Chapter 9: Frequent Pattern Mining

Ex5: instacart_online_grocery_shopping_2017_05_01

Dataset: link download https://www.instacart.com/datasets/grocery-shopping-2017 (https://www.instacart.com/datasets/grocery-shopping-2017)

Requirement:

- Read data
- Pre-process data
- Apply FPGrowth algorithm to find association rules from this dataset. Find the most popular items in a basket.

In [1]:
```
import findspark
findspark.init()
```

In [2]:
```
import pyspark
```

In [3]:
```
from pyspark import SparkContext
from pyspark.conf import SparkConf
from pyspark.sql import SparkSession
```

In [4]:
```
sc = SparkContext()
```

In [5]:
```
spark = SparkSession.builder.appName('ex_demo').getOrCreate()
```

In [6]:
```
from pyspark.ml.fpm import FPGrowth
```

In [7]:
```
# Loads data.
data = spark.read.csv('instacart_2017_05_01/order_products__train.csv',
                      header=True,
                      inferSchema=True)
```

In [8]:
```
data.count()
```

Out[8]:
```
1384617
```

In [9]:
```
data.show()
```

```
+--------+----------+-----------------+---------+
|order_id|product_id|add_to_cart_order|reordered|
+--------+----------+-----------------+---------+
|       1|     49302|                1|        1|
|       1|     11109|                2|        1|
|       1|     10246|                3|        0|
|       1|     49683|                4|        0|
|       1|     43633|                5|        1|
|       1|     13176|                6|        0|
|       1|     47209|                7|        0|
|       1|     22035|                8|        1|
|      36|     39612|                1|        0|
|      36|     19660|                2|        1|
|      36|     49235|                3|        0|
|      36|     43086|                4|        1|
|      36|     46620|                5|        1|
|      36|     34497|                6|        1|
|      36|     48679|                7|        1|
|      36|     46979|                8|        1|
|      38|     11913|                1|        0|
|      38|     18159|                2|        0|
|      38|      4461|                3|        0|
|      38|     21616|                4|        1|
+--------+----------+-----------------+---------+
only showing top 20 rows
```

In [10]:
```
# Pre-processing data
from pyspark.sql.functions import collect_list, col, count, collect_set
```

In [11]:
```
data.createOrReplaceTempView("order_products_train")
```

In [12]:
```
products = spark.sql("select distinct product_id from order_products_train")
products.count()
```

Out[12]: 39123

In [13]:
```
rawData = spark.sql("select * from order_products_train")
baskets = rawData.groupBy('order_id').agg(collect_set('product_id')\
                                          .alias('items'))
baskets.createOrReplaceTempView('baskets')
```

In [14]:
```
baskets.show(5, truncate=False)
```

```
+--------+------------------------------------------------------------------------
------------------------------------------------------------------------------
-------------------------------------------------------------+
|order_id|items
|
+--------+------------------------------------------------------------------------
------------------------------------------------------------------------------
-------------------------------------------------------------+
|1342    |[30827, 3798, 14966, 21137, 46129, 33081, 13176, 7862]
|
|1591    |[48246, 44116, 24852, 5194, 9130, 48823, 46473, 40310, 32520, 22105,
16900, 27681, 4103, 44008, 17758, 41671, 25316, 45061, 38805, 48205, 25237, 196
04, 5384, 27344, 17203, 18792, 12986, 39758, 34358, 31215, 9387]|
|4519    |[29270]
|
|4935    |[45190]
|
|6357    |[33731, 14669, 43789, 37524, 39408, 43129, 24852, 48745, 38772]
|
+--------+------------------------------------------------------------------------
------------------------------------------------------------------------------
-------------------------------------------------------------+
only showing top 5 rows
```

In [15]:
```
type(baskets)
```

Out[15]:
```
pyspark.sql.dataframe.DataFrame
```

In [16]:
```
fpGrowth = FPGrowth(itemsCol="items", minSupport=0.003,
                    minConfidence=0.003)
model = fpGrowth.fit(baskets)
```

