

Mini-Project(2020-21)

Mid-Term Report

ON

Tourist Analysis Using Big Data (Tourist Place Recommendations Dataset)



Institute of Engineering and Technology

**Submitted by-
Abhishank Pandey
(01)**

**Supervised by-
Mrs. Harvinder Kaur
Technical Trainer
Department of Computer Engineering & Applications**

Content

	Page No.
Abstract	3
1. Introduction	4
1.1 General introduction to the topic	4
1.2 Analysis of Project	5
1.3 Hardware & Software Requirements	6
2. Objectives	7
3. Implementation details	8-9
4 Area of Computer Science & Explanation	10-12
5. Screenshots & Code Implementation	13-29
6. References	30

Abstract

The recent past showed a greater interest in recommender techniques. Now-adays there are many tourism and travel packages existing from different websites to almost all the places over the world. A customer finds it very difficult to search for the best package as he/she has to browse multiple websites, contact many travel agents and etc. which is a tedious process and is time consuming. There should be a system where the user should find the best package on the Internet with a single click.

To address this issue, we adopt Travel Package Recommendation System which offers the best package among all the other packages that are on the web. This project will help tourist to suggest the best Travel Package among all the package deals on the web. On multiple demands of tourist that is, a customer will select a travel package for a particular place based on the recommendations provided by the previous customers who had experience with the package. Therefore, according to the personalized recommendations, he/she will choose the best package that is on the web. Thus, The proposed system maintains a centralized repository to make necessary travel arrangements and to retrieve information easily.

Initially, we will evaluate the particular characteristics of the current traveling packages and we mine the data on the tourists rating and the intrinsic features i.e., locations, travel seasons etc. Based on the data collected after mining, we will generate a list for personalized travel package recommendations. Furthermore, we will extract the data based on the tourist's relation with the area and season.

1.Introduction(1.1)

Tourism can be considered as most favorite pass time when people get free time. Several travel organizations are available on the web. The people or the tourist select their own Travel Package according to their personal interest. The travel companies concentrate on the interest associated with tourist making sure to increase their particular market value and supply enormous package deals. So that they can make their Travel Package more effective. Now-a-days Recommender system is becoming very famous and people are getting attracted to it, as it is helping them to choose the best package in a short time.

Recommender systems are categorized into

1. Content based system: With this, item recommendation is analyzed then it retrieves the information and filters this for research. For example, if the tourist goes to hill stations more often, then database contains “hill station” as recommendation
2. Collaborative filtering systems: It rely on the similar factors of user or items. Preferences of different users for same item are recommended by system.

There are many challenges in designing and executing Personalized Tourist Travel Package Recommendation System. The following shows some of the challenges:

1. The data for Travel is very less and scattered. For an example, recommendation for a movie may cost more to travel than the movie price.
2. Usually Travel package are location based so they are pertained to space or time to reach destination. For an example, the package contains locations which are geographically near and also vary season wise.
3. The older recommendation method is dependent upon rating and the travel data may not consist of this sort of rating.

1.2 Analysis Of Project

Recommender systems can be classified into two categories: Content - based filtering and collaborative filtering. Content-based filtering analyzes the association between user problems and the descriptions of items. To recommend new items to a user, the content-based filtering approach matches the new items descriptions to those items known to be of interest to the user. On the other hand, the collaborative filtering (CF) approach does not need content information to make recommendations. Users of social networking services can connect with each other by forming communities for online interaction. Yet as the number of communities hosted by such websites grows over time, 5 users have even greater need for effective community recommendations in order to meet more users.

A Mixed Recommendation approach:

For personalized travel package recommendation, the data is displayed for location and price. As the tourist selects the package based on location or price, the data is updated in the database and the admin or the users can view the count of people who has selected a package based on a particular location or a particular price. It is important for the user, as it helps the users in deciding the location based on the popularity of the location and also it helps when pricing plays a major role in deciding a package. It will give admin a perspective about the Travel packages which was added as it will be useful for the admin to add future packages based on the popularity of the location and the price the users have selected.

1.3 Hardware And Software Requirements

Hardware Used:

1. A portable PC or a working laptop.
2. A minimum RAM of 8GB or 32 GB hard disk drive.
3. Intel Core-i5 Processor, 8th Generation(minimum)

Software Used:

1. HADOOP/Python frameworks environment system that act as a tool for analysis & recommendations to tourist(if needed).
2. Technology: Java
3. Web technology: HTML, Javascript and CSS.
4. Database used : MySQL5.0

2.Objective

Tourism is the world's largest industry and one of the objectives of tourism development is economic gain. ... In economic terms the travel and tourism industry is able to do three key things:

- 1- It gives vigour to economies.
- 2 - It offers people jobs and career prospects.
- 3 - It stimulates development.

The purpose of the travel and tourism management system project is to create a system that automates travel processes and events and to build a system that enables all travel related operations to be carried out.

With this project, I am developing a tourism management system first by creating a user login and password form with the help of java language and some of the python frameworks which will be used for developing the code and then , giving recommendations on famous tourist places to visit in a particular area or throughout our country(India) based on user's search history on the system. It will be a simple static web application type system with not that much of lookend feel.

This application is develop to provide best travelling services to the customers and travel agents. We have developed tours and travel management system to provide a search platform where a tourist can find their tour places according to their choices. The system also helps to promote responsible and interesting tourism so that people can enjoy their holidays at their favorable places. This system also helps to develop tourism with different cultures so that they enrich the tourism experience and build pride. This system also provide a better way to connect with various events.

This system also gives tours related information like which places are tourist attractions, cities, and provinces. Tourist can also get the Map and navigation system and temperature and weather information. Tourist can also book tours

through our tours and travels management system. This system also keeps a history of visited places of its users.

3. Implementation of Project

In this project, there are two types of users, one is Admin, and another is user. An Admin logs into his account, and his role is to add, edit, and delete packages. And also can provide recommendations accordingly. A user logs in and provides personalized inputs (tourist, area, and season) to the system and the best package that is available on web which will be presented to the user.

In this, the proposed system is designed to be more efficient than the manual system. It invokes all base tasks that are now carried out manually, such as the forms transactions and reports which is added advantage. The proposed System is completely computer-based application. Thousands of records can searched and displayed without taking any significant time.

Advantages of the Proposed System:

- Gives accurate information.
- Simplifies the manual work.
- It minimizes the documentation related work.
- Provides up to date information.
 - Friendly Environment by providing warning messages.
- travelers details can be provided.
- booking confirmation notification.

Below are the modules required to develop this project and create an application user form:

1. Administrator authentication/module: This module is mainly based on admin. System will check the admin user name and password for authentication. After the verification for authorization the admin can be able to precede the process. All works are done under his control. From this module admin can view daily, weekly and monthly report. This module is develop for admin of the website and admin can add, delete, edit and view the data related to places, travels, routes, bookings from this module.
2. User Registration: This module covers the details about the registration of users which they can be register by itself by adding data like name,

password, email id and further details. After registration they can be sign in by their username and password.

3. Package Modules: User can view different tour packages available for tourist. User can select any packages from this module, he can also check the details of various travel agencies. A user can select any travel agency from this module.
4. Testimonials module: This is the module where passenger can post feedback after the journey and they can share their experience. Users of this application can post their opinions, complaints and suggestions regarding photos & videos.
5. Payment & Search Module:
 - a. Pay payment through paypal.
 - b. Pay payment through draft, credit & debit cards, UPI and netbanking.
 - c. Search city wise hotels, flights, packages, bus, rails and events.
6. Routes module; This will display the route information of source location and destination location. User can also check the best routes for his destination. From this module user can also get information related to various routes connecting sources and destinations, For each route, information such as source, destination, fare, reservation details, pick up points etc. are provided.
7. Reservations module: This module is for passengers/customers where passenger can reserve the seats by making payment. Using this module, user can book bus ticket or train ticket. From this module, user can also book tickets or cancel previously booked ticket. The module maintains the details of all reservations made so far and allows administrator to either confirm or reject the bookings. Accordingly, the administrator can take various steps to act on the complaints and suggestions.

4.Area(Scope) of Tourism in Computer Science

Tourism is considered to be an important aspect of economic growth and the development of a nation. According to the United Nations World Tourism Organization (UNWTO), global tourism is expected to reach 1.6 billion (in terms of international arrivals) by the year 2020. Tourism management is generally considered a bright and potential employment sector as it offers a wide variety of career opportunities in both the private and public sector.

