KARNATAK LAW SOCIETY’S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

**(APPROVED BY AICTE, NEW DELHI)**



*Course Activity Report*

*Submitted in the partial fulfillment for the academic requirement of 2021-22*

***6’thSemester B.E.***

***In***

***AI and ML LAB***

*Submitted by*

**MANJULA KURANAGI -2GI18CS069**

**NIKITA TERDAL - 2GI18CS080**

**PREETI KAMBLE -2GI18CS100**

**ROHAN MALADKAR -2GI18CS110**

*In the year 2020-2021*

GUIDE : Umesh Kulkarni

**Subject : AI and ML Lab**

**Team Members Details:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **USN** | **Student Name** |
| **1** | **2GI18CS069** | **ManjulaKuranagi** |
| **2** | **2GI18CS100** | **Preetikamble** |
| **3** | **2GI18CS080** | **Nikita Terdal** |
| **4** | **2GI18CS110** | **Rohan Maladkar** |

**Course Seminar report and ppt content**

**Marks allocation:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Batch No. :11 | | | | | |
| 1. | Seminar Title: | Marks Range | USN | | | |
|  |  |  |  |
| 2. | Abstract (PO2) | 0-2 |  |  |  |  |
| 3. | Application of the topic to the course (PO2) | 0-3 |  |  |  |  |
| 4. | Literature survey and its findings (PO2) | 0-4 |  |  |  |  |
| 5. | Methodology, Results and Conclusion (PO1,PO3,PO4) | 0-6 |  |  |  |  |
| 6. | Report and Oral presentation skill (PO9,PO10) | 0-5 |  |  |  |  |
|  | Total | 20 |  |  |  |  |

**\* 20 marks is converted to 10 marks for CGPA calculation**

**1.Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

**2.Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and Engineering sciences.

**3.Design/Development of solutions:**Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**4.Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**5.Modern tool usage:**Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**6.The engineer and society:**Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**7.Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need

for sustainable development.

**8.Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**9.Individual and team work:** Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

**10.Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**11. Project management and finance:** Demonstrate knowledge and understanding of the engineering management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**12. Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

**TOPIC:**

**IMPLEMENTING FP GROWTH TREE FOR CO–OCCURING WORDS IN TWITTER FEED**

**CONTENTS:**

* **Twitter**
* **FP- growth tree**
* **Algorithm for FP- growth tree**
* **Advantages &Disadvantages**
* **Problem statement**
* **Source code**
* **Output**
* **Conclusion**
* **References**

**Twitter:**

Twitter is a social media platform where 328 million monthly active users microblog (share 280-character updates) with their followers. It’s a cross between instant messaging and blogging—or social messaging, but it’s also been crucial for news reporting, event promotion, marketing and business. Whether you’re a fan or not, Twitter has become the ninth largest social network in the world and Cortex, Twitter’s in-house engineering team, has turned to the power of artificial intelligence (AI) to help enhance the platform’s user experience.

One of the ways Twitter uses artificial intelligence is to determine what tweet recommendations to suggest on users’ timelines with the goal of highlighting the most relevant tweets for every individual. Prior to this shift, Twitter would show its users tweets in reverse chronological order. Today, the algorithms scan and score thousands of tweets per second to rank them for every user’s feed.

Artificial intelligence tools are also supporting small tweaks to Twitter that improve the overall user experience. For example, its image cropping tools are using AI to automatically crop images in a much more appealing way and are monitoring live video feeds and categorising them based on subject matter to improve their searchability and to help the algorithms identify videos users might be interested in seeing in their feeds.

**FP-Growth Tree:**

A FP-tree is a compact data structure that represents the data set in tree form. Each transaction is read and then mapped onto a path in the FP-tree. This is done until all transactions have been read. Different transactions that have common subsets allow the tree to remain compact because their paths overlap.

The construction of a FP-tree is subdivided into three major steps.

1.Scan the data set to determine the support count of each item, discard the infrequent items and sort the frequent items in decreasing order.

2.Scan the data set one transaction at a time to create the FP-tree. For each transaction:

a)If it is a unique transaction form a new path and set the counter for each node to 1.

b)If it shares a common prefix itemset then increment the common itemset node counters and create new nodes if needed.

3.Continue this until each transaction has been mapped unto the tree.

**FP-Growth Algorithm:**

Fp Growth Algorithm (Frequent pattern growth). FP growth algorithm is an improvement of apriori algorithm. FP growth algorithm used for finding frequent itemset in a transaction database without candidate generation.

FP growth represents frequent items in frequent pattern trees or FP-tree.

