A

Project Report On

Implementation of Association Rule Mining

Submitted for partial fulfillment of requirement for the award of degree

Of

BACHELOR OF TECHNOLOGY

(COMPUTER SCIENCE AND ENGINEERING)

Of

Assam Down Town University

Session 2018-22

Submitted by : -

Name : - Chandra Shekhar Yadav
Roll no : - 1814010
Registration no : - (ADTU/2018-22/BCS/010)

Name : - Nilratam Sarkar
Roll no : - 1814017
Registration no : - (ADTU/L/2018-22/BCS/017)

Under The Guidence of : -

Eirene Barua, Assistant Professor
Faculty of Engineering and Technology
Computer Science and Engineering

Assam Down Town University

Sankar MadhabPath Gandhi Nagar, Panikhaiti,
781026, Guwahati, Assam,India

Assam Down Town University
Computer Science & Engineering
Faculty of Engineering & Technology
Sankar Madhab Path , Gandhi Nagar,
Panikhaiti , 781026 , Guwahati , Assam , India

CERTIFICATE

This is to certify that A Project titled **"Implementation of Association Rule Mining"** submitted by *Chandra Shekhar Yadav bearing Registration no : - ADTU/2018-22/BCS/010 & Roll no : - 1814010 and Nilratan Sarkar bearing Registration no : -ADTU/L/2018-22/BCS/017 & Roll no : - 1814017* , students of 7th sem , B.Tech C. S . E , carried under my guidance and supervision for the award of Degree Bachelor of Technology in Computer Science & Engineering of *Assam Down Town University* and the work is original and not a copy of any other project.

Date : -

(*Signature of Dean*) (*Signature of Supervisor)*

DECLARATION

We the undersigned solemnly declare that the report of the project work entitled "**Implementation of Association Rule Mining"**, is based on our own work carried out during the course of our study under the supervision of Eirene Barua .

We assert that the statements made and conclusions drawn are an outcome of the project work. We further declare that to the best of our knowledge and belief that the project report does not contain any part of any work which has been submitted for the award of any other degree in this University or any other University.

____

(Signature of the Candidate)

Name : -

Roll No .: -

Enrollment No.: -

ACKNOWLEDGEMENT

Place : -

Date : -

We would like to express our gratitude towards Mrs. Eirene Barua , Assistant Professor, Computer Science & Engineering, Faculty of Engineering & Technology , Assam Down Town University for her support in accomplishment of our project on "**Implementation of Association Rule Mining**" .

I would like to extend my deep appreciation to my fellow group member, without his support and coordination we would not have been able to complete this project.

Name : -

Roll No.: -

Enrollment No.: -

Signature : -

Contents : -

List of Contents Page No.

1 . ABSTRACT : -

Task of extracting useful and interesting knowledge from large data is called data mining. It has many aspects like clustering, classification, association mining, outlier detection, regression etc. Among them association rule mining is one of the important aspect for data mining.

Best example of association rule mining is market-basket analysis. Applications of association rule mining are stock analysis, web log mining, medical diagnosis, customer market analysis bio-informatics etc.

The APriori Data Mining Algorithm is used to create association rules from sets of items. The algorithm finds patterns of items. Algorithm uses knowledge from previous iteration phase to produce frequent item-sets that are frequently associated together. A confidence measure is created for each rule generated from the frequent item-sets.

2 . INTRODUCTION : -

- We in this project will First import a data-set .
- Then convert the data-set into one-hot encoding array ,where each unique label is represented as a column in the new array.
- Then mine frequent item-set from it with given input for minimum support metric
- Then mine association rules using apriori algorithm with metric of support , confidence , lift , leverage .
- Then visualize the output for better understanding of the data.
- At the end we will save the data in a csv file.

3 . AIM & OBJECTIVE :-

- To find frequent item-set from a given data-set.
- Mine association rules from the generated frequent item-set .
- Sort the data on basis of metric concordance and lift with given minimum threshold .
- Visualize the mined association rules for better understanding of the data.
- Then store the data in a csv file .
- The generated data can be used for many things like market basket analysis,ux design, disease predication etc and many more.

4 . PLATFORM : -

❖ **Operating system : -** Microsoft windows.

5 . Technologies used : -

❖ **Python :-** Python is an interpreted high-level general-purpose programming language. Python is often used to build websites and software, automate tasks, and conduct data analysis,data visualization,AI and machine learning.

➢ **Libraries used : -**

1. **Numpy : -** NumPy is Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

2. **Pandas : -**Pandas is a software library written for the Python programming language for data manipulation and analysis.

3. **Mlxtend : -**Mlxtend (machine learning extensions) is a Python library of useful tools for the day-to-day data science tasks.

4. **Matplotlib : -**Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

6 . Software Requirements : -

- **Anaconda :-** Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment.

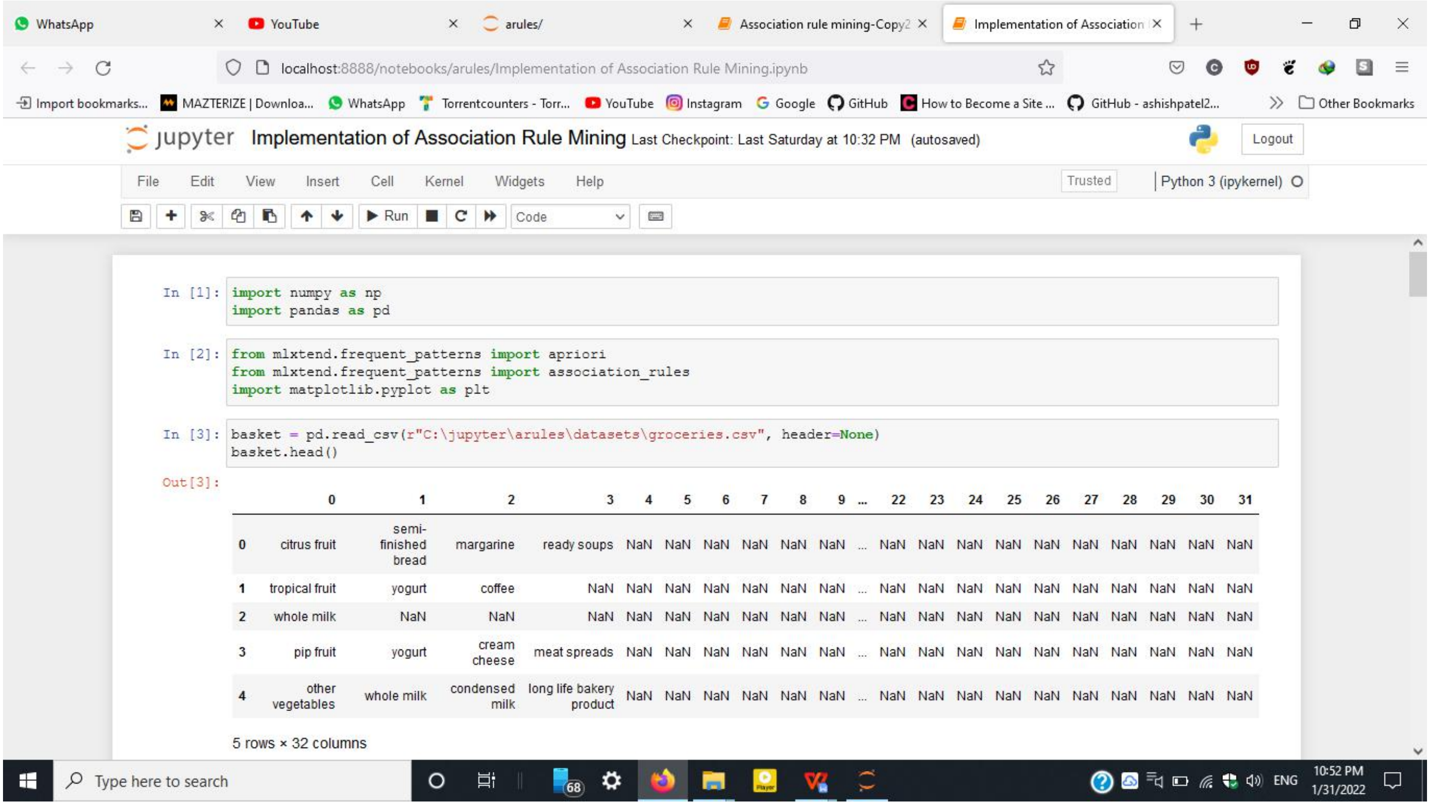
7 . Hardware Requirements : -

- Intel core i3, i5,i7 processor or equivalent or higher.
- 1GB Ram or Higher.
- 20 GB HDD or Higher.
- 1.5GB free space.

8. *Number of outputs corresponding to the given min_support : -*

Serial no.	min_support	Number Of Outputs
01	0.0838	1
02	0.0755	1
03	0.0729	2
04	0.0675	2
05	0.0664	3
06	0.0600	3
07	0.0589	4
08	0.0560	4
09	0.0559	5
10	0.0525	5
11	0.0514	6
12	0.0500	6
13	0.0490	7
14	0.0475	7
15	0.0467	8
16	0.0462	8
17	0.0460	9
18	0.0425	9
19	0.0421	10
20	0.0395	10
21	0.0391	11
22	0.0390	11
23	0.0389	12
24	0.0385	12
25	0.0384	13
26	0.0370	13
27	0.0360	14
28	0.0345	14
29	0.0341	15
30	0.0321	15

9 . EXPERIMENTS(SCREENSHOTS) : -

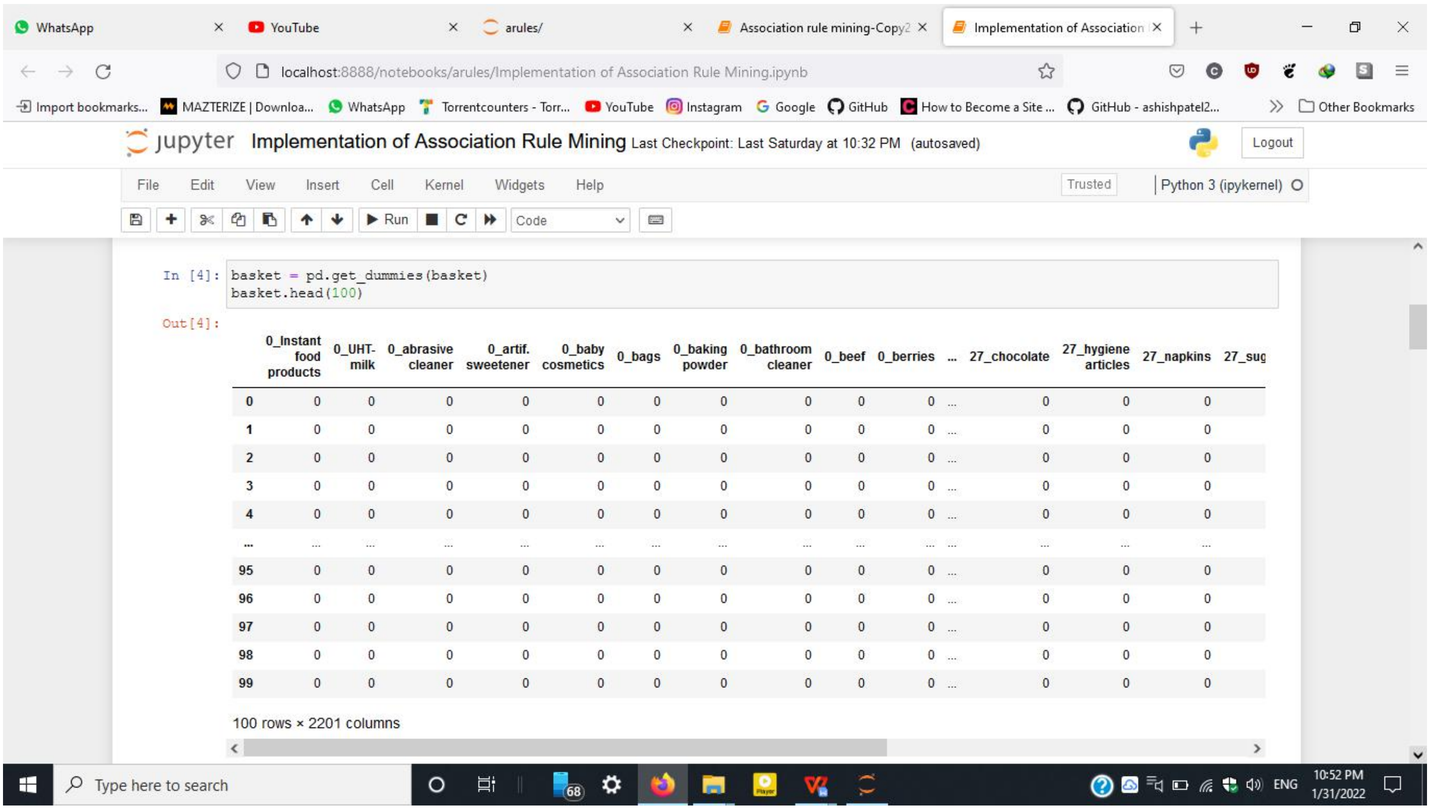



```
In [1]: import numpy as np
        import pandas as pd

In [2]: from mlxtend.frequent_patterns import apriori
        from mlxtend.frequent_patterns import association_rules
        import matplotlib.pyplot as plt

In [3]: basket = pd.read_csv(r"C:\jupyter\arules\datasets\groceries.csv", header=None)
        basket.head()
```

