Murali Krishna Chintha

Restaurant Ratings Prediction

Table of Contents

[1. Introduction 2](#_Toc42644447)

[1.1. Data 2](#_Toc42644448)

[1.2. Technologies Used: 2](#_Toc42644449)

[2. Process Flow 3](#_Toc42644450)

[2.1. Retraining the model 5](#_Toc42644451)

[2.1.1. Data Validation Check 5](#_Toc42644452)

[2.1.2. Data Splitting 6](#_Toc42644453)

[2.1.3. Data Pre-processing 6](#_Toc42644454)

[2.1.4. Training Different Models 7](#_Toc42644455)

[2.1.5. Choosing the best model based on the metric 8](#_Toc42644456)

[2.1.6. Saving the model 8](#_Toc42644457)

[2.2. Bulk Prediction 8](#_Toc42644458)

[**2.2.1.** **Data Validation Check** 9](#_Toc42644459)

[2.2.2. Data Pre-processing 9](#_Toc42644460)

[2.2.3. Loading the model and predicting using the model. 10](#_Toc42644461)

[2.3. Single Value Prediction 11](#_Toc42644462)

[2.3.1. Data Validation Check 12](#_Toc42644463)

[2.3.2. Loading the model and predicting the data 12](#_Toc42644464)

[3. Concepts 12](#_Toc42644465)

[3.1. Random Forest Classifier 12](#_Toc42644466)

[3.2. XGBoost Classifier 13](#_Toc42644467)

[3.3. GridSearchCV 13](#_Toc42644468)

1. Introduction

Restaurant Rating has become the most commonly used parameter for judging a restaurant for any individual. A lot of research has been done on different restaurants and the quality of food it serves. Rating of a restaurant depends on factors like reviews, area situated, average cost for two people, votes, cuisines and the type of restaurant.

## Data

This dataset has been obtained by scraping the TA website for information about restaurants.

|  |  |
| --- | --- |
| Feature | Description |
| **Name** | name if the restaurant |
| **City** | city location of the restaurant |
| **Cuisine Style** | cuisine style(s) of the restaurant, in a Python list object (94 046non-null) |
| **Rating** | rate of the restaurant on a scale from 1 to 5, as a float object (115 658 non-null) (Target Column) |
| **Ranking** | rank of the restaurant among the total number of restaurants in the city as a float object (115 645 non-null) |
| **Price Range** | price range of the restaurant among 3 categories , as a categorical type (77 555 non-null) |
| **Number of Reviews** | number of reviews that customers have let to the restaurant, as a float object (108 020 non-null) |
| **Reviews** | 2 reviews that are displayed on the restaurants scrolling page of the city, as a list of list object where the first list contains the 2 reviews, and the second le dates when these reviews were written (115 673 non-null) |
| **URL\_TA** | part of the URL of the detailed restaurant page that comes after 'www.tripadvisor.com' as a string object (124 995 non-null) |
| **ID\_TA** | identification of the restaurant in the TA database constructed a one letter and a number (124 995 non-null |

## Technologies Used:

|  |  |
| --- | --- |
| IDE | PyCharm |
| Database | MySQL |
| Frontend | HTML5, CSS3, Bootstrap |
| Integration | Flask |
| Deployment | Google Cloud Platform |

