CSE 587

DATA INTENSIVE COMPUTING

PROJECT - PHASE 2

Restaurant Recommendation System

NAME	UBIT	PERSON NUMBER
Shreya Dhareshwar	shreyadh	50471594
Rishab Aggarwal	rishabag	50475990

1. PROBLEM STATEMENT

“We will use zomato.csv to create a Restaurant Recommendation System and use a data driven solution to solve the problem of finding suitable locations to open restaurants”

a. Background:
Before the advent of food delivery apps like Zomato, Swiggy, Doordash and Uber Eats, the food industry operated primarily through physical restaurants and takeaways. Customers would have to visit restaurants or call them up to place their orders. The delivery of food was typically limited to a small radius around the restaurant, and it was often time-consuming and inconvenient for customers. However, the introduction of food delivery apps has brought about significant changes in the food industry like Increased convenience, Expanded customer base, Greater transparency, Improved delivery logistics, Innovation in food packaging leading to transforming the way people order food.

The purpose of this Zomato data is to analyze the location's demography for Bengaluru, India. The number of restaurants is growing with approximately 12,000 restaurants. Every day, new restaurants open their doors. However, competing with already established restaurants has become difficult for them. High real estate costs, rising food costs, a shortage of qualified workers, a fragmented supply chain, and over-licensing are among the key challenges they face.

b. Potential of the project to solve the problem:
The above problem can be solved by assisting new restaurants in determining their theme, menus, cuisine, cost, and so on for a specific location. It also aims to discover food similarities between Bengaluru neighborhoods. The dataset also includes reviews for each restaurant, which will aid in determining the overall rating for the establishment and help someone who wants to set up a new restaurant.

2. MODELLING

Different models have been implemented below to find the ideal location to open a new restaurant, different parameters like the location with maximum reviews, rating, top restaurant location, best cuisines have been taken into consideration.

1. Linear Regression

Linear regression is a statistical technique for determining the relationship between a dependent variable (usually denoted as 'y') and one or more independent variables (usually denoted as 'x'). The goal of linear regression is to find the best linear equation for the data and use it to make predictions.

Recommendation

The function *recommend_location* is used, which employs linear regression to determine the best location for a new restaurant based on certain criteria. It takes in a pandas DataFrame df as an argument.

The first step is to split the DataFrame into features and target variables. The features are 'votes', 'cost for two', and 'location', and the target variable is 'rate'.

Then we split the data into training and testing sets using train_test_split from scikit-learn. The test size is set to 20% of the data, and the random state is set to 42 for reproducibility. A linear regression model is created using LinearRegression() from scikit-learn, and it's fit to the training data using model.fit(X_train, y_train).

The model is used to make predictions on the test data using y_pred = model.predict(X_test). The performance of the model is evaluated using mean squared error (mean_squared_error(y_test, y_pred)) and R-squared score (r2_score(y_test, y_pred)).

A new DataFrame is created with all possible locations to open a restaurant. The possible values for 'votes' and 'cost for two' are generated using np.linspace() from NumPy, and the unique locations in the original DataFrame are used to create a list of possible locations.

The model is used to make predictions on the new DataFrame using y_pred = model.predict(new_df[['votes', 'cost for two', 'location']].values).

The location with the highest predicted rating is found using new_df['predicted rating'].idxmax() and new_df.loc[]

We then return a string with the model accuracy (MSE and R-squared) and the values for 'votes', 'cost for two', and 'predicted rating' for the best location.

