**DATABASE MANAGEMENT SYSTEM**

**CSE-2004**

**REVIEW 3 REPORT**

**PROF. GOVINDA K**

|  |  |
| --- | --- |
| DIVESH AMAR LAHRANI | 17BCE2281 |
| ANUSHKA VINAY DWIVEDI | 17BCE2209 |
| NAMAN L UPADHYAY | 17BCE0980 |

**RECOMMENDATION SYSTEM FOR BLOGS**

**ACKNOWLEDGMENT**

We are thankful to our faculty GOVINDA K – SCOPE for giving us this opportunity to work on this project “RECOMMENDATION SYSTEM”. He has guided us throughout the project and we are highly obliged to receive his precious suggestions and advice.

We are grateful to his continuous mentorship that helped us tackle the difficulties and obstacles faced during the making of this project which eventually led to the completion of the project

**INDEX**

**1. ABSTRACT**

**2. INTRODUCTION**

**3. LITERATURE REVIEW**

**4. PROPOSED METHOD**

**5. RESULT**

**6. COMPARISION AND CONCLUSION**

**7. REFERENCES**

**ABSTRACT**

A straightforward, vital and well known approach to share your own understanding and thought on web is BLOG. In this day and age it is exceptionally troublesome and tedious to discover which blog completes a client need to peruse and survey due to a colossal measure of information and data on web. This venture demonstrates how proposal framework are utilized in online journals of the present world, which contains huge amounts of data on web. We utilize diverse systems in this venture to indicate how writes are prescribed on a specific website page effortlessly.In the course of recent years the web has encountered an exponential development in the utilization of weblogs or websites, sites containing diary style sections displayed backward sequential request. In this paper we give an examination of the kind of proposal methodology reasonable for this space. We acquaint measures with describe the blogosphere as far as blogger and point float and we show how these measures can be utilized to build a conceivable clarification for blogger conduct. We demonstrate that the blog space is described by bloggers moving much of the time from point to theme and that blogger movement nearly tracks occasions in reality. We at that point exhibit how label cloud data inside each group enables us to distinguish the most theme applicable and steady online journals in each bunch.

**INTRODUCTION**

**PROBLEM STATEMENT**

The objective behind the recommendation system is to provide user the data or item of his choice by just predicting. If we are able to do so the user will not have to search for the data or item which he requires from million of TBs of data available to him in modern world.

If we are able to develop an algorithm which recommends the similar data which the user is searching for, the user doesn’t have to go through the entire list of data or items.

Successful implementation of recommendation system can be used on almost every online shopping , blog , websites. E.g the largest e commerce website Amazon uses recommendation system to show user what they buy along with the current purchased product.

**WHY RECOMMENDATION SYSTEM**

The measure of information in our reality has been detonating, which settles on it troublesome for the chief to distinguish helpful data. Ongoing years have seen a huge development of the blogosphere. The span of the gathering of websites on the World Wide Web has been recently showing an exponential increment. As online journals turn out to be increasingly prominent, they draw in an ever increasing number of individuals to get included and contribute new and valuable data to blogosphere. Online journals are presently one of the fundamental way to spread thoughts and data all through the Web. They examine diverse patterns, thoughts and occasions.

With the end goal to get the most incentive out of their information, the test is to guarantee the correct data is getting to the correct representatives, on the grounds that the vast measure of data makes it troublesome for clients to know about it or even glance through it. The customized recommender frameworks are critical applications that can address this issue and propose things that suit the client's needs [1]. Normally, a recommender framework thinks about the client's profile to some reference attributes, and looks to foresee the rating that a client would provide for a thing they had not yet thought about [2]. The key component of a recommender framework is the client display that contains information about the individual inclinations, which decides his or her conduct in an unpredictable situation.

Now let’s come to the special class of algorithms which are tailor-made for solving the recommendation problem. There are typically two types of algorithms – Content Based and Collaborative Filtering. You should refer to our previous article to get a complete sense of how they work. I’ll give a short recap here.

**Content based algorithms:**

**Idea:** If you like an item then you will also like a “similar” item

Based on similarity of the items being recommended

It generally works well when its easy to determine the context/properties of each item. For instance when we are recommending the same kind of item like a movie recommendation or song recommendation.