In [17]:
```
# Display frequent itemsets.
model.freqItemsets.show()
```

```
+--------------------+-----+
|               items| freq|
+--------------------+-----+
|             [13629]|  772|
|              [5194]|  475|
|             [24852]|18726|
|             [13176]|15480|
|             [35921]|  769|
|             [20345]|  473|
|             [21137]|10894|
|      [21137, 13176]| 3074|
|      [21137, 24852]| 2174|
|             [23165]|  764|
|             [13380]|  473|
|              [7969]|  472|
|             [21903]| 9784|
|      [21903, 21137]| 1639|
|[21903, 21137, 13...|  587|
|      [21903, 13176]| 2236|
|      [21903, 24852]| 2000|
|             [32478]|  763|
|             [47626]| 8135|
|      [47626, 21137]| 1017|
+--------------------+-----+
only showing top 20 rows
```

In [18]:
```
# transform examines the input items against all the association rules and summar
# consequents as prediction
mostPopularItemInABasket = model.transform(baskets)
```

In [19]:
```
mostPopularItemInABasket.show()
```

```
+--------+--------------------+--------------------+
|order_id|               items|          prediction|
+--------+--------------------+--------------------+
|    1342|[30827, 3798, 149...|[21903, 47626, 47...|
|    1591|[48246, 44116, 24...|[21137, 21903, 47...|
|    4519|             [29270]|                  []|
|    4935|             [45190]|                  []|
|    6357|[33731, 14669, 43...|[21137, 21903, 47...|
|   10362|[28522, 43789, 12...|[21137, 47626, 47...|
|   19204|[45255, 37285, 48...|                  []|
|   29601|[2716, 48057, 219...|[21137, 21903, 47...|
|   31035|[40723, 8174, 131...|[21137, 21903, 47...|
|   40011|[27292, 35213, 21...|[21137, 13176, 24...|
|   46266|[38558, 48642, 13...|[47626, 47766, 47...|
|   51607|[41390, 42752, 17...|                  []|
|   58797|[30827, 8803, 326...|[21137, 21903, 47...|
|   61793|[26348, 6184, 433...|[21137, 16797, 39...|
|   67089|[47766, 29388, 21...|[47626, 21137, 47...|
|   70863|[34791, 2618, 173...|      [13176, 24852]|
|   88674|[25659, 16262, 22...|                  []|
|   91937|[20708, 38200, 26...|                  []|
|   92317|[18105, 34969, 17...|[13176, 21903, 21...|
|   99621|[21616, 43789, 38...|[26209, 21137, 47...|
+--------+--------------------+--------------------+
only showing top 20 rows
```

Use product_name instead of product_id

In [20]:
```
product_data = spark.read.csv('instacart_2017_05_01/products.csv',
                              header=True, inferSchema=True)
```

In [21]:
```
product_data.show(5, truncate=False)
```

```
+----------+-------------------------------------------------------------+-
-------+-------------+
|product_id|product_name                                                 |a
isle_id|department_id|
+----------+-------------------------------------------------------------+-
-------+-------------+
|1         |Chocolate Sandwich Cookies                                   |6
1      |19           |
|2         |All-Seasons Salt                                             |1
04     |13           |
|3         |Robust Golden Unsweetened Oolong Tea                         |9
4      |7            |
|4         |Smart Ones Classic Favorites Mini Rigatoni With Vodka Cream Sauce|3
8      |1            |
|5         |Green Chile Anytime Sauce                                    |5
|13           |
+----------+-------------------------------------------------------------+-
-------+-------------+
only showing top 5 rows
```

In [22]:
```
product_data.createOrReplaceTempView("products")
```

In [23]:
```
rawData_1 = spark.sql('''select p.product_name, o.order_id from products p
                         inner join order_products_train o
                         where o.product_id = p.product_id''')
baskets_1 = rawData_1.groupBy('order_id').agg(collect_set('product_name')\
                                              .alias('items'))
baskets_1.createOrReplaceTempView('baskets')
```