# Process Flow

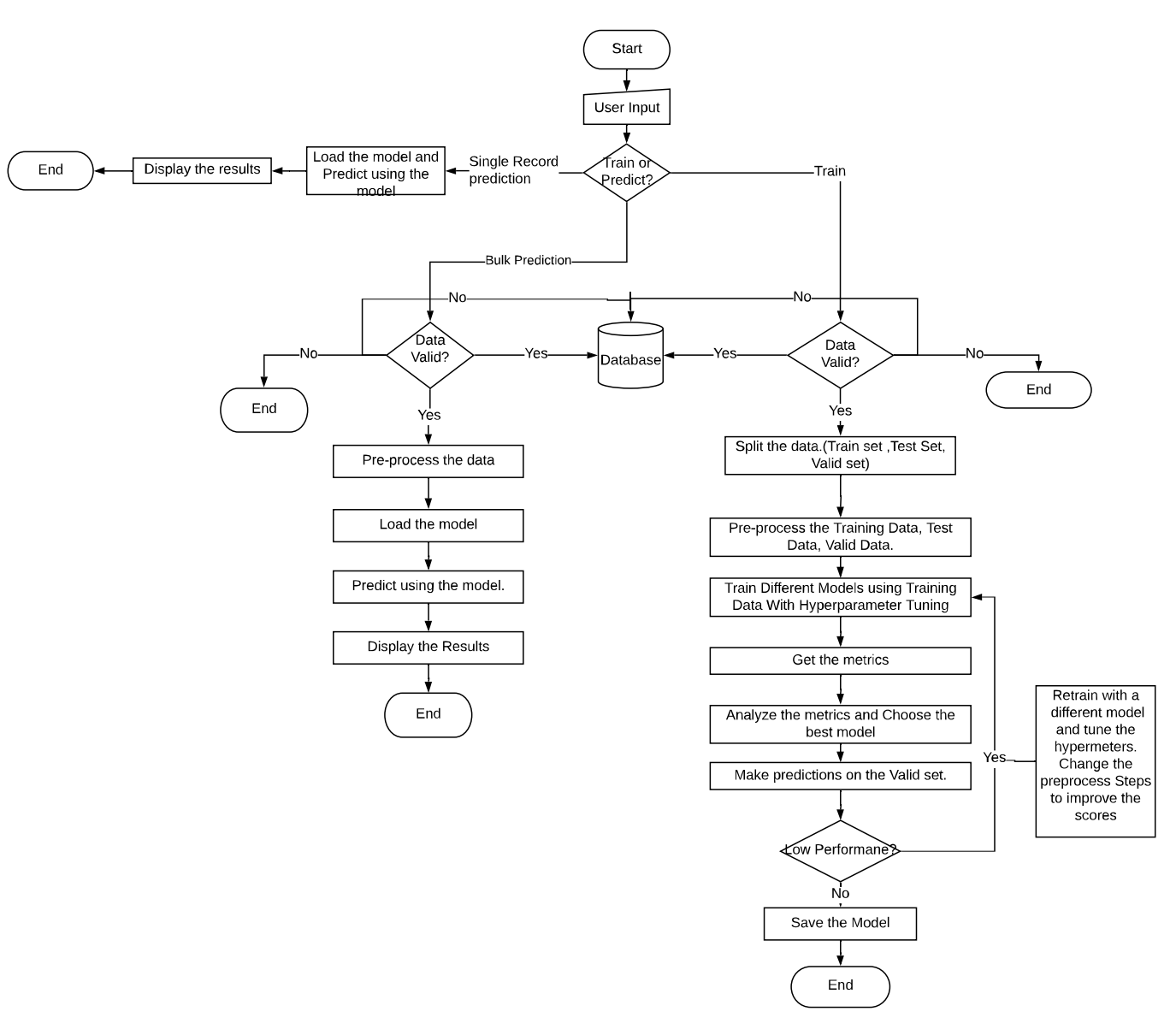


Fig 1: Flow chart explaining the process.

The Flow chart above explains the process flow of the entire solution. There are three paths to the flow based on the user input –