**Collaborative filtering algorithms:**

**Idea:** If a person A likes item 1, 2, 3 and B like 2,3,4 then they have similar interests and A should like item 4 and B should like item 1.

This algorithm is entirely based on the past behavior and not on the context. This makes it one of the most commonly used algorithm as it is not dependent on any additional information.

For instance: product recommendations by e-commerce player like Amazon and merchant recommendations by banks like American Express.

**LITERATURE REVIEW**

The imperfections of the three fundamental conventional likeness calculations. For the arrangement of those issues, we present another similitude calculation called Cosine. This calculation considers the quantity of assets which the two clients both assess, the nearness of the assets score which is given to a similar asset by the two clients and the edge between the relating uservectors. By thinking about every one of these elements, Cosine calculation[1] viably beats every one of the deformities. The trial results demonstrate that the precision of the Cosine calculation is superior to that of the Pearson.

All in all, recommender frameworks may fill two distinct needs. On one hand, they can be utilized to animate clients into accomplishing something, for example, purchasing a particular book or viewing a particular motion picture.[2]Then again, recommender frameworks can likewise be viewed as instruments for managing data over-burden, as these frameworks expect to choose the most fascinating things from a bigger set. Along these lines, recommender frameworks inquire about is likewise firmly established in the fields of data recovery and data separating. In these zones, be that as it may, the emphasis lies mostly on the issue of separating among significant and unessential records. A large number of the strategies created in these regions abuse data got from the documents'contents to rank them.

Information based suggestion frameworks key thought is to utilize principles to prescribe diverse things to various clients[3]. Maybe a few standards has struggle or incorporate others rules, so it regards have diverse weights on these tenets to organize them. A standout amongst the most dependable rating and weighting strategy is to communicate with clients to discover which rule is more imperative. Presently the significance of client cooperation is amplified.

With the improvement of WoT framework, proposal component is exceptionally essential for clients[4]. This paper initially introduces the exploration on conventional suggestion calculations: content-based proposal, communitarian separating suggestion and the cross breed suggestion. At that point it proposes a thing based suggestion calculation in WoT framework, which consolidates both the substance based proposal and community oriented separating proposal to enhance the proposal quality.

Music proposals are performed by utilizing client and thing likenesses. The objective was to give a methodology where we can use a portion of the Information recovery strategies and the direct logarithmic way to deal with understand enormous and meager networks inorder to remove right and important data[5]. Numerous algorithmic methodologies result in an exponential increment in scientific counts. A methodology like SVD can profit by decreasing high dimensionality space into low measurements.

In the framework, client based shared sifting calculation is received. All the more decisively, a watchword list is prepared to get clients' inclinations. The dynamic client gives his/her inclinations by entering the catchphrases from the watchword applicant list, and the inclinations of past clients are separated from their surveys for various items[6]. These watchwords from surveys are assembled by item and after that they are characterized in two sets one with positive significance audit and another with negative importance audit.

**PROPOSED METHOD**

**1. COSINE SIMILARITY**

We use the following softwares and libraries for this project.

1. Colab – Google’s implementation of Python Jupyter Notebook to work with various machine learning algorithms and data analytics.
2. Pandas-An important library in python to do basic manipulation on files. To read the data from the database and store them in data structures called data frame.
3. NumPy- NumPy is Fundamental library in python for array manupulations.
4. Sklearn – Sklearn is an important library in python to perform basic machine learning algorithms.

Firstly, we install all the datasets using pandas library.

import pandas as pd

import numpy as np

from sklearn.metrics import pairwise\_distances

pairwise distances is the library in sklearn which has various similarity metrics like cosine, jaccard, Euclidean, pearson’s correlation,etc.

