Predicting User Visit in Stochastic Web Surfing

Team 04

Himanshu Vinod Nirmal, Krishna Prasad Neupane, Sachin Mohan Sujir

1) Introduction

Web surfing is now common and used in every sector to access the information. Its extensive use is due to its easiness, flexibility, and ability to provide immediate information to the user. The problem of our project is to recommend the websites that a user may visit based on what other websites he or she visited. Recommending the websites that a target user may visit based on the website visits made by the other users of the cluster to which our target user belongs to.

To predict user visits of the websites we will be using unsupervised learning algorithms. The unsupervised algorithm clusters the users who visited most of the similar websites and then we predict the websites for the users based on their clusters. We will use K-means clustering as a core algorithm.

The target users are general users who use the website. Recommendation has become a trend today and users expect recommendations for faster access and search. An active user will get suggestions to visit other websites of similar content thus giving them more options to explore and gather more information. This recommendation will be based on the active user's previous search and the similarity with users who have similar searches. The active user will be suggested with a website based on other users(voting- based on other users who have visited that area) who belongs to the cluster or user nearest to the mean of the cluster.

2) Data Preprocessing

We used anonymous web data from www.microsoft.com available in the UCI KDD archive [1]. The data has different training and test set of users-website interaction. The dataset contains 32711 training data and 5000 data reserved for testing. The dataset has 294 attributes(Web Areas), which is called vroots in the dataset. We create a user website interaction matrix by representing 1 for interacted website else 0. The mean interaction of the user to the website is about 3 where the number of websites is 294. So, our matrix has a sparse representation.

The format of the dataset is ASCII-based sparse data format and each line starts with a letter representing the type of the line.

For example:

'A,1277,1,"NetShow for PowerPoint","/stream"'

Where:

'A' represents it is an attribute line,

'1277' is the attribute ID number of the website (called a Vroot),

'1' may be ignored,

'"NetShow for PowerPoint"' is the title of the Vroot,

'"/stream"' is the URL relative to "http://www.microsoft.com"

3) Algorithms:

- **User-based neighboring**

We have used User-based neighboring for the recommendation. This algorithm clusters similar users based on cosine similarity measures. We made 10 neighbors for each user and then find the

best item (in our case website) by the voting of those neighbors using a training dataset. We then again compute the cosine similarity measures between each user of test data and the mean value of each cluster. Based on their similarity value we predict the similarity probability of the respective clusters and predicted websites.

- **K-mean Clustering (Core algorithm)**

  K-means is our core algorithm which averages the vectorized points to calculate the centroid and place similar points close to that centroid based on the mean value. K-means re-calculates the centroid and repeats the process until no further change in the centroid occurs.

- **Mean shift Clustering**

  This algorithm is a nonparametric clustering algorithm that locates a maximum of a density function.

- **SVD Matrix Factorization**

  In this algorithm, we used singular value decomposition to factorize the interaction matrix of the user and website. Let the interaction matrix is 'R', website matrix 'V', and user matrix 'U' then the SVD matrix factorization is given below:

$$R = U\Sigma V$$

4) Evaluation Metrics

For the recommendation system, commonly used evaluation measures are prediction accuracy, rank accuracy, decision-support, etc. In this project, we have used rank accuracy with different TOP@N rankings.

5) Experimental Results

We have selected topN rank ranges from 1 to 10. This number can be selected differently based on user interest.

- **User-based neighboring**

We made 10 neighbors for each user and then find the best items based on number topN values. We then again compute the cosine similarity measures between each user of test data and then compare predicted websites with the actual websites. Here is the result:

Top@N	**Predicted Accuracy**
1	0.01
3	0.362
5	0.378
10	0.434

Table 1

```
Enter topN value=
10
Test accuracy using user-based neighboring method=
Backend TkAgg is interactive backend. Turning interactive mode on.
0.4344
```

Figure 1 topN=10

- **K-mean Clustering (Core algorithm)**

We have initially assigned the number of clusters as 7 to compute the performance of the algorithm but after doing fine-tuning i.e. using the elbow plot we got the number of clusters equal to 5. The results have been tabulated in Table 2.

