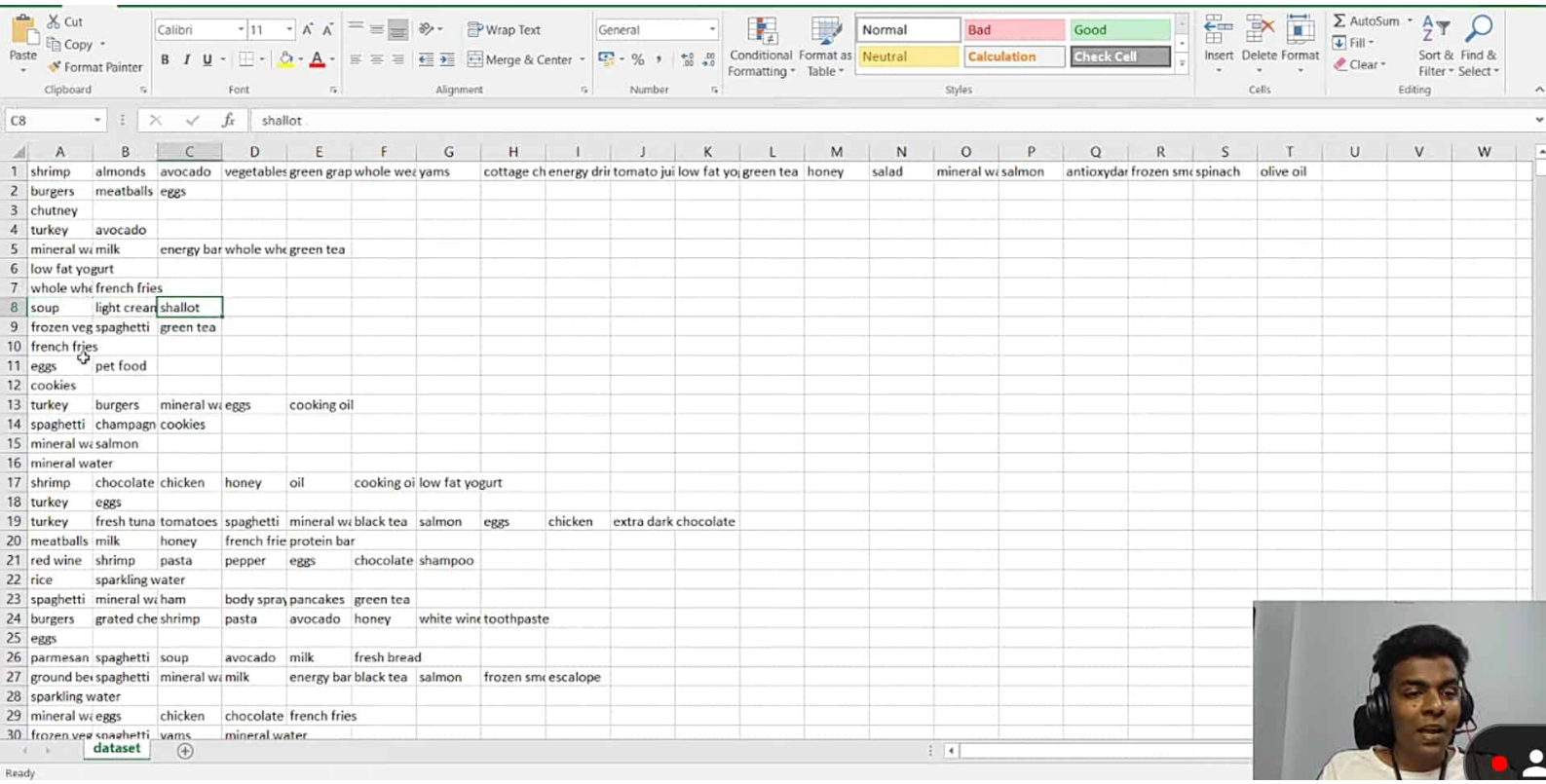
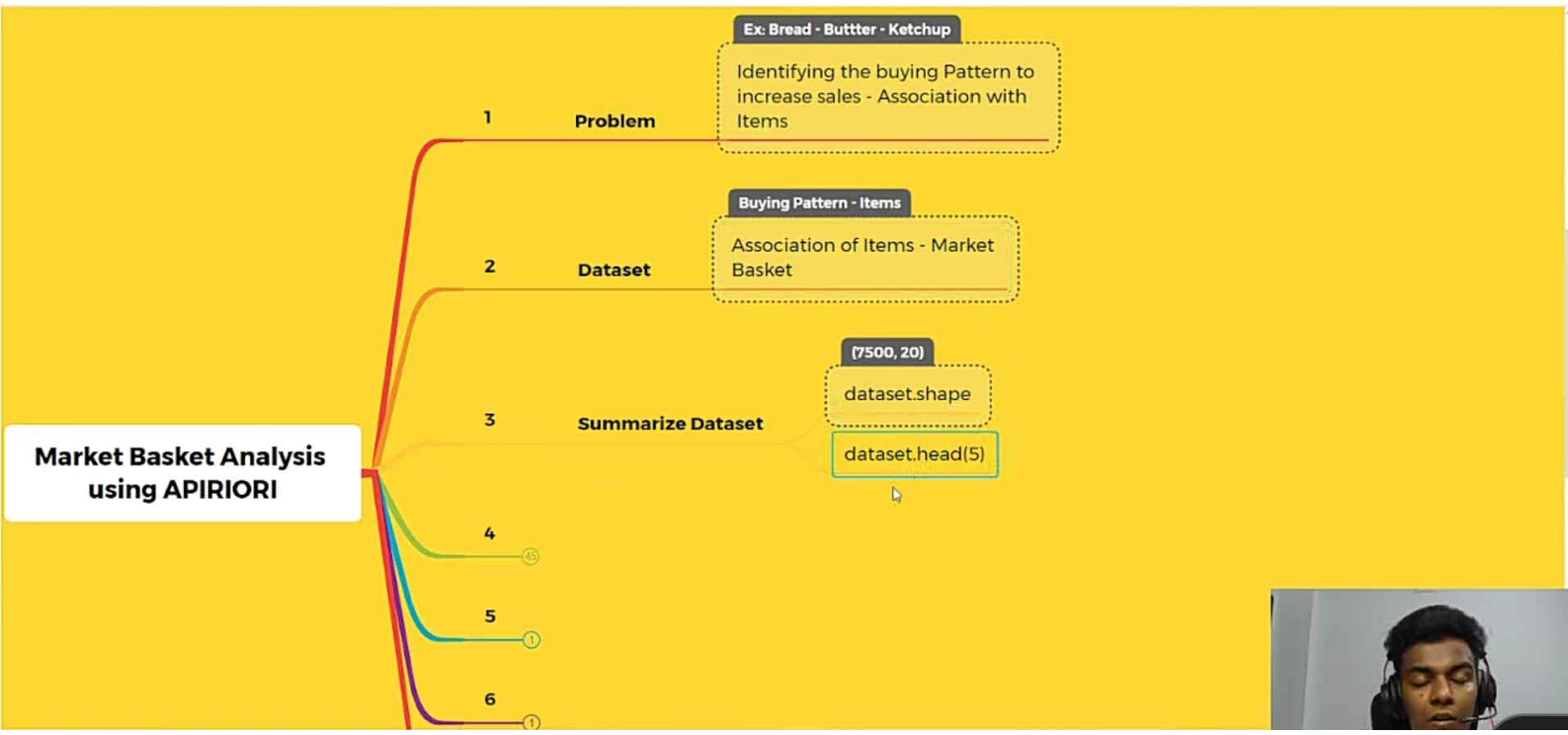