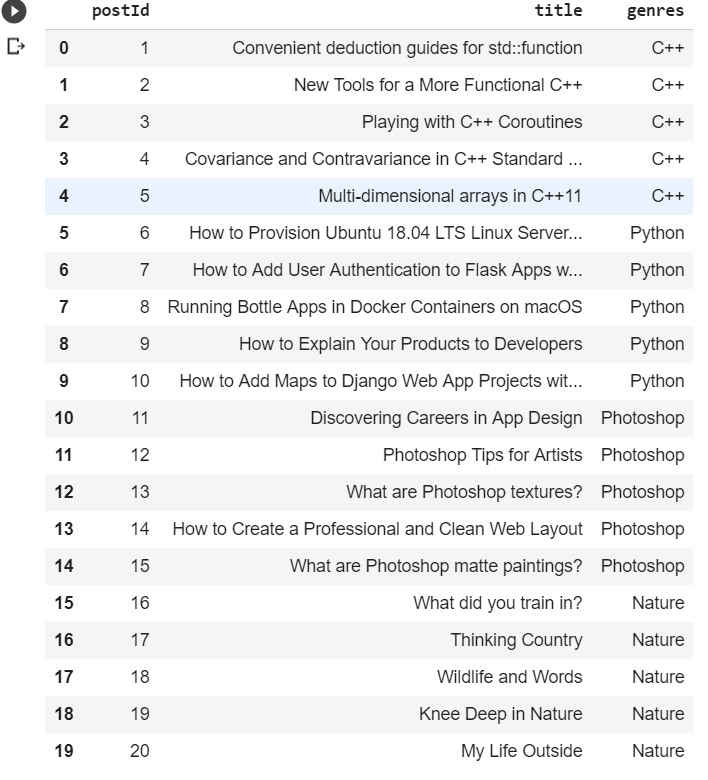
The two datasets are ratings.csv and posts.csv ratings.csv consists of the ratings posted by various users on various columns. Posts.csv consist of the information regarding various posts. The datasets are stored in .csv format. For this we use the pandas read\_csv method.

posts=pd.read\_csv('posts.csv')

posts

The data from the file is stored in the database in the dataframe called posts

The output can be seen in the following figure.



Similarly for the ratings dataset.

ratings=pd.read\_csv('Ratings.csv')

ratings



We use a different format of representation of the ratings dataframe called pivot table.

A pivot table is a another datastructure in which all the elements of two elements are compared with each others. All the elements of all the first entity is compared with all the elements of the second entity. The python code for the same is:

ratings.matrix=ratings.pivot\_table(index=['postid'],columns=['userid'],values='ratings').reset\_index(drop=True)

Here the name of the pivot table is ratings.matrix and index is changed from default of id to postid and columns are userid and rows are the ratings.

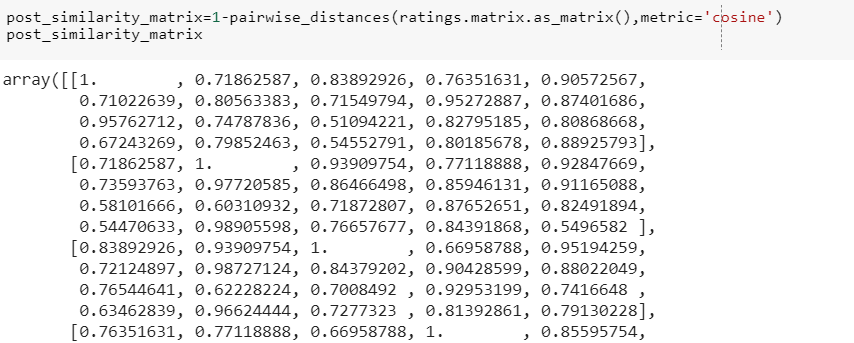


From the pivot table it can be seen that user with id 1 has given a rating 5 on post id 0 and a rating of 4 on post id.

Now we apply the cosine similarity algorithm on this pivot table.

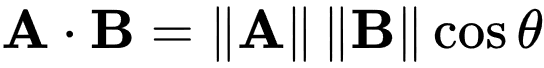
Here each post is a row. This row is converted into a n-dimensional matrix. In this case n is 5.

In this way we have have created 20 different 5-dimensional vectors. We apply cosine similarity between them. Cosine similarity calculates cosine of the angle between them. Lesser the angle more the similarity and more the cosine value.

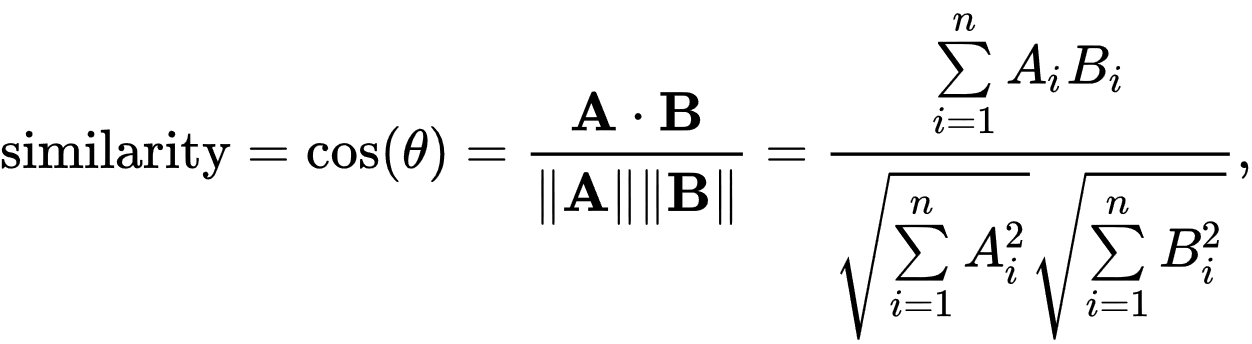


Here we can see the output is an array. Each element of this represents this extent of cosine similarity between the two elements. For example the [1,2] element represents the similarity index between the element with id=1 when compared with element with id=2;

The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:



Given two vectors of attributes, *A* and *B*, the cosine similarity, cos(θ), is represented using a dot product and magnitude as



There is a small problem with this array. The diagonal elements of this array are 1 because all the diagonal elements are of the form [i,i] so each similarity index in each of this case is 1 as the angle is 0. This raises an anamoly. So we can resolve this by replacing all the diagonal elements with 0.

np.fill\_diagonal(post\_similarity\_matrix,0)

post\_similarity=pd.DataFrame(post\_similarity\_matrix)

We store this matrix in a new DataFrame so that we can carry out our normal data manupulations.

Now we create a new function, which takes the post id as an argument and sends the best four similar movies according to the similarity index generated by the cosine similarity.

def recommend\_similar\_post(postid):

try:

posts['similarity']=post\_similarity.iloc[postid -1]

top\_n=posts.sort\_values(["similarity"],ascending=False)[0:4]

print("Similar Posts to :")

return top\_n

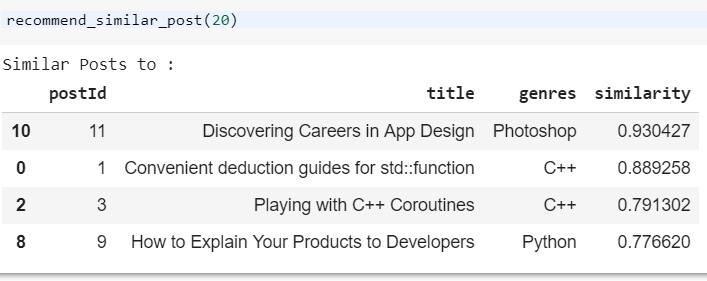
except:

print('Sorry movie not in database')

This function adds an additional column called similarity to the posts dataframe the column added is postid-1 numbered.

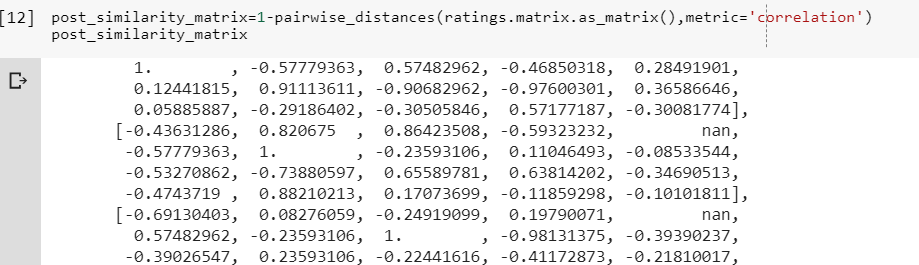
Then we sort the values in this column in descending order because we want to show the post which is having highest similarity first.

