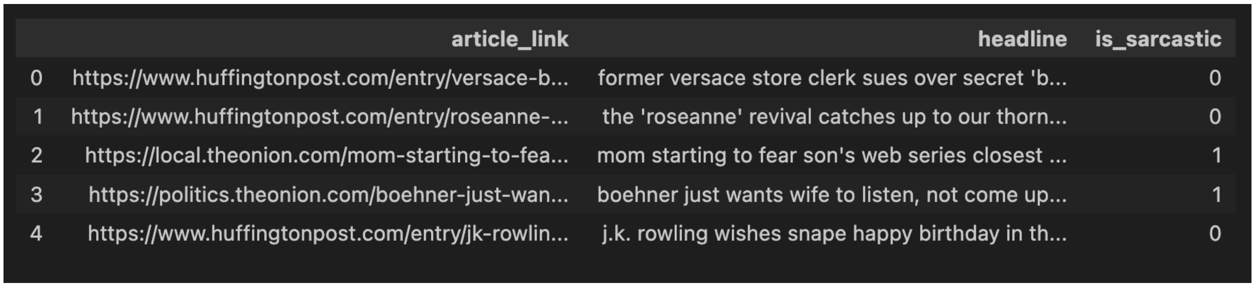
UNSUPERVISED ML
-
CLUSTERING
AND
ASSOCIATION RULES

Problem Statement : Clustering

Using k-means clustering to classify text as sarcastic or not.

About the Dataset

The data has columns such as articles, sentences to predict and the target variable. The data is provided in JSON format.

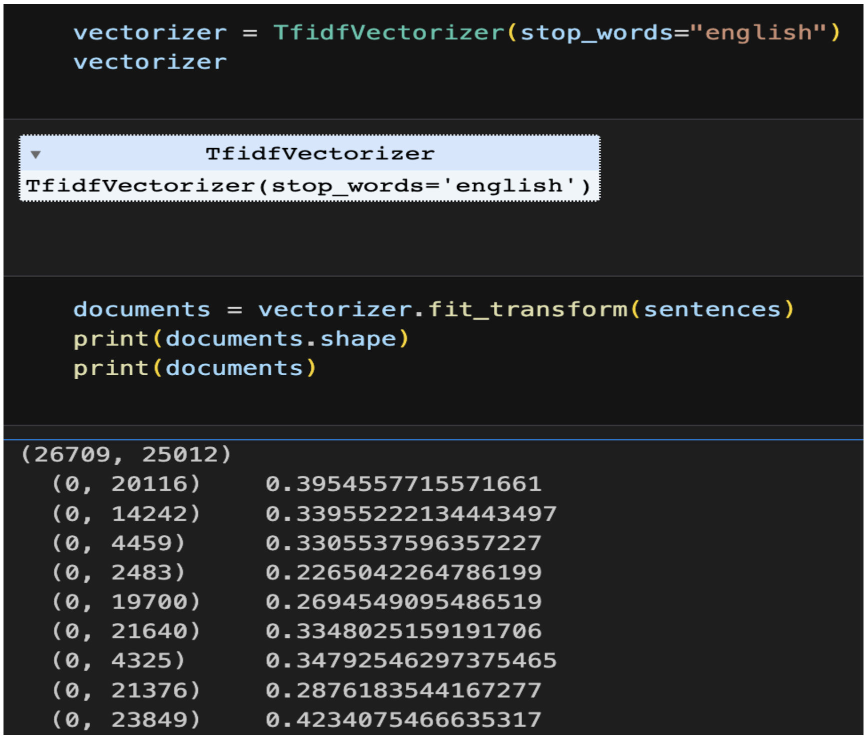


	article_link	headline	is_sarcastic
0	https://www.huffingtonpost.com/entry/versace-b...	former versace store clerk sues over secret 'b...	0
1	https://www.huffingtonpost.com/entry/roseanne-...	the 'roseanne' revival catches up to our thorn...	0
2	https://local.theonion.com/mom-starting-to-fea...	mom starting to fear son's web series closest ...	1
3	https://politics.theonion.com/boehner-just-wan...	boehner just wants wife to listen, not come up...	1
4	https://www.huffingtonpost.com/entry/jk-rowlin...	j.k. rowling wishes snape happy birthday in th...	0

View of the dataset

Steps involved

1) The text data is represented using TfidfVectorizer.



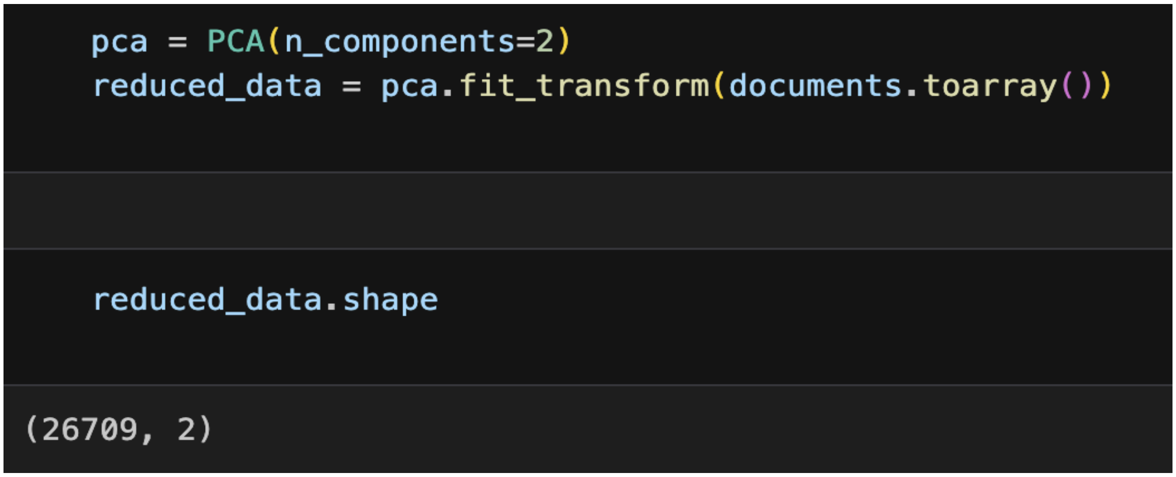
```
vectorizer = TfidfVectorizer(stop_words="english")
vectorizer
```

```
▾ TfidfVectorizer
TfidfVectorizer(stop_words='english')
```

```
documents = vectorizer.fit_transform(sentences)
print(documents.shape)
print(documents)
```

```
(26709, 25012)
  (0, 20116)    0.3954557715571661
  (0, 14242)    0.33955222134443497
  (0, 4459)     0.3305537596357227
  (0, 2483)     0.2265042264786199
  (0, 19700)    0.2694549095486519
  (0, 21640)    0.3348025159191706
  (0, 4325)     0.34792546297375465
  (0, 21376)    0.2876183544167277
  (0, 23849)    0.4234075466635317
```

2) **PCA** is used to decrease dimensionality.



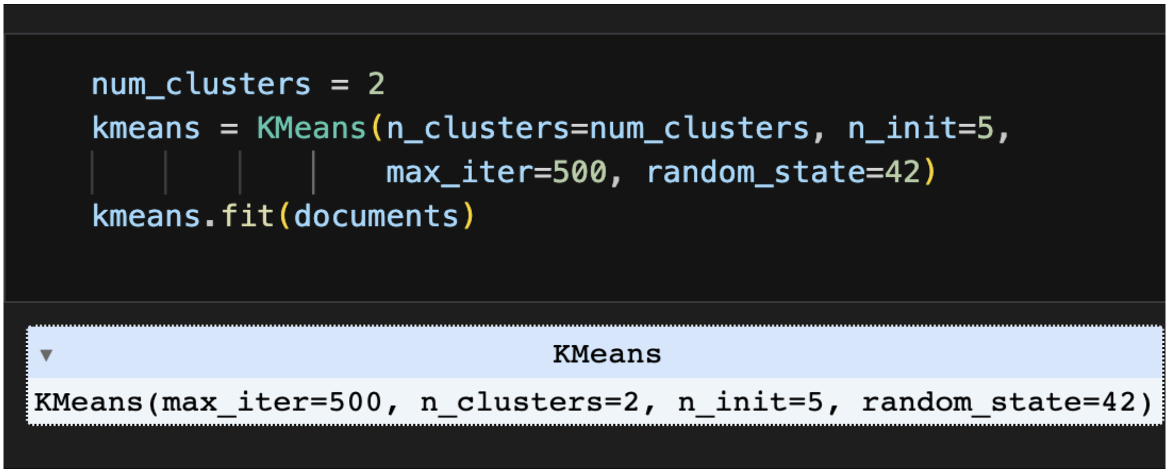
```python
pca = PCA(n_components=2)
reduced_data = pca.fit_transform(documents.toarray())
```

```python
reduced_data.shape
```

```
(26709, 2)
```

3) Clustering

- As the target variable is binary the number of clusters is 2.

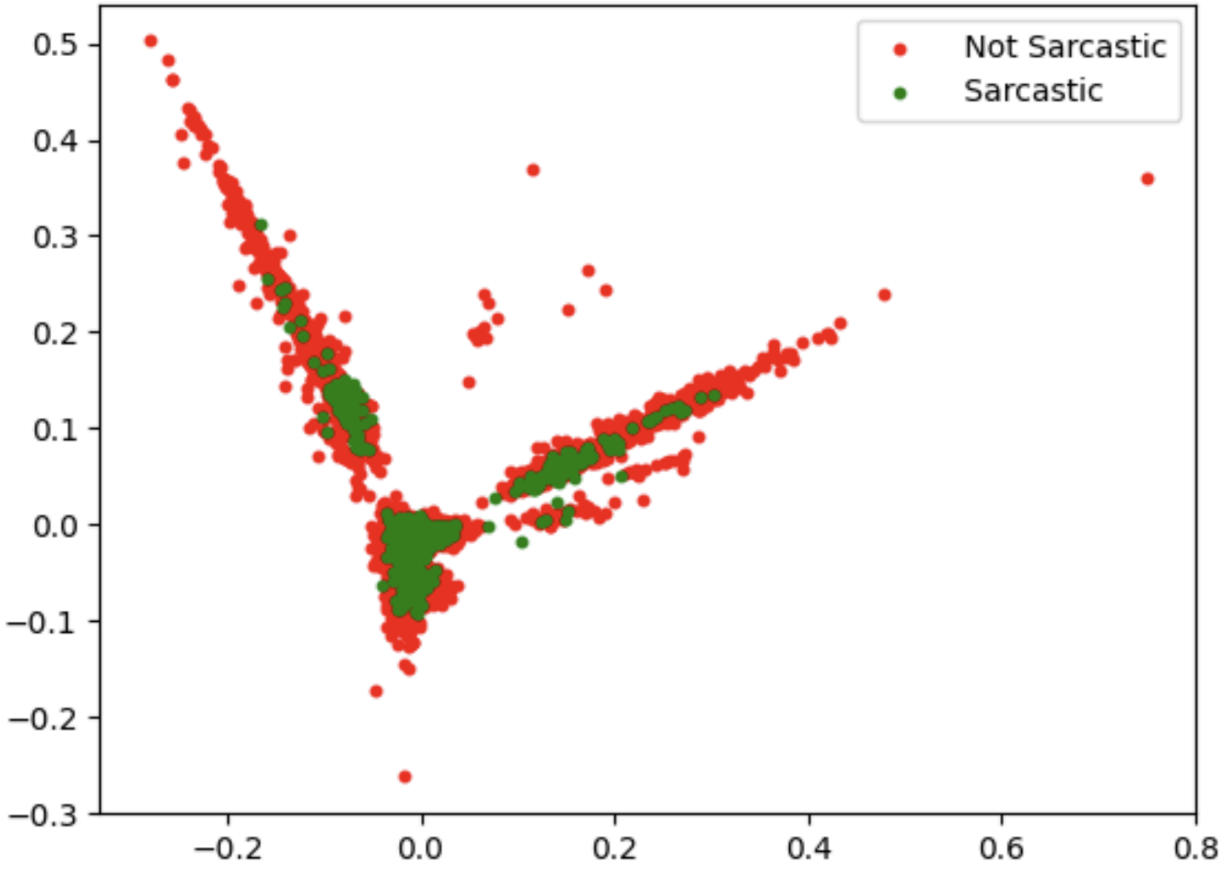


```python
num_clusters = 2
kmeans = KMeans(n_clusters=num_clusters, n_init=5,
                max_iter=500, random_state=42)
kmeans.fit(documents)
```

```
▼                          KMeans
KMeans(max_iter=500, n_clusters=2, n_init=5, random_state=42)
```

4) Visualizing

- Plotting the final clustered target variable.

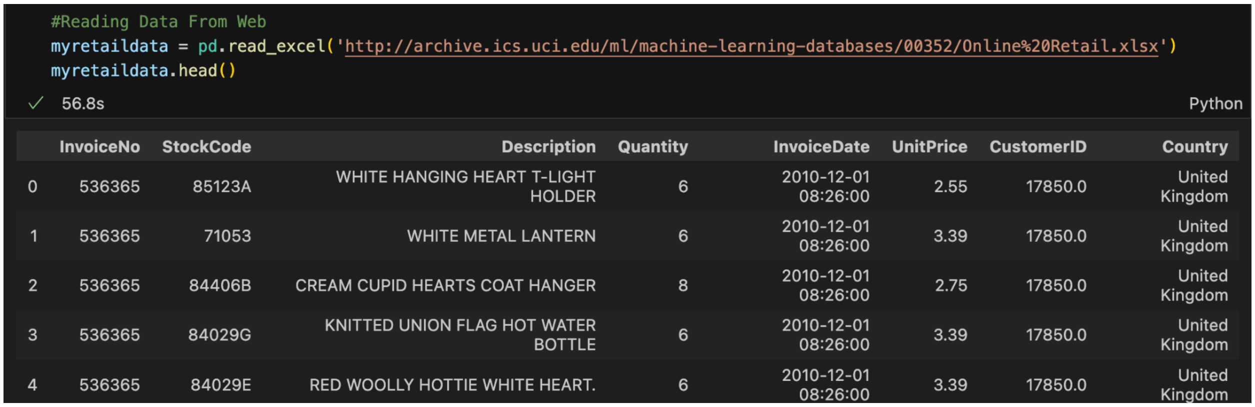


Problem Statement : Association rules - Apriori

To perform market basket analysis with association rules and apriori algorithm to find frequent patterns of the given dataset.

About the Dataset

The data consists of features - Invoice number, stock code, description, Quantity, Invoice date, Unit price, Customer Id and Country.



```
#Reading Data From Web
myretaildata = pd.read_excel('http://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx')
myretaildata.head()
```

✓ 56.8s

Python

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

View of the dataset

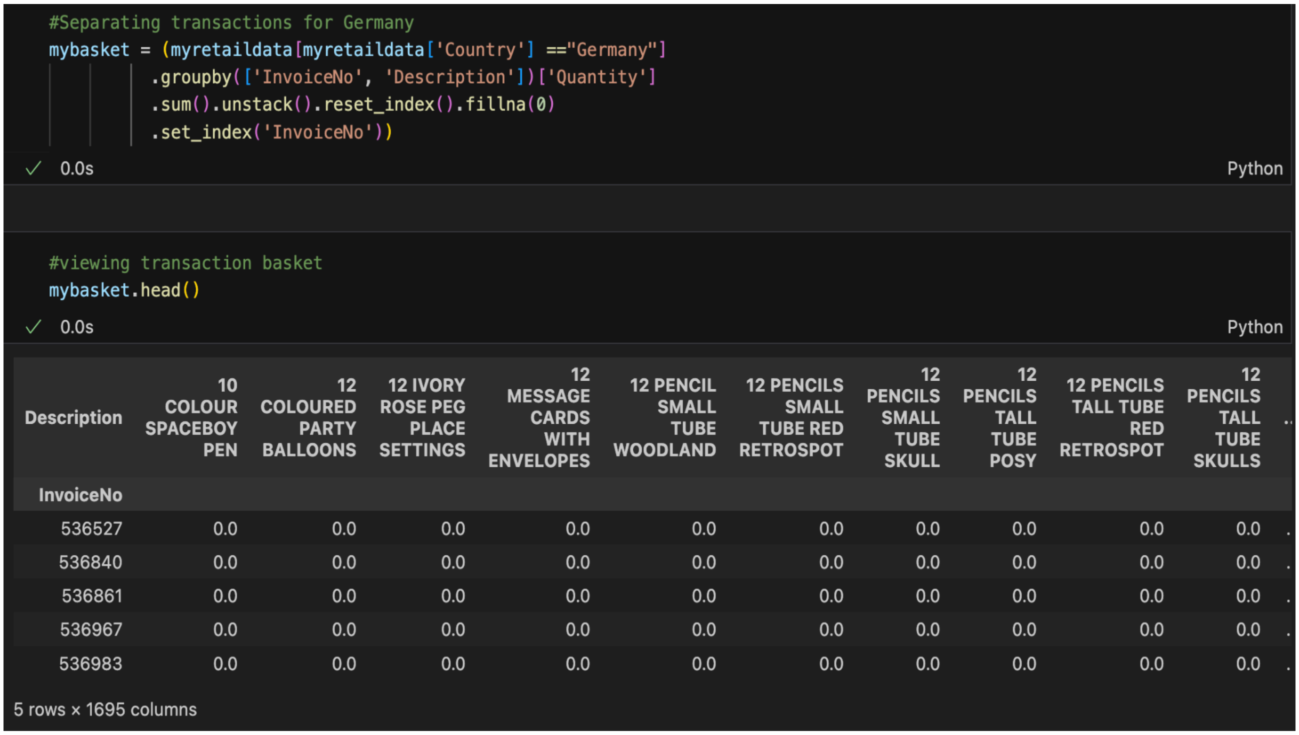
Steps involved

1) Data Preparation

- Drop the missing values for invoice number, remove the blankspaces front and back of invoice number and remove credit transactions as we are not focusing on them.

2) Converting to transactions

- We are now focusing on the country - Germany and to group by invoice number and description with sum of quantities as values.



```
#Separating transactions for Germany
mybasket = (myretaildata[myretaildata['Country'] =="Germany"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

✓ 0.0s Python

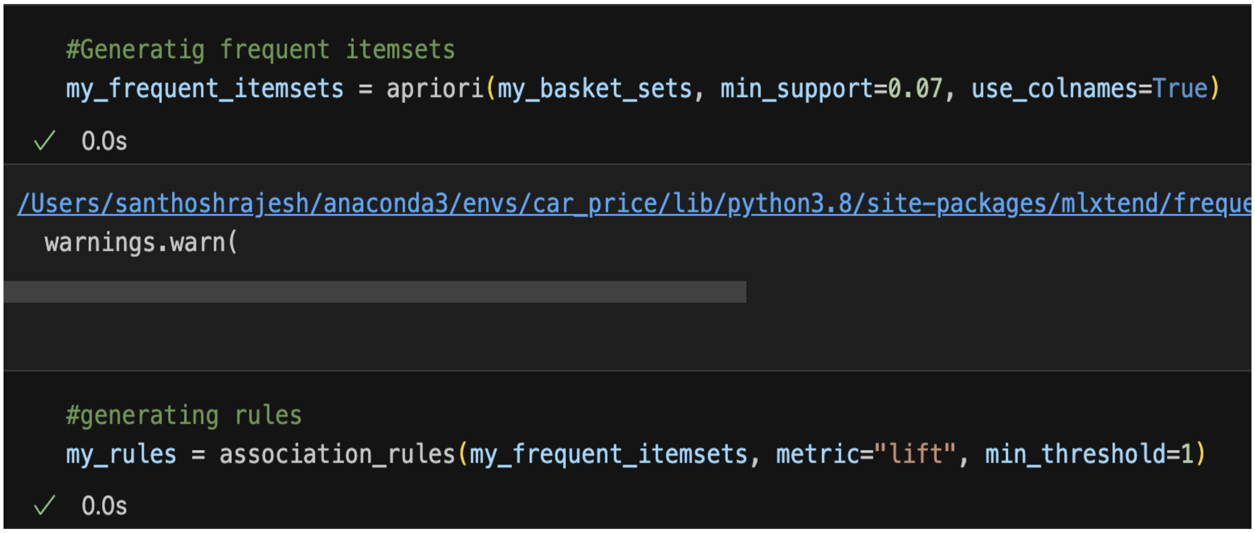
```
#viewing transaction basket
mybasket.head()
```

✓ 0.0s Python

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 IVORY ROSE PEG PLACE SETTINGS	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	12 PENCILS TALL TUBE SKULLS	...
InvoiceNo											
536527	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536840	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536861	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536967	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
536983	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

5 rows × 1695 columns

3) Model training



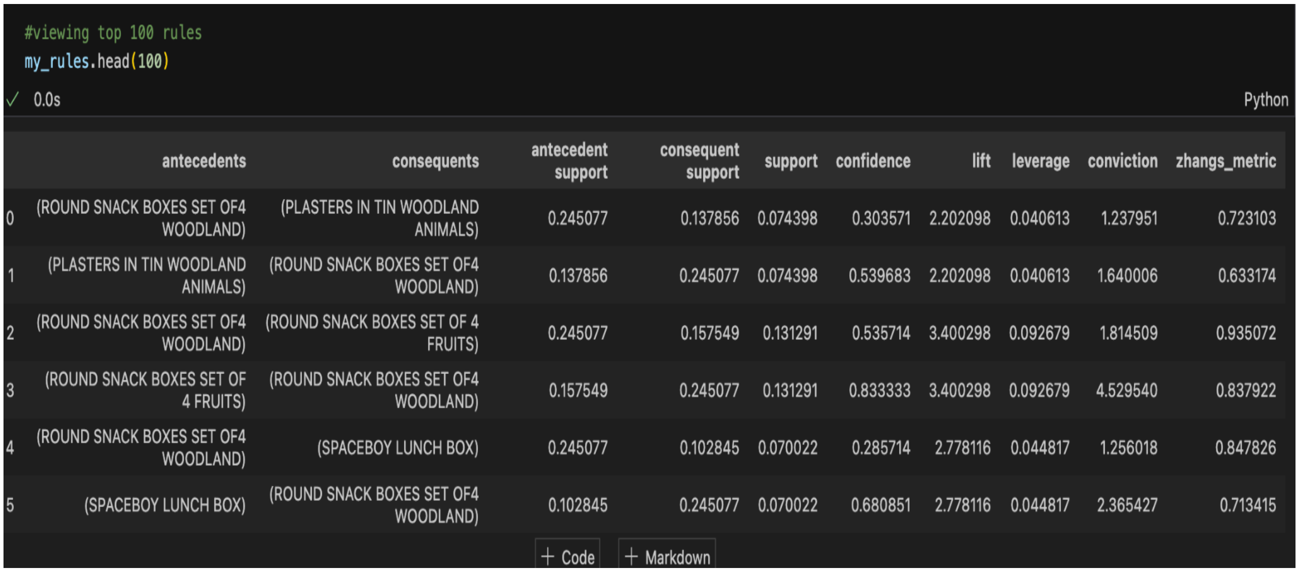
```
#Generatig frequent itemsets
my_frequent_itemsets = apriori(my_basket_sets, min_support=0.07, use_colnames=True)
```

✓ 0.0s

