RECOMMENDATION SYSTEM TO PROMOTE TOURISM

A PROJECT REPORT

Submitted by

Laraib Ahmed Ansari	**RollNo:12617001069**	**RegNo:171260110068**
Navonil Das	**RollNo:12617001083**	**RegNo:171260110082**
Neha Chowdhary	**RollNo:12617001084**	**RegNo:171260110083**
Swarnadwip Bose	**RollNo:12617005056**	**RegNo:171260110910**

Under the Supervision of

Prof.(Dr.) Anindita Kundu
Assistant Professor
Dept. of Computer Science and Engineering
Heritage Institute of Technology, Kolkata

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA

Autonomous Institute under

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

August, 2021

HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA

An autonomous Institute under

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

BONAFIDE CERTIFICATE

Certified that this project report "**RECOMMENDATION SYSTEM TO PROMOTE TOURISM**" is the bonafide work of **Laraib Ahmed Ansari**, **Navonil Das**, **Neha Chowdhary** and **Swarnadwip Bose** who carried out the project under my supervision.

SIGNATURE

Prof.(Dr.)
Subhasis Majumdar

Head of the Department

Computer Science and Engineering

Heritage Institute of Technology
Chowbaga Road T.O, Anandapur,

East Kolkata Township,

Kolkata, West Bengal,700107

SIGNATURE

Prof.(Dr.)
Anindita Kundu

Project Guide,
Assistant Professor

Computer Science and Engineering

Heritage Institute of Technology
Chowbaga Road T.O, Anandapur,

East Kolkata Township,

Kolkata,West Bengal, 700107

ACKNOWLEDGEMENT

“ No one who achieves success does so without acknowledging the help of others. The wise and confident acknowledge this help with gratitude. **“ – Alfred North Whitehead**

The success of any project depends largely on the encouragement and guidelines from other people around us. Thus, we would like to take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to thank **Prof.(Dr.) Basab Chaudhuri**, Principal, Heritage Institute of Technology, Kolkata for allowing us to form a group of four people and for providing us with all the necessary facilities to make our project work and of worth.

We are grateful to **Prof.(Dr.) Anindita Kundu**, our project coordinator and project mentor as well as the Head of the Computer Science and Engineering Department, **Prof.(Dr.) Subhashis Majumder** for guiding us and giving us invaluable insights throughout the course of the project. Their guidance, unstinted support and words of encouragement motivated us to achieve our goal and provided us with impetus to excel.

We would like to take a moment to thank **our teachers**, **faculty members** and **laboratory assistants** at The Heritage Institute of Technology, Kolkata for playing a pivotal and decisive role during the development of the project. Last, but not the least, we thank **all our friends and peers at the university** for the cooperation and encouragement that they have bestowed upon us.

(Laraib Ahmed Ansari)

(Navonil Das)

(Neha Chowdhary)

(Swarnadwip Bose)

○ Table of Contents ○

Contents

INTRODUCTION

E-commerce and retail companies are leveraging the power of data and boosting sales by implementing recommendation systems on their websites. The use cases of these systems have been steadily increasing within the last years and it's a great time to dive deeper into this amazing machine learning technique.
In this blog post, you'll learn the broad types of popular recommendation systems, how they work, and how they are used by companies in the industry.
Further, we'll discuss the high level requirements for implementing recommendation systems, and how to evaluate them.

What are recommendation systems?

(A Brief Overview)

Recommendation systems aim to predict users' interests and recommend product items that quite likely are interesting for them. They are among the most powerful machine learning systems that online retailers implement in order to drive sales.
Data required for recommendation systems stems from explicit user ratings after watching a movie or listening to a song, from implicit search engine queries and purchase histories, or from other knowledge about the users/items themselves.
Sites like Spotify, YouTube or Netflix use that data in order to suggest playlists, so-called Daily mixes, or to make video recommendations, respectively.

Why do we need recommendation systems?

Companies using recommendation systems focus on increasing sales as a result of very personalized offers and an enhanced customer experience.
Recommendations typically speed up searches and make it easier for users to access content they're interested in, and surprise them with offers they would have never searched for.
What is more, companies are able to gain and retain customers by sending out emails with links to new offers that meet the recipients' interests or suggestions of films and TV shows that suit their profiles.
The user starts to feel known and understood and is more likely to buy additional products or consume more content. By knowing what a user wants, the company gains competitive advantage and the threat of losing a customer to a competitor decrease.
Providing that added value to users by including recommendations in systems and products is appealing. Furthermore, it allows companies to position ahead of their competitors and eventually increases their earnings.

How does a recommendation system work?

Recommendation systems function with two kinds of information:

- ***Characteristic information***. This is information about items (keywords, categories, etc.) and users (preferences, profiles, etc.).
- ***User-item interactions****.* This is information such as ratings, number of purchases, likes, etc.

Based on this, we can distinguish between three algorithms used in recommendation systems.

- ***Content-based* systems**, which use characteristic information.
- ***Collaborative filtering* systems**, which are based on user-item interactions.
- Hybrid systems, which combine both types of information with the aim of avoiding problems that are generated when working with just one kind.

Next, we will dig a little deeper into content-based and collaborative filtering systems and see how they are different.

Content-based systems

These systems make recommendations using a user's item and profile features. They hypothesize that if a user was interested in an item in the past, they will once again be interested in it in the future. Similar items are usually grouped based on their features. User profiles are constructed using historical interactions or by explicitly asking users about their interests. There are other systems, not considered purely content-based, which utilize user personal and social data.

One issue that arises is making obvious recommendations because of excessive specialization (user A is only interested in categories B, C, and D, and the system is not able to recommend items outside those categories, even though they could be interesting to them).

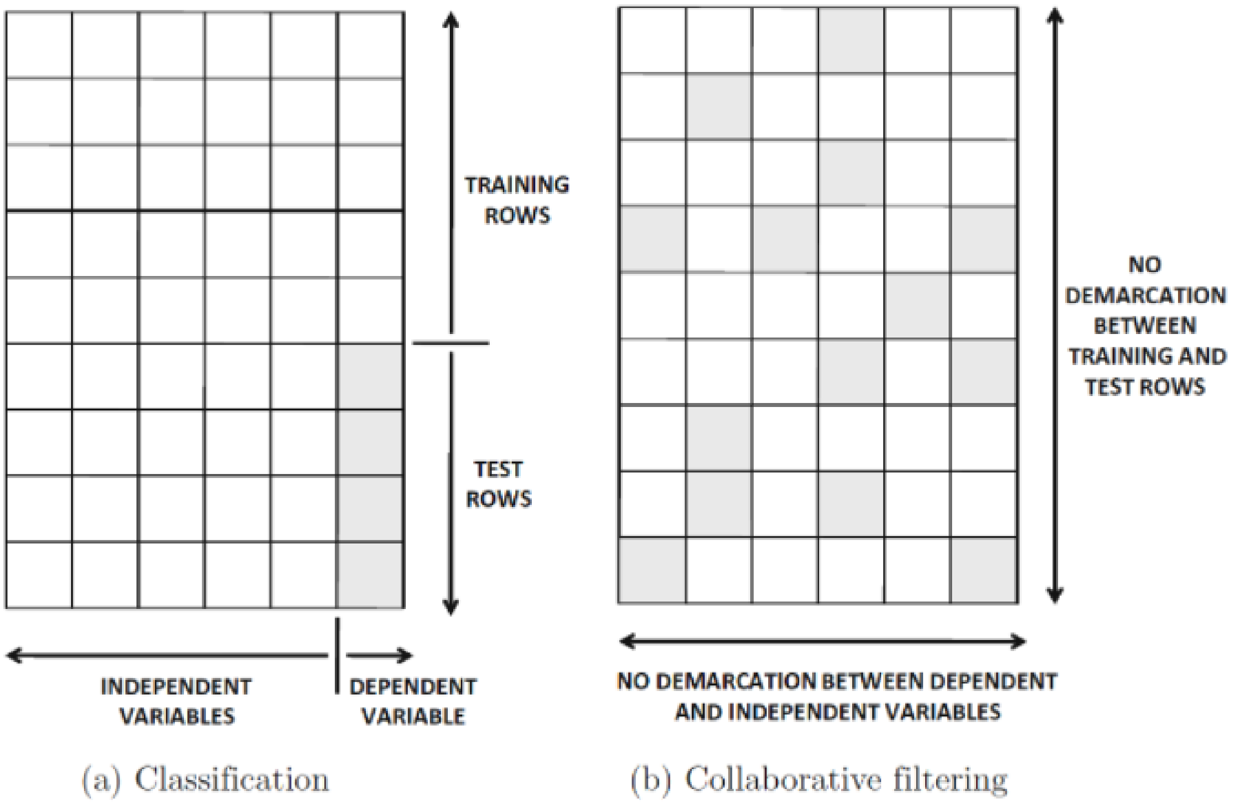
Another common problem is that new users lack a defined profile unless they are explicitly asked for information. Nevertheless, it is relatively simple to add new items to the system. We just need to ensure that we assign them a group according to their features.

Collaborative filtering systems

Collaborative filtering is currently one of the most frequently used approaches and usually provides better results than content-based recommendations. Some examples of this are found in the recommendation systems of YouTube, Netflix, and Spotify.

These kinds of systems utilize user interactions to filter for items of interest. We can visualize the set of interactions with a matrix, where each entry (i,j) (i,j) represents the interaction between user ii and item jj. An interesting way of looking at collaborative filtering is to think of it as a generalization of classification and regression. While in these cases we aim to predict a variable that directly depends on other variables (features), in collaborative filtering there is no such distinction of feature variables and class variables.