Advantages of FP growth algorithm:-

1. Faster than apriori algorithm

2. No candidate generation

3. Only two passes over dataset

Disadvantages of FP growth algorithm:-

1. FP tree may not fit in memory

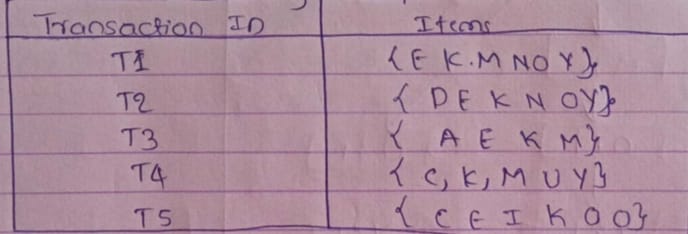
2. FP tree is expensive to build

**FP-Growth Algorithm:**

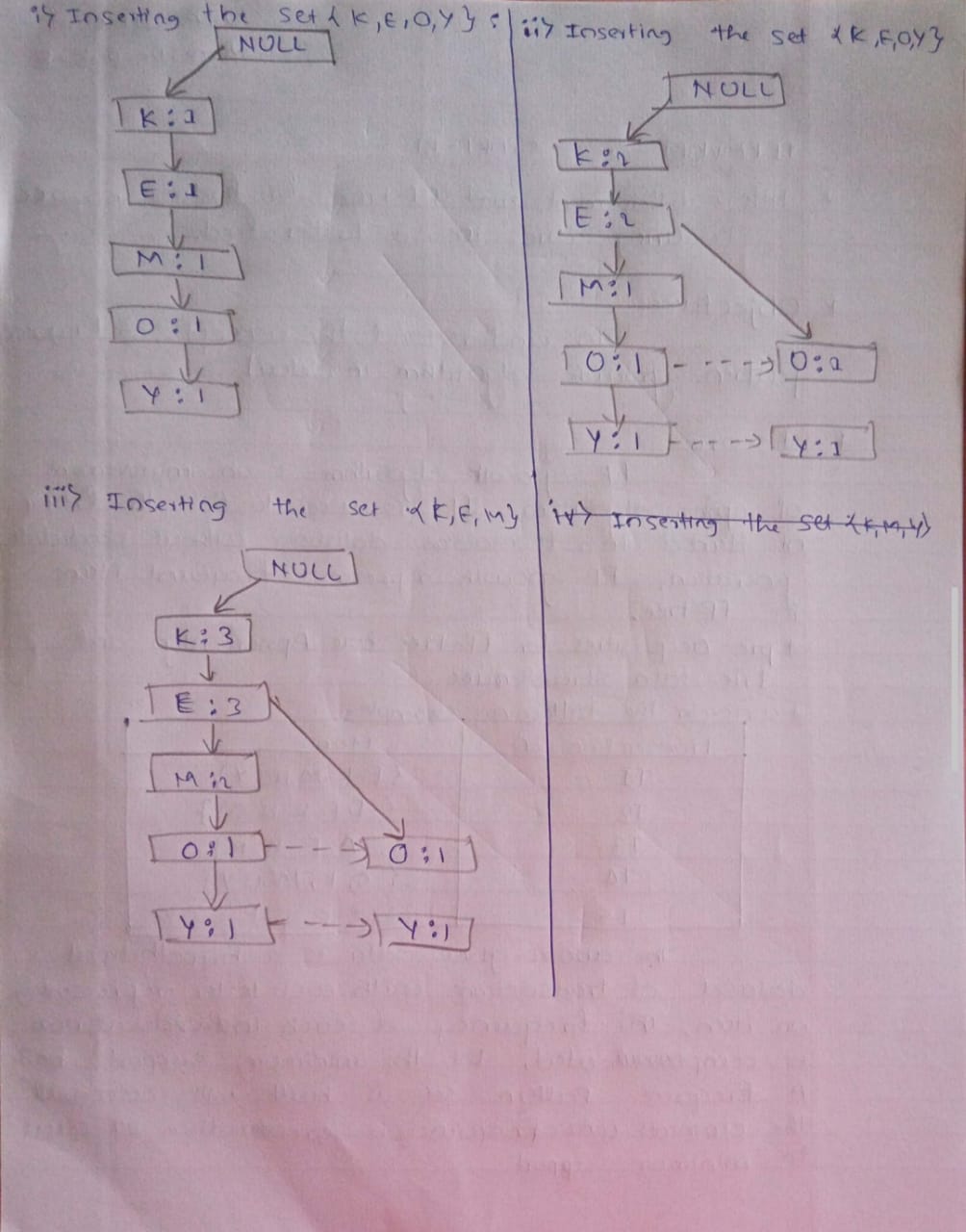
FP growth algorithm algorithm is an improvement of apriori algorithm. It is used to find the frequent itemset in a transaction database without candidate generation. FP growth represents frequent items in FP-tree.