Top@N	Predicted Accuracy
1	0.228
3	0.441
5	0.539
10	0.657

Table 2

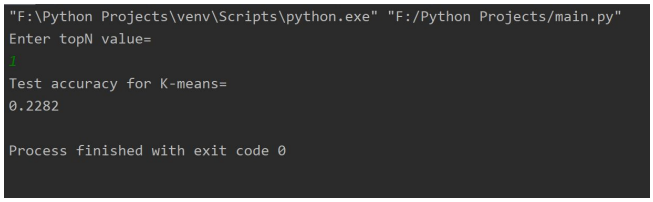

Figure 2 topN=1

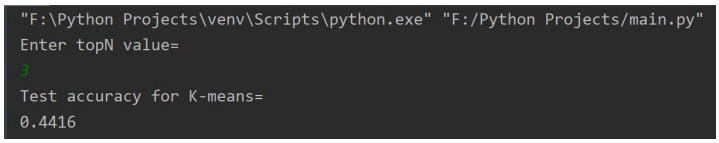

Figure 3 topN=3

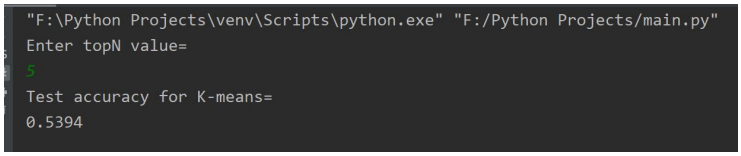

Figure 4 topN=5

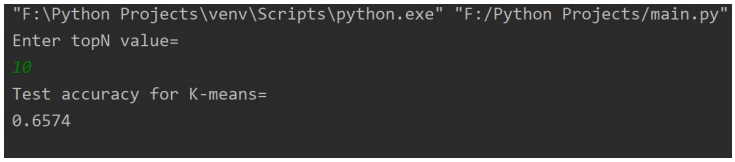

Figure 5 topN=10

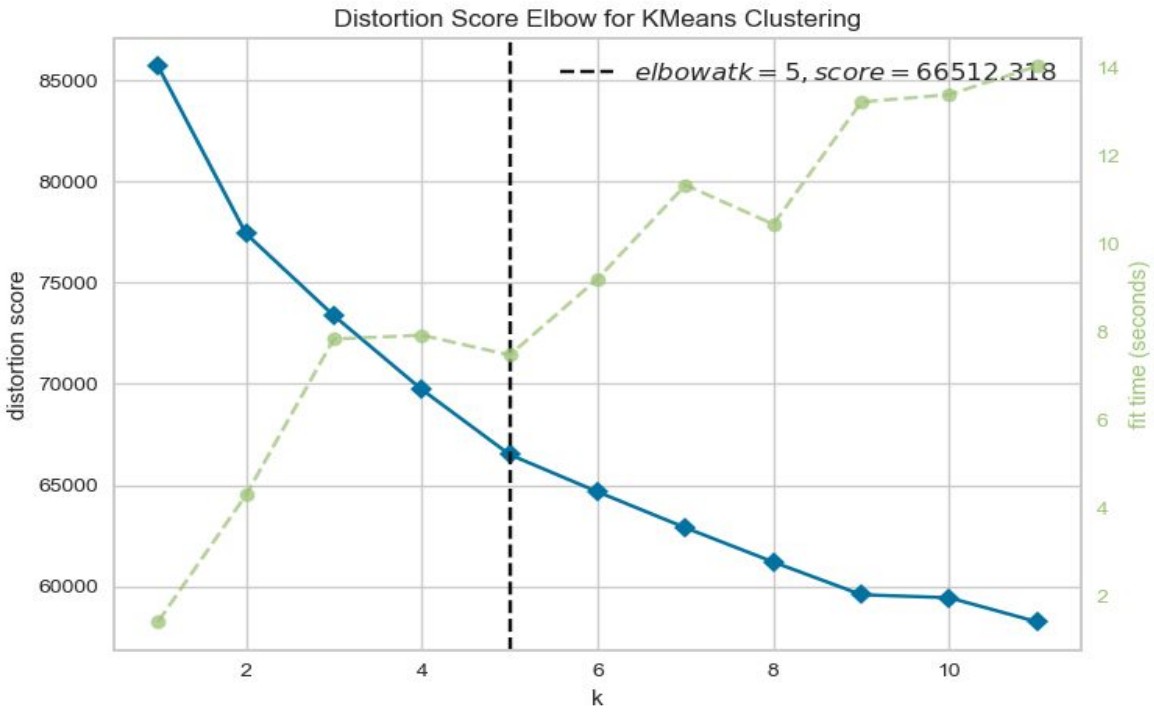

Figure 6: Elbow Curve

- **Mean shift Clustering**

Due to the memory problem while computation, we used a subset of the training data and found 3 clusters for this approach. Based on those three clusters we predicted the websites for the test users. The result is shown in the table below:

Top@N	**Predicted Accuracy**
1	0.02
3	0.43
5	0.55
10	0.63

Table 3

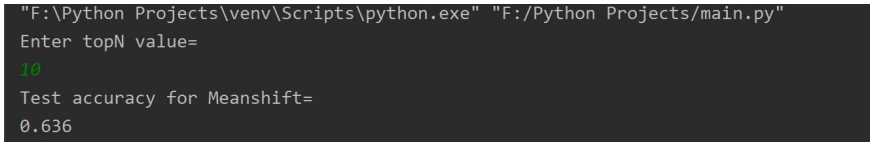

Figure 6 topN=10

- **SVD Matrix Factorization**

  For this experiment, we used only the test data because we have labels for them. This algorithm is assigned 50 latent parameters for users and websites. The result of this experiment is shown in the table below.

