Proposed Work and methodology

Proposed Work:

Proposed recommendation system is personalized recommendation. It uses clustering technique and similarity metric to produce recommendation that are user dependent.

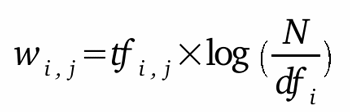
1. Scraping data from fancy news sites.
2. Preprocess data and using TfIdf vectorization technique for data vectrization.

**Why TF-IDF vectorization?**

TF represents Term Frequency which counts number of times word occur in given instance .

IDF represents Inverse Document Frequency in which counts this word in whole dataset.

Below image represents mathematical formula for TF-IDF vectorization.



TF-IDF is far better then other vectorization technique as every frequency of word signifies its values in that instance of dataset.

1. Using K-Means clustering technique.

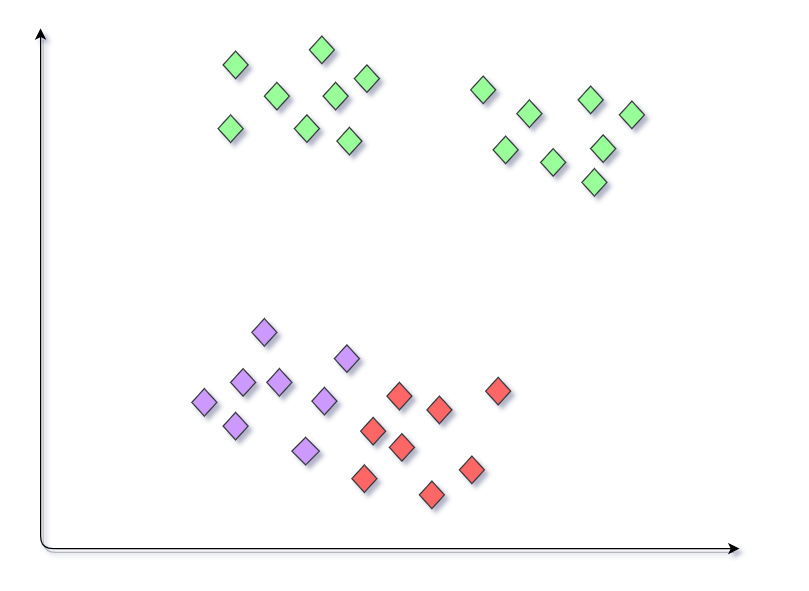
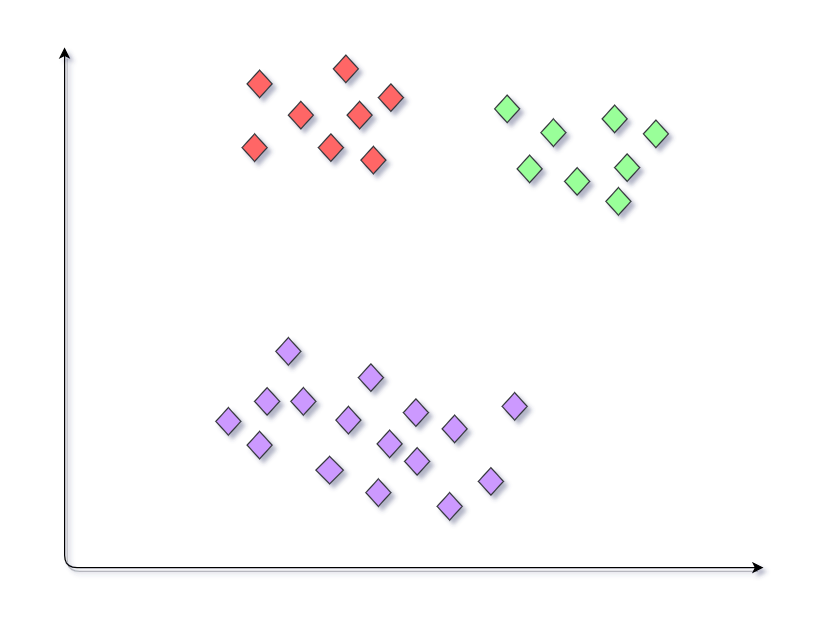
**Why K- Means CLustering?**

There are various clustering technique like hierarchical clustering,but K-means clustering offers scaling at ease for large data,can warm start positions of centroids.

Here we did not start centriods of points randomly in space, as we may fall in random initialization trap.

**What is random initialization trap?**

For same data if we assume that we have started there clustering now on randomly initializing centroids we will always get same data point in different cluster when compared other. We can below 2 images as her clustering have fallen for random initialization trap.



**How to overcome random initialization trap?**

To overcome this trap kmeans proivdes a powerful algorithm that work behind k-means for centroid initialization it is **k-means++**.

