**A MINOR PROJECT REPORT**

**on**

# ANALYSING TOURIST BEHAVIOUR

# USING BIG DATA TECHNOLOGY

***Submitted in partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)**

## By

**J. SAI HRUSHIKESH : 21Q91A67E0**

**UDAY BHASKAR : 21Q91A67H3**

**M. MADHU : 21Q91A67F7**

**K. KUSUMA SRI : 21Q91A67E7**

## Under the guidance of

**Mr. RAVI**

**Assistant Professor, Dept. of CSE-DS**



## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## (DATA SCIENCE)

**MALLA REDDY COLLEGE OF ENGINEERING**

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)

Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO 9001:2015 certified Institution

Maisammaguda, Dhulapally (Post via Kompally), Secunderabad- 500100

**2024 – 2025**

# MALLA REDDY COLLEGE OF ENGINEERING

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)

Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO

9001:2015 certified Institution

Maisammaguda, Dhulapally (Post via Kompally), Secunderabad- 500100



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**(DATA SCIENCE)**



## CERTIFICATE

This is to certify that the Minor Project report on “**ANALYSING TOURIST BEHAVIOUR USING BIG DATA TECHNOLOGY**” is successfully done by the following students **of** Department of **Computer Science & Engineering (Data Science)** of our college in partial fulfilment of the requirement for the award of Bachelor of Technology degree in the year 2024-2025. The results embodied in this report have not been submitted to any other University for the award of any diploma or degree.

**J. SAI HRUSHIKESH : 21Q91A67E0**

**UDAY BHASKAR : 21Q91A67H3**

**M. MADHU : 21Q91A67F7**

**K. KUSUMA SRI : 21Q91A67E7**

**Submitted for the viva voice examination held on: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

### INTERNAL GUIDE PROJECT COORDINATOR HOD

**Mrs.B. SWATHI Mrs.D.Kranthi Deep Dr.J.Britto**

**Asst. Professor Asst.Professor Professor & Head**

**Internal Examiner External Examiner**

### DECLARATION

We**, J. SAI HRUSHIKESH, UDAY BHASKAR, M.MADHU, K. KUSUMA SRI** with Regd.no. **21Q91A67E0, 21Q91A67H3, 21Q91A67F7, 21Q95A67E7** are hereby declaring that the minor project entitled “**ANALYSING TOURIST BEHAVIOUR USING BIG DATA TECHNOLOGY**” has done by us under the guidance of **Mr. RAVI** Assistant Professor, Department of Computer Science And Engineering (Data science) is submitted in the partial fulfilment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE).**

The Result embedded in this project report have not been submitted to any other University or institute for the award of any degree or diploma.

Signature of the Candidate

**J. SAI HRUSHIKESH :21Q91A67E0**

**UDAY BHASKAR :21Q91A67H3**

**M. MADHU :21Q91A67F7**

**K. KUSUMSA SRI :21Q91A67E7**

**DATE:**

**PLACE: Maisammaguda**

### ACKNOWLEDGEMENT

First and foremost, we would like to express our immense gratitude towards our institution Malla Reddy College of Engineering, which helped us to attain profound technical skills in the field of Computer Science & Engineering, there by fulfilling our most cherished goal.

We are pleased to thank **Sri Ch. Malla Reddy**, our Founder Chairman **MRGI**, **Sri Ch. Mahender Reddy, Secretary, MRGI** for providing this opportunity and support throughout the course.

It gives us immense pleasure to acknowledge the perennial inspiration of **Dr. M. Ashok** our beloved principal for his kind co-operation and encouragement in bringing out this task.

We would like to thank **Dr. J. Gladson Maria Britto,** Professor, HOD, CSE(DS) Department for their inspiration adroit guidance and constructive criticism for successful completion of our degree.

We would like to thank **Mrs. D. Kranthi Deep** Assistant Professor, CSE(DS), for her valuable suggestions and guidance during the exhibition and completion of this project.

We would like to thank **Mr. RAVI** Assistant Professor our internal guide, for his valuable suggestions and guidance during the exhibition and completion of this project.

Finally, we avail this opportunity to express our deep gratitude to all staff who have contribute their valuable assistance and support making our project success.

**J. SAI HRUSHIKESH :21Q91A67E0**

**UDAY BHASKER :21Q91A67H3**

**M. MADHU :21Q91A67F7**

**K. KUSUMA SRI :21Q91A67E7**

### INDEX

**CONTENT PAGE NO.**

LIST OF FIGURES i

LIST OF ABBREVATION ii

ABSTRACT iii

**CHAPTER 1 : INTRODUCTION 1**

[1.1 PROBLEM DEFINATION 2](file:///C:\Users\Akhila\AppData\Local\Microsoft\Windows\INetCache\IE\MEQO34PD\ppt_abcd_(12)%5b1%5d.pdf#_Toc99236)

[1.2 OBJECTIVE OF PROJECT 2](file:///C:\Users\Akhila\AppData\Local\Microsoft\Windows\INetCache\IE\MEQO34PD\ppt_abcd_(12)%5b1%5d.pdf#_Toc99237)

[1.3 LIMITATIONS OF PROJECT 3](file:///C:\Users\Akhila\AppData\Local\Microsoft\Windows\INetCache\IE\MEQO34PD\ppt_abcd_(12)%5b1%5d.pdf#_Toc99238)

1.4 ORAGANIZATION OF DOCUMENTATION 5

**CHAPTER 2 : LITERATURE SURVEY**  7

2.1 INTRODUCTION **8**

2.2 EXISTING SYSTEM 8

2.2.1 DISADVANTAGES OF EXISTING SYSTEM 9

2.3 PROPOSED SYSTEM 10

2.3.1 ADVANTAGES OF PROPOSED SYSTEM 10

**CHAPTER 3 : ANALYSIS**  12

3.1 INTRODUCTION 13

3.1.1 KEY COMPONENTS 13

3.2 SOFTWARE REQUIREMENT SPECIFICATIONS 14

3.2.1 HARDWARE REQUIREMENTS 14

[3.2.2 SOFTWARE REQUIREMENTS 1](file:///C:\Users\Akhila\AppData\Local\Microsoft\Windows\INetCache\IE\MEQO34PD\ppt_abcd_(12)%5b1%5d.pdf#_Toc99251)4

3.3 ALGORITHMS 15

**CHAPTER 4 : DESIGN** 18

4.1 INTRODUCTION 19

4.2 UML DIAGRAM 19

4.2.1 USE CASE DIAGRAM 21

4.2.2 CLASS DIAGRAM 22

4.2.3 SEQUENCE DIAGRAM 23

4.2.4 ACTIVITY DIAGRAM 24

**CHAPTER 5 : IMPLEMENTATION**  25

5.1 SOFTWARE ENVIRONMENT 26

5.2 MODULES USED IN PROJECT 36

5.3 CODING 38

**CHAPTER 6 : TESTING AND VALIDATION** 46

6.1 TYPES OF TESTING 47

6.2 OUTPUT SCREENS 51

**CHAPTER 7 : CONCLUSION**  58

7. CONCLUSION 59

**CHAPTER 8 : FUTURE SCOPE**  60

8. FUTURE SCOPE 61

**CHAPTER 9 : REFERENCE 62**

**9. REFERENCE 63**

### LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Name of The Figure** | **Page No.** |
| 1 | Use Case Diagram | 21 |
| 2 | Class Diagram | 22 |
| 3 | Sequence Diagram | 23 |
| 4 | Activity Diagram | 24 |
| 5 | Output Screens | 51 |

**i**

#### LIST ABBREVIATIONS

#### S.No Short Form Full Form

1. UML Unified Modeling Language
2. AI Artificial Intelligence
3. SQL StructuredQueryLanguage
4. OS Operating System

#### 

#### 

#### ii

**ABSTRACT**

In this project, we embark on the implementation of a Tourist Place Recommendation system utilizing the powerful Big Data framework, Apache Spark. With its distributed processing capabilities, Spark enables the efficient handling of datasets of any size. Our focus is on leveraging Geo-Tagged images datasets to extract location information, clustering similar tourist behaviors, and providing personalized recommendations based on user queries.This implements a comprehensive framework for recommending tourist destinations based on user interest queries. It begins by preprocessing textual data from geotagged images, such as titles, descriptions, and tags. The preprocessing involves cleaning the text by removing stop words, punctuation, and non-alphabetic characters and then transforming the cleaned data into numerical vectors using TF-IDF for feature extraction. To organize the tourist locations into meaningful groups, the code employs KMeans clustering, including a constrained version to ensure balanced cluster sizes.When a user inputs a query, it is similarly cleaned and vectorized to identify the nearest matching cluster. The system then recommends toprated tourist spots within the matched cluster, ranked by user engagement metrics such as favorites. A Random Forest classifier is used to classify user interests based on the clusters, and the SHAP (SHapley Additive exPlanations) framework is integrated to provide insights into the model’s feature importance, enhancing interpretability. The system also visualizes clusters and visitation trends, offering an overview of related destinations and their popularity.

iii

**CHAPTER 1**

**INTRODUCTION**

1. **INTRODUCTION**

In the dynamic landscape of the tourism industry, understanding and harnessing insights from tourist behavior have become imperative for destinations and businesses aiming to provide personalized and enriching experiences. Recognizing this evolving need, our project, titled "Analyzing Tourist Behavior using Big Data Technology," endeavors to employ advanced data analytics to unravel the intricate patterns and preferences that define tourist activities. By leveraging the power of big data technologies, we aim to revolutionize the way we glean insights from vast and diverse datasets.

**KEY FEATURES**

**1.Real-time Data Collection**

Big data allows for the continuous collection of real-time data from a variety of sources such as social media, mobile apps, GPS data, online bookings, and reviews.

**2. Personalization and Predictive Analytics**

By analyzing patterns, such as previous travel destinations or spending habits, companies can offer personalized recommendations, discounts, and experiences tailored to individual tourists.

**3. Segmentation of Tourists**

By analyzing demographic, behavioral, and psychographic data, tourists can be segmented into different groups based on factors like age, spending capacity, travel interests, and previous experiences.

**4. Sentiment Analysis**

Big data technology can analyze customer feedback, social media posts, and reviews to gauge tourist sentiment toward destinations, services, and experiences.

**5. Geospatial and Location-Based Insights**

By processing GPS and geolocation data, big data can map out tourist movement patterns, popular tourist hotspots, and local traffic trends.

* 1. **PROBLEM DEFINATION**

The problem tackled by this program revolves around providing personalized tourist recommendations based on user interests, leveraging large datasets of geotagged content, including images, descriptions, and tags associated with tourist locations. With the increasing volume of data generated by social media platforms, travel websites, and other sources, it has become a significant challenge to help tourists discover destinations that align with their preferences in an efficient and personalized manner. The program aims to overcome this challenge by using machine learning algorithms and big data technologies to process, analyze, and generate recommendations for tourist destinations based on historical data.

At the core of the program, the primary task is to predict and suggest tourist destinations that match a user's interests, which are derived from their query input.

* 1. **OBJECTIVE OF PROJECT**

The objective of this project is to develop a recommendation system for tourist places that provides personalized suggestions based on a user's interests. The system uses advanced machine learning techniques, such as TF-IDF vectorization, K-means clustering, and Random Forest classification, to analyze and recommend places to visit. The core functionality revolves around transforming unstructured text data (including titles, descriptions, and tags of tourist places) into meaningful numerical representations through TF-IDF. This allows the system to cluster similar places based on their textual features, creating groups of tourist spots that share common characteristics.

To handle large-scale data efficiently, the project uses Apache Spark to process the dataset, which includes geotagged images and textual data from sources like Flickr. This ensures that the system can scale and process large volumes of information.