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	shrimp	almonds	avocado	vegetables	green grap	whole wea	yams	cottage ch	energy drir	tomato jui	low fat yo	green tea	honey	salad	mineral wa	salmon	antioxydar	frozen sm	spinach	olive oil			
2	burgers	meatballs	eggs																				
3	chutney																						
4	turkey	avocado																					
5	mineral wa	milk	energy bar	whole whe	green tea																		
6	low fat yogurt																						
7	whole whe	french fries																					
8	soup	light cream	shallot																				
9	frozen veg	spaghetti	green tea																				
10	french fries																						
11	eggs	pet food																					
12	cookies																						
13	turkey	burgers	mineral wa	eggs	cooking oil																		
14	spaghetti	champagn	cookies																				
15	mineral wa	salmon																					
16	mineral water																						
17	shrimp	chocolate	chicken	honey	oil	cooking oi	low fat yogurt																
18	turkey	eggs																					
19	turkey	fresh tuna	tomatoes	spaghetti	mineral wa	black tea	salmon	eggs	chicken	extra dark	chocolate												
20	meatballs	milk	honey	french frie	protein bar																		
21	red wine	shrimp	pasta	pepper	eggs	chocolate	shampoo																
22	rice	sparkling water																					
23	spaghetti	mineral wa	ham	body spray	pancakes	green tea																	
24	burgers	grated che	shrimp	pasta	avocado	honey	white wine	toothpaste															
25	eggs																						
26	parmesan	spaghetti	soup	avocado	milk	fresh bread																	
27	ground bee	spaghetti	mineral wa	milk	energy bar	black tea	salmon	frozen sm	escalope														
28	sparkling water																						
29	mineral wa	eggs	chicken	chocolate	french fries																		
30	frozen veg	spaghetti	yams	mineral water																			