Tourism management and tourism based analysis is a multidisciplinary field that includes all activities related to the tourism and hospitality industries. It prepares candidates with the experience and training required to hold managerial positions in food, accommodation and tourism industry. The three major areas of tourism management are:

- Business administration (finance, human resources and marketing activities).
- Management theories and principles.
- Tourism industry (travel accommodation, tourist(travel) places recommendations , environmental factors and tourism organizations).

Tourism management implements marketing efforts in attracting tourists to travel to particular destinations. This involves the management of a variety of activities such as:

- Studying tour destination;
- Planning the tour;
- Making travel arrangements;
- Providing accommodation.

Tourism analysis is not a new concept. Since tourism is an important economic driver for many communities, most tourism focused cities invest significant resources to collect visitor data to inform funding decisions for tourism development initiatives. The next generation of tourism market analysis uses consumer analytics based on either transaction data or mobile device data to go deeper. It gives cities, convention and visitor's bureau(CVB), and destination

marketing organizations(DMO) a more comprehensive understanding of tourism market dynamics.

Explanation/System Specifications

Content

System Requirement Specifications

1. Java 1.8+ : JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop java applications and applets. It physically exists. It contains JRE + development tools. JDK is an implementation of any one of the below given Java Platforms released by Oracle corporation: Standard Edition Java Platform, Enterprise Edition Java Platform, Micro Edition Java Platform. The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) etc. to complete the development of a Java Application.
2. MySQL database 5.0 or above: MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

Relational Database Schema: The data is stored in a database that is developed in MySQL. The database consists of two tables. Data of a tourist means details such as ID, name, from location, to location, no. of days and the price of tour package with ID as the primary key which is placed in a table tourpackage. The user table has the data like username and password for login that enables the admin to manipulate the information regarding each tourist.

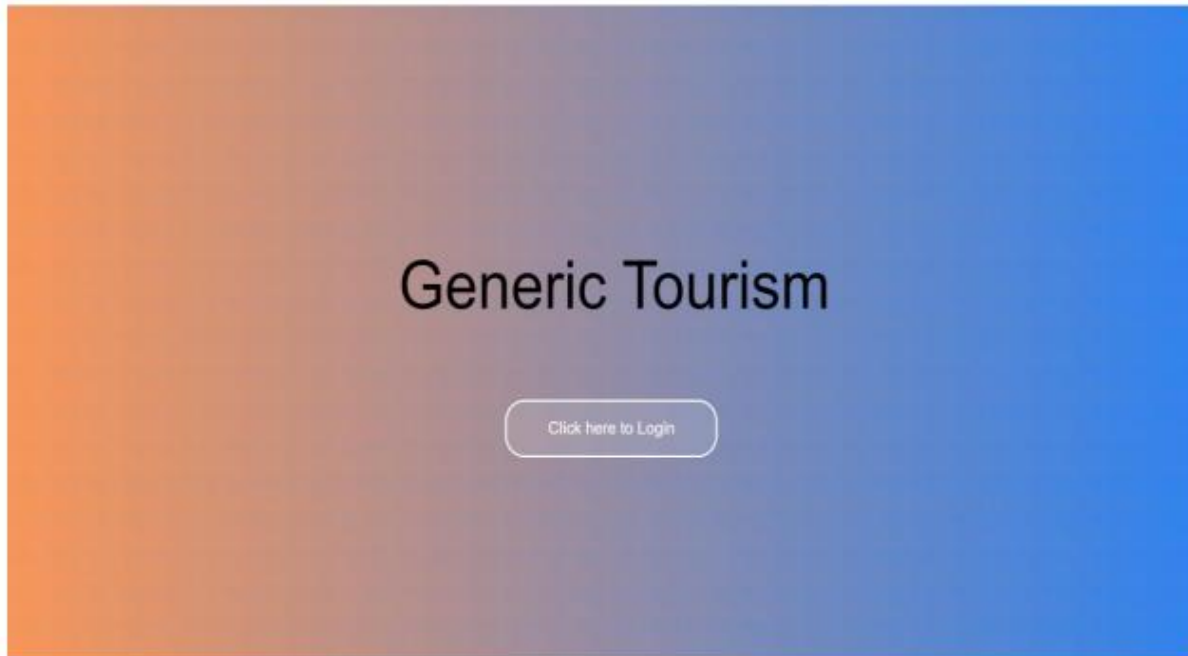
- Datasets used:
 1. data_collaborative: this has the userId, the itemId, the rating and the date that rating was given.
 2. data_content: this has the itemId, title, category, p_rating(price_rating) etc.

