Source code

Project**: Sales Data Analysis using Apache Spark on Azure**

Author: Pavan

Description:

This project reads sales data from Azure Blob Storage,

performs data cleaning, and generates business insights

using Apache Spark (PySpark) on Azure Databricks.

STEP 1: Configure Blob Storage Access

```
# NOTE:

# Do NOT hardcode real keys in GitHub.

# Paste the key only while running in Databricks.


storage_account_name = "salessparkstorage123"

storage_account_key = "<YOUR_STORAGE_ACCOUNT_KEY>"


spark.conf.set(

f"fs.azure.account.key.{storage_account_name}.blob.core.windows.net",

storage_account_key

)



# STEP 2: Load Sales Data from Azure Blob Storage

file_path = "wasbs://salesdata@salessparkstorage123.blob.core.windows.net/sales_data.csv"
```

```
df = spark.read.format("csv") \
.option("header", "true") \
.option("inferSchema", "true") \
.load(file_path)

print("Sample Data:")
df.show(5)


# STEP 3: Inspect Schema
print("Schema:")
df.printSchema()


# STEP 4: Data Cleaning
# Remove rows with null values
df_clean = df.dropna()

from pyspark.sql.functions import to_date

# Convert order_date column to Date type
df_clean = df_clean.withColumn(
"order_date",
```

```
to_date("order_date", "yyyy-MM-dd")
)

print("Cleaned Data:")
df_clean.show(5)


# STEP 5: Total Sales Analysis

from pyspark.sql.functions import sum

df_clean.select(
sum("price").alias("Total_Sales")
).show()


# STEP 6: Region-wise Sales Analysis
df_clean.groupBy("region") \
.sum("price") \
.withColumnRenamed("sum(price)", "Total_Sales") \
.orderBy("Total_Sales", ascending=False) \
.show()
```

```
# STEP 7: Top Selling Products
df_clean.groupBy("product") \
.sum("price") \
.withColumnRenamed("sum(price)", "Total_Sales") \
.orderBy("Total_Sales", ascending=False) \
.show(5)



# STEP 8: Monthly Sales Trend
from pyspark.sql.functions import month

df_clean.withColumn("Month", month("order_date")) \
.groupBy("Month") \
.sum("price") \
.withColumnRenamed("sum(price)", "Monthly_Sales") \
.orderBy("Month") \
.show()



# STEP 9: Visualization (Databricks)

# Databricks automatically provides charts
display(
df_clean.groupBy("region").sum("price")
```

)

\# END OF PROJECT

print("Sales Data Analysis Completed Successfully")

OUTPUT: