PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE | la clase ejecutiva

Instrucciones Miniproyecto 2

Descripción

Usará funciones de Spark y Spark SQL sobre datos en formato CSV, correspondiente a datos bancarios de clientes, donde se tiene una columna de si el cliente contrató o no un crédito. Usarás funciones de Spark y Spark SQL para realizar análisis sobre los datos.

Se usará un notebook de Google Colab como herramienta de ejecución de código y como informe de entrega del miniproyecto, por lo que se requiere que se cumpla el siguiente formato de entrega:

1. Debes nombrar el notebook como *<apellido>_<nombre>.ipynb*
2. Cada parte del miniproyecto debe estar contenido en una celda, y el output que se pide para la evaluación debe estar explícitamente impreso en el notebook.

El trabajo a realizar

1. [Celda 1] Usando el notebook *practica spark sql.,ipynb* como base, importe e instale Hadoop y Spark.
2. [Celda 2] Usando un *snippet* de *upload*, cargue el archivo *bank-additional-full.csv* que lo usará de input de datos.
3. [Celda 3] Cree una sesión de Spark SQL e importe los datos del CSV. Imprima en pantalla una muestra de las 5 primeras columnas y el número de registros del archivo CSV.
4. [Celda 4] Cree un Dataframe de nombre *simpleDF_yes* que contenga las columnas [“age”, “job”, “education”, “housing”] para aquellos donde la columna “y” tiene valor “yes” (que contrató un crédito), y un Dataframe *simpleDF_no* con las mismas columnas que *simpleDF_yes* pero para aquellos con el valor de “y” en “no”. Muestre las primeros 10 registros de ambos Dataframes.
5. [Celda 5] Usando funciones de Spark, y usando matplotlib.pyplot como herramienta de visualización, genere histogramas de cada columna de los Dataframes anteriores. Para la columna “age”, use intervalos de 10 años. Muestre histogramas con la función matplotlib.pyplot.bar, uno para cada columna y Dataframe. Use el ejemplo de *ejemplos_matplotlib.py* para obtener orientación en el manejo de los gráficos.
6. [Celda 6] Usando funciones de Spark SQL, cree una tabla temporal llamada *countsDF* sobre el Dataframe *simpleDF_yes*, y realiza un query SQL que agrupe los datos de acuerdo a [“job”, “education”] y que cuente la aparición de cada caso dentro de la

tabla, ordenando el conteo en forma descendente. Muestra los primeros 20 registros de la tabla resultante.

7. [Celda 7] Guarda *countsDF*, *simpleDF_yes* y *simpleDF_no* en HDFS, en formato JSON. Puedes usar la función <Dataframe>.format('json').save(<path/archivo.json>). Muestra la ruta de HDFS con los archivos (hint: usar !hdfs dfs -ls <path>).
8. [Celda 8] Usando funciones de Spark y RDD, realice el mismo procedimiento que la [Celda 6], programando todos los pasos sin usar Dataframe, desde el input del archivo usando *sc.textFile*. Imprima en pantalla los primeros 20 registros del RDD resultante.