VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

ASSYST

Submitted in Partial fulfillment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

RONJAY RUDRA (1CR16CS136)

S SUMEDHA (1CR16CS137)

SHIKHA TRIVEDI (1CR16CS157)

TANISHQ GAUTAM (1CR16CS172)

Under the Guidance of,

Dr. Jhansi Rani P

Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **Assyst** carried out by **Mr**. **Ronjay Rudra**, USN **1CR16CS136**, Ms. **S Sumedha**, USN **1CR16CS137**, **Ms**. **Shikha Trivedi**, USN **1CR16CS157, Mr**. **Tanishq Gautam**, USN **1CR16CS172,** bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree**.**

.

Dr. Jhansi Rani P	**Dr. Prem Kumar Ramesh**	**Dr. Sanjay Jain**
Professor	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled **Assyst** has been successfully completed under the guidance of Prof. Dr. Jhansi Rani P, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place:

Date:

Team members:

RONJAY RUDRA (1CR16CS136) ____________________

S SUMEDHA (1CR16CS137) ____________________

SHIKHA TRIVEDI (1CR16CS157) ____________________

TANSIHQ GAUTAM (1CR16CS172) ____________________

ABSTRACT

Tourism is an important sector for economic development and a potential application area of use of recommender systems. Recommender Systems are computer-based tools, which attempt to predict items out of large pool a user may highly likely be interested in, and to suggest him the best one. They also support systems helping users to find and/or to make choices about items that match their preferences and interests. To achieve these tasks, RS rely on characteristics and attributes of users and items as well as algorithms, which implement specific strategies and approaches, to generate a recommendation to a target user

Recommender systems are meant to be an important solution to the data overload problem that persists today in World Wide Web. The job of the recommender system is to provide the consumer with a selection of products or content which suit his/her needs so that the users are relieved from the herculean task of browsing through enormous number of web pages.

Through this project, “ASSYST”, we have used useful information from various data collected by Ministry of Tourism and other organizations working in the field of Tourism sector. Our project analyzes the data, trend of tourists coming to India in accordance to different indicators like age, seasons, interests and people accompanying to build a recommendation system to help more tourists who are yet to plan their visit. Our project helps tourists coming to India to act as their own travel agent and plan their trip with effective recommendation provided and well provided description of places that are to be visited.

ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing us a platform to pursue our studies and carry out our final year project.

We have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

We would like to thank **Dr. Prem Kumar Ramesh,** Professor and Head, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

We consider it a privilege and honor to express our sincere gratitude to our guide

Dr. Jhansi Rani P, Professor, Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of this review.

We also extend our thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged us.

Finally, we would like to thank our parents and friends for all their moral support they have given us during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
RS	Recommendation System
TRS	Travel Recommendation System
IT	Information Technology
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
POI	Point of Interest
SVD	Singular Vector Decomposition
SVM	Support Vector Machine
IFD	Information Flow Diagram
UML	Unified Modelling Language
ADL	Architectural Description Language
BRS	Buisness Requirement Specification
SRS	System Requirement Specification
CSV	Comma Separated Value
BSD	Berkeley Source Distribution
NMF	Non-negative Matrix Factorization

Chapter 1

Introduction

1.1 Relevance of the Project

The tourism field is one of the most potential application areas of Recommender Systems. From the point of view of tourism operators and service providers, employing Recommender Systems could boost tourist flows and increase revenue by recommending, at the right moment and when they are at the appropriate location, suitable items to potential tourist consumers. On the other side, from the tourists' point of view, Recommender Systems could be a valuable help while preparing a trip or searching a service among many destinations, numerous attractions and activities. The use of Recommender Systems could help tourists to save time and energy while searching for a trip or services that match their preferences and interests.

Travel Recommender System is a recommendation system which is used by the tourist and travelers to fulfil their needs which makes user to take decisions easy like deciding the travel destinations, finding nearby Point of Interest (POI), restaurants, shortest distance to travel, accommodation. Generally, travel recommender system is of different types like generic type recommender system, personalized recommender system. Generic type of recommender system would consist of destination details like tourist attractions near Kashmir. Personalized recommender system would require personal preferences like gender, type of vacation, number of people, number of days and many more.

In the tourism field, recommender system aims to match the characteristics of tourism and leisure resources or attractions with the user needs. TRS frequently combine multiple types of recommendation techniques such as the use of stereotypes (standard tourist segments), content-based and collaborative filtering techniques, personalized and ontology-based approaches.

Tourism has always been a major economy booster for India. Country's tourism industry can be made more flourished with all added benefits to the country. Every year crores of national revenue are generated from tourism industry with additional benefits of exposure to the world, globalization and exchange of culture with foreigners. This gives an obligation for developing tourism and taking it to new heights with increased efforts.

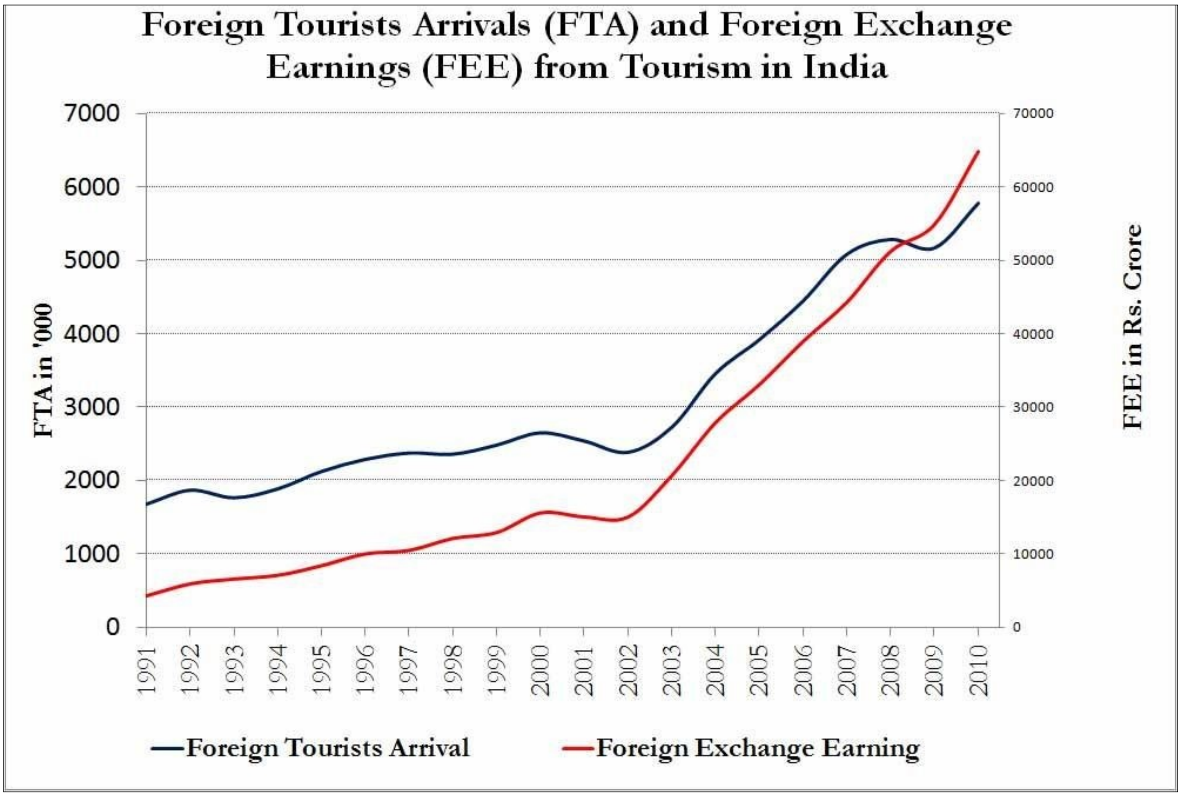


Fig 1.1. Tourist arrival & Revenue earned in India

In this technological world which is moving each and every day towards more and more sophisticated technology, especially achieving its recognition as an “IT world”, introducing IT in Tourism Industry can be very fruitful. Our country is a developing country moving towards the era of modernization and technology but has not achieved much.

Regarding the technologies introduced in Tourism industry in India, all we can get is the website maintained by Ministry Of Tourism, travel agencies with well adapted technologies like online booking of flights, hotels etc. for tourists but are serving for the purpose of pure business, information maintained in sites like Wikipedia, which are not enough and other similar technological acquaintances which are not quite adequate to make the tourism industry as e-tourism. There is certainly a need to get this whole thing to the new level, to make tourists coming to India feel as if this destination is more familiar. There is definitely a room to get all these developments possible, to let the guests know us globally, making India easier destination, helping them to know India, entice them with the culture and beauty, to let them know what the fellow tourists of different countries who have already visited think about our country, and to make India reachable globally from any part of world though not physically or infrastructurally but technologically.

The project “Assyst” could help in promoting tourism industry of India. And along with that, this project as a part of this academic project, helped us to use knowledge we have learned practically and learn new techniques and trends to develop an application and help us in the academic side too. So, “Assyst” basically aims in fulfilling both academic and social responsibility and help both tourists and concerned authorities.

1.2 Problem Statement

As technology has advanced, we have learnt to automate much of our daily life to be faster and more luxurious. After much research we realized that travelers tend to lose valuable time searching for locations and events manually. We aim to automate the process of trip planning.

According to Google, 85% of travelers decide on activities only after having arrived at the destination. Users visit multiple sites online while deciding their destination and planning activities. Majority of travelers don’t want to hire travel agents, even app-based agents like journey’.

Assyst aims to aggregate and improve upon multiple sites/application such as trivago, expedia and travel blogs to curate itineraries for travelers to effortlessly discover new and interesting places.

1.3 Objectives

Basically, Recommender system as software tools and techniques provide users with suggestions for items a user may wish to utilize. And basic theme of Assyst is to function as a recommender system. With this, objective of the project can be inferred as:

- To develop a Tourism Recommendation System based on their interests and past records.

Specifically, we aim to fulfil the followings:

- To design and develop a questionnaire for data collection for a case-study city.
- To identify features and data-processing techniques for the proposed system.
- To develop an interactive and adaptive user interface for the proposed TRS.
- Enhance tourist decision-making.

1.4 Scope

Our project "Assyst" is basically a tourist recommender system suggesting places to visit in India based on personal interest and characteristics. Our system consists of information about many places in India which are worth visiting. Additionally, with the information of age, interests, preferential seasons and past experiences of tourists coming to India our system can provide results that are accurate to much extent. As mentioned, the needs of IT and computer applications in field of Tourism and lack of the same in existing trend, this project has a lot of significance to cater the need and provide the effectiveness to existing trend of Tourism industry.

Such a system with features like information about the places, recommendations to places on the basis of parameters like age, seasons, categories, personal interests like natural beauty, adventure trekking etc. can help tourists and be a guide to them to explore India and its beauty. With the system, India can be made more reachable to number of foreigners from different countries with more ease, know about the trends of other who have already visited and at the same time explore the area of their interest all at the same platform.

And it is a general truth that anything made specifically for a person carries more significance than something which serves for general.

With all the problems mentioned, significance of developing a system like “Assyst” seems quite convincing and it is a requirement rather than just another system introduced in the area. From academic point of view, we got to learn lot of techniques, practical implementation of knowledge, research on tourism status of India, and eventually make a project we can look up with pride which adds even more importance to the project. We hope in the future this project can be a lot more importance in Tourism industry field and can be of some value to contribute in the same.

1.5 Methodology

To give user an efficient and fast recommender alongside a proper website with information about famous as well as yet to be discovered place, what can be better than a system like “Assyst”. This system well addresses the problem of need to boost IT development in tourism, give a proper and efficient recommendation as well as also act as an informative site with famous places as well as places worth visiting but less famous.

The to-be-built tourist recommendation system helps to return curated search results of tourist spots based on the user’s previous travel history, their preferences, their budget, and choices along with the current season.

Different people have different selections when they look for a place to go to. Some prefer to roam around in the heart of nature, while others want to have an adrenaline rush while having a go at fun, adventurous sports, while others have the desire to be lost in the long-forgotten stories of history.

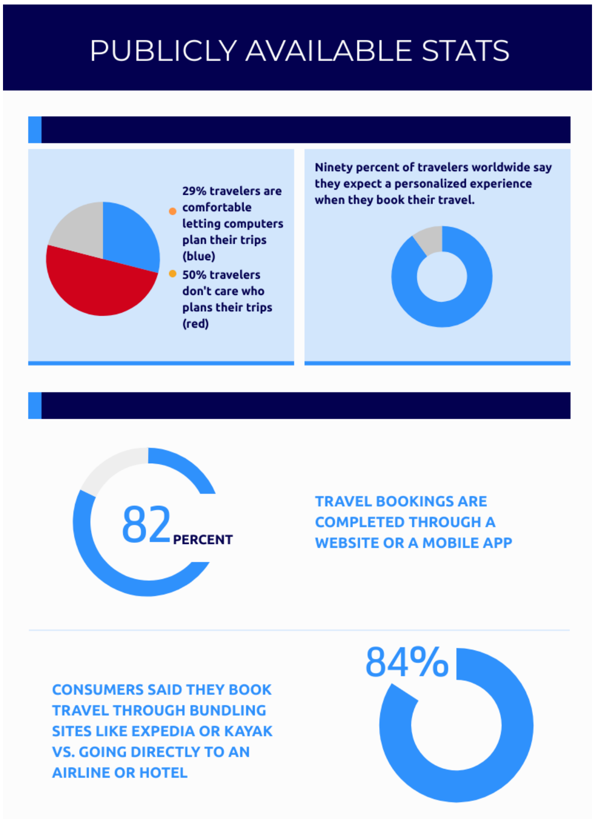
While some might prefer to do it with their friends, others might prefer the company of their better half, others might want the love of their family alongside them. Everyone has a varying budget. Some might prefer a short vacation to a nearby place while others may want a long vacation with no travel bounds.

Chapter 2

Literature Survey

As this project is centered around the user, we require information about what the intended users expect from the model. The main observations that we were looking for was user preferences such as seasonal preferences, trip duration, and other similar factors. A literature survey was conducted to obtain data about multiple opinions and views over proposed idea. A Google form was circulated and over 80 responses were recorded. The users were asked about various travel planner aspects such as their willingness and concerns to use one. Questions about their travel preferences such as duration and factors affect their plans.

2.1 Publicly available statistics



29% travelers are comfortable letting computers plan their trips (blue)

50% travelers don't care who plans their trips (red)

Ninety percent of travelers worldwide say they expect a personalized experience when they book their travel.

TRAVEL BOOKINGS ARE COMPLETED THROUGH A WEBSITE OR A MOBILE APP

CONSUMERS SAID THEY BOOK TRAVEL THROUGH BUNDLING SITES LIKE EXPEDIA OR KAYAK VS. GOING DIRECTLY TO AN AIRLINE OR HOTEL

Fig 2.1 Publicly available stats

2.2 Statistics obtained by us

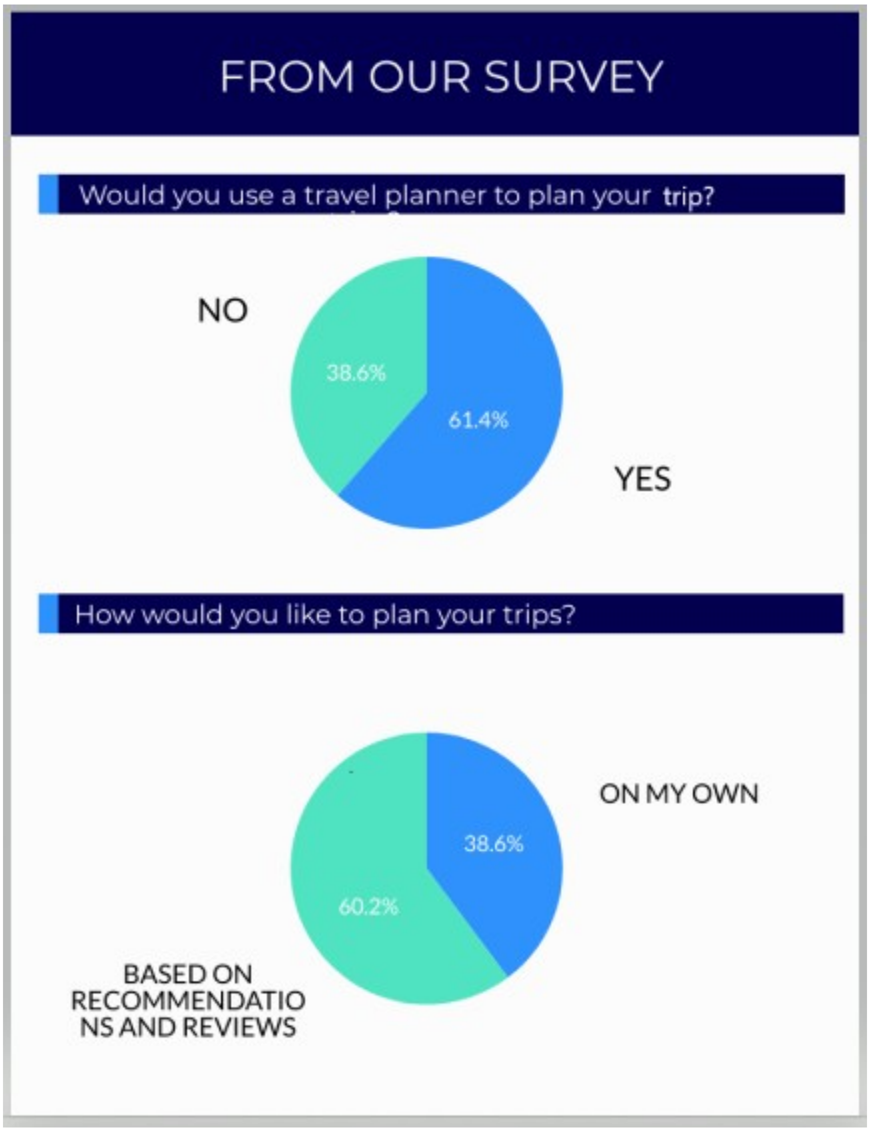
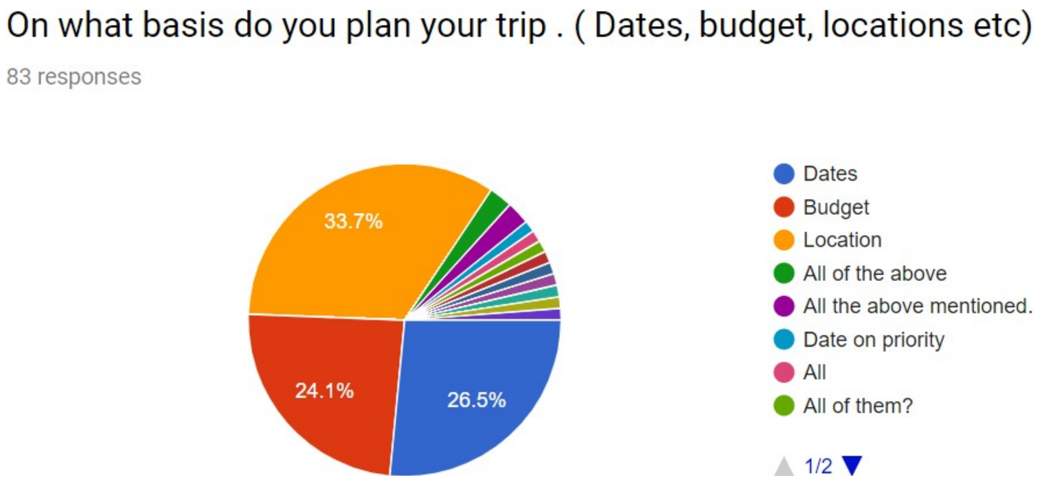
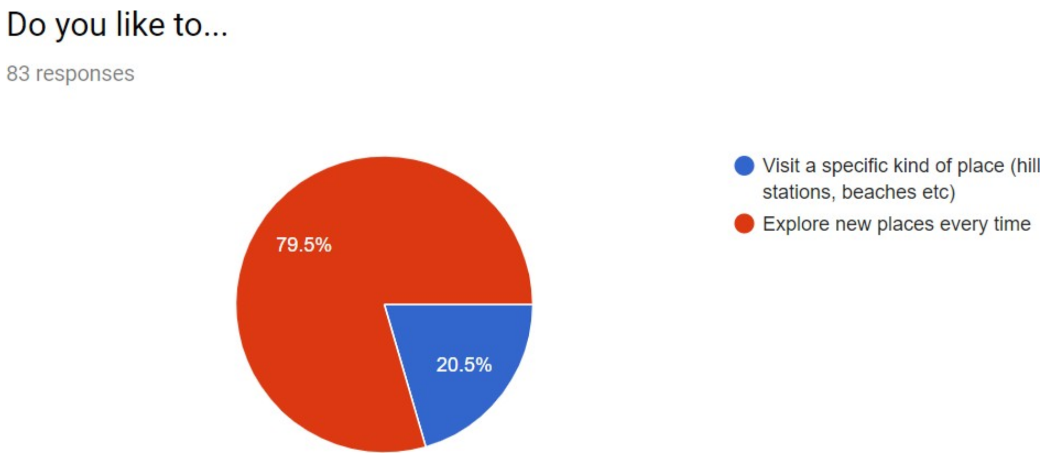


Fig 2.2 Basic Questions from our survey


On what basis do you plan your trip . (Dates, budget, locations etc)

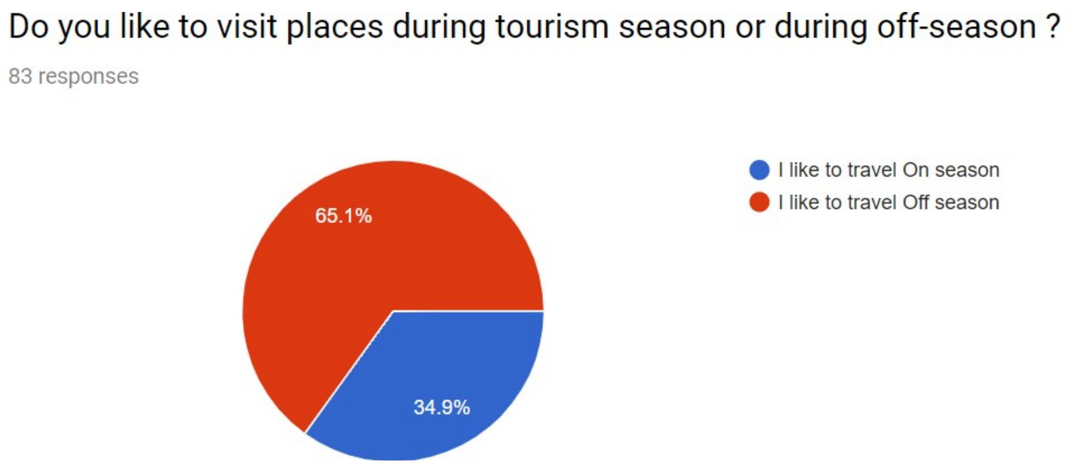
83 responses

Fig 2.3 Stats based on factors affecting trip planning


Do you like to...

83 responses

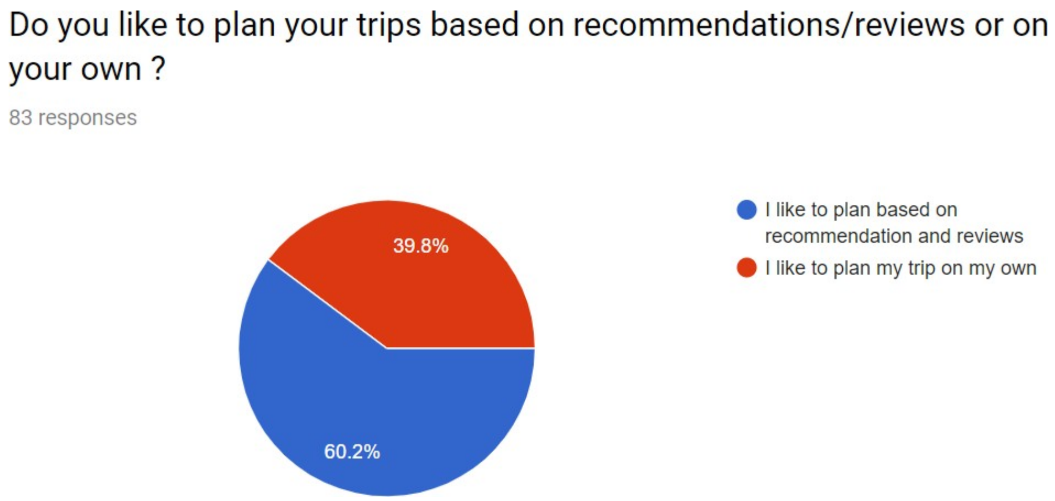
Fig 2.4 Stats based on the personality of a user



Do you like to visit places during tourism season or during off-season ?

83 responses

Fig 2.5 Stats based on tourism season preferences



Do you like to plan your trips based on recommendations/reviews or on your own ?

83 responses

Fig 2.6 Stats based on trip planning

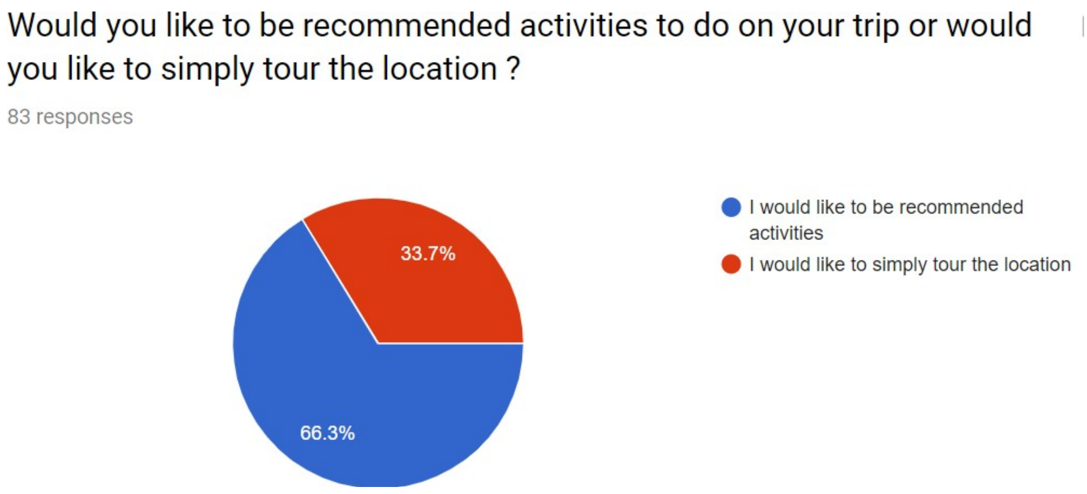


Fig 2.7 Stats based on activity preferences

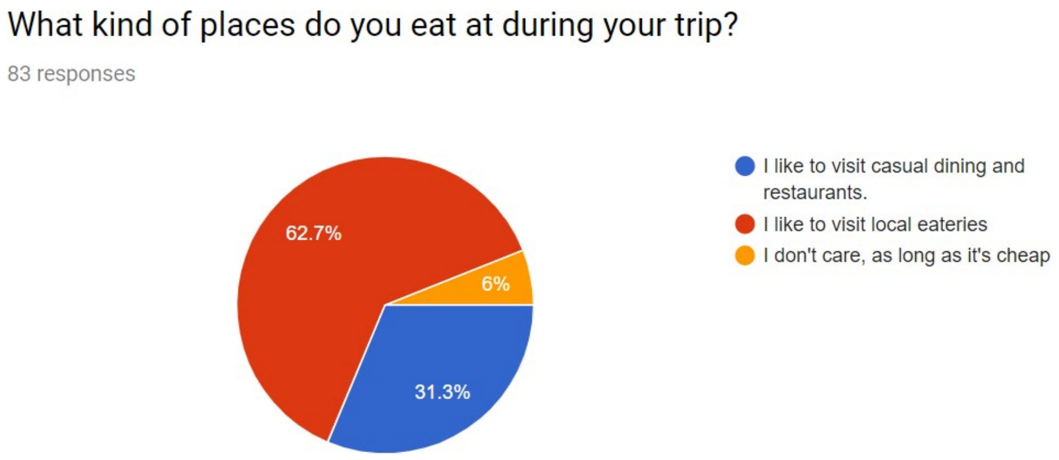


Fig 2.8 Stats based on dining preferences

We also asked responders whether they have used trip planners in the past, and if yes, their experience. 60% of the responders have not used trip planners. 20% said that they had a good experience with trip planners, because it was organized and flexible.

A major factor that goes into planning trips is duration of the trip in question. We asked users their ideal trip duration in the google form. 70% of the responses gave a range from 4 to 7 days, which gave us the observation that the most ideal trip duration would be around one week.

When asked about frequent challenges faced when planning a trip, the most challenging aspect was found to be finding new places to visit. Users also mentioned budget and time of travel in their answers.

Lastly, we asked responders for any suggestions for our proposed idea. Most of the responses asked for customization of trips and consideration of budget. Several responses asked for recommendations on eateries and hotels in the travel destinations.

Chapter 3

System requirements Specification

3.1 Functional Requirements

The functional requirements for the project are mentioned below:

- The website should be able to login the user when the he/she enters the username and password.
- On login, the user should be presented with the recommendation and the itinerary prepared for the user.
- On user signup, the website to automatically redirect the user to the questionnaire page where he/she can provide the website with valuable information regarding their past trips.
- The Machine learning model should be able to extract the best possible recommendations based on the user's interests.
- The places api should be able to extract different components of the itinerary in a quick and accurate manner.
- The user-interface should work in an efficient manner to provide the user with the fastest recommendations.
- The machine learning model should be able to provide high accuracy at all times.
- The system to should provide an admin page for the administrator to have control.
- The system should be able to take new data from the users to provide updated results.
- The system should provide efficient communication between server and client.

