A Project Report on

# TRAVEL RECOMMEDATION SYSTEM

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

**Bachelor of Engineering**

in

**COMPUTER ENGINEERING**

by

**Nakulesh Jayakrishnan (17202012)**

**Gauri Deshpande (16102046)**

**Akanksha Koshti (16102011)**

Under the Guidance of

**Prof. Mayuri Jain**


**Department of Branch Name**
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615 UNIVERSITY OF MUMBAI

**Academic Year 2017-2018**

# Approval Sheet

This Project Report entitled *"Travel Recommendation System"* Submitted by "*Nakulesh Jayakrishnan"(17202012), "Gauri Deshpande" (16102046), "Akanksha Koshti" (16102011)* is approved for the partial fulfilment of the requirement for the award of the degree of *Bachelor of Engineering* in *Computer Engineering* from *University of Mumbai*.

Prof. Mayuri Jain
(Guide)

Prof. Sachin Malave
H.O.D., Computer Engineering.

Place: A.P.Shah Institute of Technology, Thane

Date: 12th April,2019

# CERTIFICATE

This is to certify that the project entitled *"Travel Recommendation System"* submitted by *"Nakulesh Jayakrishnan"(17202012),"Gauri Deshpande"(16102046),"Akanksha Koshti" (16102011)* for the partial fulfilment of the requirement for award of a degree *Bachelor of Engineering* in *Computer Engineering* to the University of Mumbai, is a bonafide work carried out during academic year 2017-2018.

Prof. Mayuri Jain
Guide

Prof. Sachin Malave
Head Department of Computer Engineering

Dr.Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place: A. P. Shah Institute of Technology, Thane

Date: 12th April,2019

# Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/ fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Nakulesh Jayakrishnan,17202012)

_____

(Gauri Deshpande,16102046)

_____

(Akanksha Koshti,16102011)

## ABSTRACT

There is a growing need for tourism and tourist activities around the globe. Everyone is looking for a perfect travel getaway that meets their needs. In this project, our main aim is to provide a suitable destination as per the user's needs. This includes taking the user's as well as trip's attributes as an input and building a recommendation around that. User will get a personalized travel destination according to their interests. This system is aimed at individuals who need a perfect destination for a trip. The main domain that our project focuses on is Machine Learning. Our dataset will be trained according to the various user inputs and will output a suitable travel destination.

The scope of our project for this semester is limited to ten target variables i.e. ten famous hotspot destinations across India. This application is developed by keeping in mind the citizens of Mumbai city only. By taking the user's interests as attributes for training the dataset, we are making sure that the result is the most accurate.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

TRS: TRAVEL RECOMMENDATION SYSTEM

GNB: GAUSSIAN NAÏVE BAYES

MAP: MAXIMUM A POSTERIORI

AI: ARTIFICIAL INTELLIGENCE

# Introduction

In today's day and age, the tourism industry is constantly on a rise. We have developed a travel recommendation system which works according to the user's interests to suggest a travel destination across India. In this project, we have limited our scope to the hotspot travel destinations across India which are most frequently visited by people and are well-known. This application is developed particularly for people who stay in Mumbai as the scope for this year's project is related to the citizens of Mumbai. The domain of our project is Machine Learning, wherein we train the user data available and later test it to get accurate results. The algorithm used to train the data set is 'Naïve Bayes' algorithm as it proved to give the highest accuracy results.

## 1.1. Naïve-Bayes algorithm for training

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task.

The crux of the classifier is based on the Bayes theorem. In machine learning we are often interested in selecting the best hypothesis (h) given data (d).In a classification problem, our hypothesis (h) may be the class to assign for a new data instance (d).

One of the easiest ways of selecting the most probable hypothesis given the data that we have that we can use as our prior knowledge about the problem. Bayes' Theorem provides a way that we can calculate the probability of a hypothesis given our prior knowledge.

Bayes' Theorem is stated as:

$$P(h|d) = (P(d|h) * P(h)) / P(d)$$

Where,

- **P(h|d)** is the probability of hypothesis h given the data d. This is called the posterior probability.
- **P(d|h)** is the probability of data d given that the hypothesis h was true.
- **P(h)** is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
- **P(d)** is the probability of the data (regardless of the hypothesis).

You can see that we are interested in calculating the posterior probability of P(h|d) from the prior probability p(h) with P(D) and P(d|h).After calculating the posterior probability for a number of different hypotheses, you can select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the maximum a posteriori (MAP) hypothesis.

This can be written as:

$$MAP(h) = max(P(h|d))$$

or

$$MAP(h) = max((P(d|h) * P(h)) / P(d))$$

or

$$MAP(h) = max(P(d|h) * P(h))$$

The P(d) is a normalizing term which allows us to calculate the probability. We can drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize.Back to classification, if we have an even number of instances in each class in our training data, then the probability of each class (e.g. P(h)) will be equal. Again, this would be a constant term in our equation and we could drop it so that we end up with:

$$MAP(h) = max(P(d|h))$$

### 1.1.1GaussianNB() function

A Gaussian Naive Bayes algorithm is a special type of NB algorithm. It's specifically used when the features have continuous values. It's also assumed that all the features are following a gaussian distribution i.e, normal distribution.