Top@N	**Predicted Accuracy**
1	0.240
3	0.246
5	0.249
10	0.265

Table 4

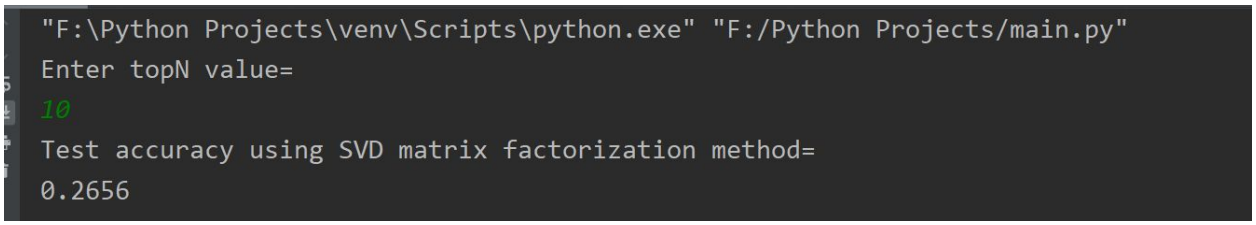

Figure 7 topN=10

Discussion:

In this section, we compare three algorithms that are trained with training data and tested with test data. The comparison is shown in the plot below.

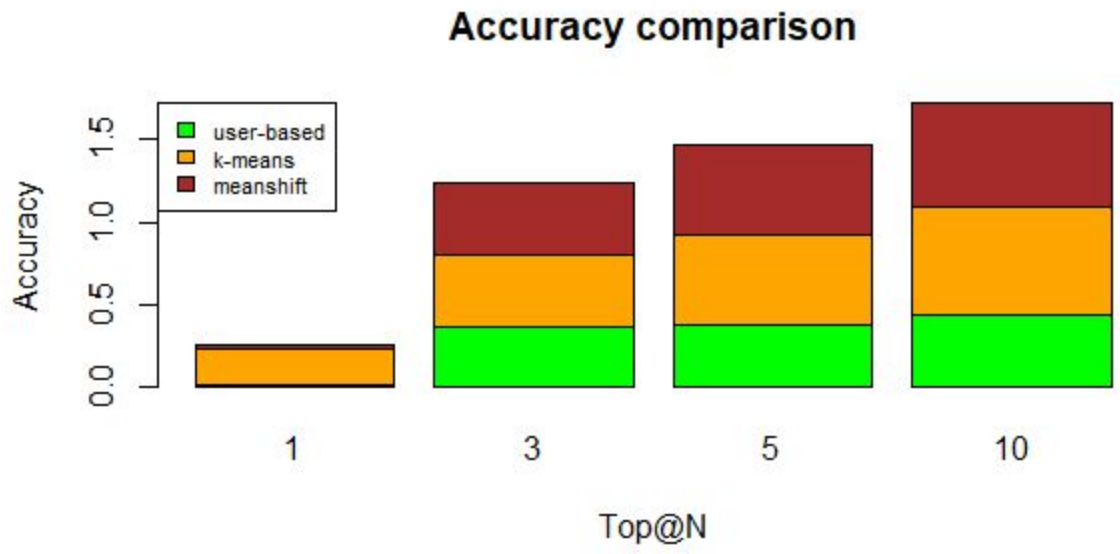

Figure 9: Performance Comparison of three algorithms.

From the figure, we can conclude that the K-means cluster is quite good because we computed clusters using fine-tuning.

Conclusion:

In this project, we tested different cluster-based algorithms for the recommendation system. Our project aims to predict the appropriate website for the user based on their previous interaction with the website. We achieved 65% accuracy with the core K-means algorithm clustering approach for TOP@10 rank websites where the number of websites is 249.

Currently, we have used training data only for the training and didn't perform any validation task by splitting training data. In the future, we can do this step to increase the performance of the test data.

References:

[1] Seth Hettich and SD Bay. The UCI KDD archive [http://kdd.ics.uci.edu]. Irvine, ca: University of California. Department of Information and Computer Science, 152, 1999.