3.2 Non- Functional Requirements

System

Requirement is a process that identifies, describes and evaluates proposed system and selects the best system for the job. During the study, the problem definition is solved and all aspects of problem to be included in the system are determined. Size of project, cost and benefits are also estimated with greater accuracy. The result of feasibility study is simply a report which is a formal document detailing the nature and scope of the proposed solution.

Economical

The economic requirements of the project includes its economic appropriateness with respect to its presented output. If a project provides results of lower significance but requiring higher budgets then that project can't be economically viable. For the case of this project, the investment is not a lot in terms of economic aspects. The only expense was of time and skill and the return is large if the project is favored by potential customers. To get the desired processing speed once the project is fully developed, we will need a high-speed GPU to keep the application up and running but in the developmental stage we can stop worrying about that.

Technical

The technical requirements of the project deals with the availability and its actual implementation ability with the existing tools and techniques available in the software market world. The following are the notable points about the technical feasibility of the project:

- Chose a template best suited for our needs and modified the template to shape it according to the project's demands.
- Used web development technologies like HTML5, CSS, JavaScript, and Bootstrap.
- Used Django for connecting the front end to the back end which is a high-level Python web framework that enables rapid development of websites.

- Coding in python to establish connectivity to the database (SQLite) by creating models, forms and views.
- Required dataset that are easily available over the internet almost free of cost on community development websites like Kaggle and in our case we build it using the information available over different government tourism websites of India.
- Implementation of decision tree algorithm, cosine similarity algorithm and SVD algorithm

With all these perspectives taken into consideration, the project is technically feasible to implement.

Behavioral

People are inherently resistant to change, but if the change tends to provide enough benefits such that it is worth accepting then the system possess behavioral feasibility. Our system benefits all variety of users as it is an effective tool to find places to travel to, based on your preferences. Secondly, this idea will help people to discover vacation spots according to their current inclination towards different categories like budget, their interests, people to keep them company, the season in which they are planning their travel and also according to their likes and dislikes about their previously travelled places. Hence, this system increases the value for money and also builds the trust of user in the system by providing the accurate results.

3.3 Hardware

The project will require a System with interactive user-interface, a system that supports latest web-browsers and technologies in order for smooth accessing of the project.

The hardware requirement has been identified as shown below:

- Processor- Intel Core CPU
- RAM- 4GB
- CPU- 32-bit or 64-bit Operating system

3.4 Software

- Atoms Text Editor
- Spyder (Anaconda)
- Django Framework
- SQLite Database
- Application of machine learning and distributed systems with various sub topics like hyper-tune, Tensor-Flow SDK framework and Scikit- learn pipelines
- Front-end Languages like HTML, CSS, JavaScript and Bootstrap
- Back-end Language – Python and its libraries used for ML

Chapter 4

System Analysis and Design

System analysis is the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. Another view sees system analysis as a problem-solving technique that breaks down a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose.

4.1 System Architecture

The software needs the architectural design to represent the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectural styles.

“Assyst” is a very simple yet effective system that addresses one of the problem people face while planning a trip. Our website can recommend cities to people which will be the most appropriate for their needs and requirements. These requirements are user’s current requirements and can be altered from time to time. The website for a guest user will contain general information, showing the most visited places in India and their reviews. Once the user decides to log in, he/she is asked to fill a registration form and once the registration is done, the user is directed to fill the questionnaire designed to get the optimal information about them. It asks the user to rate various categories like interests (for e.g. adventure, nature etc.), seasons (for e.g. winter, summer etc.), people they are planning a trip with and age group of people travelling. Along with this we take the rating of some place they might have visited in the past so that we can use them for collaborative filtering. We also ask for the budget they are ready to spare for this trip.

All this information helps us to build a strong recommendation for the cities they are likely to visit as it is the most suitable place as per their needs and wants. This is done using the SVD algorithm that extracts information from the models (database).

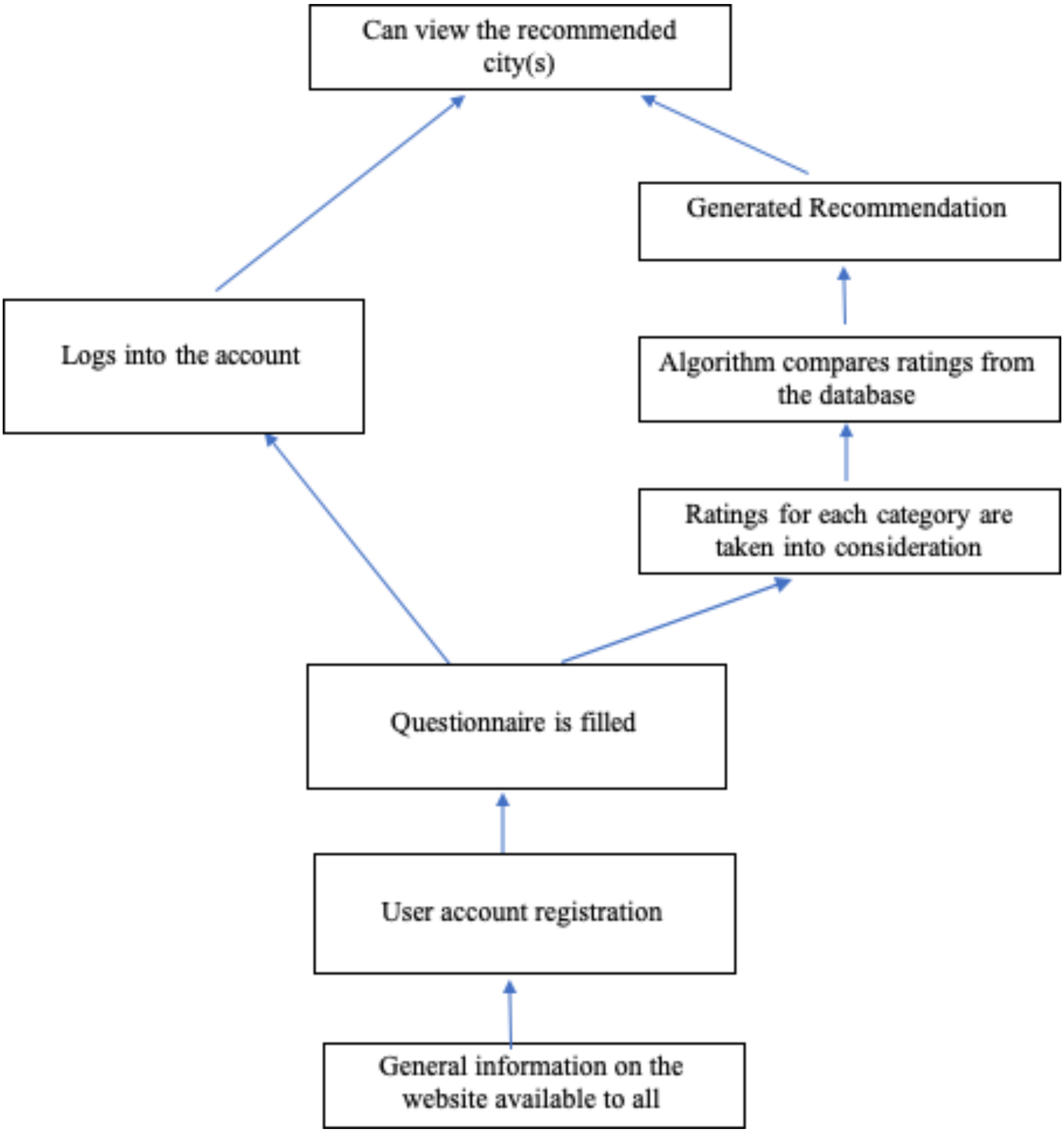

Can view the recommended city(s)
Generated Recommendation
Logs into the account
Algorithm compares ratings from the database
Ratings for each category are taken into consideration
Questionnaire is filled
User account registration
General information on the website available to all

Fig 4.1 System Architecture

4.2 Dataset / Flowchart

While dealing with machine learning projects it becomes inevitable to deal with large chunks of data. To deal with large amounts of data it requires proper organization of data with the help of various data structures. Various data structures are used to store data in various formats. For example, an array stores homogenous data while a list can store heterogeneous data too. To deal with the large amounts of dataset that we require for our project we have used the models used in Django framework. Django by default comes with SQLite database where we can store the relevant data that can be accessed by the algorithm. For this project we made two models, namely, City Database and Rating Database.

City Database contains the fields placeid (acting as its primary key), city, state and type of interests this city provides like adventure, nature, heritage etc. Other one that is the Rating Database contains the fields as userid, placeid (one from the City Database), rating, and the timestamp. Other than these data models we have a model for Questionnaire that contains all the preferences of the user in the form of rating for the corresponding field. Rating database acts as a link between City Database and the Questionnaire.

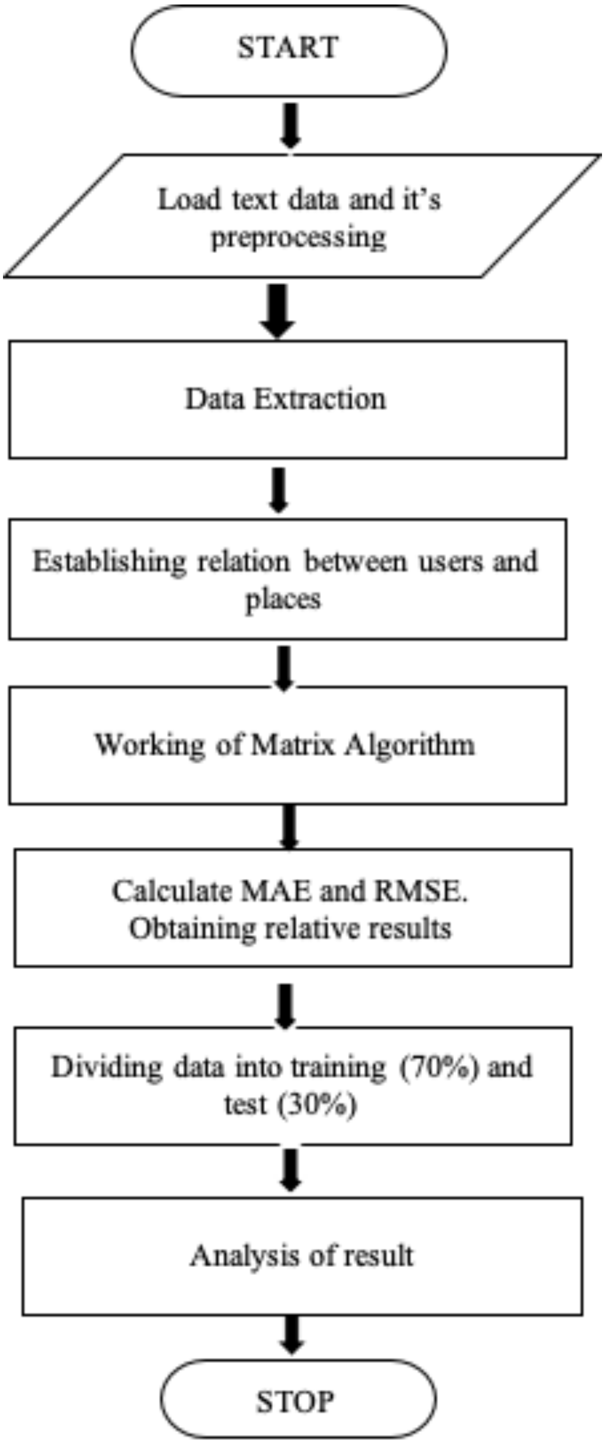

START
Load text data and it's preprocessing
Data Extraction
Establishing relation between users and places
Working of Matrix Algorithm
Calculate MAE and RMSE. Obtaining relative results
Dividing data into training (70%) and test (30%)
Analysis of result
STOP

Fig 4.2. Algorithm Flowchart

4.3 Entity-Relationship Diagram

An entity–relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).

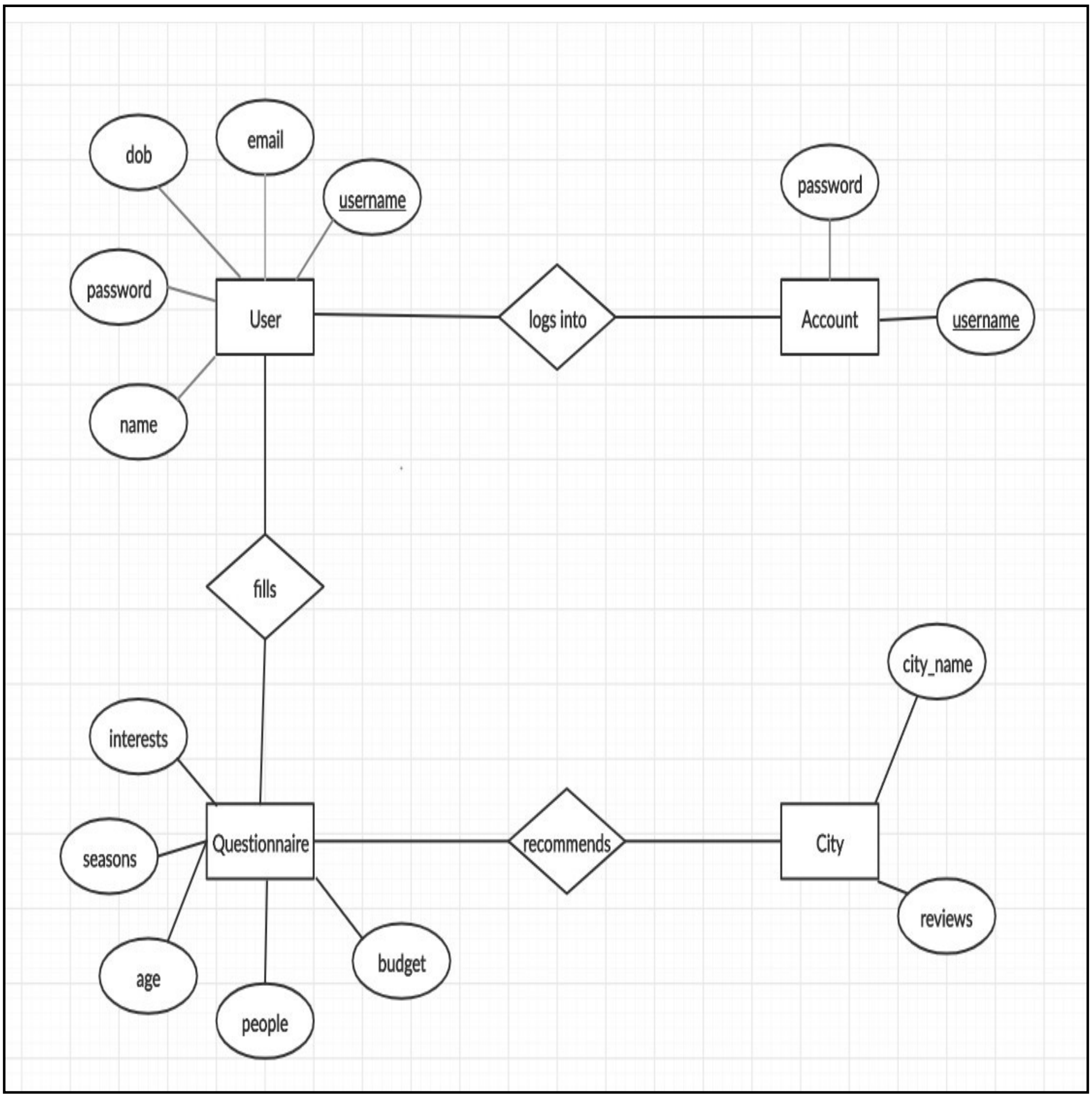


Fig 4.3. E-R Diagram of Travel Recommendation System

4.4 Activity Diagram

Activity diagram is defined as a UML diagram that focuses on the execution and flow of the behavior of a system instead of implementation. It is also called object- oriented flowchart. Activity diagrams consist of activities that are made up of actions which apply to behavioral modelling technology.