Market Basket Analysis using APIRIORI
1
Problem
Ex: Bread - Buttter - Ketchup
Identifying the buying Pattern to increase sales - Association with Items
2
Dataset
Buying Pattern - Items
Association of Items - Market Basket
3
Summarize Dataset
(7500, 20)
dataset.shape
dataset.head(5)
4
5
6

3 **Summarize Dataset**

dataset.snape

dataset.head(5)

4 **Algorithm**

APIRIORI

{Bread} -> {Butter} or {Bread, Butter} -> Ketchup

Analogy of Bread, Butter & Ketchup

LHS

Antecedent - IF

RHS

Consequent - THEN

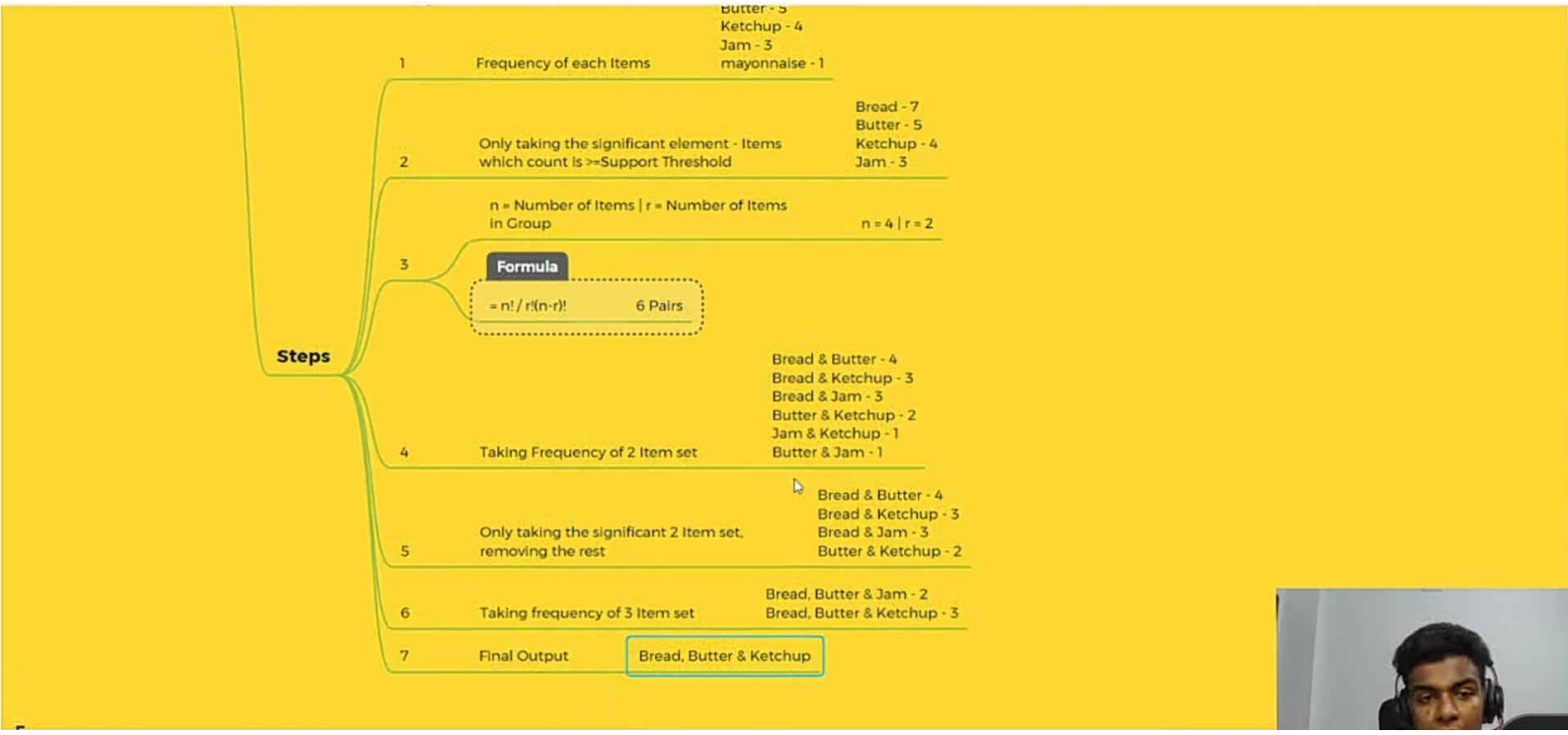
Measures

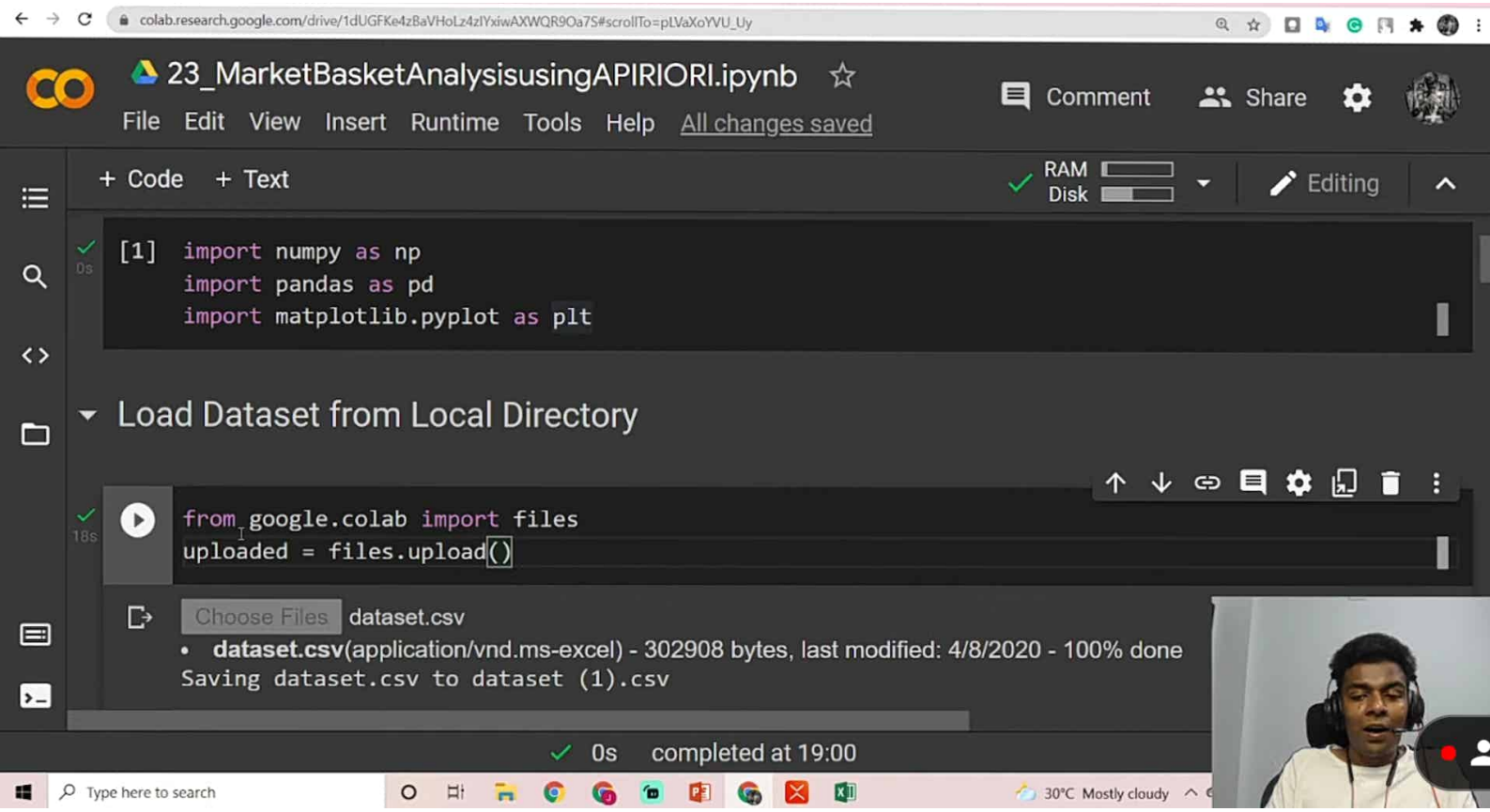
In order to select the insisting ones

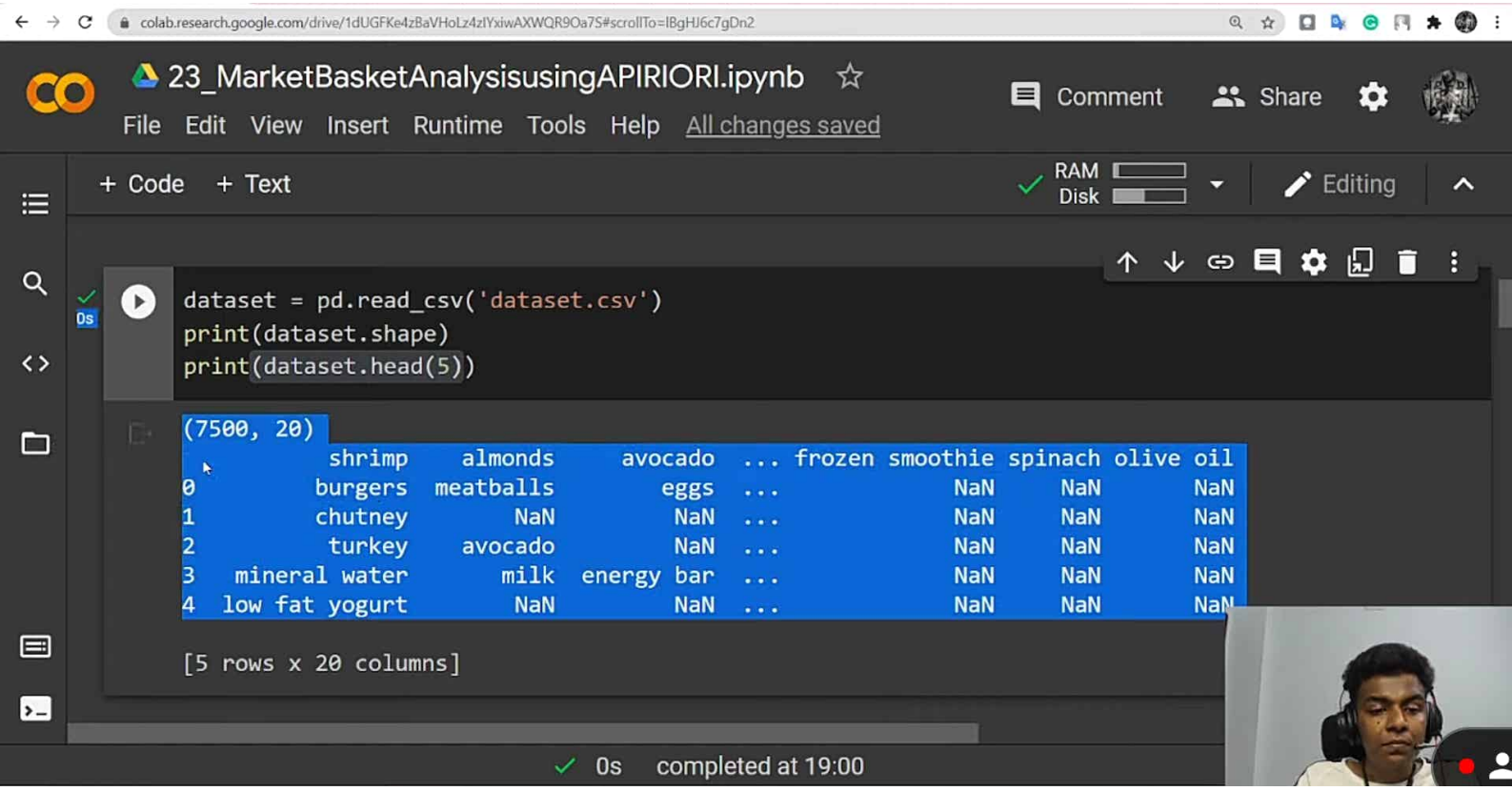
1	Support	Signifies popularity of the Item	supp(bread)=No. of Transaction of Bread / Total no. of Transaction
2	Confidence	Signifies the Likelihood of butter purchased when bread is purchased	Conf(bread->butter) = supp(bread U butter) / supp(bread)
3	Lift	If its Bayesian, there may be the equal probability of bread and butter which increases confidence. To overcome this problem we using Lift	Lift(bread->butter) = supp(bread U butter) / supp(bread) *supp(butter)
4	Conviction	Conv(bread -> butter) = 1 - supp(bread) / 1-conf(bread -> butter)	

