Data Science with Python Programming

- Presentation By Uplatz
- Contact us: https://training.uplatz.com
- Email: info@uplatz.com
- Phone: +44 7836 212635

Uplatz

Association Rule Learning

Learning outcomes:

- Association Rule Learning
- Apriori algorithm
- Working of Apriori algorithm
- Implementation of Apriori algorithm

Association Rule Learning

Association Rule Learning has the most popular applications of Machine Learning in business. It has been widely used to understand and test various business and marketing strategies to increase sales and productivity by various organizations including supermarket chains and online marketplaces. Association Rule Learning is rule-based learning for identifying the association between different variables in a database.

Association Rule Learning

One of the best and most popular examples of Association Rule Learning is the Market Basket Analysis. Market basket analysis is one of the key applications of machine learning in retail. By analysing the past buying behaviour of customers, one can find out which are the products that are bought together by the customers. For example, bread and butter are sold together, baby diapers and baby massage oil are sold together, etc. The problem analyses the association between various items that has the highest probability of being bought together by a customer.

Association Rule Learning

For example, the association rule, {onions, chicken masala} => {chicken} says that a person who has got both onions and chicken masala in his or her basket has a high probability of buying chicken also.
This process of analysing the association is called the **Association Rule Learning** and analysing the products bought together by the customers is called the Market Basket Analysis.

Association Rule Learning

In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection, continuous production, and bioinformatics.

In this session, we will discuss the association rule learning method with a practical implementation of market basket analysis in python. We will use the **Apriori** algorithm as an association rule method for market basket analysis.

Apriori algorithm

Apriori is considered an algorithm for frequent item-set mining and association rule learning over transactional databases. It proceeds just by identifying the frequent individual items in the database and then extending them to larger and larger item sets. The observation is, for as long as those item-sets appear sufficiently often in the database. The frequent item-sets that were determined by **Apriori** can be later used to determine about the association rules which highlights all the general trends that are being used in the database: this has got applications that fall in the domains such as the market basket analysis.

Apriori algorithm

Apriori Algorithm is a Machine Learning algorithm which is used to gain insight into the structured relationships between different items involved. The most prominent practical application of the algorithm is to recommend products based on the products already present in the user’s cart. **Walmart** especially has made great use of the algorithm in suggesting products to it’s users. **Apriori algorithm** is the perfect algorithm to start with association analysis as it is not just easy to understand and interpret but also to implement.

Apriori algorithm

The Apriori algorithm uses frequent item sets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected.
The algorithm was first proposed in 1994 by **Rakesh Agrawal** and **Ramakrishnan Srikant**. It is mainly used for *market basket analysis* and helps to find those products that can be bought together.

Working of Apriori algorithm

To construct association rules between elements or items, the algorithm considers 3 important factors which are, **support, confidence** and **lift.**
We will explain this concept with the help of an example. *Suppose we have a record of 1000 customers transactions and we want to find out* ***support***, ***confidence*** *and* ***lift*** *for milk and diapers. out of 1000 transactions, 120 contains a milk and 150 contains a diaper, out of this 150 transaction where a diaper is purchased 30 contains transaction contains milk as well. we will use this data to calculate* ***support, confidence*** *and* ***lift***.

Working of Apriori algorithm

Support:

Support refers to the popularity of item and can be calculated by finding the number of transactions containing a particular item divided by the total number of transactions.

Support(diaper) = (Transactions containing (diaper))/(Total Transactions)

Support(diaper) = 150 / 1000 = 15 %

Uplatz

Working of Apriori algorithm

Confidence: Confidence refers to the likelihood that an item B is also bought if item A is bought. It can be calculated by finding the number of transactions where A and B are bought together, divided by the total number of transactions where A is bought. Mathematically, it can be represented as:

Confidence(A → B) = (Transactions containing both (A and B))/(Transactions containing A)

The confidence of likelihood of purchasing a diaper if a customer purchase milk.

Confidence(milk → diaper) = (Transactions containing both (milk and diaper))/(Transactions containing milk)

Confidence(milk → daiper) =30 / 120 = 25 %

Working of Apriori algorithm

Lift: Lift refers to the increase in the ratio of the sale of B when A is sold.
Lift(A –> B) can be calculated by dividing Confidence(A -> B) divided by Support(B).
Mathematically it can be represented as:
Lift(A→B) = (Confidence (A→B))/(Support (B))
Lift(milk → diaper) = (Confidence (milk → diaper))/(Support (diaper))
Lift(milk → diaper) = 25 / 15 = 1.66
So by Lift theory, there is 1.66 times more chance of buying milk and diaper together than just buying diaper alone.

Working of Apriori algorithm

Steps for Apriori Algorithm:

Below are the steps for the Apriori algorithm:

Step-1: Determine the support of itemsets in the transactional database, and select the minimum support and confidence.

Step-2: Take all supports in the transaction with higher support value than the minimum or selected support value.

Step-3: Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

Step-4: Sort the rules as the decreasing order of lift.

Implementation of Apriori algorithm

Now that we know all about how Apriori algorithm works. Now we will implement this algorithm using a dataset. This dataset contains 1000 transactions over the course of a week at a French retail store. To implement this, we have a problem of a retailer, who wants to find the association between his shop's product, so that he can provide an offer of "Buy this and Get that" to his customers. The retailer has a dataset information that contains a list of transactions made by his customer.

Implementation of Apriori algorithm

In the dataset, each row shows the products purchased by customers or transactions made by the customer. To solve this problem, we will perform the below steps:
Before we begin our coding we need to install the apyori package. To install the package use the following code in Jupyter notebook.

```
!pip install apyori
```

Implementation of Apriori algorithm

Step-1: Importing the libraries:

```
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
```

Step-2: **Importing the Dataset**

Now lets import dataset and see how our dataset looks like, how many transactions are there and what is the shape of the dataset.

```
data = pd.read_csv('C:\Python38\Store_data.csv')
data.head()
```

Uplatz

Implementation of Apriori algorithm

Step-2:

All the rows of the dataset are showing different transactions made by the customers. The first row is the transaction done by the first customer, which means there is no particular name for each column and have their own individual value or product details.

Implementation of Apriori algorithm

Step-3: Data Preprocessing

The Apriori library we are going to use requires our dataset to be in the form of a list of lists, where the whole dataset is a big list and each transaction in the dataset is an inner list within the outer big list. Currently, we have data in the form of a pandas dataframe. To convert our pandas dataframe into a list of lists, execute the following code.

```
transactions=[]
for i in range(0, 1000):
    transactions.append([str(dataset.values[i,j])  for j
 in range(0,20)])
```

Implementation of Apriori algorithm

Step-4: Training the Apriori Model on the dataset
To train the model, we will use the **apriori function** that will be imported from the **apyroi** package. This function will return the **rules** to train the model on the dataset.
Consider the below code:

```
from apyori import apriori
rules= apriori(transactions= transactions,
min_support=0.003, min_confidence = 0.2,
min_lift=3, min_length=2, max_length=2)
```

Implementation of Apriori algorithm

Step-4: Training the Apriori Model on the dataset
In the above code, the first line is to import the apriori function. In the second line, the apriori function returns the output as the rules. It takes the following parameters:
transactions: A list of transactions.
min_support= To set the minimum support float value. Here we have used 0.003 that is calculated by taking 3 transactions per customer each week to the total number of transactions.

Implementation of Apriori algorithm

Step-4: Training the Apriori Model on the dataset

min_confidence: To set the minimum confidence value. Here we have taken 0.2. It can be changed as per the business problem.
min_lift= To set the minimum lift value.
min_length= It takes the minimum number of products for the association.
max_length = It takes the maximum number of products for the association.

Implementation of Apriori algorithm

Step-5: Displaying the result of the rules occurred from the apriori function

```
associations= list(rules)
associations
```

By executing the above lines of code, we will get the 95 rules.

Let's see the number of rules:

```
print(len(associations))
```

Implementation of Apriori algorithm

Understanding the rules:
The apriori algorithm automatically sorts the associations’ rules based on relevance, thus the topmost rule has the highest relevance compared to the other rules returned by the algorithm.
Let’s have a look at the first and most relevant association rule from the given dataset.

```
associations [0]
```

Implementation of Apriori algorithm

Understanding the rules:

Rule one is the most relevant rule that the algorithm identified from the given dataset.

The above output specifies the association between two items ‘burgers‘ and 'almonds’. The first rules, states that the almonds and burgers are bought frequently by most of the customers.

The **support vector** of 0.01 is calculated by dividing the number of transactions containing almonds divided by the total number of transactions.

Implementation of Apriori algorithm

Understanding the rules:

The confidence of 0.3571, tells us that of the total transactions, 35.71 % of transactions also contains burgers. Hence, if a customer buys almonds, it is 35.71% chances that he also buys burgers.

Finally, the lift of 4.46 tells us that there are 4.46 times chances that burger will be bought with almonds.

We can check all these things in other rules also.

Thank you

Uplatz