

بخش عملی

سوال اول)

در گام اول برای پیاده سازی در spark لازم است آن را نصب کنیم که مطابق با تصویر زیر است:

```

✓ 56s | pip install pyspark

Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
    317.0/317.0 MB 3.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488491 sha256=e55095d936983948e4ae87163be760df7c228ef743ea1a43
  Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38dddce2fdd93be545214a63e02fbd8d74fb0b7f3a6
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.1

```

در بخش بعدی ۵ سطر اول، تعداد سطرها و همچنین حذف ستون مربوط به تاریخ را انجام دهیم، ابتدا پنج سطر اول را با دستور زیر پرینت میکنیم :

✓ 0s | df.show(5)

```

➡ +-----+-----+-----+
  |Member_number|      Date| itemDescription|
  +-----+-----+-----+
  |          1808|21-07-2015| tropical fruit|
  |          2552|05-01-2015|      whole milk|
  |          2300|19-09-2015|      pip fruit|
  |          1187|12-12-2015|other vegetables|
  |          3037|01-02-2015|      whole milk|
  +-----+-----+-----+
  only showing top 5 rows

```

در ادامه تعداد سطرها را خروجی میدهیم و ستون مربوط به تاریخ را حذف میکنیم:

```
✓ [19] print(df.count())
```

38765

```
✓ [20] df = df.drop('Date')
```

در ادامه به سراغ اگریگیت کردن روی آیتم لیست های مربوط به هر شماره کاربر میپردازیم ابتدا بر مبنای آی دی هر کاربر گروه میکنیم و در نهایت روی item هایی که خریده لیست ایجاد میکنیم (همچنین با توجه به fpgrowth که در ادامه استفاده میکنیم لازم است که ست باشد و تکراری نداشته باشد)

```
✓ from pyspark.sql import functions as F

df_grouped = df.groupBy("Member_number").agg(F.collect_set("itemDescription").alias("products"))
df_grouped.show(truncate=False)
```

Member_number	products
1000	[pickled vegetables, whole milk, misc. beverages, pastry, salty snack, sausage, canned beer, semi-finished bread, hygiene article]
1001	[whole milk, beef, sausage, frankfurter, curd, rolls/buns, soda, white bread, whipped/sour cream]
1002	[whole milk, sugar, butter, butter milk, specialty chocolate, frozen vegetables, tropical fruit, other vegetables]
1003	[frozen meals, sausage, detergent, rolls/buns, root vegetables, dental care]
1004	[pastry, whole milk, pip fruit, canned beer, shopping bags, packaged fruit/vegetables, cling film/bags, frozen fish, hygiene article]
1005	[rolls/buns, margarine, whipped/sour cream]
1006	[flour, whole milk, softener, frankfurter, chicken, rice, skin care, bottled water, shopping bags, bottled beer, rolls/buns, chocolate]
1008	[liquor (appetizer), photo/film, liver loaf, yogurt, dessert, domestic eggs, white wine, soda, root vegetables, tropical fruit, other vegetables]
1009	[pastry, canned fish, ketchup, cocoa drinks, yogurt, newspapers, herbs, tropical fruit]
1010	[pip fruit, frankfurter, specialty bar, bottled water, candles, kitchen towels, rolls/buns, UHT-milk, sliced cheese, coffee]
1011	[pastry, whole milk, citrus fruit, curd cheese, yogurt, frankfurter, candles, bottled water, rolls/buns, candy, herbs, grapes, other vegetables]
1012	[whole milk, yogurt, processed cheese, frankfurter, shopping bags, rolls/buns, root vegetables, frozen vegetables, tropical fruit]

در مرحله بعد به سراغ ساخت دیتافریم جدید میرویم که ستون جدیدی هم دارد که product_count است، سپس روی آن فیلتر میزنیم، که تنها آنهایی را برگرداند که تعداد آیتمهای آن از ۱۰ بیشتر است :

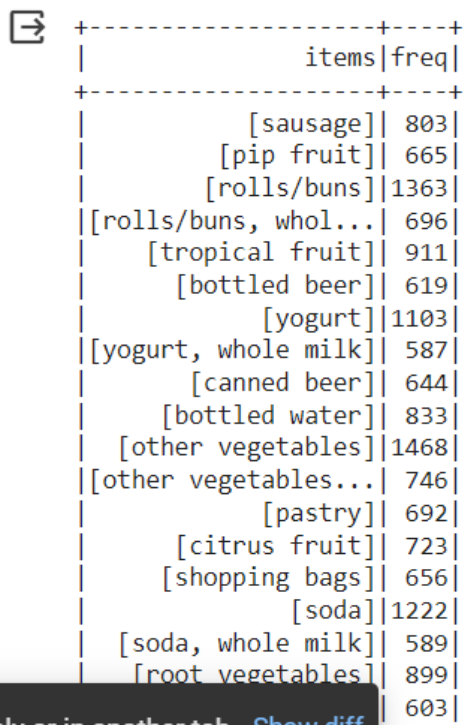
```
✓ df_grouped_with_counts = df_grouped.withColumn("product_count", F.size("products"))
df_grouped_with_counts.filter("product_count > 10").select("Member_number").show()
```

Member_number
1000
1004
1006
1008
1011
1012
1013
1023
1026
1028
1032
1033
1038
1050

در مرحله بعد به کمک فانکشن های **built-in** در اسپارک آن را فیت میکنیم و در نهایت نمایش میدهیم که به صورت زیر است :

```
from pyspark.ml.fpm import FPGrowth

fpGrowth = FPGrowth(itemsCol="products", minSupport=0.15, minConfidence=0.15)
model = fpGrowth.fit(df_grouped)
frequent_itemsets = model.freqItemsets
frequent_itemsets.show()
```



items	freq
[sausage]	803
[pip fruit]	665
[rolls/buns]	1363
[rolls/buns, whole milk]	696
[tropical fruit]	911
[bottled beer]	619
[yogurt]	1103
[yogurt, whole milk]	587
[canned beer]	644
[bottled water]	833
[other vegetables]	1468
[other vegetables...]	746
[pastry]	692
[citrus fruit]	723
[shopping bags]	656
[soda]	1222
[soda, whole milk]	589
[root vegetables]	899
[other vegetables...]	603

در نهایت به سراغ قوانین میرویم که دوباره از ویژگی خود مدل اسپارک استفاده میکنیم و آن هایی که کانفیدنس بالای 0.4 دارند و آنها را نمایش میدهیم :

```
[ ] rules = model.associationRules
rules.filter(rules.confidence >= 0.4).show()
```

antecedent	consequent	confidence	lift	support
[other vegetables]	[whole milk]	0.5081743869209809	1.1091062487222754	0.1913801949717804
[yogurt]	[whole milk]	0.5321849501359928	1.1615100423460805	0.15059004617752694
[rolls/buns]	[whole milk]	0.5106382978723404	1.1144838102499344	0.17855310415597742
[whole milk]	[other vegetables]	0.4176931690929451	1.1091062487222754	0.1913801949717804
[soda]	[whole milk]	0.48199672667757776	1.0519726990980953	0.15110312981015905

تمامی کدها در فایل نوتبوک موجود هستند.