Université des Sciences et de la Technologie Houari Boumediene

Faculté de Génie Electrique

Département Télécommunications

Master Réseaux et Télécommunications

MODULE : Programmation orienté objet

Compte rendu de TP 1 : Initiation et prise en main du langage Python (Modules : NumPy, matplotlib, Pandas, fichiers CSV, etc)

- SECTION : Réseaux et télécommunication B
- Sgrp : 04
- Presenté par:

1- BOUTKEDJIRET MED IKBAL 191931061466
2- LOUNES ELIAS 191931086878

- ***But de TP :***

Cette première séance de TP opte pour la familiarisation avec le langage de programmation Python. Nous abordons l'installation de Python, l'intégration des librairies, l'importation de modules, et la manipulation des fichiers.

➢ ***Manipulation :***

- Phase de prétraitement :

1. afficher l'ensemble de données et quelques infos générales sur les colonnes et les valeurs de données :

Tout d'abord on doit convertir notre fichier CSV en data frame à l'aide de la bibliothèque Pandas avec le code suivant

Le code :

- Si le fichier csv est sauvegardé dans le même fichier que python on utilise ce code-là :

```
import pandas as pd
df=pd.read_csv('titanic-passengers.csv',sep=';')
df
```

- Si le fichier CSV n'est pas sauvegardé sur le même fichier que python on utilise ce code pour tracer l'itinéraire d'emplacement de fichier sur le PC

```
import pandas as pd
df=pd.read_csv(r"C:\Users\Azur\OneDrive\Bureau\POO\Tp\TP1\titanic-passengers.csv",delimiter=';')
df
```

Le résultat :

On aura la data frame suivante :

Un tableau de 891 ligne et 12 colonne, on remarque que il y'a des cases remplis avec des valeurs (numérique ou pas) et d'autres cas remplis avec le mot NaN

[1]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	343	No	2	Collander, Mr. Erik Gustaf	male	28.0	0	0	248740	13.0000	NaN	S
1	76	No	3	Moen, Mr. Sigurd Hansen	male	25.0	0	0	348123	7.6500	F G73	S
2	641	No	3	Jensen, Mr. Hans Peder	male	20.0	0	0	350050	7.8542	NaN	S
3	568	No	3	Palsson, Mrs. Nils (Alma Cornelia Berglund)	female	29.0	0	4	349909	21.0750	NaN	S
4	672	No	1	Davidson, Mr. Thornton	male	31.0	1	0	F.C. 12750	52.0000	B71	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	10	Yes	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
887	61	No	3	Sirayanian, Mr. Orsen	male	22.0	0	0	2669	7.2292	NaN	C
888	535	No	3	Cacic, Miss. Marija	female	30.0	0	0	315084	8.6625	NaN	S
889	102	No	3	Petroff, Mr. Pastcho ("Pentcho")	male	NaN	0	0	349215	7.8958	NaN	S
890	428	Yes	2	Phillips, Miss. Kate Florence ("Mrs Kate Louis...	female	19.0	0	0	250655	26.0000	NaN	S

891 rows × 12 columns

2. Le prétraitement de nos données, recherche des valeurs manquantes et les remplacer par les valeurs appropriées :

- Tout d'abord on va chercher les valeurs manquant par l'instruction isnull

```
df.isnull()
```

Le résultat :

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	True	False
4	False	False	False	False	False	False	False	False	False	False	False	False

- On remarque qu'on a dans notre data frame des 'true' et 'false'
- True est lorsqu'on a des valeurs manquantes (True signifie qu'on a des NaN)
- False est lorsqu'on a des valeurs numériques.

