Prueba tecnica

Jose Antonio Rodriguez Rivera

2023-06-23

1. Importamos las bibliotecas necesarias: `SparkR` y `SparkRML`. Estas bibliotecas nos permiten trabajar con SparkR y el algoritmo de recomendación ALS (Alternating Least Squares).

2. Iniciamos una sesión de Spark utilizando `sparkR.session()`. Esto establece una conexión con el clúster de Spark y nos permite ejecutar comandos de SparkR.

3. Cargamos los datos del archivo 'ratings.csv' en un DataFrame utilizando `read.df()`. Asegúrate de que el archivo 'ratings.csv' se encuentre en el directorio correcto. Especificamos que el archivo es de formato CSV y que tiene un encabezado y las columnas deben inferirse automáticamente.

4. Convertimos las columnas del DataFrame al formato correcto utilizando `withColumn()` y `cast()`. En este caso, convertimos la columna 'userId' a tipo de datos entero, la columna 'movieId' a tipo de datos entero y la columna 'rating' a tipo de datos de punto flotante. Esto asegura que los datos estén en el formato adecuado para el modelo ALS.

5. Creamos un objeto ALS (Alternating Least Squares) utilizando `ml_als()`. Configuramos los parámetros del modelo, como el número máximo de iteraciones (`maxIter`), el parámetro de regularización (`regParam`), las columnas de usuario (`userCol`), película (`itemCol`) y calificación (`ratingCol`), y la estrategia para manejar datos en frío (`coldStartStrategy`). El modelo ALS es un algoritmo de factorización de matrices utilizado comúnmente en sistemas de recomendación.

6. Ajustamos el modelo ALS al conjunto de datos utilizando `ml_fit()`. Esto entrena el modelo utilizando los datos de entrada y encuentra los factores latentes que representan las preferencias de los usuarios y las características de las películas.

7. Creamos un evaluador de regresión utilizando `ml_regression_evaluator()`. Configuramos el nombre de la métrica a utilizar, en este caso, "rmse" (Root Mean Square Error o error cuadrático medio), y especificamos las columnas de la calificación real (`labelCol`) y la calificación predicha (`predictionCol`). El evaluador se utilizará para medir el rendimiento del modelo en base a la precisión de las predicciones.

8. Realizamos predicciones utilizando el modelo ajustado y los datos de entrada utilizando `ml_transform()`. Esto genera predicciones de calificación para cada usuario y película en el conjunto de datos.

9. Evaluamos el modelo utilizando el evaluador creado anteriormente y las predicciones generadas utilizando `ml_evaluate()`. Esto calcula el error cuadrático medio (RMSE) entre las calificaciones reales y las calificaciones predichas. El resultado nos proporciona una medida del rendimiento del modelo en términos de precisión de las predicciones.

10. Imprimimos el valor del error cuadrático medio utilizando `cat()`. Esto muestra en la consola el resultado del RMSE.

11. Generamos las 10 mejores recomendaciones de películas para cada usuario utilizando `ml_recommend_for_all_users()`. Esto utiliza el modelo entrenado para calcular las mejores recomendaciones de películas para cada usuario en base a sus preferencias y las características de las películas.

12. Mostramos las primeras filas de las recomendaciones utilizando `head()`. Esto nos permite ver un vistazo de las recomendaciones generadas por el modelo.

Asegúrate de tener el archivo 'ratings.csv' en el directorio correcto antes de ejecutar el código.