In [24]:
```
baskets_1.head(3)
```

Out[24]:
```
[Row(order_id=1342, items=['Raw Shrimp', 'Seedless Cucumbers', 'Versatile Stain
Remover', 'Organic Strawberries', 'Organic Mandarins', 'Chicken Apple Sausage',
'Pink Lady Apples', 'Bag of Organic Bananas']),
 Row(order_id=1591, items=['Cracked Wheat', 'Strawberry Rhubarb Yoghurt', 'Orga
nic Bunny Fruit Snacks Berry Patch', 'Goodness Grapeness Organic Juice Drink',
'Honey Graham Snacks', 'Spinach', 'Granny Smith Apples', 'Oven Roasted Turkey B
reast', 'Pure Vanilla Extract', 'Chewy 25% Low Sugar Chocolate Chip Granola',
'Banana', 'Original Turkey Burgers Smoke Flavor Added', 'Twisted Tropical Tango
Organic Juice Drink', 'Navel Oranges', 'Lower Sugar Instant Oatmeal  Variety',
'Ultra Thin Sliced Provolone Cheese', 'Natural Vanilla Ice Cream', 'Cinnamon Mu
ltigrain Cereal', 'Garlic', 'Goldfish Pretzel Baked Snack Crackers', 'Original
Whole Grain Chips', 'Medium Scarlet Raspberries', 'Lemon Yogurt', 'Original Pat
ties (100965) 12 Oz Breakfast', 'Nutty Bars', 'Strawberry Banana Smoothie', 'Gr
een Machine Juice Smoothie', 'Coconut Dreams Cookies', 'Buttermilk Waffles', 'U
ncured Genoa Salami', 'Organic Greek Whole Milk Blended Vanilla Bean Yogurt']),
 Row(order_id=4519, items=['Beet Apple Carrot Lemon Ginger Organic Cold Pressed
Juice Beverage'])]
```

In [25]:
```
fpGrowth_1 = FPGrowth(itemsCol="items",
                      minSupport=0.003,
                      minConfidence=0.003)
model_1 = fpGrowth.fit(baskets_1)
```

In [26]:
```
# Display frequent itemsets.
model_1.freqItemsets.show(truncate=False)
```

```
+------------------------------------------------------------------------+-----+
|items                                                                   |freq |
+------------------------------------------------------------------------+-----+
|[Organic Tomato Basil Pasta Sauce]                                      |772  |
|[Organic Spinach Bunch]                                                 |475  |
|[Banana]                                                                |18726|
|[Bag of Organic Bananas]                                                |15480|
|[Organic Large Grade A Brown Eggs]                                      |769  |
|[Organic Blue Corn Tortilla Chips]                                      |473  |
|[Organic Strawberries]                                                  |10894|
|[Organic Strawberries, Bag of Organic Bananas]                          |3074 |
|[Organic Strawberries, Banana]                                          |2174 |
|[Organic Leek]                                                          |764  |
|[Thin Crust Pepperoni Pizza]                                            |473  |
|[Lime]                                                                  |472  |
|[Organic Baby Spinach]                                                  |9784 |
|[Organic Baby Spinach, Organic Strawberries]                            |1639 |
|[Organic Baby Spinach, Organic Strawberries, Bag of Organic Bananas]|587  |
|[Organic Baby Spinach, Bag of Organic Bananas]                          |2236 |
|[Organic Baby Spinach, Banana]                                          |2000 |
|[Reduced Fat 2% Milk]                                                   |763  |
|[Large Lemon]                                                           |8135 |
|[Large Lemon, Organic Strawberries]                                     |1017 |
+------------------------------------------------------------------------+-----+
only showing top 20 rows
```

