Market_Basket_Analysis_and_Product_Recommendation_with_SparkMLi

September 6, 2023

Task 1 - Market Basket Analysis and Product Recommendation

GOAL: The goal of this task is to use Spark MLlib to build a model to generate association rules to quickly run the market basket analysis to uncover associations between different items, then further to provide recommendations for purchase on a distributed platform.

[0]:
```
# Importing the library
from pyspark.ml.fpm import FPGrowth
```

[0]:
```
# 1.3 - Creating the training set
# Reading the dataframe
train_df = spark.read.format("csv").option("header", "false").load("dbfs:/
 ↪FileStore/shared_uploads/clvrashmika@gmail.com/Lab5_Part1_TrainData.csv")
```

[0]:
```
# Displaying the dataframe
display(train_df)
```

[0]:
```
# Modifying the dataframe to perform analysis
count = 0
mylist = []
for row in train_df.collect():
    temp = (count,list(filter(None, row)))
    mylist.append(temp)
    count = count + 1
```

[0]:
```
# Printing the list to show the new format
mylist
```

```
Out[122]: [(0, ['bread']),
 (1, ['peanut butter', ' apple']),
 (2, [' peanut butter', ' bread']),
 (3, ['peanut butter', ' bread', ' apple']),
 (4, ['peanut butter', ' bread', ' milk']),
 (5, ['peanut butter', ' bread', ' milk', ' chocolate']),
 (6, ['bread', ' milk', ' orange']),
 (7, ['apple', ' chocolate', ' milk']),
 (8, ['peanut butter', ' milk', ' chocolate', ' apple']),
```

```
 (9, ['cheese', ' bread', ' milk', ' potatoes ']),
 (10, ['cheese', ' pasta', ' ketchup']),
 (11, ['milk', ' cheese', ' pasta', ' ketchup']),
 (12, ['pasta', ' ketchup', ' cheese', ' potatoes', ' milk']),
 (13, ['bread', ' milk', ' chocolate', ' pasta']),
 (14, [' milk', ' pasta', ' potatoes', ' ketchup', ' bread']),
 (15, ['apple', ' chocolate', ' pasta']),
 (16, ['milk', 'bread']),
 (17, ['apple', 'milk']),
 (18, ['milk', 'chocolate']),
 (19, ['milk'])]
```

[0]:
```
# Creating a new dataframe from the above list
train_df1 = spark.createDataFrame(mylist, ["id", "items"])
```

[0]:
```
# 1.4 - FP Growth Model
fpGrowth = FPGrowth(itemsCol="items", minSupport=0.05, minConfidence=0.07)
model = fpGrowth.fit(train_df1)
```

[0]:
```
#1.5 -  Display frequent itemsets.
model.freqItemsets.show()
```

```
+--------------------+----+
|               items|freq|
+--------------------+----+
|            [ apple]|   3|
|[ apple, peanut b…|   3|
|[ apple, peanut b…|   1|
|[ apple, peanut b…|   1|
|[ apple, peanut b…|   1|
|[ apple, peanut b…|   1|
|    [ apple,  bread]|   1|
|[ apple,  chocolate]|   1|
|[ apple,  chocola…|   1|
|     [ apple,  milk]|   1|
|             [pasta]|   1|
|   [pasta,  ketchup]|   1|
|[pasta,  ketchup,…|   1|
|  [pasta,  potatoes]|   1|
|[pasta,  potatoes…|   1|
|[pasta,  potatoes…|   1|
|[pasta,  potatoes…|   1|
|    [pasta,  cheese]|   1|
|[pasta,  cheese, …|   1|
|[pasta,  cheese, …|   1|
+--------------------+----+
only showing top 20 rows
```

[0]:
```
#1.6  Display generated association rules.
display(model.associationRules)
```

[0]:
```
# 1.7 Creating a test set
test_df = spark.createDataFrame([
    (0, ['bread']),
    (1, ['potatoes','milk']),
    (2, ['chocolate']),
    (3, ['pasta', 'cheese']),
    (4, ['apple','milk']),
    (5, ['milk']),
    (6, ['chocolate', 'bread', 'milk']),
    (7, ['bread', 'milk'])
], ["id", "items"])
```

[0]:
```
# 1.8 Making predictions
# transform examines the input items against all the association rules and␣
 ↪summarize the consequents as prediction
display(model.transform(test_df))
```

1.10 Task 1 - Additional

[0]:
```
# Reading and displaying the dataframe
extra_df = spark.read.format("csv").option("header", "false").load("dbfs:/
 ↪FileStore/shared_uploads/clvrashmika@gmail.com/groceries.csv")
display(extra_df)
```

[0]:
```
# Modifying the dataframe to perform analysis
count = 0
mylist = []
for row in extra_df.collect():
    temp = (count,list(filter(None, row)))
    mylist.append(temp)
    count = count + 1

# Creating a new dataframe from the above list
extra_df1 = spark.createDataFrame(mylist, ["id", "items"])
```

[0]:
```
# Splitting the dataset into train and test dataframes
trainDF, testDF = extra_df1.randomSplit([0.8, 0.2], seed=25)
print(trainDF.cache().count()) # Cache because accessing training data multiple␣
 ↪times
print(testDF.count())
```

```
7884
1951
```

[0]:
```
# FP Growth Model
fpGrowth = FPGrowth(itemsCol="items", minSupport=0.01, minConfidence=0.01)
model = fpGrowth.fit(trainDF)
```

[0]:
```
# Display frequent itemsets.
display(model.freqItemsets)
```

[0]:
```
# Display generated association rules.
display(model.associationRules)
```

[0]:
```
# transform examines the input items against all the association rules and␣
 ↪summarize the
# consequents as prediction
display(model.transform(testDF))
```

1.9 References:

1. Dr. Liao's Code Examples & Tutorials: Blackboard/Liao_PySpark_basic_databricks.html 2. PySpark: https://spark.apache.org/docs/2.4.0/api/python/pyspark.html
3. Frequent Pattern Mining : https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html