Ensuite on va chercher combien de valeurs manquantes (NaN) on a, à l'aide de l'instruction suivante :

```
df.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

- L’instruction en haut nous montre le nombre de valeurs manquantes dans chaque ligne
- Par exemple on a 177 V. manquantes dans la colonne ‘Age’ et 687dans la colonne ‘Cabin’ et 2 dans la colonne ‘Embarked’

Ensuite on va chercher la valeur la plus apparente dans la colonne Cabin par exemple, à l’aide d’un compteur, et remplacer les valeurs manquantes avec cette valeur

Le Code :

```
import pandas as pd
df=pd.read_csv(r"C:\Users\Azur\OneDrive\Bureau\POO\Tp\TP1\titanic-passengers.csv",delimiter=';')
num=len(df['Cabin'])
df['Cabin'].value_counts()
df['Cabin'].fillna('G6',inplace=True)
df.tail()
```

- L’instruction **value_counts** sert à compter le nombre de fois que chaque valeur est répétée dans la colonne

```
Number of elements: 891
G6             4
B96 B98        4
C23 C25 C27    4
F33            3
D              3
              ..
C91            1
D45            1
F G63          1
A34            1
E63            1
Name: Cabin, Length: 147, dtype: int64
```

- Après utilisation de compteur on a trouvé que la valeur **G6** est la valeur la plus répétitif dans la colonne
- On a remplacé les valeurs manquantes dans la colonne par la valeur G6 avec l'instruction **fillna**

Voici le résultat :

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	10	Yes	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	G6	C
887	61	No	3	Sirayanian, Mr. Orsen	male	22.0	0	0	2669	7.2292	G6	C
888	535	No	3	Cacic, Miss. Marija	female	30.0	0	0	315084	8.6625	G6	S
889	102	No	3	Petroff, Mr. Pastcho ("Pentcho")	male	NaN	0	0	349215	7.8958	G6	S
890	428	Yes	2	Phillips, Miss. Kate Florence ("Mrs Kate Louis...	female	19.0	0	0	250655	26.0000	G6	S

- On remarque que les valeurs manquantes dans la colonne 'Cabin' ont étaient remplacées par la valeur G6

On suit on va supprimer les colonnes indésirables, celles qu'on n'a pas intérêts avec dans notre recherche en utilisant l'instruction **drop**

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
df=pd.read_csv(r"C:\Users\Azur\OneDrive\Bureau\POO\Tp\TP1\titanic-passengers.csv",delimiter=';')
df.drop('Cabin',axis=1,inplace=True)
df['Age'].fillna(df['Age'].median(),inplace=True)
df['Embarked'].fillna('G6',inplace=True)
df.isnull().sum()
df
```

- Avec cette instruction on a supprimé la colonne 'Cabin'
- Et on a remplis les valeurs manquantes dans la colonne Age et Embarked avec la valeur moyenne de ces dernières, avec l'instruction **median.**

[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	343	No	2	Collander, Mr. Erik Gustaf	male	28.0	0	0	248740	13.0000	S
1	76	No	3	Moen, Mr. Sigurd Hansen	male	25.0	0	0	348123	7.6500	S
2	641	No	3	Jensen, Mr. Hans Peder	male	20.0	0	0	350050	7.8542	S
3	568	No	3	Palsson, Mrs. Nils (Alma Cornelia Berglund)	female	29.0	0	4	349909	21.0750	S
4	672	No	1	Davidson, Mr. Thornton	male	31.0	1	0	F.C. 12750	52.0000	S
...	...	...	...	...	...	...	...	...	...	...	...
886	10	Yes	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	C
887	61	No	3	Sirayanian, Mr. Orsen	male	22.0	0	0	2669	7.2292	C
888	535	No	3	Cacic, Miss. Marija	female	30.0	0	0	315084	8.6625	S
889	102	No	3	Petroff, Mr. Pastcho ("Pentcho")	male	28.0	0	0	349215	7.8958	S
890	428	Yes	2	Phillips, Miss. Kate Florence ("Mrs Kate Louis...	female	19.0	0	0	250655	26.0000	S

891 rows × 11 columns

- Voici la nouvelle data frame après ma suppression et l'écrasement.