In [27]:
```
mostPopularItemInABasket_1 = model_1.transform(baskets_1)
```

In [28]:
```
mostPopularItemInABasket_1.head(3)
```

Out[28]:
```
[Row(order_id=1342, items=['Raw Shrimp', 'Seedless Cucumbers', 'Versatile Stain
Remover', 'Organic Strawberries', 'Organic Mandarins', 'Chicken Apple Sausage',
'Pink Lady Apples', 'Bag of Organic Bananas'], prediction=['Organic Baby Spinac
h', 'Large Lemon', 'Organic Avocado', 'Organic Hass Avocado', 'Strawberries',
'Limes', 'Organic Raspberries', 'Organic Blueberries', 'Organic Whole Milk', 'O
rganic Cucumber', 'Organic Zucchini', 'Organic Yellow Onion', 'Organic Garlic',
'Seedless Red Grapes', 'Asparagus', 'Organic Grape Tomatoes', 'Organic Red Onio
n', 'Organic Baby Carrots', 'Honeycrisp Apple', 'Organic Cilantro', 'Organic Le
mon', 'Sparkling Water Grapefruit', 'Raspberries', 'Organic Fuji Apple', 'Small
Hass Avocado', 'Organic Baby Arugula', 'Organic Large Extra Fancy Fuji Apple',
'Original Hummus', 'Organic Blackberries', 'Organic Gala Apples', 'Fresh Caulif
lower', 'Organic Half & Half', 'Michigan Organic Kale', 'Organic Small Bunch Ce
lery', 'Organic Garnet Sweet Potato (Yam)', 'Organic Tomato Cluster', 'Carrot
s', 'Organic Peeled Whole Baby Carrots', 'Organic Italian Parsley Bunch', 'Orga
nic Red Bell Pepper', 'Organic Granny Smith Apple', 'Hass Avocados', 'Apple Hon
eycrisp Organic', 'Spring Water', 'Organic Unsweetened Almond Milk', 'Unsweeten
ed Almondmilk', 'Organic Ginger Root', 'Organic Whole String Cheese', 'Organic
Navel Orange', 'Large Alfresco Eggs', "Organic D'Anjou Pears", 'Organic Kiwi',
'Organic Grade A Free Range Large Brown Eggs', 'Organic Lacinato (Dinosaur) Kal
e', 'Organic Carrot Bunch', 'Organic Broccoli', 'Organic Black Beans', 'Banan
a', 'Broccoli Crown', 'Organic Banana']),
 Row(order_id=1591, items=['Cracked Wheat', 'Strawberry Rhubarb Yoghurt', 'Orga
nic Bunny Fruit Snacks Berry Patch', 'Goodness Grapeness Organic Juice Drink',
'Honey Graham Snacks', 'Spinach', 'Granny Smith Apples', 'Oven Roasted Turkey B
reast', 'Pure Vanilla Extract', 'Chewy 25% Low Sugar Chocolate Chip Granola',
'Banana', 'Original Turkey Burgers Smoke Flavor Added', 'Twisted Tropical Tango
Organic Juice Drink', 'Navel Oranges', 'Lower Sugar Instant Oatmeal  Variety',
'Ultra Thin Sliced Provolone Cheese', 'Natural Vanilla Ice Cream', 'Cinnamon Mu
ltigrain Cereal', 'Garlic', 'Goldfish Pretzel Baked Snack Crackers', 'Original
Whole Grain Chips', 'Medium Scarlet Raspberries', 'Lemon Yogurt', 'Original Pat
ties (100965) 12 Oz Breakfast', 'Nutty Bars', 'Strawberry Banana Smoothie', 'Gr
een Machine Juice Smoothie', 'Coconut Dreams Cookies', 'Buttermilk Waffles', 'U
ncured Genoa Salami', 'Organic Greek Whole Milk Blended Vanilla Bean Yogurt'],
prediction=['Organic Strawberries', 'Organic Baby Spinach', 'Large Lemon', 'Org
anic Avocado', 'Organic Hass Avocado', 'Strawberries', 'Limes', 'Organic Raspbe
rries', 'Organic Blueberries', 'Organic Whole Milk', 'Organic Cucumber', 'Organ
ic Zucchini', 'Organic Yellow Onion', 'Organic Garlic', 'Seedless Red Grapes',
'Asparagus', 'Organic Grape Tomatoes', 'Organic Red Onion', 'Yellow Onions', 'O
rganic Baby Carrots', 'Honeycrisp Apple', 'Organic Cilantro', 'Sparkling Water
Grapefruit', 'Raspberries', 'Organic Fuji Apple', 'Small Hass Avocado', 'Brocco
li Crown', 'Organic Baby Arugula', 'Red Peppers', 'Organic Large Extra Fancy Fu
ji Apple', 'Original Hummus', 'Organic Blackberries', 'Organic Gala Apples', 'F
resh Cauliflower', 'Organic Half & Half', 'Michigan Organic Kale', 'Organic Sma
ll Bunch Celery', 'Organic Garnet Sweet Potato (Yam)', 'Organic Tomato Cluste
r', 'Green Bell Pepper', 'Carrots', 'Organic Peeled Whole Baby Carrots', 'Half
& Half', 'Cucumber Kirby', 'Organic Red Bell Pepper', 'Organic Granny Smith App
le', 'Blueberries', '100% Whole Wheat Bread', 'Apple Honeycrisp Organic', 'Red
Vine Tomato', 'Unsweetened Almondmilk', 'Boneless Skinless Chicken Breasts', 'O
rganic Whole String Cheese', 'Roma Tomato', 'Bunched Cilantro', 'Jalapeno Peppe
rs', "Organic D'Anjou Pears", 'Orange Bell Pepper', 'Grape White/Green Seedles
s', 'Red Raspberries', 'Clementines, Bag', 'Unsweetened Original Almond Breeze
Almond Milk', 'Bartlett Pears']),
 Row(order_id=4519, items=['Beet Apple Carrot Lemon Ginger Organic Cold Pressed
Juice Beverage'], prediction=[])]
```