1. Retraining the model
2. Bulk Prediction
3. Single Value Prediction.

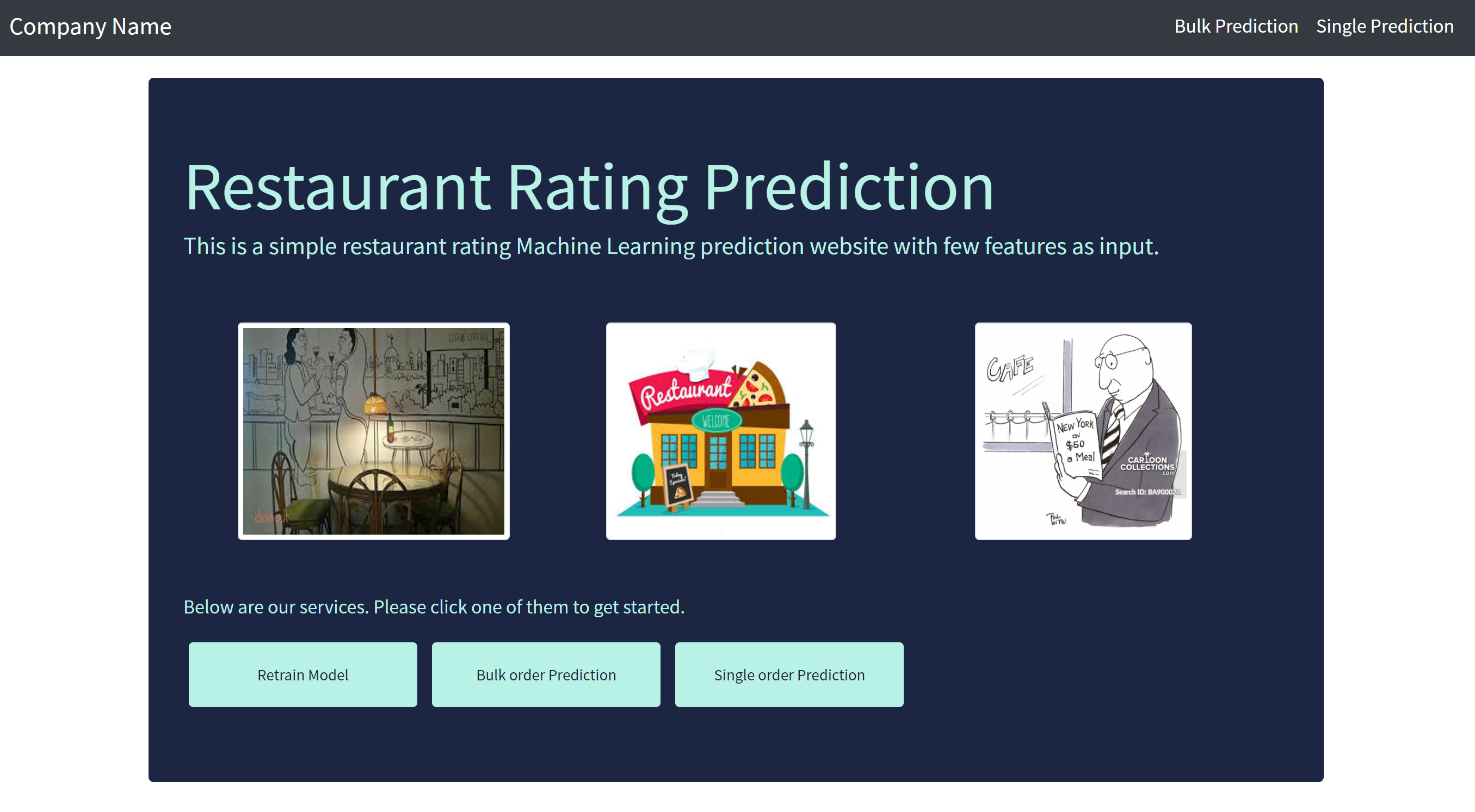


Fig 2: Home Page for user input.

## Retraining the model

The steps involved in retraining the model are

1. Data Validation Check
2. Data Splitting
3. Data Pre-processing
4. Training Different Models
5. Choosing the best model based on the metric.
6. Saving the model.

### Data Validation Check

Data Validation is performed to check if the data provided is valid or not. The main tasks in data validation check are –

1. Checking the number of columns agreed as per SLA
2. Checking the datatypes of each column agreed as per SLA.
3. Checking the column names agreed as per SLA.
4. Checking if any of the columns have more than 75% null values. In this case,data will be considered as invalid.

If the data meets all the four conditions then the data is considered to be Valid data. If it does not meet the conditions, data will be considered as invalid data.

Both Valid and Invalid data are pushed into the database.

### Data Splitting

The data for retraining the model will be split into three unequal sets in the ratio 70:15:15.

1. Training set.
2. Validation set.
3. Test set.

Training and Validation sets will be used for training different models and choosing the best one among them. Test set will be used to validate the chosen model.

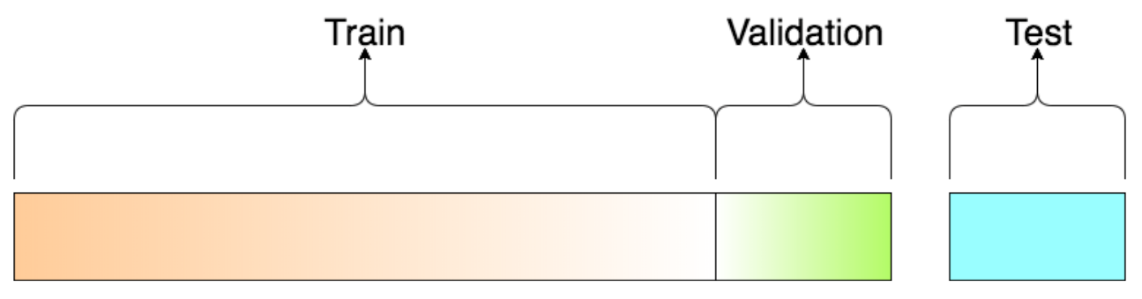


Fig 3: Diagram of Train Validation Test splitting of the data.

