Importing essential modules

In [1]:

```
import findspark
findspark.init()
```

In [2]:

```
from pyspark.sql.functions import split
```

In [3]:

```
from pyspark.ml.fpm import FPGrowth
from pyspark import SparkContext, since
```

In [4]:

```
from pyspark.sql import SQLContext as sc
```

In [5]:

```
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
```

Creating a spark context and starting a session

In [6]:

```
sc = SparkContext.getOrCreate()
spark = SparkSession(sc)
```

Reading the input data

In [7]:

```
 lines = sc.textFile("F:\Docs\Big data\Assignment\Assignmnet 4\Dataset\kosarak.dat")
```

Converting the input data into id and transactions as a list

In [8]:

```
data = []
i = 0
for line in lines.collect():
    data.append((i,list(set(map(lambda x : int(x) ,str(line).split())))))
    i = i + 1
print("The total number of transactions are : ",len(data))
```

```
The total number of transactions are :  990002
```

Creating a sparks dataframe

In [17]:

```
df = spark.createDataFrame(data, ["id", "items"])
```

In [18]:

```
print(df.show())
```

```
+---+--------------------+
| id|               items|
+---+--------------------+
|  0|           [1, 2, 3]|
|  0|                 [1]|
|  0|        [4, 5, 6, 7]|
|  0|              [8, 1]|
|  0|             [9, 10]|
|  0|[6, 11, 12, 13, 1...|
|  0|           [1, 3, 7]|
|  0|            [17, 18]|
|  0|[6, 11, 19, 20, 2...|
|  0|          [1, 3, 25]|
|  0|             [26, 3]|
|  0|[32, 33, 34, 3, 3...|
|  0|          [38, 2, 6]|
|  0|[1, 3, 6, 7, 11, ...|
|  0|       [3, 52, 53, 6]|
|  0|       [1, 55, 54, 6]|
|  0|[64, 6, 11, 56, 5...|
|  0|                 [3]|
|  0|[65, 1, 66, 67, 6...|
|  0|       [1, 11, 69, 6]|
+---+--------------------+
only showing top 20 rows

None
```

Aliasing the inbuild FPGrowth function as fpgrwoth and setting the minimumsupport to be 0.05 (that is around 50000) and min confidence as 0.75

In [19]:

```
fpGrowth = FPGrowth(itemsCol="items", minSupport=0.05, minConfidence=0.75)
```

Fitting our model

In [20]:

```
model = fpGrowth.fit(df)
```

Printing frequent item list

In [21]:

```
print("The frequent Item set is : ")
model.freqItemsets.show()
```

```
The frequent Item set is :
+-----------------+------+
|            items|  freq|
+-----------------+------+
|            [148]| 69922|
|        [148, 11]| 55759|
|     [148, 11, 6]| 55230|
|       [148, 218]| 58823|
|   [148, 218, 11]| 50098|
|[148, 218, 11, 6]| 49866|
|    [148, 218, 6]| 56838|
|         [148, 6]| 64750|
|              [6]|601374|
|              [3]|450031|
|           [3, 6]|265180|
|             [55]| 65412|
|             [11]|364065|
|          [11, 3]|161286|
|       [11, 3, 6]|143682|
|          [11, 6]|324013|
|              [1]|197522|
|          [1, 11]| 91882|
|        [1, 11, 6]| 86092|
|            [1, 3]| 84660|
+------------------+------+
only showing top 20 rows
```

Printing association rules

In [22]:

```
print("The Association rule is : ")
model.associationRules.show()
```

```
The Association rule is : 
+--------------+----------+------------------+------------------+
|    antecedent|consequent|        confidence|              lift|
+--------------+----------+------------------+------------------+
|           [7]|       [6]| 0.847085088264402|1.3944998146776124|
|  [148, 11, 6]|     [218]|0.9028788701792504|10.088849491356443|
|      [148, 6]|      [11]|0.8529729729729729|2.3194895120079906|
|      [148, 6]|     [218]|0.8778069498069498| 9.808693603950202|
|[148, 218, 11]|       [6]|0.9953690766098447|1.6386098776832714|
|         [218]|       [6]|0.8767127926138287| 1.443273932882492|
|       [7, 11]|       [6]|0.9782913410659845|1.6104959380319184|
|     [148, 11]|       [6]|0.9905127423375598|1.6306152177175417|
|     [148, 11]|     [218]|0.8984737889847378| 10.03962671891542|
|        [7, 6]|      [11]|0.7585246569759544|2.0626561945133663|
|          [11]|       [6]|0.8899866782030681|1.4651258474666247|
|    [148, 218]|      [11]|0.8516736650629856|2.3159563038459776|
|    [148, 218]|       [6]|0.9662546962922667|1.5906808106747825|
|     [218, 11]|     [148]|0.8125405475541715|11.504487388228668|
|     [218, 11]|       [6]|0.9833592837680031| 1.618838954874821|
|          [27]|       [6]|0.8237169711925029|1.3560304384867325|
|       [11, 3]|       [6]|0.8908522748409657| 1.466550821613681|
|         [148]|      [11]|0.7974457252366923| 2.168494260299056|
|         [148]|     [218]|0.8412659820943337| 9.400381552691421|
|         [148]|       [6]|0.9260318640771145|1.5244646384780045|
+--------------+----------+------------------+------------------+
only showing top 20 rows
```

Showing the predicted consequents if any for each transaction

In [23]:

```
print("Examining the input items against all the association rules and summarize the consequents a
s prediction")
model.transform(df).show()
```

```
Examining the input items against all the association rules and summarize the consequents as predi
ction
+---+--------------------+----------+
| id|               items|prediction|
+---+--------------------+----------+
|  0|           [1, 2, 3]|        []|
|  0|                 [1]|        []|
|  0|        [4, 5, 6, 7]|      [11]|
|  0|              [8, 1]|        []|
|  0|             [9, 10]|        []|
|  0|[6, 11, 12, 13, 1...|        []|
|  0|           [1, 3, 7]|       [6]|
|  0|            [17, 18]|        []|
|  0|[6, 11, 19, 20, 2...|        []|
|  0|          [1, 3, 25]|        []|
|  0|             [26, 3]|        []|
|  0|[32, 33, 34, 3, 3...|        []|
|  0|          [38, 2, 6]|        []|
|  0|[1, 3, 6, 7, 11, ...|        []|
|  0|      [3, 52, 53, 6]|        []|
|  0|      [1, 55, 54, 6]|        []|
|  0|[64, 6, 11, 56, 5...|        []|
|  0|                 [3]|        []|
|  0|[65, 1, 66, 67, 6...|        []|
|  0|[65, 1, 66, 67, 6...|          []|
|  0|       [1, 11, 69, 6]|          []|
+---+--------------------+----------+
only showing top 20 rows
```

In [ ]: