

# **Recommendation System for Anonymous Microsoft Web Data**

CMPE - 256 Large Scale Analytics Team Project Report

## **PROJECT GROUP 8**

Project Link - <https://github.com/busipallavi-reddy/cmpe256-project/>

### **Team Members**

Busi Pallavi Reddy <[busipallavi.reddy@sjsu.edu](mailto:busipallavi.reddy@sjsu.edu)> (013852800)

Maunil Swadas <[maunil.swadas@sjsu.edu](mailto:maunil.swadas@sjsu.edu)> (013850122)

Sarthak Sugandhi <[sarthak.sugandhi@sjsu.edu](mailto:sarthak.sugandhi@sjsu.edu)> (013848497)

### **Guided By**

Professor Gheorghii Guzun

# **CONTENTS**

1. INTRODUCTION
  - 1.1 INTRODUCTION AND MOTIVATION
  - 1.2 PROBLEM STATEMENT
2. PROPOSED APPROACH
  - 2.1 MODELS USED
    - 2.1.1 ITEM BASED CF
    - 2.1.2 USER BASED CF
    - 2.1.3 SVD
  - 2.2 SYSTEM DESIGN
  - 2.3 TOOLS USED
3. EXPERIMENTS AND RESULTS
  - 3.1 DATASET DESCRIPTION
  - 3.2 DATA PRE-PROCESSING
  - 3.3 RECOMMENDATION USING ITEM BASED CF
  - 3.4 RECOMMENDATION USING USER BASED CF
  - 3.5 RECOMMENDATION USING SVD
  - 3.6 EVALUATION METRICS
4. CONCLUSION

# 1. INTRODUCTION

## 1.1 INTRODUCTION AND MOTIVATION

With the boom in the field of Information technology, came the internet and social media. With the rise of platforms like Netflix, Amazon, YouTube, billions of people have started using these sites. For better user experiences and luring the customers, the big giants started working on recommender systems. Recommendation systems are models used to suggest relative items to users.

Amazon uses recommendation systems to suggest items that a targeted user may be interested in buying. Netflix uses recommendation systems to recommend movies and series of similar taste to a particular user. With such strategies, the customer base of the companies increases and thus their total revenue. From e-commerce to online advertising, recommendation systems have become very famous.

In this project we are using Microsoft web data(users and URLs visited), to recommend URLs to a user that he or she might visit. We have used Collaborative filtering to recommend URLs based on similar users and similar items. We have also used SVD(Single Value Decomposition) for dimensionality reduction and to recommend URLs.

## 1.2 PROBLEM STATEMENT

Anonymous Microsoft Web Data is a week long collection of randomly sampled users visiting [www.microsoft.com](http://www.microsoft.com). It contains the URL information, users and the sites they have visited. Our objective is to build a recommendation engine which recommends users with URLs they might be interested in.

### *Recommending URLs to users based on Collaborative Filtering*

Used Item-Item Based Collaborative filtering and User-Item based Collaborative Filtering to recommend users with URLs. This uses most similar users and items information for recommendation.

### *Dimensionality Reduction and Recommendation using SVD*

SVD is used to reduce the dimensionality of the dataset and then recommend URLs to users based on the resultant matrix.

## 2. PROPOSED APPROACH

### 2.1 MODELS USED

This study uses Collaborative filtering and SVD for recommendation.

#### 2.1.1 ITEM BASED COLLABORATIVE FILTERING

Item based collaborative filtering uses the similarity between items to recommend items to a user. It is based on the history of users and items and their ratings. In this model we build a item-user utility matrix with values set as ratings or visited for a particular item-user pair. Then, the similarity between items is computed to a particular item. For each such item we get some similar items. Then we recommend a user with items the user has previously used, rated or visited.

In our study, the items are the URLs. The utility matrix will be of users vs URLs visited. The similarity for each URL is calculated against all other URLs using correlation. Based on these values, for an item, the most similar items are computed.

