# Course Recommendation System

IMT2020001 - Srinivas Manda
IMT2020003 - Karanjit Saha
IMT2020084 - Arya Kondawar
IMT2020502 - Monjoy Narayan Choudhury

# Problem Statement

Design a Course Recommendation System using the given ratings of core courses.

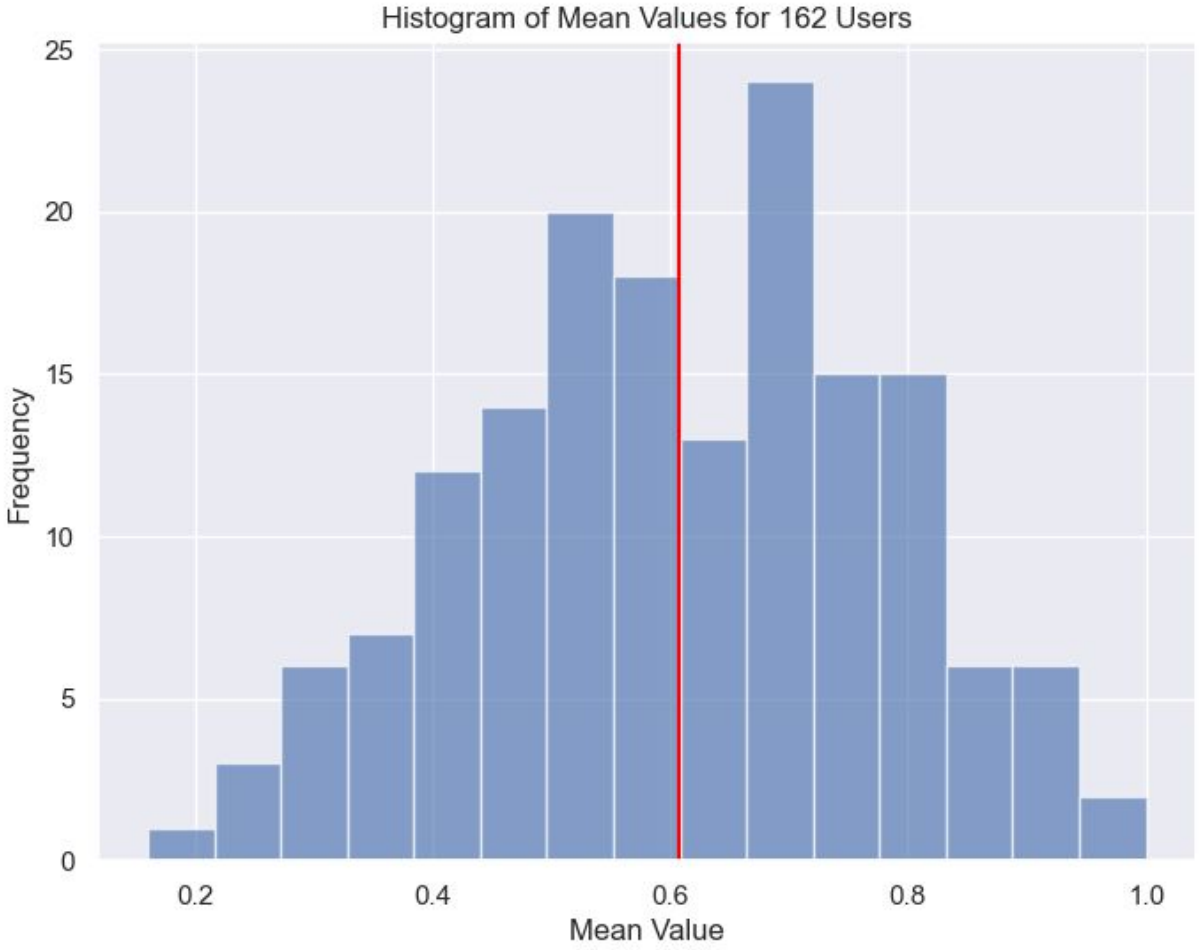
The Recommendation System will use a real - life dataset, consisting of 162 entries of students.

Our objective is to predict top n courses for a student based on the ratings of that student for core courses he/she have taken. We have designed a test client UI for the same, which highlights the proof of concept.

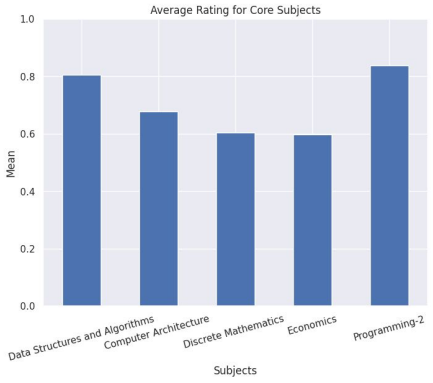
# Exploratory Data Analysis

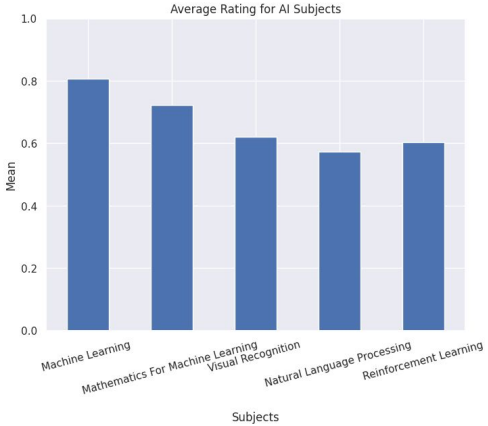
The original dataset is populated with 162 entries which has user ratings for each of the 25 courses. 5 of these courses are core courses while rest 20 are electives. It is not mandatory that each user filled for each subject

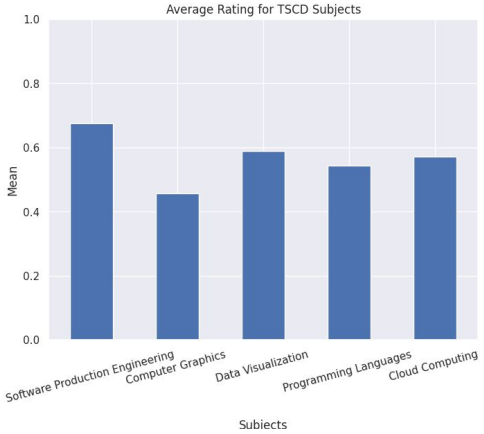
We further split this dataset in into train and test with a test ratio of = 0.05 i.e 9 test points

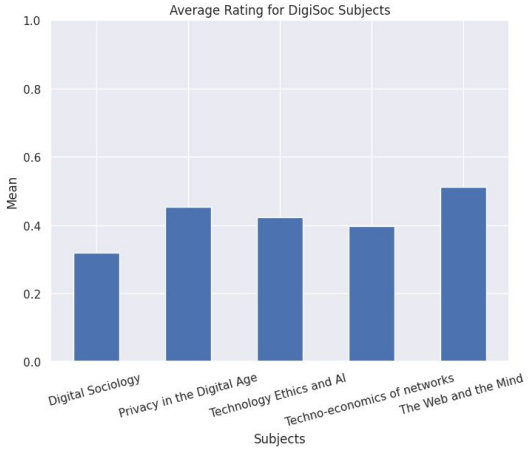


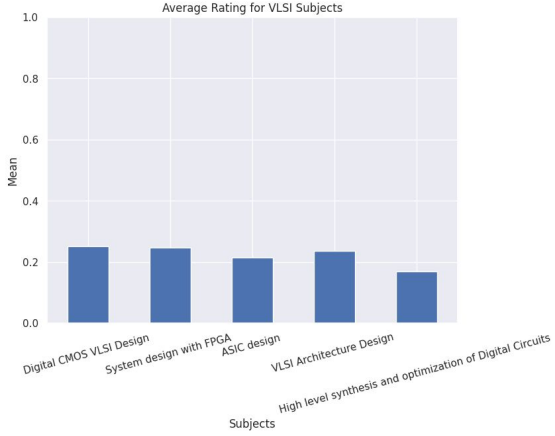
# Average Rating For Courses



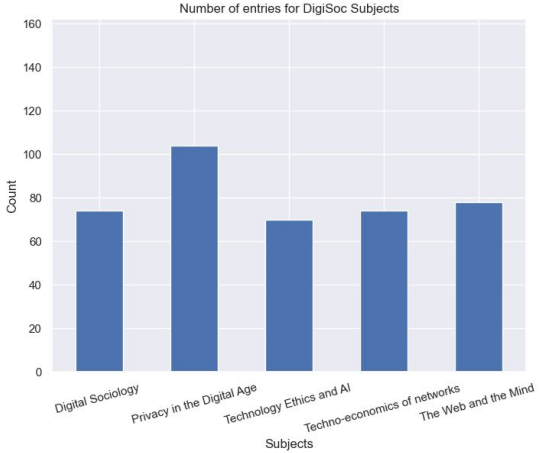



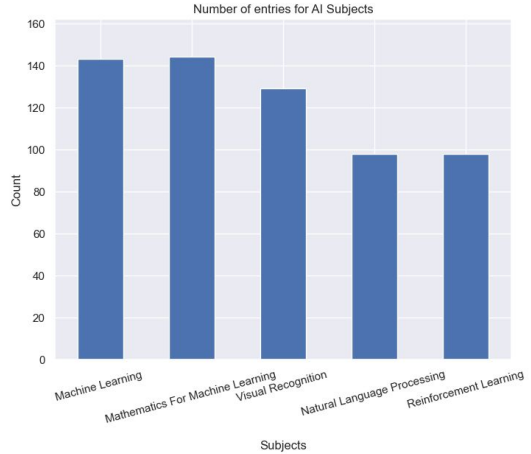



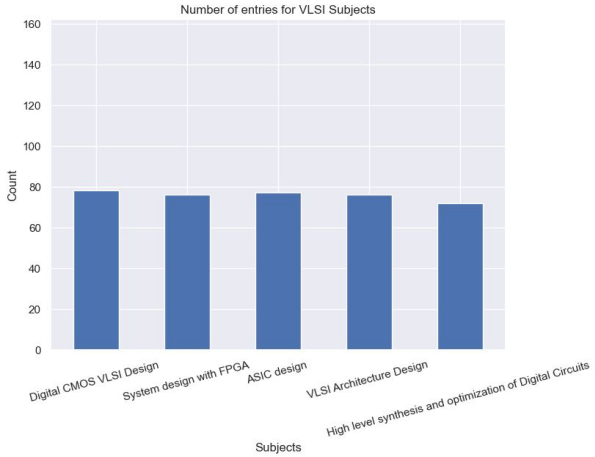



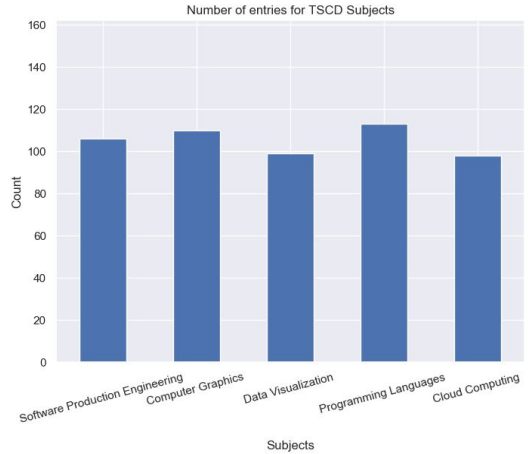



# Number of ratings per elective



# Initial Observations

From the above shown graphs we can infer the following:

- “Discrete Mathematics” is listed as core course. Since it was a core course of only CSE branch, hence people from other branches like ECE and Digital Society would not have taken this course. Due to this either they would not have filled the form or would have filled the Discrete Mathematics column with a random value. Hence our dataset would be biased towards courses taken by CSE people.

- Since Machine Learning, Mathematics for Machine Learning and Software Production Engineering have been rated by many people and have high average rating, hence for many users these 3 subjects are among the top recommendations.

- Since only few ECE people have filled the form, hence the ECE subjects have a bias against them. Hence the ECE electives should lie at the end in the list of recommended subjects.

# Components used in our Recommendation System

K Means:

- Centroid Initialisation: Randomly done with seed = 42.
- Inertia is computed for plotting the elbow plot.
    - Inertia: Sum of squared distances of samples to their closest cluster center

SVD:

- All functions were made from scratch.