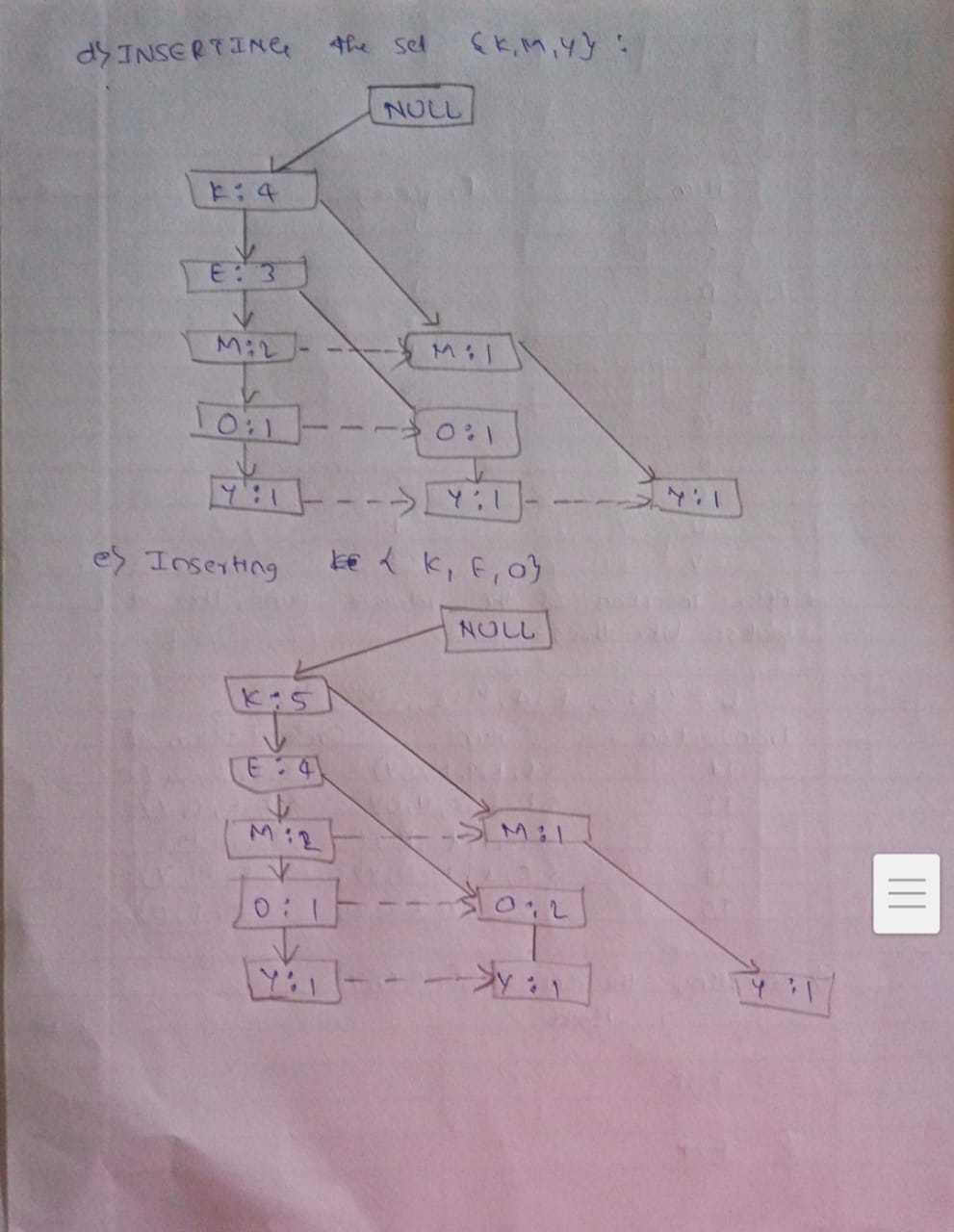
\*Pre-requisities for FP-tree are Apriori algorithm and trie-data structure.

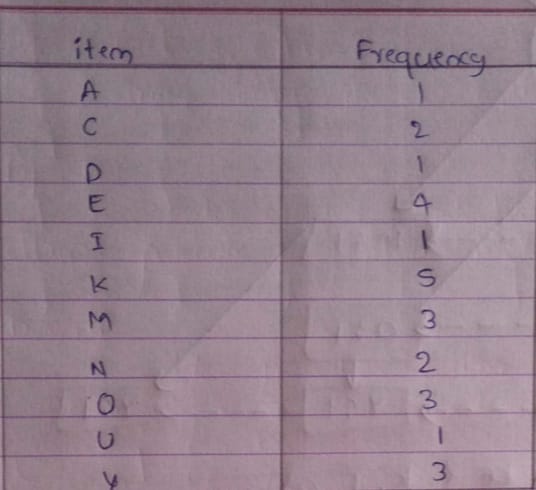
\*Consider the following example:



The above given data is a hypothetical dataset of transactions with each letter representively an item. The frequency of each individual item is computed. Let the minimum support be 3. A frequent pattern set is built will contain all the elements whose frequency is greater than or equal to minimum support.







\*After insertion of the revelant items, the set L looks: Like this-

L={K:5, E:4, M:3, O:3, Y:3}

Transaction ID Items Ordered item set

T1 {E,K,M,N,O,Y} {K,E,M,O,Y}

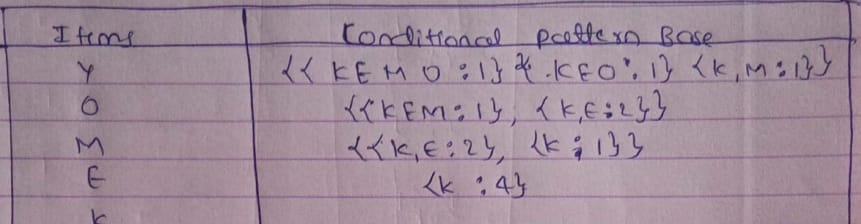
T2 {D,E,K,N,O,Y} {K,E,O,Y}

T3 {A,K,E,M} {K,E,M}

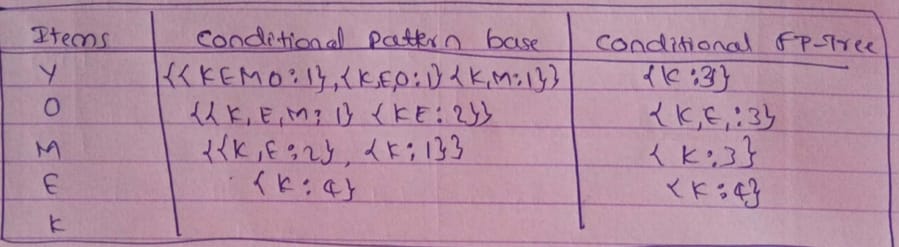
T4 {C,K,M,U,Y} {K,M,Y}

T5 {C,E,I,K,O,O} {K,E,O}

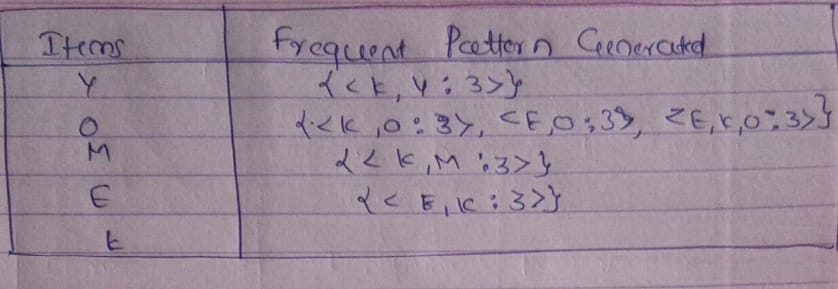
Now for each item, the conditional pattern base is computed which is path labels of all the paths which lead to any node of the given item in the frequent pattern tree.



Now for each item the conditional frequent pattern Tree built. It is done by taking set of element which is common in all paths. In the conditional pattern base of that item and calculating its support count by summing the support counts of all paths in the conditional pattern base.



From the conditional FP tree the FP rules are generated by pairing the items of each conditional FP-tree set corresponding to the item is given in below table.



**Problem Statement:**

**Implement FP Growth tree for co-occuring words in Twitter field.**

**Source Code:**

**{**

**"cells": [**

**{**

**"cell\_type": "code",**

**"execution\_count": 8,**

**"metadata": {},**

**"outputs": [],**

**"source": [**

**"#!pip install fpgrowth\_py\n",**

**"from fpgrowth\_py import fpgrowth\n",**

**"#!pip install mlxtend"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 1,**

**"metadata": {},**

**"outputs": [],**

**"source": [**

**"dataset=[['google','canon','intel','adobe','microsoft','ibm',],\n",**

**" ['imb','bmw','audi','canon','adobe','google'],\n",**

**" ['audi','apple','hyundai','bmw'],\n",**

**" ['oracle','yahoo','twite','canon','google'],\n",**

**" ['hyundai','ibm','audi','oracle','apple','twite']]\n",**

**" "**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 2,**

**"metadata": {},**

**"outputs": [**

**{**

**"name": "stderr",**

**"output\_type": "stream",**

**"text": [**

**"/home/rohan/snap/jupyter/common/lib/python3.7/site-packages/joblib/\_multiprocessing\_helpers.py:45: UserWarning: [Errno 13] Permission denied. joblib will operate in serial mode\n",**

**" warnings.warn('%s. joblib will operate in serial mode' % (e,))\n"**

**]**

**},**

**{**

**"data": {**

**"text/plain": [**

**"array([[ True, False, False, False, True, True, False, True, False,\n",**

**" True, True, False, False, False],\n",**

**" [ True, False, True, True, True, True, False, False, True,\n",**

**" False, False, False, False, False],\n",**

**" [False, True, True, True, False, False, True, False, False,\n",**

**" False, False, False, False, False],\n",**

**" [False, False, False, False, True, True, False, False, False,\n",**

**" False, False, True, True, True],\n",**

**" [False, True, True, False, False, False, True, True, False,\n",**

**" False, False, True, True, False]])"**

**]**

**},**

**"execution\_count": 2,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"import pandas as pd\n",**

**"from mlxtend.preprocessing import TransactionEncoder\n",**

**"te=TransactionEncoder()\n",**

**"te\_ary=te.fit(dataset).transform(dataset)\n",**

**"te\_ary"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 3,**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/html": [**

**"<div>\n",**

**"<style scoped>\n",**

**" .dataframe tbody tr th:only-of-type {\n",**

**" vertical-align: middle;\n",**

**" }\n",**

**"\n",**

**" .dataframe tbody tr th {\n",**

**" vertical-align: top;\n",**

**" }\n",**

**"\n",**

**" .dataframe thead th {\n",**

**" text-align: right;\n",**

**" }\n",**

**"</style>\n",**

**"<table border=\"1\" class=\"dataframe\">\n",**

**" <thead>\n",**

**" <tr style=\"text-align: right;\">\n",**

**" <th></th>\n",**

**" <th>0</th>\n",**

**" <th>1</th>\n",**

**" <th>2</th>\n",**

**" <th>3</th>\n",**

**" <th>4</th>\n",**

**" <th>5</th>\n",**

**" <th>6</th>\n",**

**" <th>7</th>\n",**

**" <th>8</th>\n",**

**" <th>9</th>\n",**

**" <th>10</th>\n",**

**" <th>11</th>\n",**

**" <th>12</th>\n",**

**" <th>13</th>\n",**

**" </tr>\n",**

**" </thead>\n",**

**" <tbody>\n",**

**" <tr>\n",**

**" <th>0</th>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>1</th>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>2</th>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>3</th>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>4</th>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" </tr>\n",**

**" </tbody>\n",**

**"</table>\n",**

**"</div>"**

**],**

**"text/plain": [**

**" 0 1 2 3 4 5 6 7 8 9 \\\n",**

**"0 True False False False True True False True False True \n",**

**"1 True False True True True True False False True False \n",**

**"2 False True True True False False True False False False \n",**

**"3 False False False False True True False False False False \n",**

**"4 False True True False False False True True False False \n",**

**"\n",**

**" 10 11 12 13 \n",**

**"0 True False False False \n",**

**"1 False False False False \n",**

**"2 False False False False \n",**

**"3 False True True True \n",**

**"4 False True True False "**

**]**

**},**

**"execution\_count": 3,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"df=pd.DataFrame(te\_ary)\n",**

**"df\n"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 4,**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/html": [**

**"<div>\n",**

**"<style scoped>\n",**

**" .dataframe tbody tr th:only-of-type {\n",**

**" vertical-align: middle;\n",**

**" }\n",**

**"\n",**

**" .dataframe tbody tr th {\n",**

**" vertical-align: top;\n",**

**" }\n",**

**"\n",**

**" .dataframe thead th {\n",**

**" text-align: right;\n",**

**" }\n",**

**"</style>\n",**

**"<table border=\"1\" class=\"dataframe\">\n",**

**" <thead>\n",**

**" <tr style=\"text-align: right;\">\n",**

**" <th></th>\n",**

**" <th>support</th>\n",**

**" <th>itemsets</th>\n",**

**" </tr>\n",**

**" </thead>\n",**

**" <tbody>\n",**

**" <tr>\n",**

**" <th>0</th>\n",**

**" <td>0.6</td>\n",**

**" <td>(5)</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>1</th>\n",**

**" <td>0.6</td>\n",**

**" <td>(4)</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>2</th>\n",**

**" <td>0.6</td>\n",**

**" <td>(2)</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>3</th>\n",**

**" <td>0.6</td>\n",**

**" <td>(4, 5)</td>\n",**

**" </tr>\n",**

**" </tbody>\n",**

**"</table>\n",**

**"</div>"**

**],**

**"text/plain": [**

**" support itemsets\n",**

**"0 0.6 (5)\n",**

**"1 0.6 (4)\n",**

**"2 0.6 (2)\n",**

**"3 0.6 (4, 5)"**

**]**

**},**

**"execution\_count": 4,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"from mlxtend.frequent\_patterns import fpgrowth\n",**

