LOVELY PROFESSIONAL UNIVERSITY

INT247

CA-2

ON

TEACHING RECOMMENDATION SYSTEM

Under the Guidance of-

Upinder Kaur

Submitted by-
Belal Azam (11913306)

Introduction

You must have seen the song recommendation systems, movie recommendation systems and other systems but you must not have seen the teacher recommendation system. As students we can understand the other students' feelings. Many times a student fails the subject because he cannot understand the teacher or the teacher cannot understand the student. Some teachers have different techniques and may think that technique is not suitable for a certain child and he may not understand the topic which will result in the hating for the subject and he will fail in that subject and people will blame him for the failure.

So to overcome this we came up with the idea called teacher recommendation system. this system will give you the type of teacher that you like for example if you like teacher 'A' then this system will provide you a teacher who is " 'A'+- delta " which is basically a teacher similar to 'A' this system will judge on various aspects like "voice of the teacher" ,"teacher’s language", "teaching speed" ,"teacher’s topic level", "teacher’s frankness", etc and based on this aspects this system will give you the teacher who is best for you.

Note: we are not blaming any teacher for the student failure, try to understand with this logic if i don't like a certain movie then it does not mean director of movie is bad rather my interest was not there in that type of movie and may be i can like another movie which will be not liked by others.

.

Design and Architecture

The design we used in our project is used by many morden big company like flipkart , amazon , etc basically product based company , this design is called Collaborative filtering, where we have data of many users and and we filter the similarity in the data and try to come across the best possible answer. basically its human behavior . We group similar types of people together and then we group their ideas together and try to find similarities between them. Just like if I like a teacher A B C and one person likes teacher B,C and he doesn't know about teacher A . Then we can recommend him about teacher A. There are high chances he is going to like the teacher.

the approach here we have used is cosine distance formula for finding the minimum distance between two objects , first we have tried with the euler distance formula but we were getting issues with that . for the two problems euler's formula was giving us same response so to improve we replaced them with cosine distance formula

Before Survey

During the initial design of our project, we as a group decided to take all the possible parameters that an online teacher can have such as -

- Language used in the lectures
- Pace of the speech
- No. of videos available from a single teacher
- Length of the videos
- Frankness
- etc

So we gave equal importance to all these parameters. Along with these, Student input such as the sentiment of his/her for a particular teacher and his/her ability to continue with the same teacher and for how long. We gave these features the same importance as given to teachers' features.

Since our Machine Learning Model takes input as a value from 0.0 to 10.0 which is the overall rating a student gave to this teacher, a Hashing function is required to compile all the ratings from these selected features.

We gave equal weightage to all the features which basically means taking the average of all the values. For eg, if a student has given ratings to a teacher as such -

- Language - 10.0
- Pace of the Lecture - 4.0
- Multiple Videos available - 8.0
- Length of the videos - 7.0
- Frankness - 10.0
- Assignment - 5.0

Then the average will be the summation of all these values and divide by 6 i.e -

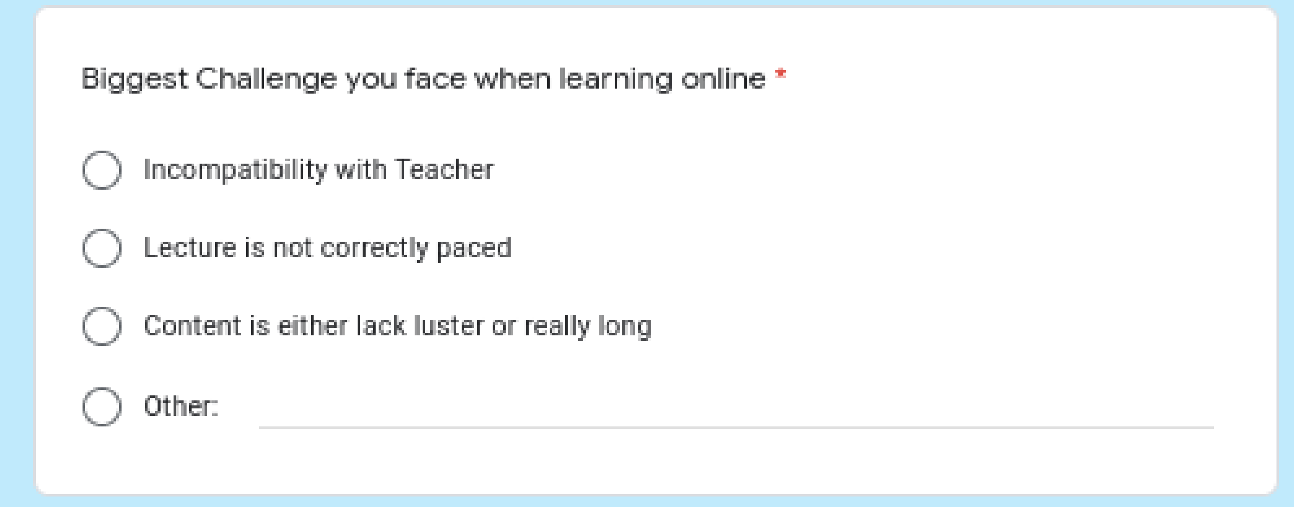
(10+4+8+7+10+5) / 6 = 44 / 6 = **7.33**

The model performed acceptably in suggesting the student's next teacher to learn from. But by extensive and rigorous testing, we discovered that the accuracy was not enough to make this a viable product. Hence we decided to survey actual real students who are our product’s target to fine tune our model’s accuracy and prediction.

The next section describes the survey and observations we got from it.

