MASTER 2 - IFRIS

TD3 – Partie A – Classification sur Python

Classification avec Python

1. Importez le fichier « iris_data.csv » dans un dataframe intitulé « iris »
2. Affichez les 5 premières lignes du dataframe « iris »
3. Combien le dataframe compte t-il de lignes et de colonnes ?
4. Ce dataframe décrit des fleurs de différentes espèces : setosa, versicolor, virginica
   Nous allons essayer de prédire l'espèce d'une plante à partir de ses caractéristiques.
   Quelles sont les variables explicatives ? Quelle est la variable cible ?
5. Séparez les variables explicatives dans un objet « X » et la variable cible dans un objet « y ». Vous pourrez utiliser le script suivant :

```
X = iris.loc[:,'sepal_length':'petal_width']
y = iris['species']
```

6. Nous allons à présent constituer un échantillon d'apprentissage et un échantillon de test pour notre modèle. Entrez les commandes suivantes :

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =train_test_split(X, y, test_size=0.35)
```

   Ces commandes permettent de constituer un échantillon d'apprentissage (X_train et y_train) ainsi qu'un échantillon de test (X_test et y_test)

7. Nous allons instancier un modèle de classification de type KNN en utilisant 3 comme nombre de voisins

```
from sklearn.neighbors import KNeighborsClassifier
#Instanciation du modèle
neigh = KNeighborsClassifier(n_neighbors=3)
```

8. Entrainez le modèle sur les données d'apprentissage

```
neigh.fit(X_train, y_train)
```

9. En utilisant le modèle entrainé, faites la prédiction sur les données de test, et stockez le résultat dans un objet « y_pred »

```
y_pred = neigh.predict(X_test)
```

10. Affichez le contenu de l'objet « y_pred »
11. En utilisant les prédictions et la réalité (y_test), affichez la matrice de confusion

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

12. Affichez l'efficacité du modèle.

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

13. Chargez les données « data_breast_cancer.csv » dans un dataframe. Ces données permettent de concevoir un modèle pour prédire la gravité d'une tumeur (variable diagnosis) à partir de ses caractéristiques.

14. En suivant les étapes nécessaires comme ci-dessus (échantillonnage, instanciation, évaluation), concevez deux modèles permettant de prédire la gravité d'une tumeur.
    Vous pourrez utiliser comme modèles : le KNN comme ci-dessus (https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html), et la régression logistique (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)