Visualizing the problem as a matrix, we don't look to predict the values of a unique column, but rather to predict the value of any given entry.



Classification vs Collaborative filtering

Image Source
https://link.springer.com/book/10.1007%2F978-3-319-29659-3

In short, collaborative filtering systems are based on the assumption that if a user likes item A and another user likes the same item A as well as another item, item B, the first user could also be interested in the second item. Hence, they aim to predict new interactions based on historical ones. There are two types of methods to achieve this goal: memory-based and model-based.

Memory-based

There are two approaches: the first one identifies clusters of users and utilizes the interactions of one specific user to predict the interactions of other similar users. The second approach identifies clusters of items that have been rated by user A and utilizes them to predict the interaction of user A with a different but similar item B. These methods usually encounter major problems with large sparse matrices, since the number of user-item interactions can be too low for generating high quality clusters.

Model-based

These methods are based on machine learning and data mining techniques. The goal is to train models to be able to make predictions. For example, we could use existing user-item interactions to train a model to predict the top-5 items that a user might like the most. One advantage of these methods is that they are able to recommend a larger number of items to a larger number of users, compared to other methods like memory-based. We say they have large coverage, even when working with large sparse matrices.

Issues with collaborative filtering systems

There are two main challenges that come up with these systems:

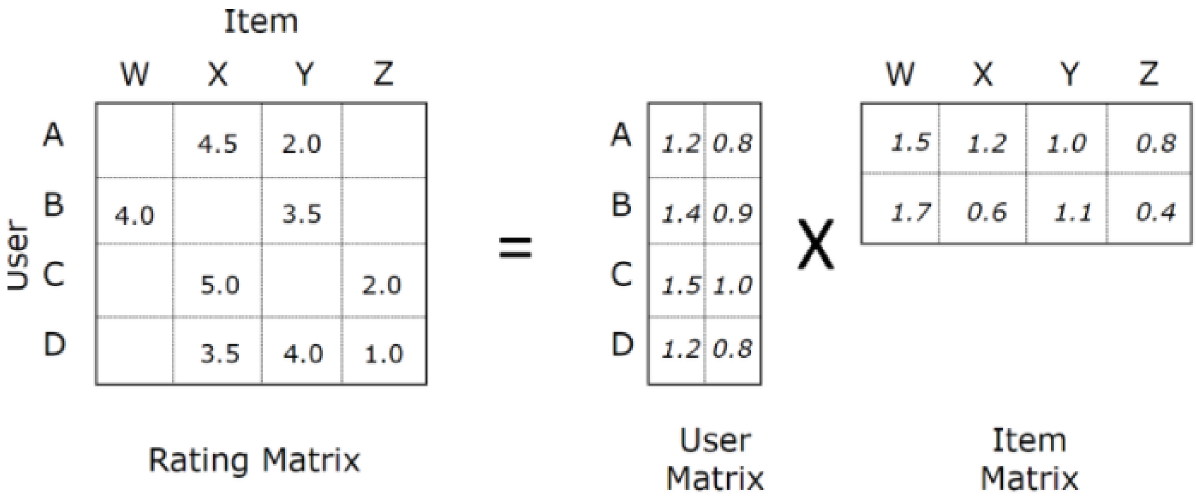
1. Cold start: we should have enough information (user-item interactions) for the system to work. If we setup a new e-commerce site, we cannot give recommendations until users have interacted with a significant number of items.
2. Adding new users/items to the system: whether it is a new user or item, we have no prior information about them since they don't have existing interactions.
3. These problems can be alleviated by asking users for other type of data at the time of sign-up (gender, age, interests, etc), and using meta information from the items in order to be able to relate them to other existing items in the database.

What technique is used to build recommendation systems?

There are two techniques for building a collaborative filtering system: fully-connected neural networks and Item2vec.

Matrix factorization

One classic approach is matrix factorization. The goal is to complete the unknowns in the matrix of user-items interactions (let's call it RR). Imagine that we somehow, magically, have two matrices UU and II, such that U \times IU×I is equal to RR in the known entries. Using the U \times IU×I product we will also have values for the unknown entries of RR, which can then be used to generate the recommendations.



Matrix factorizations

Image Source

[***https://medium.com/@connectwithghosh/simple-matrix-factorization-example-on-the-movielens-dataset-using-pyspark-9b7e3f567536***](https://medium.com/@connectwithghosh/simple-matrix-factorization-example-on-the-movielens-dataset-using-pyspark-9b7e3f567536)

A smart way to find matrices UU and II is by using a neural network.

First, we have to map each user and item to a vector with dimensions MM and NN, respectively. This means we need to learn representations of users and items, usually called embeddings (because we are embedding these concepts into a vector space). As we don't yet know the values of these vectors, we will have to start from a random initialization.

