Użyte technologie i biblioteki:

1. **Apache Spark (pyspark)**:
    - SparkSession do tworzenia sesji Spark.
    - Operacje na DataFrame do manipulacji i przetwarzania dużych zbiorów danych.
    - Funkcje transformacji i agregacji danych z pyspark.sql.functions.
2. **NumPy**:
    - Wspomaganie obliczeń matematycznych.
3. **Pandas**:
    - Przejściowe konwersje danych między Spark a Pandas DataFrame.
4. **Matplotlib**:
    - Wizualizacja danych i analiz wyników modeli.
5. **Scikit-learn**:
    - Algorytmy klastrowania (KMeans).
    - Obliczanie metryk ewaluacyjnych (Silhouette Score).
6. **Spark MLlib**:
    - Skalowanie danych (StandardScaler).
    - Klasteryzacja z KMeans Spark ML.

Etapy przetwarzania danych:

1. **Wczytywanie danych:**
   Pliki CSV wczytywane są do DataFrame przy użyciu Sparka z opcjami automatycznego wykrywania typów i nagłówków.
2. **Eksploracja i czyszczenie danych:**
    - Wyświetlenie statystyk opisowych (describe) i schematu danych (printSchema).
    - Usunięcie brakujących wartości (dropna()).
    - Filtrowanie nietypowych wartości (np. Quantity > 0) oraz usuwanie elementów odstających.
3. **Tworzenie nowych kolumn:**
    - Kolumna amount, będąca wynikiem mnożenia Quantity i UnitPrice.
4. **Modelowanie danych (RFM model):**
    - Wartości dla R (recency), F (frequency), M (monetary) wyznaczane na podstawie kolumn z DataFrame.

- Skalowanie danych przy użyciu StandardScaler.

5. **Klasteryzacja i analiza:**
    - Wybór optymalnej liczby klastrów za pomocą metody łokcia oraz metryki Silhouette.
    - Implementacja KMeans z trzema klastrami.
6. **Ewaluacja modelu:**
    - Metryki ewaluacyjne (Silhouette Score).
    - Analiza wyników i interpretacja klastrów.