Dataset Extraction

- Web Scraping: a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file.
- Scrapy: Python framework used to efficiently extract data from websites.
- Extracted the data of best tourist places in Jaipur from <https://www.trawell.in/rajasthan/jaipur/places-to-visit-things-to-do>.
- Dataset contains title, category, distance(from railway station), duration, nearby places, ratings etc.

5.Screenshots

1) Home page



2) Login page

Welcome

Username:

Password:


Login

3) Tourists List

Add New Entry Search Logout

ID	NAME	FROM-LOCATION	TO-LOCATION	NO-OF-DAYS	PRICE		
1	Tour 1	Nevada	Birmingham	4	1255.65	UPDATE	DELETE

4) Add Form



A form for saving travel details, set against a blue gradient background. The form is a light blue rectangle containing five input fields with labels: 'Name', 'From-Location', 'To-Location', 'No: of Days', and 'Price'. Below these fields is a green 'Save' button. Below the form is an orange 'Display List' button.

Name

From-Location

To-Location


No: of Days

Price

Save

Display List

5) Search By Location



A form titled 'Search by Locations' on a purple gradient background. The form is a light purple rectangle containing two input fields with labels: 'From-Location' and 'To-Location'. Below these fields is a green 'Search' button.

Search by Locations

From-Location

To-Location

Search

1	Name	Age	Sex	Category	Places
2	Brown, Mia	30	Female	Wildlife	Jaipur Zoo
3	LaRotonda, William	34	Male	Heritage	Nahargarh Fort, Amber Fort
4	Steans, Tyrone	31	Male	Heritage	Amer Fort / Amber Fort, Jal Mahal
5	Howard, Estelle	32	Female	Pilgrimage	Birla Mandir, Jama Masjid
6	Singh, Nan	30	Female	Heritage	Hawa Mahal
7	Smith, Leigh Ann	30	Female	Heritage	Maharani Ki Chhatri, Hawa Mahal
8	LeBlanc, Brandon R	33	Male	Pilgrimage	Moti Dungri Ganesh Temple, Akshardham
9	Quinn, Sean	33	Male	Park	Kanak Vrindavan Garden
10	Boutwell, Bonalyn	31	Female	Heritage	City Palace / Sawai Man Singh II Museum
11	Foster-Baker, Amy	39	Female	Pilgrimage	Akshardham Temple
12	King, Janet	63	Female	Heritage	Panna Meena Ka Kund
13	Zamora, Jennifer	38	Female	Heritage	Jantar Mantar
14	Becker, Renee	32	Female	Pilgrimage	Khoie Ke Hanuman Ji Temple, Moti Dungri
15	Goble, Talisha	46	Female	Heritage, Museum	Albert Hall Museum
16	Hernandez, Danliff	31	Male	Pilgrimage	Garh Ganesh Temple
17	Horton, Jayne	34	Female	Heritage	Jal Mahal
18	Johnson, Noelle	31	Female	Heritage, Pilgrimage	Jama Masjid / Akbari Mosque - Amer
19	Murray, Thomas	29	Male	Heritage	Rambagh Palace
20	Pearson, Randall	33	Male	Park	Central Park
21	Petrowsky, Theima	33	Female	Heritage	Jaigarh Fort
22	Roby, Lori	36	Female	Park	Jawahar Circle Garden

1	userid	itemId	rating	timestamp
2	1	1		881250949
3	1	2	3	891717742
4	1	3	1	878887116
5	1	4		880606923
6	1	5	1	886397596
7	1	6		884182806
8	1	7		881171488
9	1	8	5	891628467
10	1	9	3	886324817
11	1	10		883603013
12	1	11	2	879372434
13	1	12		879781125
14	1	13	5	876042340
15	1	14	3	891035994
16	1	15		888104457
17	1	16	3	879485318
18	1	17		879270459
19	1	18	2	879539794
20	1	19		874834944
21	1	20	2	892079237