Analysis

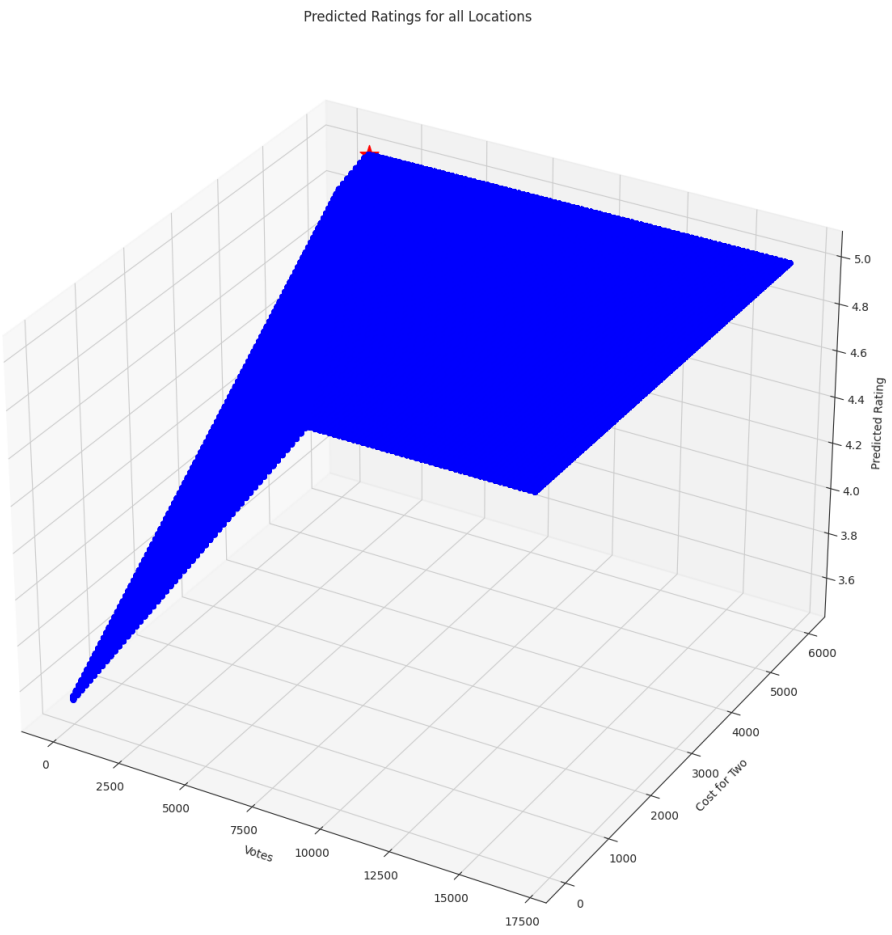
```
print("Linear Regression results:\n")
print(acc)
print("Votes:", votes)
print("Cost for two people:",cost_for_2)
print("Predicted rating:", predicted_rating)
```

```
Linear Regression results:

Model accuracy: MSE = 0.11994468835726683, R-squared = 0.23422790109697322
Votes: 1178.24
Cost for two people: 6000.0
Predicted rating: 5.0

Recommended location for opening a new restaurant from linear regression model:
'Indiranagar'
```

We can also use the predicted ratings to optimize the menu, price range, and other factors for the new restaurant that we are planning to set up.

For example, if the analysis indicates that there is a high demand for Italian cuisine in a particular area, we can focus on offering Italian dishes on our menu. So, using linear regression to predict ratings for all locations can be a powerful tool for identifying areas with high demand for good restaurants and optimizing the menu and other factors when setting up a new restaurant

Visualization



A 3D scatter plot is created using Matplotlib to visualize the predicted ratings for all locations. The location with the highest predicted rating is marked with a red star.

2. Logistic Regression

Introduction:

Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

Here in our dataset, logistic regression can be used to predict the probability of a particular location being successful for a new restaurant based on certain factors such as the cuisine type, the average cost for two people, the rating, and other factors.

We can first identify the factors that are likely to influence the success of a new restaurant. These factors can be used as predictor variables in the logistic regression model. The response variable can be whether a restaurant in a particular location is

successful or not, where success can be defined as having a high rating and a large number of positive reviews.

Recommendation

We first split the dataset into features (X) and the target variable (y). The features include the number of votes, the cost for two people, and the location. The target variable is a binary variable that indicates whether a restaurant is rated 4.0 or higher (1 if the rating is 4.0 or higher, 0 otherwise).