A sample input for the same is:



**2. Pearson’s correlation**

The steps upto making the pivot table are the same. After that we just have to give the metric as correlation instead of cosine.

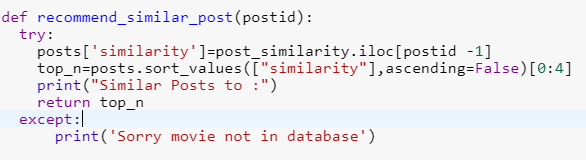


The original formula for correlation, developed by Pearson himself, uses raw data and the means of two variables, X and Y:



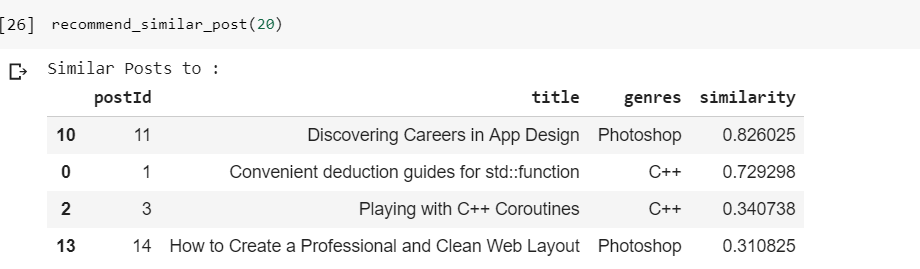
Here the similarity is calculated as the ratio of the covariance and standard deviation between the two entities. The above figure shows the similarity index for all the pairs.

Next we create a function just like we did in cosine similarity.



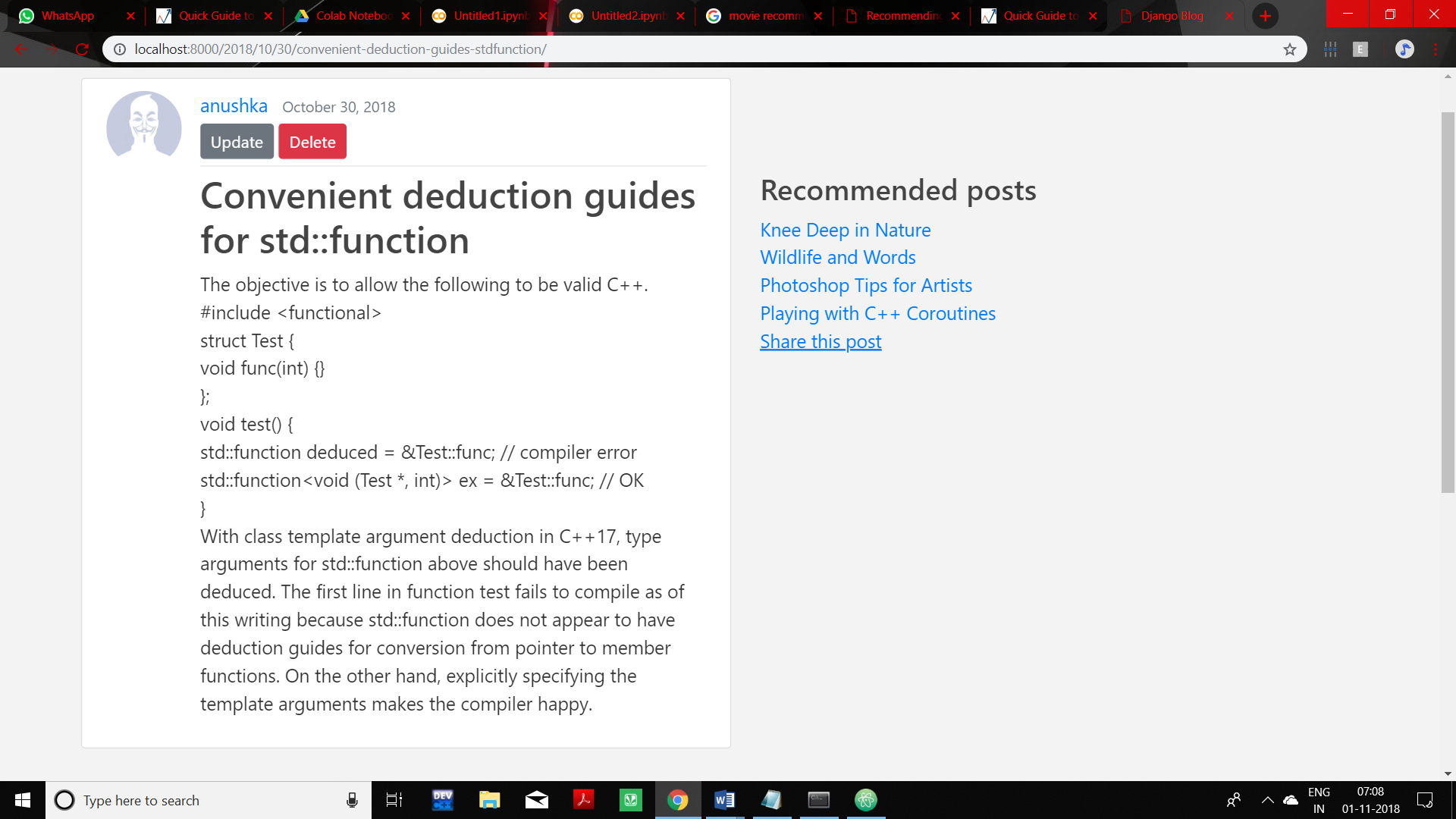
We add the column similarity to the posts dataframe. We sort the values in the similarity column and display them in descenfing order to show the posts with highest similaarity first