Out[3]:

	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28	29	30	31
0	citrus fruit	semi-finished bread	margarine	ready soups	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	tropical fruit	yogurt	coffee	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	whole milk	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	pip fruit	yogurt	cream cheese	meat spreads	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	other vegetables	whole milk	condensed milk	long life bakery product	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 32 columns




```
In [4]: basket = pd.get_dummies(basket)
        basket.head(100)
```

Out[4]:

	0_Instant food products	0_UHT-milk	0_abrasive cleaner	0_artif. sweetener	0_baby cosmetics	0_bags	0_baking powder	0_bathroom cleaner	0_beef	0_berries	...	27_chocolate	27_hygiene articles	27_napkins	27_sug
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
95	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
96	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
97	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
98	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
99	0	0	0	0	0	0	0	0	0	0	...	0	0	0	

100 rows × 2201 columns

```
In [5]: frequent_itemsets = apriori(basket, min_support=0.007, use_colnames=True)
        frequent_itemsets
```

Out[5]:

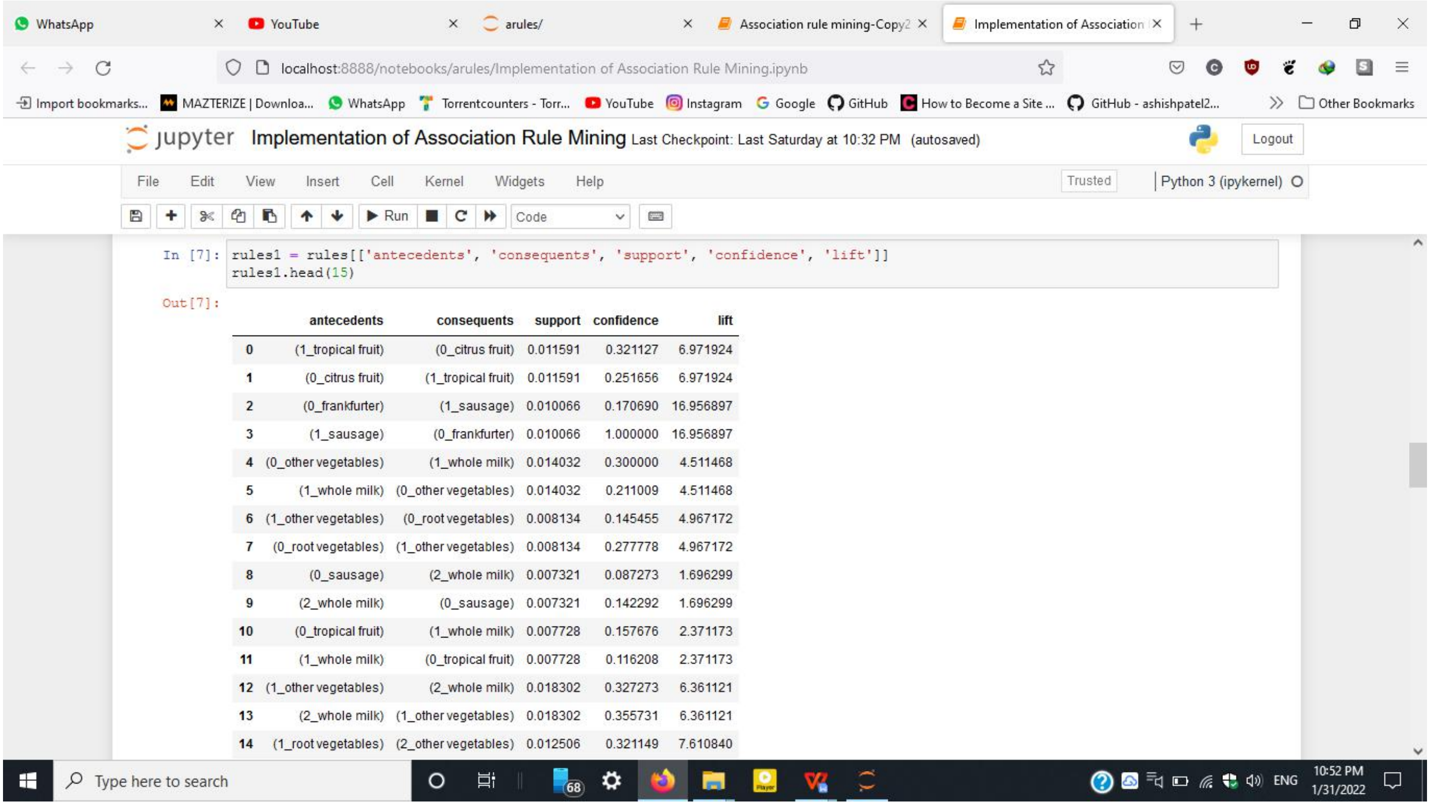
	support	itemsets
0	0.007117	(0_UHT-milk)
1	0.030910	(0_beef)
2	0.012303	(0_berries)
3	0.008134	(0_beverages)
4	0.018099	(0_bottled beer)
...	...	...
148	0.017285	(2_other vegetables, 3_whole milk)
149	0.008846	(2_root vegetables, 3_other vegetables)
150	0.007422	(2_whole milk, 3_yogurt)
151	0.010574	(4_whole milk, 3_other vegetables)
152	0.007016	(4_other vegetables, 5_whole milk)

