Market Basket Analysis using PySpark's Implementation of FPGrowth

FPGrowth is an algorithm that performs market basket analysis, similar to the Apriori algorithm. I first used it when I ran into resource issues with Apriori and I was impressed with the speed. So I am giving it a try on this dataset using pyspark. The documentation for FPGrowth is pretty straightforward and describes the hyperparameters and the results.

Import the relevant libraries

The libraries such as SparkContext and SparkSession are general pyspark libraries needed for pyspark applications. The specific function used for market basket analysis is FPGrowth.

In [ ]:

```
 # Used for a histogram
!pip install pyspark_dist_explore
```

In [22]:

```
from pyspark import SparkContext
# Rather than generally using the functions, I should explicitly import the ones
from pyspark.sql import functions as f, SparkSession, Column
from pyspark_dist_explore import hist
import matplotlib.pyplot as plt
from pyspark.ml.fpm import FPGrowth
```

In [23]:

```
# Create a spark session. All sorts of settings can be specified here.
spark = SparkSession.builder.appName("arlUsingPyspark").getOrCreate()
```

In [24]:

```
# Read the dataset
df = spark.read.csv("/Users/admin/Jupyter Examples/basket.csv", header=True).wit
#df_all = spark.read.csv("/Users/admin/Jupyter Examples/Groceries data.csv", hea
```

In [25]:

```
# Show the dataframes
df.show(5)
#df_all.show(5)
```

```
+-----------+------------------+-------------------+------+----+----+----+----+--
--+----+----+---+
|          0|                 1|                  2|     3|   4|   5|   6|   7|
8|   9|  10| id|
+-----------+------------------+-------------------+------+----+----+----+----+--
--+----+----+---+
| whole milk|            pastry|        salty snack|  NULL|NULL|NULL|NULL|NULL|NU
LL|NULL|NULL|  0|
|    sausage|        whole milk|semi-finished bread|yogurt|NULL|NULL|NULL|NULL|NU
LL|NULL|NULL|  1|
|       soda|pickled vegetables|               NULL|  NULL|NULL|NULL|NULL|NULL|NU
LL|NULL|NULL|  2|
|canned beer|   misc. beverages|               NULL|  NULL|NULL|NULL|NULL|NULL|NU
LL|NULL|NULL|  3|
|    sausage|  hygiene articles|               NULL|  NULL|NULL|NULL|NULL|NULL|NU
LL|NULL|NULL|  4|
+-----------+------------------+-------------------+------+----+----+----+----+--
--+----+----+---+
only showing top 5 rows
```

In [ ]:

```
#num_baskets = df_all.groupBy("Member_number").count()
#num_baskets.show(5)
```

In [ ]:

```
#fig, ax = plt.subplots()
#hist(ax, num_baskets.select('count'), bins = 30, color=['blue'])
```

Run PySpark's implementation of FPGrowth

First step is to collect the baskets into sets. FPGrowth requires each basket to be an array that looks like:

- ['item1','item2', 'imem3']

The basket dataframe uses wide rather than long format, with Null if the basket contains fewer than 10 items.

In [26]:

```
df_basket = df.select("id", f.array([df[c] for c in df.columns[:11]]).alias("bas
# False tells show() to not truncate the columns when printing.
df_basket.show(3, False)
```

```
+---+----------------------------------------------------------------------------
----------------+
|id |basket
|
+---+----------------------------------------------------------------------------
----------------+
|0  |[whole milk, pastry, salty snack, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL]           |
|1  |[sausage, whole milk, semi-finished bread, yogurt, NULL, NULL, NULL, NULL, N
ULL, NULL, NULL]|
|2  |[soda, pickled vegetables, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, N
ULL]            |
+---+----------------------------------------------------------------------------
----------------+
only showing top 3 rows
```

There should not be any nulls in the array. Remove using array_except()

This will be the final dataframe used for FPGrowth.

In [27]:
```
df_aggregated = df_basket.select("id", f.array_except("basket", f.array(f.lit(No
df_aggregated.show(3, False)
```

```
+---+--------------------------------------------------+
|id |basket                                            |
+---+--------------------------------------------------+
|0  |[whole milk, pastry, salty snack]                 |
|1  |[sausage, whole milk, semi-finished bread, yogurt]|
|2  |[soda, pickled vegetables]                        |
+---+--------------------------------------------------+
only showing top 3 rows
```

Hyperparameters

The hyperparameters used in FPGrowth are minimum support, minimum confidence, and number of partitions.

- minSupport - The minimum support of an item to be considered in a frequent itemset.
- minConfidence - The minimum confidence for generating an association rule from an itemset.
- numPartitions - The number of partitions used to distribute the work. This is Spark-specific.

The default number of partitions is the number of partitions for the input dataset.

In [28]:
```
# Run FPGrowth and fit the model.
fp = FPGrowth(minSupport=0.001, minConfidence=0.001, itemsCol='basket', predicti
model = fp.fit(df_aggregated)
```

```
In [29]: # View a subset of the frequent itemset.
         model.freqItemsets.show(10, False)
```

```
+-------------------------+----+
|items                    |freq|
+-------------------------+----+
|[cocoa drinks]           |16  |
|[canned fruit]           |21  |
|[specialty cheese]       |72  |
|[chocolate marshmallow]  |60  |
|[pet care]               |85  |
|[house keeping products]|45  |
|[jam]                    |34  |
|[light bulbs]            |29  |
|[beef]                   |508 |
|[beef, frankfurter]      |15  |
+-------------------------+----+
only showing top 10 rows
```

```
In [ ]: # Use filter to view just the association rules with the highest confidence.
        model.associationRules.filter(model.associationRules.confidence>0.15).show(20, F
```

Let's create a prediction based on the generated association rules

This is pretty similar to creating a prediction using other methods. The data column needs to have the same column name as the column specified in the model fit.

```
In [30]: # Create a PySpark dataframe
         columns = ['basket']
         new_data = [(['ham', 'yogurt', 'light bulbs'],), (['jam', 'cocoa drinks', 'pet c
         rdd = spark.sparkContext.parallelize(new_data)
         new_df = rdd.toDF(columns)
         new_df.show(2,False)
```

```
+-----------------------------+
|basket                       |
+-----------------------------+
|[ham, yogurt, light bulbs]   |
|[jam, cocoa drinks, pet care]|
+-----------------------------+
```

Predict!

Now that we have a new PySpark dataframe with data, predict. The first basket generates numerous predictions based on the association rules, however the second basket does not generate any.

```
In [31]: model.transform(new_df).show(5, False)
```

```
+-----------------------------+--------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
--------+
|basket                       |prediction
|
+-----------------------------+--------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
--------+
|[ham, yogurt, light bulbs]   |[beef, oil, detergent, chocolate, candy, berries, 
frankfurter, sausage, coffee, pip fruit, white bread, salty snack, domestic eggs,
root vegetables, bottled beer, specialty bar, long life bakery product, rolls/bun
s, other vegetables, soda, whole milk, canned beer, fruit/vegetable juice, desser
t, newspapers, bottled water, margarine, hamburger meat, pastry, onions, pork, ch
icken, herbs, soft cheese, frozen meals, frozen vegetables, UHT-milk, brown brea
d, citrus fruit, butter, misc. beverages, chewing gum, shopping bags, cream chees
e , waffles, whipped/sour cream, butter milk, hard cheese, napkins, curd, tropica
l fruit]|
|[jam, cocoa drinks, pet care]|[]
|
+-----------------------------+--------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
--------+
```

In [ ]: