Machine Learning

REPORT

ECE-GY-6143

Team:

Shubhankar Mishra - sm10095

NYU Tandon School of Engineering

Contents:

1. Abstract **2**

2. Introduction

a. Motivation 3

3. Dataset **4**

a. Review Data 4

b. Metadata 5

4. Technologies Used **6**

5. Recommendation System **6**

a. What are they? Why are they useful? 6

b. Collaborative Filtering 7

c. Content-Based Filtering 8

6.myApproach **9**

a. Data Extraction 9

b. Data Preprocessing 10

c. Content-Based Filtering 14

7. Results Discussion 16

a. Learnings 16

b. Future work & Limitations 16

8. Conclusion **17**

9. Acknowledgments **17**

1. Abstract

This report aims to concisely document the working ofmyproject that works around creating and implementing a recommendation system that aids the user by providing them with appropriate and accurate product recommendations that would be based on the user's past reviews on purchased products and other features such a description, categories, etc. Such a system is very useful for companies like Amazon, Etsy, Netflix, Hulu, etc as it can aid in boosting their sales and site traffic.

2. Introduction

a. Motivation

The motivation behind using a Recommender system is that it is a data science application that is based on a customer's previous reviews or browsing behavior, and I can implement data science in order to find and recommend products for the potential customer.

3. Dataset

The first step in building a recommendation system is to collect data in order to employ it in the system. For each top level category in the dataset, there are two forms of data involved and used for analysis inmydataset: Review data, and Metadata.

AMAZON FASHION	reviews (883,636 reviews)	metadata (186,637 products)
All Beauty	reviews (371,345 reviews)	metadata (32,992 products)
Appliances	reviews (602,777 reviews)	metadata (30,459 products)
Arts Crafts and Sewing	reviews (2,875,917 reviews)	metadata (303,426 products)
Automotive	reviews (7,990,166 reviews)	metadata (932,019 products)
Books	reviews (51,311,621 reviews)	metadata (2,935,525 products)
CDs and Vinyl	reviews (4,543,369 reviews)	metadata (544,442 products)
Cell Phones and Accessories	reviews (10,063,255 reviews)	metadata (590,269 products)
Clothing Shoes and Jewelry	reviews (32,292,099 reviews)	metadata (2,685,059 products)
Digital Music	reviews (1,584,082 reviews)	metadata (465,392 products)
Electronics	reviews (20,994,353 reviews)	metadata (786,868 products)
Gift Cards	reviews (147,194 reviews)	metadata (1,548 products)
Grocery and Gourmet Food	reviews (5,074,160 reviews)	metadata (287,209 products)
Home and Kitchen	reviews (21,928,568 reviews)	metadata (1,301,225 products)
Industrial and Scientific	reviews (1,758,333 reviews)	metadata (167,524 products)
Kindle Store	reviews (5,722,988 reviews)	metadata (493,859 products)
Luxury Beauty	reviews (574,628 reviews)	metadata (12,308 products)
Magazine Subscriptions	reviews (89,689 reviews)	metadata (3,493 products)
Movies and TV	reviews (8,765,568 reviews)	metadata (203,970 products)
Musical Instruments	reviews (1,512,530 reviews)	metadata (120,400 products)
Office Products	reviews (5,581,313 reviews)	metadata (315,644 products)
Patio Lawn and Garden	reviews (5,236,058 reviews)	metadata (279,697 products)
Pet Supplies	reviews (6,542,483 reviews)	metadata (206,141 products)
Prime Pantry	reviews (471,614 reviews)	metadata (10,815 products)
Software	reviews (459,436 reviews)	metadata (26,815 products)
Sports and Outdoors	reviews (12,980,837 reviews)	metadata (962,876 products)
Tools and Home Improvement	reviews (9,015,203 reviews)	metadata (571,982 products)
Toys and Games	reviews (8,201,231 reviews)	metadata (634,414 products)
Video Games	reviews (2,565,349 reviews)	metadata (84,893 products)

a. Review Data

The Review Data contains reviews given by users who purchased this product. I can identify the product with its unique id ASIN (Amazon Standard Identification number), which is present for every product that is on the Amazon Website.

The review data has characteristic attributes, which are: ASIN, Summary, Overall Ratings, Style, Review - text, summary etc.

Attached below is the screenshot of dataset schema for Review Data

```
root
 |-- asin: string (nullable = true)
 |-- image: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- overall: double (nullable = true)
 |-- reviewText: string (nullable = true)
 |-- reviewTime: string (nullable = true)
 |-- reviewerID: string (nullable = true)
 |-- reviewerName: string (nullable = true)
 |-- style: struct (nullable = true)
 |    |-- Color:: string (nullable = true)
 |    |-- Format:: string (nullable = true)
 |    |-- Size:: string (nullable = true)
 |-- summary: string (nullable = true)
 |-- unixReviewTime: long (nullable = true)
 |-- verified: boolean (nullable = true)
 |-- vote: string (nullable = true)
```

b. Metadata

Metadata holds the data pertaining to the products which are differentiated into different categories, I can identify the product with its unique id ASIN (Amazon Standard Identification number), which is present for every product that is present on the Amazon Website.

The data has characteristic attributes, some of which are exclusive to this dataset such as: ASIN, Brand, Category, Description, Price, Ratings, Views, also_buy(similar products bought by users), also_view(similar products viewed by users) etc

Attached below is the screenshot of the dataset schema

```
root
 |-- also_buy: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- also_view: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- asin: string (nullable = true)
 |-- brand: string (nullable = true)
 |-- category: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- date: string (nullable = true)
 |-- description: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- details: struct (nullable = true)
 |    |-- \n    Item Weight: \n    : string (nullable = true)
 |    |-- \n    Product Dimensions: \n    : string (nullable = true)
 |    |-- ASIN:: string (nullable = true)
 |    |-- ASIN: : string (nullable = true)
 |    |-- Apparel: string (nullable = true)
 |    |-- Audio CD: string (nullable = true)
 |    |-- Audio Cassette: string (nullable = true)
 |    |-- Blu-ray Audio: string (nullable = true)
 |    |-- DVD: string (nullable = true)
 |    |-- DVD Audio: string (nullable = true)
 |    |-- Label:: string (nullable = true)
 |    |-- MP3 Music: string (nullable = true)
 |    |-- Note on Boxed Sets:: string (nullable = true)
 |    |-- Number of Discs:: string (nullable = true)
 |    |-- Original Release Date:: string (nullable = true)
 |    |-- Please Note:: string (nullable = true)
 |    |-- Run Time:: string (nullable = true)
 |    |-- SPARS Code:: string (nullable = true)
 |    |-- Shipping Weight:: string (nullable = true)
 |    |-- UPC:: string (nullable = true)
 |    |-- Vinyl: string (nullable = true)
 |    |-- Vinyl Bound: string (nullable = true)
 |-- feature: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- fit: string (nullable = true)
 |-- imageURL: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- imageURLHighRes: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- main_cat: string (nullable = true)
 |-- price: string (nullable = true)
 |-- rank: string (nullable = true)
 |-- similar_item: string (nullable = true)
 |-- tech1: string (nullable = true)
 |-- tech2: string (nullable = true)
 |-- title: string (nullable = true)
```

4. Technologies Used

All code was written and compiled in Python using Jupyter Notebooks. I usedmyknowledge of Big Data and implementedmyrecommendation system using PySpark.

I used and implemented my working knowledge of Spark in order to work with a large dataset and efficiently execute this project. I used both Spark and SparkML.

For data preprocessing I utilized several libraries from SparkML, namely: CountVectorize, StringIndexer, LSH, Tokenizer.

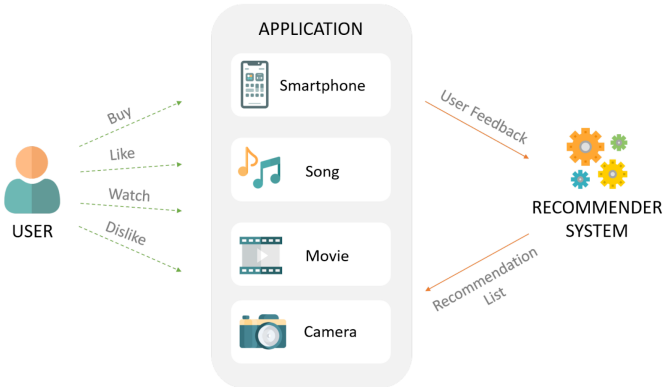
5. Recommendation System

a. What are they? Why are they useful?