### Data Pre-processing

Data pre-processing is the most important step in training the model. In this step, we will prepare the data to be fed the model for training. Pre-processing includes –

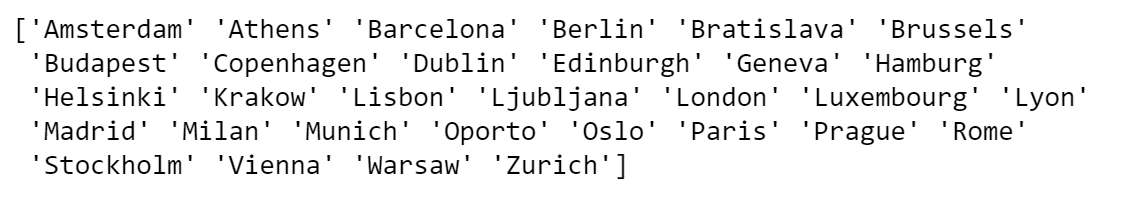
1. Dealing with null values - Null values in the “Price Range” column will be replaced with the median value. After replacing null values in the ‘Price Range’ columns, the rows containing null values will be dropped.

Imputation of the data to remove the null values cannot be used in these datasets because some of the columns like ‘Ranking’ loose their distribution if the null values are imputed.

Note: If the null values in any column are greater than 50% of the total data. The data will be considered as invalid.

1. Converting categorical variables – Categorical columns are converted into numerical columns. In this dataset the categorical variables are
2. City – Since City has 31 categories. Feature Hasher is to convert the column into

7 different columns. These columns will be appended to the original dataset and ‘City’ will be dropped. The categories in ‘City’ column are



1. Price Range – This column has 3 categories namely



These categories are converted into-

'$' - 0

'$$ - $$$' – 1

'$$$$' - 2

1. Feature Extraction – Only features that explains the variance in the dataset are used. All the other columns will be dropped. The dropped columns in this data are
2. Name
3. URL\_TA
4. ID\_TA
5. Creating new features from the existing features – New features have been created from the existing features from two columns.
6. Review – Review columns consists of string containing two reviews and corresponding dates. I have extracted the two reviews and added two new features which contains the sentiment of the reviews. Vader sentiment analyser has been used to analyse the sentiment of the reviews.
7. Cuisines offered – This column consists of a list of cuisines offered by reach restaurant. Since list of cuisines does not add any variance to the data. Column named ‘Number of cuisines offered’ has been added to the data which contains the number of cuisines offered by each restaurant.

### Training Different Models

After pre-processing, the preprocessed data will be used to train different models. The models used are

1. Random Forest Classifier
2. XGBoost Classifier
3. OneVsRest Classifier (with XGBoost as the base classifier)

GridSearchCV has been used for hyperparameter optimization of the models. Each model will be trained on the pre-processed data using GridSearchCV.

### Choosing the best model based on the metric

After training different and tuning the respective hyperparameters, best model will chose based on best f1 score. The metric used here is the F1-Score. The model with the highest F1-score will be chosen.

### Saving the model

After the model has been chosen. The existing model will be deleted and the new model will be dumped into a pickle file which can be used for loading the model any number of times.

## Bulk Prediction



Fig 4: Page for Bulk predictions

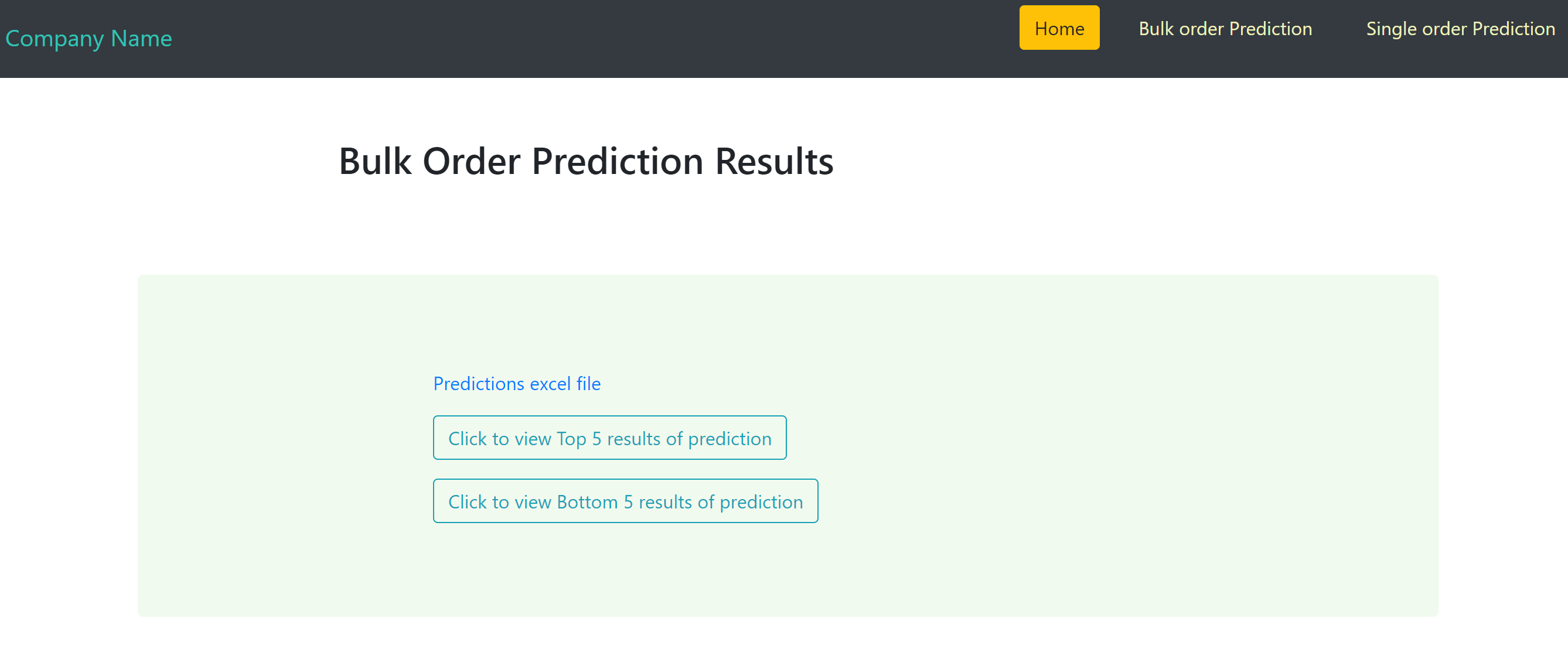


Fig 5: Bulk prediction results page

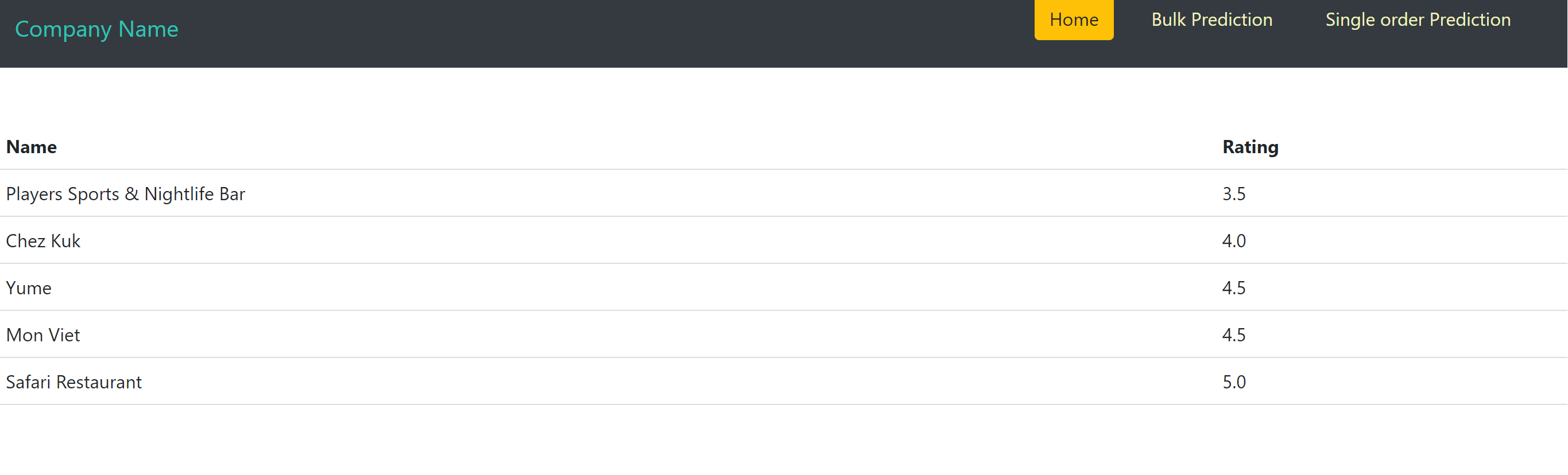


Fig 6: Page showing the top 5 rows of the bulk prediction.