**"\n",**

**"fpgrowth(df, min\_support=0.6)"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 5,**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/html": [**

**"<div>\n",**

**"<style scoped>\n",**

**" .dataframe tbody tr th:only-of-type {\n",**

**" vertical-align: middle;\n",**

**" }\n",**

**"\n",**

**" .dataframe tbody tr th {\n",**

**" vertical-align: top;\n",**

**" }\n",**

**"\n",**

**" .dataframe thead th {\n",**

**" text-align: right;\n",**

**" }\n",**

**"</style>\n",**

**"<table border=\"1\" class=\"dataframe\">\n",**

**" <thead>\n",**

**" <tr style=\"text-align: right;\">\n",**

**" <th></th>\n",**

**" <th>adobe</th>\n",**

**" <th>apple</th>\n",**

**" <th>audi</th>\n",**

**" <th>bmw</th>\n",**

**" <th>canon</th>\n",**

**" <th>google</th>\n",**

**" <th>hyundai</th>\n",**

**" <th>ibm</th>\n",**

**" <th>imb</th>\n",**

**" <th>intel</th>\n",**

**" <th>microsoft</th>\n",**

**" <th>oracle</th>\n",**

**" <th>twite</th>\n",**

**" <th>yahoo</th>\n",**

**" </tr>\n",**

**" </thead>\n",**

**" <tbody>\n",**

**" <tr>\n",**

**" <th>0</th>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>1</th>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>2</th>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>3</th>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>4</th>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>False</td>\n",**

**" <td>True</td>\n",**

**" <td>True</td>\n",**

**" <td>False</td>\n",**

**" </tr>\n",**

**" </tbody>\n",**

**"</table>\n",**

**"</div>"**

**],**

**"text/plain": [**

**" adobe apple audi bmw canon google hyundai ibm imb intel \\\n",**

**"0 True False False False True True False True False True \n",**

**"1 True False True True True True False False True False \n",**

**"2 False True True True False False True False False False \n",**

**"3 False False False False True True False False False False \n",**

**"4 False True True False False False True True False False \n",**

**"\n",**

**" microsoft oracle twite yahoo \n",**

**"0 True False False False \n",**

**"1 False False False False \n",**

**"2 False False False False \n",**

**"3 False True True True \n",**

**"4 False True True False "**

**]**

**},**

**"execution\_count": 5,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"df=pd.DataFrame(te\_ary,columns=te.columns\_)\n",**

**"df"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 6,**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/html": [**

**"<div>\n",**

**"<style scoped>\n",**

**" .dataframe tbody tr th:only-of-type {\n",**

**" vertical-align: middle;\n",**

**" }\n",**

**"\n",**

**" .dataframe tbody tr th {\n",**

**" vertical-align: top;\n",**

**" }\n",**

**"\n",**

**" .dataframe thead th {\n",**

**" text-align: right;\n",**

**" }\n",**

**"</style>\n",**

**"<table border=\"1\" class=\"dataframe\">\n",**

**" <thead>\n",**

**" <tr style=\"text-align: right;\">\n",**

**" <th></th>\n",**

**" <th>support</th>\n",**

**" <th>itemsets</th>\n",**

**" </tr>\n",**

**" </thead>\n",**

**" <tbody>\n",**

**" <tr>\n",**

**" <th>0</th>\n",**

**" <td>0.6</td>\n",**

**" <td>(audi)</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>1</th>\n",**

**" <td>0.6</td>\n",**

**" <td>(canon)</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>2</th>\n",**

**" <td>0.6</td>\n",**

**" <td>(google)</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>3</th>\n",**

**" <td>0.6</td>\n",**

**" <td>(google, canon)</td>\n",**

**" </tr>\n",**

**" </tbody>\n",**

**"</table>\n",**

**"</div>"**

**],**

**"text/plain": [**

**" support itemsets\n",**

**"0 0.6 (audi)\n",**

**"1 0.6 (canon)\n",**

**"2 0.6 (google)\n",**

**"3 0.6 (google, canon)"**

**]**

**},**

**"execution\_count": 6,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"from mlxtend.frequent\_patterns import apriori\n",**