Survey Frame

Q1-



Biggest Challenge you face when learning online *

○ Incompatibility with Teacher

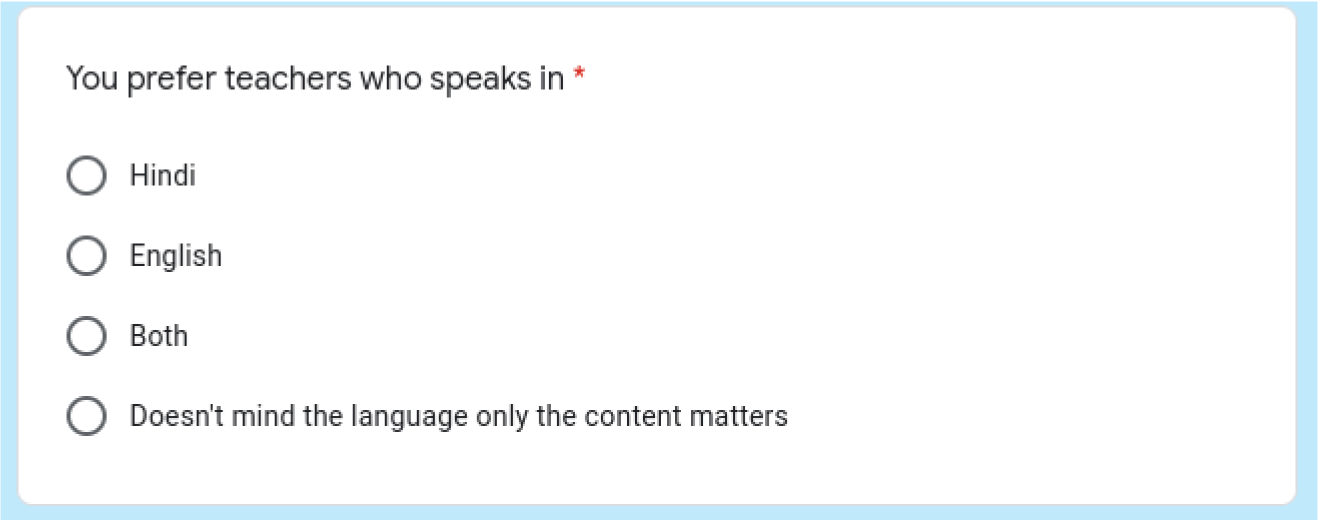
○ Lecture is not correctly paced

○ Content is either lack luster or really long

○ Other: ______

Motive - We chose this question to initialize our survey with the most important question since most people never complete the survey. We need crucial inferences from those responses. This question will define our most dominating factor in our recommendation system since this is the core problem we are trying to solve.

Q2-



You prefer teachers who speaks in *

○ Hindi

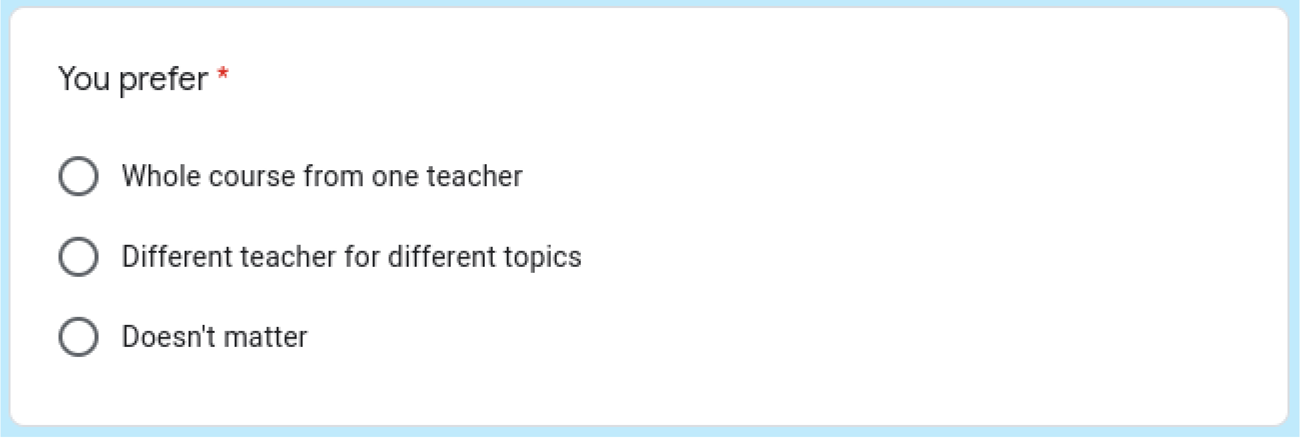
○ English

○ Both

○ Doesn't mind the language only the content matters

Motive - Since there are many students who get stuck in the language loop. We want to streamline this issue in our recommendation system. Based on the majority responses for this question, we will decide whether language can be a factor for recommending lectures.

Q3-



You prefer *

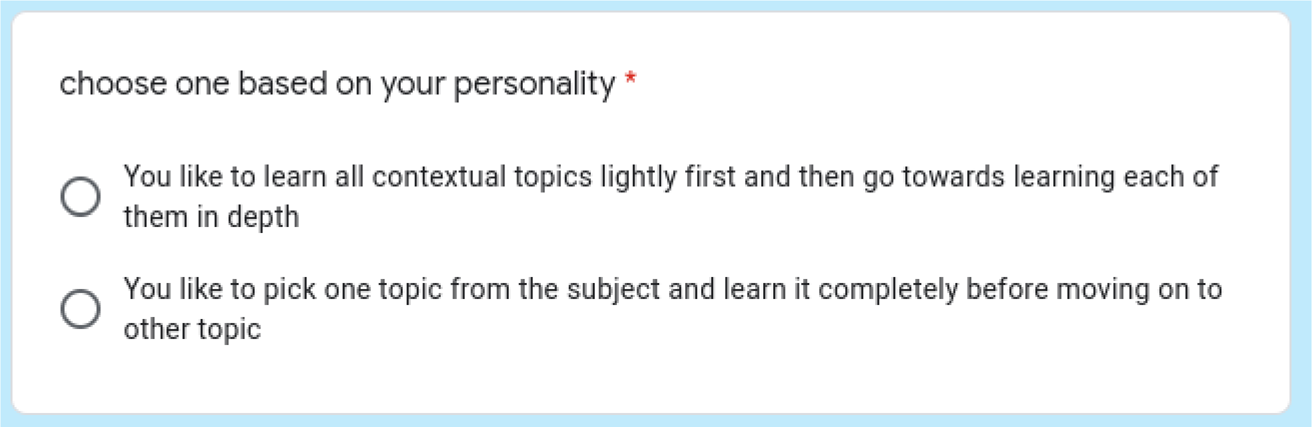
○ Whole course from one teacher

○ Different teacher for different topics

○ Doesn't matter

Motive - This is really important for our recommendation system. This basically defines our machine learning model's initial search space. If the majority of people prefer Whole course from the same teacher then our model must look for next lectures from the same teacher if not found then go for different teachers based on different parameters.

Q4-



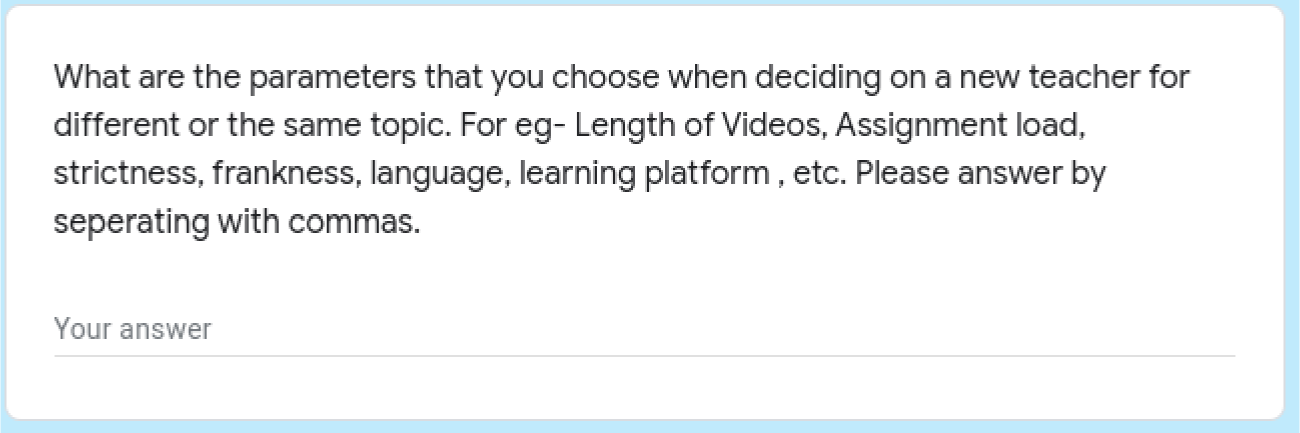
choose one based on your personality *

○ You like to learn all contextual topics lightly first and then go towards learning each of them in depth