#### 2.1.2 USER BASED COLLABORATIVE FILTERING

User based collaborative filtering uses the similarity between users for recommending items to a user. This takes into account with which users the current user's taste matches and then predict the items accordingly. It is based on the history of users and items and their ratings. In this model we build a user-item utility matrix with values set as ratings or visited for a particular user-item pair. Then, the similarity between users is computed for all items. For each such user we get some similar users. Then we recommend a user with items the similar user has used, rated or visited.

In our study, the items are the URLs. The utility matrix will be of users vs URLs visited. The similarity for each user is calculated against the other users using correlation. Based on these values, for a user we recommend those URLs that have been visited by similar users.

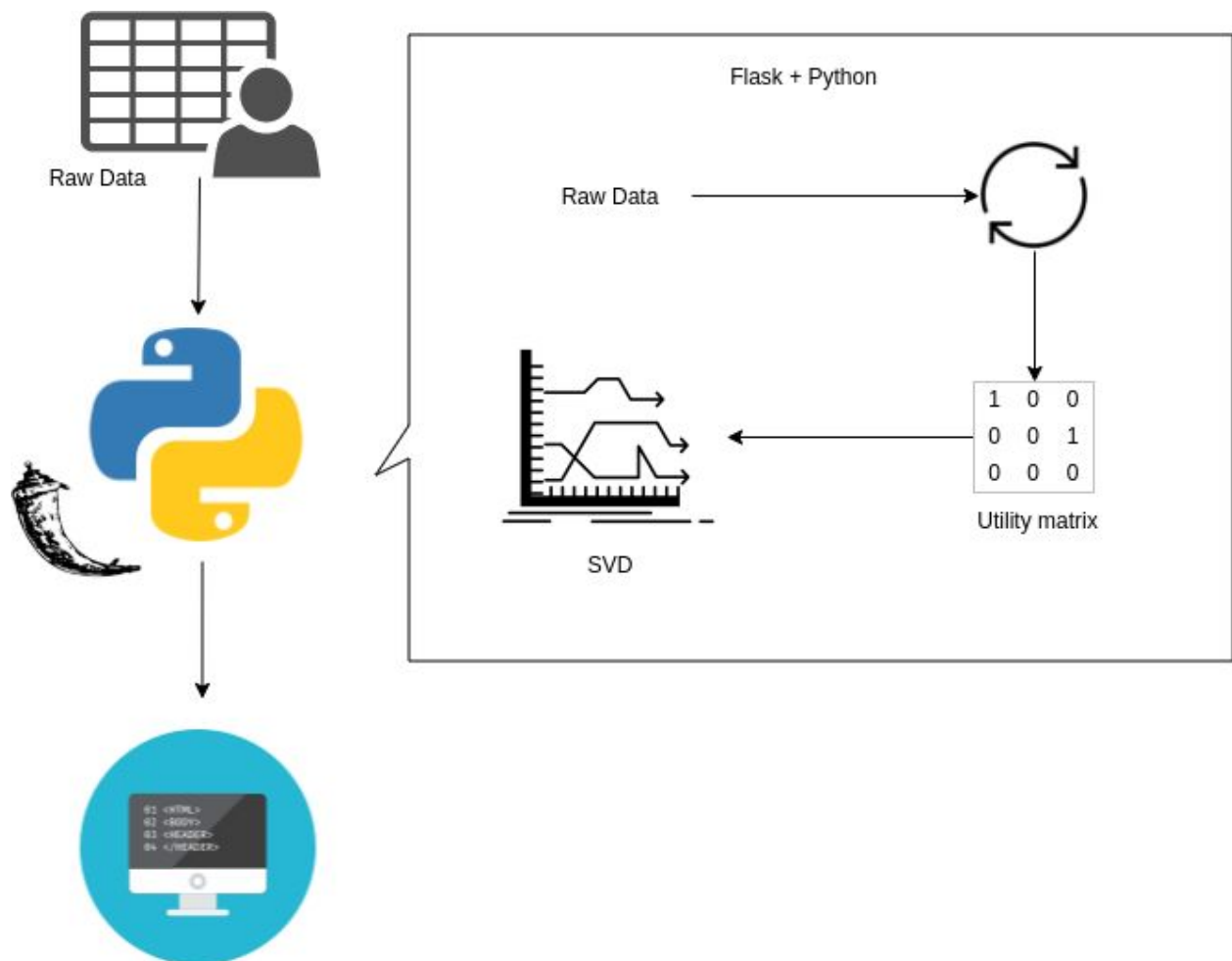
#### 2.1.3 SINGULAR VALUE DECOMPOSITION

SVD is a dimensionality reduction technique used to reduce the number of factors in the dataset. This algorithm performs matrix factorization on the user-item utility matrix to obtain a product of 3 matrices.