153 rows × 2 columns

```
In [6]: rules = association_rules(frequent_itemsets, metric="lift", min_threshold=0.8)
        rules.head(15)
```

Out[6]:

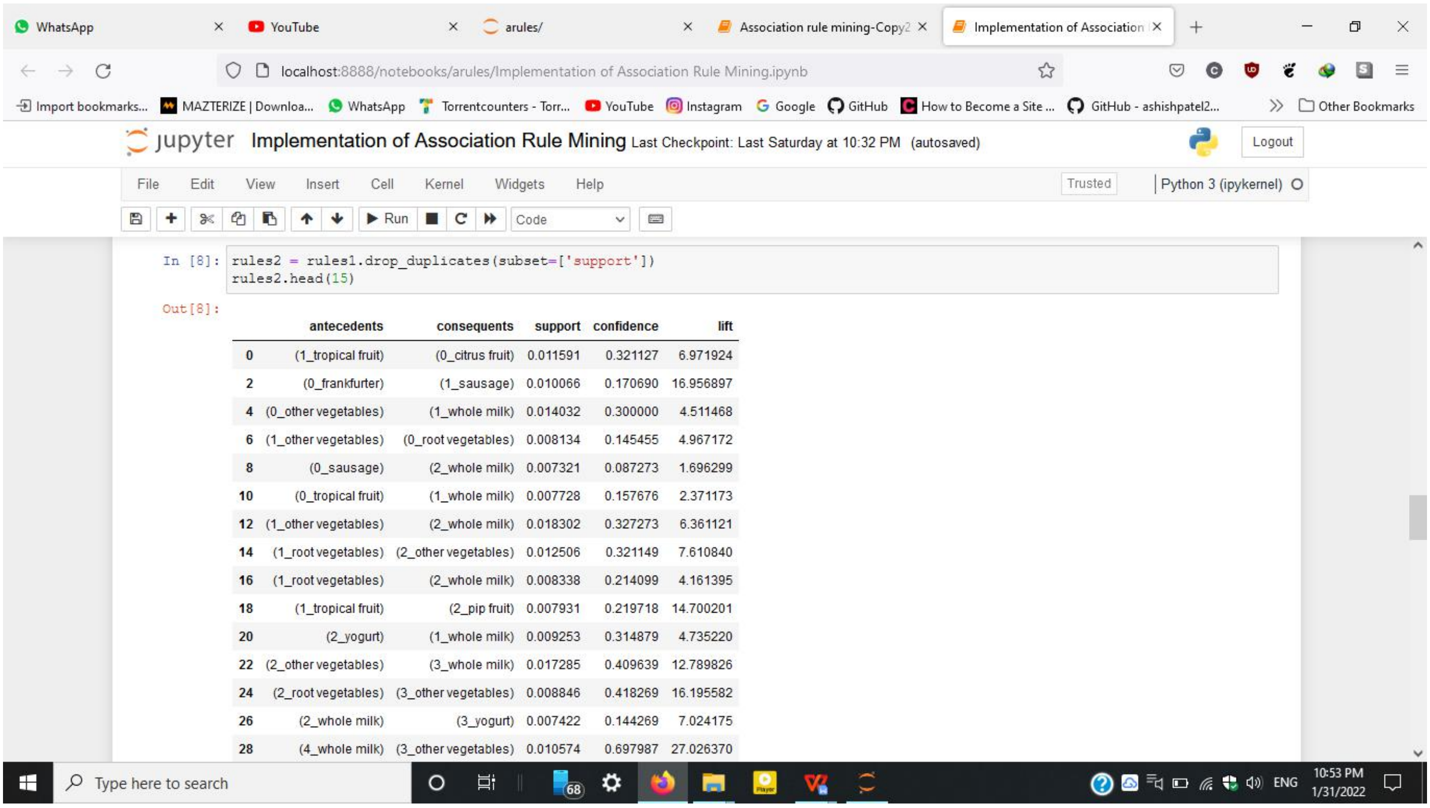
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(1_tropical fruit)	(0_citrus fruit)	0.036096	0.046060	0.011591	0.321127	6.971924	0.009929	1.405181
1	(0_citrus fruit)	(1_tropical fruit)	0.046060	0.036096	0.011591	0.251656	6.971924	0.009929	1.288049
2	(0_frankfurter)	(1_sausage)	0.058973	0.010066	0.010066	0.170690	16.956897	0.009472	1.193683
3	(1_sausage)	(0_frankfurter)	0.010066	0.058973	0.010066	1.000000	16.956897	0.009472	inf
4	(0_other vegetables)	(1_whole milk)	0.046772	0.066497	0.014032	0.300000	4.511468	0.010921	1.333575
5	(1_whole milk)	(0_other vegetables)	0.066497	0.046772	0.014032	0.211009	4.511468	0.010921	1.208161
6	(1_other vegetables)	(0_root vegetables)	0.055923	0.029283	0.008134	0.145455	4.967172	0.006497	1.135945
7	(0_root vegetables)	(1_other vegetables)	0.029283	0.055923	0.008134	0.277778	4.967172	0.006497	1.307184
8	(0_sausage)	(2_whole milk)	0.083884	0.051449	0.007321	0.087273	1.696299	0.003005	1.039249
9	(2_whole milk)	(0_sausage)	0.051449	0.083884	0.007321	0.142292	1.696299	0.003005	1.068098
10	(0_tropical fruit)	(1_whole milk)	0.049009	0.066497	0.007728	0.157676	2.371173	0.004469	1.108247
11	(1_whole milk)	(0_tropical fruit)	0.066497	0.049009	0.007728	0.116208	2.371173	0.004469	1.076035
12	(1_other vegetables)	(2_whole milk)	0.055923	0.051449	0.018302	0.327273	6.361121	0.015425	1.410008
13	(2_whole milk)	(1_other vegetables)	0.051449	0.055923	0.018302	0.355731	6.361121	0.015425	1.465347
14	(1_root vegetables)	(2_other vegetables)	0.038943	0.042196	0.012506	0.321149	7.610840	0.010863	1.410919



```
In [7]: rules1 = rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']]
        rules1.head(15)
```