### 1.1.2 Training and testing using GaussianNB():

The following code shows how the GaussianNB() function is used to train and test the data in sklearn:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.externals import joblib
data=pd.read_csv('C:/Users/nakul/Downloads/trs-dataset01.csv')
#print (data.head(10))
y=data.rp
x=data.drop('rp',axis=1)
#print(x)
x_train, x_test, y_train, y_test = train_test_split(x, y,test_size=0.2,shuffle=False)
#print(x_train)
#print(x_train.shape)  #prints no. of rows,columns of training set
#print("\n Test data")
#print(x_test)
#print(x_test.shape)    #prints no. of rows,columns of testing set
#clf = DecisionTreeClassifier()
#clf.fit(x_train, y_train)
#y_pred=clf.predict(x_test)
#print("Accuracy Score DT:\n")
#print(metrics.accuracy_score(y_test,y_pred))

#Gaussian classifier
gnb=GaussianNB()
gnb.fit(x_train, y_train)  #training the model
y_pred=gnb.predict(x_test)            #predicting the class for test data and expected is y_test
#joblib.dump(gnb, 'trs.pkl')
#print("Classification Report:\n")
#print(metrics.classification_report(y_test,y_pred))
#print("Confusion matrix:\n")
#print(metrics.confusion_matrix(y_test,y_pred))
print("Accuracy Score gnb:\n")
print(metrics.accuracy_score(y_test,y_pred))
```

*Figure 1.1: Training and testing using GaussianNB`*

### 1.1.3 Comparing Decision Tree algorithm and Naïve Bayes algorithm:

When decision tree algorithm and naïve bayes algorithm were compared to see which algorithm gave the most accuracy, it was the naïve bayes algorithm that gave the most accuracy. Same is observed below:


*Figure 1.2: Comparing Decision tree and Naive Bayes Algorithm*

# Literature Review

According to a recent study done by K. Kesorn, W. Juraphanthong and A. Salaiwarakul, a personalized attraction recommendation system can be developed for tourists using their social media check-in data. Online social networks now play a prominent role in our daily lives and our decisions and behaviors in many areas. Of particular interest here is the application of social network data to give users access to tourist information. There is a growing need for information on tourism and tourist activities to satisfy user queries in this domain. Social networks, such as Facebook, Twitter, and Foursquare, among others, store substantial volumes of check-in data, which are a valuable resource for recommending tourism attractions. However, using Facebook check-in data has rarely been considered in conventional recommendation systems (RSs). This presents not only a new research challenge for the computer science and information technology fields but also an interesting opportunity for the tourism industry: knowing what kind of attractions tourists are interested in and how to acquire their user preferences without adding tasks to users of an RS. We propose a tourism RS that is based on its recommendations on data dynamically aggregated and extrapolated from the Facebook check-in data. In addition, the so-called “cold-start”problem has been resolved by using users' Friends' check-in data to analyze ongoing Facebook activity and update user profiles in the system.

However, we came to a conclusion that social media data can be highly fictional as not everyone enters the accurate information and hence it can be highly misleading. Therefore, to restrict our scope to the user, we decided to take the user input instead so as to customize the project according to the user’s requirement. Here, the user has to answer a few basic questions and according to his/her needs, a travel destination would then be suggested to them.

# System Flow

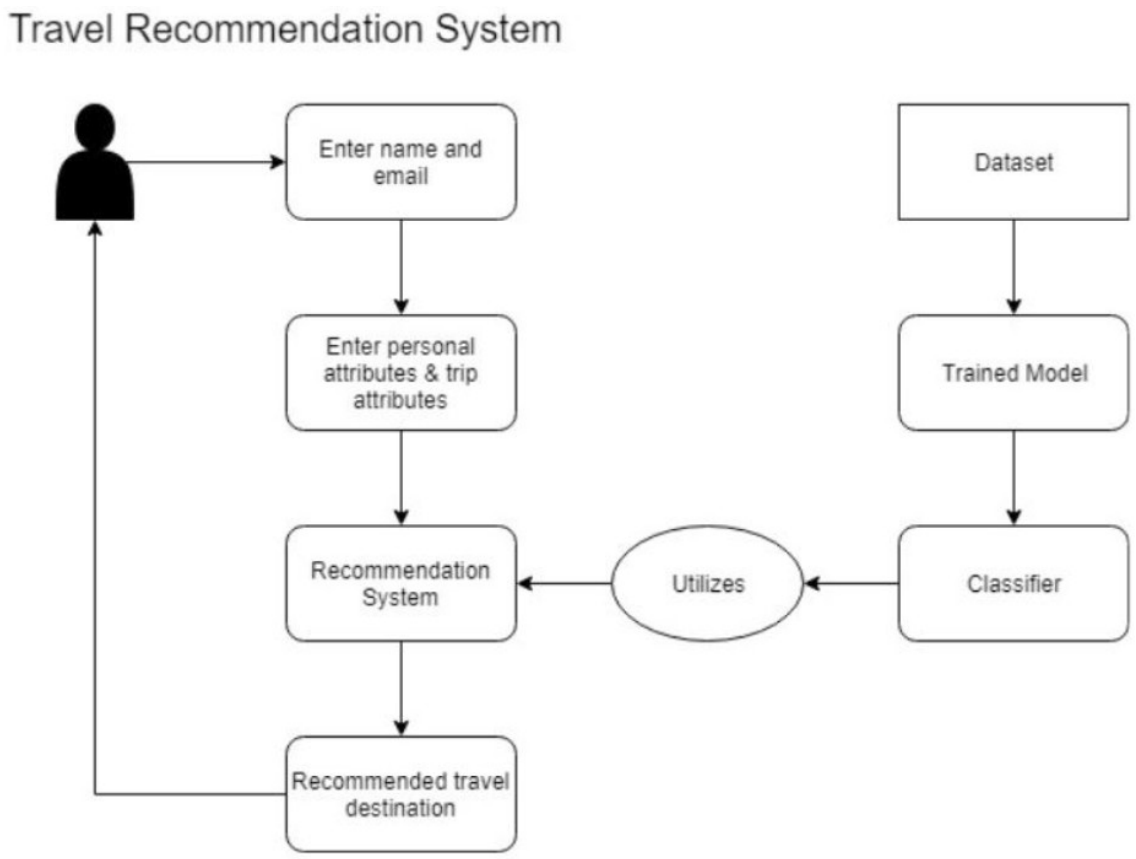


*Figure 3.1: System flow diagram*

The above diagram represents how a user interacts with the system and its modules

- The user first enters their valid credentials such as name & e-mail.
- Then, the user is expected to enter the user's & the trip's attributes.
- The details are fed to the recommendation system. The system utilizes trained classifier to produce the required output i.e. the destination
- The classifier is a Gaussian Naive Bayes classifier trained with the help of a prepared dataset.

# Dataset and System Snapshots

*Table 4.1: Data set for the TRS model:*

| age | occupation | accompanied by | area of interest | season of travel | duration of stay | rec. place |
|---|---|---|---|---|---|---|
| <30 | student | friends | sightseeing | summer | <=7 | mysore |
| <30 | student | family | sightseeing | summer | <=7 | mysore |
| <30 | student | friends | sightseeing | summer | <=7 | mysore |
| <30 | student | family | sightseeing | summer | <=7 | mysore |
| <30 | student | friends | sightseeing | winter | <=7 | mysore |
| <30 | student | family | sightseeing | winter | <=7 | mysore |
| <30 | student | friends | sightseeing | winter | <=7 | mysore |
| <30 | student | family | relaxation | winter | <=7 | mysore |
| <30 | salaried | friends | sightseeing | summer | <=7 | mysore |
| <30 | salaried | family | sightseeing | summer | <=7 | mysore |
| <30 | salaried | friends | sightseeing | summer | <=7 | mysore |
| <30 | salaried | family | sightseeing | summer | <=7 | mysore |
| <30 | salaried | friends | sightseeing | winter | <=7 | mysore |
| <30 | salaried | family | sightseeing | winter | <=7 | mysore |
| <30 | self-employed | friends | sightseeing | summer | <=7 | mysore |
| <30 | self-employed | family | sightseeing | summer | <=7 | mysore |
| <30 | self-employed | friends | sightseeing | summer | <=7 | mysore |
| <30 | self-employed | family | sightseeing | summer | <=7 | mysore |
| <30 | self-employed | friends | sightseeing | winter | <=7 | mysore |
| <30 | self-employed | family | sightseeing | winter | <=7 | mysore |
| <30 | self-employed | friends | sightseeing | winter | <=7 | mysore |
| <30 | self-employed | family | sightseeing | winter | <=7 | mysore |
| >30 | salaried | friends | sightseeing | summer | <=7 | mysore |
| >30 | salaried | family | sightseeing | summer | <=7 | mysore |
| >30 | salaried | friends | sightseeing | winter | <=7 | mysore |

| | | | | | | |
|---|---|---|---|---|---|---|
| >30 | salaried | family | sightseeing | winter | <=7 | mysore |
| >30 | salaried | friends | sightseeing | summer | <=7 | mysore |
| >30 | salaried | family | sightseeing | summer | <=7 | mysore |
| >30 | self-employed | friends | sightseeing | summer | <=7 | mysore |
| >30 | self-employed | family | sightseeing | summer | <=7 | mysore |
| >30 | self-employed | friends | sightseeing | winter | <=7 | mysore |
| >30 | self-employed | family | sightseeing | winter | <=7 | mysore |