Then we split the data into train and test sets using the "train_test_split" function from the Scikit-learn library and created a Logistic Regression model using the "LogisticRegression" function from Scikit-learn, and fits the model to the training data. Then we made predictions on the test dataset using the "predict" method of the Logistic Regression model.

The model's performance is evaluated using the "accuracy_score" function from Scikit-learn.

The model is used to make predictions on the new dataframe using the "predict_probability" method of the Logistic Regression model. The predicted probability of success is calculated for each location in the dataframe.

Analysis

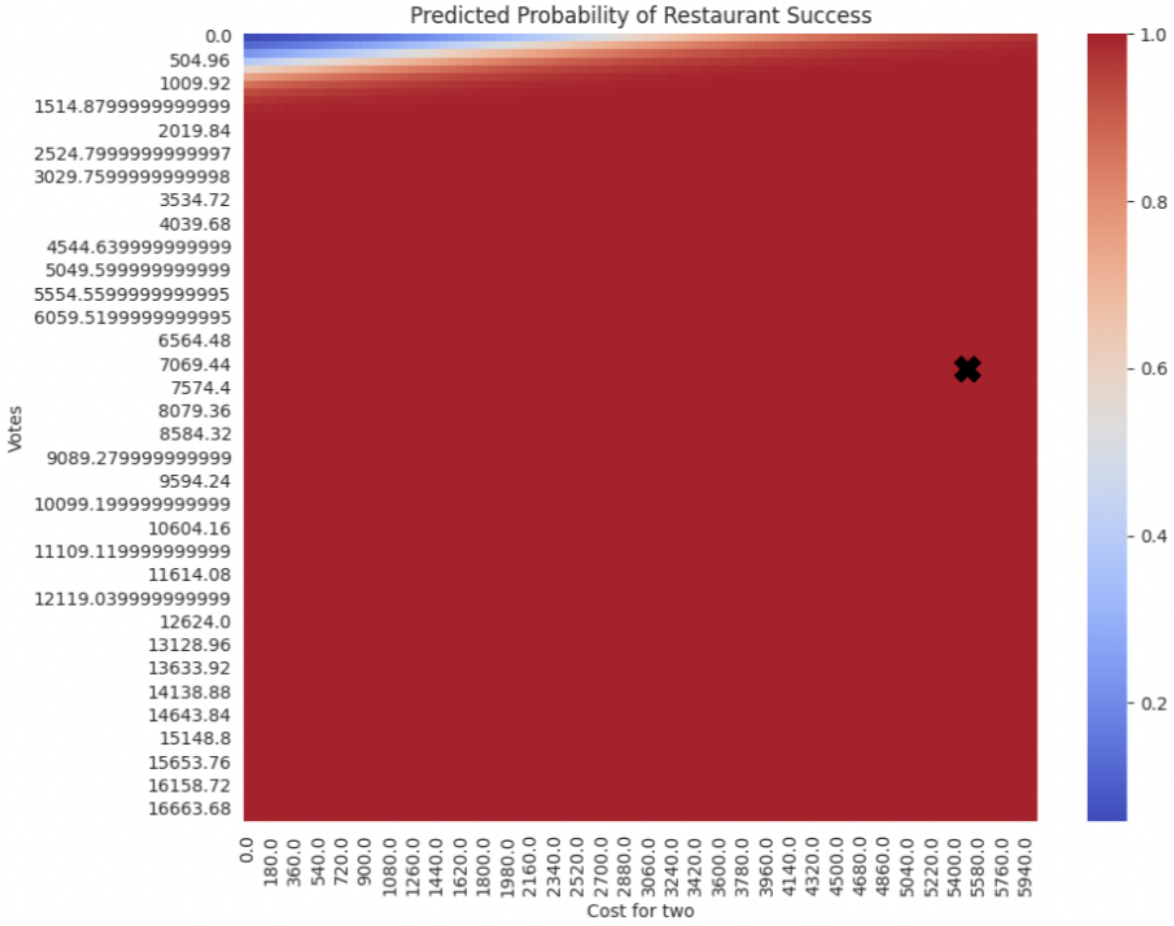
```
print("Logistic Regression results:\n")
print(acc)
print("Votes:", votes)
print("Cost for two people:",cost_for_2)
print("Predicted Probability of success:", predicted_rating)
```

```
Logistic Regression results:

Model accuracy: Accuracy score = 0.8589896579156723
Votes: 7237.759999999999
Cost for two people: 5520.0
Predicted Probability of success: 1.0

Recommended location for opening a new restaurant from logistic regression model:
'Indiranagar'
```

With the highest predicted probability of success, we can evaluate the feasibility and profitability(using additional market research) of opening a restaurant in that area. This may include studying the competition, assessing the local demand for a particular cuisine, and analyzing the demographics of the area. By combining the insights provided by logistic regression with additional research, we can make a more informed decision on where to set up a new restaurant in Bangalore.

Visualization

The results are visualized using a heatmap that shows the predicted probability of success for each combination of votes and cost for two people. The best location is also marked on the heatmap represented as 5760 which is int value of Indiranagar.



3. KNN

Introduction

K-Nearest Neighbors (KNN) is a popular algorithm used in recommendation systems for item-based collaborative filtering. In KNN, items are represented as vectors of features, such as genres, actors, or ratings. The algorithm identifies the K closest neighbors (items with similar feature vectors) to a given item, and recommends items that those neighbors have been rated highly. KNN is a simple and intuitive algorithm that doesn't require training or complex calculations.

Recommendation

K-Nearest Neighbors (KNN) algorithm is used to recommend a location for opening a restaurant in Bangalore, based on data on votes, cost for two, and rating of existing restaurants in the city. Again first we split the dataset into features and target variables (X and y), and then split the data into training and testing sets using the train_test_split function from the sklearn library. It then creates a KNN model with 5 neighbors and fits it to the training data using the fit method.Then we predict the target variable (y_pred) for the testing data and calculate the accuracy of the model using the accuracy_score function from sklearn.

It then uses the KNN model to make predictions on this new dataset, and calculates the predicted probability of success for each location.

Then we find the location with the highest predicted probability of success (best_location) and plot the predicted probabilities for different locations using matplotlib. Then we return a tuple with the model accuracy, the votes and cost for two of the recommended locations, and the predicted probability of success for that location.

Analysis

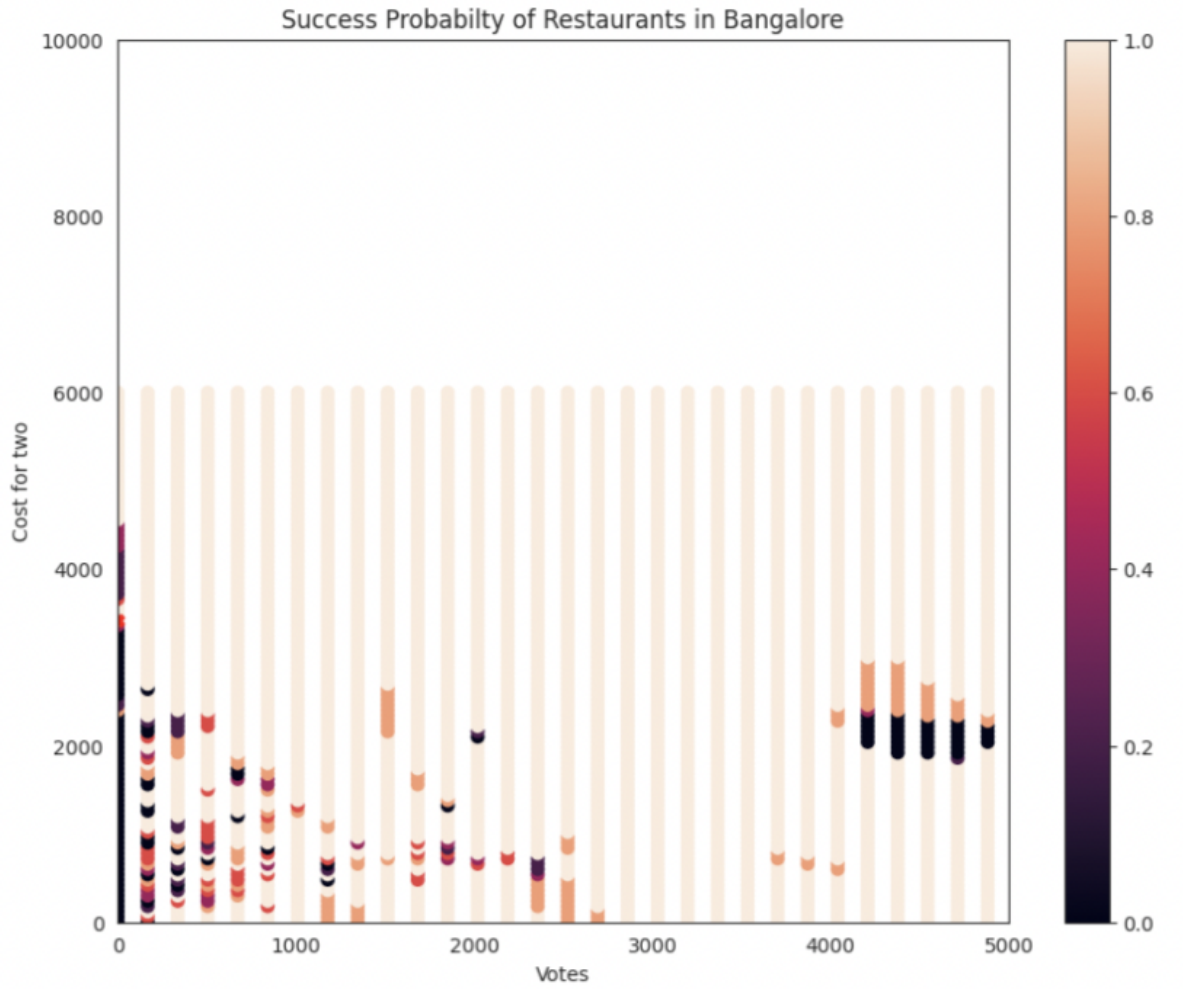
```
print("KNN results:\n")
print(acc)
print("Votes:", votes)
print("Cost for two people:",cost_for_2)
print("Predicted Probability of success:", predicted_rating)
```

```
KNN results:

Model accuracy: 0.9120922832140016
Votes: 0.0
Cost for two people: 3420.0
Predicted Probability of success: 1.0
```

```
Recommended location for opening a new restaurant from KNN model:
'Brigade Road'
```

Visualization

We plot the predicted probabilities for different locations using matplotlib. Then we return a tuple with the model accuracy, the votes and cost for two of the recommended locations, and the predicted probability of success for that location.



4. Matrix Factorization

Introduction

Matrix Factorization is a technique for reducing the dimensionality of a large matrix by decomposing it into two or more matrices of lower dimension. In this code, matrix factorization is used to find the best location to set up a new restaurant with a particular cuisine type.

Recommendation

Here we filter the dataframe to include only restaurants serving the specified cuisine. Then, we create a pivot table of the filtered dataset, with the average rating of each location serving the specified cuisine.

Next, we convert the pivot table into a matrix where the rows represent the locations and the columns represent the users (in this case, there is only one user per location). Then split this matrix into training and test sets, and train a neural network model using the training data.

After training the model, the function which we created evaluates it on the test data and computes the mean squared error. It then uses the model to get embeddings for each location and computes the pairwise cosine similarities between the locations based on their embeddings.

Finally, we get the mean squared error, the name of the location with the highest similarity score, and the highest similarity score.