Out[7]:

	antecedents	consequents	support	confidence	lift
0	(1_tropical fruit)	(0_citrus fruit)	0.011591	0.321127	6.971924
1	(0_citrus fruit)	(1_tropical fruit)	0.011591	0.251656	6.971924
2	(0_frankfurter)	(1_sausage)	0.010066	0.170690	16.956897
3	(1_sausage)	(0_frankfurter)	0.010066	1.000000	16.956897
4	(0_other vegetables)	(1_whole milk)	0.014032	0.300000	4.511468
5	(1_whole milk)	(0_other vegetables)	0.014032	0.211009	4.511468
6	(1_other vegetables)	(0_root vegetables)	0.008134	0.145455	4.967172
7	(0_root vegetables)	(1_other vegetables)	0.008134	0.277778	4.967172
8	(0_sausage)	(2_whole milk)	0.007321	0.087273	1.696299
9	(2_whole milk)	(0_sausage)	0.007321	0.142292	1.696299
10	(0_tropical fruit)	(1_whole milk)	0.007728	0.157676	2.371173
11	(1_whole milk)	(0_tropical fruit)	0.007728	0.116208	2.371173
12	(1_other vegetables)	(2_whole milk)	0.018302	0.327273	6.361121
13	(2_whole milk)	(1_other vegetables)	0.018302	0.355731	6.361121
14	(1_root vegetables)	(2_other vegetables)	0.012506	0.321149	7.610840



```
In [8]: rules2 = rules1.drop_duplicates(subset=['support'])
        rules2.head(15)
```

Out[8]:

	antecedents	consequents	support	confidence	lift
0	(1_tropical fruit)	(0_citrus fruit)	0.011591	0.321127	6.971924
2	(0_frankfurter)	(1_sausage)	0.010066	0.170690	16.956897
4	(0_other vegetables)	(1_whole milk)	0.014032	0.300000	4.511468
6	(1_other vegetables)	(0_root vegetables)	0.008134	0.145455	4.967172
8	(0_sausage)	(2_whole milk)	0.007321	0.087273	1.696299
10	(0_tropical fruit)	(1_whole milk)	0.007728	0.157676	2.371173
12	(1_other vegetables)	(2_whole milk)	0.018302	0.327273	6.361121
14	(1_root vegetables)	(2_other vegetables)	0.012506	0.321149	7.610840
16	(1_root vegetables)	(2_whole milk)	0.008338	0.214099	4.161395
18	(1_tropical fruit)	(2_pip fruit)	0.007931	0.219718	14.700201
20	(2_yogurt)	(1_whole milk)	0.009253	0.314879	4.735220
22	(2_other vegetables)	(3_whole milk)	0.017285	0.409639	12.789826
24	(2_root vegetables)	(3_other vegetables)	0.008846	0.418269	16.195582
26	(2_whole milk)	(3_yogurt)	0.007422	0.144269	7.024175
28	(4_whole milk)	(3_other vegetables)	0.010574	0.697987	27.026370

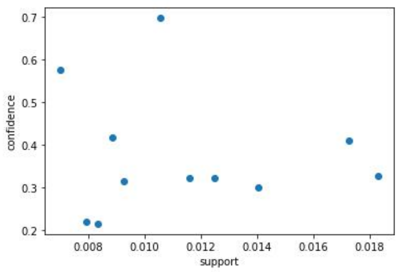
```
In [9]: rules3 = rules2[rules2['confidence'] >=0.20]
        rules3.sort_values('confidence', ascending = False)
```

Out[9]:

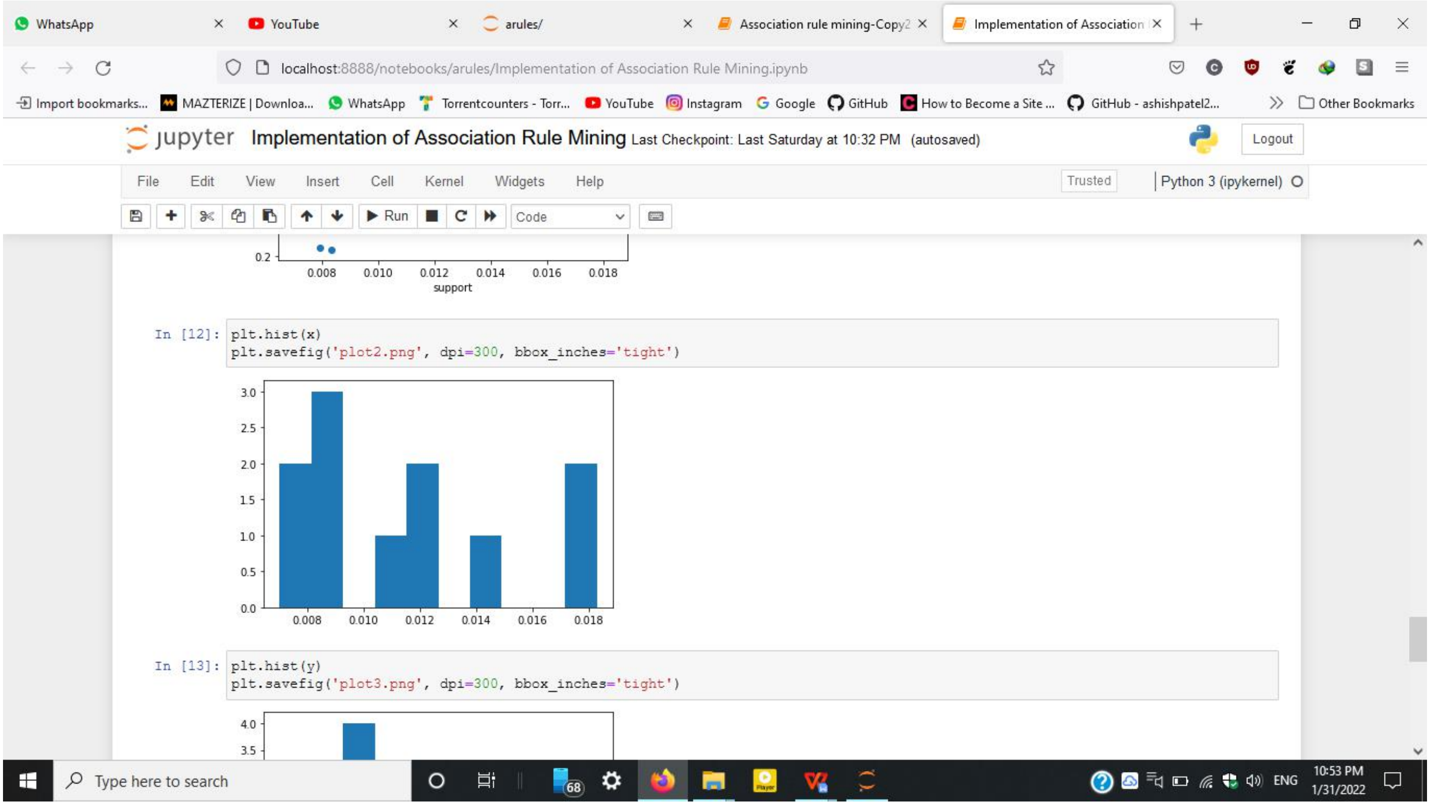
	antecedents	consequents	support	confidence	lift
28	(4_whole milk)	(3_other vegetables)	0.010574	0.697987	27.026370
30	(4_other vegetables)	(5_whole milk)	0.007016	0.575000	65.001437
24	(2_root vegetables)	(3_other vegetables)	0.008846	0.418269	16.195582
22	(2_other vegetables)	(3_whole milk)	0.017285	0.409639	12.789826
12	(1_other vegetables)	(2_whole milk)	0.018302	0.327273	6.361121
14	(1_root vegetables)	(2_other vegetables)	0.012506	0.321149	7.610840
0	(1_tropical fruit)	(0_citrus fruit)	0.011591	0.321127	6.971924
20	(2_yogurt)	(1_whole milk)	0.009253	0.314879	4.735220
4	(0_other vegetables)	(1_whole milk)	0.014032	0.300000	4.511468
18	(1_tropical fruit)	(2_pip fruit)	0.007931	0.219718	14.700201
16	(1_root vegetables)	(2_whole milk)	0.008338	0.214099	4.161395

