

Nordic Academy

Recommendation System

For e-Commerce using Collaborative Filtering

Amruthaa S

04-Oct-2021

ACKNOWLEDGEMENTS

The success of this project would not have been possible without the encouragement and support of my mentor, Dr. Gnanavel Mutharasu. I would also like to express my gratitude to Nordic Academy for providing me with the opportunity to work on this project.

ABSTRACT

A recommendation system is used to provide personalized suggestions or recommendations for products or services that a user is interested in. These systems have gained a lot of attention for use in areas such as e-Commerce, retail, telecom, tourism etc. as they increase user satisfaction and increase revenue for service providers. Various techniques for creating recommendation systems have been developed, item-based collaborative filtering being the one used for this project. The dataset utilized for this project contains product information, namely the product name, category tree, product description and brand. Similarities between products were computed using a cosine similarity function. A recommendation system was created using this data and was then executed through the command-line interface using command-line arguments. This report gives an insight into the technique used to create this recommendation system, and the workflow of this particular project. The complete code for this project can be found at <https://github.com/amruthaa08/Recommendation-system> .

CONTENTS

1	Introduction	7
2	Techniques.....	8
2.1	Collaborative Filtering.....	8
2.1.1	Item-based Collaborative Filtering.....	9
2.2	Cosine-based Similarity	9
3	Workflow	10
3.1	Acquiring data.....	10
3.2	Cleaning data.....	10
3.3	Preprocessing data	11
3.4	Applying cosine function.....	11
3.5	Getting recommendations	12
3.6	Using a Command Line Interface (CLI)	13
3.6.1	CLI Demo	13
4	Conclusion.....	14
5	Further developments.....	15
6	References.....	16

List of Figures

Figure 1 Techniques used to create Recommendation Systems.....8

Figure 2 Mathematical formula for cosine similarity.....9

Figure 3 Project workflow.....10

Figure 4 Visualizing similarity scores.....12

Figure 5 CLI Demo.....13

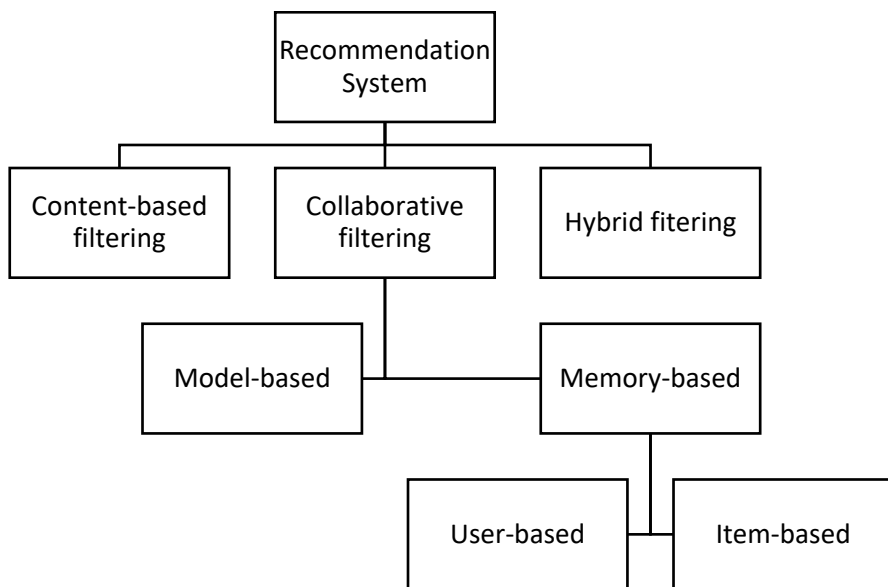
1 INTRODUCTION

Recommender systems can be defined as programs which attempt to recommend the most suitable items (products or services) to particular users, by predicting a user's interest in an item. This can be based on information about the items, the users, or user activity. The aim of developing recommender systems is to filter and obtain the most relevant information and products from a huge amount of data, thereby providing personalized services. The applications of recommender systems include recommending movies, music, television programs, books, documents, websites, conferences, tourism scenic spots and learning materials, and involve the areas of e-commerce, e-learning, e-library, e-government and e-business services. The objectives of this project were to create a recommendation system using item-based collaborative filtering, and to execute this system using command line arguments.

2 TECHNIQUES

Different techniques have been developed to create recommendation systems, as seen in Figure 1. The subsequent sections will be explaining the particular techniques used in this project.

