Spark_Python_in_Databricks

September 6, 2023

0.0.1 AIT 614 - Big Data Essentials

Lab 4: Spark with Python in Databricks Purpose for helping students to learn PySpark for Data Science in Databricks

Creatd by Dr. Liao

Please type into your course section # and your full name:

Course Section #: 005 Student's Full Name: Rashmika Calve

Please follow Dr. Liao's code examples/tutorials to complete these tasks:

Load a data file

[0]:
```
df1 = spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/
↪shared_uploads/rcalve@gmu.edu/EmployeeAttrition.csv")
```

(5 points) 1. Count the employees whose TotalWorkingYears are greater than 20.

[0]:
```
df1.filter(df1.TotalWorkingYears > 20).count()
```

```
Out[14]: 207
```

(15 points) 2. Find EmployeeNumber, EducationField, JobRole for all the employees whose Age is between 25 and 30 and Education is 5. Use display() to display EmployeeNumber, EducationField, and JobRobe only.

[0]:
```
df_2 = df1.filter((df1.Age.between(25,30)) & (df1.Education == 5)).
↪select('EmployeeNumber', 'EducationField', 'JobRole')
display(df_2)
```

(15 points) 3. For all the women employees having Age between 35 and 40 and TotalWorkingYears < 5, sort EmployeeNumber in an ascending order. Use display() to show EmployeeNumber and Department in the output.

[0]:
```
df_3 = df1.filter((df1.Age.between(35,40)) & (df1.TotalWorkingYears < 5) & (df1.
↪Gender == "Female")).select('EmployeeNumber', 'Department').sort(df1.
↪EmployeeNumber.cast('int').asc())
# we are changing the datatype of Employee Number from string to integer with␣
↪the
#help of cast()
```

```
display(df_3)
```

(15 points) 4. Find employees whose HourlyRate is greater than 100 or DailyRate is greater than 1490. Display Age, HourlyRate, DailyRate, and Department only and sort DailyRate in a descending order.

[0]:
```
df_4 = df1.filter((df1.HourlyRate > 100) | (df1.DailyRate > 1490)).
 ↪select('Age', 'HourlyRate', 'DailyRate', 'Department').sort('DailyRate',␣
 ↪ascending = False)
display(df_4)
```

(20 points) 5. For each JobRole, find the average MonthlyIncome. Print out the formatted monthly incomes in hundredth and arrange them in descending order?

[0]:
```
from pyspark.sql.functions import avg, round
df_5 = df1.groupBy('JobRole').agg(round(avg('MonthlyIncome'),2).
 ↪alias('Avg_Monthly_Income')).sort('Avg_Monthly_Income', ascending = False)
display(df_5)
```

[0]:
```
# Displaying the output as a bar chart
df_5.display()
```

```
Output can only be rendered in Databricks
```

(20 points) 6. Count the different MaritalStatus when Attrition is Yes and Age is greater than 35 in the dataset. Arrange the count in descending order.

[0]:
```
df_6 = df1.filter((df1.Attrition == 'Yes') & (df1.Age > 35)).
 ↪groupBy('MaritalStatus').count().sort("count", ascending = False)
display(df_6)
```

[0]:
```
#Displaying the output as a pie chart
df_6.display()
```

```
Output can only be rendered in Databricks
```

(10 points) References:

1) Dr. Liao's Code Examples and Tutorials Series - Spark with Python for Data Queries and Basic Analysis in Databricks (Blackboard)
2) Databricks Visualizations : https://docs.databricks.com/notebooks/visualizations/index.html
3) PySpark Dataframe : https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.sql.DataFran
4) PySpark : https://spark.apache.org/docs/2.4.0/api/python/pyspark.html

[0]:
```
```