Day 10 - Datasets Analysis with PySpark

📅 Date: 18th June 2025

# Objective

To apply real-world data engineering skills on a retail dataset (Walmart sales data), focusing on data ingestion, transformation, exploration, and aggregation using PySpark.

# Dataset Overview: Walmart Sales Data

The dataset likely contains:  
- Store-wise sales records  
- Weekly sales figures  
- Promotional markdowns  
- Holiday flags  
- Temperature, Fuel Price, CPI, and Unemployment rates  
  
These fields simulate real-world challenges in retail analytics.

# Step 1: Data Loading

df = spark.read.csv("/FileStore/walmart\_sales.csv", header=True, inferSchema=True)  
df.show(5)  
df.printSchema()

# Step 2: Exploratory Data Analysis (EDA)

Basic Stats:  
df.describe().show()  
df.select("Weekly\_Sales").summary().show()  
  
Count of Holidays:  
df.groupBy("Holiday\_Flag").count().show()  
  
Number of stores:  
df.select("Store").distinct().count()

# Step 3: Data Cleaning

Null Handling:  
df.na.drop() or df.na.fill(value)  
  
Duplicate Removal:  
df.dropDuplicates()  
  
Type Casting:  
df = df.withColumn("Weekly\_Sales", df["Weekly\_Sales"].cast("double"))

# Step 4: Business Questions Solved

1. Top Performing Store:  
df.groupBy("Store").agg({"Weekly\_Sales": "sum"}).orderBy("sum(Weekly\_Sales)", ascending=False).show(1)  
  
2. Weekly Trends:  
df.groupBy("Date").agg({"Weekly\_Sales": "sum"}).orderBy("Date").show()  
  
3. Holiday vs Non-Holiday Sales:  
df.groupBy("Holiday\_Flag").agg({"Weekly\_Sales": "avg"}).show()  
  
4. Effect of Unemployment on Sales:  
df.select("Unemployment", "Weekly\_Sales").orderBy("Unemployment").show()

# Step 5: Advanced Aggregations

from pyspark.sql.functions import sum, avg, max  
  
df.groupBy("Store").agg(  
 sum("Weekly\_Sales").alias("Total\_Sales"),  
 avg("Weekly\_Sales").alias("Average\_Sales"),  
 max("Weekly\_Sales").alias("Max\_Sale")  
).show()

# Step 6: Save Processed Data

df.write.mode("overwrite").parquet("/FileStore/processed/walmart\_summary")

# Insights

- Spark is highly efficient for large datasets, such as Walmart's.  
- Retail datasets often require multi-column groupings and filters.  
- In-memory operations using DataFrames accelerate complex aggregations.  
- DataFrames + Spark SQL offer a powerful combo for data engineers.

# Summary

- Practiced real-world analytics using Walmart retail data.  
- Performed end-to-end data engineering: ingestion → cleaning → transformation → analysis.  
- Applied PySpark’s DataFrame API to answer business-critical questions.  
- Learned how to clean, analyze, and export large-scale sales data in Spark.