Then, for each user-item interaction (u,x)(u,x) we will concatenate both embeddings of user and item xx to give us a single vector. As we already know the value of this user-item interaction, we can force the output of the network for this vector to be such. Then, the network will use backpropagation to adjust both its own weights and the embeddings themselves, so the result matches what we expect. Thus, the network will be learning the best way to represent users and items and will be useful to predict interactions which it hasn't seen before by feeding it with the resulting embeddings.

For example, let's look at the image above and suppose that 'User Matrix' and 'Item Matrix' are our randomly initialized embeddings. For interaction (A, X)(A,X), we'll feed our neural network with the vector [1.2, 0.8, 1.2, 0.6][1.2,0.8,1.2,0.6] and force its output to equal 4.54.5. For this example, we could use MSE as the loss function. If we had a binary matrix of interactions, it would be appropriate to use a more common loss function in classification problems, like cross entropy.

A very interesting result of this approach is that the embeddings usually contain certain semantic information. Thus, we don't end up with only the predictions of unknown interactions, but we gather insights that we can make to be actionable. For examples, similar users will end up closer to each other in the user vector space. This could, for example, be useful for studying how customers behave.

What are the prerequisites for building a recommendation system?

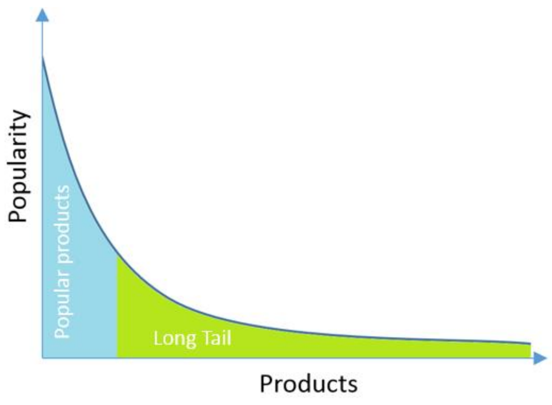
Data is the single most important asset. Essentially, you need to know some details about your users and items. If metadata is all you have available, you can start with content-based approaches. If you have a large number of user interactions, you can experiment with more powerful collaborative filtering.

The larger the data set in your possession, the better your systems will work. Furthermore, you have to be sure you're staffed with a team that is able to understand the data and manipulate it correctly to allow for it to be ingested by the techniques you'll utilize.

Some things to keep in mind regarding the user-item interactions:

You should define the interactions with respect to your system so that data can be extracted For example, if you're working on an e-commerce site, the interactions could include clicks on an item, searches, visits, favorite items, purchases, explicit ratings, elements in a shopping cart, or even discarded products, among others.

- The interactions can be defined as explicit or implicit. Explicit is characterized by situations such as when the user shows either positive or negative interest in an item, such as ranking it or leaving a review. Implicit is when the user's interest is derived from their actions, like searching for or buying an item.
- The larger the number of interactions per user and item, the better the final results will be.
- Typically, there are very popular items that users interact with a lot and others that they don't, which comprise what is known as the Long Tail. Recommendation systems usually work pretty well on popular items, although that's probably not very interesting to users as they most likely already know about them. The items in the Long Tail are the most interesting ones, because they may not be considered by the user at all if they aren't recommended.



Long tail

Image Source

http://www.dataminingapps.com/dma_research/marketing-analytics/

Why this problem?

1. Experience

To gain a first-hand experience in creating life size problems which reflect real life usage. Trip planning can undoubtedly be an intimidating task, especially when one has not done something like this before. Our project prevents people from getting bamboozled by the abundance of data available all over the internet through blogs, social media and guidebooks. Our system helps an user to carve out a satisfactory and optimized tour plan without having to break a sweat and in reasonably less time as well.

2. Improvement

To improve and optimize the already existing problems we can provide our services to a larger client-base. Our project looks to provide a solution which is free from some of the shortcomings of other already existing systems. The stand-out features that effectively serve the purpose include incorporation of real-time weather, Collaboration with local guides, and giving utmost importance to rigorously working on feedbacks.

3. Varied Usage

This system can be used in various platforms and the idea can also be implemented in other similar fields. Just by varying the domain knowledge, similar content-based filtering technology can be used in case of some other tourism objects such as cruises. In that case, instead of tourist destinations, the system would have to describe a catalogue of cruises—that is, build a catalogue using a selected set of cruise features.

Limitation of other people's work:

1. Lack of personal touch:

While most of the existing such systems are competent, one major forte they lack in is in providing a personal touch. In our recommendation system we plan to connect local tourist guides with our customer. For example, if someone is travelling to Kolkata, we will get in touch with a tour guide and connect them with the client. This will also improve employment for guides who are more local.

