Recommender

Alex COMMEAU

Ryan HEADLEY

Geoffroy HUCK

Jonathan KHALIFA

Fabrice SUMSA

EPITECH

T-DAT-901 Big Data

January 30, 2022

Abstract *(for professional papers)*

A system of product recommendations was created based on a dataset provided, which contained the product purchases from a wide range of clients. The system itself takes a client ID, and returns a list of recommended products. Two different algorithms are offered: The first is a collaborative based filtering method, which creates a rank for each product purchased by each client; the second is a content based method using NLP to detect similarity between product titles. The resulting recommendations are provided with a categorical weighted average.

Keywords: Collaborative filtering, item based, Artificial Intelligence

Recommender

Introduction

The goal of the project was to create an application of recommendation, based on the sales dataset provided. This application takes a client id and returns a list of recommended products that have a high probability of being compatible with said client. Evidently, any product already purchased by a customer will not be included in the list of recommendations.

All information that follows, details the methodology applied to arrive at this application. The dataset and all data calculated from therein was stored in a free MongoDB.

Background

The dataset provided for this project contains millions of bought items, divided into three categories. Each row indicates the sale of 1 product by 1 client, denoted with the sale price, month of sale and purchase ticket id. This data, and only this data, was used to create the system of recommendation. A summary of the dataset can be seen in *Table 1* below.

Total sales	7.245.522
Unique clients	853.514
Unique Famille	9
Unique Maille	34
Unique Univers	105
Unique Libelle (products)	1.484
Unique Months	12
Null Values	0 in all columns

Table 1: Summary of dataset provided

Method

Participants

First and foremost, it must be stated that all work completed in this project was equally divided and prepared by all five members of the team.

Pre-Processing

After taking an initial look at the dataset, the first steps were to clean and organize the data, and then begin calculating Key Performance Indicators (KPI). The dataset provided contained no null values, and all column data types were accurately detected.

As a result of the project requiring rapid access to large amounts of static data, a NoSQL MongoDB was chosen to store the purchase data calculated by the model, and all relevant information for the functionality of the application.

The calculation of the KPIs was divided into two main sections: the Client data and the Store data.

Store Data

Before beginning to classify and group clients, information related to the products was essential. Once a general overview of the categories and quantities of the data was determined, the following KPIs were calculated

- Total number of products
- Products per category
- Most purchased products
- Average price per category
- Total sales

Recommender

Client Data

Once the product data was organized and sorted, the following step was to begin the classification of clients. For this to be accomplished, a sufficient amount of data had to be available for each individual client, which provided their profile and thus their personal interests. This personal interest data was then used to find the relative interests between clients and arrive at recommendations within the client dataset. A list of the client data KPIs can be seen below :

- Total number of carts
- Total expenses
- Most expensive cart
- Least expensive cart
- Average cart price
- Most expensive product
- Expenses / month
- Number of carts / month
- Purchase frequency / product
- Purchase frequency / maille
- Purchase frequency / univers
- Purchase frequency / famille
- Gender supposition

Where a cart represents all products bought by a single client. These KPIs provided the model with the price space of the dataset, as well as the frequency related to each product and its category. Therefore, it was made possible to gain an insight into a customer's behavior.

New Data

While the dataset provided a large amount of information for the client and store data, it was not sufficient to create a functional recommendation system. As a result there were several KPIs that were created and added to the above sections to furnish the recommendations systems with the required data. This information is specified in the description for each system below.

Recommendation System

There are several types of popular recommendation systems that form the majority of examples used today, being the collaborative filtering, content-based approach and knowledge based system. The latter was immediately eliminated from the possible options, since direct feedback from users is unavailable under the scope of this project. The other two options were both possible solutions, and therefore models for both were created and applied to provide the user with an option between them. Both models take a client id, as their only parameter, and return a list of product recommendations. Details of the models are explained below.

Content Based Approach

With this approach, recommendations are created by comparing the similarity between the items. Since the data set provided was quite limited with respect to product information, an Natural Language Processing (NLP) model was created that produces a product similarity matrix based off of the text columns provided: *Libelle, Maille, Univers and Famille*.

Taking a client id, the model vectorized the products already purchased by the client and finds the closest product vectors to them in the product space created by the dataset. While this method at first seemed to be illogical, based on the quantity of text available to describe each product, it was found to provide accurate results in a minimal amount of time.

Collaborative Filtering

Where the first approach is based on the content of each item, which was minimal, this approach takes advantage of the size of the dataset to determine similarity between purchases. This is done by splitting the data into two sections :

Recommender

1. The product data, containing their information provided and adding their pricing category.
2. The client data, containing their product purchases with the quantity for each product and their rank for said purchase.

In other words, these sections represent the *item to item* matix and the *client to item* matrix. These two matrices provide the ability to vectorize the items purchased by a user, calculate the user's ranking of these purchases, and then attribute a ranking for each user to every product available that they have not yet purchased. Since the resulting predictions matrix requires a significant amount of computing power to parse, the top 5 recommendations for each client were stored in the database thus providing the application with a rapid list of recommendations given a client id.

Results

The resulting recommendations system was found to provide accurate results. The following provides details for each.

Collaborative Filtering

Based on user product rankings that were calculated, a collaborative filtering algorithm was applied to the data. The resulting prediction matrix, on a basic personal laptop, was calculated in two minutes and then stored in the database. Thus providing recommendations instantaneously to the user of the system.

Content Based

Based on the text descriptions of the product name and categories, a NLP algorithm was applied to the data to measure the similarity between products. The system provides

recommendations on an average of less than one second, which allows the system to recalculate the recommendations on each request.

Discussion

The results of the project were proven to be accurate and effective for the objectives set. However this does not imply that there are no improvements that can be made.

Firstly, the formulas applied for calculating the rank of a user for a product can be drastically improved to widen the spectrum of possibilities, which is currently at 5. While this wasn't a hurdle for the current scope, to widen the dataset and increase the probability of accurate recommendations, this will be obliged.

Secondly, the application is set on static data with no means for new users or purchases. Ideally, new users would be automatically added to the database as well as every new purchase. The models would be updated on a daily basis and not directly on every new entry, since it is computational demanding. After the results would be added to their respective database locations.

Lastly, two of the three popular recommendation systems were used in this project and found to provide correct results. However the combination of both is another methodology often applied and shown to provide the benefits of both systems. This possibility is worth exploration.

Appendix

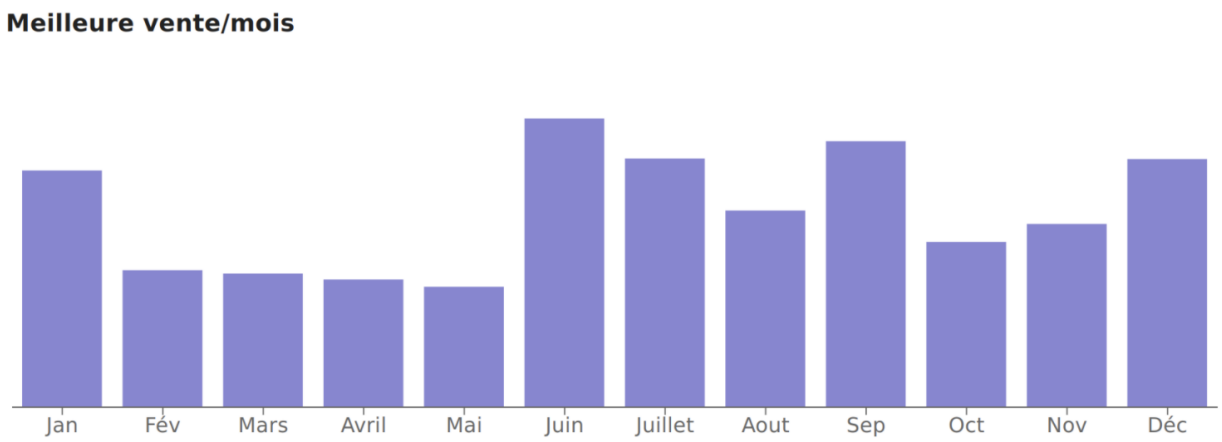

Figure 1: Best sales per month from dataset

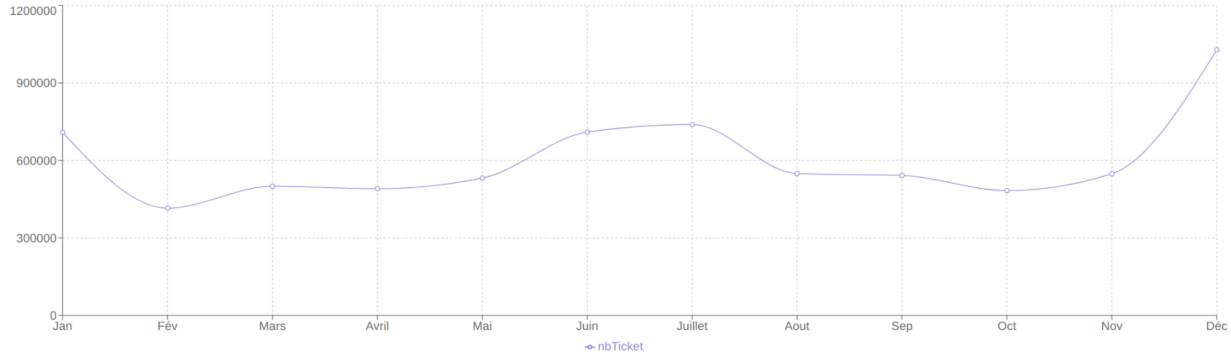

Figure 2: Total sales from dataset