- We initially tried power iteration and noticed the following
    - In Eigen Decomposition: For small eigenvalues power iteration never converge to that value irrespective of the large number of iteration.
        - Convergence: We compare nth and n-1th iteration's eigen vectors and check whether they below the certain threshold/tolerance (chosen 1e-6).

# Metrics For Evaluation

1. Top n accuracy: In this we see whether our predicted subjects based on decreasing order of preference/rating lies in the top n rated subjects of the test set. For our study we used n = 5.

2. MAE: Mean Absolute error to compare the value of predicted ratings and actual ratings

# Our approaches

We have used the following approaches for our building our course recommendation system:-

- K-Means Clustering and then using Reduced SVD for noise reduction.
- Performing Reduced SVD and applied K-Means.
- Collaborative Filtering (User based) as a benchmark.

We are now going to look at each of the above approaches in detail in the following slides.

# Approach 1

We do the following steps:

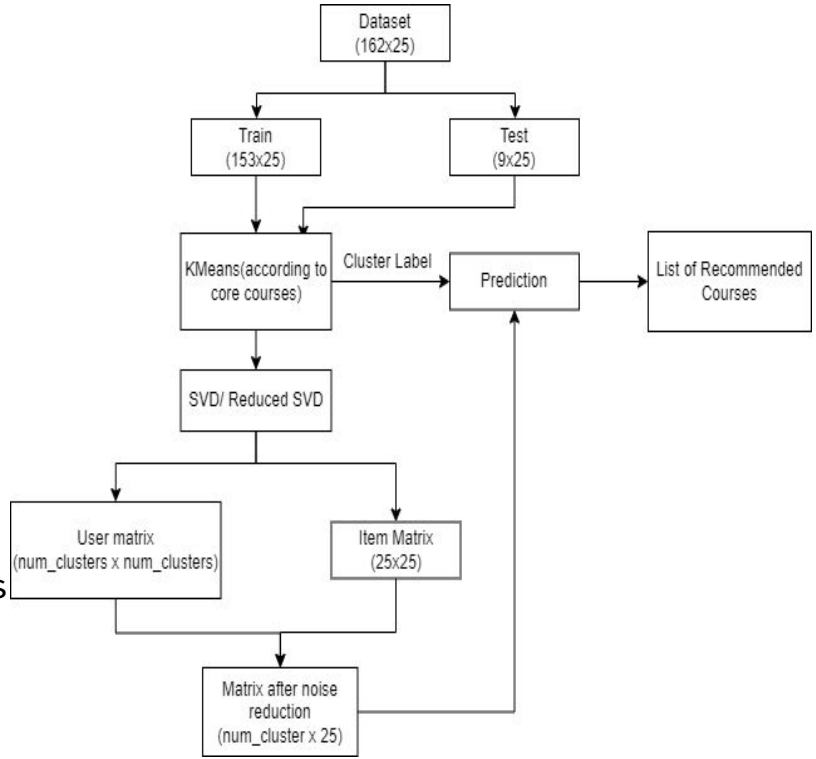
1. First we train KMeans object with train dataset and then create a matrix of size num_clusters x 25( storing average rating of every subject for users in the cluster)

2. We then pass the above created matrix through SVD to get user and item matrix.

3. We multiply the above 2 matrices to get a new matrix

4. For testing dataset we predict their labels using KMeans object.

5.We use the above obtained label and the matrix to make predictions and then recommend the courses based on the predicted ratings for courses.