○ You like to pick one topic from the subject and learn it completely before moving on to other topic

Motive - This is also very important for our machine learning model because most teachers divide complex topics into parts of videos such as Topic part 1, topic part 2, etc. So based on the majority vote. If the vote goes in favor of in-depth, we will put an inductive bias in our machine learning model so as to prefer the continued parts of the last learned lecture and if it favors the other, we will make sure our immediate next suggestion will not be the continued part of the previous lecture.

Q5-



What are the parameters that you choose when deciding on a new teacher for different or the same topic. For eg- Length of Videos, Assignment load, strictness, frankness, language, learning platform , etc. Please answer by seperating with commas.

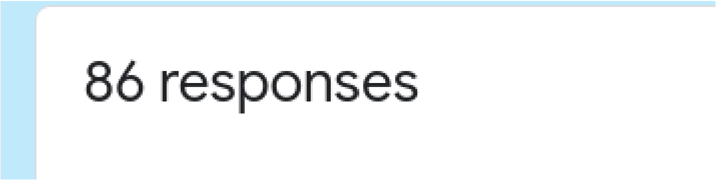
Your answer

Motive - Now this is our one of the most important and finalizing parameter. From this we can deduce the next most important features that our algorithm should work upon. Since our group's mindset is limited in terms of designing the recommendation system. We wanted new and fresh inputs from users that we are targeting since their choice of parameters will only make this system more accurate.

Based on the majority input from this question, we will try to create more features that are feasible and can lead to better recommendations.

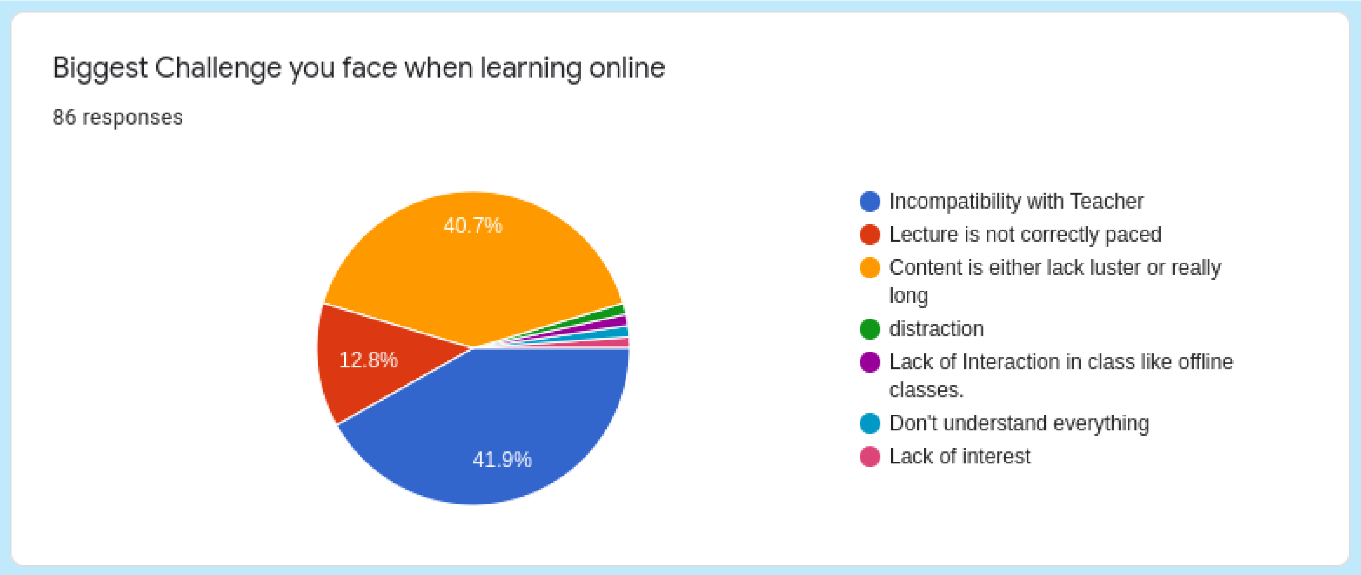
Survey Responses

Responses so far-



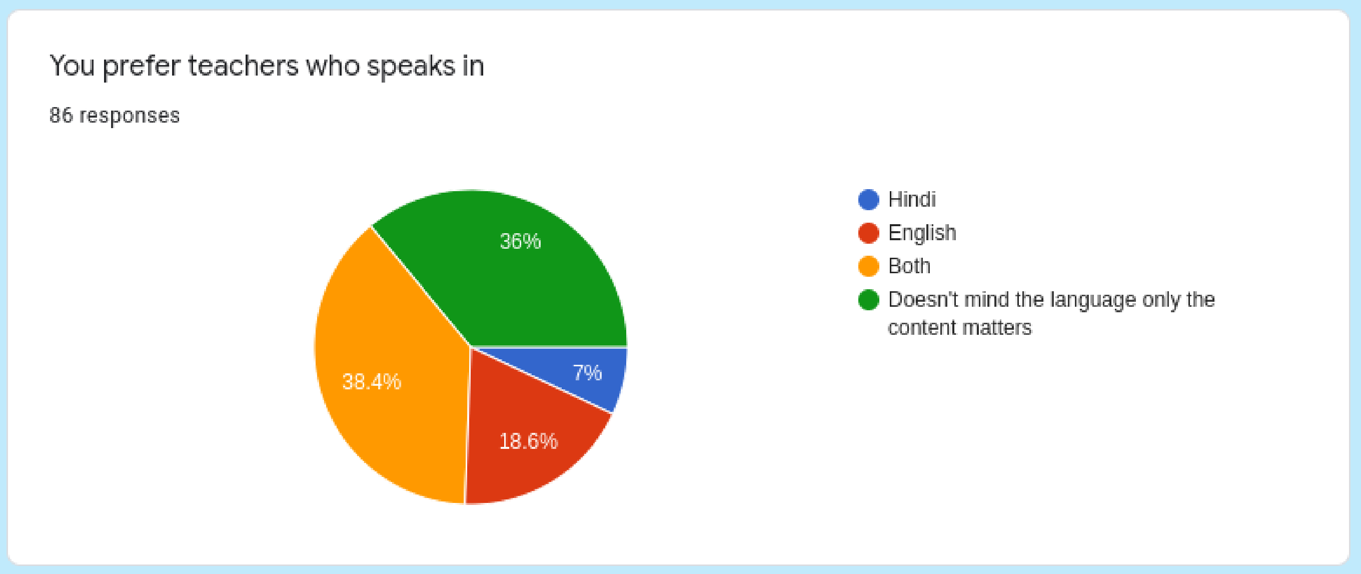
86 responses

Q1-



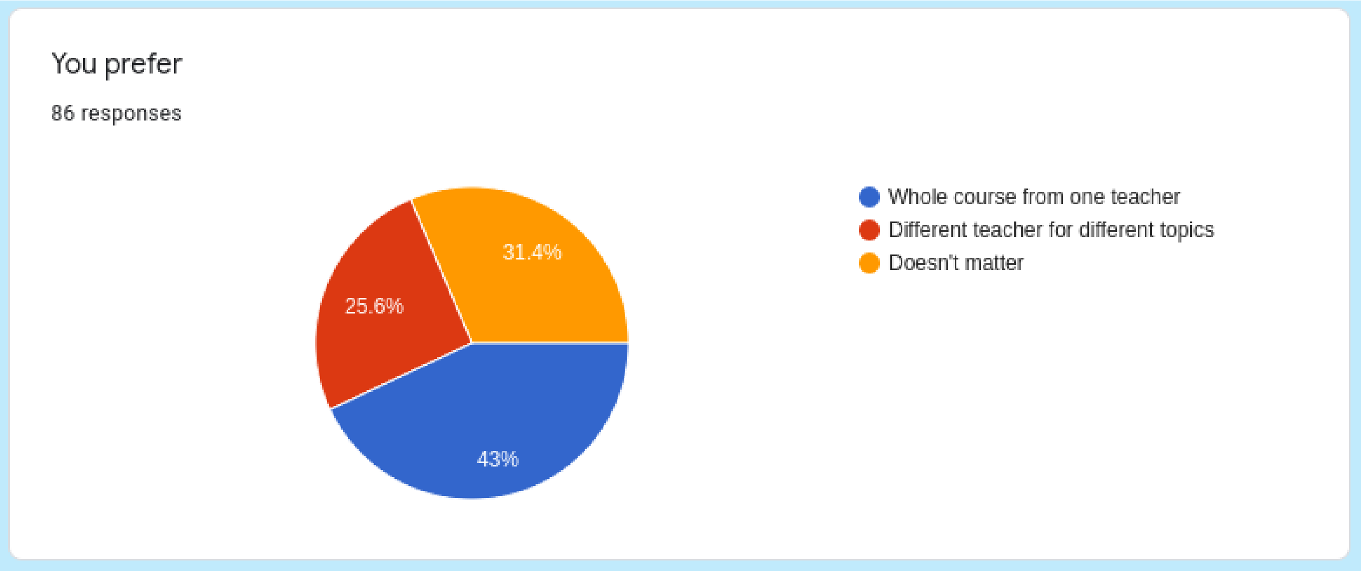
Inference- From this, it is clear that most of the participants are having issues with compatibility with the teacher and about the content, So we will use this to improve our model.

Q2-



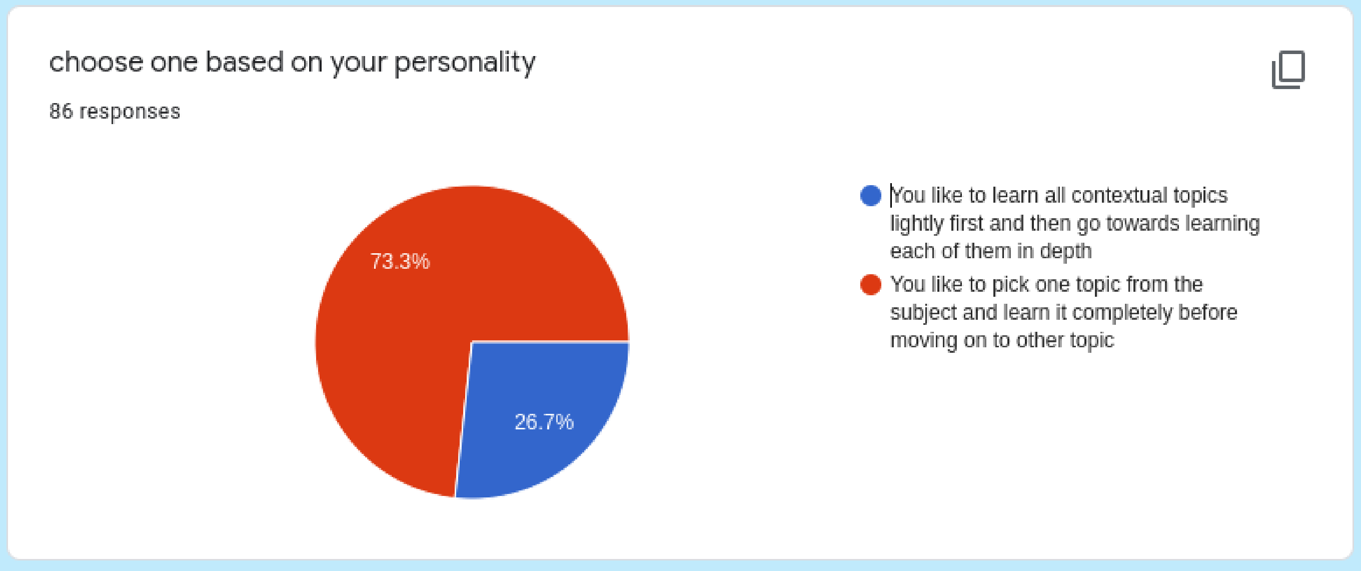
Inference- Most participants don't mind language. So we will reduce the dependence of language on our recommendation model.