A Recommendation System is a data science application that is used to predict or offer products to customers based on their past purchase, browsing history, reviews etc. It employs several Machine Learning algorithms which help in predicting possible recommendations for users.

A recommendation system can have a variety of applications, for example, companies like Netflix, IMDB, Youtube, etc. Have been using recommendation systems for tailoring their content recommendation in order to create a vast array of content that is differentiated by the content consumed by the user.

The mechanism behind the working of a recommendation system is that it has an engine that helps in filtering the data using different algorithms and recommends the most relevant items to users. There are many techniques and algorithms used. In this project I will deal with two of them namely. Collaborative filtering and Content-Based filtering.



Picture cited from google

But a recommendation system has several steps involved that lead to the recommendation these are:

I. Data Collection
II. Data Storage
III. Data Filtering

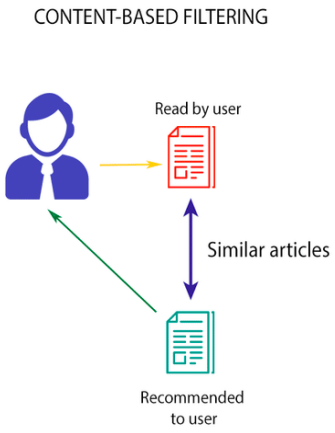
Out of all the steps involved in creating a well functioning Recommendation System, the most important step is Data Filtering.

b. Content-Based Filtering

The other famous approach is Content-Based Filtering. Where the characteristics and features of a product are used to find similar products. It is best suited for scenarios where the similarity between different products is calculated on the basis of the attributes of products. An example of its working would be for a movie recommender, the similarities between the movies is calculated on the basis of genres, actor, director etc.

Content-based recommenders treat recommendation as a user-specific classification problem and tailor for the classifier to learn from the user's behavior, such as the likes and dislikes based on product features.

Inmyapproach I will be using the description, categories a products belongs and its main features to calculate similarities between products.



The diagram above is from a Medium article

6. Approach

In this section I will covermyapproach of how I solved the problem. The approach includes four main steps. The first step is extracting the data from the dataset. The second step is preprocessing the data, cleaning them up, and bringing them in a format that is friendly both for meas developers to use for the filtering algorithms and for third parties readingmyproject. Finally the last step include creating recommendations by performing content based filtering.

a. Data Extraction

As discussed in the Dataset Description section, this Amazon datasets provides many top level categories for reviews and metadata (products).myintent is to makemysystem work with any of those. Or any combination of those. For the purposes of this report and the presentation I will be using data from the Software category. This includes 500,000 reviews for roughly 27,000 products. Some of the other top level categories provided reviews to the magnitude of millions however some of these were very large and with a limited memory capacity on Jupyter I tried to avoid them. However, as mentioned abovemysystem would work similarly well with any category, or combination of categories, as the principle is the same.

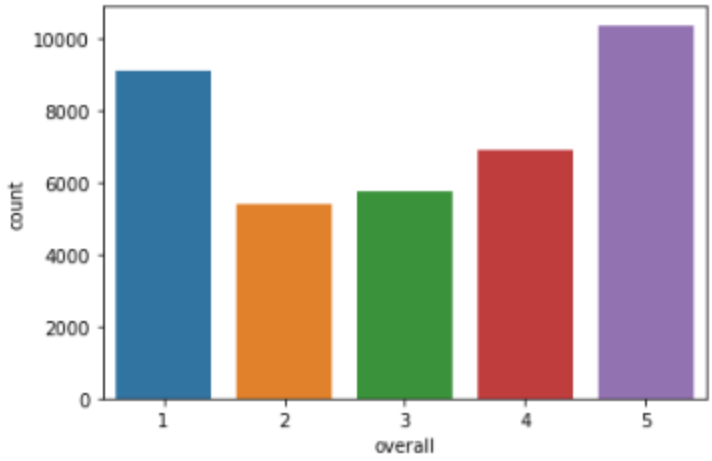
Extracting and importing the data into pyspark is actually very simple. Using the !get command I download the json files intomyJupyter local directory, then these can simply be imported into dataframes.

```
# Downloading the same file many times it will just create copies in your home directory, avoid it.
!wget http://deepyeti.ucsd.edu/jianmo/amazon/metaFiles2/meta_Software.json.gz
!wget http://deepyeti.ucsd.edu/jianmo/amazon/categoryFiles/Software.json.gz
```