The exact algorithm is as follows:

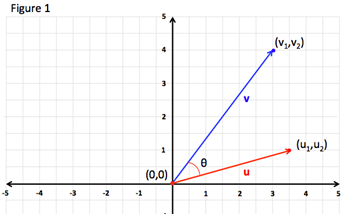
1. Choose one center uniformly at random from among the data points.
2. For each data point x, compute D(x), the distance between x and the nearest center that has already been chosen.
3. Choose one new data point at random as a new center, using a weighted probability distribution where a point x is chosen with probability proportional to D(x)2.
4. Repeat Steps 2 and 3 until k centers have been chosen.
5. Now that the initial centers have been chosen, proceed using standard k-means clustering.

After clustering all we have to is to use cosine similaritiy to find top recommendation.here we are going to use Cosine similarity.

**What is cosine similarity?**

Cosine similarity works between two vectors for cause how similar they are? Similarity between these vectors can be decided on basis cosine angle between two vectors.

Less angle represents more closer and overlapping represents same vector.



so we can consider that two vectors are news article 1 and article 2. and since we already used tfidf vectorizer that creates a sparse matrix of vector for each article.

Now we can retrieve list of article with maximum values of cosine from same cluster class user is reading. This will be used for recommendation of user.

Methodology:

Data Scraping:

1. In this project we are going to use open source library to collect data called **Scrapy.**
2. Using **scrapy** we created a spider/web-crawler.
3. for this project we are going to scrap only following attributes.
4. News headline
5. Short description of news
6. News Category
7. Link of news
8. Date

Import scraped data to system using **pandas** library.

Data Preprocessing:

1. Taking headline now we use **re** library to parse according to defines regular expression to keep only small alphabets, capital alphabets, space.
2. Lower each headline to reduce word frequency by just difference of small and capital letter. (Eg - nice and Nice)
3. Removing unwanted pronunciation and words called stopwords. Folowing are some stop words [“i”, “me”, ‘ours’]
4. Now we apply **vectorization.** Its process of converting each word to a vector representation according to chosen algorithm. Here we choose **TF-IDF vectorization** for our dataset.

This vectorization is implemented from **sklearn.preprocessing.text.TfidfVectoizer**

Now our data is ready, lets see model implementaion.

**Model design:**

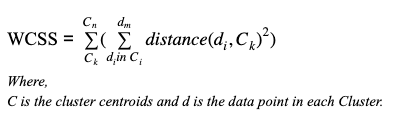
1. Here we are going o implement K-means clustering to apply k-means we can use module from **sklearn.cluster.KMeans.**
2. But before applying k-means clustering we need to find optimal number of clusters. This can be done by plotting **Elbow Plot.**

**Eblow Plot**  is plotted between:

WCSS (Y-axis)

Number of clusters (X-axis)

1. WCSS - it stands for Within Clusters Sum of Squares, here we find its value by given mathematical formula.

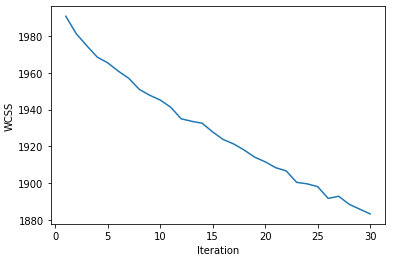


During applying k-means clustering there is initialization of points ,but since they are in group algorithm always takes their centroid. Now this initialization can be done by choosing algorithm accordingly.

Plotting a graph is quite simple we can loop for various values from [1,n]

To find their WCSS values from cluster-1 to cluster-n and graph can be plotted.

The graph of our prototype model is shown below.



Here idea is find minimum but optimal WCSS value.To find optimal cluster we point from top of graph and find a **Elbow** like structure/ like interruption (cringe point where value is suddenly decreased smoothly and further increased smoothly) that point represent our optimal cluster count.

**Generating recommendations**

As user visits a article

1. We can predict the cluster class of article.
2. Apply cosine similarity of that article with article belonging to same cluster class.
3. From various generated cosine values take top 10 and recommend article corresponding to it.

To get fast access to cluster one can maintain dictionary to lower its search time. Like cluster\_id:[list of article].

**Tools and technology used**

**Software used:**

1. Python 3.x
2. Scikit-learn
3. Numpy
4. Pandas
5. Matplotlib.pyplot
6. Re
7. Nltk

**Hardware used:**

Descent working machine can be considered.

**References:**

1. Machine Learing A-ZTM:Hands on Pyhton and R in Data Science-<https://www.udemy.com/course/machinelearning/>
2. Scrapy documentation- <https://docs.scrapy.org/en/latest/>
3. Sentdex channel- <https://www.youtube.com/user/sentdex>