* 1. **LIMITATIONS OF PROJECT**

**Limited Text Understanding:** The model doesn’t capture the full meaning of place descriptions and may miss nuances, leading to imperfect recommendations.

**Cluster Quality:** KMeans clustering can group places in ways that are hard to interpret or may mix dissimilar places together.

**Basic Query Matching:** The system only matches user queries based on text, lacking deeper understanding of user interests and without personalization.

**Cold Start Problem**: New places with little data or user feedback can't be recommended effectively. Scalability Issues: As the dataset grows, the model may struggle with performance and memory usage.

**Complex Explanations**: SHAP values provide explanations but might be too technical for non- experts to understand.

**Bias Toward Popular Places**: The system may favor well-known places based on popularity (faves), ignoring niche preferences.

**Data Quality:** Over-simplified text cleaning and lack of synonym handling may reduce the model's accuracy.

**Limited Generalization:** The system may struggle with regions underrepresented in the data or new, unseen places.

* 1. **ORGANIZATION FOR DOCUMENTATION**

Organizing documentation for our code is crucial for maintaining clarity, ease of use, and scalability.

**INTRODUCTION**

This project aims to recommend tourist places based on user preferences using machine learning techniques and big data analytics. The system uses geotagged images and textual descriptions of tourist destinations, analyzes user queries, and generates recommendations based on clusters.

**LITERATURE SURVEY**

This program leverages machine learning techniques such as KMeans clustering and RandomForest for personalized tourism recommendations, utilizing big data processing with Apache Spark. It combines text processing, user preferences, and SHAP for model interpretability to suggest relevant tourist destinations.

**SYSTEM ANALYSIS**

The system analyzes user interests through text-based queries, clusters similar tourist destinations using KMeans and provides personalized recommendations by leveraging machine learning, big data processing, and SHAP for model interpretability.

**SYSTEM DESIGN**

The system design involves data preprocessing, TF-IDF vectorization, KMeans clustering, RandomForest classification for prediction, and SHAP for model explainability, all integrated with Apache Spark for large-scale data processing and personalized tourism recommendations.

**IMPLEMENTATION**

The implementation involves data preprocessing, TF-IDF vectorization, KMeans clustering for tourist place grouping, RandomForest classification for predicting user interests and SHAP for model explanation, all integrated with Apache Spark for large-scale data handling.

**TESTING**

System testing involves validating the accuracy of clustering, recommendation relevance, and prediction performance using test queries, while ensuring proper model explainability through SHAP visualizations and assessing data processing efficiency with Apache Spark.

**RESULTS AND DISCUSSION**

The results demonstrate accurate tourist place recommendations based on user interests, with effective clustering, predictive accuracy from the RandomForest model.

**FUTURE SCOPE**

The future scope includes integrating real-time data, incorporating multimodal inputs like images and reviews, and enhancing personalization using deep learning models for more accurate and dynamic tourism recommendations.

**CONCLUSION**

The program successfully utilizes machine learning, big data, and model explainability techniques to provide personalized and accurate tourism recommendations, enhancing user experience through scalable and interpretable results.

**REFERENCE**

References for the program include works on KMeans clustering (MacQueen, 1967), RandomForest (Breiman, 2001), SHAP explainability (Lundberg & Lee, 2017), and TF-IDF text vectorization (Ramos,2003).

**CHAPTER 2**

**LITERATURE SURVEY**

# 

# 2. LITERATURE SURVEY

**2.1 INTRODUCTION**

* **Title** : A tourist flows analysis system based on phone big data
* **Author** : [Dan-Dan Lu](https://ieeexplore.ieee.org/author/37085777448); [Yong-De Zhong](https://ieeexplore.ieee.org/author/37958745100)
* **Abstract:** The analysis of tourist flows behavior can help to achieve the reasonable allocation of social resources for tourist area and respond effectively to the tourist destination traffic pressure, maintain social public security, etc. The traditional manual analysis methods, such as questionnaire survey, have the characters of the high cost and low efficiency.
* **Title :** Understanding Travel Patterns of Tourists from Mobile Phone Data: A Case Study in
* Hainan o **Author** : [Qingqing Chen;](https://ieeexplore.ieee.org/author/37086037073) [Zheng Hu](https://ieeexplore.ieee.org/author/37600666100); [Hang Su;](https://ieeexplore.ieee.org/author/37087656282) [Xiaosheng Tang](https://ieeexplore.ieee.org/author/37086420838); [Ke Yu](https://ieeexplore.ieee.org/author/37674212900)
* **Abstract:** Large scale of locational data generated by mobile devices present an opportunity to change the structure of traditional research in tourism. However, tourist-focusing mobility patterns haven't been explored enough, which are tremendously useful for optimizing tourism resource allocation. To fulfill the need, we design a new analysis framework for understanding travel patterns of tourists by using the massive anonymous CDRs. The analysis framework consists of three layers that are data layer, algorithm layer and application layer.
* **Title** : User Behavior Patterns and Activity Recognition Based on Mobile Big Data
* **Author** : [Yong Liu;](https://ieeexplore.ieee.org/author/37089301510) [Na Zhao;](https://ieeexplore.ieee.org/author/37088873842) [Wei Han;](https://ieeexplore.ieee.org/author/37089303156) [Qinghua Zhu](https://ieeexplore.ieee.org/author/37089302451)
* **Abstract:** User behavior patterns and activity recognition have a wide range of applications in location-based information mining, such as early warning systems, traffic flow planning, urban computing, mobile marketing, social networking, and user portraits. warning systems, traffic flow planning, urban computing, mobile marketing, social networking, and user portraits.

**2.2 EXISTING SYSTEM**

# The current methodologies for analyzing tourist behavior often grapple with the challenges posed by the sheer volume, velocity, and variety of data generated in the tourism sector. Conventional analytics tools may struggle to keep pace with the dynamic nature of tourist preferences, leading to limitations in accurately capturing the nuances of their behavior. The existing systems may not fully exploit the potential of big data technologies, missing opportunities to provide deeper and more nuanced insights. Our project seeks to address these limitations by bridging the gap between traditional analytics and cutting-edge big data-driven approaches, unlocking a more comprehensive understanding of tourist behavior.

**2.2.1 DISADVANTAGES OF EXISTING SYSTEM**

**Scalability Issues:** Traditional analytics tools often struggle to handle the massive volume, velocity, and variety of data generated in the tourism sector, limiting their ability to scale efficiently with growing datasets.

**Limited Real-time Analysis:** Conventional methods fail to provide real-time insights, making it challenging to track rapidly changing tourist preferences and behaviors.

**Inability to Capture Dynamic Preferences:** These systems may not accurately account for the evolving nature of tourist preferences, as they often rely on static datasets and predefined models that cannot adapt quickly.

**Lack of Personalization:** Existing systems typically offer generalized insights, missing the opportunity for highly personalized recommendations that are crucial in understanding and influencing individual tourist behaviors.

**Simplicity in Analysis**: Traditional analytics often lack advanced techniques like machine learning or deep learning, which can provide more nuanced insights into complex, non-linear relationships within tourist behavior.

**2.3 PROPOSED SYSTEM**

Our proposed system aims to usher in a new era of tourist behavior analysis, underpinned by the capabilities of big data technology. By leveraging robust frameworks such as Apache Hadoop or Apache Spark, we seek to process and analyze vast datasets comprising social media interactions, transaction records, and geospatial data. Advanced analytics and machine learning algorithms will be deployed to identify patterns, predict trends, and uncover valuable insights that can inform decision- making in the tourism sector.

To ensure scalability and real-time adaptability, cloud-based solutions will be explored, providing a flexible infrastructure for seamless integration and analysis. The envisioned system not only promises a more nuanced understanding of tourist behavior but also sets the stage for the development of personalized and data-driven strategies that can elevate the overall tourism experience. In essence, this project aspires to redefine the landscape of tourist behavior analysis, showcasing the transformative potential of big data technologies in shaping the future of the tourism industry.

* + 1. **ADVANTAGES OF PROPOSED SYSTEM**

**1. Personalized Recommendations:**

User-Centric: By processing user input queries and mapping them to relevant clusters of tourist places, the system provides personalized recommendations based on the user’s specific interests, improving the user experience.

**2. Effective Use of Geotagged Data:**

Leveraging Real-World Data: The system uses geotagged data from platforms like Flickr, allowing it to recommend places based on actual visitor data. This adds authenticity and relevance to the recommendations.

**3. Flexibility in Query Handling:**

Real-Time Recommendations: The system can handle dynamic, real-time queries from users, making it useful in a variety of scenarios.

**4. Scalability and Extensibility:**

Scalability to Larger Datasets: The system is built with scalability in mind, leveraging technologies like Spark for big data processing. This makes it capable of handling large datasets of tourist places and geotagged images without significant performance degradation.

**5. Data-Driven Decision Making:**

Data-Backed Insights: The system's recommendations are based on data from actual visitors and tourist places, ensuring that the suggestions are not arbitrary but rooted in real-world patterns.

**6. Visualizations for Insights:**

Cluster Visualization: The cluster graph allows users to see the relationships between tourist places and their popularity, making it easier to understand where certain places fit within the broader landscape of tourist interests.

**7. Integration of Machine Learning and Natural Language Processing:**

Unified System: The system combines several advanced techniques, including machine learning (Random Forest, KMeans), natural language processing (TF-IDF, text cleaning), and explainability (SHAP), into a single unified solution for tourism recommendations. This combination makes it robust and adaptable to a wide range of potential user interests and scenarios.

**CHAPTER 3**

**ANALYSIS**

# 3. ANALYSIS

## 3.1. INTRODUCTION

The program you've shared is a complex tourist recommendation system using various machine learning techniques, including clustering (KMeans), classification (RandomForest), and model explainability (SHAP). It takes into account geo-tagged data from tourist places, processes textual information (titles, descriptions, tags), and provides recommendations based on user queries.

### 3.1.1. KEY COMPONENTS

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

## ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

## TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

## SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### 3.2. SOFTWARE REQUIREMENTS SPECIFICATIONS

### 3.2.1 HARDWARE REQUIREMENTS

* RAM : Minimum 8GB
* Processor : Minimum 4GB cores
* Storage : Minimum 50GB free disk space

**3.2.2 SOFTWARE REQUIRMENTS**

* Operating System : Windows​
* Coding Language : Python​
* Python Version : Python 3.7​

### 3.3. ALGORITHMS

**Algorithm for Tourism Place Clustering and Recommendation**

1. Start
2. Load and clean the dataset.
3. Convert text to numeric form using TF-IDF.
4. Apply KMeans clustering to group similar places.
5. Train a Random Forest to predict clusters.
6. Use SHAP for model interpretation.
7. Process the user's query and recommend places.
8. Save models and visualize the results.
9. End

### Algorithm to load a geotagged photo dataset

1. Start
2. Define schema to specify column types.
3. Create Spark session to read and process data.
4. Load CSV file into a Spark DataFrame.
5. Convert to Pandas DataFrame and clean the data.
6. Display the final dataset after cleaning.
7. End

**Algorithm for process of cleaning the text**

1. Start
2. Prepare stopwords, stemmer, and lemmatizer.
3. Split the text into words.
4. Remove punctuation and unwanted words (stopwords, short words).
5. Stem and lemmatize each word.
6. Join the words back into a cleaned sentence.
7. Return the cleaned text.
8. End

**Alogorithm for loads existing processed data or processes and saves new data**

1. Start
2. Check for existing processed data.
3. If data exists:

Load TF-IDF model and vectors.

1. If data does not exist:

Clean and preprocess the dataset.