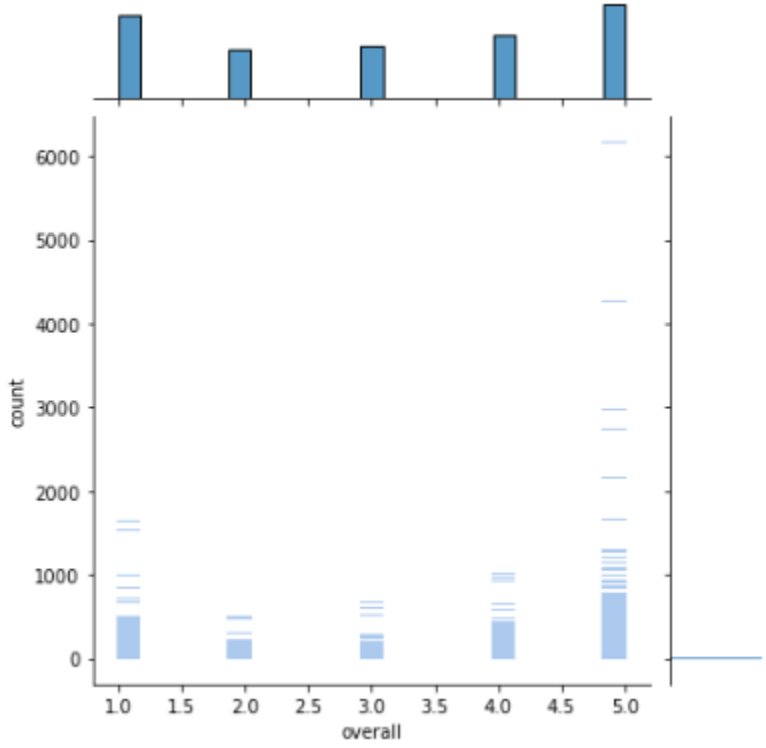
```
# Load data into dataframes
products = spark.read.json("meta_Software.json.gz")
reviews = spark.read.json("Software.json.gz")
```

b. Data Visualisation

Data Visualisation is an important step as it helps us to better understand the data that we are working with.



