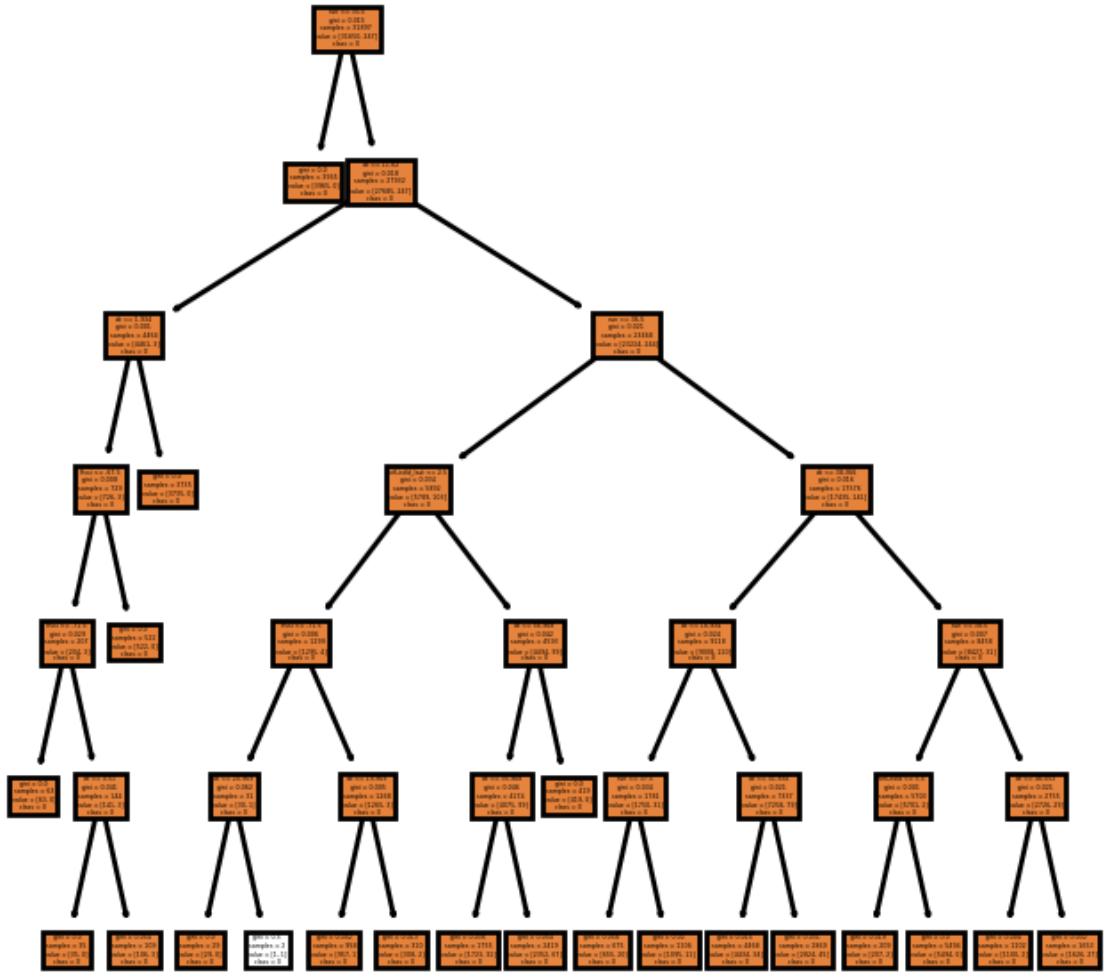
Mode: Command Ln 1, Col 1 MLRFID_Projet_S8_Random_Forest.ipynb

2.2) En réalisant une étude à l'aide d'une matrice de corrélation(nous avons maintenant la colonne loc_epc qui représente la position de l'epc)

[117]:
```
import seaborn as sb

corr = df_1.corr()
corrMat = plt.matshow(corr, fignum = 2)
plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
plt.yticks(range(len(corr.columns)), corr.columns)
sb.heatmap(corr,annot=True)
plt.show()
```





Evaluation du modèle(score)

[127]:
```
print(tree_clf.score(X_test, y_test))
```

```
0.9896551724137931
```

Matrice de confusion

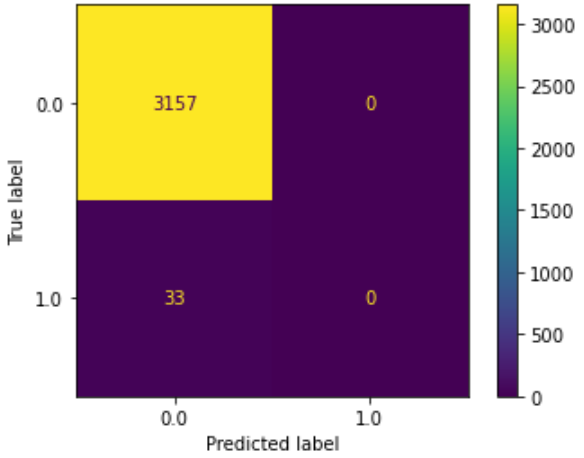
[128]:
```
y_pred = tree_clf.predict(X_test)
```

[129]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

[129]:
```
array([[3157,    0],
       [  33,    0]], dtype=int64)
```

[130]:
```
import matplotlib.pyplot as plot
from sklearn.datasets import make_classification
from sklearn.metrics import plot_confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
plot_confusion_matrix(tree_clf, X_test, y_test)
```

[130]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x29cdd916d60>
```



On créé un dictionnaire pour l'affichage sur l'interface

[131]:
```
mon_dictionnaire = {"score":tree_clf.score(X_test, y_test),"TN":confusion_matrix(y_test,y_pred)[0][0],"FP":confusion_matrix(y_test,y_pred)[0][1],"FN":confusion_matrix(y_test,y_pred)[1][0
```

[132]:
```
mon_dictionnaire
```

[132]:
```
{'score': 0.9896551724137931, 'TN': 3157, 'FP': 0, 'FN': 33, 'TP': 0}
```

df_timing_slices