Big Data Residency Weekend Group Project

Satyanarayana Reddy Muttana

Sushila Sipai

Yanjie Liu

2025 Summer - Big Data (MSDS-632-M51) - Full Term

Dr. Moody Amakobe

06/15/2025

**Prerequisites to Install Apache Spark:**

1. Java

2. Python

3, Winutils

4. Spark

**1. Java (installed Java 11)**

1. Apache Spark is written in Scala, which runs on the Java Virtual Machine (JVM).
2. Spark needs Java to compile, run its engine, and manage RDDs, Data Frames, etc.
3. Without Java, you cannot start the Spark Shell or run any Spark job.
4. Minimum requirement: Java 8 or higher (Java 11 is compatible with Spark 3.5.6).

**2. Spark (installed Spark 3.5.6)**

1. This is the main framework you are using.
2. Installing Spark provides the binaries and scripts to launch spark-shell, pyspark, and run jobs locally or on a cluster.

**3. Python (installed Python 3.13.4)**

1. Spark Shell has multiple entry points:
2. spark-shell → Scala REPL
3. pyspark → Python REPL
4. Python is needed only if you're using pyspark (Python API for Spark).
5. If you're using just spark-shell (Scala-based), Python is not mandatory.

**4. Winutils (Windows only)**

1. Spark was originally designed for Linux-based environments (like Hadoop clusters).
2. On Windows, Spark expects certain Hadoop-related native libraries and file system utilities.
3. winutils.exe acts as a dummy implementation to let Spark run without throwing permission/file system errors on Windows.

Use case: Without winutils.exe, you might see errors like:

java.io.IOException: Could not locate executable null\bin\winutils.exe in the Hadoop binaries.

Note: After installing Java, Python, Spark, and Winutils, make sure to set the corresponding environment variables (JAVA\_HOME, SPARK\_HOME, HADOOP\_HOME) and update the system PATH to include their /bin directories.

This ensures Spark can run correctly from the command line using spark-shell or pyspark.

Apache Spark on Windows – Required Components Overview:

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Purpose | Is It Mandatory? | Notes |
| Java (e.g., Java 11) | Runs the JVM-based Spark engine (Scala/Java) | Yes | Spark is built on Scala, which runs on the JVM |
| Spark (e.g., Spark 3.5.6) | The actual big data engine; provides spark-shell and pyspark | Yes | Needed to run Spark applications |
| Python (e.g., Python 3.13.4) | Required only for using PySpark (Python API) | No (for spark-shell), Yes (for pyspark) | Use the correct Python version for compatibility with Spark |
| Winutils | Mimics Hadoop file system utilities on Windows | Yes (on Windows only) | Required to avoid Hadoop-related errors in Spark |

Successful Launch of Apache Spark Shell on Windows:

A computer screen with white text

AI-generated content may be incorrect.

**Dataset Selection:**

**Dataset Name**: Online Retail Dataset

**Source:** UCI Machine Learning Repository – Online Retail Dataset(https://archive.ics.uci.edu/ml/datasets/online+retail)

**File Format:** .csv

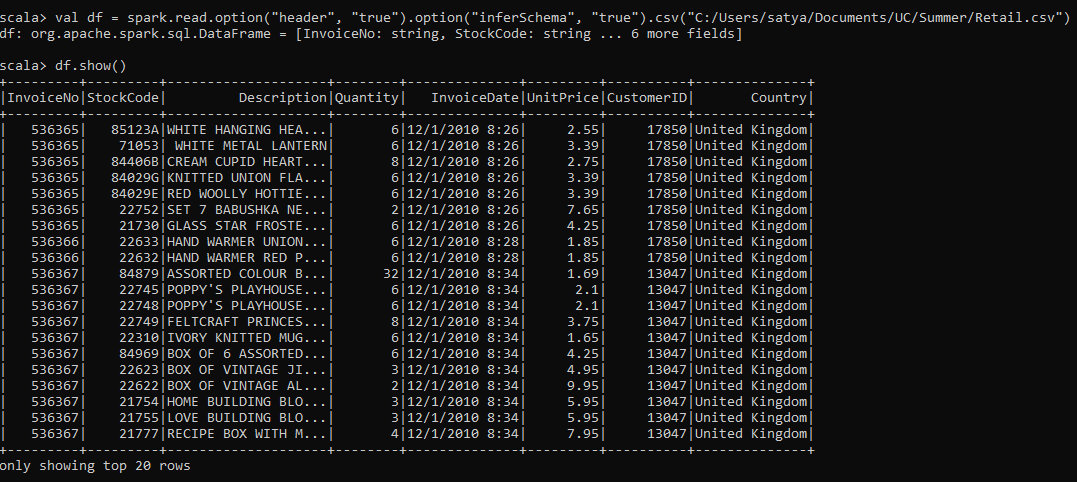
Selected From: Public dataset provided by UCI ML Repository, often used for retail analytics, customer segmentation, and RFM analysis.

**Description:**

The Online Retail dataset contains transactional data from a UK-based and registered non-store online retail company. The company mainly sells unique all-occasion gifts. The data includes information such as invoice numbers, stock codes, product descriptions, quantities, invoice dates, unit prices, customer IDs, and country of origin. This dataset is widely used for exploring real-world e-commerce data for data preprocessing, clustering, and association rule mining tasks.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| InvoiceNo | Unique identifier for each transaction |
| StockCode | Product/item code |
| Description | Name of the product |
| Quantity | Number of items purchased |
| InvoiceDate | Date and time of the transaction |
| UnitPrice | Price per unit in GBP |
| CustomerID | Unique customer identifier |
| Country | Country of the customer |

The below displayed DataFrame shows the first 20 rows of the Online Retail dataset loaded in Spark. It contains transactional data including invoice numbers, product descriptions, quantities, and timestamps, with most sales originating from the United Kingdom.



Meaningful results based on analytical queries:

**Top 10 Selling Products**

df.groupBy("Description").sum("Quantity").orderBy(org.apache.spark.sql.functions.desc("sum(Quantity)")).show(10)

A computer screen with text on it

AI-generated content may be incorrect.

2.Revenue per Country

val revenueDF = df.withColumn("Revenue", $"Quantity" \* $"UnitPrice")

revenueDF.show()

A screen shot of a computer screen

AI-generated content may be incorrect.

1. Sales Over Time (By Month):

import org.apache.spark.sql.functions.\_

spark.conf.set("spark.sql.legacy.timeParserPolicy", "LEGACY")

val df = spark.read.option("header", "true").csv("Retail.csv")

val dfWithDate = df.withColumn("InvoiceTimestamp", to\_timestamp(col("InvoiceDate"), "M/d/yyyy H:mm"))

val salesByMonth = dfWithDate.groupBy(month(col("InvoiceTimestamp")).alias("Month")).agg(sum("Quantity").alias("TotalQuantity")).orderBy("Month")

salesByMonth.show()

A computer screen with text on it

AI-generated content may be incorrect.

4. Top Customers by Revenue

revenueDF.groupBy("CustomerID").agg(round(sum("Revenue"), 2).as("TotalSpent")).orderBy(desc("TotalSpent")).show(10)

A black screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A graph showing a line of blue lines

AI-generated content may be incorrect.

A computer code with text

AI-generated content may be incorrect.

A graph of blue bars

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

A graph with blue and white lines

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.