1	category	distance	duration	nearby_places
2	Wildlife	6 Kms	1-2 Hours	None
3	Heritage	6 Kms	1-2 Hours	None
4	Heritage	13 Kms	2-3 Hours	Jaigarh Fort (1 km by walk), Srijagat Siromanjiji Temple (500m), Panna Meena ka Kund (1 km) & Jama Masjid / Akbari Mosque - Amer (1 km) are situated near Amer Fort
5	Pilgrimage	6 Kms	1-2 Hours	None
6	Heritage	6 Kms	30 Mins	None
7	Heritage	9 Kms	30 Mins	None
8	Pilgrimage	6 Kms	30 Mins	None
9	Park	11 Kms	1 Hour	None
10	Heritage	5.5 Kms	1-2 Hours	Hawa Mahal (One the edge of City Palace), Jantar Mantar (adjacent to City Palace), Govind Devji Temple (Inside City Palace Complex), Sargasuli Tower / Isar Lat (Near Tripolia Gate of City Palace)
11	Pilgrimage	6.5 Kms	1-2 Hours	None
12	Heritage	14 Kms	30 Mins	None
13	Heritage	5.5 Kms	1 Hour	This is UNESCO World Heritage Site
14	Pilgrimage	11 Kms	1-2 Hours	None
15	Heritage, Museum	6 Kms	1-2 Hours	None
16	Pilgrimage	7 Kms	30 Mins	None
17	Heritage	11 Kms	1 Hour	Maharani ki Chhatri (1 km from Jal Mahal), Kanak Vrindavan Garden (1 km from Jal Mahal) can be visited along with Jal Mahal.
18	Heritage, Pilgrimage	13.5 Kms	30 Mins	None
19	Heritage	6 Kms	1 Hour	This is currently a private hotel, but visitors are allowed to take a tour of the palace.
20	Park	4 Kms	1 Hour	None
21	Heritage	15 Kms	1-2 Hours	This is usually visited from Amer Fort (1 km by walk)
22	Park	13 Kms	1-2 Hours	None

	title	url	p_rating	count	itemid
	Jaipur Zoo	/rajasthan/jaipur/jaipur-zoo	2.2	158	1
	Nahargarh Fort	/rajasthan/jaipur/nahargarh-fort	4.6	652	2
(1 km) are situated near Amer Fort and can be visited together.	Amer Fort / Amber Fort	/rajasthan/jaipur/amer-fort-amber-fort	5	783	3
	Birla Mandir	/rajasthan/jaipur/birla-mandir	4.9	482	4
	Hawa Mahal	/rajasthan/jaipur/hawa-mahal	4.2	890	5
	Maharani Ki Chhatri	/rajasthan/jaipur/maharani-ki-chhatri	1.9	656	6
	Moti Dungri Ganesh Temple	/rajasthan/jaipur/moti-dungri-ganesh-temple	4.5	350	7
	Kanak Vrindavan Garden	/rajasthan/jaipur/kanak-vrindavan-garden	2	700	8
, Sargasuli Tower / Isar Lat (Near Tripolia Gate of City Palace) can be visited along with City Palace.	City Palace / Sawai Man Singh II Museum	/rajasthan/jaipur/city-palace-sawai-man-singh-ii-museum	3.2	25	9
	Akshardham Temple	/rajasthan/jaipur/akshardham-temple	5	345	10
	Panna Meena Ka Kund	/rajasthan/jaipur/panna-meena-ka-kund	4.8	120	11
	Jantar Mantar	/rajasthan/jaipur/jantar-mantar	1.5	456	12
	Khole Ke Hanuman Ji Temple	/rajasthan/jaipur/khole-ke-hanuman-ji-temple	3.5	990	13
	Albert Hall Museum	/rajasthan/jaipur/albert-hall-museum	4	222	14
	Garh Ganesh Temple	/rajasthan/jaipur/garh-ganesh-temple	1.5	43	15
	Jal Mahal	/rajasthan/jaipur/jal-mahal	4.2	678	16
	Jama Masjid / Akbari Mosque - Amer	/rajasthan/jaipur/jama-masjid-akbari-mosque-amer	2	789	17
	Rambagh Palace	/rajasthan/jaipur/rambagh-palace	3.2	567	18
	Central Park	/rajasthan/jaipur/central-park	3.8	245	19
	Jaigarh Fort	/rajasthan/jaipur/jaigarh-fort	4.5	890	20
	Jawahar Circle Garden	/rajasthan/jaipur/jawahar-circle-garden	4.2	13	21
	Gatore ki Chhatriyan	/rajasthan/jaipur/maharaja-ki-chhatri-ga'tor	1	123	22