In [29]:
```
type(mostPopularItemInABasket_1)
```

Out[29]:
```
pyspark.sql.dataframe.DataFrame
```

In [30]:
```
# chuyển list array thành string
from pyspark.sql.types import StringType
```

In [31]:
```
mostPopularItemInABasket_1.printSchema()
```

```
root
 |-- order_id: integer (nullable = true)
 |-- items: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- prediction: array (nullable = true)
 |    |-- element: string (containsNull = true)
```

In [33]:
```
mostPopularItemInABasket_1.createOrReplaceTempView("popular_items")
```

In [34]:
```
DF_cast = mostPopularItemInABasket_1.select('order_id',
                    mostPopularItemInABasket_1.items.cast(StringType()),
                    mostPopularItemInABasket_1.prediction.cast(StringType()))

DF_cast.printSchema()
```

```
root
 |-- order_id: integer (nullable = true)
 |-- items: string (nullable = true)
 |-- prediction: string (nullable = true)
```

In [35]:
```
DF_cast.head(3)
```

Out[35]:
```
[Row(order_id=1342, items='[Raw Shrimp, Seedless Cucumbers, Versatile Stain Rem
over, Organic Strawberries, Organic Mandarins, Chicken Apple Sausage, Pink Lady
Apples, Bag of Organic Bananas]', prediction="[Organic Baby Spinach, Large Lemo
n, Organic Avocado, Organic Hass Avocado, Strawberries, Limes, Organic Raspberr
ies, Organic Blueberries, Organic Whole Milk, Organic Cucumber, Organic Zucchin
i, Organic Yellow Onion, Organic Garlic, Seedless Red Grapes, Asparagus, Organi
c Grape Tomatoes, Organic Red Onion, Organic Baby Carrots, Honeycrisp Apple, Or
ganic Cilantro, Organic Lemon, Sparkling Water Grapefruit, Raspberries, Organic
Fuji Apple, Small Hass Avocado, Organic Baby Arugula, Organic Large Extra Fancy
Fuji Apple, Original Hummus, Organic Blackberries, Organic Gala Apples, Fresh C
auliflower, Organic Half & Half, Michigan Organic Kale, Organic Small Bunch Cel
ery, Organic Garnet Sweet Potato (Yam), Organic Tomato Cluster, Carrots, Organi
c Peeled Whole Baby Carrots, Organic Italian Parsley Bunch, Organic Red Bell Pe
pper, Organic Granny Smith Apple, Hass Avocados, Apple Honeycrisp Organic, Spri
ng Water, Organic Unsweetened Almond Milk, Unsweetened Almondmilk, Organic Ging
er Root, Organic Whole String Cheese, Organic Navel Orange, Large Alfresco Egg
s, Organic D'Anjou Pears, Organic Kiwi, Organic Grade A Free Range Large Brown
Eggs, Organic Lacinato (Dinosaur) Kale, Organic Carrot Bunch, Organic Broccoli,
Organic Black Beans, Banana, Broccoli Crown, Organic Banana]"),