One of the result is shown:



**RESULT**

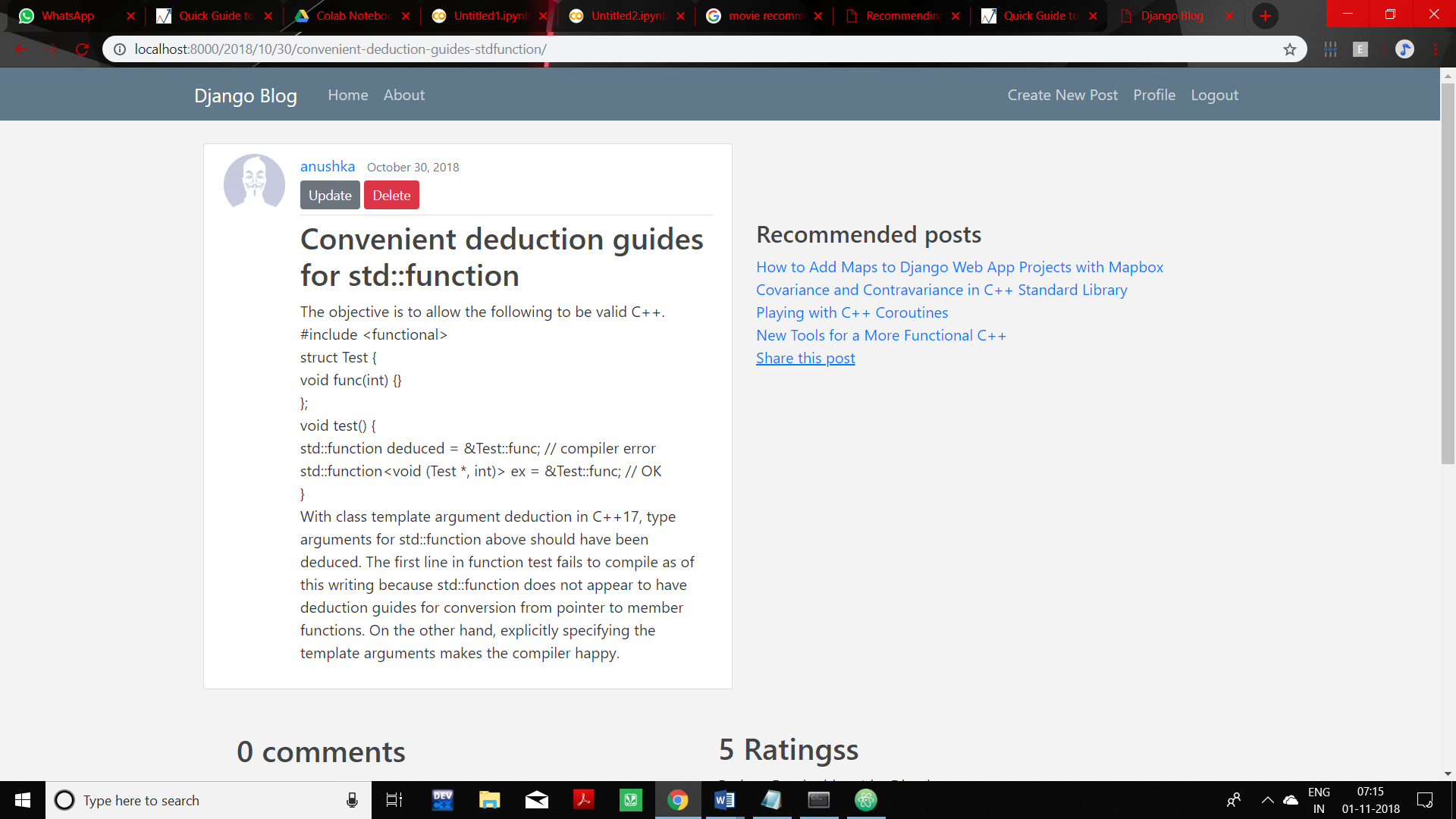
**Pearson correlation**



As it can be seen in the above figure the posts are recommended for the post ‘Convention deduction guides for std : function’ according to pearson coorelation.

Here the posts recommended are not of same category as that of the current post. However, most of them have similar ratings like the current post.

In the cosine similarity, the posts recommended are shown in the figure below:



Here the posts recommended are of the same category as the current post. Also the posts recommended are of similar ratings.

**Comparative graph study**

To compare the two methods we choose the metric accuracy

To compare the two methods we choose the metric time .

**CONCLUSION**

From the above graphs it is clear that cosine similarity has much higher accuracy when compared to Pearson Correlation but at the same time cosine similarity consumes more time. For small scale websites cosine similarity will work fine as time is not the primary resource. Whereas in case of large scale websites time as well as accuracy both are primary resource. In such cases a new hybrid algorithm can be used. The future applications of such comparison can be used in almost every field such as movie website and online shopping.

This helps the website to be much interactive and user friendly.

