Predicting User Visit in Stochastic Web Surfing

Team 04

Himanshu Vinod Nirmal, Krishna Prasad Neupane, Sachin Mohan Sujir

1) 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. The tabulated sample results are shown in Table 1.

User	Predicted Website	Probability
13452	1041	0.86
	1026	0.89
13440	1000	0.73
	1014	0.73
13434	1067	0.79
	1026	0.72

Table 1

2) K-means with K=7 (Random K)

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.

We have initially assigned the number of clusters as 7 to compute the performance of the algorithm. The results have been tabulated in Table 2. Here, predicted websites and probability are computed similar to the user-based neighbor strategy.

User	Predicted Website	Probability
13452	1018	0.30
	1034	0.51
	1026	0.66
13440	1008	0.06
	1004	0.02
	1018	0.03
13434	1034	0.40
	1018	0.41
	1008	0.71

Table 2

Fine-Tuning:

Our core algorithm is K-mean clustering so we did fine-tuning using the elbow curve method.

Elbow Curve Analysis:

Since K-means is an unsupervised technique, the performances are difficult to capture. Hence for performance comparison, we use the elbow curve method. The underlying idea is to get the optimal number of clusters suitable for the vector data we provide. So we run the k-means

algorithm for a range of ‘k’ values: 1 to 12. For each value, the distortion is mapped. As the number of clusters increases, the distortion score tends to get low as each data point will have it

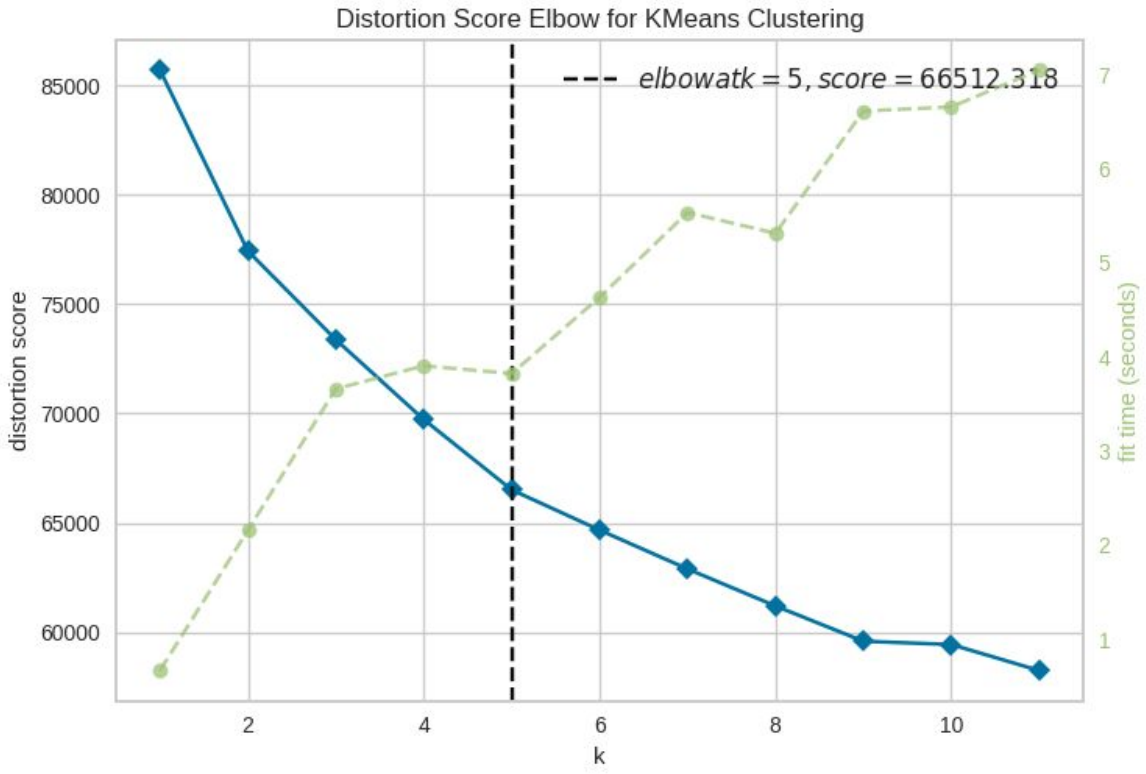

Figure 1

own cluster as the number keeps increasing. We choose the value of ‘k’ at the ‘elbow’ of the curve- the point where the distortion starts to decrease in a linear manner as shown in figure 1, where the elbow is at k=5. So we again train our algorithm with k as 5 clusters. The sample results are tabulated in Table 3.

User	Prediction	Probability
13452	1018	0.40
	1009	0.46
	1008	0.83
13440	1009	0.02
	1001	0.03
	1034	0.06
13434	1008	0.59
	1018	0.28
	1009	0.86

Table 3

Though there aren't significant changes seen after changing the number of clusters to the optimal number from the elbow curve, we can see a small increase in the probabilities predicted for the website. But for some data points, there are significant changes in the probabilities. Hence, we will have k=5, as the number of clusters for the algorithm.