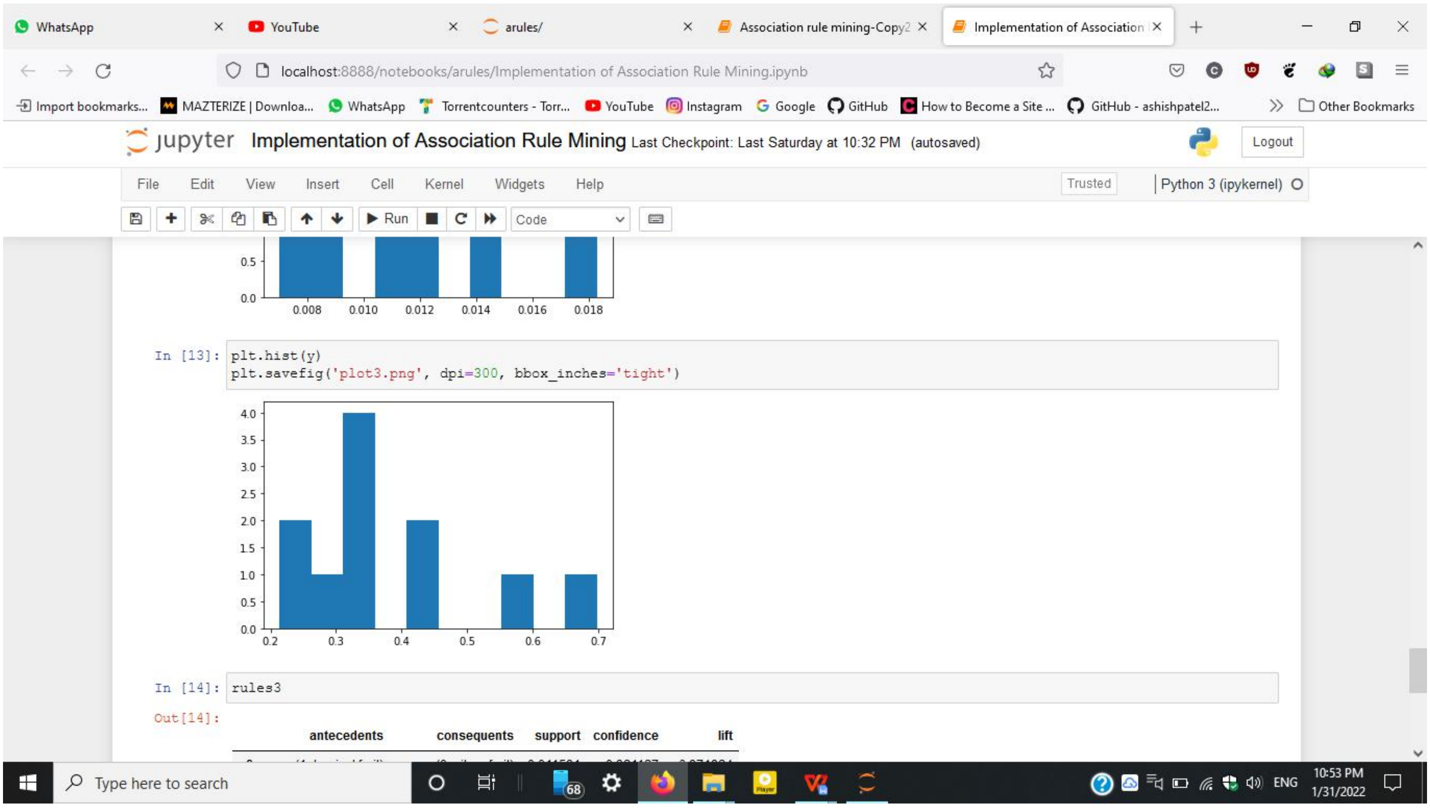
```
In [10]: x = rules3.support
         y = rules3.confidence
```