A Countplot of the Ratings for all Existing products

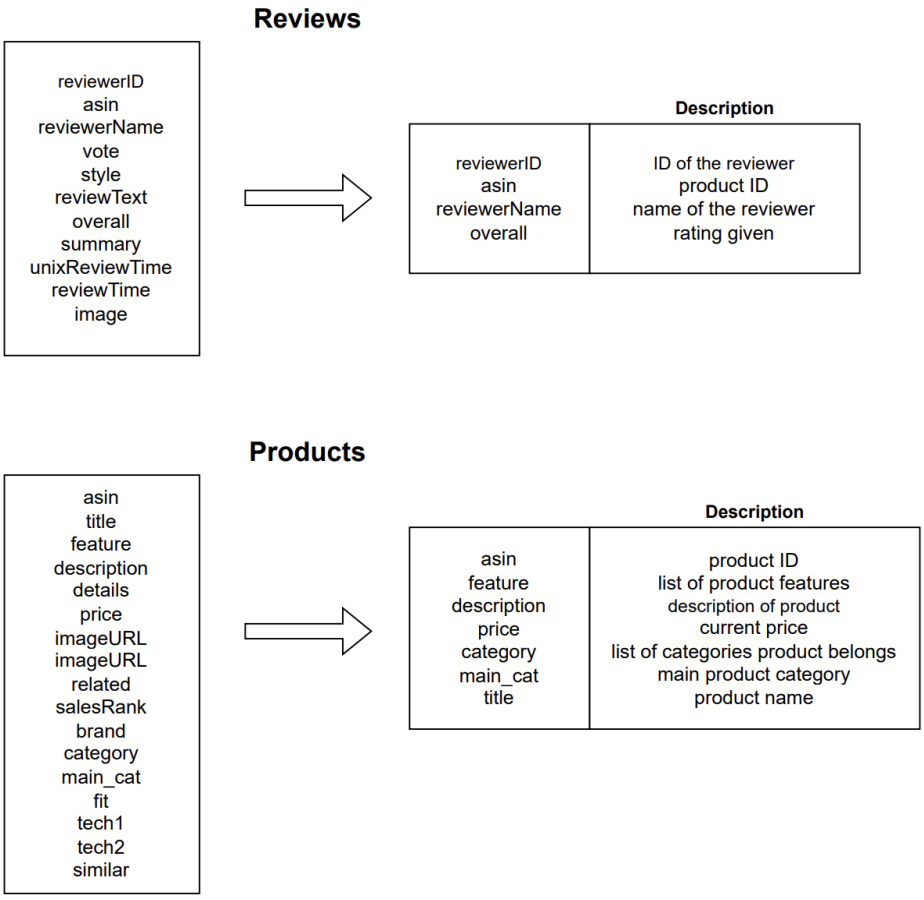


A jointplot using a Histogram plotting the Ratings v Product ID

c. Data Preprocessing

Both products and reviews dataframes have many columns, I will not be using all of them. Therefore, it is a good practice to drop the columns I will not be needing.

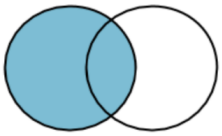
Specifically these are the columns that are useful formyrecommendation system:



This was the first step of data preprocessing. At this point I inspected the products dataframe and realized that there Ire a few rows where “categories” and “feature” columns were empty. Therefore I decided to remove these rows as it will make performance of content based filtering worse and also created errors.

```
products = products.where(size('category') > 0).where(size('feature') > 0)
```

Next I realized that there are some reviews for products that do not exist in the products dataset. Therefore I wanted to remove these reviews. This is done by performing a left semi join on asin (productID).



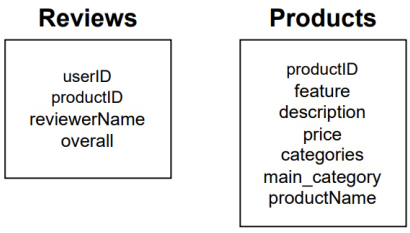
Left Join

on reviews.asin == products.asin

```
reviews = reviews.join(products, reviews["asin"] == products["asin"],"leftsemi")
```

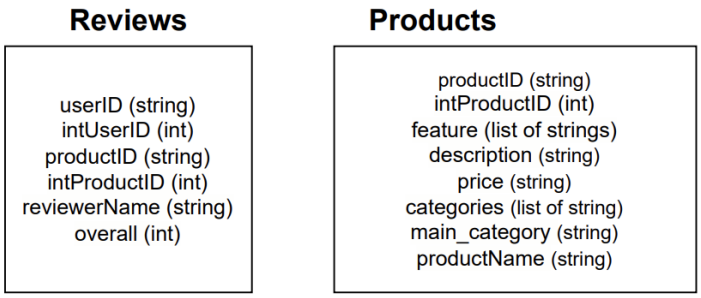
Next step was a simple duplicate removal, as both dataframes contained a few duplicates. At this point I wanted to make the dataset more understandable to third parties, so I decided to perform some column renaming.

Dataframes after the renaming process:



The ML library that I used for Collaborative filtering requires that userID and productID are of Integer type, currently they are strings containing both numbers and characters (eg. B000OON842). Thus, at this point I used StringIndexer from MLib to convert these two columns to unique Integer IDs. In addition, overall (overall contained values 0.0, 1.0, 2.0 … 5.0, not values in between it denoted the number of stars given by the user) was also converted with a simple cast to Integer from Double.

This is the final state of the dataframes after preprocessing:
At this point I also saved them into two dataframes to be used by other notebooks ofmyproject.