The steps involved in predicting the model are

1. Data Validation Check
2. Data Pre-processing
3. Loading and Predicting using the model

### **Data Validation Check**

Data Validation is performed to check if the data provided is valid or not. The main tasks in data validation check are –

1. Checking the number of columns agreed as per SLA
2. Checking the datatypes of each column agreed as per SLA.
3. Checking the column names agreed as per SLA.
4. Checking if any of the columns have more than 75% null values. In this data will considered as invalid.

If the data meets all the three conditions then the data is considered to be Valid data. If it does not meet the conditions, data will be considered as invalid data.

### Data Pre-processing

Similar to the steps in training the model, Data pre-processing will be performed on the prediction data. In this step, we will prepare the data for prediction. Pre-processing includes –

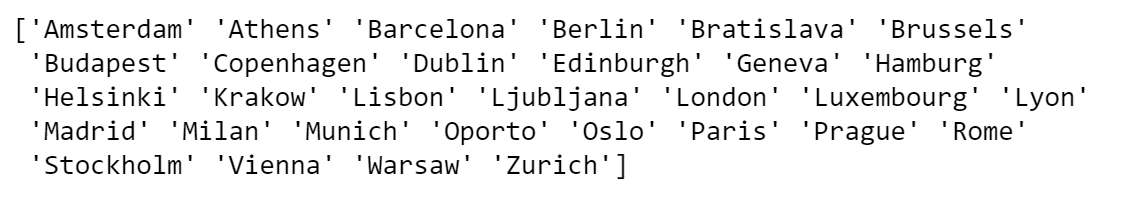
1. Dealing with null values - Null values in the “Price Range” column will be replaced with the median value. After replacing null values in the ‘Price Range’ columns, the rows containing null values will be dropped.

Imputation of the data to remove the null values cannot be used in these datasets because some of the columns like ‘Ranking’ loose their distribution if the null values are imputed.

Note: If the null values in any column are greater than 50% of the total data. The data will be considered as invalid.

1. Converting categorical variables – Categorical columns are converted into numerical columns. In this dataset the categorical variables are
2. City – Since City has 31 categories. Feature Hasher is to convert the column into

7 different columns. These columns will be appended to the original dataset and ‘City’ will be dropped. The categories in ‘City’ column are



1. Price Range – This column has 3 categories namely



These categories are converted into-

'$' - 0

'$$ - $$$' – 1

'$$$$' - 2

1. Feature Extraction – Only features that explains the variance in the dataset are used. All the other columns will be dropped. The dropped columns in this data are
2. Name
3. URL\_TA
4. ID\_TA
5. Creating new features from the existing features – New features have been created from the existing features from two columns.
6. Review – Review columns consists of string containing two reviews and corresponding dates. I have extracted the two reviews and added two new features which contains the sentiment of the reviews. Vader sentiment analyser has been used to analyse the sentiment of the reviews.
7. Cuisines offered – This column consists of a list of cuisines offered by reach restaurant. Since list of cuisines does not add any variance to the data. Column named ‘Number of cuisines offered’ has been added to the data which contains the number of cuisines offered by each restaurant.

### Loading the model and predicting using the model.

After pre-processing the data for prediction, the saved model will be loaded from the pickle file and the prediction data will be given to the model as input. The output will be combined with the Restaurant name and will be displayed.