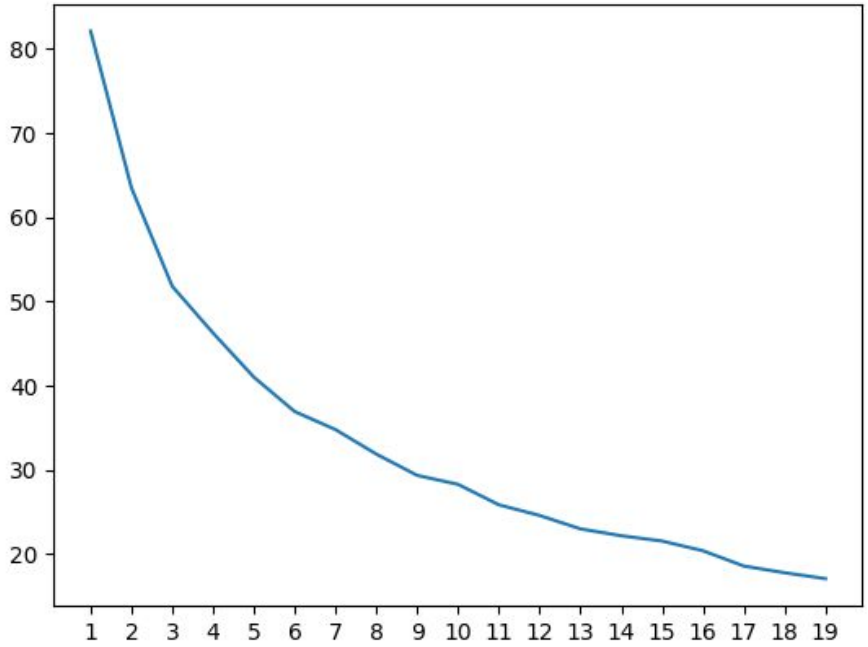


# Design Choices

1. Selecting “K” in K-Means:
   For this we take the elbow plot to come up with the best K. However, as you can see there is no proper single “elbow” seen. We can see kinks (inflections) at 4,5,7,12. We decided to go ahead to try the approach for these values of K.

   Intuitively we also thought that elective courses count under different specialisation and the core subject affinity should have some say in what students would prefer to continue pursuing. Hence we also thought of trying k=4 irrespective of the elbow.

2. Amount of Singular Values to remove in “Reduced SVD”
3. Imputing NaN values with 0.5 or mean rating of a subject.

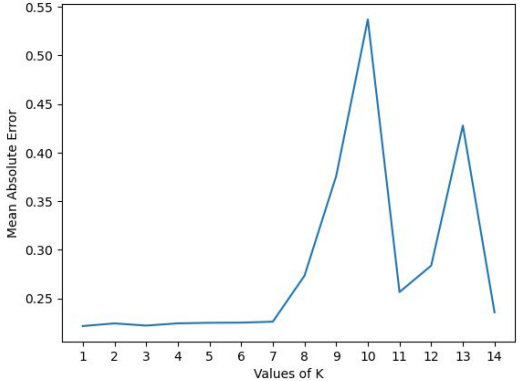



**We applied our own implementation of grid search on number of clusters and number of singular values to remove and applied it on train data and found the best results. Following plots are results obtained on train-set**

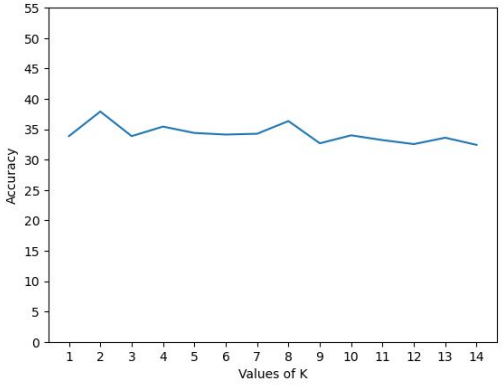
# Observation With NaNs = 0.5

**Final Parameters:**
number of clusters = 8
singular Values removed = 2

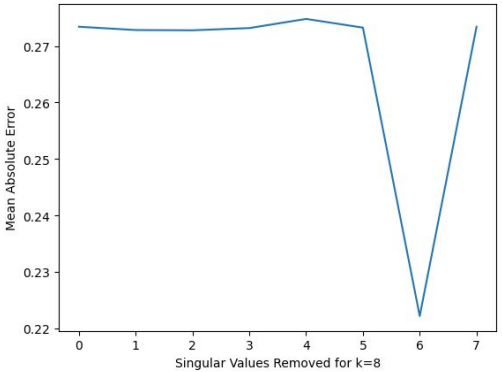
| Values of K | Mean Absolute Error |
|---|---|
| 1 | 0.221655 |
| 2 | 0.224379 |
| 3 | 0.222134 |
| 4 | 0.224437 |
| 5 | 0.224994 |
| 6 | 0.225169 |
| 7 | 0.226130 |
| 8 | 0.273407 |
| 9 | 0.375850 |
| 10 | 0.537149 |
| 11 | 0.256616 |
| 12 | 0.283795 |
| 13 | 0.427912 |
| 14 | 0.235773 |