Q3-



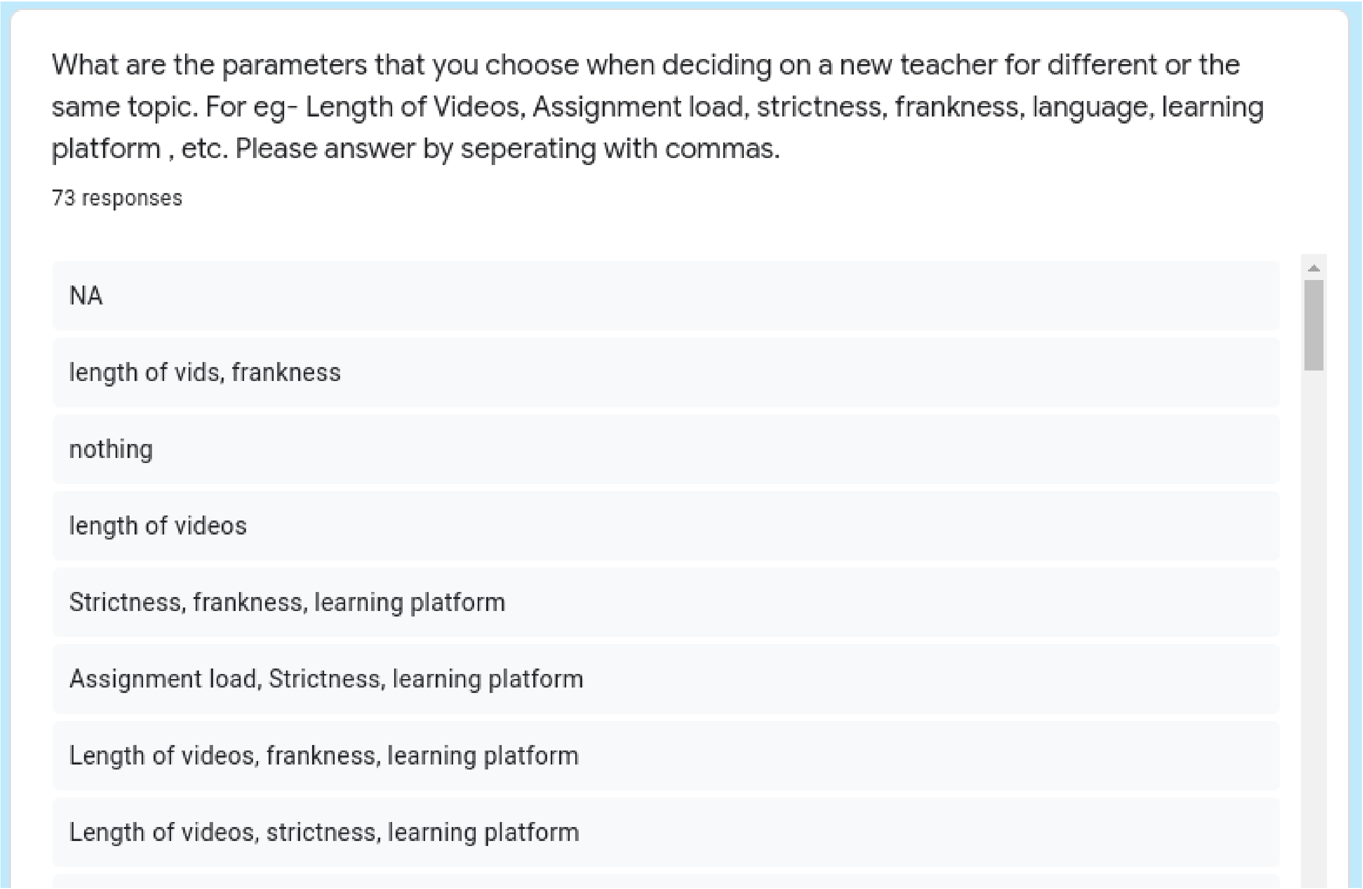
Inference- Since the responses are divided evenly we will try to recommend the next teacher without any bias for these factors.

Q4-



Inference- Most participants prefer to pick one topic and learn it fully then moving on to the next topic. We will use this to make our algorithm biased towards next parts of same topic videos.

Q5-



What are the parameters that you choose when deciding on a new teacher for different or the same topic. For eg- Length of Videos, Assignment load, strictness, frankness, language, learning platform , etc. Please answer by seperating with commas.

73 responses

NA

length of vids, frankness

nothing

length of videos

Strictness, frankness, learning platform

Assignment load, Strictness, learning platform

Length of videos, frankness, learning platform

Length of videos, strictness, learning platform

Inference- From most of the responses from this form, we understood that most common quantitative factors for the preference of the next lecture are-

1- length of videos
2- frankness
3- learning platform

We will work towards turning these into different feature sets so as to incorporate this in our machine learning model.

After Survey

From the survey, We discovered that the students choose their next teacher mostly based on the sentiment towards that particular teacher and after that, the length of the videos matter, then students prefer the pace of the lecture given. From then it was evident from the survey responses that the next most valuable feature to choose the next teacher is assignment followed by frankness and then comes last the no. of videos available from the teacher.

So we changed the weight given to these features which was initially set to 1 to all. Now we will give appropriate weight to these features based on the importance of features we understood through the survey responses.

The weightage has been given like this -

- Language - x1
- Pace of the Lecture - x6
- Multiple Videos available - x2
- Length of the videos - x10
- Frankness - x3
- Assignment - x4

So now for the same previous example how the values will change by multiplying -

- Language - 10.0 x 1 = 10.0
- Pace of the Lecture - 4.0 x 6 = 24.0
- Multiple Videos available - 8.0 x 2 = 16.0
- Length of the videos - 7.0 x 10 = 70.0
- Frankness - 10.0 x 3 = 30.0
- Assignment - 5.0 x 4 = 20.0

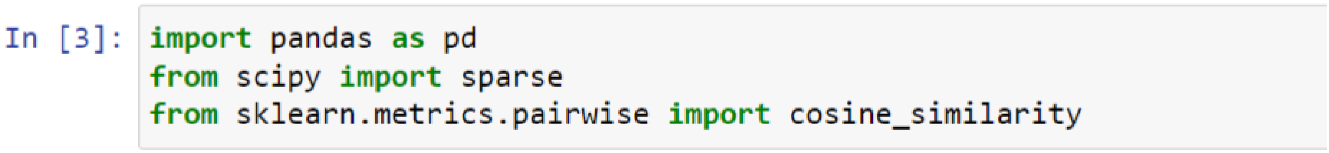
So now we can find the hashed value like this-

$$(10+24+16+70+30+20) / (1+6+2+10+3+4) = 170 / 26 = \mathbf{6.53}$$

After modifying the weightage, we observed from testing that the predictions were a lot more accurate and the model is ready to be converted into the final product. So we can say that the survey which we did in the initial phase of the project helped a lot in finalizing the actual math which works behind the algorithm.