**REFERENCES**

[1] Huangfu Dapeng, Lin Qianhui, Zhou Jingmin, An Improved Similarity Algorithm for Personalized Recommendation.

[2] Mohammad Hamidi Esfahani, Farid Khosh Alhan ,Hybrid Recommendation System Based On C-Means Clustering Method

[3] Zhong Lian1，Chen Hui Research on Recommendation Algorithms in Web of Things System

[4] M.Sunitha , Dr.T.Adilakshmi , Mobile Based Music Recommendation System

[5] Ms. Vinaya B. Savadekar Mr. Manoj E. Patil, Improved Recommendation System with Review Analysis

[6] Ms. Shakila Shaikh , Dr. Sheetal Rathi,Asst Prof. Prachi Janrao ,Recommendation system in E-commerce websites: A Graph Based Approached

[7] Yana Aditia Gerhana , Aldy Rialdy Atmadja , Wildan Budiawan Zulfikar , Nurida Ashanti , The Implementation of K-Nearest Neighbor Algorithm in Case-Based Reasoning Model for Forming Automatic Answer Identity and Searching Answer Similarity of Algorithm Case

[8] Kunal Shah , Akshaykumar Salunke, Saurabh Dongare , Kisandas Antala Recommender Systems: An overview of different approaches to recommendations

[9] Ms. Nishigandha Karbhari ,Prof. Asmita Deshmukh , Dr. Vinayak D. Shinde ,Recommendation System using Content Filtering

[10] Maake Benard Magara , Sunday O. Ojo , Tranos Zuva , A Comparative Analysis of Text Similarity Measures and Algorithms in Research Paper Recommender Systems.