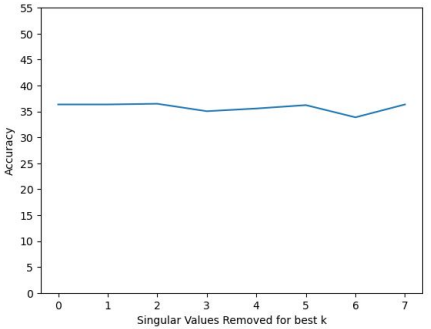

| Values of K | Accuracy |
|---|---|
| 1 | 33.856209 |
| 2 | 37.908497 |
| 3 | 33.856209 |
| 4 | 35.424837 |
| 5 | 34.379085 |
| 6 | 34.117647 |
| 7 | 34.248366 |
| 8 | 36.339869 |
| 9 | 32.679739 |
| 10 | 33.986928 |
| 11 | 33.202614 |
| 12 | 32.549020 |
| 13 | 33.594771 |
| 14 | 32.418301 |




| Singular Values Removed | Mean Absolute Error |
|---|---|
| 0 | 0.537149 |
| 1 | 0.537171 |
| 2 | 0.533484 |
| 3 | 0.532175 |
| 4 | 0.531912 |
| 5 | 0.530886 |
| 6 | 0.529573 |
| 7 | 0.528284 |

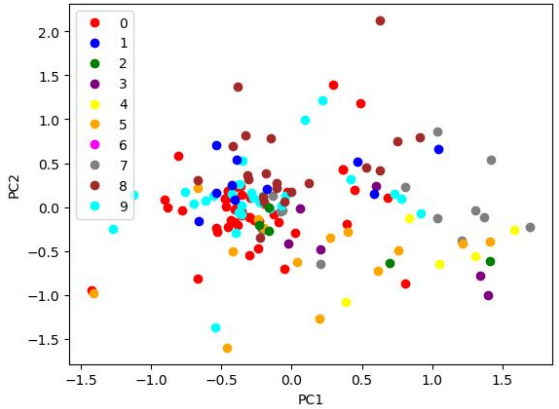


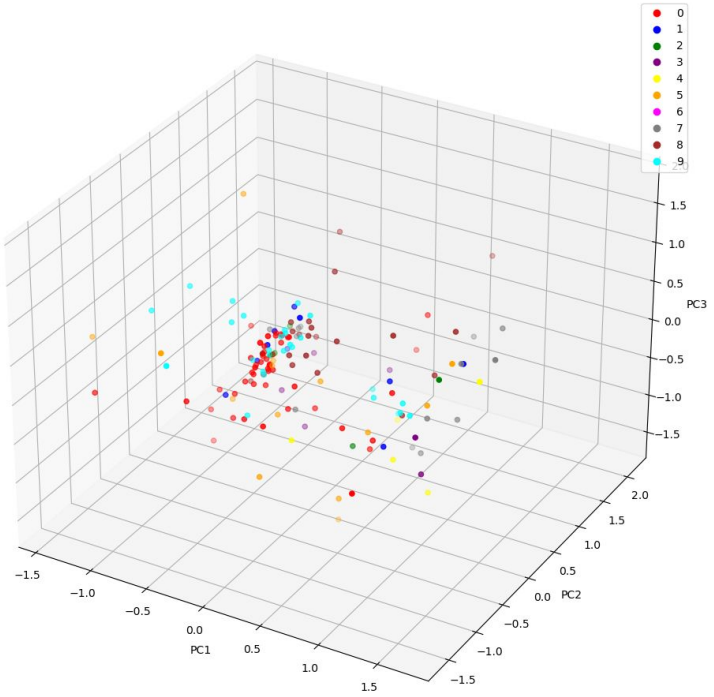

| Removed values | Accuracy |
|---|---|
| 0 | 33.986928 |
| 1 | 35.032680 |
| 2 | 33.986928 |
| 3 | 33.464052 |
| 4 | 33.986928 |
| 5 | 33.333333 |
| 6 | 34.117647 |
| 7 | 33.856209 |




From this approach, we see an accuracy of 48.88% on test data.

