

# RESTAURANTS DATASET



Laury FYOT-TAMADON  
Sohana HASSANALY

# Data Analysis with Python

- Introduction
- Exploratory Data Analysis
- Model Predictions
- Flask API
- Conclusion

# Introduction

## Presentation of the project :

The goal was to analyze our dataset by first performing exploratory data analysis. We then made model predictions by applying Machine Learning techniques in order to select the best one. With our final model, we created a Flask API to make some fun predictions about restaurants.

By following these steps, here is what we present to you :

# Introduction

Dataset : **Zomato Restaurants**

Source : <https://www.kaggle.com/shrutimehta/zomato-restaurants-data>

Link to code: <https://github.com/SHFRANCE/Data-Analysis-Project->

For more details on our analysis, please check our code 😊

## Description of the dataset:

We are using the Zomato Restaurants dataset and the Country Code dataset. The first dataset contains information about 9951 restaurants from 15 countries. Using the information about these 9951 restaurants, the challenge is to build a model to predict the **Average Cost for two** based on the following fields:

- **Restaurant ID** (int64) - ID of the restaurant
- **Restaurant Name** (object) - Name of the restaurant
- **Country Code** (int64) - Country code
- **City** (object) - City where the restaurant is located
- **Address** (object) - Postal address of the restaurant
- **Locality** (object) - Location of the restaurant in the city
- **Locality Verbose** (object) - Detailed description of the location
- **Longitude** (float64) - Longitude of the location
- **Latitude** (float64) - Latitude of the restaurant
- **Cuisines** (object) - Types of cuisine served in the restaurant
- **Average Cost for two** (int64) - Average cost of a two-persons meal
- **Currency** (object) - Currency of the country
- **Has Table booking** (object) - Possibility of booking a table or not (Yes/No)
- **Has Online delivery** (object) - Possibility of online delivery or not (Yes/No)
- **Is delivering now** (object) - Possibility of delivery at the moment (Yes/No)
- **Switch to order menu** (object) - Possibility of ordering with a switch button (Yes/No)
- **Price range** (int64) - Price range of the restaurant
- **Aggregate rating** (float64) - Average rating of the restaurant
- **Rating color** (object) - Category of 5 colors representing the average rating
- **Rating text** (object) - Category of 5 attributes to qualify the quality of the restaurant
- **Votes** (int64) - Number of votes for the restaurant

The second dataset represents the country names of the first dataset associated to their country codes.

# Exploratory Data Analysis

The main goal of Exploratory Data Analysis is to dive into the dataset and explore everything it has to offer.

## What Data are we Using ?

We first got to know the dataset, so we decided to start by exploring the different columns and all the values they contain.

- The first main discovery we made is that **8652** restaurants of this dataset are located in **India**.
- At first, it seems that our dataset does not explicitly contain more than 9 missing values. However, when we looked more precisely, we realized that restaurants with a total rating of 0 or a rating set as 'Not rated' were implicit missing values.

We then decided to separate our study with the **continuous features** on one hand and **categorical features** on the other.

# Exploratory Data Analysis

## Read in Data

In this section, we checked the values in the columns. We also looked at missing values and noticed that some values were incomplete and decided to treat them as missing values.

We dropped the columns that we did not need.

# Exploratory Data Analysis

## Explore Continuous Features

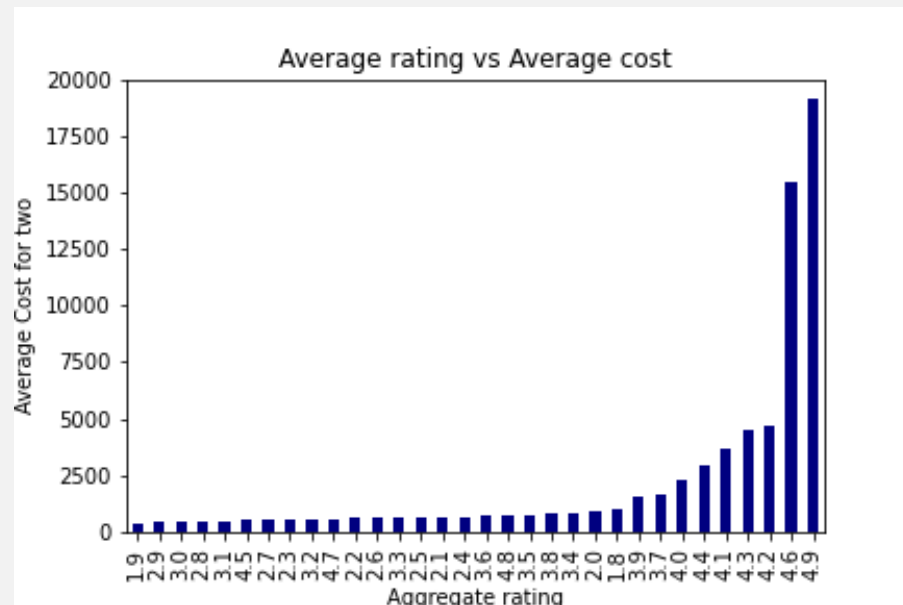
In this section, we focused on the quantitative values: *Aggregate rating*, *Average Cost for two*, *Price range* and *Votes*.