Analysis

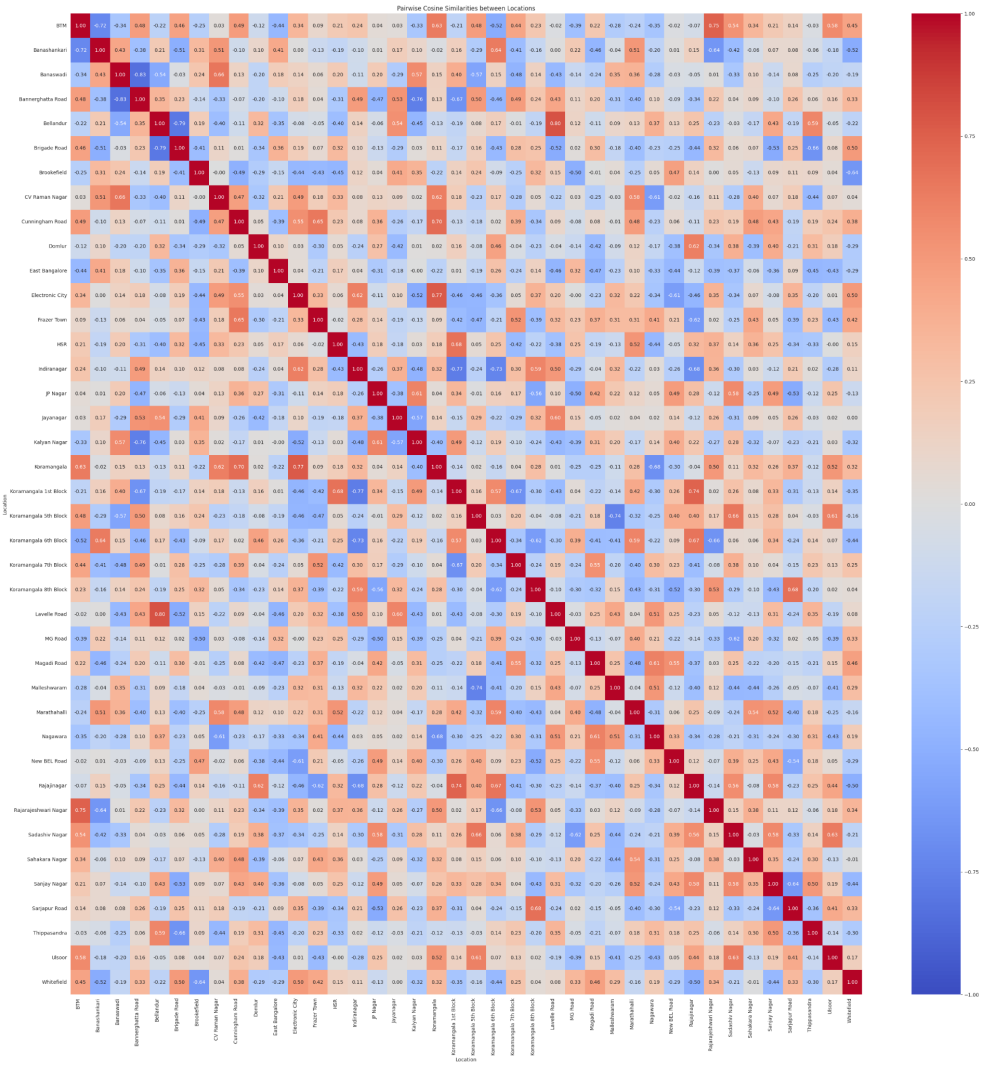
```
print("Matrix factorization results:\n")
print(acc)
print("Similarity score:", similarity_score)
```

```
Matrix factorization results:

Mean Squared Error: 12.396944999694824
Similarity score: 1.0000001
```

```
Recommended location for opening a new restaurant for Mexican cuisine from Matrix factorization:
 Cunningham Road
```

Visualization

Heatmap of the cosine similarity matrix is created using seaborn. It is used to plot the similarity scores between 2 locations.



5. Random Forest

Introduction

Random Forest is a machine learning algorithm that can be used for both classification and regression tasks. It is an ensemble learning method that creates multiple decision trees and combines their predictions to obtain a more accurate and stable prediction.

