Lab2 : K-Nearest Neighbors

(KNN)

Réalisé par : Oussama LOUATI – Khouloud Taouchikht – Achraf Benomar

Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

dImporting the Dataset

To import the dataset and load it into our pandas dataframe, execute the following code:

```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']
# Read dataset to pandas dataframe
dataset = pd.read_csv(url, names=names)
```

1. Describe the features and the classes of the data set. You may use the dataset.head() command to see what the dataset actually looks like.

   Le Dataset contient des variables, et après la commande dataset.head() il représente maintenant quatre mesures pour chaque class qui sont en ensemble cinq.

2. What the argument « names » has been useful to ?

   L'argument « names » qui est une liste des noms était utile dans la représentation du dataset pout nommer chaque variable de ce dernier.

Preprocessing

Use the following code

```
X = dataset.iloc[:, :3].values
y = dataset.iloc[:, 4].values
# Or X = dataset.iloc[:, :-1].values and y = dataset.iloc[:, 4].values
```

1. What are the values of X and y variables ? what have you performed by the code?

   X et y représentent des séries des lignes et des colonnes du dataset.
   X représente les 3 premiers colonnes de toutes les lignes du data.
   y représente le quatrième colonnes de toutes les lignes du data.

   On a visualisé le data frame d'une manière plus claire à l'aide de ce code.

Train Test Split

We will divide our dataset into training and test splits, which gives us a better idea as to how our algorithm performed during the testing phase.

To create training and test splits, execute the following script:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

2. What are advantages of separating the train and test data dataset for the algorithm performance? (what happens if we do not separate train and test data)

- Pour comprendre les performances du modèle, en divisan l'ensemble de données en un ensemble d'apprentissage

- vous permettent de comparer les performances des algorithmes d'apprentissage automatique pour votre problème de modélisation prédictive.

3. What are the rates of this split ? how many rows does it correspond to?

X_train contient 120 lignes

X-test contient 30 lignes.

Le même chose pour y_train et y_test

Training rate = 80 %

Testing rate = 20 %

Feature Scaling

The following script performs feature scaling:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

4. Define the scaling in your words

Le Scaling est une technique permettant de normaliser les fonctionnalités indépendantes présentes dans les données dans une plage fixe. Il est effectué lors du prétraitement des données pour gérer des grandeurs ou des valeurs ou des unités très variables.

Training and Predictions

In the following we will train the KNN algorithm and make predictions with it.

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train, y_train)
 y_pred = classifier.predict(X_test)
```

1. What is the name of the KNN class and with which argument it is called?
   `y_pred = classifier.predict(X_test)`

   Le nom de la knn class est KNeighborsClassifier et elle prend en parametre l'argument n_neighbors (les 5 proches class).

2. What does the fit() function is used for ?

   La fonction fit est la mesure de la capacité d'un modèle d'apprentissage automatique à généraliser des données similaires à celles avec lesquelles il a été formé

3. What does the predict() function is used for ?

   La fonction predict () nous permet de prédire les étiquettes des valeurs de données sur la base du modèle formé.

Evaluating the Algorithm

For evaluating an algorithm, most commonly used metrics are computed in the following.

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

4. Explain the difference between y_test and y_pred.
   y_test = Les valeurs de référence.
   y_pred = Les valeurs prévus.

5. Provide the meaning of each computed measure.

   La précision peut être considérée comme une mesure de l'exactitude d'un classificateur. Pour chaque classe, il est défini comme le rapport des vrais positifs à la somme des vrais et des faux positifs.

   recall :Le rappel est une mesure de l'exhaustivité du classificateur ; la capacité d'un classificateur à trouver correctement toutes les instances positives. Pour chaque classe, il est défini comme le rapport des vrais positifs à la somme des vrais positifs et des faux négatifs.

   f1 score : Le score F1 est une moyenne harmonique pondérée de précision et de rappel telle que le meilleur score est de 1,0 et le pire est de 0,0. De manière générale, les scores F1 sont inférieurs aux mesures de précision car ils intègrent la précision et le rappel dans leur calcul. En règle générale, la moyenne pondérée de F1 doit être utilisée pour comparer les

modèles de classificateur, et non la précision globale.

Support : le nombre d'occurrences réelles de la classe dans l'ensemble de données spécifié. Un soutien déséquilibré dans les données de formation peut indiquer des faiblesses structurelles dans les scores rapportés du classificateur et pourrait indiquer la nécessité d'un échantillonnage stratifié ou d'un rééquilibrage. Le support ne change pas entre les modèles, mais diagnostique plutôt le processus d'évaluation.

accuracy :la précision globale du modèle (notez que la précision n'est pas une mesure relative à une certaine classe, mais une performance dans toutes les classes).

Comparing Error Rate with the K Value

First calculate the mean of error for all the predicted values where K ranges from 1 and 40. Execute the following script:

```
error = []
# Calculating error for K values between 1 and 40
for i in range(1, 40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, y_train)
    pred_i = knn.predict(X_test)
    error.append(np.mean(pred_i != y_test))
```

And then plot the error.

```
plt.figure(figsize=(12, 6))
plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
plt.title('Error Rate K Value')
plt.xlabel('K Value')
plt.ylabel('Mean Error')
```

1. Change the K range. Identify the best K interval. Justify.

- Les meilleurs intervalles du K sont : [9 , 11] ; [19 , 20] ; [22]

➔ Parce l'erreur moyen dans ces valeurs de K est minimale d'après le graphe .