# Attempt to analyse using PCA - 3, 2 dimension




We notice that from the 3d plot that cluster 6 (magenta) has no members in it. Also cluster 2,3,4 (green,purple,yellow) have very less members and most go to cluster 0 (red)

| Cluster Number | Number of points in it |
|---|---|
| 0 | 52 |
| 1 | 10 |
| 2 | 5 |
| 3 | 6 |
| 4 | 5 |
| 5 | 14 |
| 6 | 0 |
| 7 | 13 |
| 8 | 21 |
| 9 | 27 |

# Top 5 subjects of each cluster representative

Note that cluster 6 has no data as its an empty cluster

## Cluster 0

Machine Learning
Mathematics For Machine Learning
Software Production Engineering
Visual Recognition
Cloud Computing

## Cluster 1

Machine Learning
Mathematics For Machine Learning
Software Production Engineering
Reinforcement Learning
Natural Language Processing

## Cluster 2

Machine Learning
Software Production Engineering
Mathematics For Machine Learning
Cloud Computing
Programming Languages

## Cluster 3

Machine Learning
Mathematics For Machine Learning
Privacy in the Digital Age
Technology Ethics and AI
The Web and the Mind

## Cluster 4

Machine Learning
Mathematics For Machine Learning
Natural Language Processing
Reinforcement Learning
Visual Recognition

## Cluster 5

Machine Learning
Software Production Engineering
Mathematics For Machine Learning
Cloud Computing
Visual Recognition

## Cluster 7

Machine Learning
Mathematics For Machine Learning
Natural Language Processing
Reinforcement Learning
Visual Recognition

## Cluster 8

Machine Learning
Mathematics For Machine Learning
Software Production Engineering
Visual Recognition
Cloud Computing

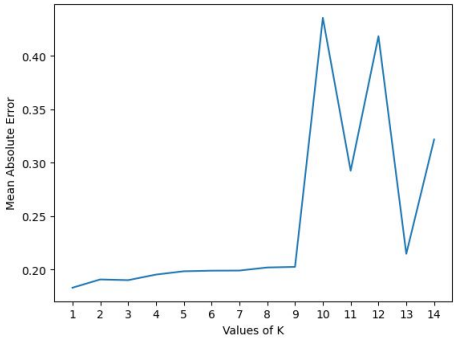
## Cluster 9

Machine Learning
Mathematics For Machine Learning
Software Production Engineering
Visual Recognition
Reinforcement Learning

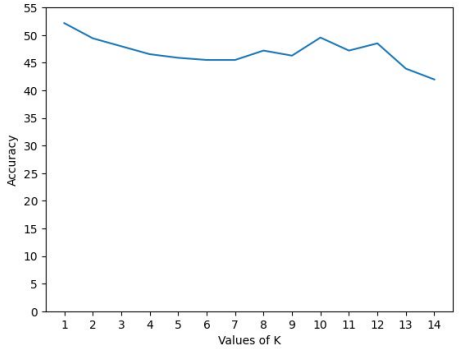
# Observation With NaNs = Mean rating of Subject

**Final Parameters:**
number of clusters = 10
singular Values removed = 7

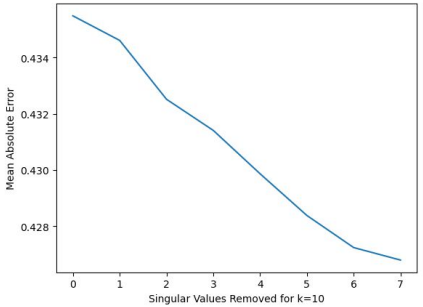
| Values of K | Mean Absolute Error |
|---|---|
| 1 | 0.183020 |
| 2 | 0.190758 |
| 3 | 0.190090 |
| 4 | 0.195289 |
| 5 | 0.198413 |
| 6 | 0.198957 |
| 7 | 0.199071 |
| 8 | 0.201933 |
| 9 | 0.202508 |
| 10 | 0.435482 |
| 11 | 0.292483 |
| 12 | 0.418309 |
| 13 | 0.214732 |
| 14 | 0.321667 |



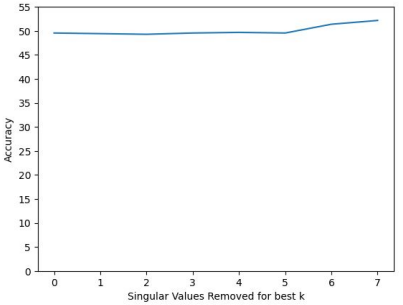

| Values of K | Accuracy |
|---|---|
| 1 | 52.156863 |
| 2 | 49.411765 |
| 3 | 47.973856 |
| 4 | 46.535948 |
| 5 | 45.882353 |
| 6 | 45.490196 |
| 7 | 45.490196 |
| 8 | 47.189542 |
| 9 | 46.274510 |
| 10 | 49.542484 |
| 11 | 47.189542 |
| 12 | 48.496732 |
| 13 | 43.921569 |
| 14 | 41.960784 |




| Singular Values Removed | Mean Absolute Error |
|---|---|
| 0 | 0.435482 |
| 1 | 0.434608 |
| 2 | 0.432514 |
| 3 | 0.431406 |
| 4 | 0.429863 |
| 5 | 0.428380 |
| 6 | 0.427238 |
| 7 | 0.426793 |