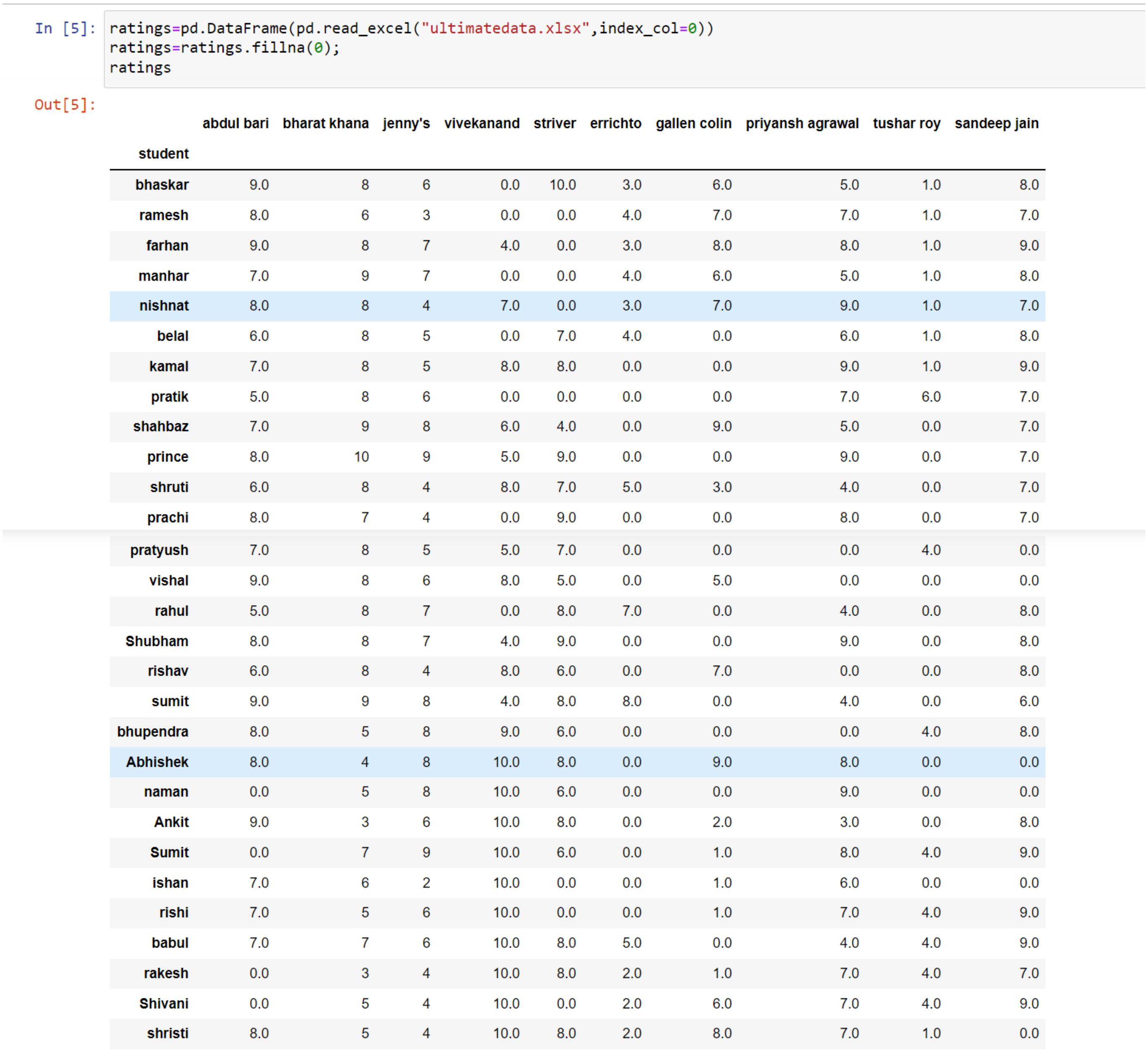
Project Progress

These are some screenshot from our working model:



```
In [3]: import pandas as pd
        from scipy import sparse
        from sklearn.metrics.pairwise import cosine_similarity
```

In this we are importing pandas packages from python libraries.

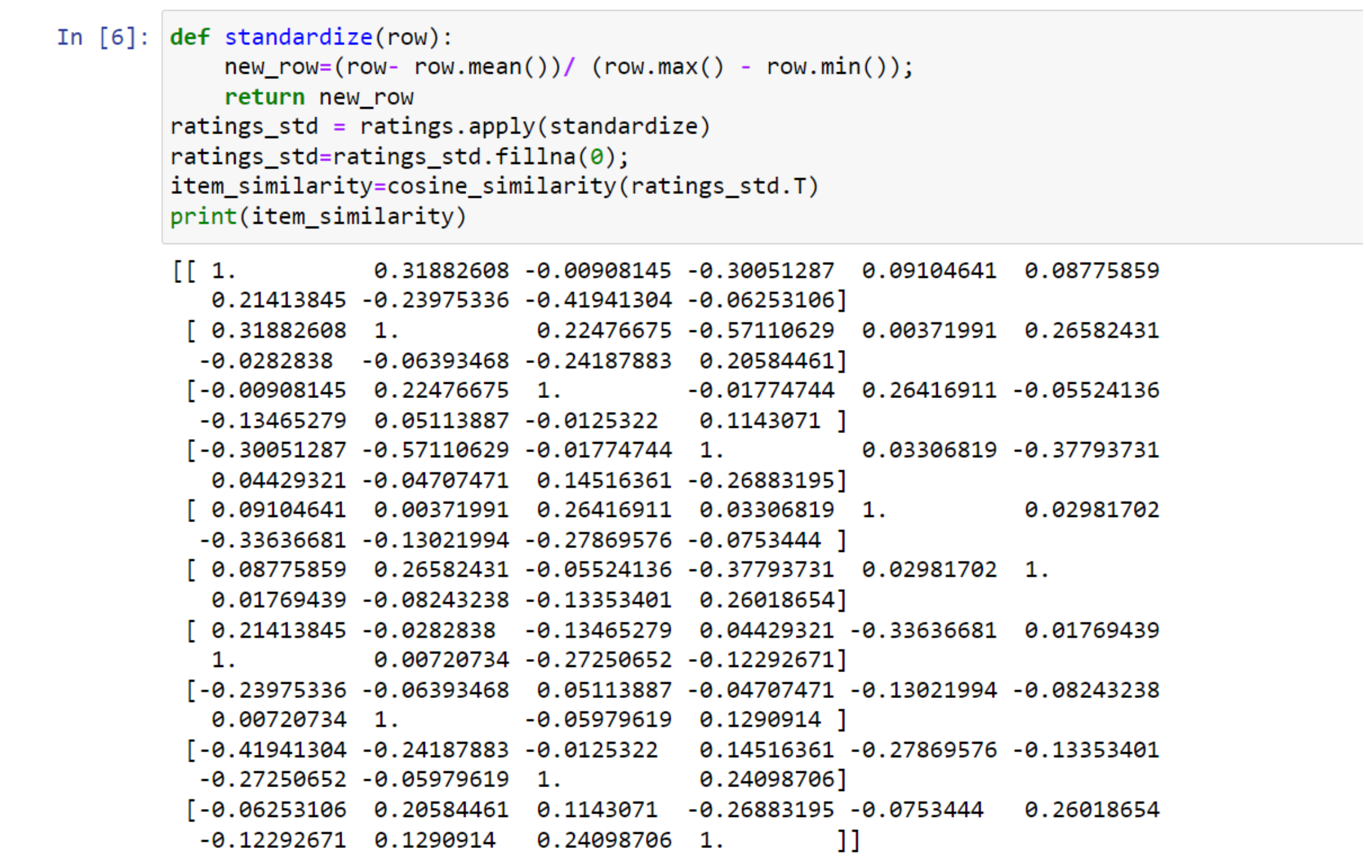


```
In [5]: ratings=pd.DataFrame(pd.read_excel("ultimatedata.xlsx",index_col=0))
        ratings=ratings.fillna(0);
        ratings
```