| <30 | student | friends | adventure | winter | >7 | andaman |
| <30 | student | friends | relaxation | winter | >7 | andaman |
| <30 | student | family | adventure | winter | >7 | andaman |
| <30 | student | family | relaxation | winter | >7 | andaman |
| <30 | salaried | friends | adventure | winter | >7 | andaman |
| <30 | salaried | friends | relaxation | winter | >7 | andaman |
| <30 | salaried | family | adventure | winter | >7 | andaman |
| <30 | salaried | family | relaxation | winter | >7 | andaman |
| <30 | self-employed | friends | adventure | winter | >7 | andaman |
| <30 | self-employed | friends | relaxation | winter | >7 | andaman |
| <30 | self-employed | family | adventure | winter | >7 | andaman |
| <30 | self-employed | family | relaxation | winter | >7 | andaman |
| >30 | salaried | friends | adventure | winter | >7 | andaman |
| >30 | salaried | friends | relaxation | winter | >7 | andaman |
| >30 | salaried | family | adventure | winter | >7 | andaman |
| >30 | salaried | family | relaxation | winter | >7 | andaman |
| >30 | self-employed | friends | adventure | winter | >7 | andaman |
| >30 | self-employed | friends | relaxation | winter | >7 | andaman |
| >30 | self-employed | family | adventure | winter | >7 | andaman |
| >30 | self-employed | family | relaxation | winter | >7 | andaman |
| <30 | student | friends | sightseeing | winter | <=7 | delhi |
| <30 | student | family | sightseeing | winter | <=7 | delhi |
| <30 | salaried | friends | sightseeing | winter | <=7 | delhi |
| <30 | salaried | family | sightseeing | winter | <=7 | delhi |
| <30 | self-employed | friends | sightseeing | winter | <=7 | delhi |
| >30 | salaried | friends | sightseeing | winter | <=7 | delhi |

| | | | | | | |
|---|---|---|---|---|---|---|
| >30 | salaried | family | sightseeing | winter | <=7 | delhi |
| >30 | self-employed | friends | sightseeing | winter | <=7 | delhi |
| >30 | self-employed | family | sightseeing | winter | <=7 | delhi |
| <30 | student | friends | relaxation | winter | <=7 | goa |
| <30 | student | friends | adventure | winter | <=7 | goa |
| <30 | student | friends | relaxation | winter | <=7 | goa |
| <30 | student | family | sightseeing | winter | <=7 | goa |
| <30 | student | family | sightseeing | winter | <=7 | goa |
| <30 | student | family | sightseeing | winter | <=7 | goa |
| <30 | salaried | friends | relaxation | winter | <=7 | goa |
| <30 | salaried | friends | adventure | winter | <=7 | goa |
| <30 | salaried | friends | relaxation | winter | <=7 | goa |
| <30 | salaried | family | sightseeing | winter | <=7 | goa |
| <30 | salaried | family | adventure | winter | <=7 | goa |
| <30 | salaried | family | sightseeing | winter | <=7 | goa |
| <30 | self-employed | friends | relaxation | winter | <=7 | goa |
| <30 | self-employed | friends | adventure | winter | <=7 | goa |
| <30 | self-employed | friends | relaxation | winter | <=7 | goa |
| <30 | self-employed | family | sightseeing | winter | <=7 | goa |
| <30 | self-employed | family | sightseeing | winter | <=7 | goa |
| <30 | self-employed | family | sightseeing | winter | <=7 | goa |
| >30 | salaried | friends | relaxation | winter | <=7 | goa |
| >30 | salaried | friends | adventure | winter | <=7 | goa |
| >30 | salaried | friends | relaxation | winter | <=7 | goa |
| >30 | salaried | family | relaxation | winter | <=7 | goa |
| >30 | salaried | family | relaxation | winter | <=7 | goa |
| >30 | salaried | family | sightseeing | winter | <=7 | goa |
| >30 | self-employed | friends | relaxation | winter | <=7 | goa |
| >30 | self-employed | friends | adventure | winter | <=7 | goa |
| >30 | self-employed | friends | relaxation | winter | <=7 | goa |
| >30 | self-employed | family | relaxation | winter | <=7 | goa |
| >30 | self-employed | family | adventure | winter | <=7 | goa |

| | | | | | | |
|---|---|---|---|---|---|---|
| >30 | self-employed | family | sightseeing | winter | <=7 | goa |
| <30 | student | friends | relaxation | monsoon | <=7 | munnar |
| <30 | student | friends | sightseeing | monsoon | <=7 | munnar |
| <30 | student | family | sightseeing | monsoon | <=7 | munnar |
| <30 | student | family | sightseeing | monsoon | <=7 | munnar |
| <30 | salaried | friends | sightseeing | monsoon | <=7 | munnar |
| <30 | salaried | friends | sightseeing | monsoon | <=7 | munnar |
| <30 | salaried | friends | relaxation | monsoon | <=7 | munnar |
| <30 | salaried | family | sightseeing | monsoon | <=7 | munnar |
| <30 | salaried | family | sightseeing | monsoon | <=7 | munnar |
| >30 | salaried | friends | adventure | winter | >7 | andaman |
| >30 | self-employed | friends | adventure | winter | >7 | andaman |
| <30 | student | family | sightseeing | winter | >7 | andaman |
| <30 | salaried | family | sightseeing | winter | >7 | andaman |
| <30 | student | friends | sightseeing | winter | >7 | andaman |
| <30 | student | family | sightseeing | winter | >7 | andaman |
| <30 | student | friends | adventure | winter | <=7 | goa |
| <30 | self-employed | friends | adventure | winter | >7 | goa |
| <30 | salaried | friends | adventure | winter | >7 | goa |
| >30 | self-employed | friends | adventure | winter | <=7 | goa |
| >30 | salaried | friends | adventure | winter | <=7 | goa |
| <30 | salaried | friends | relaxation | winter | <=7 | goa |
| <30 | self-employed | friends | relaxation | winter | <=7 | goa |
| <30 | student | friends | relaxation | winter | <=7 | goa |
| <30 | self-employed | family | sightseeing | winter | <=7 | goa |
| <30 | salaried | family | sightseeing | winter | <=7 | goa |
| >30 | self-employed | family | sightseeing | winter | <=7 | goa |
| >30 | salaried | friends | adventure | summer | >7 | leh ladak |
| >30 | self-employed | friends | adventure | summer | >7 | leh ladak |
| <30 | salaried | friends | adventure | summer | >7 | leh ladak |
| <30 | self-employed | friends | adventure | summer | >7 | leh ladak |
| >30 | self-employed | friends | adventure | summer | >7 | leh ladak |

| | | | | | | |
|---|---|---|---|---|---|---|
| <30 | student | friends | adventure | summer | >7 | leh ladak |
| >30 | salaried | family | sightseeing | winter | <=7 | jaipur |
| >30 | salaried | family | sightseeing | winter | <=7 | jaipur |
| <30 | self-employed | family | sightseeing | winter | <=7 | jaipur |
| <30 | student | family | sightseeing | winter | <=7 | jaipur |
| >30 | self-employed | friends | sightseeing | winter | <=7 | jaipur |
| >30 | salaried | friends | sightseeing | winter | <=7 | jaipur |
| >30 | self-employed | family | sightseeing | summer | >7 | manali |
| >30 | salaried | family | sightseeing | summer | >7 | manali |
| <30 | student | family | sightseeing | summer | >7 | manali |
| <30 | salaried | family | sightseeing | summer | >7 | manali |
| <30 | self-employed | family | sightseeing | summer | >7 | manali |
| <30 | student | friends | relaxation | summer | <=7 | coorg |
| <30 | salaried | friends | relaxation | summer | <=7 | coorg |
| <30 | self-employed | friends | relaxation | summer | <=7 | coorg |
| <30 | salaried | family | relaxation | summer | <=7 | coorg |
| <30 | self-employed | family | relaxation | summer | <=7 | coorg |
| <30 | student | family | relaxation | summer | <=7 | coorg |
| >30 | salaried | friends | relaxation | summer | <=7 | coorg |
| >30 | self-employed | friends | relaxation | summer | <=7 | coorg |
| >30 | salaried | friends | adventure | summer | >7 | leh ladak |
| >30 | self-employed | friends | adventure | summer | >7 | leh ladak |
| <30 | salaried | friends | adventure | summer | >7 | leh ladak |
| <30 | self-employed | friends | adventure | summer | >7 | leh ladak |
| >30 | salaried | friends | adventure | summer | >7 | leh ladak |
| >30 | self-employed | friends | adventure | summer | >7 | leh ladak |
| <30 | student | friends | adventure | summer | >7 | leh ladak |
| <30 | student | family | sightseeing | winter | <=7 | delhi |
| <30 | salaried | family | sightseeing | winter | <=7 | delhi |
| >30 | salaried | family | sightseeing | winter | <=7 | delhi |
| >30 | self-employed | family | sightseeing | winter | <=7 | delhi |
| <30 | salaried | friends | relaxation | monsoon | >7 | munnar |

| | | | | | | |
|---|---|---|---|---|---|---|
| <30 | self-employed | friends | sightseeing | monsoon | <=7 | munnar |
| >30 | salaried | family | sightseeing | monsoon | >7 | munnar |
| >30 | self-employed | family | sightseeing | monsoon | <=7 | munnar |
| <30 | salaried | friends | sightseeing | monsoon | <=7 | munnar |
| <30 | self-employed | friends | sightseeing | monsoon | <=7 | munnar |
| <30 | student | friends | relaxation | monsoon | <=7 | munnar |
| <30 | student | family | sightseeing | monsoon | <=7 | munnar |
| <30 | salaried | family | sightseeing | monsoon | >7 | munnar |
| <30 | salaried | friends | relaxation | monsoon | <=7 | coorg |
| <30 | self-employed | friends | relaxation | monsoon | <=7 | coorg |
| >30 | salaried | family | relaxation | monsoon | <=7 | coorg |
| >30 | self-employed | family | relaxation | monsoon | <=7 | coorg |
| <30 | salaried | family | relaxation | monsoon | <=7 | coorg |
| <30 | self-employed | family | relaxation | monsoon | <=7 | coorg |
| <30 | student | friends | relaxation | monsoon | <=7 | coorg |
| <30 | student | family | relaxation | monsoon | <=7 | coorg |
| <30 | salaried | family | relaxation | monsoon | <=7 | coorg |
| <30 | student | friends | relaxation | monsoon | >7 | darjeeling |
| <30 | student | family | relaxation | monsoon | >7 | darjeeling |
| <30 | salaried | family | relaxation | monsoon | >7 | darjeeling |
| >30 | student | family | relaxation | monsoon | >7 | coorg |
| >30 | salaried | family | relaxation | monsoon | >7 | coorg |
| <30 | salaried | friends | sightseeing | monsoon | <=7 | coorg |
| <30 | self-employed | friends | sightseeing | monsoon | <=7 | coorg |
| >30 | salaried | family | sightseeing | monsoon | <=7 | coorg |