Bread - 7

Butter - 5


Steps
1
Frequency of each Items
Butter - 5
Ketchup - 4
Jam - 3
mayonnaise - 1
2
Only taking the significant element - Items which count is >=Support Threshold
Bread - 7
Butter - 5
Ketchup - 4
Jam - 3
3
n = Number of Items | r = Number of Items in Group
n = 4 | r = 2
Formula
= n! / r!(n-r)!
6 Pairs
4
Taking Frequency of 2 Item set
Bread & Butter - 4
Bread & Ketchup - 3
Bread & Jam - 3
Butter & Ketchup - 2
Jam & Ketchup - 1
Butter & Jam - 1
5
Only taking the significant 2 Item set, removing the rest
Bread & Butter - 4
Bread & Ketchup - 3
Bread & Jam - 3
Butter & Ketchup - 2
6
Taking frequency of 3 Item set
Bread, Butter & Jam - 2
Bread, Butter & Ketchup - 3
7
Final Output
Bread, Butter & Ketchup


colab.research.google.com/drive/1dUGFKe4zBaVHoLz4zIYxiwAXWQR9Oa7S#scrollTo=pLVaXoYVU_Uy
23_MarketBasketAnalysisusingAPIRIORI.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
0s
Load Dataset from Local Directory
from google.colab import files
uploaded = files.upload()
18s
Choose Files dataset.csv
dataset.csv(application/vnd.ms-excel) - 302908 bytes, last modified: 4/8/2020 - 100% done
Saving dataset.csv to dataset (1).csv
0s completed at 19:00
Type here to search
30°C Mostly cloudy


colab.research.google.com/drive/1dUGFKe4zBaVHoLz4zIYxiwAXWQR9Oa7S#scrollTo=lBgHJ6c7gDn2
23_MarketBasketAnalysisusingAPIRIORI.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
dataset = pd.read_csv('dataset.csv')
print(dataset.shape)
print(dataset.head(5))
(7500, 20)
shrimp almonds avocado ... frozen smoothie spinach olive oil
0 burgers meatballs eggs ... NaN NaN NaN
1 chutney NaN NaN ... NaN NaN NaN
2 turkey avocado NaN ... NaN NaN NaN
3 mineral water milk energy bar ... NaN NaN NaN
4 low fat yogurt NaN NaN ... NaN NaN NaN
[5 rows x 20 columns]
0s completed at 19:00

colab.research.google.com/drive/1dUGFKe4zBaVHoLz4zIYxiwAXWQR9Oa7S#scrollTo=pm6q3gWXgG_1

23_MarketBasketAnalysisusingAPIRIORI.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

```
[5 rows x 20 columns]
```

Data Pre-Processing

```
transactions = []
for i in range(0, 7500):
  transactions.append([str(dataset.values[i,j]) for j in range(0, 20)])
transactions
```

```
 'frozen vegetables',
 'mineral water',
 'muffins',
 'cereals',
```

0s completed at 19:00

colab.research.google.com/drive/1dUGFKe4zBaVHoLz4zIYxiwAXWQR9Oa7S#scrollTo=wOuthXLIT0GI

23_MarketBasketAnalysisusingAPIRIORI.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

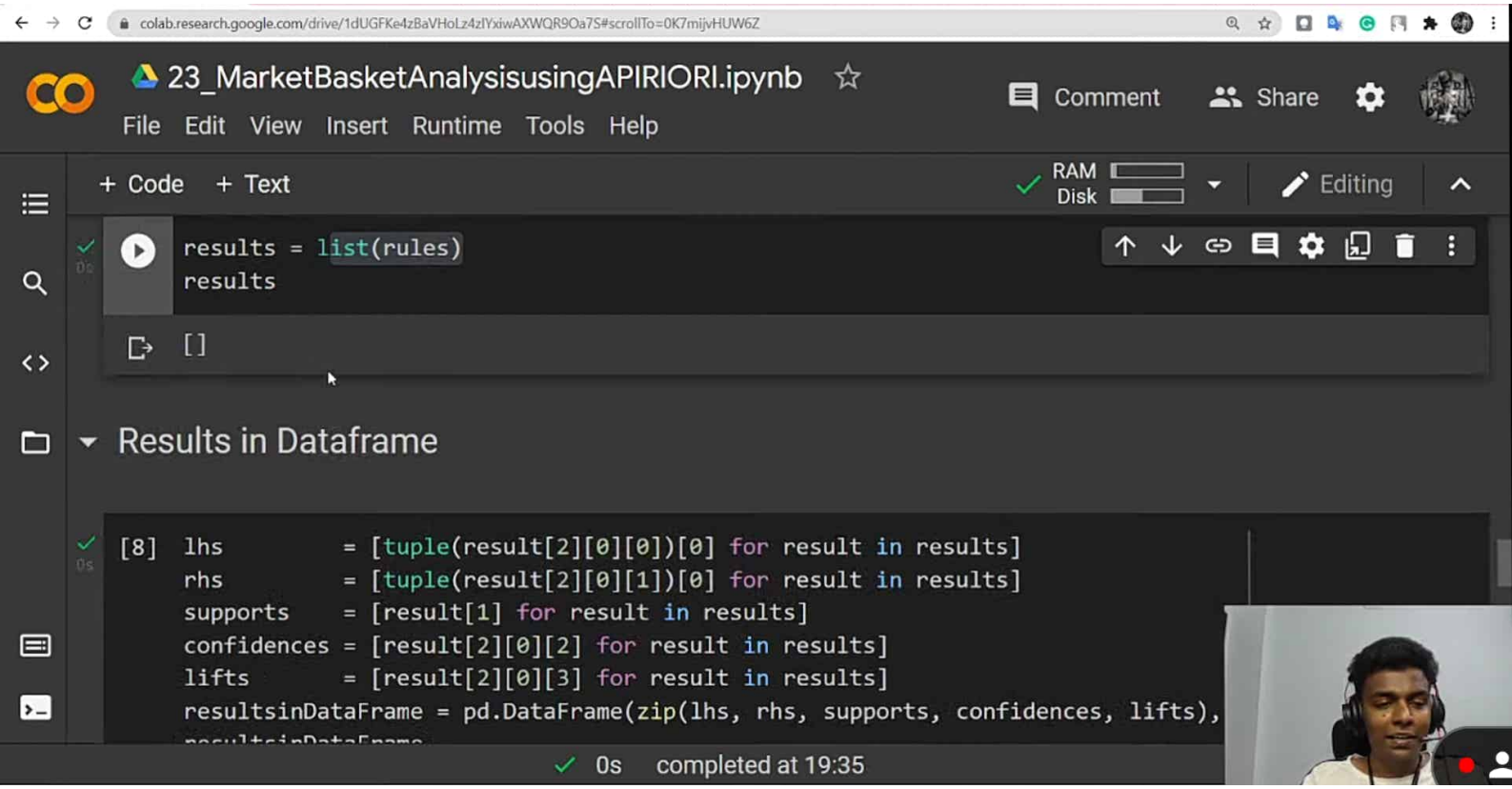
```
        'nan',
        'nan',
        'nan'
```