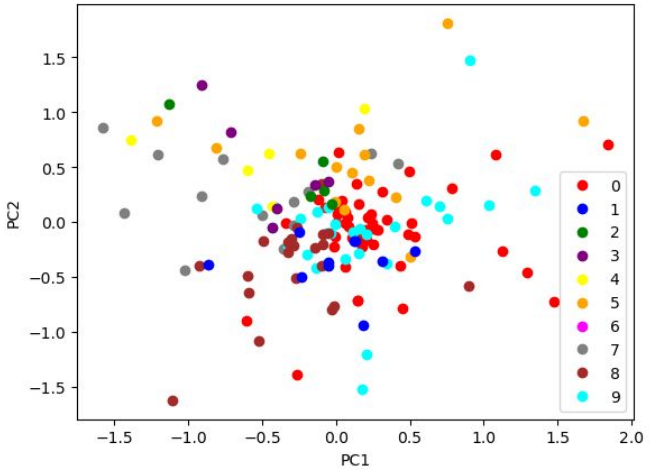


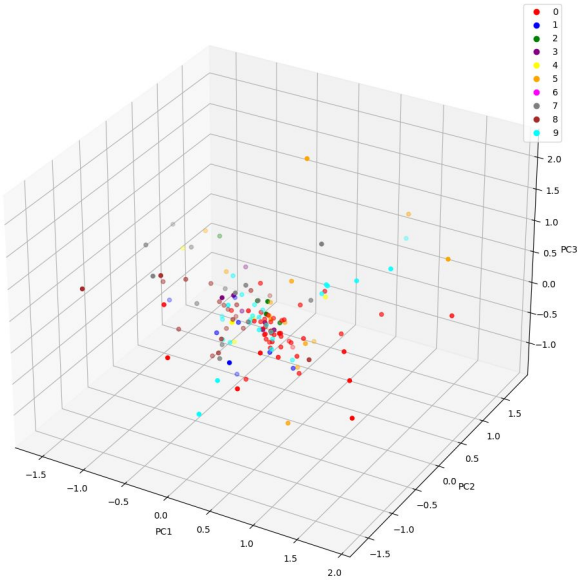

| | Removed values | Accuracy |
|---|---|---|
| 0 | 0 | 47.189542 |
| 1 | 1 | 47.189542 |
| 2 | 2 | 45.751634 |
| 3 | 3 | 49.019608 |
| 4 | 4 | 50.065359 |
| 5 | 5 | 50.588235 |
| 6 | 6 | 52.156863 |
| 7 | 7 | 47.189542 |




From this approach, we see a better accuracy of 51.11% on test data so we decide to use mean ratings of the subject as NaN replacement. Also a significant increase in training accuracy occurs

# Attempt to analyse using PCA - 3, 2 dimension




| Cluster Number | Number of points in it |
|---|---|
| 0 | 52 |
| 1 | 10 |
| 2 | 5 |
| 3 | 6 |
| 4 | 5 |
| 5 | 14 |
| 6 | 0 |
| 7 | 13 |
| 8 | 21 |
| 9 | 27 |

Here also we notice that from the 3d plot that cluster 6 (magenta) has no members in it. Also cluster 2,3,4 again (green,purple,yellow) have very less members and most go to cluster 0 (red)

# Top 5 subjects of each cluster representative

Note that cluster 6 has no data as its an empty cluster
The ones in italic represent a change in order while the one in bold suggests a newer subject came after we changed the NaN values to user mean.

## Cluster 0

Machine Learning
Mathematics For Machine Learning
Software Production Engineering
Visual Recognition
**Reinforcement Learning**

## Cluster 1

Machine Learning
Mathematics For Machine Learning
Software Production Engineering
Reinforcement Learning
Natural Language Processing

## Cluster 2

Machine Learning
Software Production Engineering
Mathematics For Machine Learning
Cloud Computing
Programming Languages

## Cluster 3

Machine Learning
Mathematics For Machine Learning
Privacy in the Digital Age
**The Web and the Mind**
**Data Visualization**

## Cluster 4

Machine Learning
Mathematics For Machine Learning
**Software Production Engineering**
Reinforcement Learning
*Natural Language Processing*

## Cluster 5

Machine Learning
Software Production Engineering
Mathematics For Machine Learning
Cloud Computing
Visual Recognition

## Cluster 7

Machine Learning
Mathematics For Machine Learning
*Reinforcement Learning*
*Natural Language Processing*
**Software Production Engineering**

## Cluster 8

Machine Learning
Mathematics For Machine Learning
Software Production Engineering
Visual Recognition
Cloud Computing

## Cluster 9

Machine Learning
Mathematics For Machine Learning
Software Production Engineering
Visual Recognition
Reinforcement Learning