Figure 1. Techniques used to create Recommendation Systems



2.1 Collaborative Filtering

In the context of recommendation systems, Collaborative Filtering (CF) is a method of making predictions about the interest of a user by analyzing the taste of similar users. Multiple viewpoints are collaborated to filter data; hence the name 'collaborative filtering'. CF can be model-based or memory-based. Model based CF uses machine learning models to learn and predict ratings, whereas memory-based CF uses the similarity between items or users to filter data.

2.1.1 Item-based Collaborative Filtering

In the item-based CF approach, the similarity between items is used to provide recommendations for a particular item. The main step in item-based CF is computing the similarity between items. Several similarity measures such as Pearson correlation-based similarity, constrained Pearson correlation-based similarity, cosine-based similarity, or adjusted cosine-based measures are available, out of which the cosine-based similarity has been utilised in this project.

2.2 Cosine-based Similarity

Cosine similarity is a metric used to measure how similar items are. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. It is the dot product of the two vectors divided by the product of the two vectors' lengths (or magnitudes).

Figure 2. Mathematical formula for cosine similarity

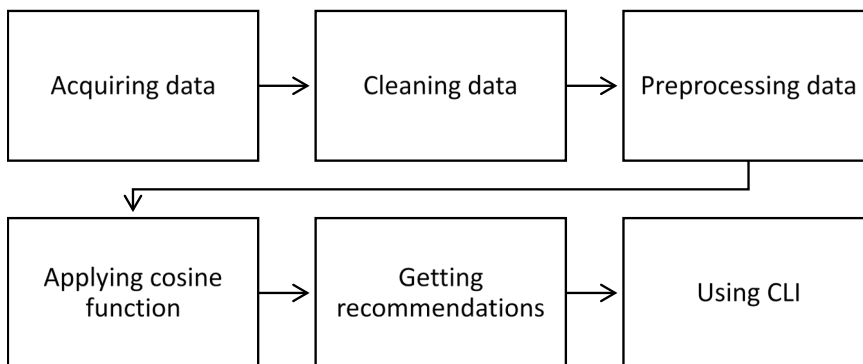
$$\text{Cos}\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$ is the dot product of the two vectors.

3 WORKFLOW

Figure 3 illustrates the workflow of the project, from acquiring data to executing the system through a command line interface.

Figure 3. Project workflow



This project utilises open-sourced data from Kaggle. The original dataset contains 15 fields, and was created by extracting data from Flipkart.com, a widely-used e-Commerce website in India.

3.2 Cleaning data

In this step, unwanted columns were removed from the data. Only columns containing the product name, product category tree, product description and brand

were retained, since item-based collaborative filtering requires only product-related features. Duplicate rows were removed from the data and fill values were added as required.

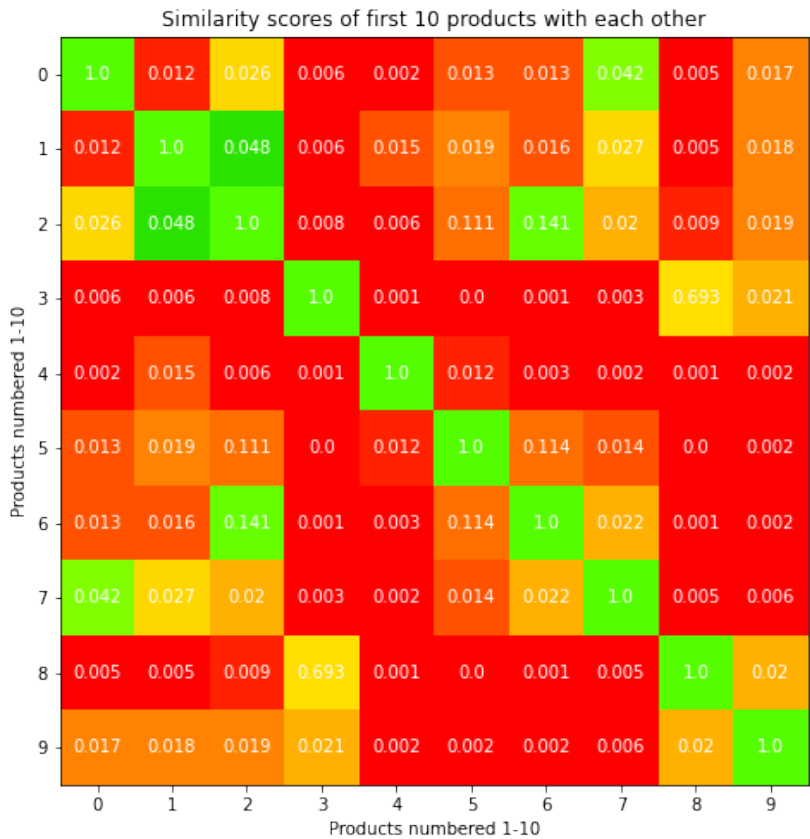
3.3 Preprocessing data

The text in the data was preprocessed using the natural language toolkit (nltk) python library. the data was then tokenized and stop words and punctuation were removed from the text. Stemming was applied using the Porter Stemmer. This resulted in preprocessed data ready for applying the similarity measure.