## Single Value Prediction

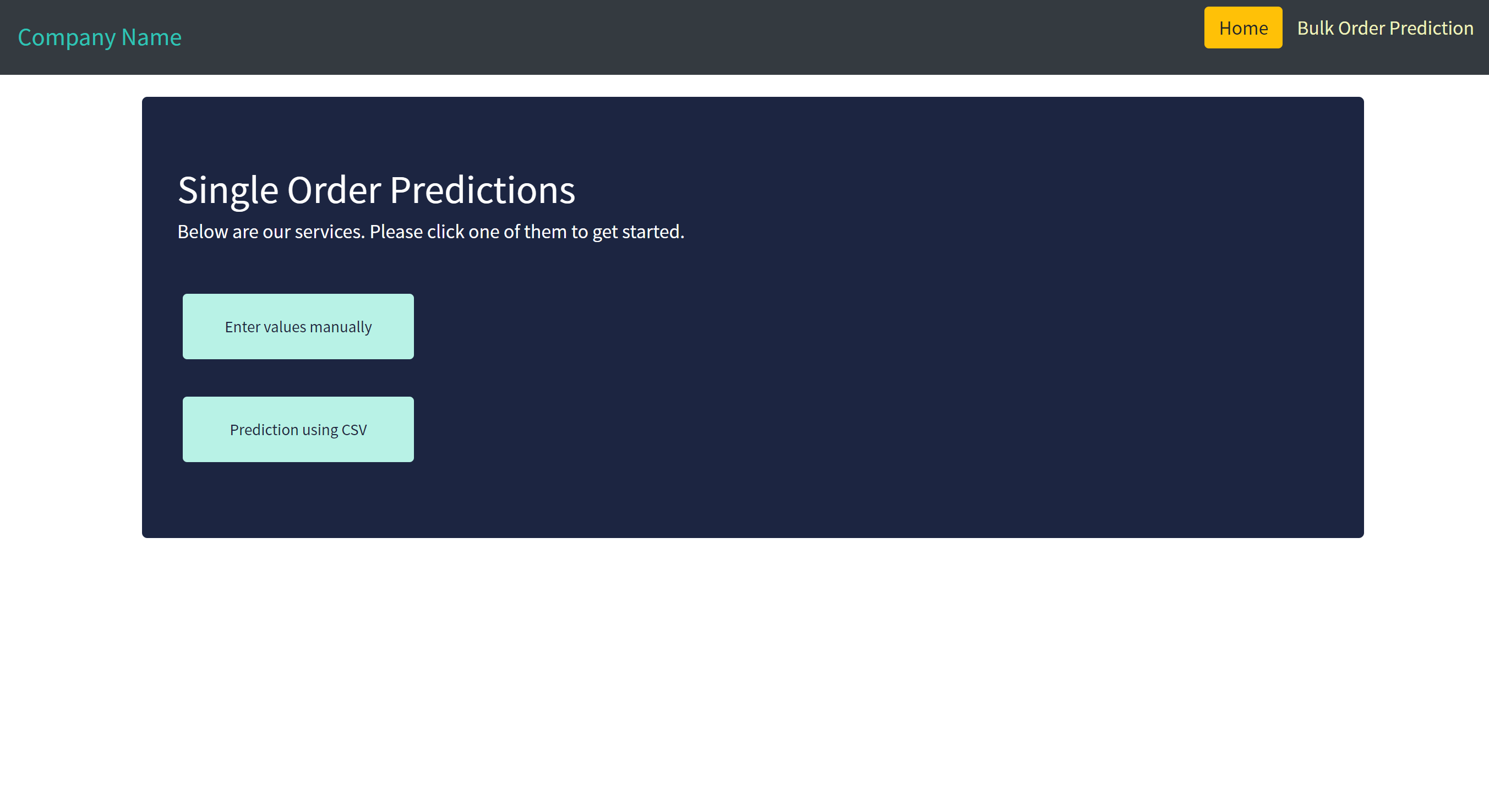


Fig 4: Single Value Prediction page.

The user can choose to enter the values manually or can upload the CSV file containing a single record.