**"frequent\_itemsets=apriori(df,min\_support=0.6,use\_colnames=True)\n",**

**"frequent\_itemsets"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 7,**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/html": [**

**"<div>\n",**

**"<style scoped>\n",**

**" .dataframe tbody tr th:only-of-type {\n",**

**" vertical-align: middle;\n",**

**" }\n",**

**"\n",**

**" .dataframe tbody tr th {\n",**

**" vertical-align: top;\n",**

**" }\n",**

**"\n",**

**" .dataframe thead th {\n",**

**" text-align: right;\n",**

**" }\n",**

**"</style>\n",**

**"<table border=\"1\" class=\"dataframe\">\n",**

**" <thead>\n",**

**" <tr style=\"text-align: right;\">\n",**

**" <th></th>\n",**

**" <th>antecedents</th>\n",**

**" <th>consequents</th>\n",**

**" <th>antecedent support</th>\n",**

**" <th>consequent support</th>\n",**

**" <th>support</th>\n",**

**" <th>confidence</th>\n",**

**" <th>lift</th>\n",**

**" <th>leverage</th>\n",**

**" <th>conviction</th>\n",**

**" </tr>\n",**

**" </thead>\n",**

**" <tbody>\n",**

**" <tr>\n",**

**" <th>0</th>\n",**

**" <td>(google)</td>\n",**

**" <td>(canon)</td>\n",**

**" <td>0.6</td>\n",**

**" <td>0.6</td>\n",**

**" <td>0.6</td>\n",**

**" <td>1.0</td>\n",**

**" <td>1.666667</td>\n",**

**" <td>0.24</td>\n",**

**" <td>inf</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>1</th>\n",**

**" <td>(canon)</td>\n",**

**" <td>(google)</td>\n",**

**" <td>0.6</td>\n",**

**" <td>0.6</td>\n",**

**" <td>0.6</td>\n",**

**" <td>1.0</td>\n",**

**" <td>1.666667</td>\n",**

**" <td>0.24</td>\n",**

**" <td>inf</td>\n",**

**" </tr>\n",**

**" </tbody>\n",**

**"</table>\n",**

**"</div>"**

**],**

**"text/plain": [**

**" antecedents consequents antecedent support consequent support support \\\n",**

**"0 (google) (canon) 0.6 0.6 0.6 \n",**

**"1 (canon) (google) 0.6 0.6 0.6 \n",**

**"\n",**

**" confidence lift leverage conviction \n",**

**"0 1.0 1.666667 0.24 inf \n",**

**"1 1.0 1.666667 0.24 inf "**

**]**

**},**

**"execution\_count": 7,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"from mlxtend.frequent\_patterns import association\_rules\n",**

**"rules=association\_rules(frequent\_itemsets,metric=\"confidence\",min\_threshold=0.8)\n",**

**"rules"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 8,**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/html": [**