Visualisation :

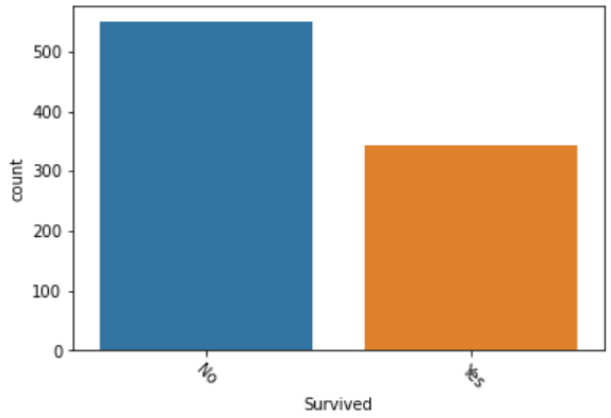
Il existe plusieurs bibliothèques sur python pour visualiser les graphes, tel que seaborn, matplotlib

- Pour visualiser les variables non numériques tels que (yes/no ou bien homme/femme) on utilise la bib seaborn, dans notre data frame on prend la colonne Survived comme exemple

[20]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
df=pd.read_csv(r"C:\Users\Azur\OneDrive\Bureau\POO\Tp\titanic-passengers.csv",delimiter=';')
sns.countplot(x='Survived',data=df)
plt.xticks(rotation=-45)
```

- On a importé la bibliothèque seaborn as sns et ensuite on a visualisé la colonne Survived

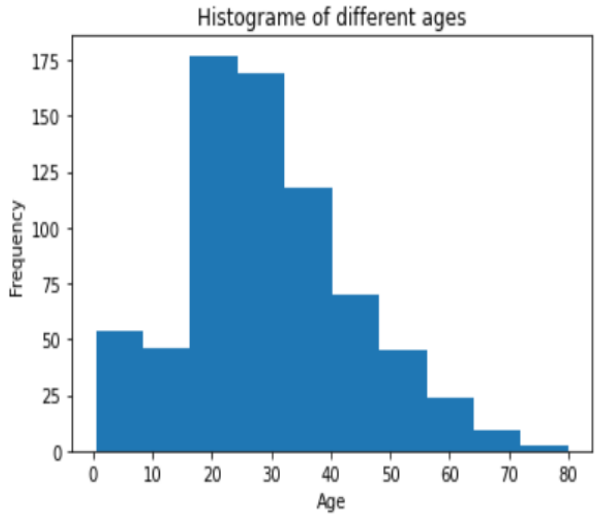
[20]: (array([0, 1]), [Text(0, 0, 'No'), Text(1, 0, 'Yes')])



Et pour visualiser les valeurs numériques on utilise la bib **matplotlib**, dans notre data frame on prend la colonne Age comme exemple

```
[17]: import pandas as pd
      import matplotlib.pyplot as plt
      import seaborn as sns
      import numpy as np
      df=pd.read_csv(r"C:\Users\Azur\OneDrive\Bureau\POO\Tp\titanic-passengers.csv",delimiter=';')
      plt.title('Histograme of different ages')
      plt.xlabel('Age')
      df['Age'].plot.hist()
```

```
[17]: <AxesSubplot:title={'center':'Histograme of different ages'}, xlabel='Age', ylabel='Frequency'>
```



- A l'aide de la bibliothèque matplotlib on a dessiné un histogramme avec l'instruction **plot.hist()**

La corrélation :

En utilisant le script donné à l'énoncé du TP on va étudier la corrélation entre les colonnes de notre data frame

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
df=pd.read_csv(r"C:\Users\Azur\OneDrive\Bureau\POO\Tp\TP1\titanic-passengers.csv",delimiter=';')
corr = df.corr()
s , ax = plt.subplots( figsize =( 12 , 10 ) )
cmap = sns.diverging_palette( 220 , 10 , as_cmap = True )
s = sns.heatmap(corr,cmap = cmap,square=True,cbar_kws={ 'shrink' : .9 },ax=ax,annot = True,annot_kws = { 'fontsize' : 12 })
df
```

- Avec l'instruction **.corr()** on étudie la corrélation entre les colonnes de notre df

- On remarque que la corrélation est optimale quand la valeur est égale à 1
- La corrélation est minimale lorsqu'elle tend vers 0
- La valeur de corrélation est comprise entre -1 et 1

➢ CONCLUSION :

Dans ce TP on a utilisé le programme PYTHON et ces bibliothèques pour faire des data frame à partir d'un fichier csv et ensuite manipuler cette data frame en trouvant les valeurs manquantes, appliquer l'encrassement et finalement visualiser les valeurs en diagramme

FIN.