$$A = U\Sigma V$$

On this factorized matrix, we find the similarity between the unvisited URLs of  $U\Sigma$  with  $V$ . With this we get the similar items and then recommend users with URLs.

## 2.1.4 SYSTEM DESIGN



*Proposed Framework of the Recommendation System using SVD on Python Flask*

## 2.1.5 TOOLS USED

We have used Python3 and Jupyter Notebook. Libraries used - Pandas, numpy, scikitlearn, matplotlib. We have used Github for collaboration. Also we deployed the recommendation system using SVD as a Python Flask application. We created a webpage for viewing the recommendations as a HTML page.

### 3. EXPERIMENTS AND RESULTS

#### 3.1 DATASET DESCRIPTION

Number of Instances - 37711

Number of Attributes - 294

Attribute Characteristics - Categorical

Missing Values - N/A

Total Values - 11,087,034

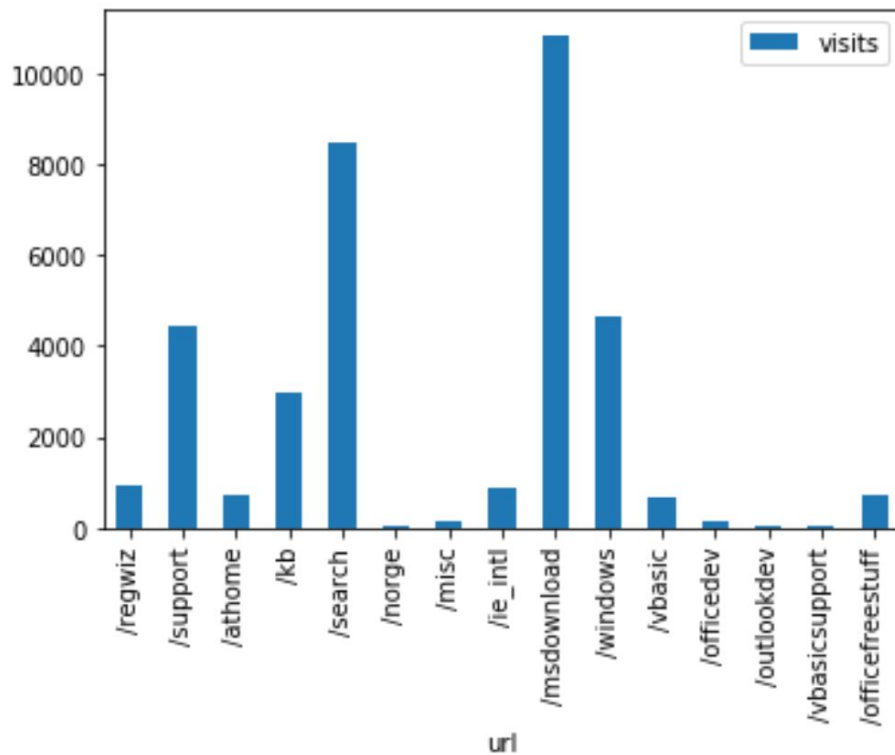
Link to Dataset - <https://archive.ics.uci.edu/ml/datasets/Anonymous+Microsoft+Web+Data>

The dataset is a collection of randomly selected 38,000 users of [www.microsoft.com](http://www.microsoft.com). It consists of the search URLs, these users visited over a week.