Convert text into TF-IDF vectors.

1. Save the processed data.
2. Display results (processed text).
3. End

**Algorithm for grouping tourist places into clusters**

1. Start
2. Apply KMeans clustering on the data.
3. Train the KMeans model.
4. Predict cluster labels for each data point.
5. Add cluster labels to the dataset.
6. Create a scatter plot to visualize the clusters.
7. Display the plot.
8. End

**Algorithm for Tourist Recommendation System**

1. Start
2. Collect user input
3. Preprocess the input
4. Convert input to model format
5. Predict cluster and retrieve relevant places
6. Display results
7. Explain model using SHAP
8. End

**CHAPTER 4**

**DESIGN**

# 4. DESIGN

## 4.1. INTRODUCTION

The advent of Big Data technologies, like Apache Spark, HDFS, and machine learning, has made it possible to process and analyze large-scale datasets efficiently, providing deeper insights into tourist behavior. This program leverages these technologies to analyze and predict tourist preferences, uncover patterns, and offer personalized recommendations. By processing large volumes of geospatial data, user reviews, and image metadata, we can segment tourists based on their interests, predict future trends, and understand the factors that influence tourist decisions.

## 4.2. UML DIAGRAM

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Metamodel and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

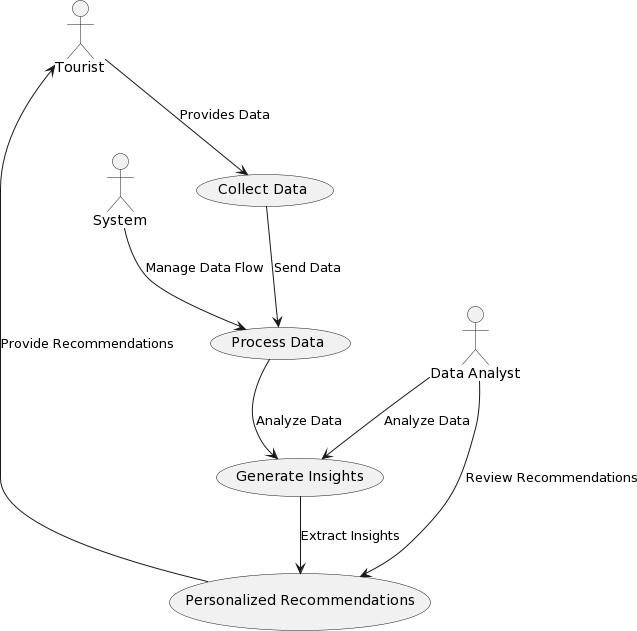
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

### 4.2.1. USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

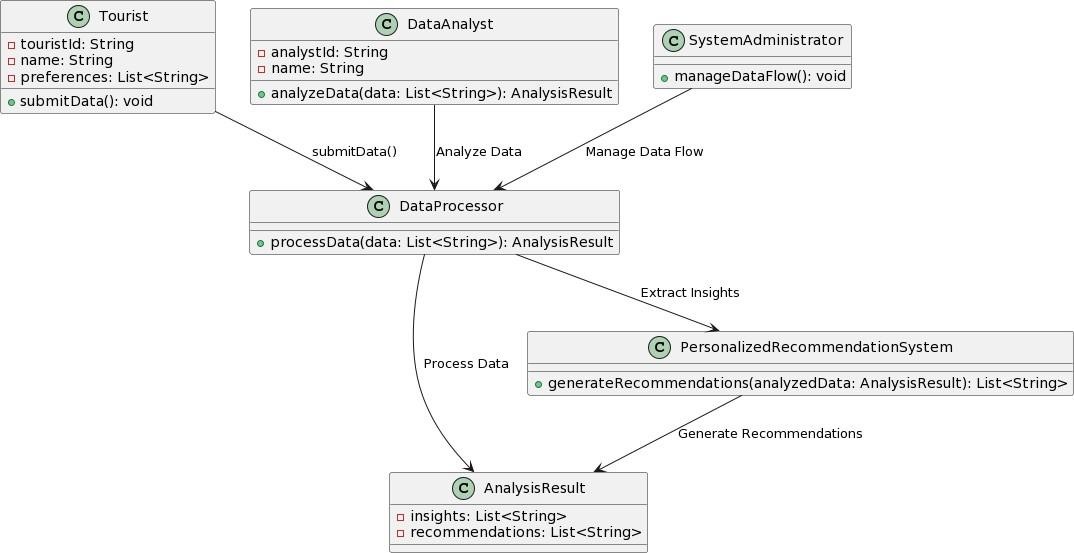


**Figure 1** : Use Case Diagram

A Use Case Diagram shows how users (called actors) interact with a system to perform tasks (called use cases). It highlights what the system does from the user’s perspective. Actors can be people or other systems, and use cases represent actions like logging in or making a purchase.

### 4.2.2. CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

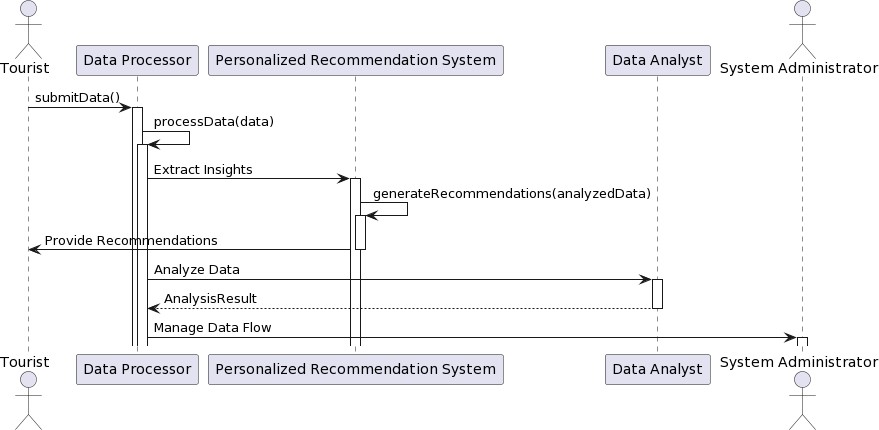


**Figure 2** : Class Diagram

A Class Diagram is a type of UML diagram that shows the structure of a system by displaying its classes, attributes, and methods. Classes are blueprints for objects in the system, and each class has attributes (data) and methods (actions).

### 4.2.3. SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

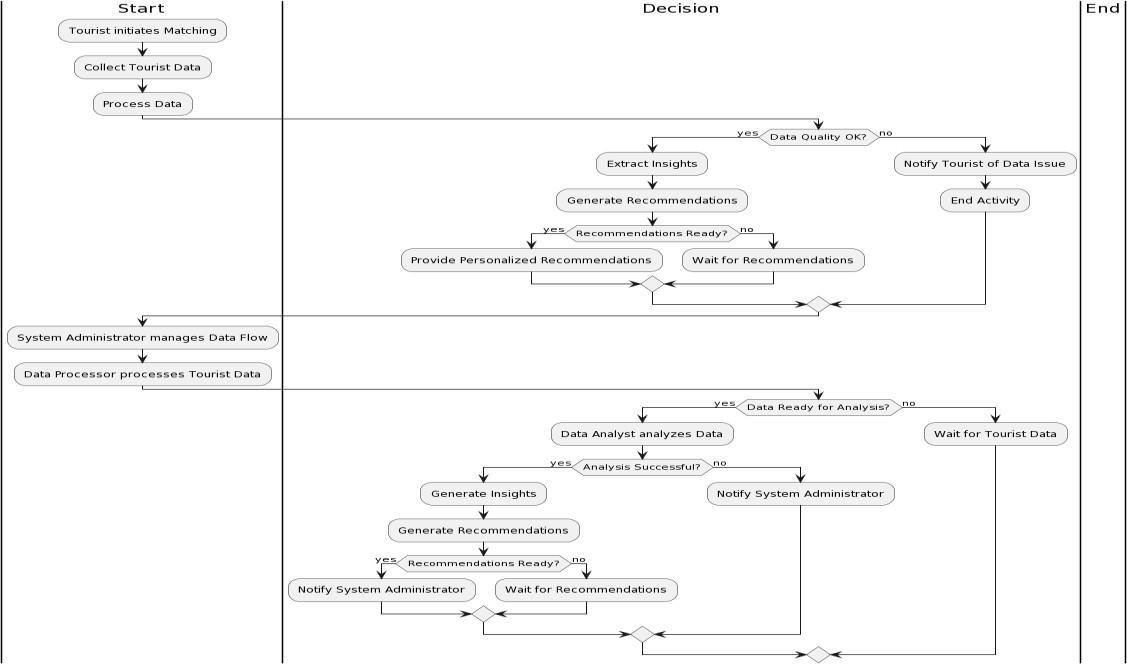


**Figure 3 :** Sequence Diagram

A Sequence Diagram shows how objects in a system interact with each other over time. It illustrates the order of messages or actions between objects to complete a specific task. The diagram typically includes objects, the messages they send to each other, and the sequence of events. For example, in an online shopping system, a sequence diagram might show how a Customer interacts with the System to place an order, step by step.

### 4.2.4. ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**Figure 4 :** Activity Diagram

An Activity Diagram shows the flow of activities or tasks in a system. It visualizes how processes are carried out, including decision points, actions, and the order in which they happen. The diagram uses shapes like ovals for actions and diamonds for decisions. For example, in a shopping process, an activity diagram might show steps like "Select Items," "Add to Cart," "Checkout," and "Make Payment," along with decisions like "Is the payment successful?" to guide the flow

**CHAPTER 5**

**IMPLEMENTATION**

# 5. IMPLEMENTATION

**5.1 SOFTWARE ENVIRONMENT :**

**What is Python :**

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
* GUI Applications (like Kivy, Tkinter, PyQt etc. )
* Web frameworks like Django (used by YouTube, Instagram, Dropbox)
* Image processing (like Opencv, Pillow)
* Web scraping (like Scrapy, BeautifulSoup, Selenium)
* Test frameworks
* Multimedia

**Advantages of Python :-**

Let’s see how Python dominates over other languages.

1. **Extensive Libraries**

Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.* So, we don’t have to write the complete code for that manually.

1. **Extensible**

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

1. **Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

1. **Improved Productivity**

The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things.

1. **IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things.. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

1. **Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

1. **Object-Oriented**

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world.

A class allows the encapsulation of data and functions into one.

1. **Free and Open-Source**

Like we said earlier, Python is freely available. But not only can you [download Python](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

1. **Portable**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

1. **Interpreted**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

**Advantages of Python Over Other Languages :**

1. **Less Coding**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

1. **Affordable**

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

1. **Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [machine learning,](https://data-flair.training/blogs/machine-learning-tutorials-home/) automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

**Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

1. **Speed Limitations**

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in slow execution. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

1. **Weak in Mobile Computing and Browsers**

While it serves as an excellent server-side language, Python is much rarely seen on the clientside. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle..

1. **Design Restrictions**

As you know, Python is dynamically-typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck-typing. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck.

1. **Underdeveloped Database Access Layers**

Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python’s database access layers.

1. **Simple**

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**What is Machine Learning : -**

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted toobserved data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

**Categories Of Machine Leaning :-**

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.*Supervised learning* involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities.

**Challenges in Machines Learning :-**

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of lowquality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well- defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

**Applications of Machines Learning :-**

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

**Advantages of machine learning:**

**1.Automation of Tasks**

Reduced Human Effort: Machine learning enables automation of repetitive tasks, freeing up human workers to focus on more complex or creative tasks. For example, ML can automate data analysis, email filtering, and customer service interactions (e.g., chatbots).

**2.Handling Large Data Sets**

Big Data Processing: ML algorithms excel at processing large volumes of data that would be too complex or time-consuming for humans to analyze. This allows organizations to extract valuable insights from vast amounts of data, such as customer behavior patterns, financial data, or sensor data.

**3.Improved Accuracy and Precision**

Data-Driven Decisions: Machine learning models can analyze complex patterns in data and make more accurate predictions compared to traditional rule-based systems or human intuition. For example, in healthcare, ML algorithms can help diagnose diseases with greater accuracy based on medical images or patient records.

**4.Personalization and Customization**

Personalized Experiences: ML enables highly personalized user experiences in areas like e- commerce, entertainment (e.g., Netflix or Spotify recommendations), and digital advertising. By analyzing user behavior, preferences, and interactions, ML models can suggest products, services, or content tailored to individual users.

**Disadvantages of Machine Learning :**

**1. Data Acquisition**

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

**2. Time and Resources**

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

**3. Interpretation of Results**

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

**4. High error-susceptibility**

Machine learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**5.2 Modules Used in Project :**

# Tensorflow

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming a](https://en.wikipedia.org/wiki/Library_(computing))cross a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks.](https://en.wikipedia.org/wiki/Neural_networks) It is used for both research and production at [Google.](https://en.wikipedia.org/wiki/Google)

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

## Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

## Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

## Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery.](https://matplotlib.org/gallery/index.html)For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

## Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

## 5.3 CODING

#import python classes and packages for tourism place recommendation

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.preprocessing import MinMaxScaler

from sklearn.model\_selection import train\_test\_split

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

import shap

from sklearn.ensemble import RandomForestClassifier

#defining column datatype for spark dataset

schema = StructType([

StructField("photo\_id", StringType()),

StructField("title", StringType()),

StructField("description", StringType()),

StructField("tags", StringType()),

StructField("faves", DoubleType())

])

#load geotagged photo dataset with places and ratings using using BIG Data Spark Framework

#this geotagged images taken from flickr and by giving photo ID we can get image

spark = SparkSession.builder.appName("HDFS").getOrCreate()

sparkcont = SparkContext.getOrCreate(SparkConf().setAppName("HDFS"))

logs = sparkcont.setLogLevel("ERROR")

filePath = os.path.abspath("Dataset/tourism.csv")

dataset=spark.read.format('csv').schema(schema).options(header='true', inferSchema='true').load("file:///"+filePath)

dataset = dataset.toPandas()

dataset = dataset.dropna()

dataset

#define object to remove stop words and other text processing

stop\_words = set(stopwords.words('english'))

lemmatizer = WordNetLemmatizer()

ps = PorterStemmer()

#define function to clean text by removing stop words and other special symbols

def cleanText(doc):

tokens = doc.split()

table = str.maketrans('', '', punctuation)

tokens = [w.translate(table) for w in tokens]

tokens = [word for word in tokens if word.isalpha()]

tokens = [w for w in tokens if not w in stop\_words]

tokens = [word for word in tokens if len(word) > 1]

tokens = [ps.stem(token) for token in tokens]

tokens = [lemmatizer.lemmatize(token) for token in tokens]

tokens = ' '.join(tokens)

return tokens

#process dataset

if os.path.exists('model/tfidf\_X.txt.npy'):

with open('model/tfidf.txt', 'rb') as file:

tfidf = pickle.load(file)

file.close()

tfidf\_X = np.load("model/tfidf\_X.txt.npy")

#load tfidf vector if dataset already cleaned

else:

#if not process then clean all text data from dataset

temp = dataset.values

tfidf\_X = []

for i in range(len(temp)):#loop all geotagged text from dataset

title = temp[i,1]#read title, description and tags

desc = temp[i,2]

tags = temp[i,3]

data = title.strip()+" "+desc.strip()+" "+tags.strip()

#concatenate all values

data = data.lower().strip()#convert to lower letter

data = cleanText(data)#clean data

tfidf\_X.append(data)#add clean data to X array

#convert clean text into numeric vector

tfidf = TfidfVectorizer(stop\_words=stop\_words, use\_idf=True, smooth\_idf=False, norm=None, decode\_error='replace', max\_features=200)

tfidf\_X = tfidf\_vectorizer.fit\_transform(tfidf\_X).toarray()

np.save("model/tfidf\_X.txt",tfidf\_X)

with open('model/tfidf.txt', 'wb') as file:

pickle.dump(tfidf, file)

file.close()

print("Numeric vector generated from cleaned Geotagged text data")

print(tfidf\_X)

#now convert all tourist geotagged places into kmeans cluster

kmeans = KMeansConstrained(n\_clusters=10, size\_min=200, size\_max=tfidf\_X.shape[0], random\_state=0)

kmeans.fit(tfidf\_X)

clusters = kmeans.predict(tfidf\_X)

dataset['cluster'] = clusters

dataset

#plot cluster graph with similar tourist interest and number of visits

plt.figure(figsize=(12,4))

for i in range(0, 10):

cls1 = dataset[clusters == i]

plt.scatter(cls1.values[1:5,1] , cls1.values[1:5,4])

plt.xticks(rotation=90)

plt.xlabel("Similar Tourist Places Clusters")

plt.ylabel("Number of Visits")

plt.title("Similar Interest Tourist Places Graph")

#plt.tight\_layout()

plt.show()

#possible tourist places to search from geo tagged

'''

britishmuseum london sculpture roman buckinghampalace themall hydepark parkline unitedkingdom towerhill unitedkingdom

'''

query = input("Enter your interest : ")#take user interest query as input

data = query.lower()

data = cleanText(data)#clean the query

temp = []

temp.append(data)

temp = tfidf.transform(temp).toarray()#convert query to vector

XX = []

for i in range(0,tfidf\_X.shape[0]):

XX.append(temp[0])

temp = np.asarray(XX)

#get user interest places from kmeans cluster

predict = kmeans.predict(temp)

predict = predict[0]#display predicted user interest places from clusters

output = dataset[clusters == predict]

output = output.nlargest(5,"faves")

total\_visited = np.sum(output['faves'])

print("Total Number of Visitors : "+str(total\_visited))

print()

output

#implementing shap modelling tool to explain model

rf = RandomForestClassifier()

rf.fit(tfidf\_X[0:300], clusters[0:300])

explainer = shap.TreeExplainer(rf)

shap\_values = explainer.shap\_values(tfidf\_X[0:300])#asking explainer to explain about features and model

shap.summary\_plot(shap\_values, tfidf\_X[0:100])#plot explaining graph

**CHAPTER 6**

**TESTING AND VALIDATION**

## 6 .TESTING & VALIDATION

### 6.1 TYPES OF TESTS:

### 

**Unit testing :**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose.

It is purpose. It is used to test areas that cannot be reached from a black box level.

### Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

### Test objectives

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

### Features to be tested

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

### Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

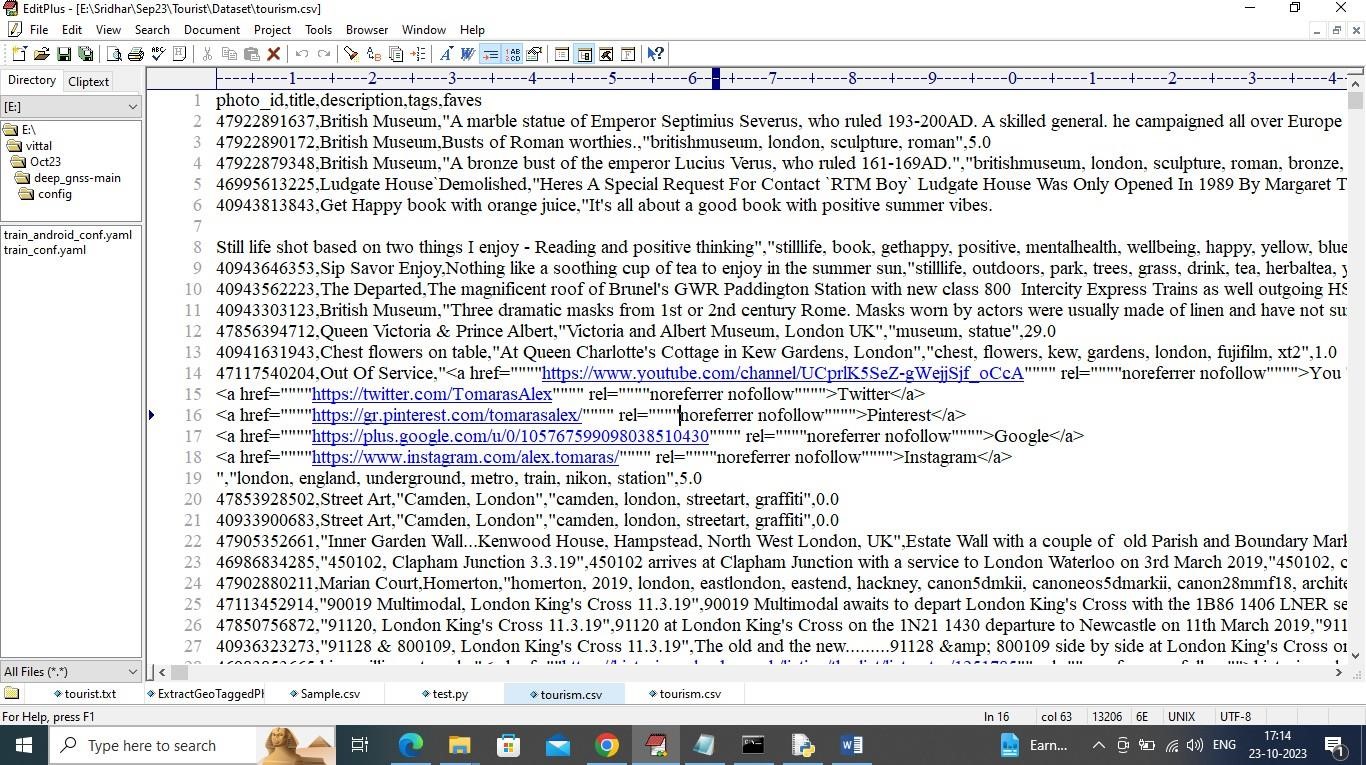
**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