5.a Code Implementation

```
import pandas as pd
import numpy as np
import re, math
from collections import Counter
from googlemaps import convert
from googlemaps import Client
from googlemaps.convert import as_list
WORD = re.compile(r'\w+')
def get_cosine(vec1,vec2):
    intersection = set(vec1.keys()) & set(vec2.keys())
    numerator = sum([vec1[x] * vec2[x] for x in intersection])
    sum1 = sum([vec1[x]**2 for x in vec1.keys()])
    sum2 = sum([vec2[x]**2 for x in vec2.keys()])
    denominator = math.sqrt(sum1) * math.sqrt(sum2)
    if not denominator:
        return 0.0
    else:
        return float(numerator) / denominator
def text_to_vector(text):
    words = WORD.findall(text)
    return Counter(words)

def clean_data(x):
    if
    is
    in
    st
    a
    n
    c
    e(
    x,
    li
    st
```

```

):
    return [str.lower(i.replace(" ", "")) for i in
x]
else:
    if isinstance(x, str):
        return str.lower(x.replace(" ", ""))
    else:
        return "

metadata =
pd.read_csv('
/home/sukrat
i/Desktop/B
TP/Code/dat
a4.csv',
low_memor
y=False)

print("Select your preferred category:\n1.wildlife
\n2.heritage \n3.pilgirmage\n4.park\n5.museum")
text1 = input("Enter User Interests: ")
vector1 = text_to_vector(text1)
C = metadata['p_rating'].mean()
m = metadata['count'].quantile(0.75)

def weighted_rating(x,m=m,C=C):

    v
    =
    x
    ['
    c
    o
    u
    n
    t'
    ]

    R = x['p_rating']
    return (v/(v+m) * R) + (m/(m+v) * C)
    metad
    ata['c
    ategor

```

```

y'] =
metad
ata['c
ategor
y'].ap
ply(cl
ean_d
ata)

metadata['score'] =
metadata.apply(weighted_rating,
axis=1)
cos=[]
for i in list(metadata['category']):

text2 = i
    vector2 = text_to_vector(text2)
    cosine = get_cosine(vector1, vector2)
    cos.append(cosine)
metadata['cosine']=cos
x=metadata['cosine']>0.0
rec=pd.DataFrame(metadata[x])
rec=rec.sort_values('score',ascending=F
alse)
src=input("Enter your location: ")
dest=list(rec['title'])
def
dista
nce_
matr
ix(cl
ient,
origi
ns,
desti
nati
ons,

mode=None,
language=None, avoid=None,
units=None,

```

```

departure_time=None,
arrival_time=None,
transit_mode=None,

transit_routing_preference=None, traffic_model=None,
region=None):
    params = {
        "origins":
convert.location_list(origins),
        "destinations":
convert.location_list(destinations)
    }
    if mode:

        if mode not in ["driving",
"walking", "bicycling",
"transit"]:
            raise
ValueError("Invalid travel
mode.")
        params["mode"] = mode
        if language:
            params["language"] =
language
        if avoid:
            if avoid not in ["tolls",
"highways", "ferries"]:
                raise
ValueError("Invalid route
restriction.")
            params["avoid"] = avoid
        if units:
            params["units"] = units
        if departure_time:
            params["departure_time"]
= convert.time(departure_time)
        if arrival_time:
            params["arrival_time"] =

```

```

convert.time(arrival_time)
    if departure_time and
arrival_time:
        raise ValueError("Should
not specify both
departure_time and"
                        "arrival_time.")
    if transit_mode:
        params["transit_mode"] =
convert.join_list("|",
transit_mode)
    if
transit_routing_preference:

params["transit_routing_prefer
ence"] =
transit_routing_preference
    if traffic_model:
        params["traffic_model"]
= traffic_model
    if region:
        params["region"] = region

    return
client._request("/maps/api/dist
ancematrix/json", params)
client =
Client(key='AIzaSyA8wq3R8
WASxgUqTvWCh5blEmGzU
8njVZ0')
dist=[]
dur=[]
for d in dest:
    d=d+",Jaipur"

output=distance_matrix(client,
src,d)