Users are identified by Serial Numbers like #1003.

The dataset consists of 3 types of lines[1]:

- Attribute lines:  
A,1205,1,"Hardware Supprt","/hardwaresupport"  
Where:  
'A' marks this as an attribute line,  
'1205' is the attribute ID number for an area of the website  
(called a Vroot),  
'1' may be ignored,  
"Hardware Supprt" is the title of the Vroot,  
"/hardwaresupport" is the URL relative to "<http://www.microsoft.com>"
- Case and Vote Lines:  
For each user, there is a case line followed by zero or more vote lines.  
For example:  
C,"10027",10027  
V,1008,1  
V,1046,1  
V,1034,1  
Where:  
'C' marks this as a case line,  
'10027' is the case ID number of a user,'V' marks the vote lines for this case,  
'1008', 1046', 1034' are the attributes ID's of Vroots that a user visited.  
'1' may be ignored.



*Bar Graph showing URLs vs Number of Visits*

### 3.2 DATA PRE-PROCESSING

For the purpose of recommendations, we transformed the dataset into a utility Matrix of User-Item, where each entry is either a 1(if a user has visited a URL) or 0(if a user hasn't visited the URL). The attribute 'A' line gives the information on the URLs. The attribute 'C' line has the user information followed by the attribute 'V' lines which are the URLs that the user in the 'C' line visited. This matrix is used by Collaborative filtering and SVD techniques for URL recommendation.

### 3.3 RECOMMENDATION USING ITEM BASED CF

On the above formed Utility Matrix, this method is used. As the dataset is for 38000 users and 294 websites, the utility matrix was too sparse. So we used the 100 most popular URLs as our items. On this, we computed the correlation for an item with other items.

For the item '/athome', the similar URLs with similarity is as follows:

/athome	1.000000
/support	0.076860

```
/windowssupport 0.067837
/moneyzone      0.056557
/windows        0.050309
```

So for users who have visited '/athome', we will also recommend them /support, /windowssupport, /moneyzone, /windows URLs.

### 3.4 RECOMMENDATION USING USER BASED CF

On the above formed Utility Matrix, this method is used. As the dataset is for 38000 users and 294 websites, the utility matrix was too sparse. So we used the 100 most popular URLs as our items. On this, we computed the correlation for a user with other users.

For example for the user with ID 10011, the similar users with their similarity are:

```
user
21000 0.74000
16662 0.70014
17563 0.70014
37130 0.70014
21724 0.70014
```

So the users similar to user 10011 are users 2100, 16662, 17563, 37130, 21724. So the URLs visited by these similar users might be of interest to user 10011. The URLs visited by these similar users are:

	visited	visits	ID	title	url
287	1.0	5330	1018	isapi	/isapi
212	1.0	5108	1017	Products	/products
78	1.0	4451	1001	Support Desktop	/support
138	1.0	287	1016	MS Excel	/excel

Out of these, URLs visited by user 10011 are '/excel', '/isapi', '/mspowerpoint', '/products'. So we recommend the unvisited URL, '/support' to the user 10011.

To avoid cold start for a new user, we are recommending the most popular websites to such users. For a new user, the recommended URLs are '/msdownload', '/ie', '/search', '/isapi', '/products'