**"<div>\n",**

**"<style scoped>\n",**

**" .dataframe tbody tr th:only-of-type {\n",**

**" vertical-align: middle;\n",**

**" }\n",**

**"\n",**

**" .dataframe tbody tr th {\n",**

**" vertical-align: top;\n",**

**" }\n",**

**"\n",**

**" .dataframe thead th {\n",**

**" text-align: right;\n",**

**" }\n",**

**"</style>\n",**

**"<table border=\"1\" class=\"dataframe\">\n",**

**" <thead>\n",**

**" <tr style=\"text-align: right;\">\n",**

**" <th></th>\n",**

**" <th>antecedents</th>\n",**

**" <th>consequents</th>\n",**

**" <th>antecedent support</th>\n",**

**" <th>consequent support</th>\n",**

**" <th>support</th>\n",**

**" <th>confidence</th>\n",**

**" <th>lift</th>\n",**

**" <th>leverage</th>\n",**

**" <th>conviction</th>\n",**

**" <th>antecedent\_len</th>\n",**

**" </tr>\n",**

**" </thead>\n",**

**" <tbody>\n",**

**" <tr>\n",**

**" <th>0</th>\n",**

**" <td>(google)</td>\n",**

**" <td>(canon)</td>\n",**

**" <td>0.6</td>\n",**

**" <td>0.6</td>\n",**

**" <td>0.6</td>\n",**

**" <td>1.0</td>\n",**

**" <td>1.666667</td>\n",**

**" <td>0.24</td>\n",**

**" <td>inf</td>\n",**

**" <td>1</td>\n",**

**" </tr>\n",**

**" <tr>\n",**

**" <th>1</th>\n",**

**" <td>(canon)</td>\n",**

**" <td>(google)</td>\n",**

**" <td>0.6</td>\n",**

**" <td>0.6</td>\n",**

**" <td>0.6</td>\n",**

**" <td>1.0</td>\n",**

**" <td>1.666667</td>\n",**

**" <td>0.24</td>\n",**

**" <td>inf</td>\n",**

**" <td>1</td>\n",**

**" </tr>\n",**

**" </tbody>\n",**

**"</table>\n",**

**"</div>"**

**],**

**"text/plain": [**

**" antecedents consequents antecedent support consequent support support \\\n",**

**"0 (google) (canon) 0.6 0.6 0.6 \n",**

**"1 (canon) (google) 0.6 0.6 0.6 \n",**

**"\n",**

**" confidence lift leverage conviction antecedent\_len \n",**

**"0 1.0 1.666667 0.24 inf 1 \n",**

**"1 1.0 1.666667 0.24 inf 1 "**

**]**

**},**

**"execution\_count": 8,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"rules[\"antecedent\_len\"]=rules[\"antecedents\"].apply(lambda x:len(x))\n",**

**"rules\n"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 9,**

**"metadata": {},**

**"outputs": [**

**{**

**"name": "stdout",**

**"output\_type": "stream",**

**"text": [**

**"3.01 ms ± 74.1 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)\n"**

**]**

**}**

**],**

**"source": [**

**"from mlxtend.frequent\_patterns import apriori\n",**

**"%timeit apriori(df,min\_support=0.6)"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": null,**

**"metadata": {},**

**"outputs": [],**

**"source": []**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": null,**

**"metadata": {},**

**"outputs": [],**

**"source": []**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": null,**

**"metadata": {},**

**"outputs": [],**

**"source": []**

**}**

**],**

**"metadata": {**

**"kernelspec": {**

**"display\_name": "Python 3",**

**"language": "python",**

**"name": "python3"**

**},**

**"language\_info": {**

**"codemirror\_mode": {**

**"name": "ipython",**

**"version": 3**

**},**

**"file\_extension": ".py",**

**"mimetype": "text/x-python",**

**"name": "python",**

**"nbconvert\_exporter": "python",**

**"pygments\_lexer": "ipython3",**

**"version": "3.7.3"**

**}**

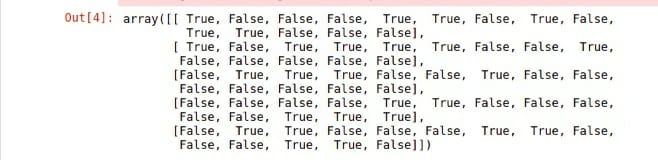
**},**

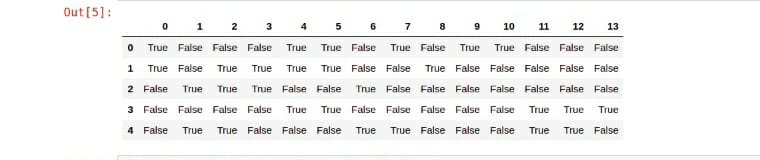
**"nbformat": 4,**

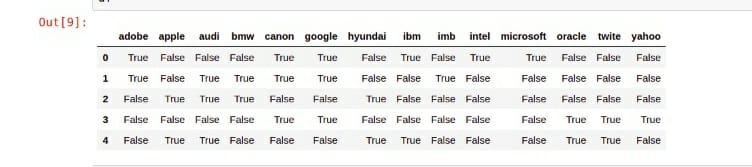
**"nbformat\_minor": 4**

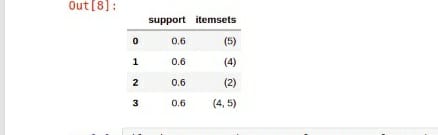
**}**

**OUTPUT:**











**ADVANTAGES & DISADVANTAGES:**

* Advantages:
* Faster than apriori algorithm
* No canididate generation
* Only two passes over dataset
* Disadvantages:
* It may not fit in memory
* FP tree is extensive to build

**CONCLUSION:**

* Understood the FP growth tree construction &algorithm
* Understood how to implement the FP growth tree to find co-occuring words in twitter feed
* Understood the concept of Frequent pattern tree and we implemented that successfully.

**REFERENCES:**

[https://www.geeksforgeeks.org](https://www.geeksforgeeks.org/)>

[http://www.lastnightstudy.com](http://www.lastnightstudy.com/)

[https://www.mygreatlearning.com](https://www.mygreatlearning.com/)