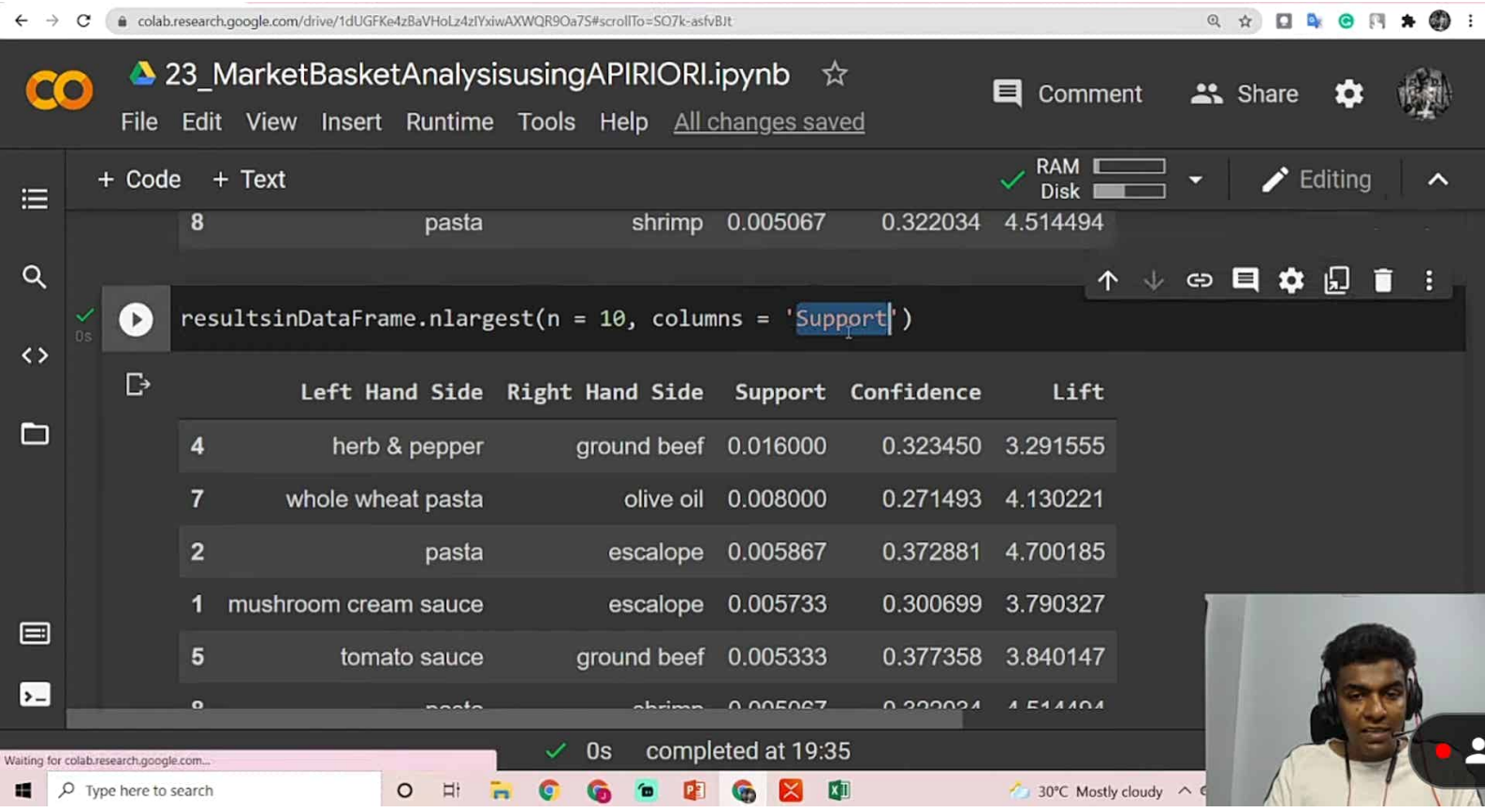
Training APRIORI

```
pip install apyori
from apyori import apriori
rules = apriori(transactions = transactions, min_support = 0.003, min_confidence = 0.2, min_lift =
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=5974 sh
```

0s completed at 19:00


colab.research.google.com/drive/1dUGFKe4zBaVHoLz4zIYxiwAXWQR9Oa7S#scrollTo=0K7mijvHUW6Z
23_MarketBasketAnalysisusingAPIRIORI.ipynb
Comment
Share
File Edit View Insert Runtime Tools Help
+ Code + Text
RAM
Disk
Editing
results = list(rules)
results
[]
Results in Dataframe
[8] lhs = [tuple(result[2][0][0])[0] for result in results]
rhs = [tuple(result[2][0][1])[0] for result in results]
supports = [result[1] for result in results]
confidences = [result[2][0][2] for result in results]
lifts = [result[2][0][3] for result in results]
resultsinDataFrame = pd.DataFrame(zip(lhs, rhs, supports, confidences, lifts),
0s completed at 19:35


colab.research.google.com/drive/1dUGFKe4zBaVHoLz4zIYxiwAXWQR9Oa7S#scrollTo=SO7k-asfvBJt
23_MarketBasketAnalysisusingAPIRIORI.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
8 pasta shrimp 0.005067 0.322034 4.514494
resultsinDataFrame.nlargest(n = 10, columns = 'Support')
0s
Left Hand Side Right Hand Side Support Confidence Lift
4 herb & pepper ground beef 0.016000 0.323450 3.291555
7 whole wheat pasta olive oil 0.008000 0.271493 4.130221
2 pasta escalope 0.005867 0.372881 4.700185
1 mushroom cream sauce escalope 0.005733 0.300699 3.790327
5 tomato sauce ground beef 0.005333 0.377358 3.840147
0s completed at 19:35
Waiting for colab.research.google.com...
Type here to search
30°C Mostly cloudy