Task No: Use Case 1 Date:22–10–25	Data analytics using Apache Spark on Amazon food dataset	CO2 K3

AIM: To achieve the task of finding pairs of items frequently reviewed together in the Amazon food dataset using Apache Spark.

Procedure:

1. Load the dataset into an RDD (Resilient Distributed Dataset).

2. Transpose the dataset to create a PairRDD of the form user-id -> list of product-ids reviewed by user-id.

3. Generate pairs of products for each user's list of reviewed products.

4. Count the frequencies of these pairs.

5. Filter the pairs that appear more than once.

6. Sort the filtered pairs by frequency.

7. Write the results to an output folder.

Implementation:

```
from pyspark import SparkContext, SparkConf

# Initialize Spark

conf = SparkConf().setAppName("Frequently Reviewed Products")

sc = SparkContext(conf=conf)

# Load the Amazon food dataset as an RDD (replace 'your_input_path' with the actual path)

lines = sc.textFile("your_input_path")


# Define a function to parse each line and extract user-id and product-id

def parse_line(line):

    elements = line.split(',')

    user_id = elements[0].strip()

    product_id = elements[1].strip()

    return (user_id, product_id)


# Parse the dataset and create a PairRDD of user-id -> list of product-ids reviewed

user_product_rdd = lines.map(parse_line).groupByKey()
```

```
# Generate pairs of products for each user's list of reviewed products
product_pairs = user_product_rdd.flatMapValues(lambda products: [(p1, p2) for p1 in products for p2 in products if p1 < p2])

# Count the frequencies of product pairs
pair_counts = product_pairs.map(lambda pair: (pair, 1)).reduceByKey(lambda x, y: x + y)

# Filter pairs that appear more than once
frequent_pairs = pair_counts.filter(lambda x: x[1] > 1)

# Sort the pairs by frequency in descending order
sorted_pairs = frequent_pairs.sortBy(lambda x: x[1], ascending=False)
# Write the results to an output folder (replace 'your_output_path' with the desired path)
sorted_pairs.saveAsTextFile("your_output_path")
# Stop Spark
sc.stop()
```

output:

```
(('user1', 'product1', 'product2'), 3)
(('user2', 'product2', 'product3'), 2)
(('user3', 'product1', 'product3'), 4)
...
```

Result: This Spark application will transpose the dataset, count the frequencies of product pairs, filter frequent pairs, sort them by frequency, and save the results in the specified output folder.