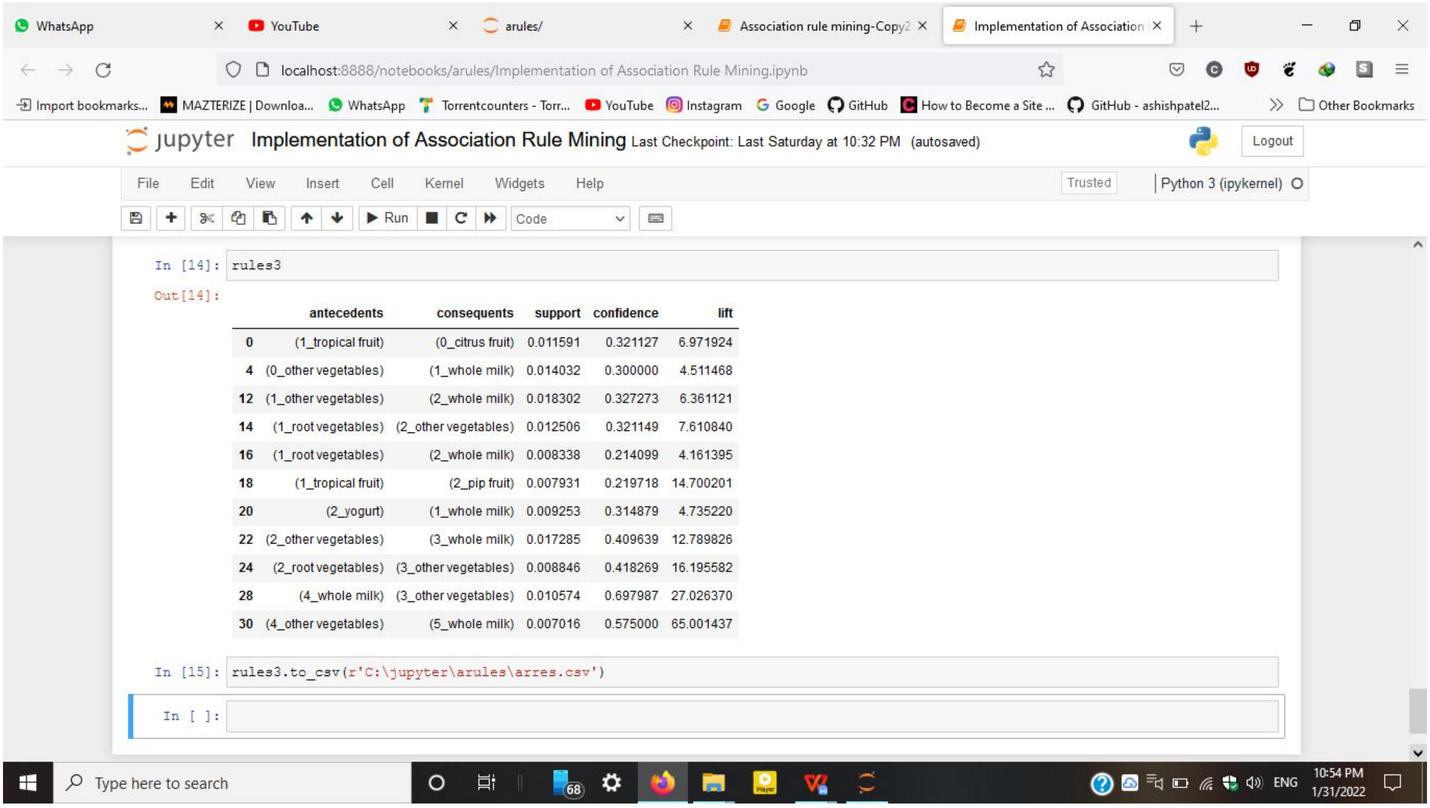
```
In [11]: plt.scatter(x,y)
         plt.xlabel('support')
         plt.ylabel('confidence')
         plt.savefig('plot1.png', dpi=300, bbox_inches='tight')
```




```
In [12]: plt.hist(x)
         plt.savefig('plot2.png', dpi=300, bbox_inches='tight')
```


jupyter Implementation of Association Rule Mining Last Checkpoint: Last Saturday at 10:32 PM (autosaved)
localhost:8888/notebooks/arules/Implementation of Association Rule Mining.ipynb
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
support
In [12]: plt.hist(x)
plt.savefig('plot2.png', dpi=300, bbox_inches='tight')
3.0 2.5 2.0 1.5 1.0 0.5 0.0
0.008 0.010 0.012 0.014 0.016 0.018
In [13]: plt.hist(y)
plt.savefig('plot3.png', dpi=300, bbox_inches='tight')
Type here to search
10:53 PM
1/31/2022

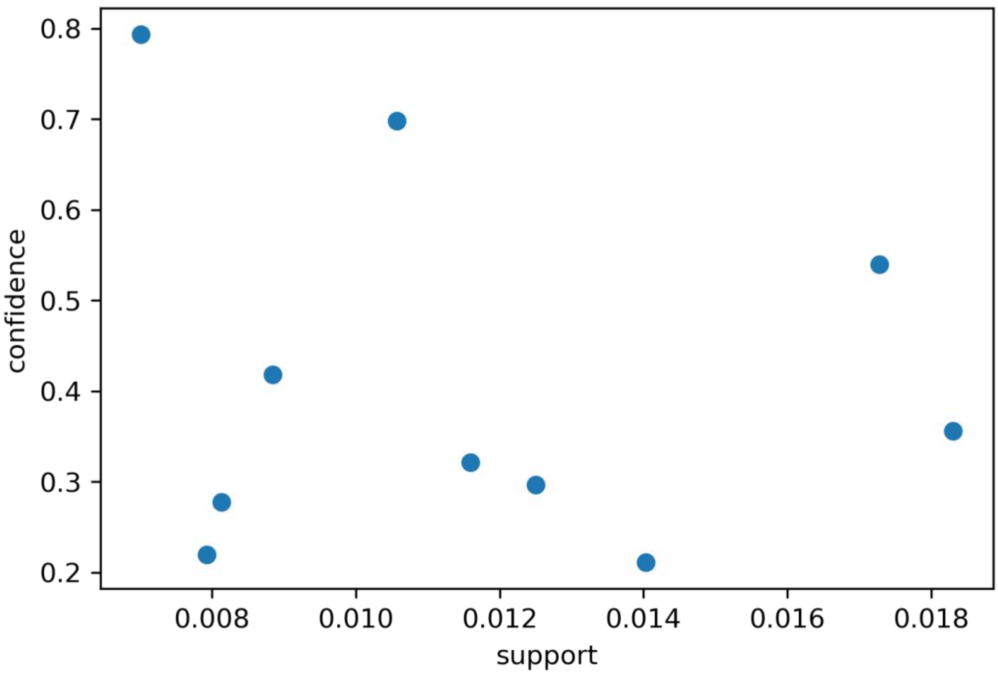

jupyter Implementation of Association Rule Mining Last Checkpoint: Last Saturday at 10:32 PM (autosaved)
localhost:8888/notebooks/arules/Implementation of Association Rule Mining.ipynb
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
0.008 0.010 0.012 0.014 0.016 0.018
In [13]: plt.hist(y)
plt.savefig('plot3.png', dpi=300, bbox_inches='tight')
4.0 3.5 3.0 2.5 2.0 1.5 1.0 0.5 0.0
0.2 0.3 0.4 0.5 0.6 0.7
In [14]: rules3
Out[14]:
antecedents consequents support confidence lift
Type here to search
10:53 PM
1/31/2022