2. Real time Weather:

We plan to optimize our travel recommendations according to the weather. Suppose a person is travelling for four days in Sikkim then we provide them an optimized travel plan wherein weather changes don't impact their tour.

3. Monitor Feedback:

One of the prominent characteristics of the existing recommendation systems is that they take feedback from customers on a regular basis but fail to act on them quickly. We plan to change it so that we can improve our services as soon as possible.

LITERATURE REVIEW

Recommendation systems are machine learning systems that help users discover new product and services. Every time we shop online, a recommendation system guides us towards the most likely product we might purchase.

Recommendation systems are an essential feature in our digital world, as users are often overwhelmed by choice and need help finding what they are looking for. This leads to happier customers and, of course, more sales. Recommendation systems are like salesmen who know, based on our history and preferences, what we like.

Recommendation System is one of the most useful applications of machine learning. They are collection of simple algorithms (For instance, K-means Clustering Algorithm in our case) which tend to provide most relevant and accurate data as per user's requirement. Travel and Tourism domain is one of the important economic area of a nation and recommendation systems in this domain would cater to not only the tourists but also to the governments.

Travelling is a combination of journey, transportation, travel-time, accommodation, weather, events, and other aspects which are likely to be experienced by most of the people at some point in their life. To enhance such experience, we generally look for assistance in planning a tour. Today, the information available on tourism-related aspects on the Internet is boundless and exploring suitable travel package/product/service may be time-consuming. A Recommendation System (RS) can assist for various tour-related queries such as top destinations for vacation, preferable climate conditions for tracking, or the fastest way to transport.

Our project provides a tailor-made travel itinerary for users using their travel details like destination, start and end dates of travel and their preferences of mode of travelling(lavish, moderate or aggressive). Our project significantly reduces the time spent on planning for a satisfactory vacation.

Research Methodology

Extraction data from various websites

It’s not always easy to get web data into a spreadsheet for analysis or machine learning. Copying and pasting information from websites is time-consuming, error-prone and not feasible.

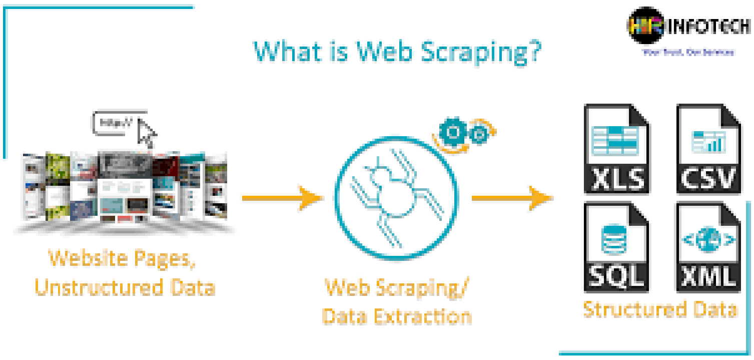
Web scraping is a way to get data from a website by sending a query to the requested page, then combing through the HTML for specific items and organizing the data. If you don’t have an engineer on hand, Import.io provides a no-coding, point and click web data extraction platform that makes it easy to get web data.

Web scraping

Web scraping is the process of using bots to extract content and data from a website.

Unlike screen scraping, which only copies pixels displayed onscreen, web scraping extracts underlying HTML code and, with it, data stored in a database. The scraper can then replicate entire website content elsewhere.

Web scraping is used in a variety of digital businesses that rely on data harvesting. Legitimate use cases include:



- Search engine bots crawling a site, analyzing its content and then ranking it.
- Price comparison sites deploying bots to auto-fetch prices and product descriptions for allied seller websites.
- Market research companies using scrapers to pull data from forums and social media (e.g., for sentiment analysis).
- Web scraping is also used for illegal purposes, including the undercutting of prices and the theft of copyrighted content. An online entity targeted by a scraper can suffer severe financial losses,

especially if it's a business strongly relying on competitive pricing models or deals in content distribution.

Forming the database

- In order to properly store data in a CSV file, you need to consider the separator. As the name says, comma-separated values — CSV is a file where the values are separated by commas. But if your data also has lots of commas, then your file will become a huge mess.
- You can then either remove every comma from your data or use a different separator. If the commas are in numbers like "1,000,000", then it might be easier to remove them since they won't be useful. But if it's the text that you're scraping, then the commas might be relevant and it's better to change the separator to a semi-colon, for instance.