3.4 Applying cosine function

Features were extracted from the data using the `TfidfVectorizer` from the `scikit learn` python module. This returned a sparse matrix containing the extracted features. Cosine similarity was then applied to these extracted features. This in turn returned a similarity matrix containing the similarity scores of all of the products with each other. Similarity scores are numerical values that represent the similarity between any 2 items. Higher the similarity scores, more similar the products. Figure 4 shows a part of the similarity matrix obtained, containing the similarity scores of the first 10 products from the dataset with each other.

Figure 4. Visualizing similarity scores



3.5 Getting recommendations

A simple python function was created to obtain recommendations. Taking a product name as an argument, the function used the similarity scores for the given product to return the top recommendations.

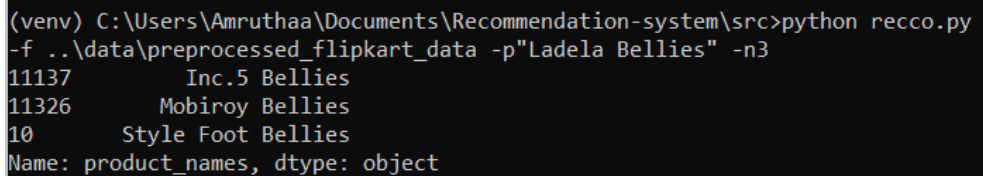
3.6 Using a Command Line Interface (CLI)

The python argparse library was used to create three command line arguments to control the recommendations obtained. The name of the file used to get recommendations, the name of the product for which recommendations were required, and the number of recommendations required were obtained from the command line. The recommendation system was then executed using the arguments.

3.6.1 CLI Demo

Figure 5 demonstrates using the CLI to execute the recommendation system.

Figure 5. CLI Demo



```
(venv) C:\Users\Amruthaa\Documents\Recommendation-system\src>python recco.py
-f ../data/preprocessed_flipkart_data -p "Ladela Bellies" -n3
11137      Inc.5 Bellies
11326      Mobiroy Bellies
10        Style Foot Bellies
Name: product_names, dtype: object
```

The ‘python’ command is used to run the file ‘recco.py’ to execute the recommendation system. Three arguments are provided to the CLI; the ‘f’ argument obtains the name of the file containing data used to get recommendations, which in this case is ‘../data/preprocessed_flipkart_data’. The ‘p’ argument is used to obtain the name of the product for which recommendations are required, which in this case is “Ladela Bellies”. The final argument, ‘n’, obtains the number of recommendations required for the given product, which in this case was 3. Hence three recommendations were returned for the product “Ladela Bellies”.

4 CONCLUSION

Recommendation systems are a powerful way of obtaining additional value for a business from its item or user databases. These systems help users find and identify items they want to buy from a business. Conversely, they help the business by generating more sales and revenue. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web.

This report gave an insight into the techniques and workflow used in this project to create a recommendation system. A recommendation system for e-Commerce was created using collaborative filtering. The system was then executed through a command line interface, using command line arguments.

5 FURTHER DEVELOPMENTS

This project can be developed further by adding different choices of algorithms to get the similarity between products. Collaborative filtering can be applied to datasets containing user reviews and user activity to create a recommendation system. Such different phases of recommendation systems could be integrated to form a single pipeline.

6 REFERENCES

- [1] PromptCloud. (2017). Flipkart Products. [Online].
<https://www.kaggle.com/PromptCloudHQ/flipkart-products>. [Accessed 9 Sep 2021]
- [2] Prabhakaran, S. (2018, Oct 22). Cosine Similarity – Understanding the math and how it works (with python codes). <https://www.machinelearningplus.com/nlp/cosine-similarity/>
- [3] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, Guangquan Zhang, Recommender system application developments: A survey, Decision Support Systems, Volume 74, 2015, Pages 12-32, ISSN 0167-9236, <https://doi.org/10.1016/j.dss.2015.03.008>.
- [4] Sarwar, Badrul & Karypis, George & Konstan, Joseph & Riedl, John. (2001). Item-based Collaborative Filtering Recommendation Algorithms. Proceedings of ACM World Wide Web Conference. 1. 10.1145/371920.372071.