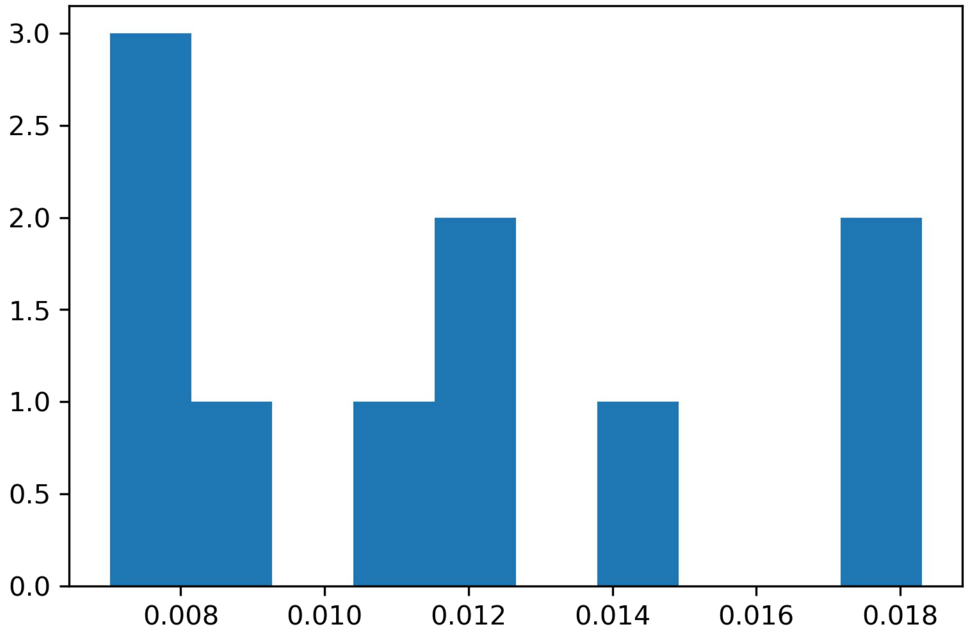
	antecedents	consequents	support	confidence	lift
0	(1_tropical fruit)	(0_citrus fruit)	0.011591	0.321127	6.971924
4	(0_other vegetables)	(1_whole milk)	0.014032	0.300000	4.511468
12	(1_other vegetables)	(2_whole milk)	0.018302	0.327273	6.361121
14	(1_root vegetables)	(2_other vegetables)	0.012506	0.321149	7.610840
16	(1_root vegetables)	(2_whole milk)	0.008338	0.214099	4.161395
18	(1_tropical fruit)	(2_pip fruit)	0.007931	0.219718	14.700201
20	(2_yogurt)	(1_whole milk)	0.009253	0.314879	4.735220
22	(2_other vegetables)	(3_whole milk)	0.017285	0.409639	12.789826
24	(2_root vegetables)	(3_other vegetables)	0.008846	0.418269	16.195582
28	(4_whole milk)	(3_other vegetables)	0.010574	0.697987	27.026370
30	(4_other vegetables)	(5_whole milk)	0.007016	0.575000	65.001437

10. Plotted image of the output : -

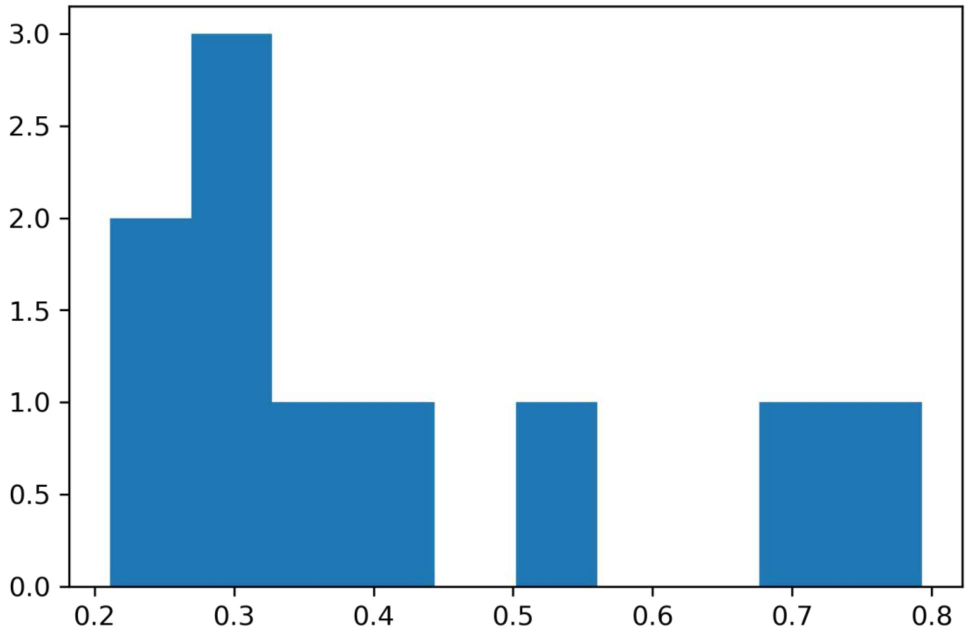
A. Each value in the data set is represented by a dot : -



B. Horizontal axis represents support & vertical axis represents number of rules: -



C. Horizontal axis represents confidence & vertical axis represents number of rules:-



11 . Conclusion : -

The project on **“Implementation of Association Rule Mining”** greatly helped us to develop our knowledge on Python .Association rule mining can be greatly useful for businesses willing to increase their sales whether it's a small retail store or e-commerce website .