Out[5]:

	abdul bari	bharat khana	jenny's	vivekanand	striver	errichto	gallen colin	priyansh agrawal	tushar roy	sandeep jain
student										
bhaskar	9.0	8	6	0.0	10.0	3.0	6.0	5.0	1.0	8.0
ramesh	8.0	6	3	0.0	0.0	4.0	7.0	7.0	1.0	7.0
farhan	9.0	8	7	4.0	0.0	3.0	8.0	8.0	1.0	9.0
manhar	7.0	9	7	0.0	0.0	4.0	6.0	5.0	1.0	8.0
nishnat	8.0	8	4	7.0	0.0	3.0	7.0	9.0	1.0	7.0
belal	6.0	8	5	0.0	7.0	4.0	0.0	6.0	1.0	8.0
kamal	7.0	8	5	8.0	8.0	0.0	0.0	9.0	1.0	9.0
pratik	5.0	8	6	0.0	0.0	0.0	0.0	7.0	6.0	7.0
shahbaz	7.0	9	8	6.0	4.0	0.0	9.0	5.0	0.0	7.0
prince	8.0	10	9	5.0	9.0	0.0	0.0	9.0	0.0	7.0
shruti	6.0	8	4	8.0	7.0	5.0	3.0	4.0	0.0	7.0
prachi	8.0	7	4	0.0	9.0	0.0	0.0	8.0	0.0	7.0
pratyush	7.0	8	5	5.0	7.0	0.0	0.0	0.0	4.0	0.0
vishal	9.0	8	6	8.0	5.0	0.0	5.0	0.0	0.0	0.0
rahul	5.0	8	7	0.0	8.0	7.0	0.0	4.0	0.0	8.0
Shubham	8.0	8	7	4.0	9.0	0.0	0.0	9.0	0.0	8.0
rishav	6.0	8	4	8.0	6.0	0.0	7.0	0.0	0.0	8.0
sumit	9.0	9	8	4.0	8.0	8.0	0.0	4.0	0.0	6.0
bhupendra	8.0	5	8	9.0	6.0	0.0	0.0	0.0	4.0	8.0
Abhishek	8.0	4	8	10.0	8.0	0.0	9.0	8.0	0.0	0.0
naman	0.0	5	8	10.0	6.0	0.0	0.0	9.0	0.0	0.0
Ankit	9.0	3	6	10.0	8.0	0.0	2.0	3.0	0.0	8.0
Sumit	0.0	7	9	10.0	6.0	0.0	1.0	8.0	4.0	9.0
ishan	7.0	6	2	10.0	0.0	0.0	1.0	6.0	0.0	0.0
rishi	7.0	5	6	10.0	0.0	0.0	1.0	7.0	4.0	9.0
babul	7.0	7	6	10.0	8.0	5.0	0.0	4.0	4.0	9.0
rakesh	0.0	3	4	10.0	8.0	2.0	1.0	7.0	4.0	7.0
Shivani	0.0	5	4	10.0	0.0	2.0	6.0	7.0	4.0	9.0
shristi	8.0	5	4	10.0	8.0	2.0	8.0	7.0	1.0	0.0

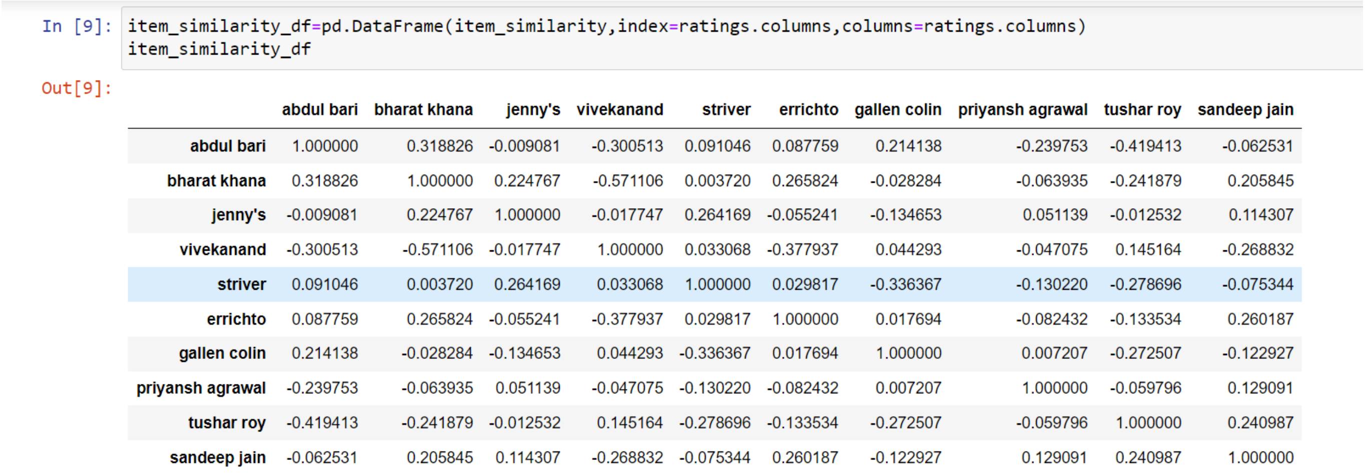
We are exporting raw data from our data sheet to pandas so that we can run our algorithms. For the students who haven't given any rating we are replacing the blanks with zeros.



```
In [6]: def standardize(row):
            new_row=(row- row.mean())/ (row.max() - row.min());
            return new_row
        ratings_std = ratings.apply(standardize)
        ratings_std=ratings_std.fillna(0);
        item_similarity=cosine_similarity(ratings_std.T)
        print(item_similarity)
```

```
[[ 1.          0.31882608 -0.00908145 -0.30051287  0.09104641  0.08775859
   0.21413845 -0.23975336 -0.41941304 -0.06253106]
 [ 0.31882608  1.          0.22476675 -0.57110629  0.00371991  0.26582431
  -0.0282838  -0.06393468 -0.24187883  0.20584461]
 [-0.00908145  0.22476675  1.         -0.01774744  0.26416911 -0.05524136
  -0.13465279  0.05113887 -0.0125322   0.1143071 ]
 [-0.30051287 -0.57110629 -0.01774744  1.          0.03306819 -0.37793731
   0.04429321 -0.04707471  0.14516361 -0.26883195]
 [ 0.09104641  0.00371991  0.26416911  0.03306819  1.          0.02981702
  -0.33636681 -0.13021994 -0.27869576 -0.0753444 ]
 [ 0.08775859  0.26582431 -0.05524136 -0.37793731  0.02981702  1.
   0.01769439 -0.08243238 -0.13353401  0.26018654]
 [ 0.21413845 -0.0282838  -0.13465279  0.04429321 -0.33636681  0.01769439
   1.          0.00720734 -0.27250652 -0.12292671]
 [-0.23975336 -0.06393468  0.05113887 -0.04707471 -0.13021994 -0.08243238
   0.00720734  1.         -0.05979619  0.1290914 ]
 [-0.41941304 -0.24187883 -0.0125322   0.14516361 -0.27869576 -0.13353401
  -0.27250652 -0.05979619  1.          0.24098706]
 [-0.06253106  0.20584461  0.1143071  -0.26883195 -0.0753444   0.26018654
  -0.12292671  0.1290914   0.24098706  1.        ]]
```

We are replacing zeros with standardized values so that it doesn't impact the teachers score negatively.

