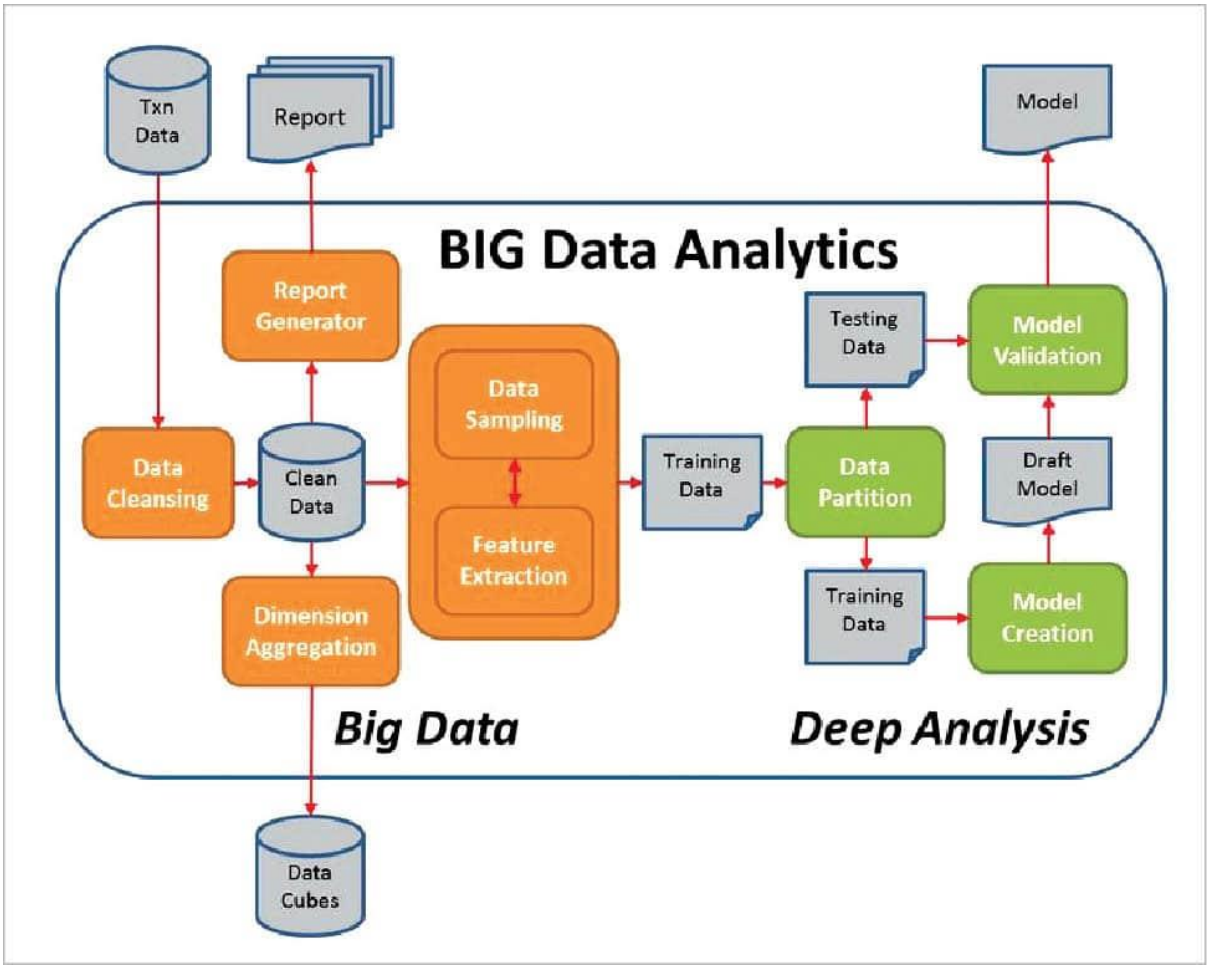
ANALYSING TOURIST BEHAVIOUR USING BIG DATA TECHNOLOGY

Objective:

Big data from social media platforms has made it possible to provide decision-makers with a wealth of new information. Nonetheless, not many big data analytics research have shown that strategic decision-making is supported. Furthermore, there is still a need for a rigorous methodology to analyse huge data created by social media for decision assistance, especially in the travel and tourist industry. This project is to develop and assess a "big data analytics" technique to assist strategic decision-making in tourism destination management using a design science research methodology. Melbourne, Australia is used as a representative case to demonstrate the applicability of the method in helping destination management organisations analyse and predict tourist behavioural patterns at specific destinations through the use of geotagged photos uploaded by tourists to the photo-sharing social media site, Flickr. Utility was proven utilising both another destination and directly with stakeholder audiences. Within a genuine issue area, the built object illustrates a method for studying unstructured big data to improve strategic decision making. The suggested approach is general, and it is addressed if it may be applied to other massive data streams.

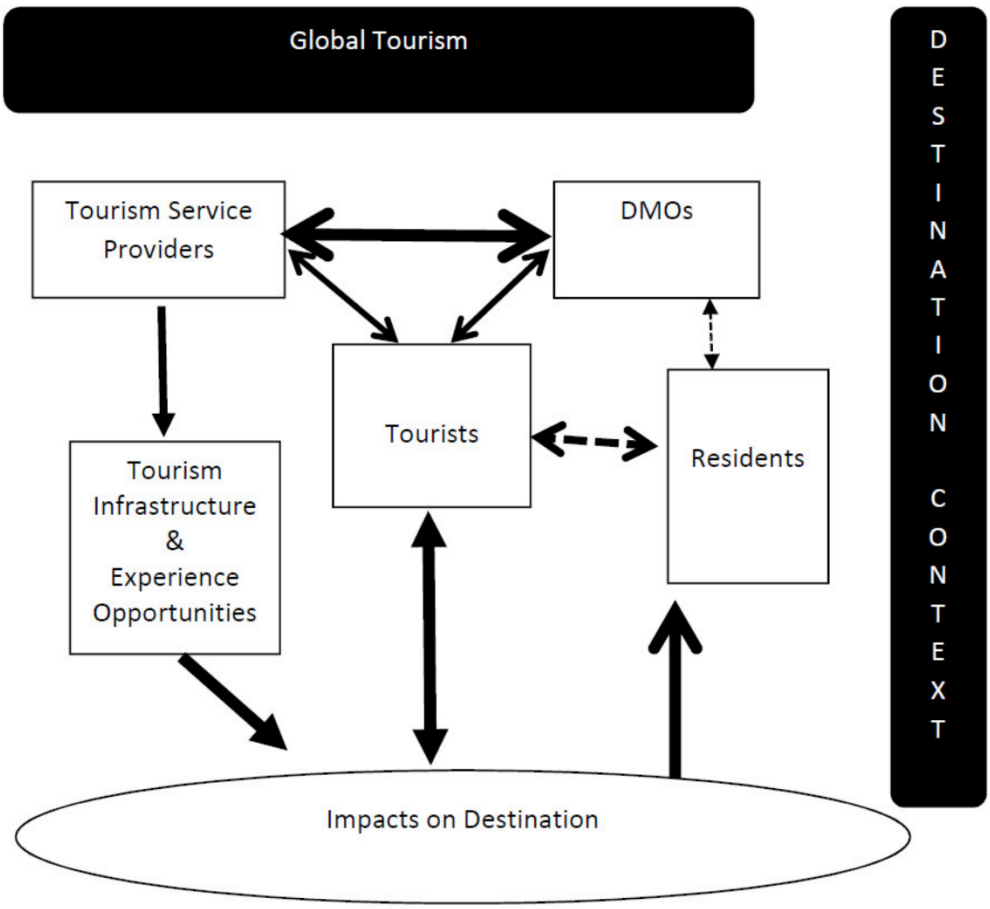
Introduction:

The voluntary disclosure of private data and posted content on numerous social media websites has opened up numerous avenues for insightful research. Because so many people voluntarily post content on social networking sites (like Facebook, Twitter, Flickr, etc.) and upload digital photos and videos, a variety of data kinds are constantly expanding within these platforms. According to the infographic, 72 hours of new video files are uploaded to YouTube every minute. This is a representation of the massive volume of data that needs to be managed in any real-world analytics project. But when the volume and velocity of pertinent data rise, traditional data management techniques are unable to effectively handle the growth and upkeep of such a vast and diverse amount of data.



The characteristics of big data are Volume (far larger than standard data sets), Velocity (the pace at which it is produced and made available), Variety (especially of formats), Variability (over time and across a variety of sources), and Volatility (variable output levels). The term "big data analytics" refers to the processes that go into defining, gathering, storing, accessing, and analysing these kinds of datasets in order to interpret their contents and utilise their potential for decision-making.