DataFrame to CSV

- The first way to create a CSV file with web scraping is to use the DataFrame.to_csv() method. This is pretty straightforward and just exports a DataFrame as a CSV file.
- However, in order to export the DataFrame, you first need to have your data as a DataFrame. A simple way to achieve this is to create a big list of lists containing all the data you scraped. Each list inside this list of lists would represent a single row of the DataFrame, containing data from a single page.
- When you're done gathering data, you can just transform the list to a DataFrame and then export it as a CSV file.

```
list_of_lists = []
for page in pages:
    list_of_page = []

    '''
    Here's your scraper
    '''

    list_of_page.append(value_1)
    list_of_page.append(value_2)
    list_of_page.append(value_3)
    list_of_lists.append(list_of_page)

# Creating the DataFrame
df = pd.DataFrame(list_of_lists, columns=['v_1', 'v_2', 'v_3'])
# Exporting the DataFrame as csv
df.to_csv('quotes-list.csv', index=False, sep=';')
```

Exporting data to DataFrame to CSV

NOTE→ We have used only Dataframe in this project to process data.

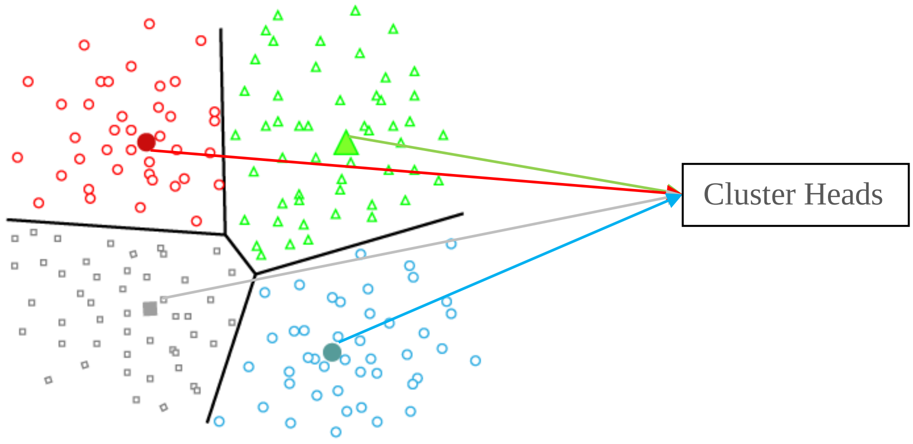
Clustering datasets according to the road distance

Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset. It's a way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group.

We've used K-Means clustering algorithm to make the clusters of data points.

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

Here, choosing the cluster no (i.e. value of K) depends on the no of days a person stays in that particular are.



Clustering Data points

Image Source

[***https://www.geeksforgeeks.org/clustering-in-machine-learning***](https://www.geeksforgeeks.org/clustering-in-machine-learning)

Proposed Algorithms

K-Means Algorithm

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.
It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:

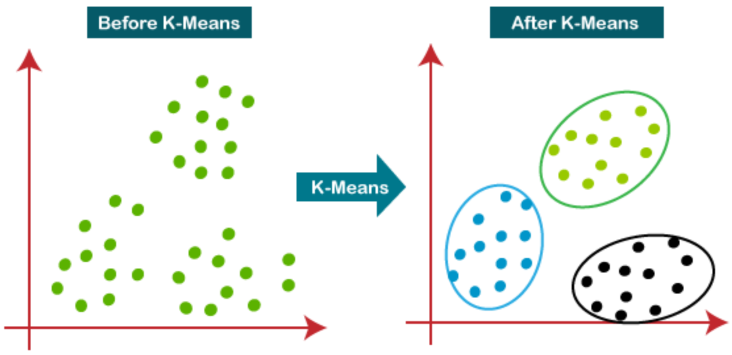


Image Source

https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each data point to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

How to choose the value of "K number of clusters" in K-means Clustering?

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways (Elbow method) to find the optimal number of clusters, but here we've chosen the cluster no (i.e. value of K) depends on the no of days a person stays in that particular are.

Note→

We have used Google API use calculate the distance not by Euclidian Method. The distance is the road distance calculated by Google API.