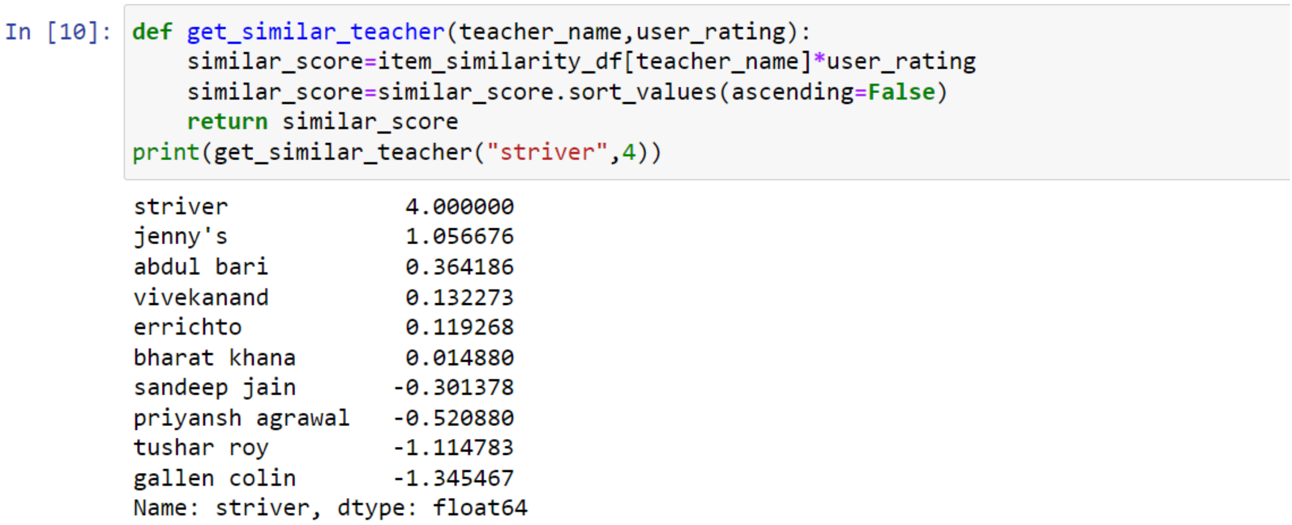


```
In [9]: item_similarity_df=pd.DataFrame(item_similarity,index=ratings.columns,columns=ratings.columns)
        item_similarity_df
```

Out[9]:

	abdul bari	bharat khana	jenny's	vivekanand	striver	errichto	gallen colin	priyansh agrawal	tushar roy	sandeep jain
abdul bari	1.000000	0.318826	-0.009081	-0.300513	0.091046	0.087759	0.214138	-0.239753	-0.419413	-0.062531
bharat khana	0.318826	1.000000	0.224767	-0.571106	0.003720	0.265824	-0.028284	-0.063935	-0.241879	0.205845
jenny's	-0.009081	0.224767	1.000000	-0.017747	0.264169	-0.055241	-0.134653	0.051139	-0.012532	0.114307
vivekanand	-0.300513	-0.571106	-0.017747	1.000000	0.033068	-0.377937	0.044293	-0.047075	0.145164	-0.268832
striver	0.091046	0.003720	0.264169	0.033068	1.000000	0.029817	-0.336367	-0.130220	-0.278696	-0.075344
errichto	0.087759	0.265824	-0.055241	-0.377937	0.029817	1.000000	0.017694	-0.082432	-0.133534	0.260187
gallen colin	0.214138	-0.028284	-0.134653	0.044293	-0.336367	0.017694	1.000000	0.007207	-0.272507	-0.122927
priyansh agrawal	-0.239753	-0.063935	0.051139	-0.047075	-0.130220	-0.082432	0.007207	1.000000	-0.059796	0.129091
tushar roy	-0.419413	-0.241879	-0.012532	0.145164	-0.278696	-0.133534	-0.272507	-0.059796	1.000000	0.240987
sandeep jain	-0.062531	0.205845	0.114307	-0.268832	-0.075344	0.260187	-0.122927	0.129091	0.240987	1.000000

Here is a correlation graph of the teachers. It defines that the student is likely to take a teacher of whom there is positive rating than the teachers which have negative rating.



```
In [10]: def get_similar_teacher(teacher_name,user_rating):
             similar_score=item_similarity_df[teacher_name]*user_rating
             similar_score=similar_score.sort_values(ascending=False)
             return similar_score
         print(get_similar_teacher("striver",4))
```

```
striver            4.000000
jenny's            1.056676
abdul bari         0.364186
vivekanand         0.132273
errichto           0.119268
bharat khana       0.014880
sandeep jain      -0.301378
priyansh agrawal  -0.520880
tushar roy        -1.114783
gallen colin      -1.345467
Name: striver, dtype: float64
```

Here there is a list of students which the student will definitely like given only one teacher he likes to the algorithm.

This model is perfectly working and the hashing connection with the main machine learning model is still work in progress. Currently it is being done manually but we will soon get it working completely.

Conclusion

Learning is a vital process that benefits us in a lot of ways. So Improving the learning experience of every student is the goal we are trying to achieve.

We are building a system where students can pick the teachers whom they find most comfortable and are most compatible with. We made a model based on Language used in the lectures , pace of the speech, No. of videos available from a single teacher, length of the videos and frankness. But after further processing we found out that the model wasn't accurate in suggesting teachers which will be suitable for the student.

We gathered data from surveys about what the students want most out of their learning experience and after gathering the data we found out that students were basically choosing their teachers based on sentiment and after that length of the videos were taken into account.

So after examining the data from the surveys we changed our model to incorporate the changes. Previously we assigned each category with the same weightage, which wasn't proving very accurate so we added different weights to different categories depending upon their importance. This was more accurate and so we were successful in creating a working model.

References

1. **Coursera** (*https://www.coursera.org/*) **-**
    - Completed MOOC from here.

2. **Kaggle** (*https://www.kaggle.com/*)**-**
    - Searched for Project Ideas and got the data set from here.

3. **Towards Data Science** (*https://towardsdatascience.com/*) **-**
    - Used to learn about several methods and algorithms and finalised the one used in the project.

4. **DeepLearning.AI** (*https://www.deeplearning.ai/*) **-**
    - Helped a lot in designing the solution for the chosen Problem.

THANK YOU