# Some possible explanation to why such changes ?

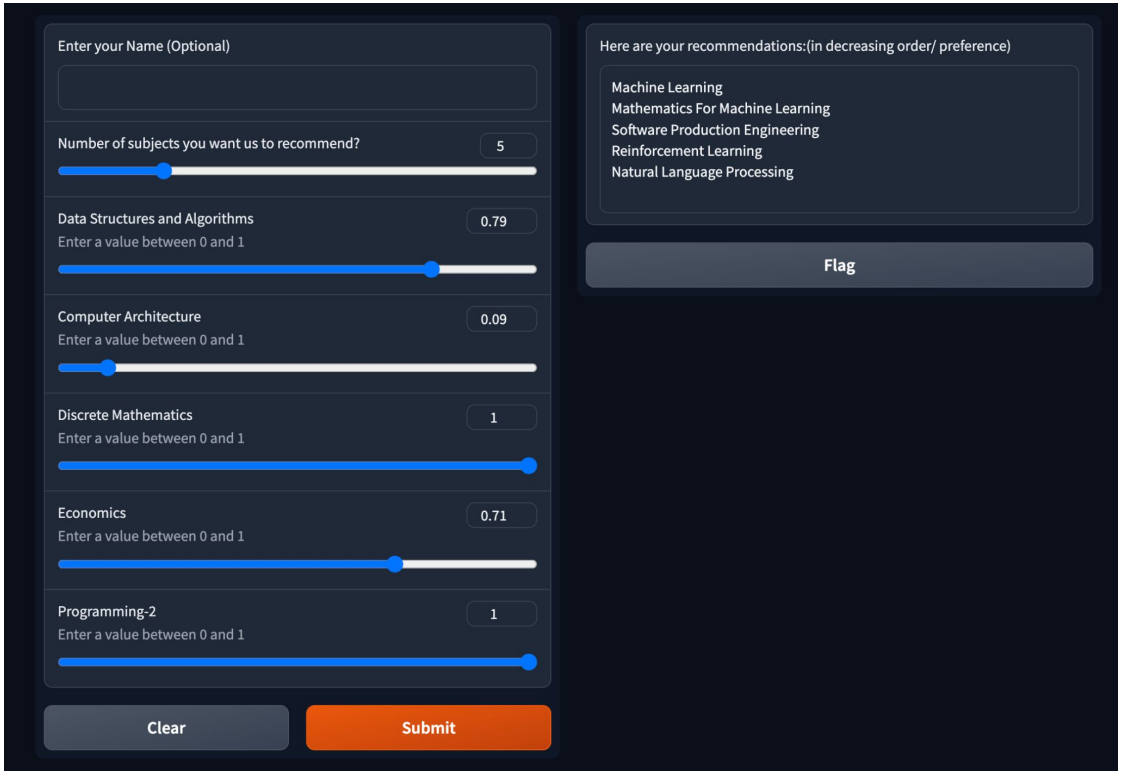
- Since we have replaced the NaN values with 0.5 in the first case, there is a chance that the overall mean of that subject reduces. This happens especially in cases where the number of person who have filled is low.

- Now when using mean of the column. Irrespective of how many filled the value is used to replace NaN, will be decided by the ratings given by those users.

- In case of SPE and VR(notably). The only logic we could come up with is that the overall rating values is dictated by the number of users who rated as well as the ratings and we don't expect a big difference in value of the ratings of those two subjects.

# Final Model

NaN filled with mean rating of the subject.
Number of Cluster = 10
Singular Values removed = 7
Test Ratio = 0.05
Accuracy based on top 5 (test) = 51.11%




A test client for demo

# Approach 2

Reduced SVD followed by K-Means

- We apply reduced SVD and perform a noise removal on the dataframe itself.
- Using that matrix as input, we take the core subjects and apply K Means on the train set created.
- For testing dataset we predict their labels using KMeans instantiation.
- We use the above obtained label and the Matrix to make predictions and then recommend the courses based on the predicted ratings for courses.

# Performance

- We applied our own implementation of grid search on number of clusters and number of singular values to remove and applied it on train data and found the best results
- For NaNs replaced in dataset by 0.5
  - We observed that number of clusters = 6 and number of singular values to remove = 1 , gave the best training accuracy. Corresponding to this we got accuracy of 53.33% on the test data.
- For NaNs replaced in dataset by mean of the subject rating.
  - For the same values of number of clusters and number of singular values removed to 53.33% on the test set (The accuracy values in both cases vary by a negligible amount).
- Finally we use the following values:
  - Number of clusters = 6
  - Singular values to remove = 1
  - NaNs replaced with mean of the subject rating

# Approach 3

We perform user based collaborative filtering to see how our K means approach fare.

- We kept the same train test ratio
- We performed the following steps on the data:

1. Find similar user based on ratings on common items: Here everyone mandatorily fills core so it's easy. We use cosine similarity on mean normalised data to calculate similarity. Also we had replaced all empty entries with the mean rating of the particular user.
2. Identify items rated by similar users for the user in question:
3. Calculated weighted average scores.
4. Rank them.

# Performance

---

- It performs better in terms of accuracy for the given test ratio from both the previous approaches.

- Accuracy (top 5): 62.222%

# Things we can try

- We had replaced the NaNs one way or another. We also tried not replacing them with anything and run approach 1. The results were suboptimal (22% accuracy on test set). However we have not extensively checked our implementation for all values i.e run a grid search on all possible values of n and values to remove.
- Since we tried mean of subject ratings, we can also try to replace NaNs with the mean of user ratings.
- We found out about more centroid initialisation methods in KMeans like KMeans++ but we did not implement it yet. This can maybe give better results due to better centroid initialisation than random.
- Centroid initialisations use some sort of randomisation and doesn't generally use any data driven technique. We found from a paper titled "A News Recommendation Algorithm Based on SVD and Improved K-means". This paper suggests that: We can try to implement this as a proof of concept on the given dataset, however we are bit skeptical about the use cases the paper address. The papers talks about how the centroid of a cluster can be initialised with the mean of the cluster.