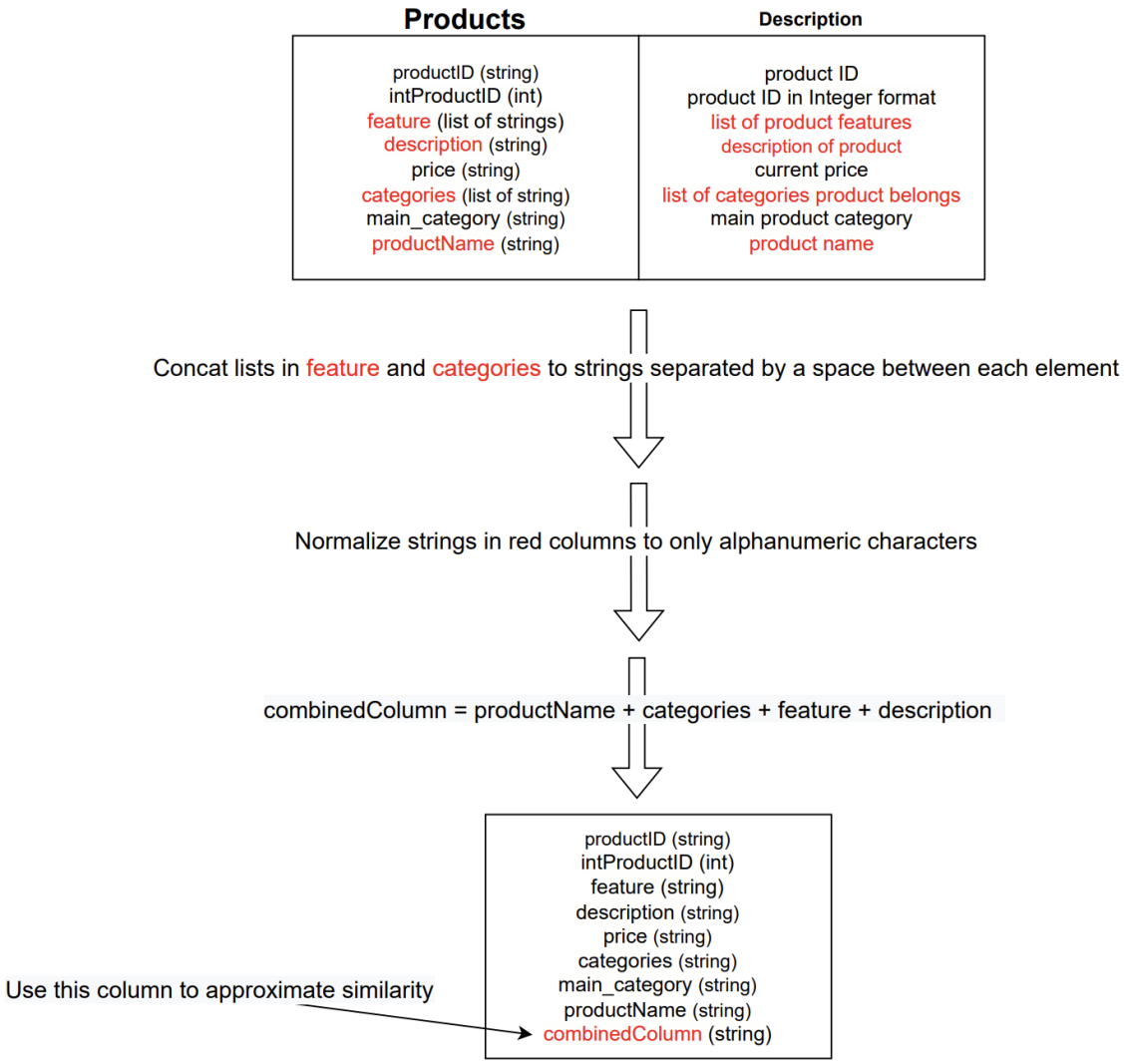
d. Content-Based Filtering

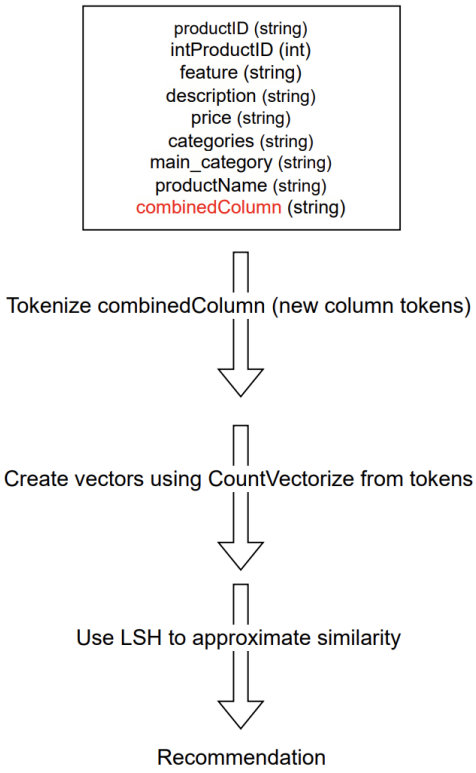
The idea behind content based filtering is to use a product's characteristics in order to find similar products. By finding similar products to what a user has already bought, make recommendations to them.

For content based filtering I are only using the data from the products(metadata), provided by the dataset. I ignore the reviews given by users.

Our approach here is to combine several of the columns into one column and then vectorize this column and use LSH to approximate similarities to other products.

I are using the columns in red as characteristics of each product.





Now let's try to get some recommendations. I picked a random product with the ID=3000, which turned out to be Microsoft Office. Therefore, I are trying similar products to the following:

```
+----------+--------------------+
| productID|         description|
+----------+--------------------+
|B0006I0WP4|microsoft office ...|
+----------+--------------------+
```

By performing all of the steps described above I got the following recommendations, note that from the results given from approxSimilarityJoin() in LSH I sort with ascending distance values and pick the first five.

recommendedProduct	intProductID	dist
mavis beacon teaches office xp	3000	0.0
learn office essentials	1980	6.4031242374328485
simon amp schuster walking with dinosaurs windo...	1283	7.0
tune transfer win mac jewel case	4369	7.211102550927978
pc treasures cluefinders 6th grade adventures n...	8535	7.280109889280518

I can see that the result is very promising, the recommendation products are Digital Softwares that are similar to Microsoft Office. Overall the algorithm and approach worked really well.

7. Results

a. Learnings

Throughout this project I gained a lot of knowledge both theoretical and technical on how recommendation systems work. I had the opportunity to work with Machine learning
I learned that Content-based Recommendation Systems might not work appropriately some times. For example, imagine the case where a user buys a mobile phone. Using this mobile phone product and content based recommendations, more mobile phones will be recommended. It would be more appropriate to recommend accessories for the phone not devices.