Random Forest can be used to identify the important factors that influence the success of a restaurant in a given location. These factors could include the cuisine type, cost for two, location, and rating of nearby restaurants, and other demographic factors like population density, income levels, and age distribution of the local population.

Using a dataset like the Zomato Bangalore restaurants dataset, we could use Random Forest to train a model that can predict the success of a restaurant based on these factors. This would involve splitting the dataset into training and test data.

Recommendation

Like other models, here also we created one recommend_location which uses a Random Forest Classifier to predict the probability of a restaurant being rated above 4.0 based on its votes and cost for two.

The function takes a pandas dataframe df as input, which is assumed to have columns named 'votes', 'cost for two', and 'rate'. The 'rate' column is assumed to contain the rating of the restaurant in question.

The code first splits the data into training and testing sets, and then trains a Random Forest Classifier on the training set using the sklearn.ensemble module. It then predicts the ratings for the testing set and calculates the accuracy of the model using the sklearn.metrics module.

The code then creates a new dataframe of possible locations based on the maximum values of the 'votes' and 'cost for two' columns in the input dataframe, and predicts the probability of each location being rated above 4.0 using the trained model. The function then returns a string with the model's accuracy, as well as the 'votes', 'cost for two', and predicted probability for the location with the highest predicted probability.

. ### Analysis

```
print("Random Forest Classifier results:\n")
print(acc)
print("Votes:", votes)
print("Cost for two people:",cost_for_2)
print("Predicted Probability of success:", success_probability)
```

```
Random Forest Classifier results:

Model accuracy: 0.927903739061257
Votes: 168.32
Cost for two people: 2400.0
Predicted Probability of success: 1.0
```

```
Recommended location for opening a new restaurant for North Indian cuisine from Random Forest Classifier :
'Brigade Road'
```

From above image,

Random Forest Classifier results:

Model accuracy: 0.927903739061257

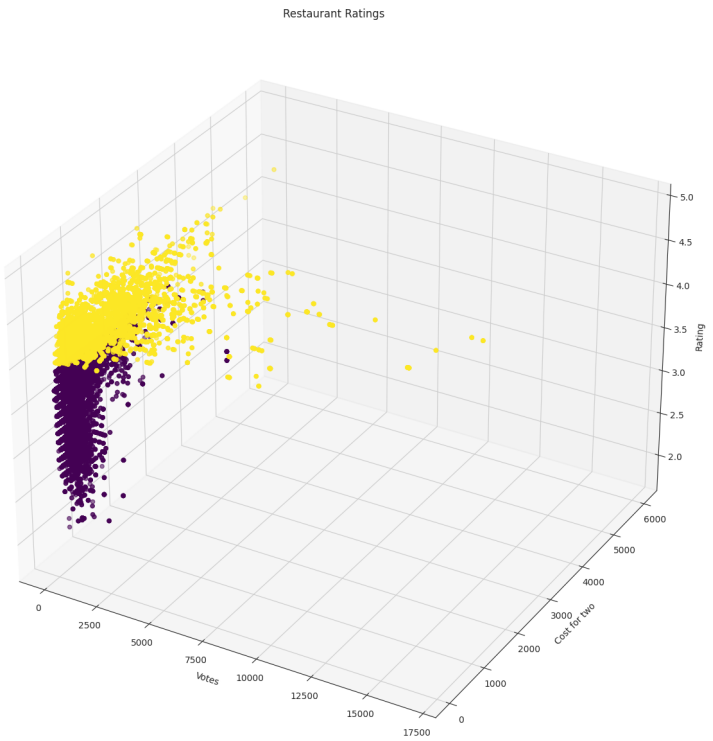
Votes: 168.32

Cost for two people: 2400.0

Predicted Probability of success: 1.0

Visualization

A 3D scatter plot using the matplotlib module is created to visualize the relationship between the three variables: 'votes', 'cost for two', and 'rate'.



6. Bag of Words

Introduction:

A bag-of-words model is a technique used to represent text as a collection of the frequency of individual words that appear in a document. In this case, the system uses customer reviews of restaurants to recommend the top-rated cuisines among the top-rated restaurants. It is to visualize and analyze textual data, such as customer reviews or menu items, for a set of restaurants.

