**ABSTRACT:**

Recommendation Systems are becoming an integral part of our everyday life. There are many techniques that are currently in use for this purpose. Here, I propose a K-clustering based recommendation system which was originally used to recommend k topics from a given set of n topics that were put up for display as the varied research areas on offering by IIT Kharagpur for each of their courses. Instead of using the traditional K-Means clustering algorithm which has the pitfalls of randomly choosing initial clusters while being limited to numeric data only and also requiring a certain number of iterations to achieve a good result, I used a vector based approach for fixed initial cluster center initialization which allows for repeatable clustering results whilst working with string data and requiring only one iteration to produce the desired final clusters. \*\***While K-means can be modified for use with string datatypes, the clusters produced using my technique gave better results wherein similar subjects were grouped together more accurately**.\*\* This system can be used for recommending k subjects, their sub topics or even be extended for any other object, based on the underlying k clusters.

1. **INTRODUCTION:**

A recommendation system is an important component in many walks of life, especially in online services. Such systems help users to find more content in the domain of their liking while also helping service providers by increasing their service usage which ultimately leads to an increase in revenue. A unique problem in this field of recommendation systems was put up by the team responsible for developing a new website for IIT Kharagpur.

Each stream branch at any institute has a vast number of subjects to offer which a student can pursue as his / her research topic. But not all of these subjects can be uploaded to the institute’s website as its research offering since it makes the website huge and clunky. Usually, teachers of each branch meet up and decide on a minimum number of topics which best represent their stream which are then put up on their website. The amount of time and manual labour required for such a task becomes huge as all decisions need to be accounted for from each individual professor. Thus, the need of a recommendation system arose which, given a set of topics or subjects, could recommend the required solution.

Clustering methods require inter-cluster distance to be maximum while minimizing intra-cluster distance. In general K-means is a popular partitional clustering algorithm which aims for the same objective. The problem with random center cluster initialization for this particular problem is that if any topics which are closely similar to each other are chosen as centroids, then the resulting clusters become redundant or insignificant. We require a method wherein the K centroids chosen are topics which are of least similarity.

Given a set of objects, first we need to identify a similarity measure as the basis of our clustering. For this particular problem, subjects can be distinguished from one another using the topics, sub-topics and their description that are present within them. Working with non-numeric data requires their conversion into meaningful numeric values. So, once the collected data for each subject was pre-processed, tf-idf was applied which yielded high dimensional vectors. Using cosine-similarity, the topics furthest away from each other and the rest of the data were selected as the starting points. Each vector corresponding to their respective topics were then put into the closest matching cluster. After the clustering was done, which required only one iteration over the entire dataset, the topics which best represent their respective clusters were chosen and then recommended.

1. **METHODS:**
   1. **Dataset:**

26 topics from both Computer Science and Engineering along with Electronics and Communication Engineering departments were taken separately for clustering. The similarity measure chosen was the syllabus, taken from IIT Kgp’s Website itself for each topic from both streams as the syllabus consists of topics, sub topics and their description which easily allows the subjects to be distinguished from one another.

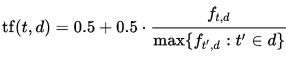
* 1. **Pre-processing:**

Data for each topic was loaded into python and \*\****topia.termextract***\*\* package, which uses a simple Parts Of Speech(POS) tagging algorithm to identify important terms in a document, was used to extract necessary terms from the data which were then stored in a separate file after which stop-word removal was performed in case any unnecessary words were left behind.

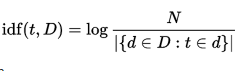
* 1. **Algorithm:**

In order to work with non-numeric data, they must be first converted into meaningful numeric values. \*\****tf–idf***\*\*, short for \*\****term frequency–inverse document frequency***\*\*, is a numerical statistic that is intended to reflect how important a word is to a [document](https://en.wikipedia.org/wiki/Document) in a collection or [corpus](https://en.wikipedia.org/wiki/Text_corpus). It is often used as a weighting factor in information retrieval and [text mining](https://en.wikipedia.org/wiki/Text_mining).

In the case of the \*\***term frequency**\*\* tf(*t*,*d*), the *augmented frequency* method was considered, in order to prevent a bias towards longer documents, i.e. *raw frequency*, which is the number of times a term *t* occurs in document *d*, divided by the maximum raw frequency of any term in the document i.e. If we denote the raw frequency of *t* by *ft*,*d*, then the augmented *tf* scheme is

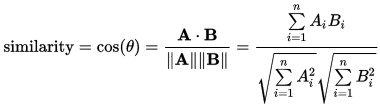


The \*\***inverse document frequency**\*\* is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. It is the [logarithmically scaled](https://en.wikipedia.org/wiki/Logarithmic_scale) inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.



Where, **N** is the total number of documents in the corpus **N = | D|,** and is the number of documents where the term ***t*** appears i.e., **tf(t, d) ≠ 0**. If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to **.**

Using tf-idf, the data for each topic was converted to a high dimensional vector. In order to compare these vectors a distance metric was required which was generated using **cosine similarity**. It is a measure of similarity between two non-zero vectors of an [inner product space](https://en.wikipedia.org/wiki/Inner_product_space) that measures the [cosine](https://en.wikipedia.org/wiki/Cosine) of the angle between them. The cosine of 0° is 1 and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude. The cosine of two non-zero vectors can be derived by using the [Euclidean dot product](https://en.wikipedia.org/wiki/Euclidean_vector#Dot_product) formula:



Where, and are components of vector ***A*** and ***B***. So a score of 1 between two vectors means that they are exactly the same while a 0 indicates that they are completely different.

A NxN matrix was constructed to keep track of the similarity score between each vector with all other vectors. Now for each vector in the matrix, its similarity score with all other vectors where i **≠** j was summed up and the result was sorted in a non-decreasing order. The first K vectors were considered as centroids for the K clusters since they would be the most dis-similar topics. This ensured \*\***high inter cluster distance**\*\*. Now for each remaining topic, its corresponding vector was put into the cluster that resulted in the maximum similarity score on comparison with each of the centroids. This ensured \*\***low intra cluster distance**. \*\*

Once the clustering was done, K topics were required to be output. Because of the nature of the clusters, it is not necessary that the centroids best represent their respective clusters because such a topic could be present that encompasses all other topics and is thus the best representative of its cluster. To solve this problem, similarity score for each topic against all other topics was summed up. This time, the subject with the maximum score means that it is strongly connected with all the other topics and would thus best represent the contents of its cluster. This was iterated over all the K clusters to finally get the recommendation, K topics which best represent the institute’s research variety.