Université Mohammed V de Rabat
Ecole Nationale Supérieure d'Arts et Métiers
Cycle Ingénieur Filière INDIA (3ème année) A.U. : 2023/2024 Pr. A. El Qadi

TP n4 en Big Data Analytics

Exercice 1 : Spark ML est une autre bibliothèque d'apprentissage automatique qui s'exécute au-dessus de Spark. Une tâche d'apprentissage automatique comprend les étapes suivantes :

1. Lire les données.	5. Entraîner un modèle avec un jeu de données d'entraînement.
2. Prétraiter ou préparer les données pour le traitement.	6. Ajuster le modèle à l'aide de techniques de validation croisée.
3. Extraire les fonctionnalités.	7. Évaluer le modèle sur un ensemble de données de test
4. Diviser les données pour la formation, la validation et les tests.	8. Déployer le modèle.

Chaque étape représente une étape dans un pipeline d'apprentissage automatique. Spark ML fournit des abstractions pour ces étapes. Par rapport à MLlib, il est plus facile d'assembler les étapes précédentes dans un pipeline d'apprentissage automatique en utilisant Spark ML.

1. Créer un RDD à partir du jeu de données iris.data (iris.txt) : val lignes = sc.textFile("d:/tpspark/iris.txt")
2. Un algorithme d'apprentissage automatique effectue plusieurs passages sur un ensemble de données, mettre en cache le RDD en mémoire.
3. Supprimer (filtrer) les lignes vides du jeu de données
4. Extraire les caractéristiques et les étiquettes. Les données sont au format CSV, diviser donc chaque ligne.
5. Les algorithmes MLlib fonctionnent sur RDD de LabeledPoint, Transformer l'analyse (parsed) en RDD de LabeledPoint. Cependant, les étiquettes sont stockées sous forme de chaînes alphabétiques, qui doivent être converties en étiquettes numériques.
6. Créer un RDD de LabeledPoint à partir de l'analyse (parsed)

```
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.mllib.linalg.{Vector, Vectors}
val labeledPoints = parsed.map{a =>
LabeledPoint(textToNumeric(a(4)),
   Vectors.dense(a(0).toDouble, a(1).toDouble, a(2).toDouble, a(3).toDouble))}
```

7. Diviser l'ensemble de données en données d'apprentissage et de test (20% pour les données de test).
8. Former le modèle NaiveBayes.

```
import org.apache.spark.mllib.classification.NaiveBayes
```

9. Evaluer le modèle sur le jeu de données de test. La première étape de l'évaluation d'un modèle consiste à lui faire prédire une étiquette pour chaque observation dans l'ensemble de données de test
10. Donner les valeurs de l'accuracy (taux de classification), et la matrice de confusion du modèle (utiliser la classe MulticlassMetrics)

```
import org.apache.spark.mllib.evaluation.MulticlassMetrics
val metrics = new MulticlassMetrics(predictionsAndLabels)
```

Exercice 2 : Créer les données suivantes en tant que logdata.log avec des délimiteurs virgules, comme indiqué. Le schéma de ces données est : Heure, Adresse IP, URL, Emplacement

```
10:24:25,10.192.123.23,http://www.google.com/searchString,ODC1
10:24:21,10.123.103.23,http;//www.amazon.com,ODC1
10:24:21,10.112.123.23,http;//www.amazon.com/Electronies,ODC1
10:24:21,10.124.123.24, http://www.amazon.com/Electronies/storagedevices,ODC1
10:24:22,10.122.123.23,http;//www.gmall.com,ODC2
10:24:23,10.122.143.21,http://www.flipkart.com/ODC2
10:24:21,10.124.123.23,http://www.flipkart.com/offers,ODC1
```

1. Créer un DataFrame du fichier journal créé (à l'aide de spark.read.csv).
2. Créer une vue temporaire nommée 'LogData'.
3. Créer une vue temporaire globale nommée 'LogData_Global'.
   Observer la différence entre la vue temporaire et la vue temporaire globale en exécutant la requête avec une vue temporaire dans une autre session Spark.
4. Écrire et exécuter des requêtes SQL pour les exigences suivantes.
   a. Combien de personnes ont accédé au domaine Flipkart dans chaque emplacement ?
   b. Qui a accédé au domaine Flipkart dans chaque emplacement ?
   c. Indiquer leur adresse IP.
   d. Combien d'utilisateurs Internet distincts sont disponibles dans chaque emplacement ?
   e. Dresser la liste des emplacements uniques disponibles.