```
/Users/santhoshrajesh/anaconda3/envs/car_price/lib/python3.8/site-packages/mlxtend/freque
  warnings.warn(
```

```
#generating rules
my_rules = association_rules(my_frequent_itemsets, metric="lift", min_threshold=1)
```

✓ 0.0s



```
#viewing top 100 rules
my_rules.head(100)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(ROUND SNACK BOXES SET OF4 WOODLAND)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.245077	0.137856	0.074398	0.303571	2.202098	0.040613	1.237951	0.723103
1	(PLASTERS IN TIN WOODLAND ANIMALS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.137856	0.245077	0.074398	0.539683	2.202098	0.040613	1.640006	0.633174
2	(ROUND SNACK BOXES SET OF4 WOODLAND)	(ROUND SNACK BOXES SET OF 4 FRUITS)	0.245077	0.157549	0.131291	0.535714	3.400298	0.092679	1.814509	0.935072
3	(ROUND SNACK BOXES SET OF 4 FRUITS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.157549	0.245077	0.131291	0.833333	3.400298	0.092679	4.529540	0.837922
4	(ROUND SNACK BOXES SET OF4 WOODLAND)	(SPACEBOY LUNCH BOX)	0.245077	0.102845	0.070022	0.285714	2.778116	0.044817	1.256018	0.847826
5	(SPACEBOY LUNCH BOX)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.102845	0.245077	0.070022	0.680851	2.778116	0.044817	2.365427	0.713415

From the rules generated we can find the most frequent items bought together and recommend them to the customers.