 Row(order_id=1591, items='[Cracked Wheat, Strawberry Rhubarb Yoghurt, Organic
Bunny Fruit Snacks Berry Patch, Goodness Grapeness Organic Juice Drink, Honey G
raham Snacks, Spinach, Granny Smith Apples, Oven Roasted Turkey Breast, Pure Va
nilla Extract, Chewy 25% Low Sugar Chocolate Chip Granola, Banana, Original Tur
key Burgers Smoke Flavor Added, Twisted Tropical Tango Organic Juice Drink, Nav
el Oranges, Lower Sugar Instant Oatmeal  Variety, Ultra Thin Sliced Provolone C
heese, Natural Vanilla Ice Cream, Cinnamon Multigrain Cereal, Garlic, Goldfish
Pretzel Baked Snack Crackers, Original Whole Grain Chips, Medium Scarlet Raspbe
rries, Lemon Yogurt, Original Patties (100965) 12 Oz Breakfast, Nutty Bars, Str
awberry Banana Smoothie, Green Machine Juice Smoothie, Coconut Dreams Cookies,
Buttermilk Waffles, Uncured Genoa Salami, Organic Greek Whole Milk Blended Vani
lla Bean Yogurt]', prediction="[Organic Strawberries, Organic Baby Spinach, Lar
ge Lemon, Organic Avocado, Organic Hass Avocado, Strawberries, Limes, Organic R
aspberries, Organic Blueberries, Organic Whole Milk, Organic Cucumber, Organic
Zucchini, Organic Yellow Onion, Organic Garlic, Seedless Red Grapes, Asparagus,
Organic Grape Tomatoes, Organic Red Onion, Yellow Onions, Organic Baby Carrots,
Honeycrisp Apple, Organic Cilantro, Sparkling Water Grapefruit, Raspberries, Or
ganic Fuji Apple, Small Hass Avocado, Broccoli Crown, Organic Baby Arugula, Red
Peppers, Organic Large Extra Fancy Fuji Apple, Original Hummus, Organic Blackbe
rries, Organic Gala Apples, Fresh Cauliflower, Organic Half & Half, Michigan Or
ganic Kale, Organic Small Bunch Celery, Organic Garnet Sweet Potato (Yam), Orga
nic Tomato Cluster, Green Bell Pepper, Carrots, Organic Peeled Whole Baby Carro
ts, Half & Half, Cucumber Kirby, Organic Red Bell Pepper, Organic Granny Smith
Apple, Blueberries, 100% Whole Wheat Bread, Apple Honeycrisp Organic, Red Vine
Tomato, Unsweetened Almondmilk, Boneless Skinless Chicken Breasts, Organic Whol
e String Cheese, Roma Tomato, Bunched Cilantro, Jalapeno Peppers, Organic D'Anj
ou Pears, Orange Bell Pepper, Grape White/Green Seedless, Red Raspberries, Clem
entines, Bag, Unsweetened Original Almond Breeze Almond Milk, Bartlett Pear
s]"),
 Row(order_id=4519, items='[Beet Apple Carrot Lemon Ginger Organic Cold Pressed
Juice Beverage]', prediction='[]')]
```

In [36]:
```
DF_cast.write.csv('mostPopularItemInABasket.csv')
```