### 6.2. OUTPUT SCREENS

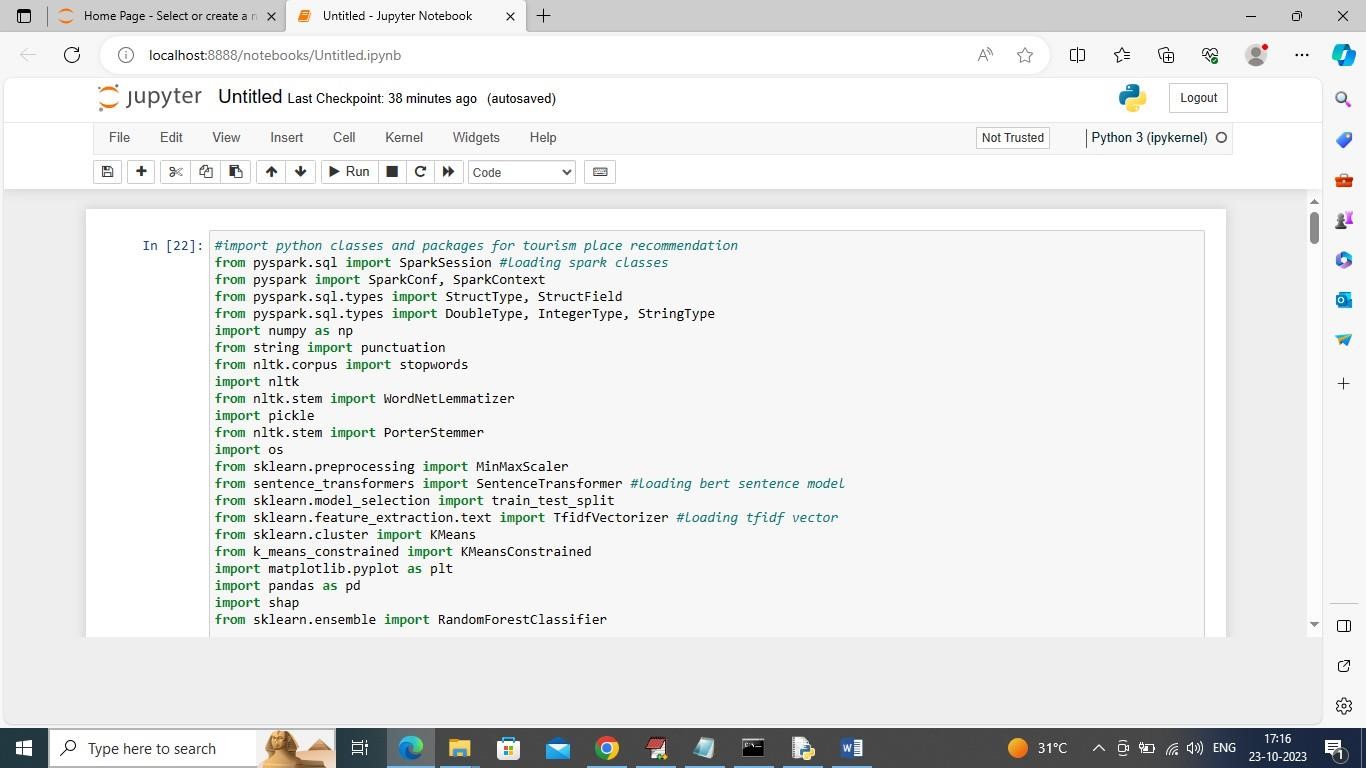
To cluster user behaviour we have used KMEANS algorithm and then the user features and cluster label will get trained with Random Forest Classifier to explain model and for explanation we have used SHAP framework instead of LIME. This model will explain which features contribute most to predict particular label.

For this project we have used below tourism data which is Geo Tagged using FLICKER

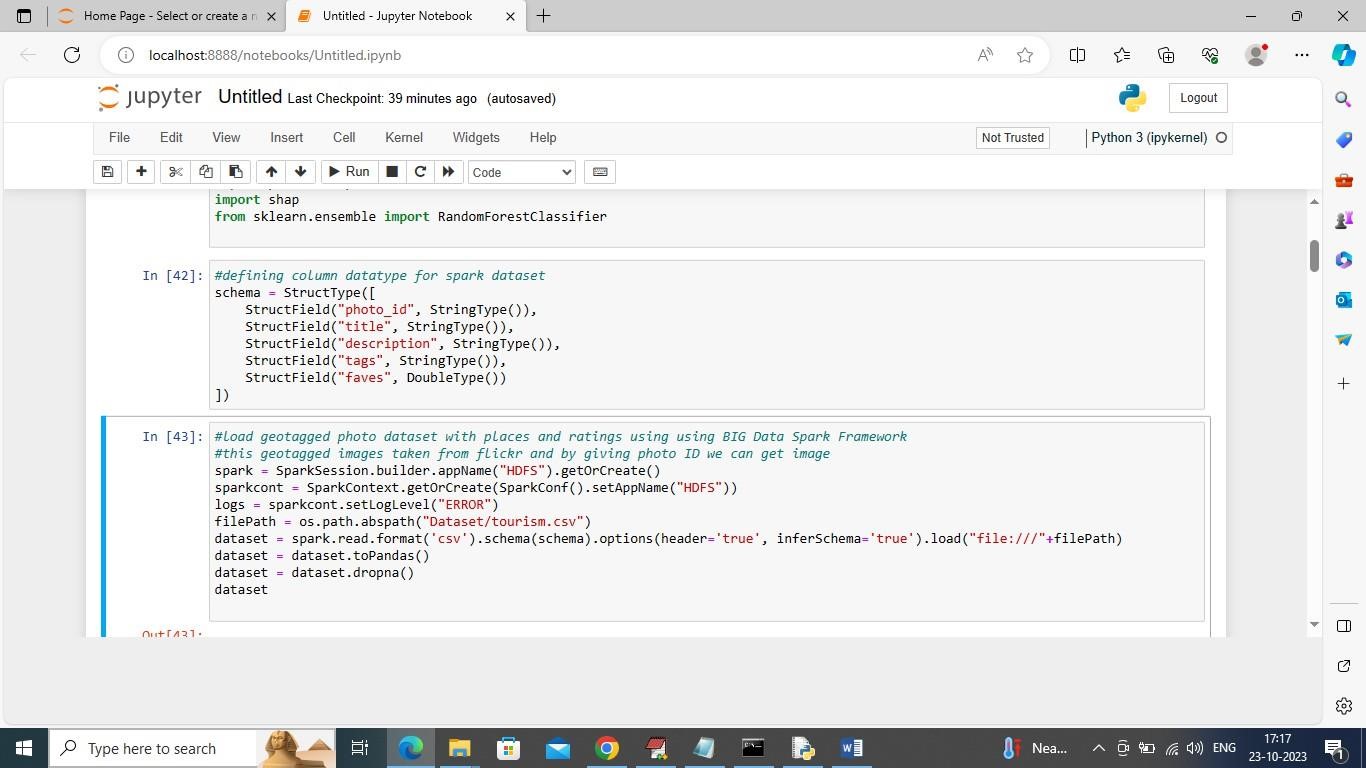


In above dataset first row represents column names and remaining rows represents dataset values and in columns we have names as “Photo ID, Description, tags, favourites as Number of time visited etc.”. So by using above dataset we will cluster and recommend places for new user.

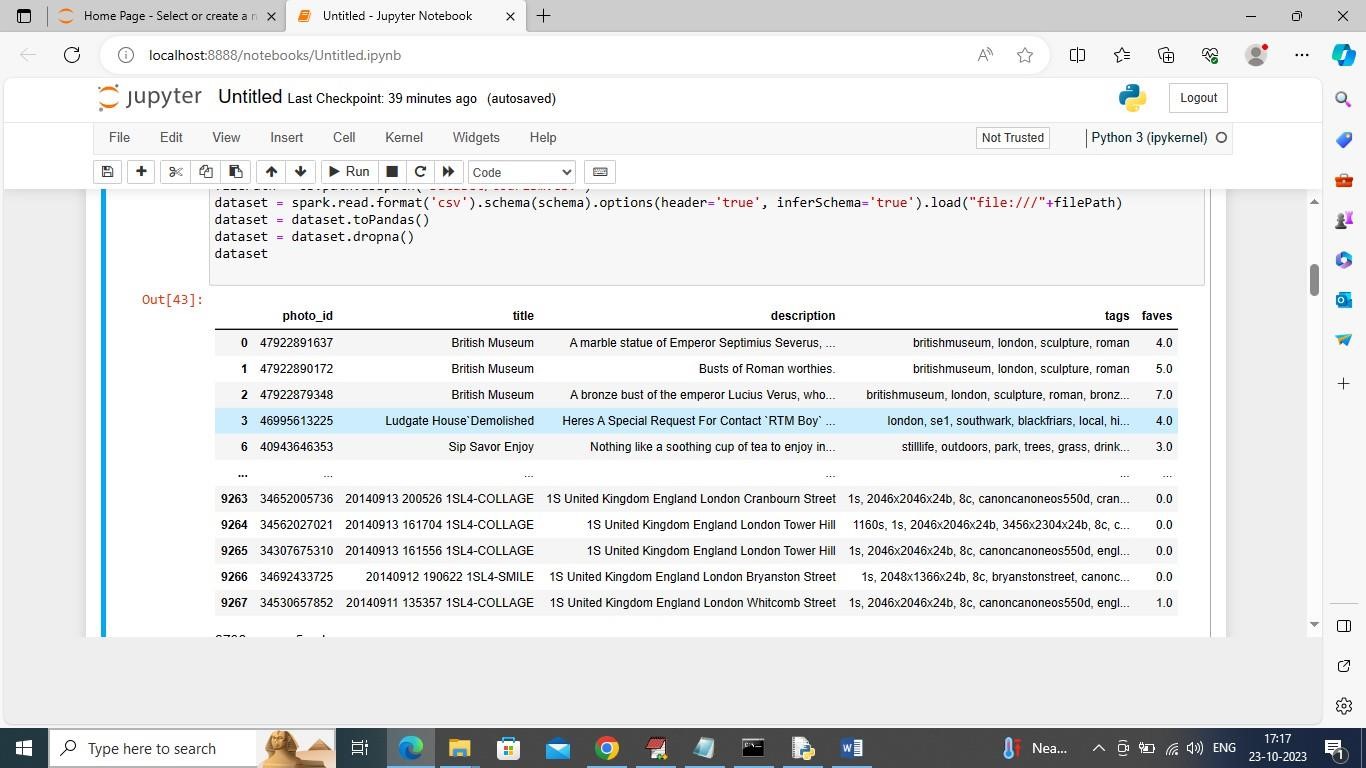
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments.



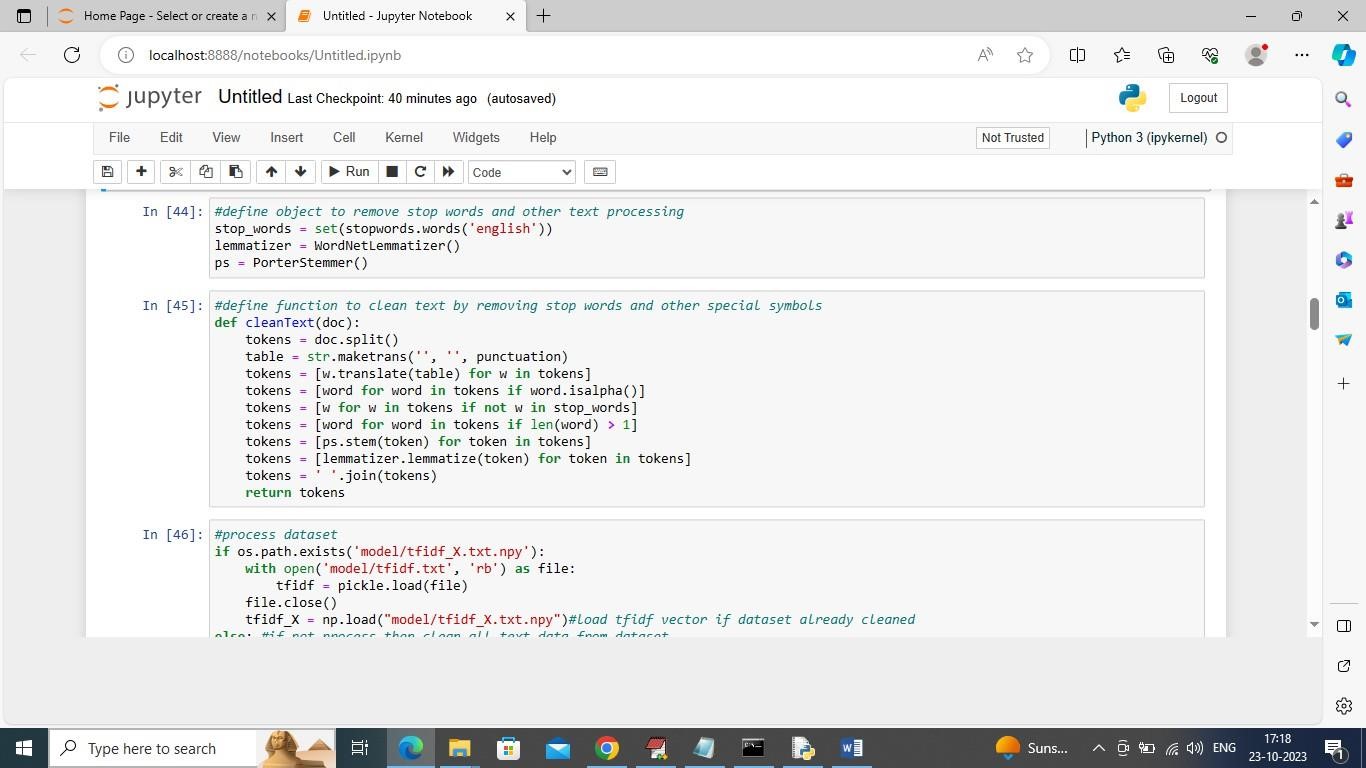
In above screen importing spark, NLP (natural language processing API to remove stop words, special symbols from geo tag text dataset) and then importing KMEANS and other classes.