The use of big data by academics and corporate decision-makers has grown: Examples in the tourist industry include destination marketing, hospitality management, customer relationship management, and knowledge generation for strategic planning objectives in tourism destinations (TDs). Even though social media is thought to be a trustworthy and helpful source of traveller information, there is still a dearth of research on the analysis of big data generated by social media, especially in TD management. As there are currently few use cases for big data analytics in supporting strategic decision-making, our study focuses on helping this industry make strategic decisions.



A tourist destination (TD) is defined as a region that provides visitors with a wide range of attractions and activities, and that region is backed by all the hospitality and other services that the visitor may need. At its core, a tourist destination (TD) is a group of actual places where visitors spend their time and come to see sights (both man-made and natural), engage in activities (like skiing, swimming, and learning), and have fun (like going to bars, events, stores, and restaurants). Generally speaking, destination management organisations (DMOs) are in charge of overseeing development plans, managing and advertising the TD, and communicating with the regional tourism sector. As a result, they have to be aware of the demands of the upcoming market and the necessity for cooperation between the different parties.

Within this framework, the big data produced by individual travelers—that is, content or materials meant for internet distribution—may contain insightful and worthwhile information. Large amounts of data are accessible on a number of social media platforms that are used for conversation (Facebook), photo and remark sharing (Flickr), video sharing (YouTube), and instant response sharing (Twitter). Tourism authorities seldom gather such data, even though they could provide valuable insights about visitor behaviour and preferences that are pertinent to TD management. Outside of major organisations, current tools for evaluating and transforming such massive data into useful decision support information are not generally accessible. This is partially due to the fact that the vast quantity and diversity of big data sets surpass the capabilities of the most popular analytics tools [4].

Related work:

Host type and pricing on Airbnb: seasonality and perceived market power (2022):

The diversity of the host community is emphasised in the literature on short-term rentals. Some contend that the pricing strategies of professional and opportunistic hosts are different. This study demonstrates how the perception and information of the market vary, resulting in a price disparity between players who are professionals and those who are not. We analyse the pricing behaviour of Airbnb sellers in Corsica (France), based on a huge dataset of nearly 9000 properties and 73,000 observations, and we propose an original and accurate definition of professional hosts. By utilising OLS and the double-machine learning techniques, we prove that professional and opportunistic sellers have distinct prices. Furthermore, we evaluate the influence of demand seasonality on the magnitude and orientation of this price difference. During the peak season, experts experience a higher degree of market power than others, which enables them to increase their revenues.

Two decades of customer experience research in hospitality and tourism: a bibliometric analysis and thematic content analysis (2022):

The state-of-the-art systematic review of the evolution and structure of empirical research on customer experience in hospitality and/or tourist settings presented in this article is based on 1248 articles that were published between January 1998 and May 2021 in 13 prestigious journals. Through a mixed-methods strategy combining quantitative bibliometric and qualitative content analyses, research articles on customer experience were extracted and analysed using the Web of Science database and the PRISMA technique to identify this topic's intellectual evolution and important themes. The academic customer experience research could be scientifically investigated and visualised thanks to the bibliometric analysis. Following the establishment of theoretical foundations for the customer experience through content analysis based on grounded theory, a conceptual model of this idea was produced. In summary, this study delves deeply into the literature on customer experience in the hospitality and tourism industries. The results offer a comprehensive picture of the customer experience, highlight the scholarly development of the subject, and indicate important avenues for further investigation.

Methodology:

We are using the SPARK bigdata framework in this project to implement recommendations for popular tourist destinations. Spark can handle any size of data since it can process data in a distributed fashion. The proposed method uses a geotagged image dataset to extract location information, and then clusters this text data to group tourists with similar behaviours together. When a new user enters a query, the clustering algorithm predicts similar clusters based on the query and recommends the top 5 frequently visited locations that are similar in interest.

We utilised the KMEANS method to cluster user behaviour. Next, we trained the Random Forest Classifier on the user features and cluster label to explain the model. Rather than using

LIME, we used the SHAP framework for explanation. This model will clarify which features are most important in predicting a certain label.

In dataset first row represents column names and remaining rows represents dataset values and in columns we have names as “Photo ID, Description, tags, favourites as Number of time visited etc.”. So by using dataset we will cluster and recommend places for new user.

Result discussion:

We have coded this project in python and executed in google colab.

