**Hepsiburada Case Study**

Market Basket Analysis finds patterns to determine connections between purchases in order that stores can increase their cross-selling potential. The main idea behind this analysis is to achieve valuable insights by identifying which items are frequently purchased together.

The Apriori algorithm is a data mining technique for identifying the frequent itemsets and relevant association rules in the database. Support, confidence and lift are the three main components of the Apriori Algorithm.

Measure 1: Support.

This says how popular an itemset is, it is number of times appear in total number of transaction. in other word we say frequency of item.

Measure 2: Confidence.

This says how likely item Y is purchased when item X is purchased, expressed as {X -> Y}. This is measured by the proportion of transactions with item X, in which item Y also appears.

Measure 3: Lift.

İt is ratio of expected confidence to observed confidence. it is described as confidence of Y when item X was already known(x/y) to the confidence of Y when X item is unknown.

***support = occurrence of item / total no of transaction.***

***confidence = support ( X Union Y) / support(X).***

***lift = support (X Union Y)/ support(X) \* support(Y) .***

This algorithm can be used to determine what products to discount. Also, it can increase sales and customer satisfaction. It is important to realize that there are many other areas in which it can be applied.

1. **Model Data**

Model data includes events.json and meta.json files. In python code these files are imported and merged to reach the data shown below.

Graphical user interface

Description automatically generated with medium confidence

After this stage some pre-processing activities are made on the data and reached the data below. Data is now in binary format every column defines a product is available in a cart or not.

Table

Description automatically generated

1. **Model Building**

Apriori algorithm is used for model and predictions are obtained. First frequent items in the data are found by the model and after that algorithm found some rules for the products.

Some of the frequent itemsets are shown below. Based on this we can say that “Domates Pembe 500 gr” product is can be bought with “Nane” product. Which means we can offer customers who buy one of the products to buy the other one too or we can change the shelf array of the products to or we can offer some discounts to related products.

Table

Description automatically generated

Based on the frequent items we found earlier we can now create association rules to decide which rules are more important. Confidence denotes the likelihood of certain items are purchased together. For example, we can say that there is a %50 chance for someone who purchased “Name” and “Maydanoz” would also buy “Dereotu” too. Lift is also a similar metric to define relationship between products. If the lift is 1 it means there is no relationship between ancedents and consequents.

Table

Description automatically generated

In the next stage using this algorithm I managed to create a function that calculates top 10 related products as recommendation. For example, if you put “Patates 1 kg” as the product algorithm gives a list containing 10 recommended products. Which you can see the detail in the python notebook.

1. **API Implementation**

After building the model I wrote the code for api implementation (recommender.py) and I connected to my local host to work my model from the api.

Here is the docs page of my api. In POST argument there is a function called Predict which is used for scoring the input and getting an output.

Graphical user interface, text, application

Description automatically generated

When we click POST we can see the input area which we will enter our input value.

Graphical user interface, application

Description automatically generated

When we press “try it out button” we can see the input screen that we can enter the input.

After writing the input we press execute and wait for the result.

Graphical user interface, text, application

Description automatically generated

After execution we can see the results in response body part.

Graphical user interface, application, background pattern

Description automatically generated with medium confidence

1. **Pros and Cones**

Some of the pros are:

* Algorithm is understandable and manageable compared to other machine learning algorithm.
* There are a lot use cases in the market in this area.
* Results of the study can be used for different purposes such as discounts, campaigns, shelf arrangement etc.
* Model is easy to implement and deploy.
* Model doesn’t need a lot of data cleaning.

Some of the cones are:

* It takes more time to create results compared to other machine learning algorithms.
* Since it is a more simple algorithm it may not address very complicated patterns

1. **Future Improvements**

There are some modifications to the model that I could make if I had more time.

In the present model, each product is examined individually, and ten recommendations are made. Instead of examining every product, I would create a model that would examine a user defined number of products and provide 10 recommendations.