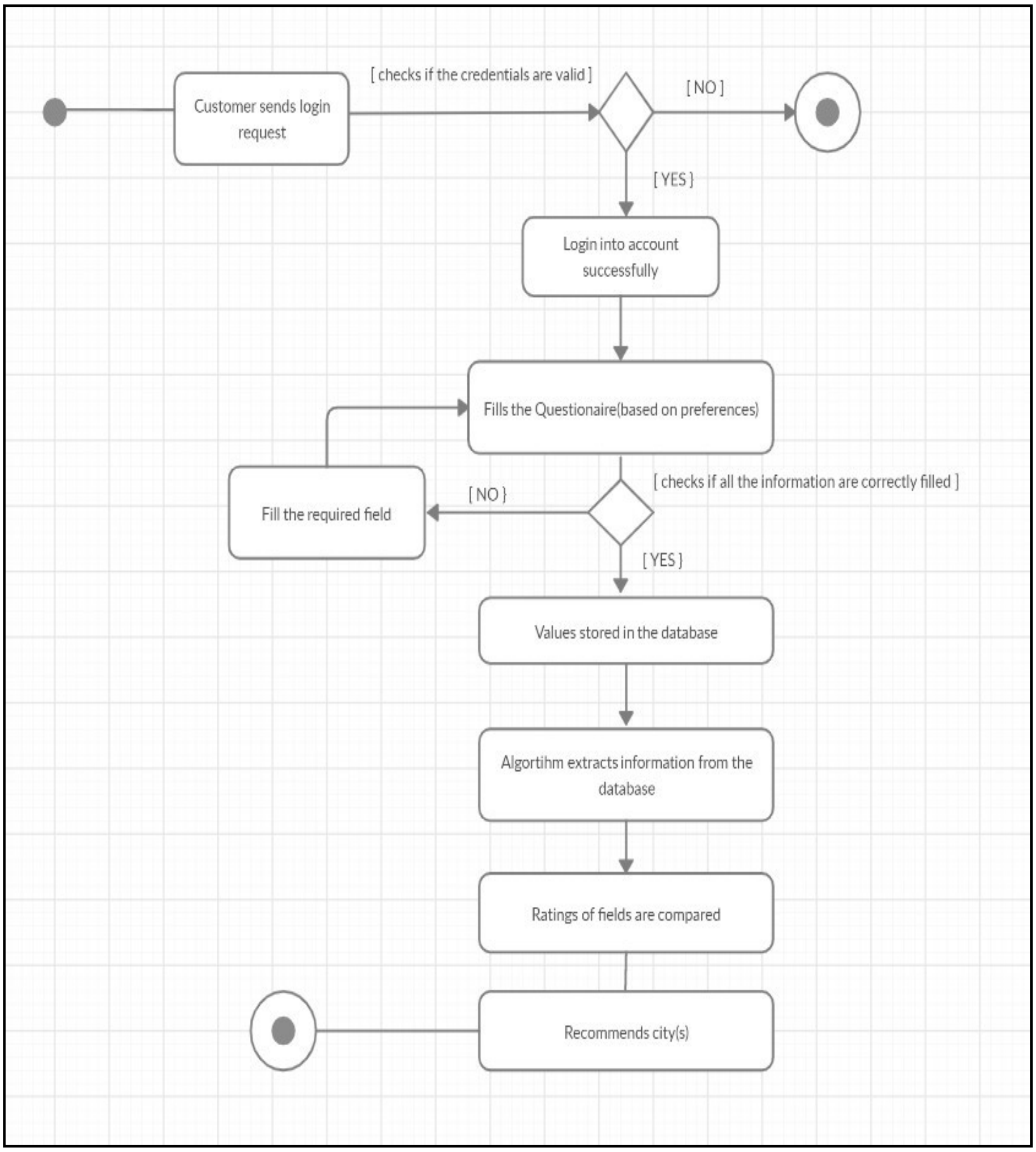

Customer sends login request
[checks if the credentials are valid]
[NO]
[YES]
Login into account successfully
Fills the Questionaire(based on preferences)
Fill the required field
[NO }
[checks if all the information are correctly filled]
[YES }
Values stored in the database
Algortihm extracts information from the database
Ratings of fields are compared
Recommends city(s)

Fig 4.4. Activity Diagram of Travel Recommendation System

4.5 Use case Diagram

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

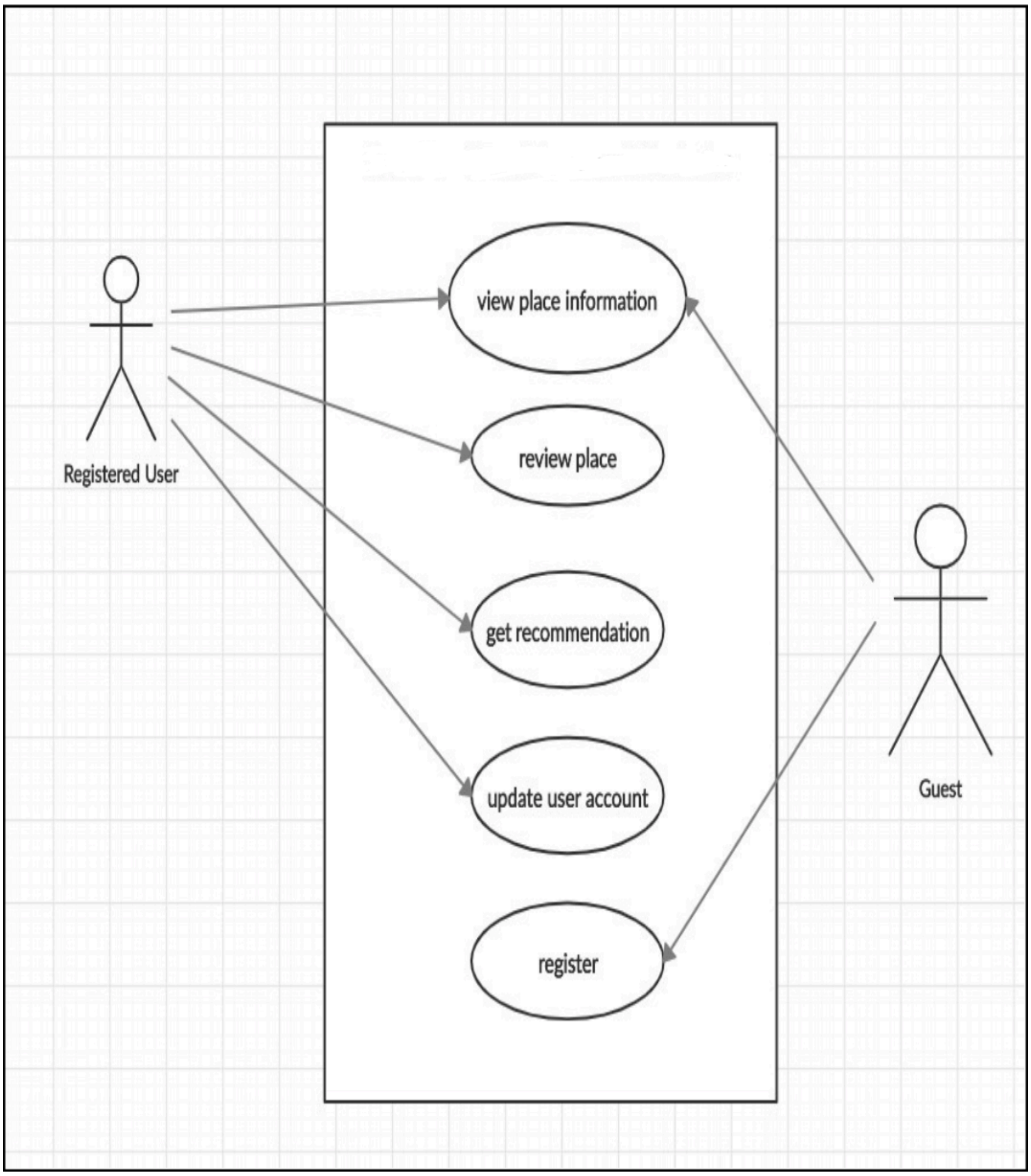


Fig 4.5. Use Case Diagram of Travel Recommendation System

4.6 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language. These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

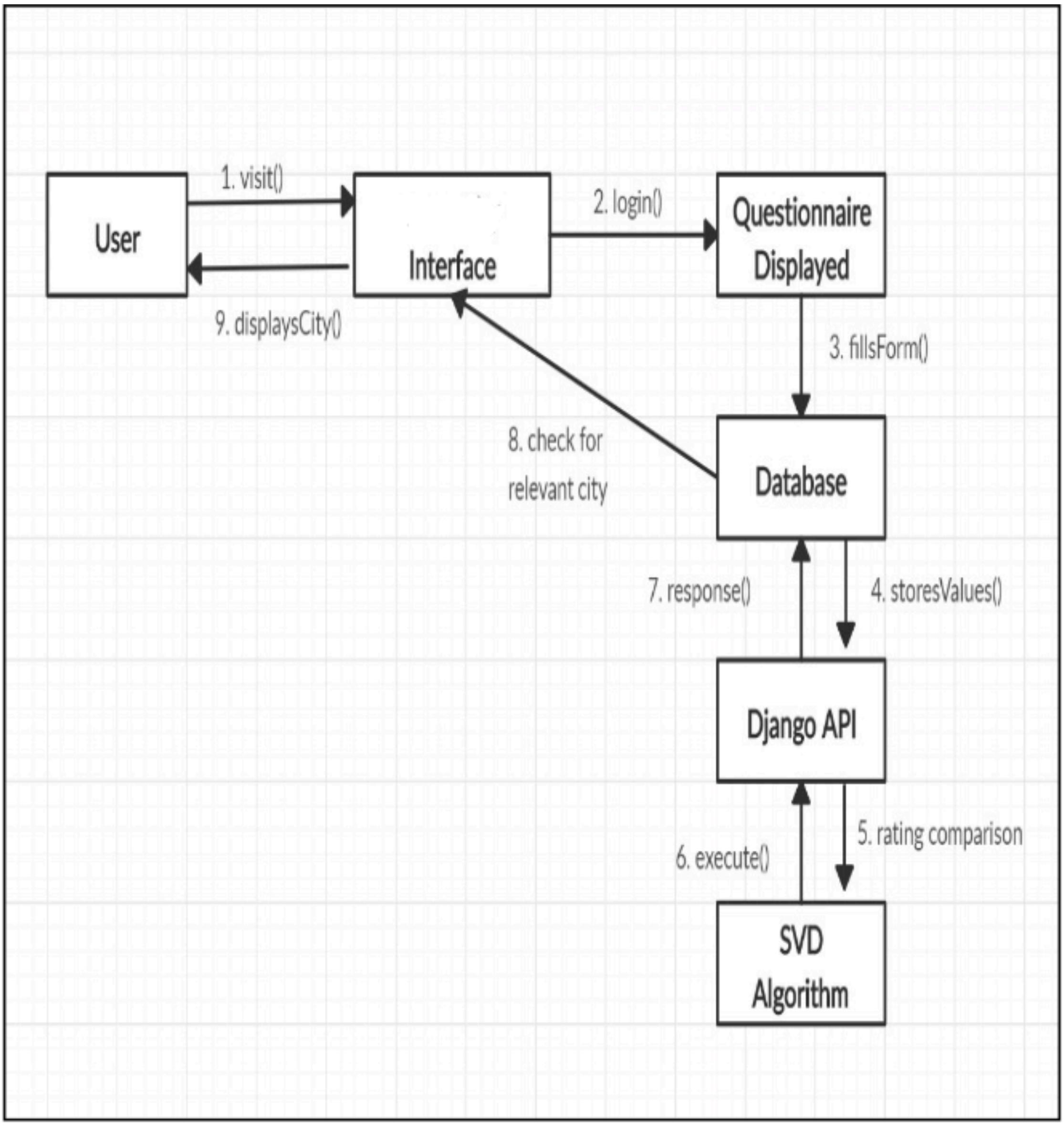


Fig 4.6. Collaboration Diagram of Travel Recommendation System

4.7 Class Diagram

A class diagram in the Unified Modelling Language 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 objects.