Google Map API to find road distance

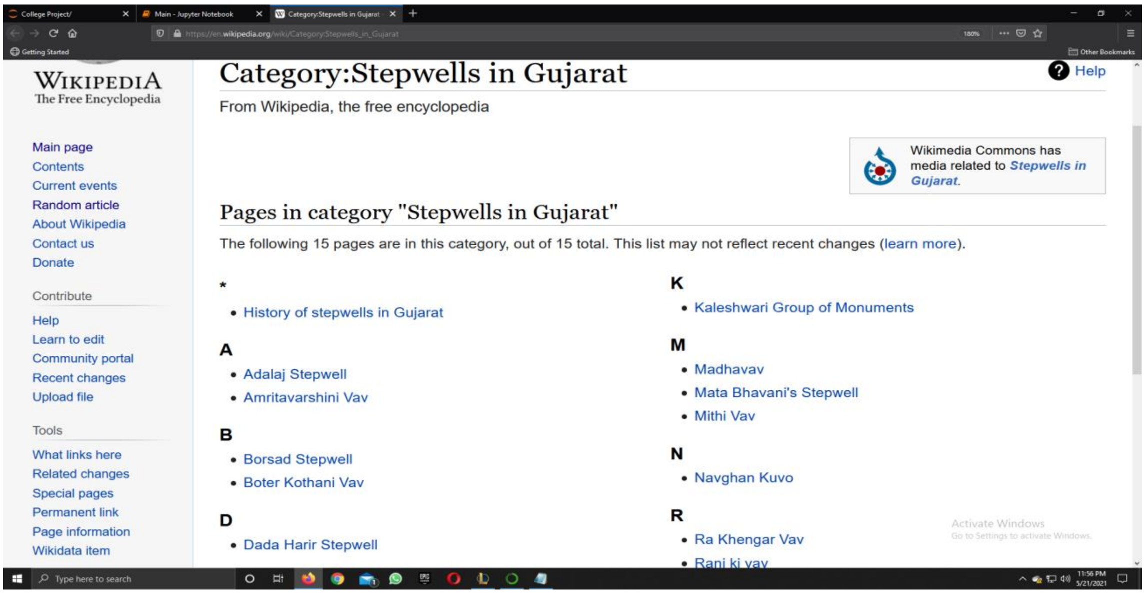
Origins-The starting point for calculating travel distance and time. You can supply one or more locations separated by the pipe character (|), in the form of a place ID, an address, or latitude/longitude coordinates

Destinations — One or more locations to use as the finishing point for calculating travel distance and time. The options for the destinations parameter are the same as for the origins parameter, described above.

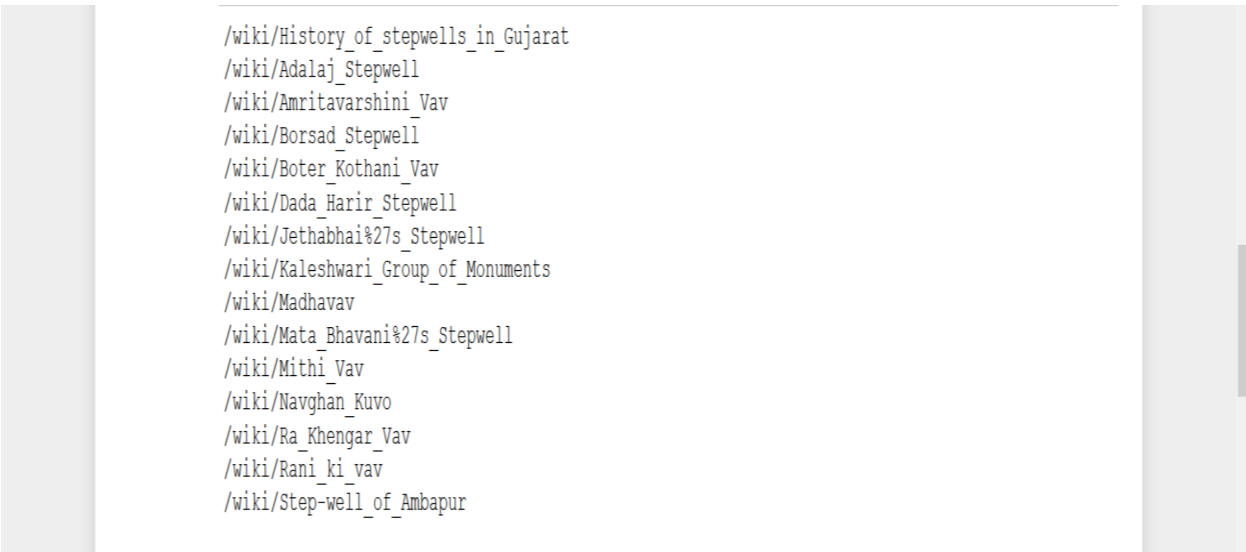
```
origins=41.43206,-81.38992|-33.86748,151.20699
```

Outcome

The information of the category of locations------



The extracted information from that page-----



Each location with their longitude and latitude-----------

```
[['Adalaj Stepwell', '23°10'01"N', '72°34'49"E', 'Archaeological Site'], ['Amritavarshini Vav', '23°01'30"N', '72°35'50"E', 'Archaeological Site'], ['Borsad Stepwell', '22°24'40"N', '72°54'02"E', 'Archaeological Site'], ['Boter Kothani Vav', '23°36'12"N', '72°24'05"E', 'Archaeological Site'], ['Dada Harir Stepwell', '23°02'25"N', '72°36'19"E', 'Archaeological Site'], ["Jethabhai's Stepwell", '22°58'28"N', '72°36'12"E', 'Archaeological Site'], ['Kaleshwari Group of Monuments', '23°19'20"N', '73°35'07"E', 'Archaeological Site'], ['Madhavav', '22°42'34"N', '71°40'28"E', 'Archaeological Site'], ["Mata Bhavani's Stepwell", '23°02'40"N', '72°36'25"E', 'Archaeological Site'], ['Mithi Vav', '24°10'27"N', '72°25'59"E', 'Archaeological Site'], ['Navghan Kuvo', '21°31'26"N', '70°28'09"E', 'Archaeological Site'], ['Ra Khengar Vav', '21°29'30"N', '70°23'00"E', 'Archaeological Site'], ['Rani ki vav', '23°51'32"N', '72°6'6"E', 'Archaeological Site'], ['Step-well of Ambapur', '23°09'07"N', '72°36'39"E', 'Archaeological Site']]
```