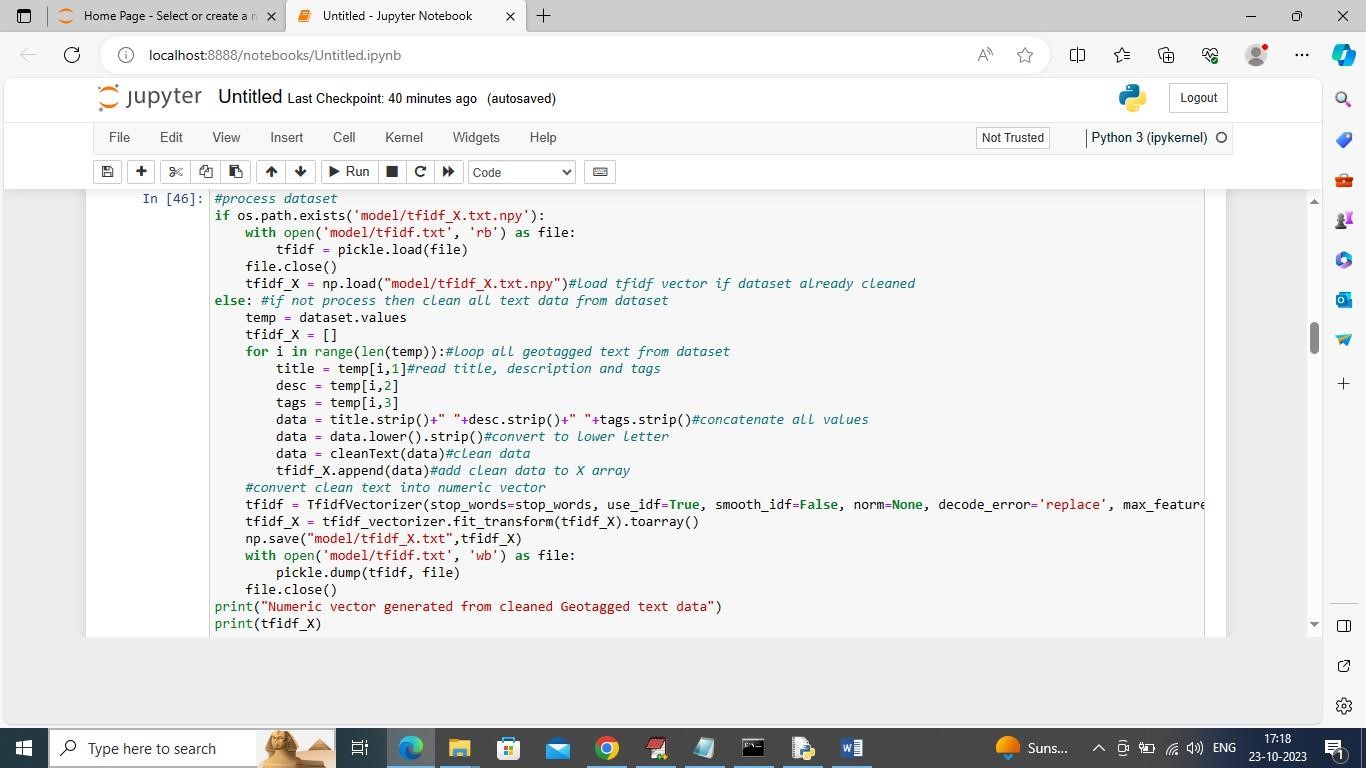
In above screen using Spark we are loading Tourism dataset and after loading will get below output



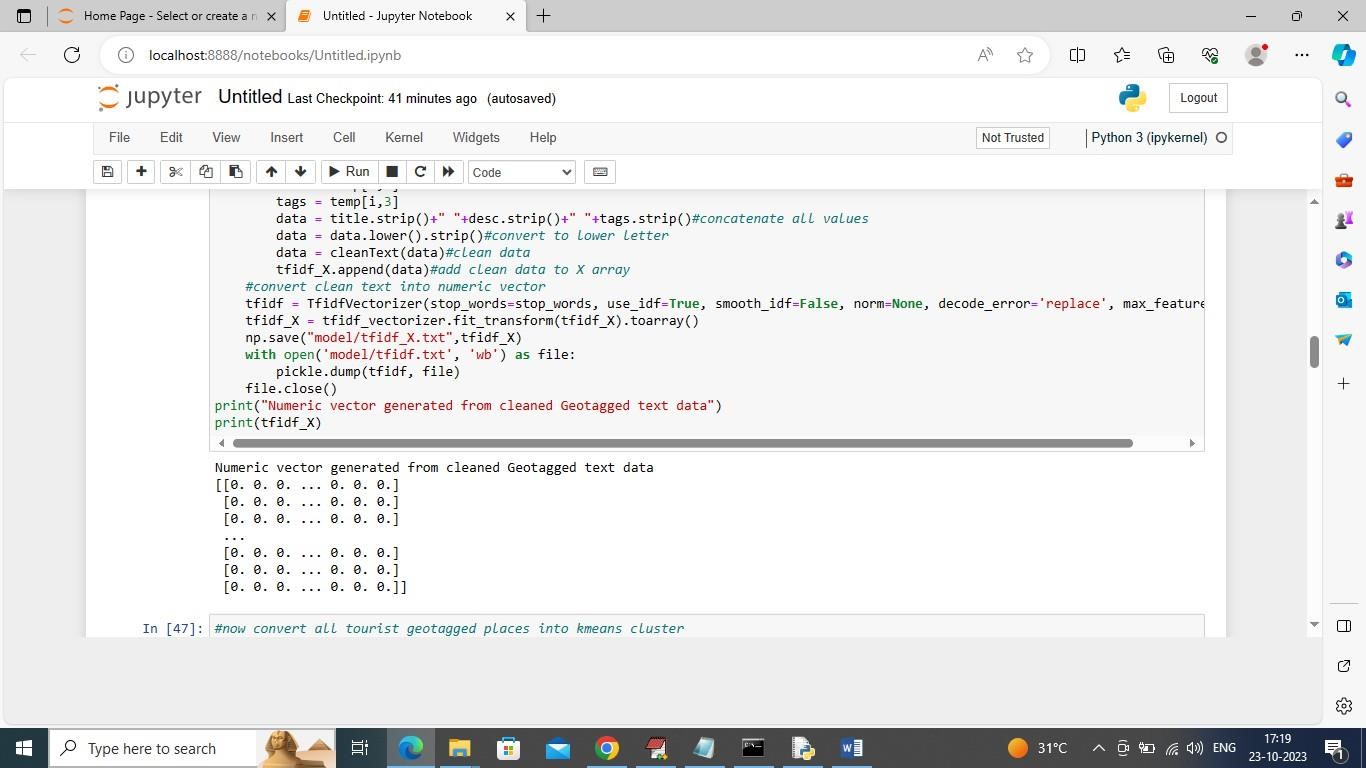
In above screen we can see loaded dataset values



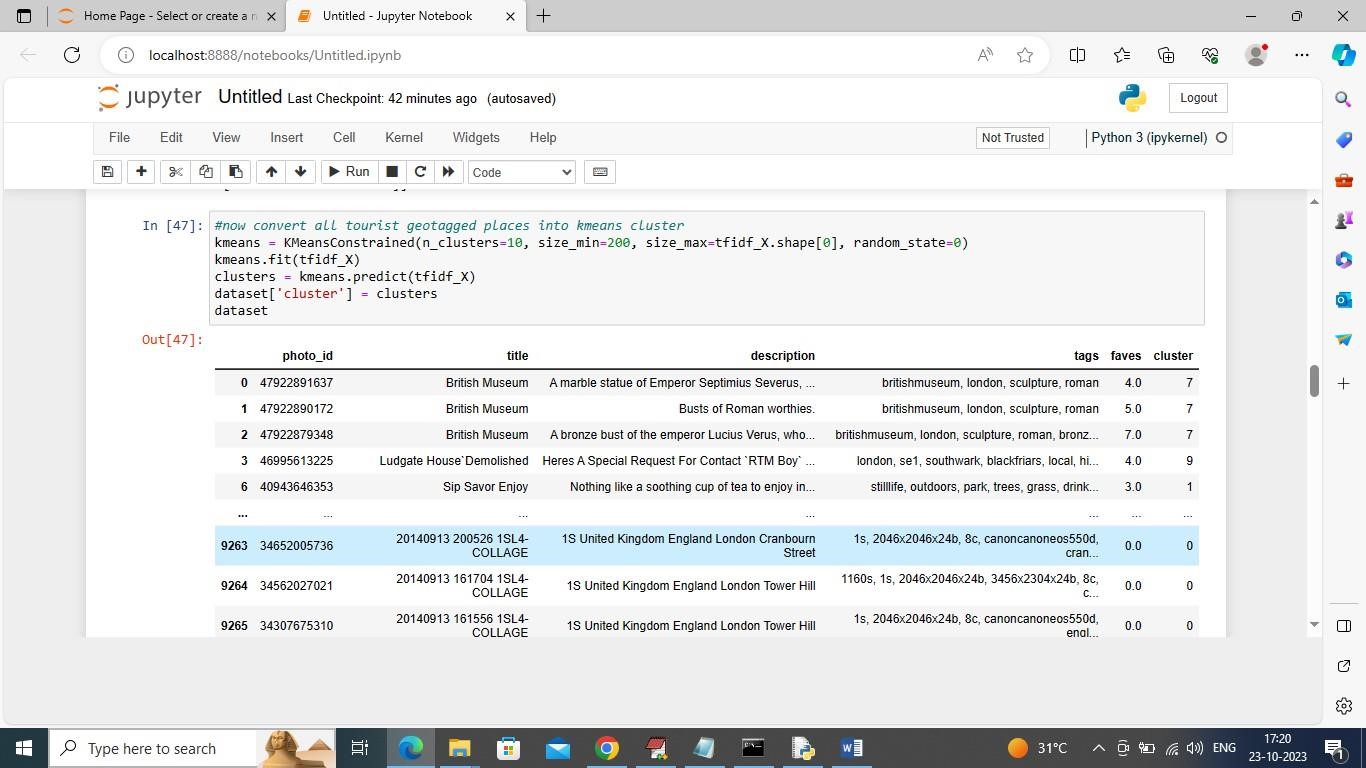
In above screen defining function to clean geo tagged text data