Below are the screen shots of code and output.

```
#importing classes from Libraries
from pyspark.sql import SparkSession #loading spark classes
from pyspark import SparkConf, SparkContext
from pyspark.sql.types import StructType, StructField
from pyspark.sql.types import DoubleType, IntegerType, StringType
import numpy as np
from string import punctuation
from nltk.corpus import stopwords
import nltk
from nltk.stem import WordNetLemmatizer
import pickle
from nltk.stem import PorterStemmer
import os
from sklearn.feature_extraction.text import TfidfVectorizer #loading tfidf vector
from sklearn.cluster import KMeans
from k_means_constrained import KMeansConstrained
import matplotlib.pyplot as plt
import pandas as pd
```

In above screen importing spark, NLP (natural language processing API to remove stop words, special symbols from geo tag text dataset) and then importing KMEANS and other classes

```
[3]  #defining column datatype for spark dataset
     schema = StructType([
         StructField("photo_id", StringType()),
         StructField("title", StringType()),
         StructField("description", StringType()),
         StructField("tags", StringType()),
         StructField("faves", DoubleType())
     ])
```

```
     # Creating a SparkSession named "HDFS" for working with Spark
     spark = SparkSession.builder.appName("HDFS").getOrCreate()
     sparkcont = SparkContext.getOrCreate(SparkConf().setAppName("HDFS"))
     logs = sparkcont.setLogLevel("ERROR")

     # Getting the absolute path of the CSV file in HDFS
     filePath = os.path.abspath("/content/tourism.csv")

     # Reading the CSV file into a Spark DataFrame with specified options
     dataset = spark.read.csv("file:///" + filePath, header=True, inferSchema=True, schema=schema)

     # Converting the Spark DataFrame to a Pandas DataFrame for local data manipulation
     dataset = dataset.toPandas()

     # Dropping rows with missing values (NaN)
     dataset = dataset.dropna()

     # Dropping duplicate rows in the dataset
     dataset = dataset.drop_duplicates()

     # Displaying the resulting Pandas DataFrame
     dataset
```

In above screen using Spark we are loading Tourism dataset and after loading will get below output

	photo_id	title	description	tags	faves
0	47922891637	British Museum	A marble statue of Emperor Septimius Severus, ...	britishmuseum, london, sculpture, roman	4.0
1	47922890172	British Museum	Busts of Roman worthies.	britishmuseum, london, sculpture, roman	5.0
2	47922879348	British Museum	A bronze bust of the emperor Lucius Verus, who...	britishmuseum, london, sculpture, roman, bronz...	7.0
3	46995613225	Ludgate House`Demolished	Heres A Special Request For Contact `RTM Boy` ...	london, se1, southwark, blackfriars, local, hi...	4.0
6	40943646353	Sip Savor Enjoy	Nothing like a soothing cup of tea to enjoy in...	stilllife, outdoors, park, trees, grass, drink...	3.0
...	...	...	...	...	...
9263	34652005736	20140913 200526 1SL4-COLLAGE	1S United Kingdom England London Cranbourn Street	1s, 2046x2046x24b, 8c, canoncanoneos550d, cran...	0.0
9264	34562027021	20140913 161704 1SL4-COLLAGE	1S United Kingdom England London Tower Hill	1160s, 1s, 2046x2046x24b, 3456x2304x24b, 8c, c...	0.0
9265	34307675310	20140913 161556 1SL4-COLLAGE	1S United Kingdom England London Tower Hill	1s, 2046x2046x24b, 8c, canoncanoneos550d, engl...	0.0
9266	34692433725	20140912 190622 1SL4-SMILE	1S United Kingdom England London Bryanston Street	1s, 2048x1366x24b, 8c, bryanstonstreet, canonc...	0.0
9267	34530657852	20140911 135357 1SL4-COLLAGE	1S United Kingdom England London Whitcomb Street	1s, 2046x2046x24b, 8c, canoncanoneos550d, engl...	1.0

In above screen we can see loaded dataset values

```
#Defining cleanText method
def cleanText(doc):
    tokens = doc.split() # Splitting the input document into tokens (words)
    table = str.maketrans('', '', punctuation)
    tokens = [w.translate(table) for w in tokens] # Creating a translation table to remove punctuation from each token
    tokens = [word for word in tokens if word.isalpha()] # Removing non-alphabetic words
    tokens = [w for w in tokens if not w in stop_words] # Removing stopwords
    tokens = [word for word in tokens if len(word) > 1] # Removing short words (length less than 2 characters)
    tokens = [ps.stem(token) for token in tokens] # Stemming each token using PorterStemmer
    tokens = [lemmatizer.lemmatize(token) for token in tokens] # Lemmatizing each token using WordNetLemmatizer
    tokens = ' '.join(tokens)
    # Returning the cleaned text
    return tokens
```

