Name: Shivam Vilas Gaikwad

Email: [shivamgaikwad39@gmail.com](mailto:shivamgaikwad39@gmail.com)

With the progress of the technology of information and the need for extracting useful information regarding the businesses of various people from the provided dataset, data mining and its techniques are used to achieve the goal.

Data mining is the essential process of discovering hidden and interesting patterns from massive amounts of data which is stored in data warehouses, OLAP (on line analytical process), databases and other repositories of information.

This data may reach more than terabytes. As a result, Data mining, also called as knowledge discovery in databases (KDD), includes an integration of techniques from many disciplines such as statistics, neural networks, database technology, machine learning and information retrieval, etc.

Various interesting patterns are extracted at reasonable time by using KDD’s techniques. The KDD process has several steps, which are performed to extract patterns to the user, such as data cleaning, data selection, data transformation, data preprocessing, data mining and pattern evaluation.

It is mining for association rules in a database of sales transactions between items which helps us detect unknown relationships, produce results which can perform the basis for decision making and prediction. The discovery of association rules is divided into two phases, detection of the frequent item sets and generation of association rules.

The problem that is addressed in association mining is finding the correlation among different items from a large set of transactions efficiency.

Mining of frequent item sets is an important phase in association mining which discovers frequent item sets in transaction databases. It is the core in many tasks of data mining that try to find interesting patterns from datasets, such as association rules, episodes, classifiers, clustering and correlation, etc. Many algorithms are proposed to find frequent item sets, but all of them can be catalogued into two classes: candidate generation or pattern growth.

We have used one such algorithm known as ‘Apriori’. It is represented under the candidate generation approach. It generates length (k+1) candidate item sets based on length (k) frequent item sets. The frequency of item sets is defined by counting their occurrence in transactions.

The Apriori algorithm is easy to execute and very simple, and is used to mine all frequent itemsets in the database.

The algorithm makes many searches in the database to find frequent item sets where k-itemsets are used to generate k+1-itemsets. Each k-itemset must be greater than or equal to minimum support threshold to be frequency. Otherwise, it is called candidate itemsets. In the first, the algorithm scans the database to find the frequency of 1-itemsets that contains only one item by counting each item in the database. The frequency of 1-itemsets is used to find the itemsets in 2- itemsets which in turn is used to find 3-itemsets and so on until there are not any more k-itemsets.If an itemset is not frequent, any large subset from it is also non-frequent; this condition prune from search space in database.

However ‘Apriori algorithm’ suffers from some weakness in spite of being clear and simple. The main limitation is costly wasting of time to hold a vast number of candidate sets with much frequent itemsets, low minimum support or large itemsets.

While using the algorithm we had made use of certain tools namely, Pandas; NumPy and matplotlib.pbt

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules, Pandas, Matplotlib, etc. complete the Python Machine Learning Ecosystem.

NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation. NumPy’s main object is the homogeneous multidimensional array. It is a table with same type elements, i.e, integers or string or characters (homogeneous), usually integers. NumPy also provides basic mathematical and statistical functions like mean, min, max, sum, prod, std, var, summation across different axes, transposing of a matrix, etc.

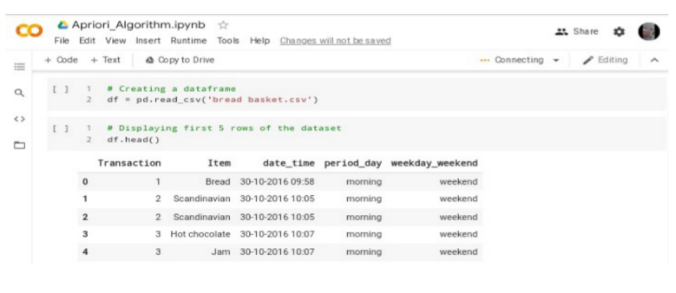
Now, similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike the NumPy library which provides objects for multi-dimensional arrays, Pandas provides an in-memory 2d table object called Dataframe. It is like a spreadsheet with column names and row labels. Hence, with 2d tables, pandas is capable of providing many additional functionalities like creating pivot tables, computing columns based on other columns and plotting graphs. Pandas is a fast and efficient DataFrame object for data manipulation with integrated indexing; it provides tools for reading and writing data between in-memory data structures and different formats: CSV and text files, Microsoft Excel, SQL databases, and the fast HDF5 format. Responsible for intelligent data alignment and integrated handling of missing data: gain automatic label-based alignment in computations and easily manipulate messy data into an orderly form. Allows flexible reshaping and pivoting of data sets;Intelligent label-based slicing, fancy indexing, and subsetting of large data sets and can aggregate or transform data with a powerful group by engine allowing split-apply-combine operations on data sets. Its hierarchical axis indexing provides an intuitive way of working with high-dimensional data in a lower-dimensional data structure. Most importantly it provides time series-functionality: date range generation and frequency conversion, moving window statistics, date shifting and lagging. Even create domain-specific time offsets and join time series without losing data. Hence Python with pandas is in use in a wide variety of academic and commercial domains, including Finance, Neuroscience, Economics, Statistics, Advertising, Web Analytics, and more.