I used and implemented my working knowledge of Spark in order to work with a large dataset and efficiently execute this project

b. Future work & Limitations

One of the things that limits my system is the RAM limit on Jupyter Notebook. I used 500,000 reviews to create the model containing only Software products. I would be very interested to test my system with millions of reviews from multiple product categories.myexpectation is that it would work equally well. Therefore in the future I aim to use a real environment with multiple clusters that will allow me to deal with many gigabytes of data. Another improvement would be to use more features of the dataset for content based recommendation. Another improvement would be to implement collaborative filtering and combine them both in order to create a Hybrid system.

8. Conclusion

In conclusion, this was a very exciting project. I gained a deep insight into recommendation systems. The recommendation system I implemented Content-Based filtering, provided very good results. I even dealt with different obstacles, some I overcame and some I didn't, thus creating limitations. I also understood how my project could be improved in the future.

9. Acknowledgments

- I would like to cite the following paper for motivating my research : **Justifying recommendations using distantly-labeled reviews and fined-grained aspects by** Jianmo Ni, Jiacheng Li, Julian McAuley *Empirical Methods in Natural Language Processing (EMNLP)*, 2019

- MLlib: https://spark.apache.org/docs/latest/ml-features.html#stringindexer

- Dataset: http://deepyeti.ucsd.edu/jianmo/amazon/

- Understanding the working of a recommendation system using Spark:

  https://towardsdatascience.com/building-a-recommendation-system-with-spark-ml-and-elasticsearch-abbd0fb59454

  https://schaper.io/2017/10/building-a-recommendation-engine-with-spark-and-emr/

- The Data used in the project is similar to my Project Submission for my Big Data course CS GY 6513, but I am submitting only a Content-Filtering based approach, and only the data used is similar. And any code implemented is unplagiarised as it is my own work.