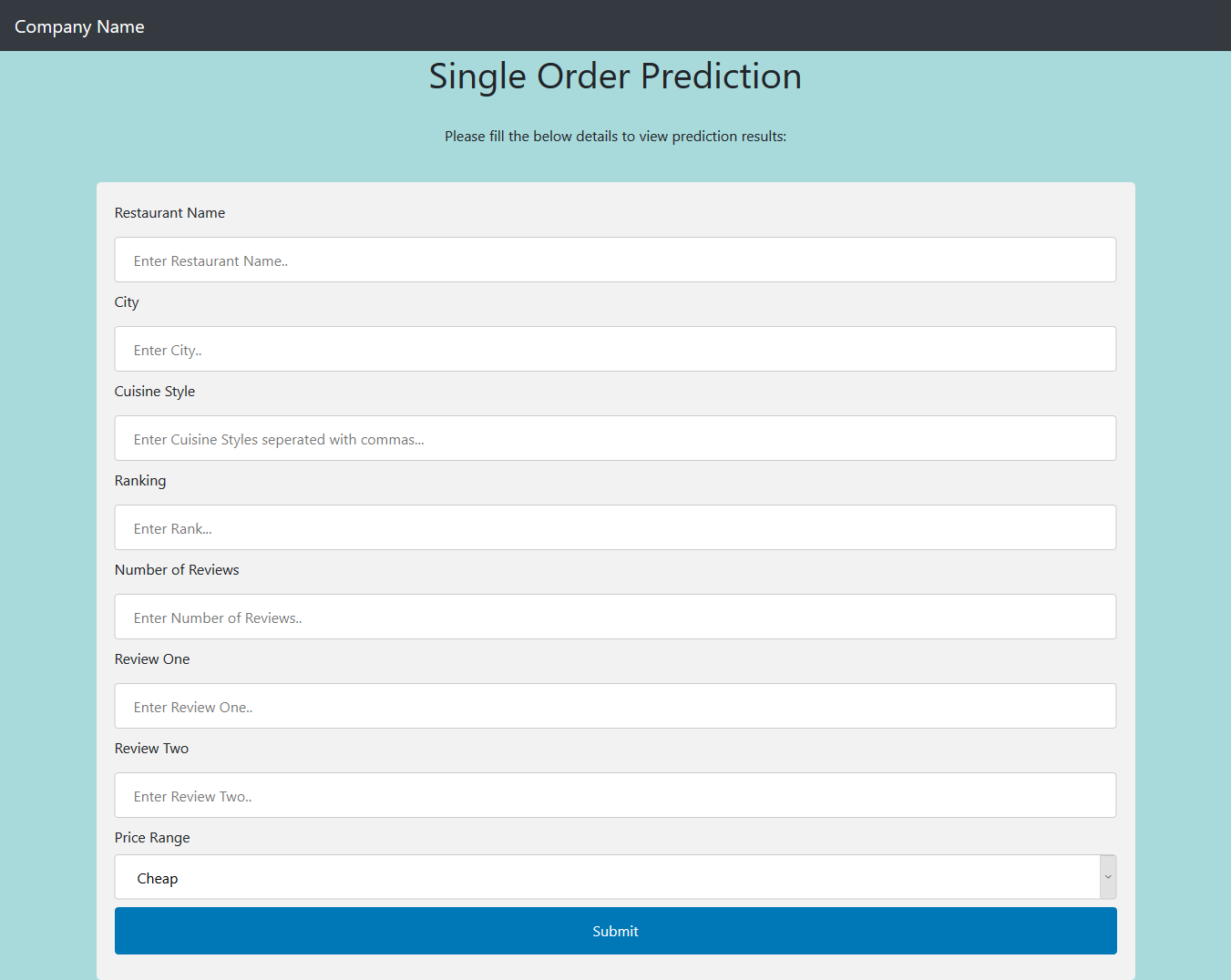


Fig 5: Page displaying the fields for entering manually.

The Steps for prediction if the user chose to enter the values manually are

1. Data Validation Check.
2. Loading the model and predicting the data.

### Data Validation Check

Data Validation is performed to check if the data provided is valid or not. The main tasks in data validation check are –

1. Checking the data types of each of the variables entered by the user.
2. Checking if any of the entered values are null values.

If the data meets all the conditions then the data is considered to be Valid data. If it does not meet the conditions, data will be considered as invalid data.

### Loading the model and predicting the data

After validating the data for single value prediction, the saved model will be loaded from the pickle file and the prediction data will be given to the model as input. The output will be combined with the Restaurant name and will be displayed.

\*\*\*\* Add a pic after the results of single value prediction\*\*\*\*

# Concepts

## Random Forest Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. In random forests, each tree in the ensemble is built from a sample drawn with replacement from the training set. Also, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model. In our model, the number of estimators used are 5 and we have considered ‘Entropy’ as a measure of the quality of a split.

## XGBoost Classifier

Extreme Gradient Boosting (XGBoost) is built on the principles of gradient boosting framework. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. XGBoost uses a more regularized model formalization to control over-fitting, which gives it better performance. In our model, the number of estimators used are 100. The model internally uses log-linear classifier for regularizing the model with λ = 1.

## GridSearchCV

GridSearchCV is a library function that is a member of sklearn’s model selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

In addition to that, you can specify the number of times for the cross-validation for each set of hyperparameters.