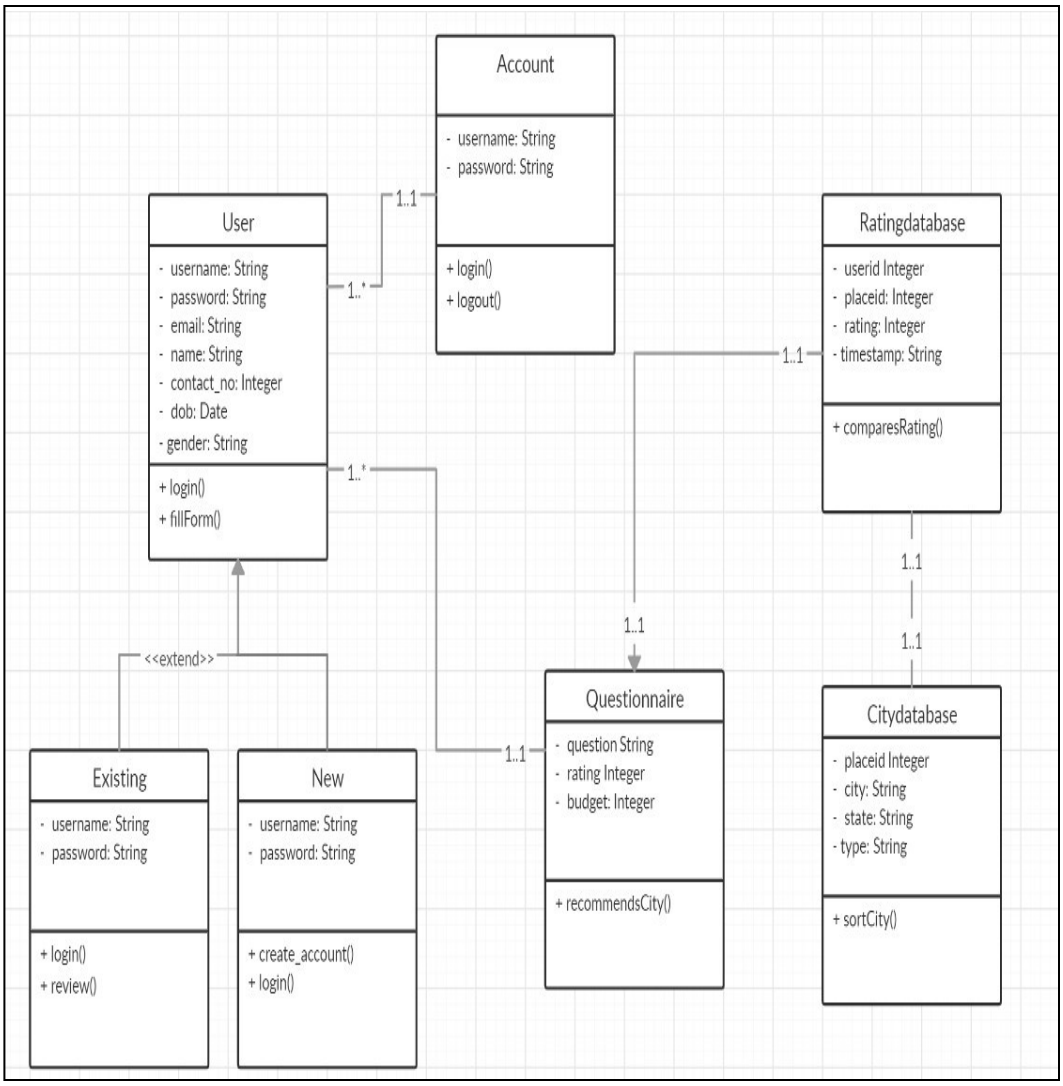


Fig 4.7 Class Diagram of Travel Recommendation System

Chapter 5

Implementation

5.1 Modular and Procedural Approach

Python provides various inbuilt modules for machine learning purposes. In this project we have used various python modules and libraries for various purposes. Since this is a ML project the libraries that we used are Surprise package and worked on Django framework. Secondly for mathematical computations and clustering purposes we used, pandas, NumPy, Sci-kit learn, SciPy and other libraries. For the development of GUI, we used front-end technologies.

5.1.1 Modules Used

5.1.1.1 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three- clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Pandas is mainly used for machine learning in form of data frames. Pandas allow importing data of various file formats such as csv, excel etc. Pandas allows various data manipulation operations such as group by, join, merge, melt, concatenation as well as data cleaning features such as filling, replacing or imputing null values.

Major Features of this library are as follows:

- DataFrame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and subsetting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
- Provides data filtration.

5.1.1.2 NumPy

NumPy is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. In comparison, MATLAB boasts a large number of additional toolboxes, notably Simulink, whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language. Moreover, complementary Python packages are available; SciPy is a library that adds more MATLAB- like functionality and Matplotlib is a plotting package that provides MATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient linear algebra computations.

5.1.1.2 Sci-kit Learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

5.1.1.3 SciPy

SciPy is a scientific python open source, distributed under the BSD licensed library to perform Mathematical, Scientific and Engineering Computations.

The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The SciPy library is built to work with NumPy arrays and provides many user-friendly and efficient numerical practices such as routines for numerical integration and optimization. Together, they run on all popular operating systems, are quick to install and are free of charge.

5.2 SVD Algorithm

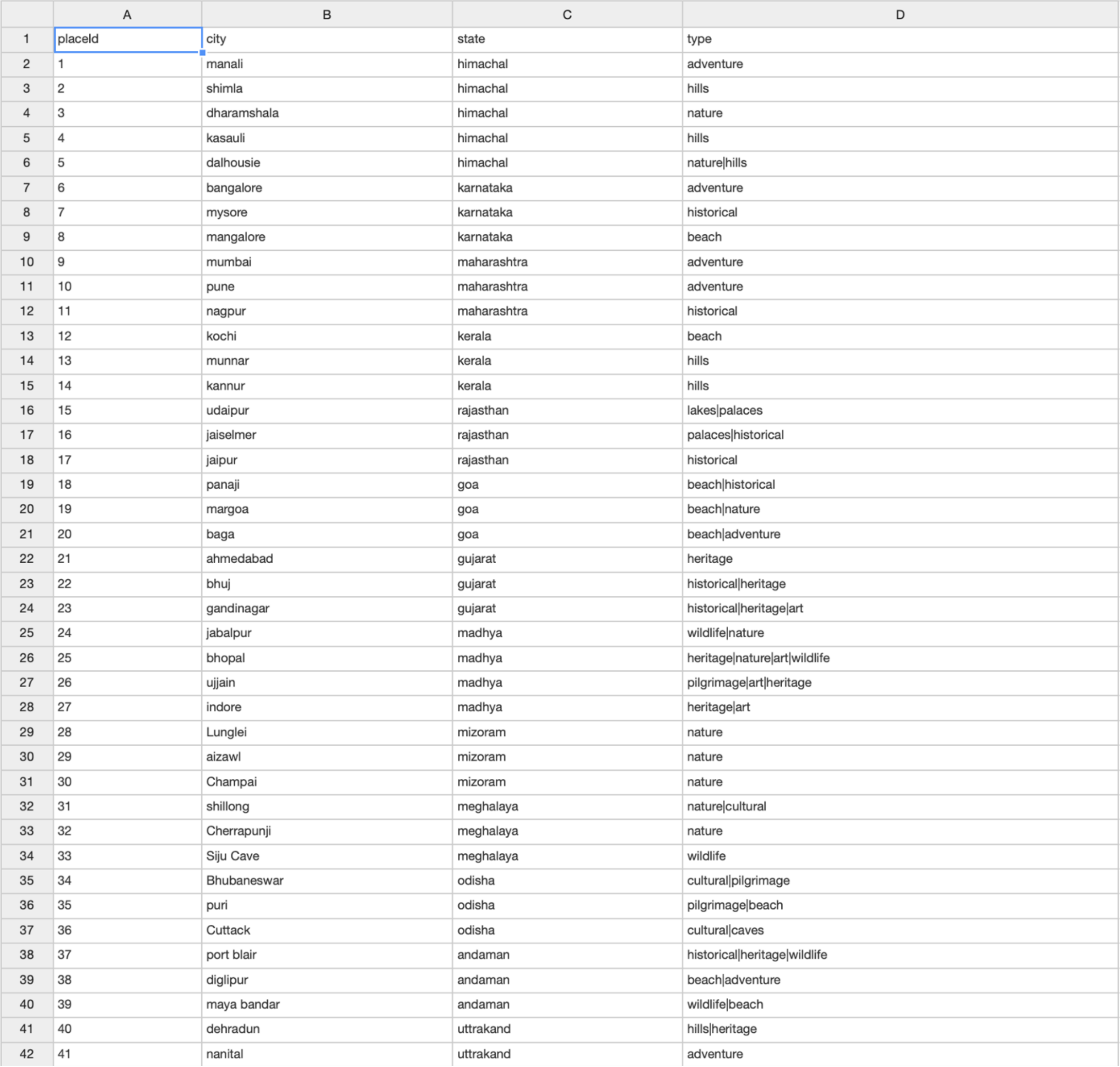
The Singular Value Decomposition (SVD) is a well-known matrix factorization technique that factors an m by n matrix X into three matrices as follows:

$$\overset{X}{\underset{m \times n}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \overset{U}{\underset{m \times r}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \overset{S}{\underset{r \times r}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \overset{V^{\top}}{\underset{r \times n}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$

Fig: 5.2. Matrix Factorization

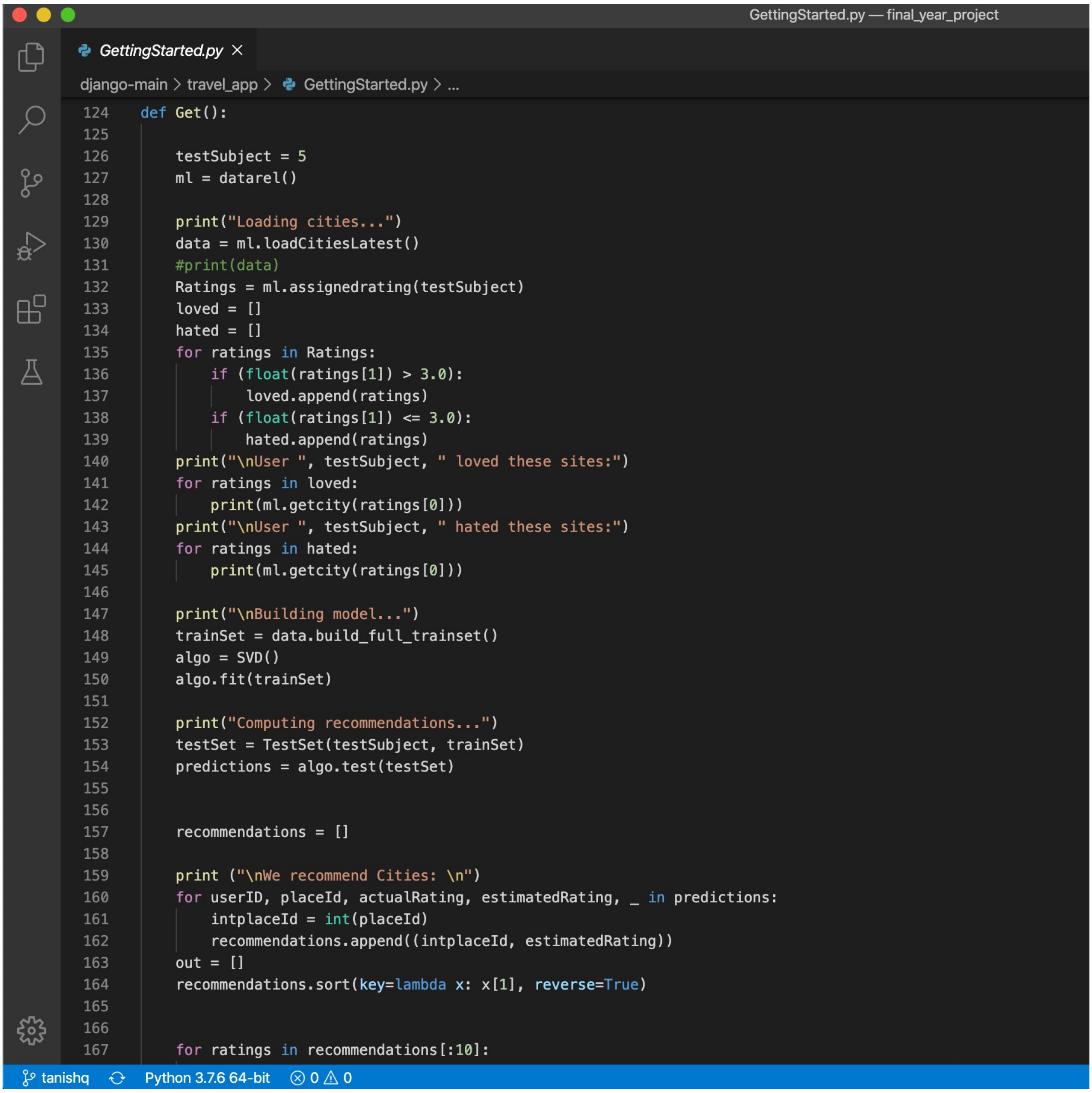
The matrix **S** is a diagonal matrix containing the *singular values* of the matrix **X**. There are exactly *r* singular values, where *r* is the *rank* of **X**. The rank of a matrix is the number of linearly independent rows or columns in the matrix. Recall that two vectors are linearly independent if they cannot be written as the sum or scalar multiple of any other vectors in the space. If every user who liked Manali also liked Kasauli, the two city vectors would be linearly dependent and would only contribute one to the rank.

Given that the SVD somehow reduces the dimensionality of our dataset and captures the "features" that we can use to compare users, how do we actually predict ratings? The first step is to represent the data set as a matrix where the users are rows, cities are columns, and the individual entries are specific ratings. In order to provide a baseline, we fill in all of the empty cells with the average rating for that movie and then compute the svd.