In above screen defining function to clean geo tagged text data

```
[7] #Creating tfidf method
    temp = dataset.values
    tfidf_X = []
    for i in range(len(temp)):
        # Loop through all geotagged text from the dataset
        title = temp[i, 1]
        desc = temp[i, 2]
        tags = temp[i, 3]

        # Concatenate all values
        data = title.strip() + " " + desc.strip() + " " + tags.strip()

        # Convert to lowercase
        data = data.lower().strip()

        # Clean data using the 'cleanText' function (assumed to be defined elsewhere)
        data = cleanText(data)

        # Add clean data to the list
        tfidf_X.append(data)

    # Create a TF-IDF vectorizer
    tfidf_vectorizer = TfidfVectorizer(stop_words= None, use_idf=True, smooth_idf=False, norm=None, decode_error='replace', max_features=200)

    # Fit and transform the text data to obtain the TF-IDF matrix
    tfidf_X = tfidf_vectorizer.fit_transform(tfidf_X).toarray()
```

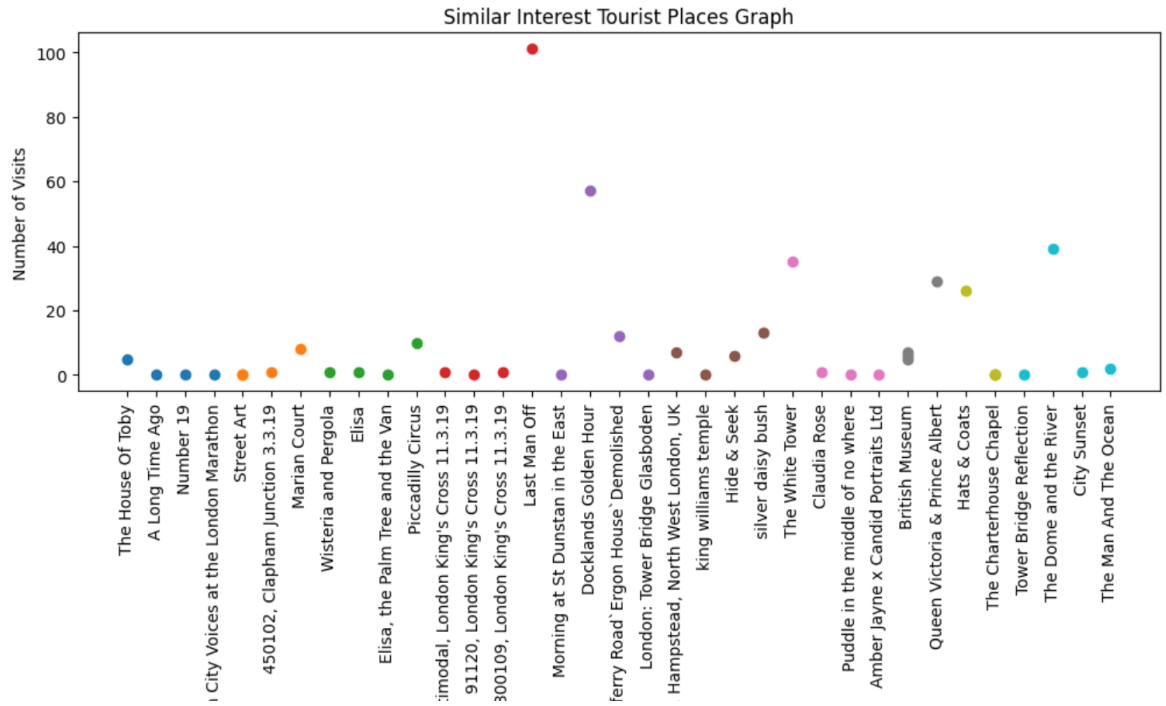
In above screen reading all Geo Tagged text data and then converting all clean text data into numeric TFIDF vector and this vector contains average frequency of each words and if word does not contains then vector will have 0 and by using this vector KMEANS will perform clustering

```
# Creating a KMeansConstrained object with 10 clusters and size constraints
kmeans = KMeansConstrained(n_clusters=10, size_min=200, size_max=tfidf_X.shape[0], random_state=0)

# Fitting the KMeansConstrained model to the TF-IDF matrix (tfidf_X)
kmeans.fit(tfidf_X)

# Predicting the cluster labels for each data point in the TF-IDF matrix
clusters = kmeans.predict(tfidf_X)

# Adding a new column 'cluster' to the original dataset to store the cluster labels
dataset['cluster'] = clusters

# Displaying the dataset with the assigned cluster labels
dataset
```