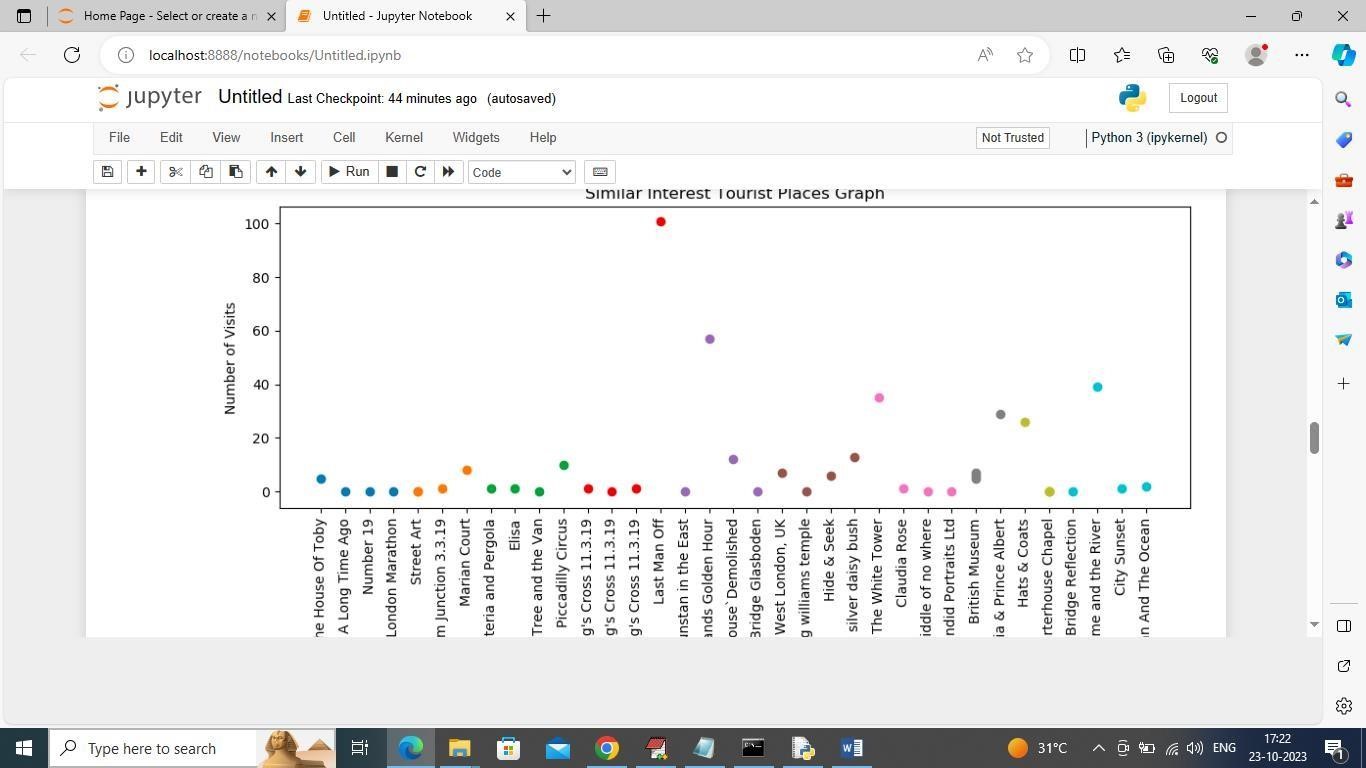
In above screen reading all Geo Tagged text data and then converting all clean text data into numeric TFIDF vector and this vector contains average frequency of each words and if word does not contains then vector will have 0 and by using this vector KMEANS will perform clustering.



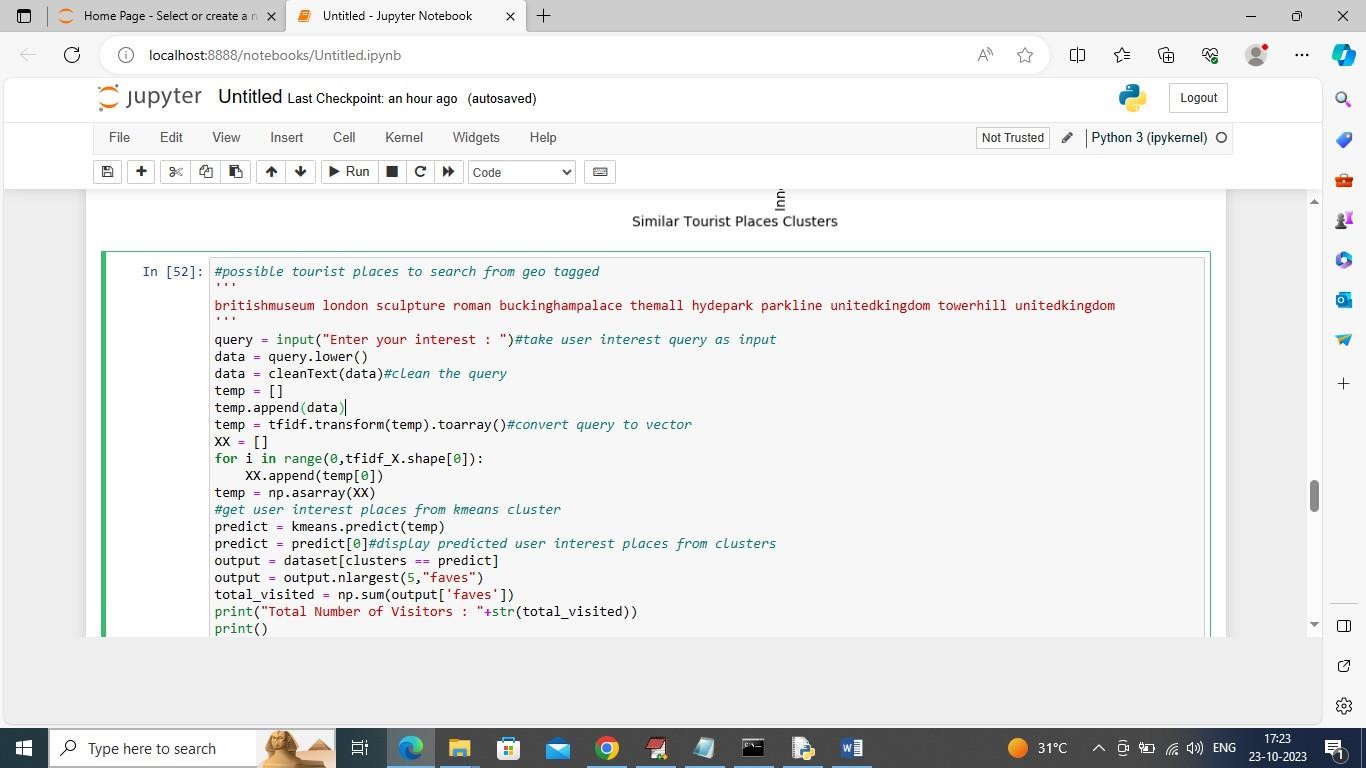
In above screen displaying TFIDF vector for few rows from dataset



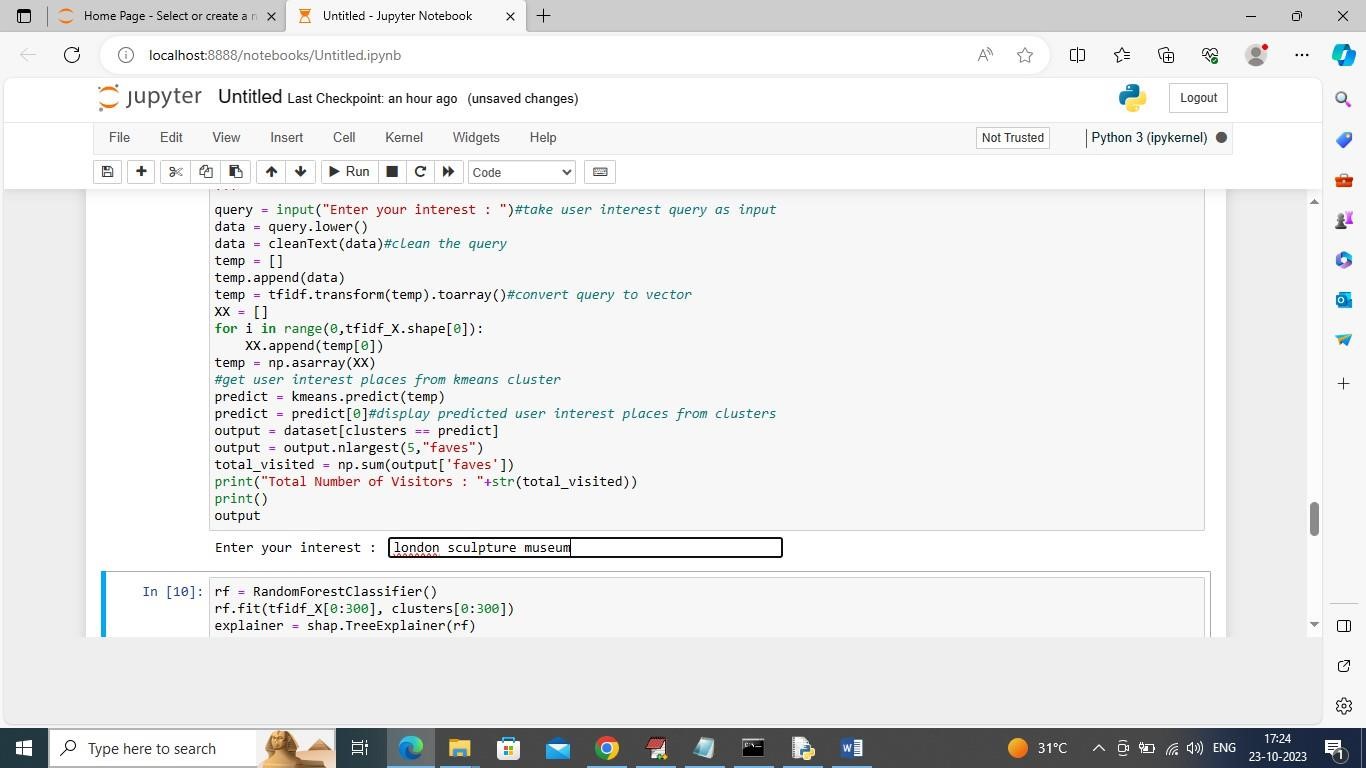
In above screen we performed clustering on all text data and after clustering in output last column we can see which row or user place goes to which cluster. In above table in last column we can see cluster label for each records. In above clustering we have created 10 clusters so all rows will be distributed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