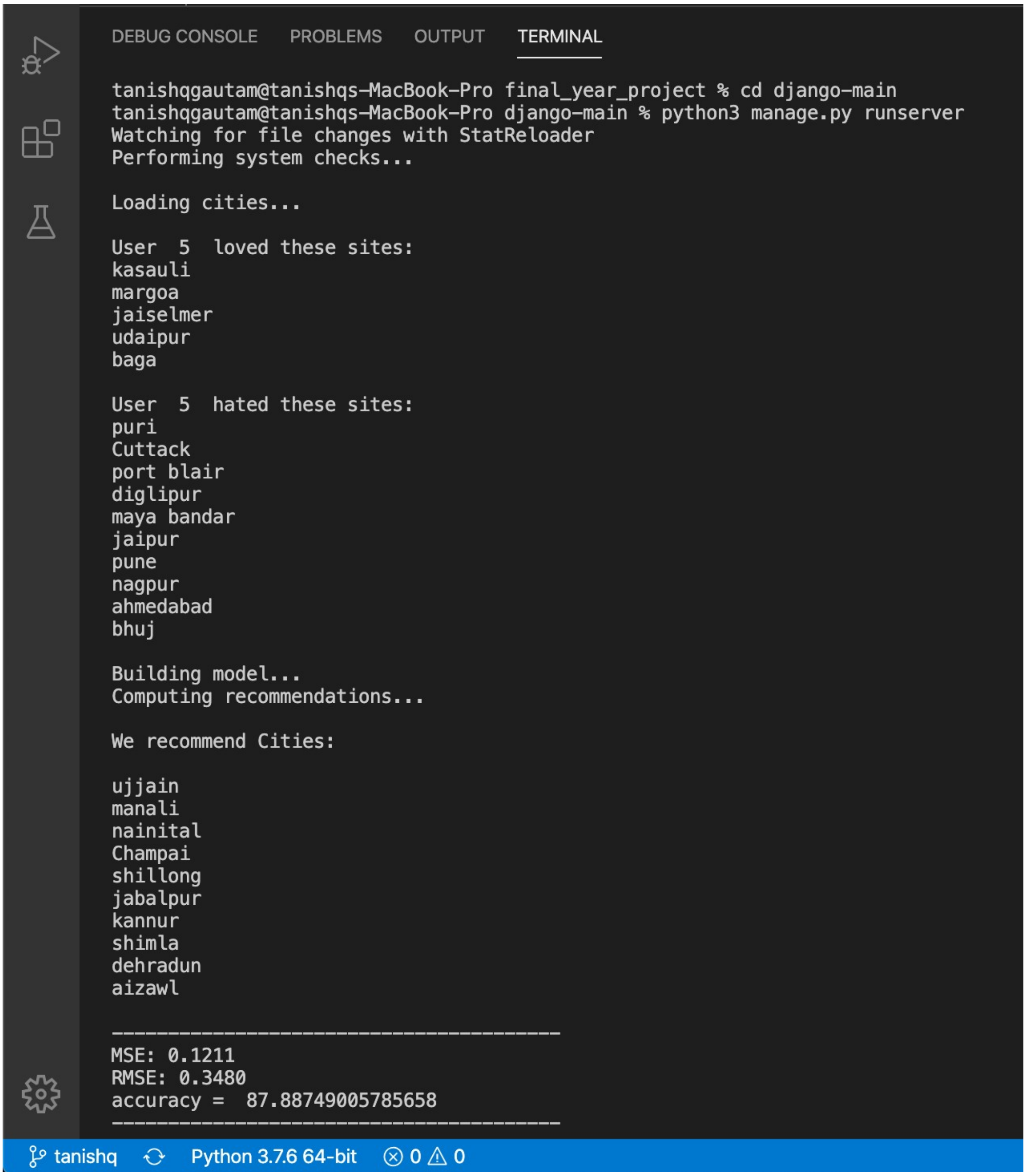
	A	B	C	D
1	placeId	city	state	type
2	1	manali	himachal	adventure
3	2	shimla	himachal	hills
4	3	dharamshala	himachal	nature
5	4	kasauli	himachal	hills
6	5	dalhousie	himachal	nature|hills
7	6	bangalore	karnataka	adventure
8	7	mysore	karnataka	historical
9	8	mangalore	karnataka	beach
10	9	mumbai	maharashtra	adventure
11	10	pune	maharashtra	adventure
12	11	nagpur	maharashtra	historical
13	12	kochi	kerala	beach
14	13	munnar	kerala	hills
15	14	kannur	kerala	hills
16	15	udaipur	rajasthan	lakes|palaces
17	16	jaiselmer	rajasthan	palaces|historical
18	17	jaipur	rajasthan	historical
19	18	panaji	goa	beach|historical
20	19	margoa	goa	beach|nature
21	20	baga	goa	beach|adventure
22	21	ahmedabad	gujarat	heritage
23	22	bhuj	gujarat	historical|heritage
24	23	gandinagar	gujarat	historical|heritage|art
25	24	jabalpur	madhya	wildlife|nature
26	25	bhopal	madhya	heritage|nature|art|wildlife
27	26	ujjain	madhya	pilgrimage|art|heritage
28	27	indore	madhya	heritage|art
29	28	Lunglei	mizoram	nature
30	29	aizawl	mizoram	nature
31	30	Champai	mizoram	nature
32	31	shillong	meghalaya	nature|cultural
33	32	Cherrapunji	meghalaya	nature
34	33	Siju Cave	meghalaya	wildlife
35	34	Bhubaneswar	odisha	cultural|pilgrimage
36	35	puri	odisha	pilgrimage|beach
37	36	Cuttack	odisha	cultural|caves
38	37	port blair	andaman	historical|heritage|wildlife
39	38	diglipur	andaman	beach|adventure
40	39	maya bandar	andaman	wildlife|beach
41	40	dehradun	uttrakand	hills|heritage
42	41	nanital	uttrakand	adventure

Fig 5.2.1 Dataset


GettingStarted.py — final_year_project
GettingStarted.py
django-main > travel_app > GettingStarted.py > ...
tanishq
Python 3.7.6 64-bit
ⓧ 0 ⚠ 0

```python
def Get():

    testSubject = 5
    ml = datarel()

    print("Loading cities...")
    data = ml.loadCitiesLatest()
    #print(data)
    Ratings = ml.assignedrating(testSubject)
    loved = []
    hated = []
    for ratings in Ratings:
        if (float(ratings[1]) > 3.0):
            loved.append(ratings)
        if (float(ratings[1]) <= 3.0):
            hated.append(ratings)
    print("\nUser ", testSubject, " loved these sites:")
    for ratings in loved:
        print(ml.getcity(ratings[0]))
    print("\nUser ", testSubject, " hated these sites:")
    for ratings in hated:
        print(ml.getcity(ratings[0]))

    print("\nBuilding model...")
    trainSet = data.build_full_trainset()
    algo = SVD()
    algo.fit(trainSet)

    print("Computing recommendations...")
    testSet = TestSet(testSubject, trainSet)
    predictions = algo.test(testSet)


    recommendations = []

    print ("\nWe recommend Cities: \n")
    for userID, placeId, actualRating, estimatedRating, _ in predictions:
        intplaceId = int(placeId)
        recommendations.append((intplaceId, estimatedRating))
    out = []
    recommendations.sort(key=lambda x: x[1], reverse=True)


    for ratings in recommendations[:10]:
```

Fig 5.2.2 SVD algorithm



```
tanishqgautam@tanishqs-MacBook-Pro final_year_project % cd django-main
tanishqgautam@tanishqs-MacBook-Pro django-main % python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

Loading cities...

User  5  loved these sites:
kasauli
margoa
jaiselmer
udaipur
baga

User  5  hated these sites:
puri
Cuttack
port blair
diglipur
maya bandar
jaipur
pune
nagpur
ahmedabad
bhuj

Building model...
Computing recommendations...

We recommend Cities:

ujjain
manali
nainital
Champai
shillong
jabalpur
kannur
shimla
dehradun
aizawl

----------------------------------------
MSE: 0.1211
RMSE: 0.3480
accuracy =  87.88749005785658
----------------------------------------
```

Fig 5.2.3 Recommendation

Important quantities to be calculated:

- **Mean Absolute Error** - Mean Absolute Error (MAE) is the average vertical distance between each point and the identity line. MAE is also the average horizontal distance between each point and the identity line.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

- **Root Mean Square Error** – It is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed.

RMSE
(Root Mean Square Error)

$$= \sqrt{\frac{\sum_{t=1}^{T} (\hat{y}_t - y_t)^2}{T}}.$$

- **Euclidean Distance** - the Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space.

Euclidean Distance

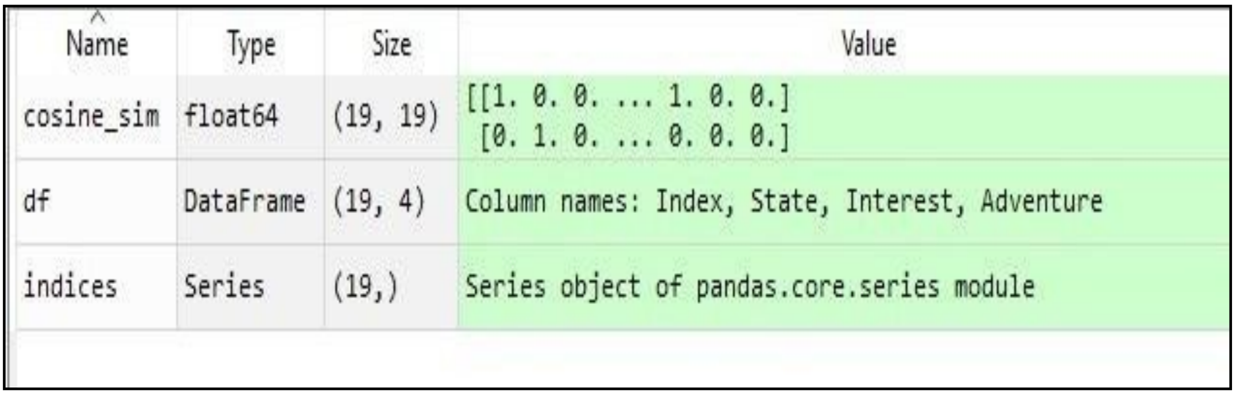
$$= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

5.3 Cosine Similarity

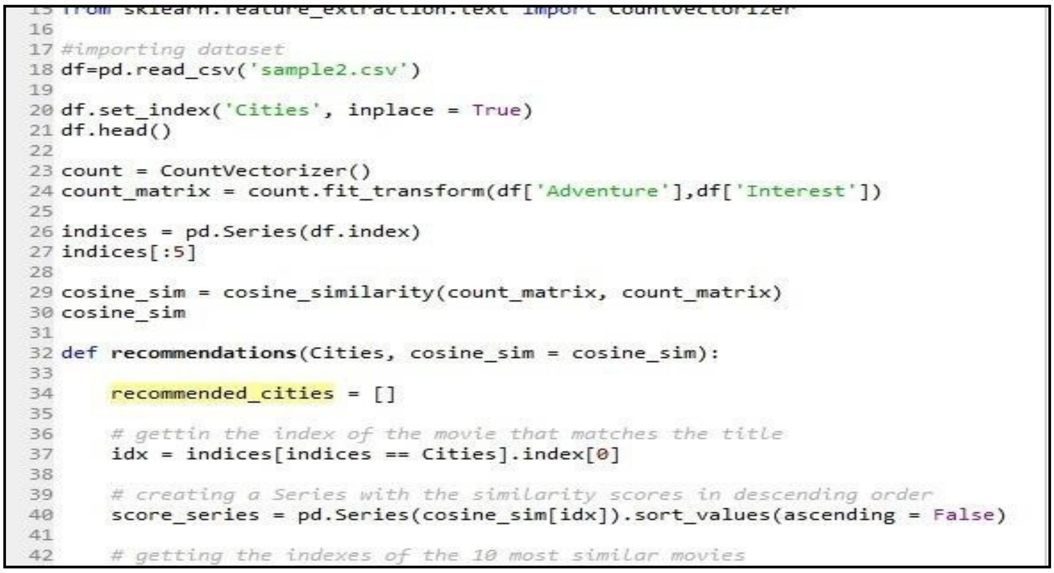
Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any angle in the interval (0, π] radians. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors oriented at 90° relative to each other have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude.

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$



Name	Type	Size	Value
cosine_sim	float64	(19, 19)	[[1. 0. 0. ... 1. 0. 0.] [0. 1. 0. ... 0. 0. 0.]
df	DataFrame	(19, 4)	Column names: Index, State, Interest, Adventure
indices	Series	(19,)	Series object of pandas.core.series module

Fig: 5.3.1 Variable Explorer



```
from sklearn.feature_extraction.text import CountVectorizer

#importing dataset
df=pd.read_csv('sample2.csv')

df.set_index('Cities', inplace = True)
df.head()

count = CountVectorizer()
count_matrix = count.fit_transform(df['Adventure'],df['Interest'])

indices = pd.Series(df.index)
indices[:5]

cosine_sim = cosine_similarity(count_matrix, count_matrix)
cosine_sim

def recommendations(Cities, cosine_sim = cosine_sim):

    recommended_cities = []

    # gettin the index of the movie that matches the title
    idx = indices[indices == Cities].index[0]

    # creating a Series with the similarity scores in descending order
    score_series = pd.Series(cosine_sim[idx]).sort_values(ascending = False)

    # getting the indexes of the 10 most similar movies
```

Fig: 5.3.2 Cosine Similarity Algorithm

5.4 Comparative Study

In order to obtain the desired results, we studied number of algorithms that work for recommender systems. This whole process helped us to figure out the algorithm with the most efficient results, which is the SVD algorithm that works on rating. Comparative study of some algorithms is listed below:

Cosine Similarity

- Cosine Similarity works on a vector space model thus counting frequency.
- Due to frequency count it fixes its prediction. Rather than predicting new predictions, it generates same prediction.

Random Forest

- Random forest is a tree algorithm. It is overfitting the data in constructing tree thus predicting same city more than once.

Decision Tree

- Decision tree is another tree algorithm, that have an issue of overfitting thus predicting same city more than once.

Assyst	RMSE	MAE	Time
SVD	0.2718	0.2301	0:00:11
SVD++	0.4249	0.3479	0:09:03
NMF	1.1105	0.7600	0:00:15
K-NN	1.3215	1.1611	0:00:10
Co-Clustering	1.0218	0.8612	0:00:03
Random	1.2712	1.1157	0:00:01

Table: 5.4.1 Comparative Study of Algorithms

The above table shows the values of RMSE and MAE for various algorithms. The listed algorithm all work fine for the job but the best algorithm will be one with following properties:

- Less time of execution
- Algorithm whose RMSE and MAE values are as comparable as possible.
  Based on the results we got, the best algorithm was found to be SVD algorithm.

5.5 Data Design

In this project most of the data sets used, is in the form of CSV files while training and then ultimately the models of the databases all are built by us and thus it does not involve much data design.

5.5.1 Data objects and Resultant Data Structures

A data object is a region of storage that contains a value or group of values. Each value can be accessed using its identifier or a more complex expression that refers to the object. In addition, each object has a unique data type. The data type of an object determines the storage allocation for that object and the interpretation of the values during subsequent access. The data objects used in the system are: Strings, Integers and Floating-point numbers. The resultant Data structures are in the form of tables that are the models build in Django. All the operations and functions are applied to these models to work with data and obtain results.