Several datasets-------

- Dataset_Arunachal_Pradesh
- Dataset_Assam
- Dataset_Bihar
- Dataset_Chattisgarh
- Dataset_Goa
- Dataset_Gujarat
- Dataset_Haryana
- Dataset_Jharkhand
- Dataset_Kerala
- Dataset_Madhya_Pradesh
- Dataset_Project
- Dataset_Punjab
- Dataset_West_Bengal.csv

Example of dataset Kolkata----------

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

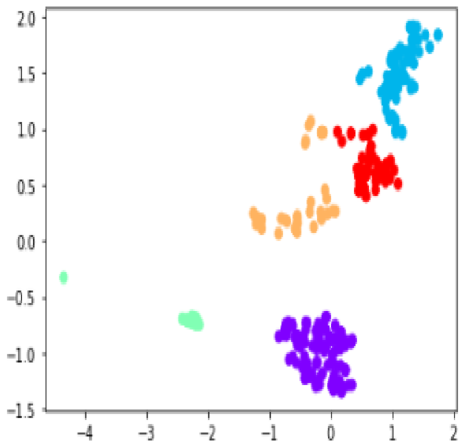
Run Code

18	Howrah Bridge	Howrah to Kolkata	22°35′06″N	88°20′49″E
19	B. B. D. Bagh	Kolkata	22°34′19″N	88°20′56″E
20	Dakshineswar Kali Temple	North 24 Parganas district (Dakshineswar)	22°39′18″N	88°21′28″E
21	Second Hooghly Bridge	Kolkata to Howrah	22°33′25″N	88°19′40″E
22	Belur Math	Howrah (Belur Math)	22°37′57″N	88°21′23″E
23	Victoria Memorial	Kolkata	22°32′42″N	88°20′33″E
24	Sunderbans delta of India	Southern 24 Parganas District	21°57′N	89°11′E
25	Digha	Purba Medinipur	21°38′18″N	87°30′35″E
26	Mandarmani	Purba Medinipur	21°39′58″N	87°42′18″E
27	St. Paul's Cathedral	Kolkata	22°32′39″N	88°20′48″E
28	Sundarbans National Park	Sunderbans delta of India	21°50′17″N	88°53′07″E
29	Mayapur	Nabadwip	23°26′18″N	88°23′34″E
30	Kumortuli	North Kolkata	22°36′00″N	88°21′41″E
31	Salt Lake Stadium	Bidhannagar	22°34′08″N	88°24′33″E
32	Howrah station	Howrah	22°34′54″N	88°20′32″E
33	Eden Gardens	Kolkata	22°33′52″N	88°20′36″E

Clustering Data points-----------

```
In [17]: plt.scatter(df_feat['Latitude'],df_feat['Longitude'],c=kmeans.labels_,cmap='rainbow')

Out[17]: <matplotlib.collections.PathCollection at 0x1864bf0feb0>
```



References

- Dataset creation and cleaning : Web Scraping using Python

https://towardsdatascience.com/dataset-creation-and-cleaning-web-scraping-using-python-part-1-33afbf360b6b

- How to get Latitude and Longitude with Python

https://stackoverflow.com/questions/25888396/how-to-get-latitude-longitude-with-python

- How to Scrape Multiple Pages of a Website Using a Python Web Scraper

https://betterprogramming.pub/how-to-scrape-multiple-pages-of-a-website-using-a-python-web-scraper-4e2c641cff8

- List of tourist attractions in Kolkata

https://en.wikipedia.org/wiki/List_of_tourist_attractions_in_Kolkata

- An Introduction to Clustering and Different Methods of Clustering

https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/

- A Tutorial on Clustering Algorithms

https://matteucci.faculty.polimi.it/Clustering/tutorial_html/index.html

- Data Science K-means Clustering : In-depth Tutorial

https://data-flair.training/blogs/k-means-clustering-tutorial/

- Other links
  - https://builtin.com/data-science/recommendation-systems
  - https://medium.com/analytics-vidhya/understanding-recommendation-systems-introduction-3be54e937625
  - https://www.researchgate.net/publication/332151806_A_Survey_of_Travel_Recommendation_System
  - http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/mvoget/cluster/cluster.html

THANK YOU