| >30 | self-employed | family | sightseeing | monsoon | <=7 | coorg |
| <30 | salaried | family | sightseeing | monsoon | <=7 | coorg |
| <30 | self-employed | family | sightseeing | monsoon | <=7 | coorg |
| <30 | student | family | sightseeing | monsoon | <=7 | coorg |
| <30 | student | family | sightseeing | monsoon | >7 | darjeeling |
| <30 | salaried | family | sightseeing | monsoon | >7 | darjeeling |
| >30 | student | family | sightseeing | monsoon | >7 | darjeeling |

| | | | | | | |
|---|---|---|---|---|---|---|
| >30 | salaried | family | sightseeing | monsoon | >7 | darjeeling |
| <30 | student | friends | relaxation | monsoon | >7 | darjeeling |
| <30 | salaried | friends | relaxation | monsoon | >7 | coorg |
| >30 | student | friends | relaxation | monsoon | >7 | coorg |
| >30 | salaried | friends | relaxation | monsoon | >7 | coorg |
| >30 | self-employed | friends | adventure | summer | >7 | leh ladak |
| <30 | salaried | friends | adventure | summer | >7 | leh ladak |
| >30 | salaried | friends | adventure | summer | >7 | leh ladak |
| <30 | salaried | friends | sightseeing | winter | <=7 | mysore |
| <30 | salaried | family | sightseeing | winter | <=7 | mysore |

This is the Home page of the Travel Recommendation System Website.

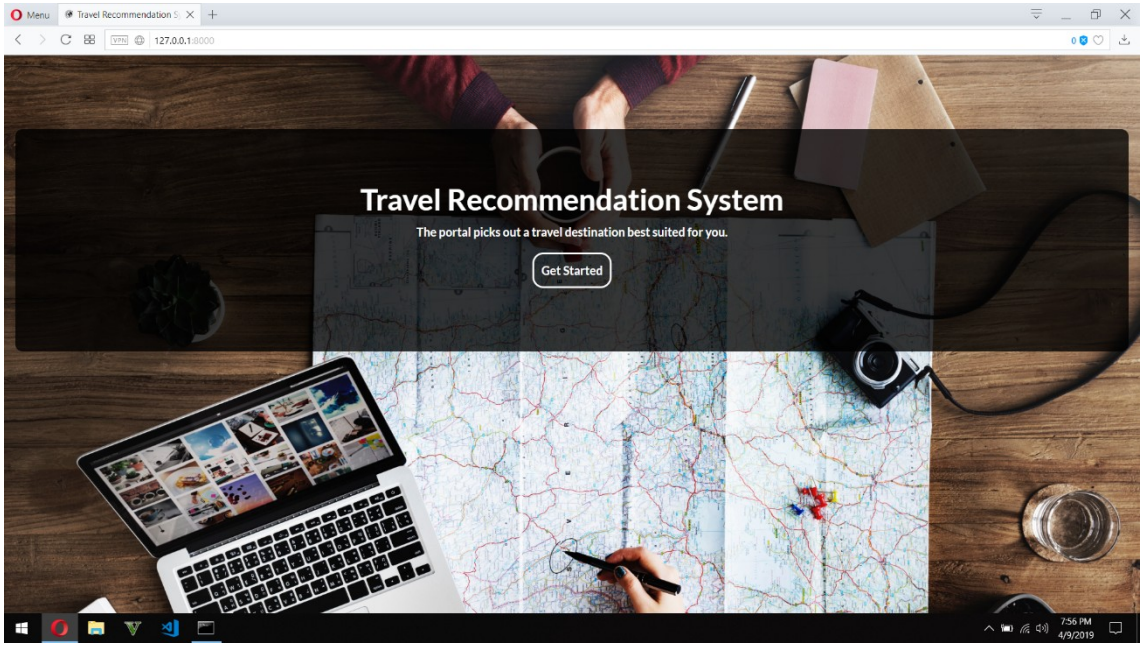


*Figure 4.1: Home page*

First, the user has to enter the details i.e Name and E-mail Id:

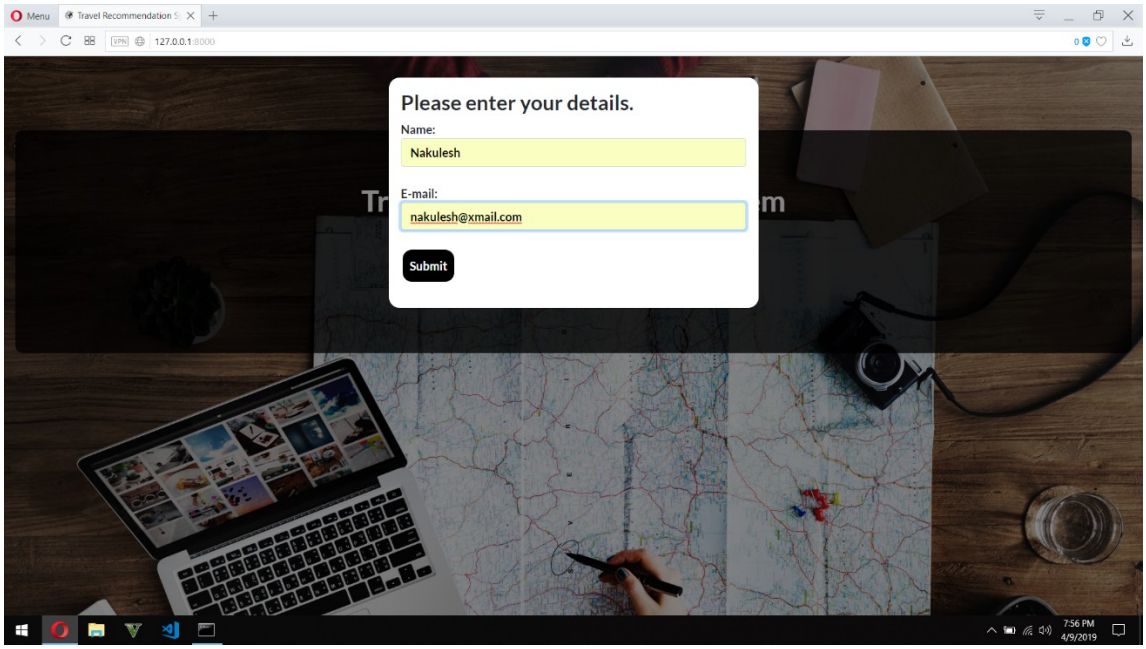


*Figure 4.2: Registration page*

Then, the user's personal details like age, Occupation and travel details like accompanied by, their month of travel, their interests (adventure, relaxation, sight-seeing), and the duration of their stay, are fed into the recommendation system.

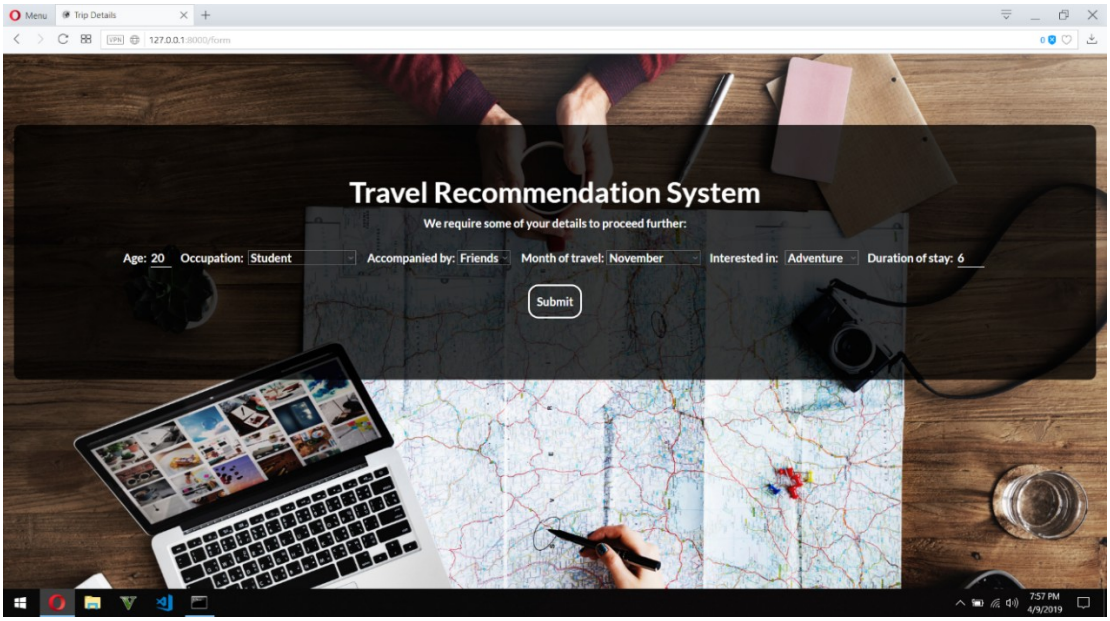


*Figure 4.3: Details Page*

The travel details are processed in the backend and according to those details, a place is recommended to the user.

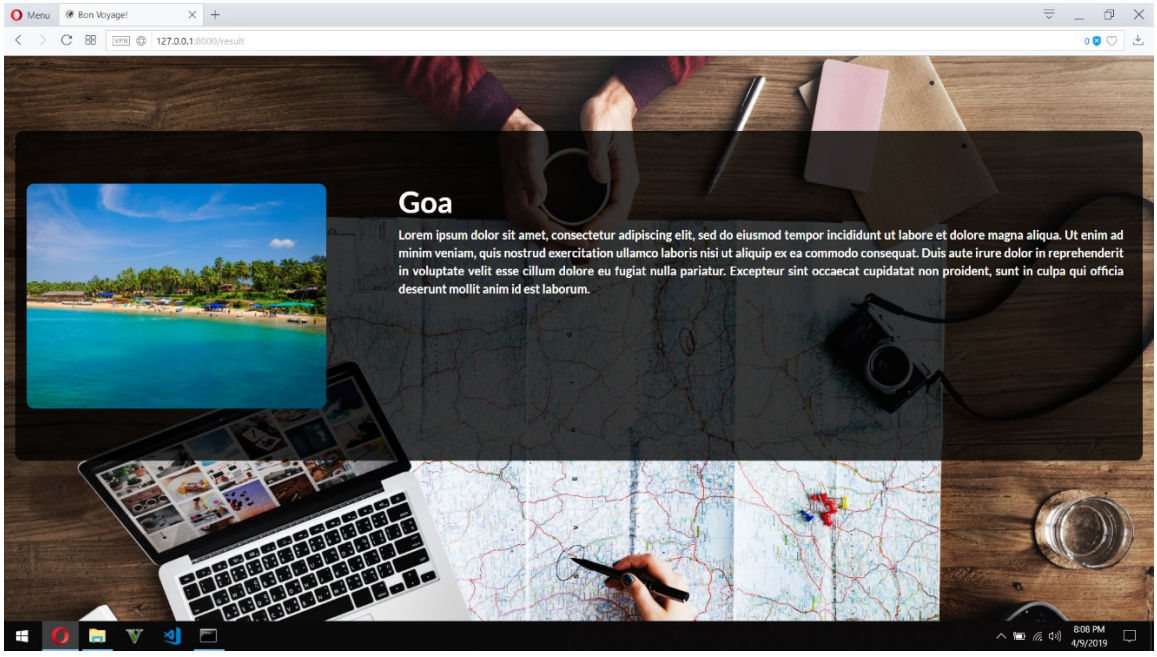


*Figure 4.4: Result page*

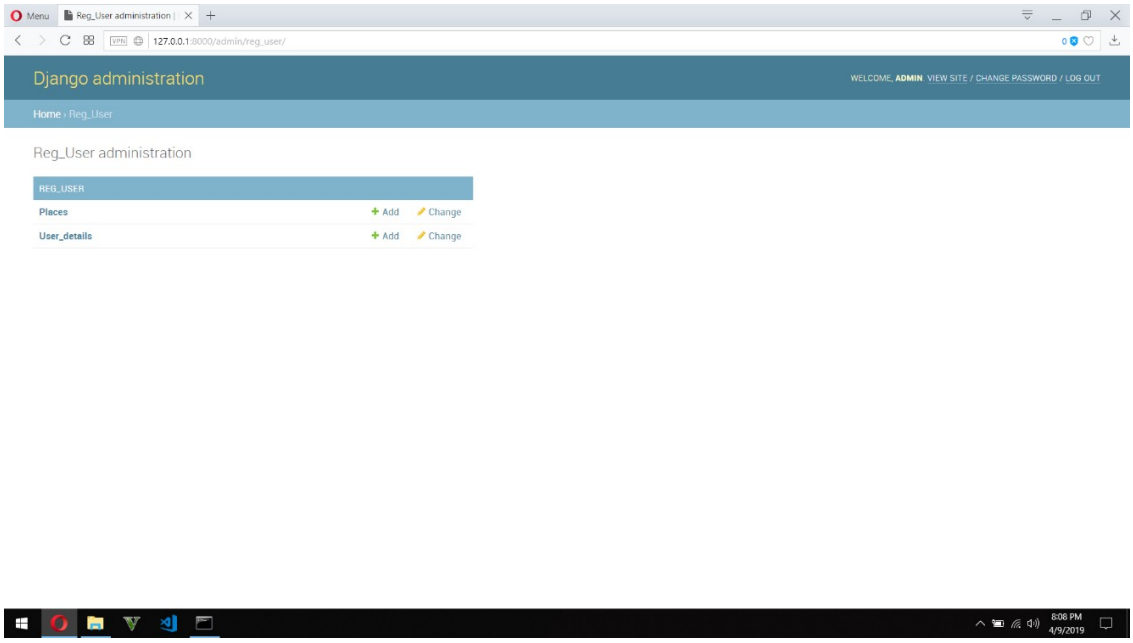


*Figure 4.5: Backend*

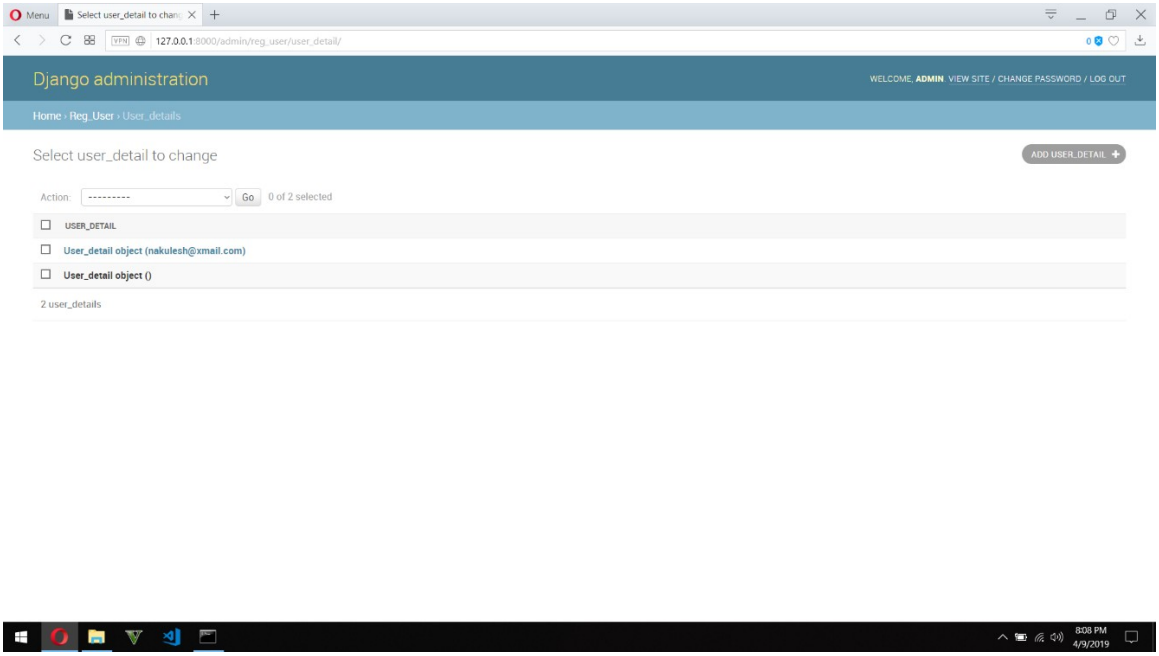


*Figure 4.6: E-mail List*

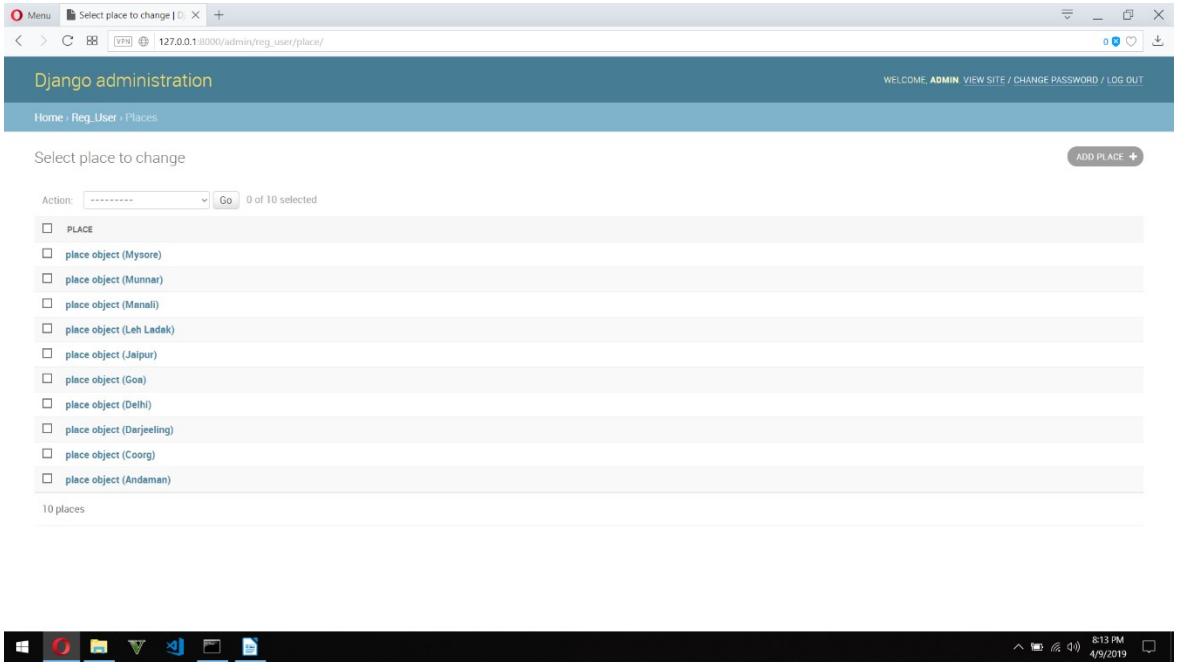


*Figure 4.7: Place details*

# Result

After a thorough study on Gaussian Naïve Bayes classifier and various other sklearn modules, we came to a conclusion that out of all the available modules available in sklearn for training the data, GaussianNB() function gave the most accurate results. While testing the output accuracy reached a maximum of 70% which proved to give the required output at runtime.

The Travel Recommendation System made sure that the user's needs were catered. Instead of depending on any kind of feedback or social media data, we decided to take user's input instead for getting more accurate results. We asked the users a few basic questions and these were taken as attributes for the module. The attributes included:

- User's age
- User's Interest
- User's month of travel
- User's accompanied by
- User's duration of travel
- User's occupation

Hence, we made sure that the system focused solely on the interests of the user and therefore we ended up getting accurate results of travel destination as output.

# Conclusions and Future Scope

Today, machine learning's massive success has led to it becoming the most dominant subset of AI that is practiced around the world. Whether its better fraud detection and prevention, the handy online recommendations made by Netflix and Amazon, revolutionary facial recognition technology, of futuristic self-driving cars, machine learning is powering the current artificial intelligence revolution. Machine learning can achieve some pretty impressive feats in AI buts its also responsible for simpler, but still incredibly useful applications.

There are many different algorithms in the machine learning domain. But, in our project, after making a comparison of two classification algorithms, Naïve Bayes Classifier algorithm gave the highest accurate results and therefore was used in the project.

In this project, we have restricted our scope to hotspot travel destinations across India. The user has to answer a few basic questions and a travel destination would be recommended to the user. In our future scope, the user can also give a feedback regarding the recommended place. If the user gives a positive feedback then that tuple would be appended to the dataset which in turn will help us in enhancing the accuracy of our model.

Also, the user will get a personalised itinerary which will include the places you can visit, what clothes you should carry during that season. It will also recommend places to visit, according to who you are travelling with. For example: friends, family. Also, it will recommend places according to the user's interest that was taken as an input. For example: adventure, relaxation or sight-seeing. Therefore, the user will get a personalised itinerary and can give a feedback accordingly.

# Bibliography


1. K. Kesorn, W. Juraphanthong, A. Salaiwarakul, “Personalised attraction recommendation system for tourists”, https://ieeexplore.ieee.org/document/8123926

2. Xinyu Wu, Zhou Huang, Xia Peng, Yiran Chen, Yu Liu, “Building a spatially embedded network for tourism hotspots using geo-tagged locations”, https://ieeexplore.ieee.org/document/8341493

3. Naive Bayes, https://scikit-learn.org/stable/modules/naive_bayes.html

4. Splitting Data into Training and testing data, https://medium.com/@contactsunny/how-to-split-your-dataset-to-train-and-test-datasets-using-scikit-learn-e7cf6eb5e0d

# Appendices

The Travel Recommendation System is a web application that is developed using the Django framework.

## 8.1. Appendix-A: Django Download and Installation

1. Install Python from https://www.python.org/downloads

2. On the command line type: pip install Django

3. To run Django server, from the command line interface navigate to the web app directory and type 'python manage.py runserver'

4. Run the web app using: https://127.0.0.1:8000

# Acknowledgement

We have great pleasure in presenting the report on **Travel Recommendation System.** We take this opportunity to express our sincere thanks towards our guide **Prof. Mayuri Jain** & Department of Computer, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Sachin Malave** Head of Department, Computer, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Amol Kalugade** BE project co-ordinator, Department of Computer, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

**Nakulesh Jayakrishnan**

**ID:17202012**

**Gauri Deshpande**

**ID: 16102046**

**Akanksha Koshti**

**ID:16102011**