5.6 Interface Design

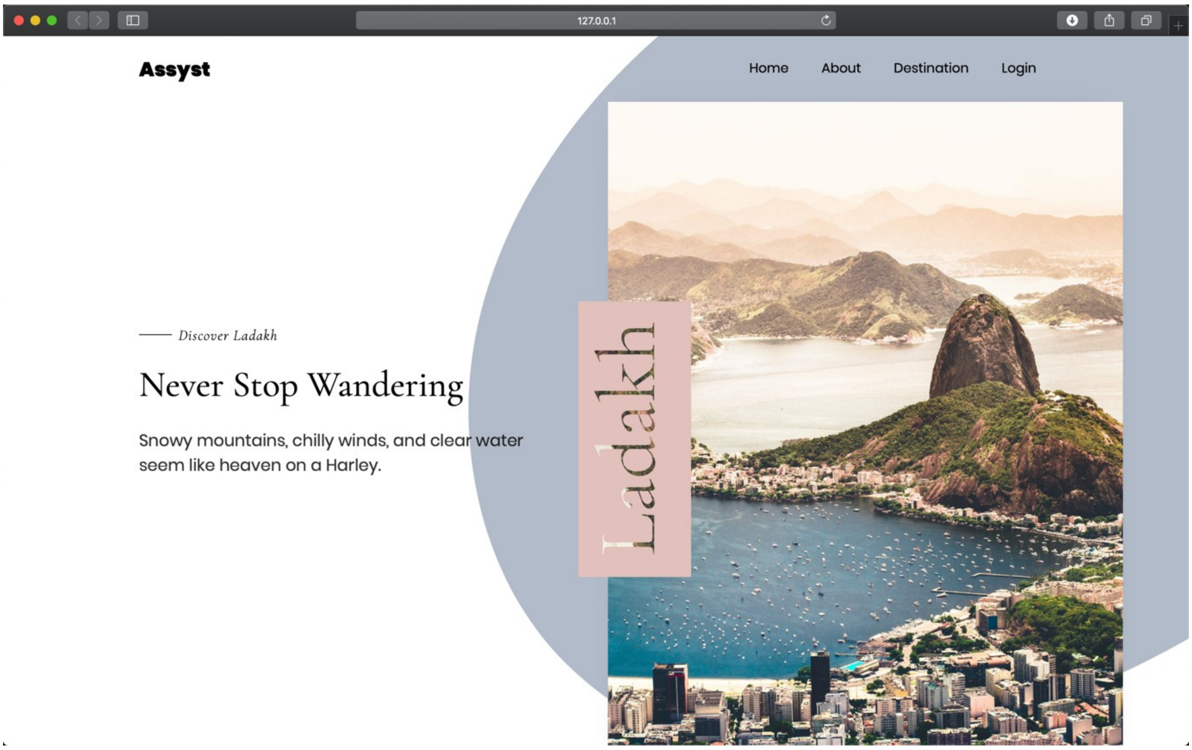
The interface of our project is quite realistic and lively as we have developed an interactive, quick responsive and functional front-end for our website that gives us a great platform to interact with our users. We have designed login and registration pages, as well as user dashboard and connected them with an SQLite database using Django.

We have developed a questionnaire to collect the ratings given by users according to their preferences and the same is being stored in a model of the same name that can be later used by the algorithm to produce the desired results.

Currently, our website serves only for Indian cities but considering the wider scope and utility of our project, this can be expanded on an International scale as well.

5.6.1 Human Machine Interface Design Specification

The human machine interface design of our project gives look and feel of a website that is easy to use and handle by people of any age group with basic computer knowledge. We have designed a template that will accentuate the euphoria people feel while planning a trip. Most people have to wait for a long time to get out of their boring routine and to be able to actually spend some good time during travel. We completely understand this feeling and our services try to provide them that in the best possible way.

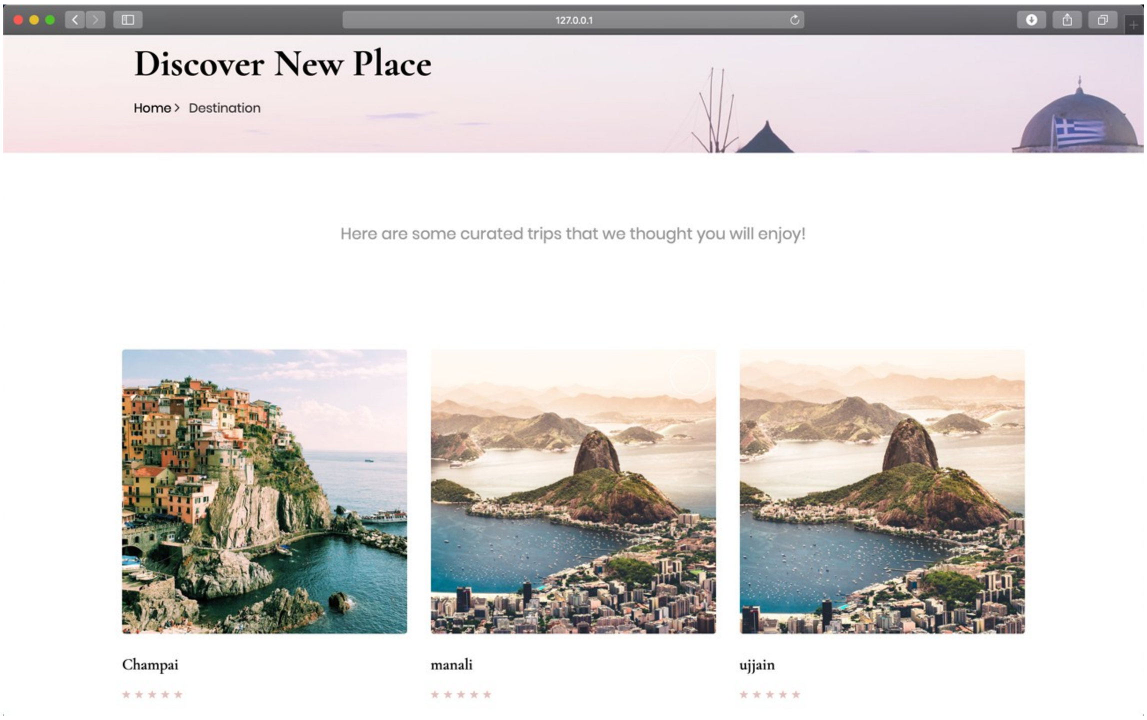


Discover Ladakh

Never Stop Wandering

Snowy mountains, chilly winds, and clear water seem like heaven on a Harley.

Fig: 5.6.1.1 Index Page of Assyst



127.0.0.1

Discover New Place

Home > Destination

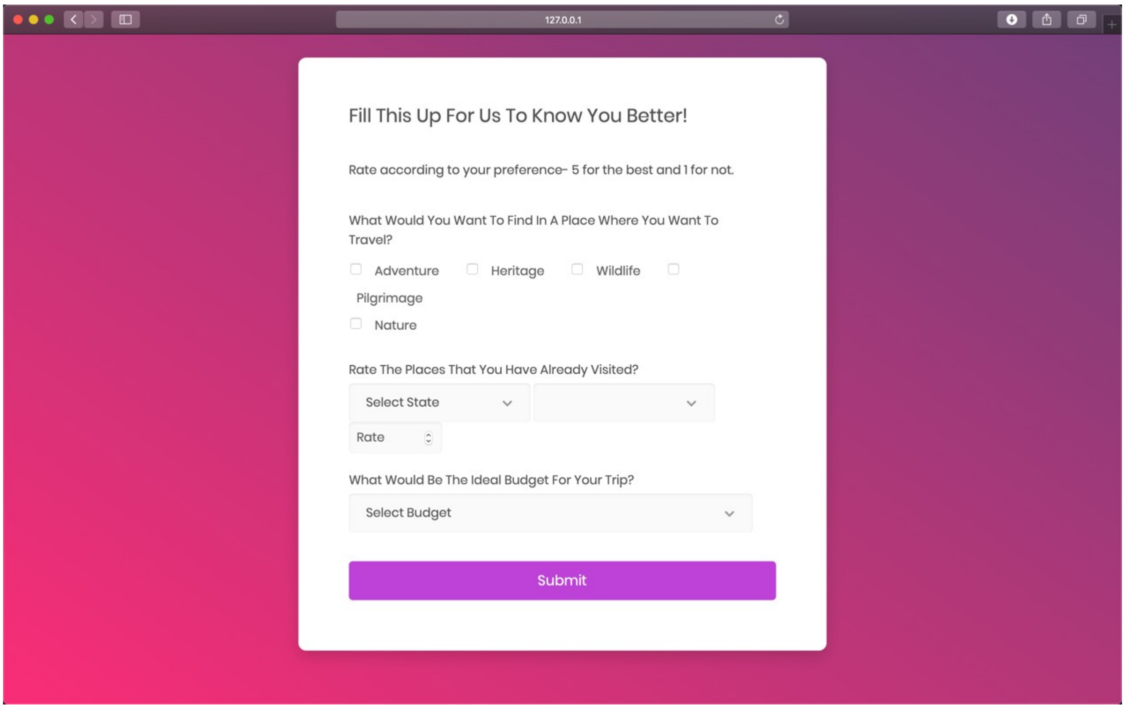
Here are some curated trips that we thought you will enjoy!

Champai

manali

ujjain

Fig: 5.6.1.2 User Recommendation



127.0.0.1

Fill This Up For Us To Know You Better!

Rate according to your preference- 5 for the best and 1 for not.

What Would You Want To Find In A Place Where You Want To Travel?

□ Adventure □ Heritage □ Wildlife □ Pilgrimage
□ Nature

Rate The Places That You Have Already Visited?

Select State

Rate

What Would Be The Ideal Budget For Your Trip?

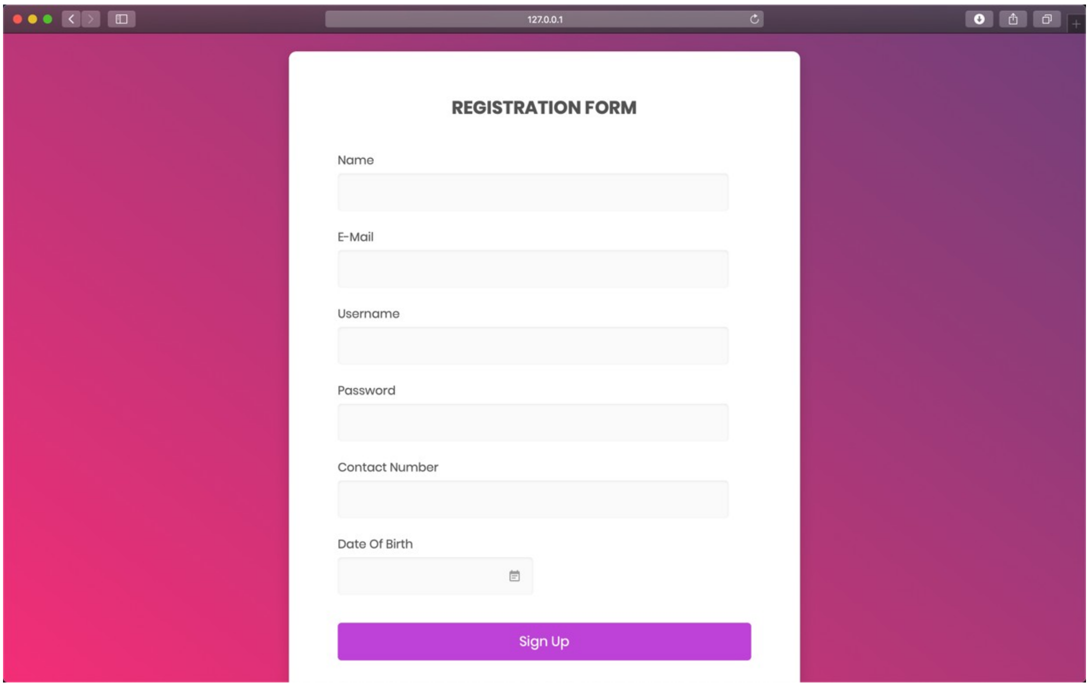
Select Budget

Submit

Fig: 5.6.1.3 Questionnaire

5.6.2 I/O Forms

The input of the user is taken using the website by using the frontend technologies like HTML, CSS and JavaScript. The two main input forms are the Registration Form and the Questionnaire Form that help us to know the user better. Once these values are taken, Django comes into play and they are stored in the models of SQLite. Once the algorithm does its job, the output is taken from the model and printed on the user dashboard for the user to see.



REGISTRATION FORM

Name

E-Mail

Username

Password

Contact Number

Date Of Birth

Fig 5.6.2.1 Registration Page

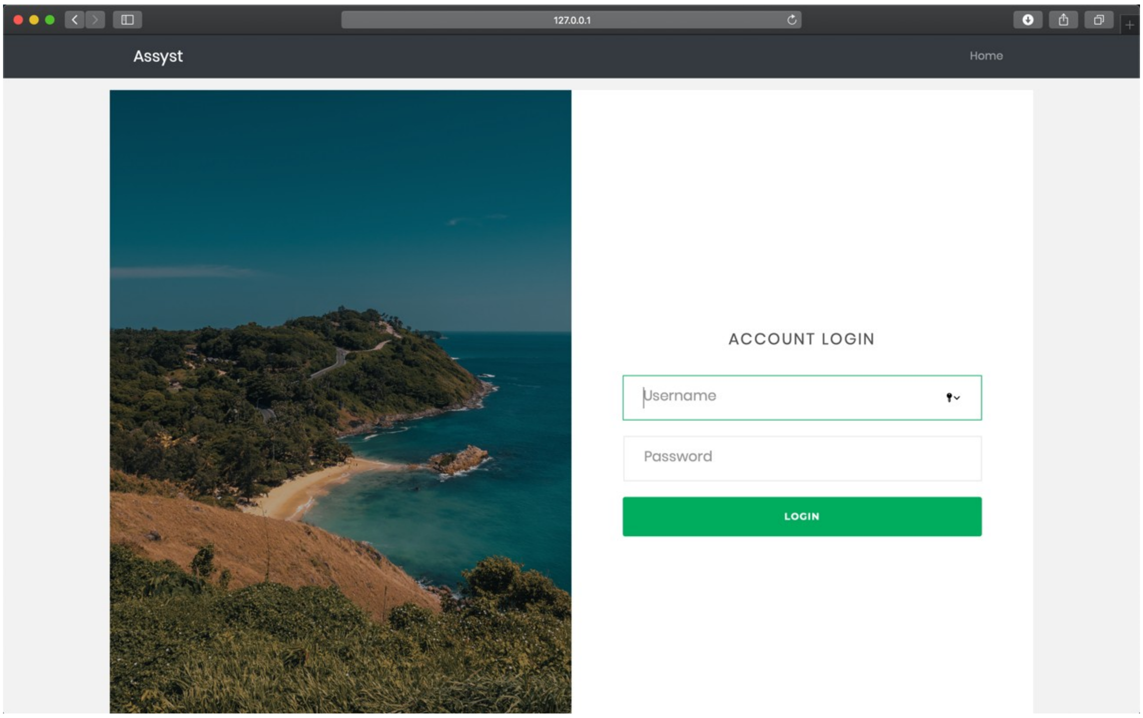
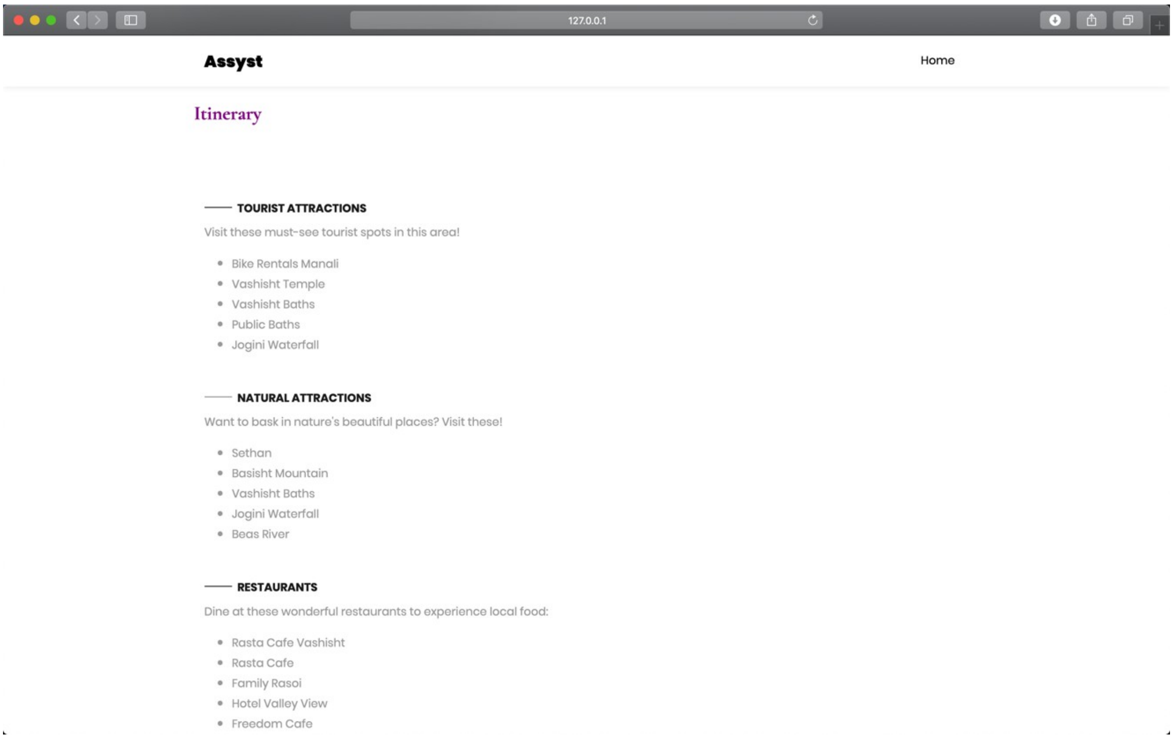


Fig: 5.6.2.2. Login



Itinerary

TOURIST ATTRACTIONS

Visit these must-see tourist spots in this area!