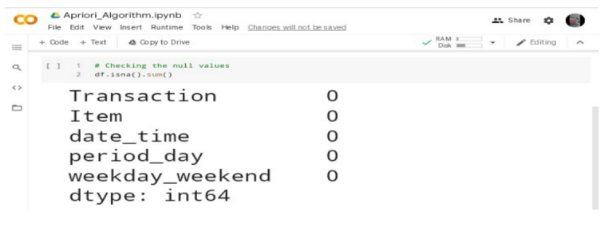
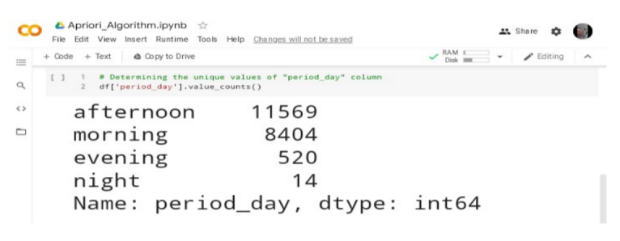
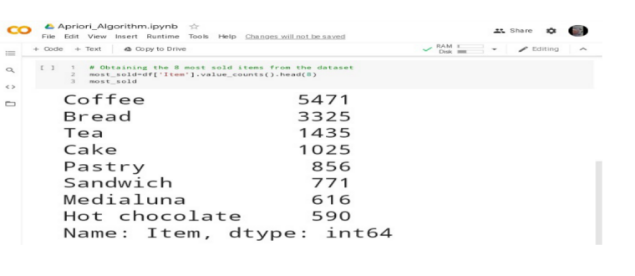
As we know Data visualizations in python can be done via many packages. We’ll be discussing the matplotlib package. Matplotlib is a 2-D plotting library that helps in visualizing figures. Matplotlib emulates Matlab like graphs and visualizations. Matlab is not free, is difficult to scale and as a programming language, is tedious. So, matplotlib in Python is used as it is a robust, free and easy library for data visualization.

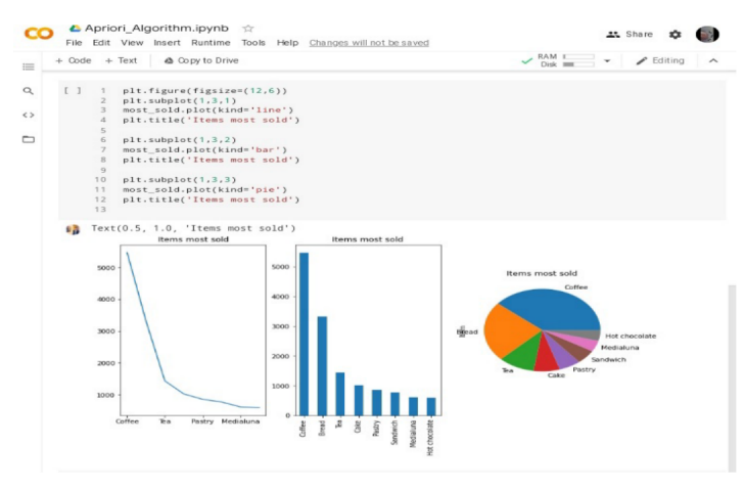
Plotting Matplotlib is quite easy. Generally, while plotting they follow the same steps in each and every plot. Matplotlib has a module called pyplot which aids in plotting figures. Hence Matplotlib is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. In matplotlib various states are preserved across function calls, so that it keeps track of things like the current figure and plotting area, and the plotting functions are directed to the current axes.

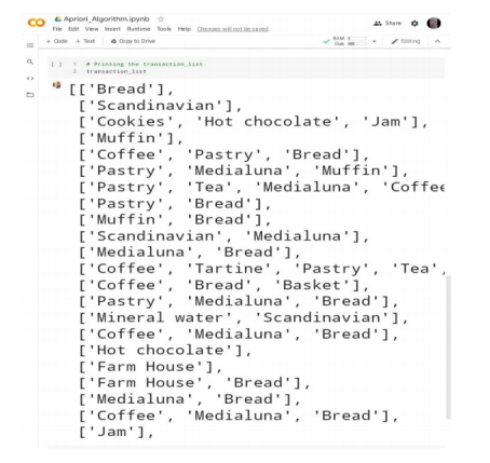
While working with the dataset and using Apriori algorithm we proceeded as follows:

STEP 1- Importing the libraries STEP 2- Creating the dataset/dataframe

STEP 3- Displaying the first five rows

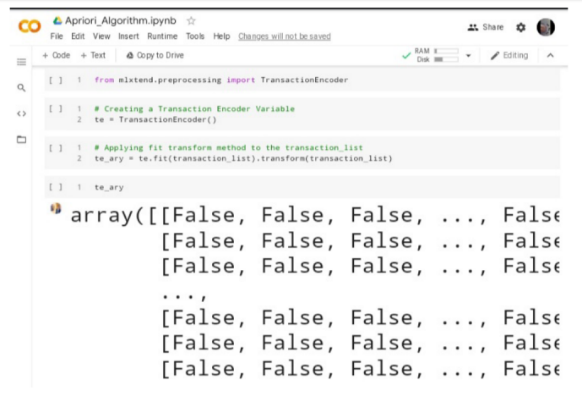
STEP 4- Checking for null values, if anySTEP 5- Listing the unique values of "period\_day" columnSTEP 6- Obtaining the 8 most sold items from the data set

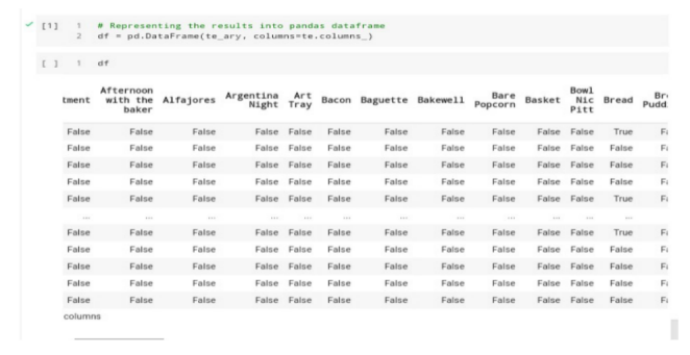
STEP 7- Developing a figure (plot) based on most sold itemsSTEP 8- Converting the item column into the list

STEP 9- Printing of the transaction list

STEP 10- Importing Transaction Encoder

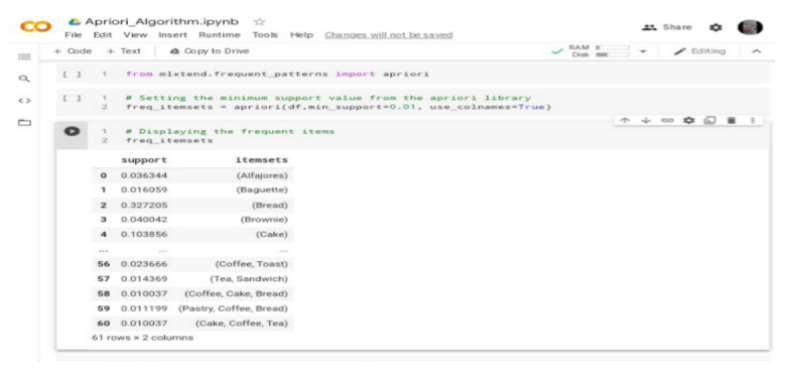
STEP 11- Creating a Transaction Encoder Variable

STEP 12- Applying fit transform on the transaction list.

STEP 13- Representing the results into Pandas data frame.

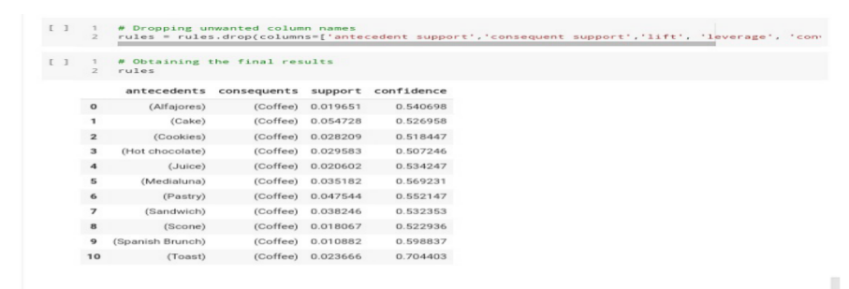
STEP 14- Importing Apriori from mlxtend.frequent\_patterns

STEP 15- Setting the minimum support value from Apriori Library

STEP 16- Displaying of frequent items

STEP 17- Importing association\_rules from mlxtend.frequent\_patterns

STEP 18- Applying the association rules on the dataset by having the confidence %STEP 19- Dropping unwanted column names

STEP 20- Obtaining the final results Link for the Google Colab- https://colab.research.google.com/drive/1uezSIpEhJVSKNUiEZy2pa1RViW3L0yJb?usp=sharing

Link for Tableau Dashboard- https://public.tableau.com/shared/8X76DG829?:display\_count=n&:origin=viz\_share\_link