	photo_id	title	description	tags	faves	cluster
0	47922891637	British Museum	A marble statue of Emperor Septimius Severus, ...	britishmuseum, london, sculpture, roman	4.0	1
1	47922890172	British Museum	Busts of Roman worthies.	britishmuseum, london, sculpture, roman	5.0	1
2	47922879348	British Museum	A bronze bust of the emperor Lucius Verus, who...	britishmuseum, london, sculpture, roman, bronz...	7.0	1
3	46995613225	Ludgate House`Demolished	Heres A Special Request For Contact `RTM Boy` ...	london, se1, southwark, blackfriars, local, hi...	4.0	5
6	40943646353	Sip Savor Enjoy	Nothing like a soothing cup of tea to enjoy in...	stilllife, outdoors, park, trees, grass, drink...	3.0	2
...	...	...	...	...	...	...

In above screen we performed clustering on all text data and after clustering in output last column we can see which row or user place goes to which cluster. In above table in last column we can see cluster label for each records. In above clustering we have created 10 clusters so all rows will be distributed between 1 to 10 clusters.

```
# Plotting a cluster graph with similar tourist interests and number of visits

# Setting the figure size for the plot
plt.figure(figsize=(12, 4))

# Iterating over each cluster (assumed to be 10 clusters)
for i in range(0, 10):
    # Selecting data points belonging to the current cluster
    cls1 = dataset[clusters == i]

    # Plotting a scatter plot for the selected data points
    plt.scatter(cls1.values[1:5, 1], cls1.values[1:5, 4])

# Rotating x-axis labels for better visibility
plt.xticks(rotation=90)

# Setting the x-axis label
plt.xlabel("Similar Tourist Places Clusters")

# Setting the y-axis label
plt.ylabel("Number of Visits")

# Setting the title of the plot
plt.title("Similar Interest Tourist Places Graph")

# Displaying the plot
plt.show()
```



In above graph x-axis represents place names and y-axis represents Number of visited and small dots represents number of time place is visited

```
# Taking user input for their interest
query = input("Enter your interest: ")

# Converting the user input to lowercase and cleaning the text using the cleanText function
data = query.lower()
data = cleanText(data)

# Creating a list containing the cleaned user input
temp = [data]

# Transforming the query using the TF-IDF vectorizer
temp_tfidf = tfidf_vectorizer.transform(temp).toarray()

# Using kmeans.predict on the transformed query to predict the cluster
predict = kmeans.predict(temp_tfidf, size_min=None, size_max=None)
predict = predict[0]

# Selecting data points from the dataset belonging to the predicted cluster
output = dataset[clusters == predict].nlargest(5, "faves")

# Calculating the total number of visitors for the selected data points
total_visited = np.sum(output['faves'])

# Displaying the total number of visitors
print("Total Number of Visitors: " + str(total_visited))
print()

# Displaying the top 5 tourist places based on user's interest and cluster
print(output)
```

In above screen we define function which will read user query and then recommend similar places from cluster based on user interest query

```
Enter your interest: museum
Total Number of Visitors: 748.0

         photo_id                            title  \
3690  46528176411                      London classic
8269  47362502411  London, Natural History Museum
3821  46207905102                  adequate scenery
2502  40606920563               365in5theparcelyard
2214  33821314158                  Room with a View

                                          description  \
3690  12mm ultra wide view into the hall of Natural ...
8269                                          SONY DSC
3821             Bride @ Natural History Museum, London
2502  During a five year period, starting 01/01/17, ...
2214             A View from The Tate Modern in London.

                                                  tags  faves  cluster
3690  london, england, uk, unitedkingdom, naturalhis...  264.0        1
8269                           london, museum, nex3  232.0        1
3821  greatbritain, england, grosbritannien, london,...  105.0        1
2502  beer, glass, ale, realale, theparcelyard, king...   75.0        1
2214  tatemodern, view, fromthetop, museum, cityscap...   72.0        1
```

In above screen the query is given as ‘museum’ and press enter key to get top recommended places. Similarly you can enter query and get popular tourist places from cluster. Above output can be consider as future tourism places which will be in demand.