```

```
a1=(output['rows'][0]['elements'][0]['distance']['text'])
```

```
a2=(output['rows'][0]['elements'][0]['duration']['text'])
    dist.append(a1)
    dur.append(a2)
rec['distance']=dist
rec['duration']=dur
final=pd.DataFrame(rec,index
=None,columns=['title','category','score','distance','duration'])
print(final)
```

```
import numpy as np
```

```
import pandas as pd
```

```
data
=pd.
read
_csv
('data4_1
.csv'
)
```

```
placeInfo=pd.read_csv('data4.csv')
data=pd.merge(data,placeInfo,left_on='itemId',right_on='itemId')
userIds=data.userId
userIds2=data[['userId']]
data.loc[0:10,['userId']]
```



```
np.nanmean(user2)
```

```
commonItemIds  
=[i for i in  
range(len(user1)  
) if user1[i]>0  
and user2[i]>0]  
    if  
len(commonItemIds)==0:  
        return 0  
    else:
```

```
user1=np.array(  
[user1[i] for i in  
commonItemIds  
)
```

```
user2=np.array(  
[user2[i] for i in  
commonItemIds  
)
```

```
    return  
correlation(user1,user2)  
except  
ZeroDivisionError:  
    or:
```

```
        print("You  
can't divide by  
zero!")
```

```
def  
nearestNeighborRatings(active  
User,K):  
    try:
```

```
similarityMatrix  
=pd.DataFrame(
```

```
index=userItem  
RatingMatrix.in  
dex,columns=['  
Similarity'])  
    for i in  
userItemRating  
Matrix.index:
```

```
similarityMatrix  
.loc[i]=similarit  
y(userItemRatin  
gMatrix.loc[acti  
veUser],userIte  
mRatingMatrix.  
loc[i])
```

```
similarityMatrix  
=pd.DataFrame.  
sort_values(simi  
larityMatrix,['Si  
milarity'],ascend  
ing=[0])
```

```
nearestNeighbo  
urs=similarityM  
atrix[:K]
```

```
neighbourItemR  
atings=userItem  
RatingMatrix.lo  
c[nearestNeighb  
ours.index]
```

```
predictItemRati  
ng=pd.DataFra  
me(index=userIt  
emRatingMatrix  
.columns,  
columns=['Ratin  
g'])
```

```

        for i in
userItemRating
Matrix.columns:

predictedRating
=np.nanmean(us
erItemRatingMa
trix.loc[activeU
ser])
        for j in
neighbourItemR
atings.index:
            if
userItemRating
Matrix.loc[j,i]>
0:

predictedRating
+=
(userItemRating
Matrix.loc[j,i]-
np.nanmean(use
rItemRatingMat
rix.loc[j]))*near
estNeighbours.l
oc[j,'Similarity']

predictItemRati
ng.loc[i,'Rating'
]=predictedRati
ng
        except
ZeroDivisionErr
or:
            print("You
can't divide by
zero!")
        return
predictItemRati
ng

```

```

def
topNRecommen
dations(activeU
ser,N):
    try:

        predictItemRati
ng=nearestNeig
hbourRatings(ac
tiveUser,10)

        placeAlreadyW
atched=list(user
ItemRatingMatr
ix.loc[activeUse
r]

        .loc[userItemRa
tingMatrix.loc[a
ctiveUser]>0].in
dex)

        predictItemRati
ng=predictItem
Rating.drop(pla
ceAlreadyWatc
hed)

        topRecommend
ations=pd.DataF
rame.sort_value
s(predictItemRa
ting,

        ['Rating'],ascend
ing=[0])[:N]

        topRecommend
ationTitles=(pla
ceInfo.loc[place

```

```
Info.itemId.isin(
topRecommend
ations.index]])
    except
ZeroDivisionErr
or:
    print("You
can't divide by
zero!")
    return
list(topRecomm
endationTitles.ti
tle)
activeUser=int(i
nput("Enter
userid: "))
print("The
recommended
places for you
are: ")
print(topNReco
mmendations(ac
tiveUser,4))
```

6. References

1. www.myproject.com

2. www.google.com

3. w3schools.com

4. www.wikipedia.org