- Bike Rentals Manali
- Vashisht Temple
- Vashisht Baths
- Public Baths
- Jogini Waterfall

NATURAL ATTRACTIONS

Want to bask in nature's beautiful places? Visit these!

- Sethan
- Basisht Mountain
- Vashisht Baths
- Jogini Waterfall
- Beas River

RESTAURANTS

Dine at these wonderful restaurants to experience local food:

- Rasta Cafe Vashisht
- Rasta Cafe
- Family Rasoi
- Hotel Valley View
- Freedom Cafe

Fig 5.2.2.3 Itinerary

Chapter 6

Testing

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

6.1 Testing Objectives

Software Testing has different goals and objectives. The major objectives of Software testing are as follows:

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.

Testing can either be done manually or using an automated testing tool:

- **Manual** – This testing is performed without taking help of automated testing tools. The software tester prepares test cases for different sections and levels of the code, executes the tests and reports the result to the manager. Manual testing is time and resource consuming. The tester needs to confirm whether or not right test cases are used. Major portion of testing involves manual testing.

- **Automated -** This testing is a testing procedure done with aid of automated testing tools. The limitations with manual testing can be overcome using automated test tools.

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

6.2 Testing Scope

Scope of testing includes the process to determine all those features or functionality as one may say that will be considered for Testing during a particular level of testing in a particular release. The scope is determined during the Test planning phase where in the test plan we mention the scope of testing we will consider.

The project has two parts one is the user interface, i.e. the front-end and the back-end functionality and the other one is the algorithm testing and coding. The front-end didn't needed testing while the back-end needed to manage the models that makes the base where the algorithm functions that required some amount of testing.

The main part of the project that required testing was the algorithm working and data sets. As we didn't get the ideal dataset online, the process to design one that works with all the algorithm we tried was little tricky. Also, in order to perform the training of the machine we started with a comparatively smaller data set, gradually increasing the size in later stages.

The process of settling to the right algorithm was again something that required decent amount of testing. We went through a couple of algorithms and tested them on our data set until we achieved the desired result. Also, we asked our friends and other people to test the system and its working to achieve smooth functioning.

6.3 Testing Principles

6.3.1 Exhaustive testing is not possible. Instead, we need the optimal amount of testing based on the risk assessment of the application.

6.3.2 Defect Clustering which states that a small number of modules contain most of the defects detected, i.e., approximately 80% of the problems are found in 20% of the modules.

6.3.3 If the same sets of repetitive tests are conducted, the method will be useless for discovering new defects.

6.3.4 Testing talks about the presence of defects and don't talk about the absence of defects.

6.3.5 i.e. Software Testing reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.

6.3.6 It is possible that software which is 99% bug-free is still unusable.

6.3.7 Testing should start as early as possible in the Software Development Life Cycle.

6.3.8 Testing is context dependent which basically means that the way you test an e- commerce site will be different from the way you test a commercial off the shelf application.

Black-box testing

It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok' and problematic otherwise.

In this testing method, the design and structure of the code are not known to the tester, and testing engineers and end users conduct this test on the software.

Black-box testing techniques:

- **Equivalence class** – The input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.
- **Boundary values** – The input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.
- **Cause-effect graphing** – In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.
- **Pair-wise Testing** – The behavior of software depends on multiple parameters. In pairwise testing, the multiple parameters are tested pair-wise for their different values.
- **State-based testing** – The system changes state on provision of input. These systems are tested based on their states and input.

White-box testing

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as 'Structural' testing.

In this testing method, the design and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

Below are some White-box testing techniques:

- **Control-flow testing** – The purpose of the control-flow testing to set up test cases which cover all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.
- **Data-flow testing** – This testing technique emphasis to cover all the data

variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified.

Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels –

Unit Testing

While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

Integration Testing

Even if the units of software are working fine individually, there is a need to find out if the units, if integrated together would also work without errors. For example, argument passing and data updating etc.

6.4 Testing Methods used

We applied testing methods at various levels that are generally carried out during the development of a software project. We applied white box testing and unit testing.

The testing started in right from very initial stages, when we started dealing with various algorithms. The major part of this project deals with the correct working of the algorithms. Right from the beginning we were quite clear as to what we want as the outcome of this project. Thus, to achieve that as accurately as possible proper testing methods were required.

We wanted our project to be able to recommend city on the basis of a questionnaire and past experiences of the user. This process took a while, as we had to select the best algorithm for the job. We started with cosine similarity. This algorithm worked fine with some issues. It recommended the same cities again and again, corresponding to the same input. Whereas, we wanted the user to be able to get different recommendations every time.

Next, we tried decision tree and random forest algorithm. The biggest problem we faced while working with these algorithms was that, it required the data set to be in 1NF form. Once this was achieved, the results obtained weren't satisfactory. There were some cities that were repeating, i.e. it was being recommended more than once. At last, we went on with SVD algorithm. It required us to provide ratings for the cities but then gave good results. These recommendations were tested for many test cases.

Each one of us had a part to play in this project and the project itself contained many modules to work with. Hence, unit testing was performed so as to check for the smooth working of each individual module.

6.5 Test Cases

To test the algorithm, we defined various test cases. We gradually increased the data set values and changed different algorithms with variable inputs. We reiterated this whole process a number of times till satisfactory results were obtained.

Case 1: Using cosine similarity with normal data set.

Case 2: Using decision tree and random forest with a dataset which is in 1NF.

Case 3: Using SVD with data set having ratings.

Using the combination of cases in Case 1, Case 2 and Case 3 we tested the software for different algorithms with different data set combination.
It is important here to note that given any set of conditions the algorithm reached its maximum efficiency in its third run.

Chapter 7

Results and Discussion

7.1 Sample Test Data and Results

Case 1: Using cosine similarity

a. **Working with normal data set for the city 'Bangalore'**

The cities predicted were same, irrespective of the number of times we tried it for 'Bangalore. To check it again we tried cosine similarity for the city 'Mumbai'.

b. **Working with normal data set for the city 'Mumbai'**

The results didn't improve. Predicted cities came out to be same, whenever Mumbai was the city in the input.

Case 2: Using decision tree for the data which is in 1NF

As mentioned before, decision tree works well only with the data which is in 1NF. The above figure is an output of decision tree algorithm, the data set used was in 1NF and hence the result obtained was quite similar to the one which was predicted.

If the data set would not have been in 1NF then the results would not be satisfactory.

Case 3: Using random forest for the data which is not in 1NF

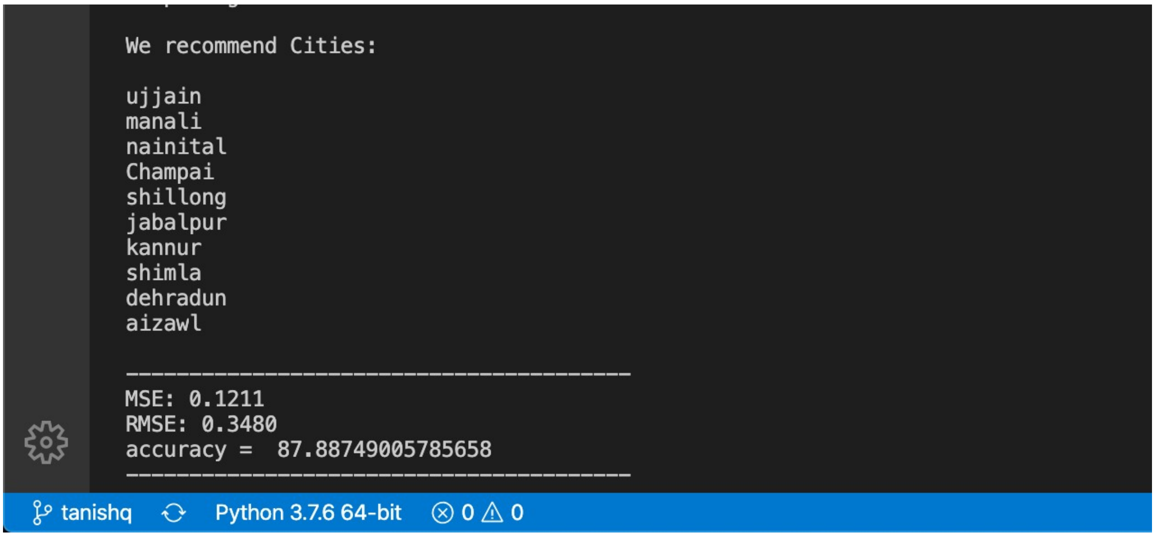
The random forest algorithm works just the same as decision tree and requires data to be in 1NF for its smooth working. It serves as an evidence to this where the algorithm fails when the data set is not in 1NF.

Case 4: Using SVM for the data which is in 1NF

This is again an example of the algorithm named SVM i.e. the Support Vector Machine. This algorithm has the similar way of working as the decision tree and random forest. If the data set given is in 1NF, which is the case here, the algorithm produced satisfactory results.

Case 6: Using SVD for a normal data set having ratings

The figure shows the output of SVD algorithm where we test the system by listing down the cities liked by the user and then in the output, we got the recommended cities and the cities predicted every time were different in order and values. These results were satisfactory and we decided to go forward with this algorithm.



```
We recommend Cities:

ujjain
manali
nainital
Champai
shillong
jabalpur
kannur
shimla
dehradun
aizawl

----------------------------------------
MSE: 0.1211
RMSE: 0.3480
accuracy =  87.88749005785658
----------------------------------------
```

Fig 8.1 SVD Result

Chapter 8

Conclusion and Future Scope

8.1 Conclusion

Planning a trip is a complex decision process as it involves taking in account all the variables on users' interests, their experiences, behavior and the tourist item factors such as accessibility, their revenue point. Because of the possibilities of tourism development, recommender system has been an area of so much interest. As a result, number of organizations being involved in tourism industry, needs to have this system so that they can understand what would be most appropriate for the user and which place can make their travel most memorable. This project has been very helpful in learning skills in the field of Recommendation System. The objective of the project is to find out the unexplored places in India, so as to increase the tourism in the country and also recommend them as per their interest. The system has been optimized using a data set acquired using the various available websites of tourism in India. For every recommender system, it is very important to hold specific information about users and their interests as a profile. The development of new learning mechanisms to analyze interactions of a user with the system and its ability to convert it into user preference can make recommender system more dynamic in providing suggestions. This project is an attempt to achieve similar goals.

8.2 Contribution

Each one of us has worked on different aspects of the project to integrate everything on a single platform that we have thought of. The whole process went in a reverse chronological order where, first, we identified our needs from the project and then we went on creating the website.

The integral part of the project was to design an algorithm that works fairly on our data set and produce the required results. Cosine similarity gave us result but due to frequency count it fixes its prediction and rather than predicting new results it generated the same prediction. Next, we tried Random Forest and Decision Tree, but both the algorithms faced the problem of overfitting which led to the prediction of the same city more than once.

After all the not so successful approaches we decided to implement the SVD algorithm. SVD algorithm worked on the ratings obtained by the user and helped us to obtain the results we desired from the beginning. Every time the predicted cities were different and we achieved pretty good results.

Then to develop the GUI of the software we used a template and modified it according to our needs. Once we got the whole template prepared, we worked on the Django framework to do all the back-end functionalities using the models, forms and views of Django. This helped our website to be recognized as a whole working project.

8.3 Future Scope

- There are a number of travel websites on the Internet that help you to plan your trips but they only suggest packages on the basis of budget.
- Our website is designed to understand your heart's desire and find you places to travel that will suit your every need, and also keeping in account your budgets like the other websites.
- Every person has a different choice and also their preferences can vary for every individual holiday they plan. We take into account every factor and here is a list of destinations that you will find perfect.
- This website is a great idea in our busy schedules as now we don't have to go through a long list of places in this vast country and then sit around comprehending what is the best and what is not.
- Currently, our website just aims at discovering and planning your destination only for India. This can be expanded to an international scale to suit every need.
- The explosive growth and variety of information available on the Web and the rapid introduction of new e-business services frequently overwhelms users, leading them to make poor decisions. And In the face of the information overload and the fastidious task of preparing a trip, recommender systems bring more easiness in accessing information about travel destinations and tourist attraction.

References

[1] Kuan-Hua Lai ; Neil Y. Yen ; Mu-Yen Chen , Design of an AI-Empowered Recommender System for Travelling Support: Individual Traveler as an Instance , IEEE , 29th October 2018

[2] Kumar Abhishek , M.P.Singh , Saswata Ghosh Abhishek Anand , Weather Forecasting Model using Artificial Neural Network , C3IT , 2011

[3] Qian Li , Yi Kong , Shi-ming Xia , A method of weather recognition based on outdoor images , IEEE , 12 October 2015

[4] Na Wang , Pengyuan Wang , Baowei Zhang , An improved TF-IDF weights function based on information theory , IEEE , 12 August 2010

[5] Kwan Hui Lim , Personalized recommendation of Travel Itineraries based on Tourist interest and preference , 2015

[6] Aldy Gunawan · Hoong Chuin Lau · Kun Lu , A Fast Algorithm for Personalized Travel Planning Recommendation