- We first seen that the price range affects the rating of restaurants.
- We noticed



# Exploratory Data Analysis

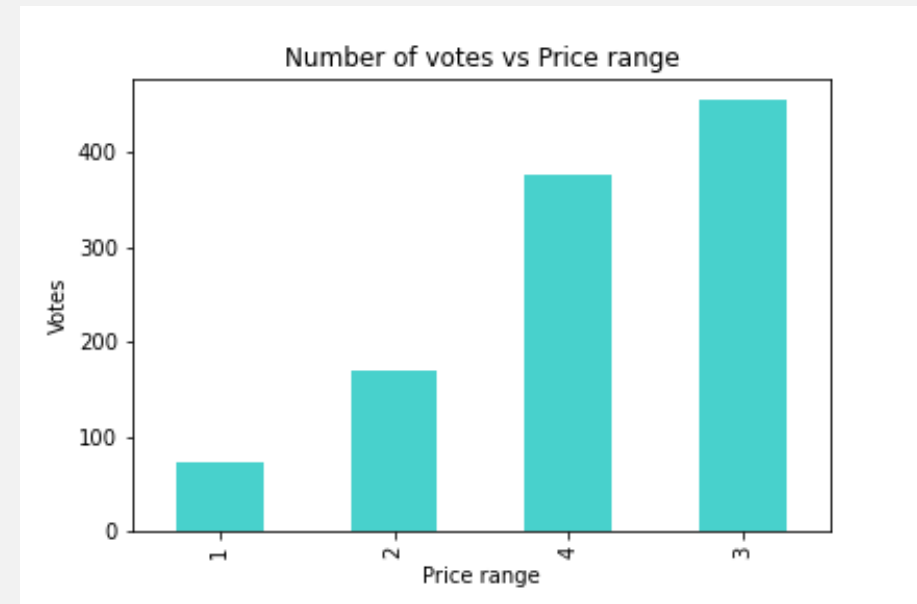
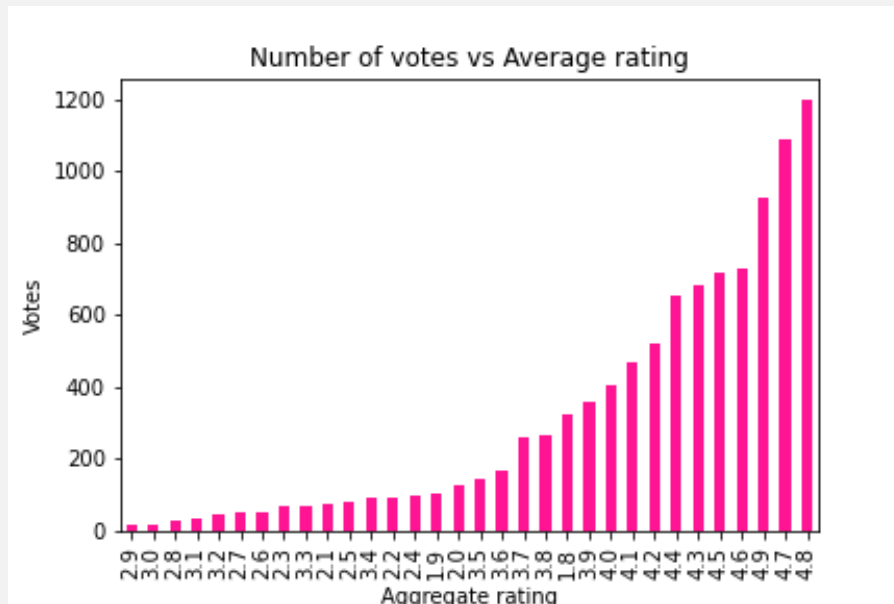
## Plot Continuous Features



We notice that the more the price goes up, the more the rating increases.

