“Assignment”

Introduction to PySpark in Databricks: Exploratory Data Analysis (EDA) Lab

Welcome to your first lab using PySpark in the Databricks environment! In this lab, you will learn how to conduct Exploratory Data Analysis (EDA) using PySpark. The focus will be on basic PySpark functions, understanding the Spark DataFrame API, and getting hands-on with common EDA tasks.

We’ll use Databricks, a popular cloud-based platform for Spark. In this lab, you will:

1. **Set up your Databricks workspace**
2. **Load a dataset into Spark**
3. **Explore the dataset using basic PySpark functions**
4. **Perform exploratory data analysis (EDA) using PySpark**
5. **Visualize data using built-in functions and libraries**

Step 1: Setting Up Your Databricks Workspace

1. **Log in to Databricks Community Edition** (if you don’t have an account, create one at databricks.comLinks to an external site.)
2. **Create a new notebook**:
    - Once logged in, go to the **Workspace** section and create a new notebook.
    - Give your notebook a meaningful name, for example: **"ALY 6110 PySpark Lab"**.
    - Set the language to **Python**.
    - (If Needed) Create a new Resource and select the default cluster. Wait for it to attach.
3. **Question:** What are the advantages of using a cloud-based platform like Databricks for working with Spark?
4. **Question:** How does Spark's cluster computing model differ from traditional single-machine data processing?

Step 2: Setting Up Spark Session

Before performing any analysis, you need to create a Spark session. This will allow you to run Spark code in the Databricks environment. This step is usually automatic in Databricks, but we will initialize it for clarity.

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("ALY6110_EDA_Lab").getOrCreate()
```

Step 3: Loading the Dataset

For this lab, we'll use the a population and housing sales **dataset.**

```
     # Load sample data

     df = spark.read.csv("/databricks-datasets/samples/population-vs-price/da
ta_geo.csv", header=True, inferSchema=True)

     df.show()
```

Question: What are the key differences between a Spark DataFrame and a Pandas DataFrame in Python?

Step 4:Basic Cleaning of the Dataset

```
df = df.dropna()

# Rename the columns

df = df.withColumnRenamed("2014 Population estimate", "Population").withColum
nRenamed("2015 median sales price", "Sales Price")

# Show the DataFrame with renamed columns

df.show()
```

Question: In the above code snipped, we remove all rows that are missing data. Explain the challenges of this decisions in a big data set.

Step 5: Perform Basic Data Operations:

Perform basic data operations such as filtering, grouping, and aggregating data using Spark DataFrames.

```
df.printSchema()
```

```
# Filter data

filtered_df = df.filter(df['State'] == 'California')

filtered_df.show()

from pyspark.sql.functions import sum

total_population = df.agg(sum("Population").alias("Total_Population"))

total_population.show()
```

```
from pyspark.sql.functions import round, avg, col,sum

# Group and aggregate data

grouped_df = df.groupBy("State").agg(

    round(avg(col("Sales Price")), 2).alias("Average_Sales"),

    sum(col("Population")).alias("Population")

)

grouped_df.show()
```

Question: How do these operations in Spark compare to similar operations in Python or R?

```
from pyspark.sql.functions import round

# Add a new column

transformed_df = df.withColumn("Price_Squared", df["Sales Price"] * df["Sales Price"])

# Assuming you want to round to 2 decimal places

transformed_df =transformed_df.withColumn("Price_Squared", round(col("Price_Squared"), 2))
```

```
transformed_df.show()
```

Question: What are the benefits of using Spark for large-scale data transformations?

```
# Describe the data
df.describe().show()

# Calculate correlation
correlation = df.stat.corr("Population", "Sales Price")
print(f"Correlation between Population and Price: {correlation}")
```

Step 6:VIsualizing the Data

Use Python libraries like matplotlib or seaborn to visualize data processed with Spark.

```
import matplotlib.pyplot as plt
# Convert Spark DataFrame to Pandas DataFrame for visualization
pandas_df = df.toPandas()
# Plot data
plt.scatter(pandas_df['Population'], pandas_df['Sales Price'])
plt.xlabel('2014 Population estimate')
plt.ylabel('2015 median sales price')
plt.title('Population vs Price')
plt.show()
```

Question: How can visualizing data processed with Spark help in understanding the data better?

Step 7:Introduction to SparkML

Load the taxi dataset into Spark.

```
df = spark.read.table("samples.nyctaxi.trips")

df.show(5)
```

Select the the columns to build our model.

```
data = df.select("trip_distance", "fare_amount")
```

Prepare the data

```
from pyspark.ml.feature import VectorAssembler

assembler = VectorAssembler(inputCols=["trip_distance", "fare_amount"], outputCol="features")

feature_data = assembler.transform(data)
```

Create the model

```
from pyspark.ml.clustering import KMeans

kmeans = KMeans(k=3, seed=1)  # Adjust the number of clusters (k) as needed

model = kmeans.fit(feature_data)
```

Examine the predictions

```
predictions = model.transform(feature_data)

predictions.select("trip_distance", "fare_amount", "prediction").show()
```

Evaluate the model

```
from pyspark.ml.evaluation import ClusteringEvaluator

evaluator = ClusteringEvaluator()

silhouette = evaluator.evaluate(predictions)

print(f"Silhouette with squared euclidean distance = {silhouette}")
```

Visualize the results

```
import pandas as pd

import matplotlib.pyplot as plt

pandas_df = predictions.select("trip_distance", "fare_amount", "prediction").toPandas()

plt.figure(figsize=(10, 6))

plt.scatter(pandas_df["trip_distance"], pandas_df["fare_amount"], c=pandas_df["prediction"], cmap="viridis", marker="o")

plt.xlabel("Trip Distance")

plt.ylabel("Fare Amount")

plt.title("Cluster Analysis of NYC Taxi Trips")

plt.colorbar(label="Cluster")

plt.show()
```

Question: How does building a machine learning model in SparkML compare to using traditional machine learning libraries like scikit-learn in Python?

Reflection

Question: Write a brief reflection on your experience using Spark. What did you find most challenging? What did you find most interesting?