Recommendation:

We first start by extracting and cleaning the relevant data from a given dataset and converting it into a Pandas DataFrame. Selection of the top-rated restaurants is done based on the number of reviews and creates word clouds for each of them to get a sense of the commonly used words. Filtering of the data to include only the top-rated restaurants is then done and a new DataFrame consisting of the reviews, ratings, and cuisines is created. The system then groups the data by cuisine and computes the mean rating for each cuisine to identify the most popular cuisines among the top-rated restaurants.

Analysis

By creating word clouds for the top 9 restaurants in the provided dataset, we can gain insights into what customers are saying about these restaurants and what types of cuisine they offer. By analyzing the cuisines offered by successful restaurants in the area, one can gain insights into the preferences of the local customers. One can then use this information to tailor the menu to suit the tastes of the target market.

Visualization

Word cloud for the top 9 restaurants

Grouping and sorting of the top rated restaurants by cuisines

	cuisines	rating
4	Burger, Fast Food, Roast Chicken, Finger Food	4.666667
7	Desserts, Ice Cream	4.364865
17	North Indian, Fast Food	4.000000
15	Mithai, Street Food	3.992138
8	Desserts, Ice Cream, Beverages, Sandwich	3.955556
22	Pizza, Cafe, Italian	3.922348
13	Ice Cream, Desserts, Beverages, Sandwich	3.819853
11	Ice Cream	3.661058
18	North Indian, Fast Food, Street Food	3.566489
2	Biryani, Kerala, Mughlai, Arabian, North India...	3.555556
20	North Indian, South Indian	3.500000
21	North Indian, South Indian, Kerala	3.500000
3	Biryani, North Indian, Andhra	3.500000
19	North Indian, Mughlai, South Indian, Chinese	3.494050
10	Fast Food, Burger	3.332519
5	Cafe	3.317164
23	Roast Chicken, Burger	3.138434
14	Kerala, Seafood, South Indian, Chinese, North ...	3.093023
12	Ice Cream, Desserts	3.070243
6	Cafe, Fast Food	3.041096
1	Bakery, Desserts	2.781013
9	Fast Food	2.601724
0	Bakery	2.329856
16	North Indian	2.000000

The most popular cuisines among the top restaurants is: Burger, Fast Food, Roast Chicken, Finger Food

Modeling Summary and Evaluation

We applied several models to analyze the Zomato dataset and make recommendations for the best location and cuisine for opening a restaurant in Bangalore. We used Linear regression, Logistic regression, Random Forest, and KNN to recommend the best location for opening a restaurant. Among these models, KNN had the highest accuracy. Additionally, we used Matrix Factorization to recommend the best location for opening a restaurant of a particular cuisine type. Finally, we applied the Bag of Words model to find top-rated restaurants based on reviews, grouped the data by cuisine, and determined the cuisine of the best-rated restaurants. Overall, our analysis provides valuable insights into the restaurant industry in Bangalore and demonstrates what would be the best locations for opening a new restaurant in bangalore using machine learning models.

References

1. kaggle - https://www.kaggle.com/datasets/himanshupoddar/zomato-bangalore-restaurants
2. matplotlib - https://matplotlib.org/stable/plot_types/index
3. pandas - https://pandas.pydata.org/docs/reference/general_functions.html
4. seaborn - https://seaborn.pydata.org/api.html
5. w3schools - https://www.w3schools.com/python/matplotlib_intro.asp
6. techtarget - https://www.techtarget.com/searchbusinessanalytics/definition
7. towardsdatascience - https://towardsdatascience.com/all-machine-learning-models-explained-in-6-minutes-9fe30ff6776a
8. Random Forest- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
9. Bag of words- https://machinelearningmastery.com/gentle-introduction-bag-words-model/
10. Matrix factorisation : https://realpython.com/build-recommendation-engine-collaborative-filtering/