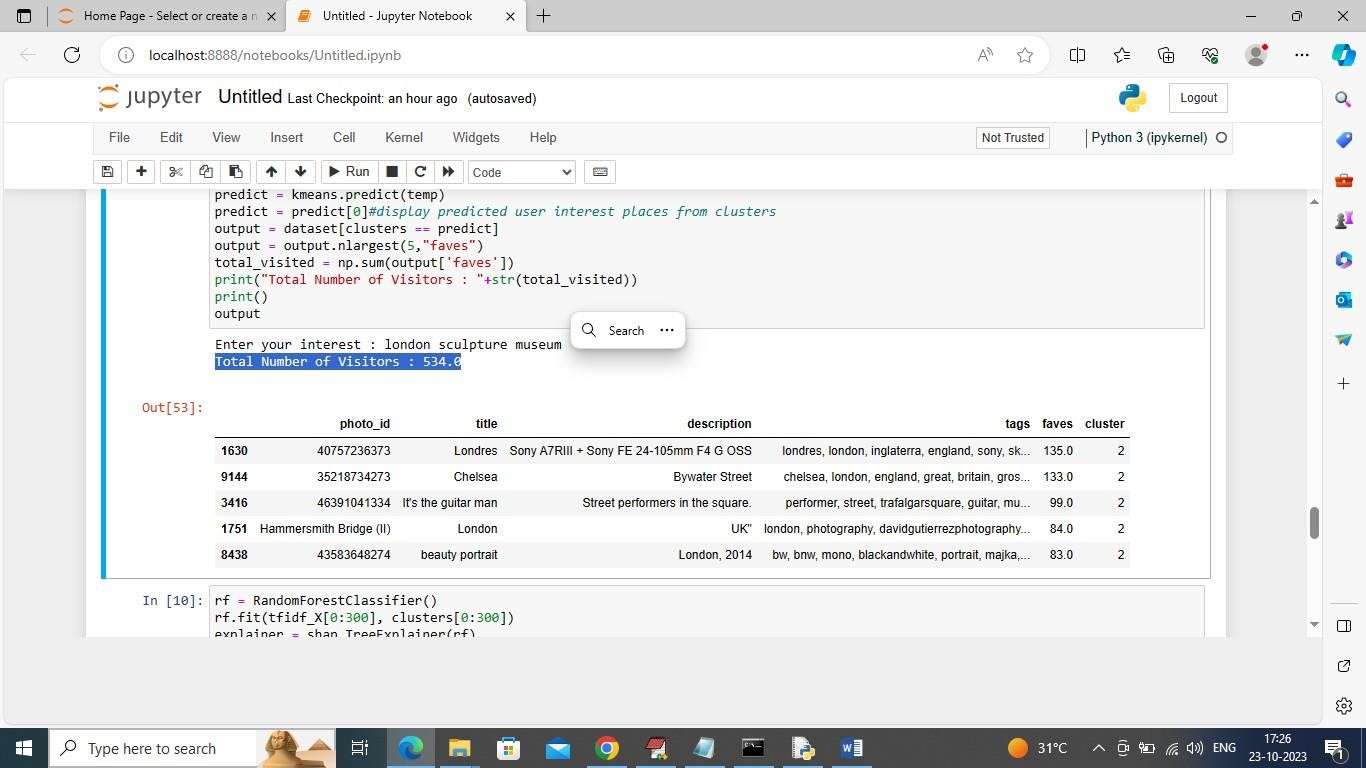
In above graph x-axis represents place names and y-axis represents Number of visited and small dots represents number of time place is visited.



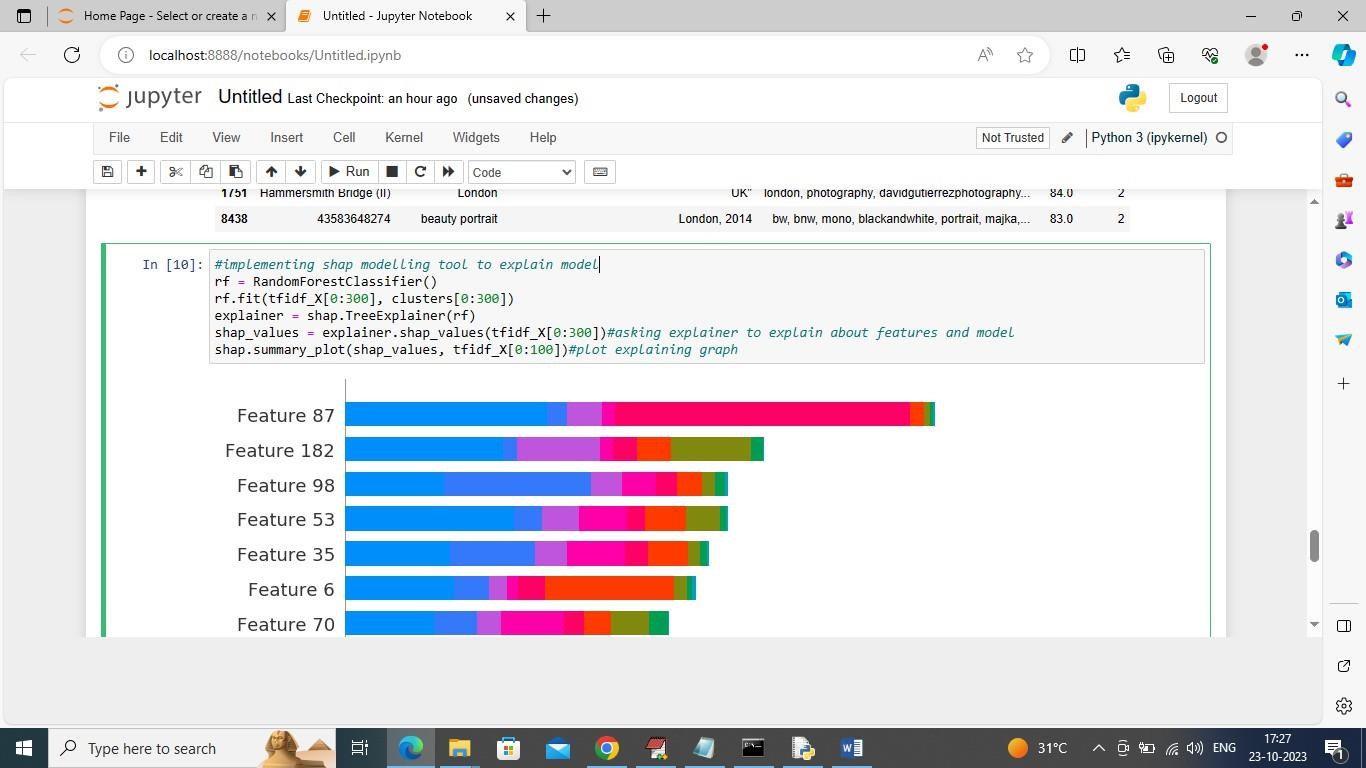
In above screen we define function which will read user query and then recommend similar places from cluster based on user interest query.



In above screen in Text Box I gave query as ‘london sculpture museum’ and press enter key to get top 5 recommended places like below screen.



In above table we got top 5 recommended places from cluster 6 and in blue colour text we can see number of users visited those 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.



The above screen we have added SHAP modelling tool to explain about the model who is using which features most for prediction.

**CHAPTER 7**

**CONCLUSION**

# 7.CONCLUSION

In conclusion, the "Analyzing Tourist Behavior using Big Data Technology" project represents a significant leap forward in the realm of tourism analytics, bringing forth a paradigm shift in how we comprehend and utilize tourist behavior data. Through the integration of advanced big data technologies, including Apache Hadoop or Apache Spark, our project aims to empower the tourism industry with a deeper and more nuanced understanding of the intricate patterns, preferences, and trends that define tourist activities.

The exploration of vast and diverse datasets, encompassing social media feeds, transaction records, and geospatial data, has allowed us to unravel valuable insights. Advanced analytics and machine learning algorithms applied to this data promise to not only capture current patterns but also predict future trends, offering stakeholders the foresight needed to make informed decisions in real-time.

The proposed system sets the stage for personalized and data-driven strategies that can enhance the overall tourism experience, providing destinations and businesses with the tools to tailor their offerings to the evolving needs and expectations of tourists. By bridging the gap between traditional analytics and big datadriven approaches, this project not only addresses the limitations of existing systems but also showcases the transformative potential of technology in shaping the future of the tourism industry.

As we move forward, the insights gained from this project can pave the way for data-driven innovation, improved resource allocation, and the creation of more immersive and tailored experiences for tourists. The "Analyzing Tourist Behavior using Big Data Technology" project embodies the convergence of technology and tourism, heralding a new era where data becomes a catalyst for informed decision-making, strategic planning, and the continual evolution of the tourism landscape.

**CHAPTER 8**

**FUTURE SCOPE**

# 8.FUTURE SCOPE

Here’s a concise future scope :

**1.Personalized Recommendations:** Tailor suggestions based on user preferences or history.

**2.Real-time Updates:** Continuously update tourist data to keep recommendations fresh.

**3.Geospatial Search:** Recommend places based on user location and nearby attractions.

**4.Multilingual Support:** Expand to handle multiple languages for global users.

**5.Better User Interaction:** Implement voice search or a graphical interface for ease of use.

**6.Hybrid Recommender System:** Combine content-based and collaborative filtering for better accuracy.

**7.Sentiment Analysis:** Factor in user reviews and sentiments for smarter recommendations.

**8.Improved Visualizations:** Use interactive maps and dashboards for better insights.

**9.API Integrations:** Integrate travel data sources (like TripAdvisor) for richer recommendations.

**10.Explainability:** Improve clarity on why certain places are recommended to build trust.

**11.Scalability:** Enhance performance for large datasets using distributed frameworks like Spark.

These enhancements can make the system more personalized, scalable, and user-friendly.

The future of analyzing tourist behavior using big data technology offers vast potential to revolutionize the travel and tourism industry. By harnessing AI, machine learning, IoT, and advanced data analytics, tourism organizations can provide personalized travel experiences, offering tailored recommendations and dynamic pricing based on individual preferences. Predictive analytics will allow for better forecasting of tourist demand, enabling more effective resource management and proactive decision-making. Additionally, the integration of IoT and real-time data can help optimize operational efficiency in destinations, from transportation systems to crowd control, while also enhancing smart destinations. Big data can improve sustainability by tracking the environmental impact of tourism and distributing visitors more evenly to prevent overcrowding.

**CHAPTER 9**

**REFERENCE**

# 9.REFERENCES

1. J. G. Bao and Y. F. Chu Tourism, Geography, Beijing:Higher Education Press, pp. I-5, 2013.
2. J. Bowden, "A cross-national analysis of international tourist flows in China", Tourism Geographies, vol. 5, pp. 257-279, Mar. 2003.
3. Y. F. Ma, T. S. Li and X. P. Liu, Research on Chinese Inbound Tourism, Beijing:Science Press, pp. 1015, 1999.
4. V. K. Jansen and R. Spee, "A regional analysis of tourist flows within Europe", Tourism Management, vol. 16, pp. 73-80, Jan. 1995.
5. G. Richards, "Tourism attraction systems: Exploring cultural behavior", Annals of Tourism Research, vol. 28, pp. 1048-1064, Apr. 2002.
6. F. J. Liu, J. Zhang and J. H. Zhang, "Analysis of basic method of collecting the spatial data of tourist flows: A study review and comparison both at home and abroad" in Tourism Tribune, Beijing, vol. 27, pp. 101-109, Jun. 2012.
7. J. Chen, B. Hu and X .Q. Zuo, "Personal Profile Mining Based on Mobile Phone Location Data", Geometrics and Information Science of Wuhan University Wuhan, vol. 39, pp. 734-738, Jun. 2014.
8. J. White and I. Wells, "Extracting origin destination information from mobile phone data", 11th International Conference on Road Transport Information and Control, pp. 30-34, 2002.
9. N. Caceres and J. P. Wideberg, "Benitez F G. Deriving origin destination data from a mobile phone network", Intelligent Transport Systems IET, vol. 1, pp. 15-26, Jan. 2007.
10. M. S. Iqbal, C. F. Choudhury and P. Wang, "Development of origin-destination matrices u sing mobile phone call data", Transportation Research Part C Emerging Technologies, vol. 40, pp. 63-74, Jan. 2014.
11. S. Isaacman, R. Becker and R. Caceres, "Identifying Important Places in people's live from cellular network data", Lecture Notes in Computer Science, vol. 6696, pp. 133-151, 2011.
12. V. A. Traag, A. Browet and F. Calabrese, "Social Event Detection in Massive Mobile Phone Data