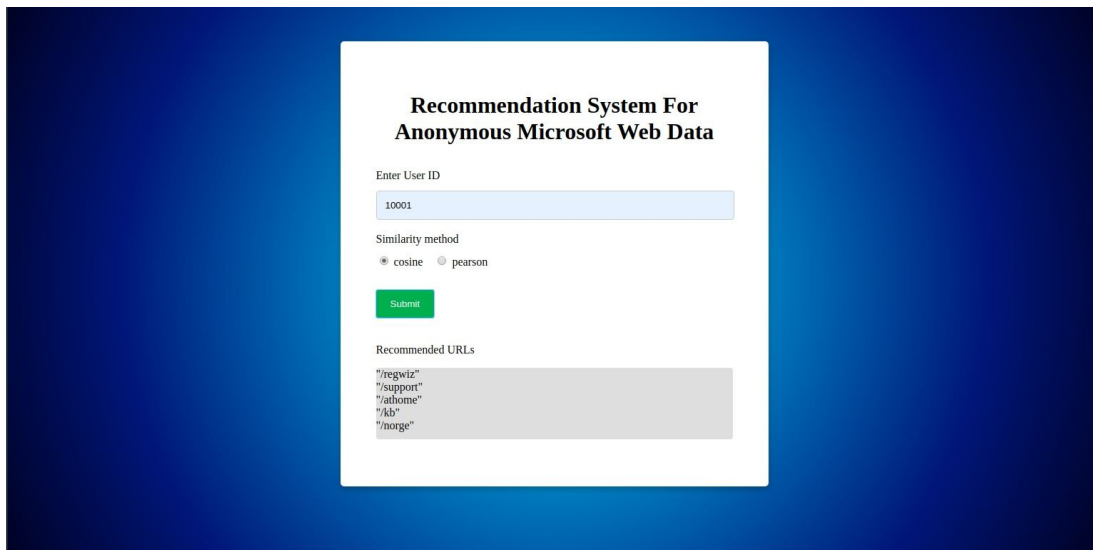


### 3.5 RECOMMENDATION USING SVD

On the utility matrix, SVD matrix factorization was done using numpy. On this, 90% energy was expressed in the matrix in the first 65 sigma.

For the user 10101, we gave the following recommendations:

	description	url
X1000	"regwiz"	"/regwiz"
X1002	"End User Produced View"	"/athome"
X1004	"Microsoft.com Search"	"/search"
X1005	"Norway"	"/norge"
X1006	"misc"	"/misc"



**Recommendation System For Anonymous Microsoft Web Data**

Enter User ID  
10001

Similarity method  
☒ cosine ☐ pearson

Submit

Recommended URLs  
"/regwiz"  
"/support"  
"/athome"  
"/kb"  
"/norge"

UI showing recommended URLs for 10001

### 3.6 EVALUATION METRICS

We used RMSE(Root Mean Squared Error) to evaluate the different models. Partitioned the dataset - 0.9 for training and 0.1 for testing purposes. RMSE for the different techniques used are:

SVD with Pearson Correlation - 0.8745362102

User Based Collaborative Filtering - 0.9112323498

Item Based Collaborative Filtering - 0.9111432323

## 4. CONCLUSION

After evaluating Item and User Based Collaborative filtering models and SVD, we observed that we obtained better results using SVD. On top of this recommender system, we built a python flask application that returns the recommended URLs using SVD with a selected type of similarity (pearson/cosine). This study explores Recommendation of URLs on Anonymous Microsoft Web Dataset using SVD, User and Item Based Collaborative Filtering on pre-processed dataset. Thus, we explored the different kinds of Recommendation Models.

## REFERENCES

1. <https://archive.ics.uci.edu/ml/datasets/Anonymous+Microsoft+Web+Data>
2. <http://queirozf.com/entries/pandas-dataframe-plot-examples-with-matplotlib-pyplot>
3. <https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.svd.html>
4. <https://towardsdatascience.com/singular-value-decomposition-example-in-python-dab2507d85a0>
5. <https://towardsdatascience.com/how-to-build-a-simple-recommender-system-in-python-375093c3fb7d>
6. <https://medium.com/@wwwbbb8510/python-implementation-of-baseline-item-based-collaborative-filtering-2ba7c8960590>
7. <https://towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1>
8. <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>
9. [https://medium.com/@zhang\\_yang/python-code-examples-of-pca-v-s-svd-4e9861db0a71](https://medium.com/@zhang_yang/python-code-examples-of-pca-v-s-svd-4e9861db0a71)
10. <https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-the-fundamentals-dfee054f978e>
11. <https://github.com/daviddwlee84/MachineLearningPractice>