# Exploratory Data Analysis

## Plot Continuous Features



We notice that the best rated restaurants are the ones with the most votes.

# Exploratory Data Analysis

## Explore Categorical Features

In this section, we focused on categorical values.

- We first converted all the currencies into US Dollars and created a new column ***AVG\_US\_Dollars***
- We then checked the coherence between the columns *Rating text* and *Rating color*
- We noticed that in the column *Cuisines*, there were different types of cuisine but in the same string. We decided to split this strings into substrings and put them in new columns ***Type*** with one type of cuisine in each *Type*.

# Exploratory Data Analysis

## Plot Categorical Features

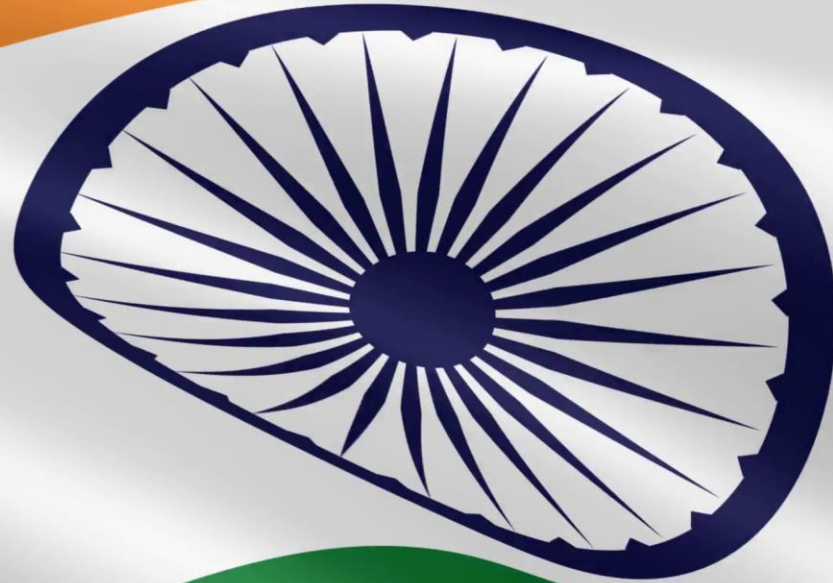
We plotted categorical values and made some conclusions. We discovered some fun conclusions, as you can check in the code.

# Exploratory Data Analysis

At this point in the study, we decided to focus on restaurants located in India because it contained the most rows and it would be more adequate to perform model predictions.

Our new dataset *df\_india* now contains **6504** rows and **27** columns.  
We also decided to transform all of the categorical values into quantitative values in another dataset called *df\_num*.

# RESTAURANTS IN INDIA



# Model Predictions

Now that we know better the dataset we are working on, it is time to make some predictions.

The first part of making model predictions is to decide what we are going to predict and what features we are going to consider.

The second step is to split our dataset in two parts :

- The training set : we train our model on this dataset
- The testing set : we test our model and make some predictions on this dataset

Finally, the most important part is to calculate the accuracy of our model to evaluate its performance. When we obtain the best model possible, we can make predictions with new data.

The goal of this study is to test different models by changing the features and the parameters and select the best one to make our final predictions.

# Model Predictions

Here are the fields of the dataset we are using for our model predictions:

- **City** (int64) - City where the restaurant is located
- **Average Cost for two** (int64) - Average cost of a two-persons meal
- **Has Table booking** (int64) - Possibility of booking a table or not
- **Has Online delivery** (int64) - Possibility of online delivery or not
- **Price range** (int64) - Price range of the restaurant
- **Type 0** (int64) -
- **Type 1** (float64) -
- **Type 2** (float64) -
- **Type 3** (float64) -
- **Type 4** (float64) -
- **Type 5** (float64) -
- **Type 6** (float64) -
- **Type 7** (float64) -

The dataset contains only information about the restaurants located in India.

We decide to make predictions on **Average Cost** for two using the following features: **City**, **Has Table booking**, **Has Online delivery**, **Price range**, **Type 0**, **Type 0**, **Type 0**, **Type 0**, **Type 0**, **Type 0** and **Type 0**.

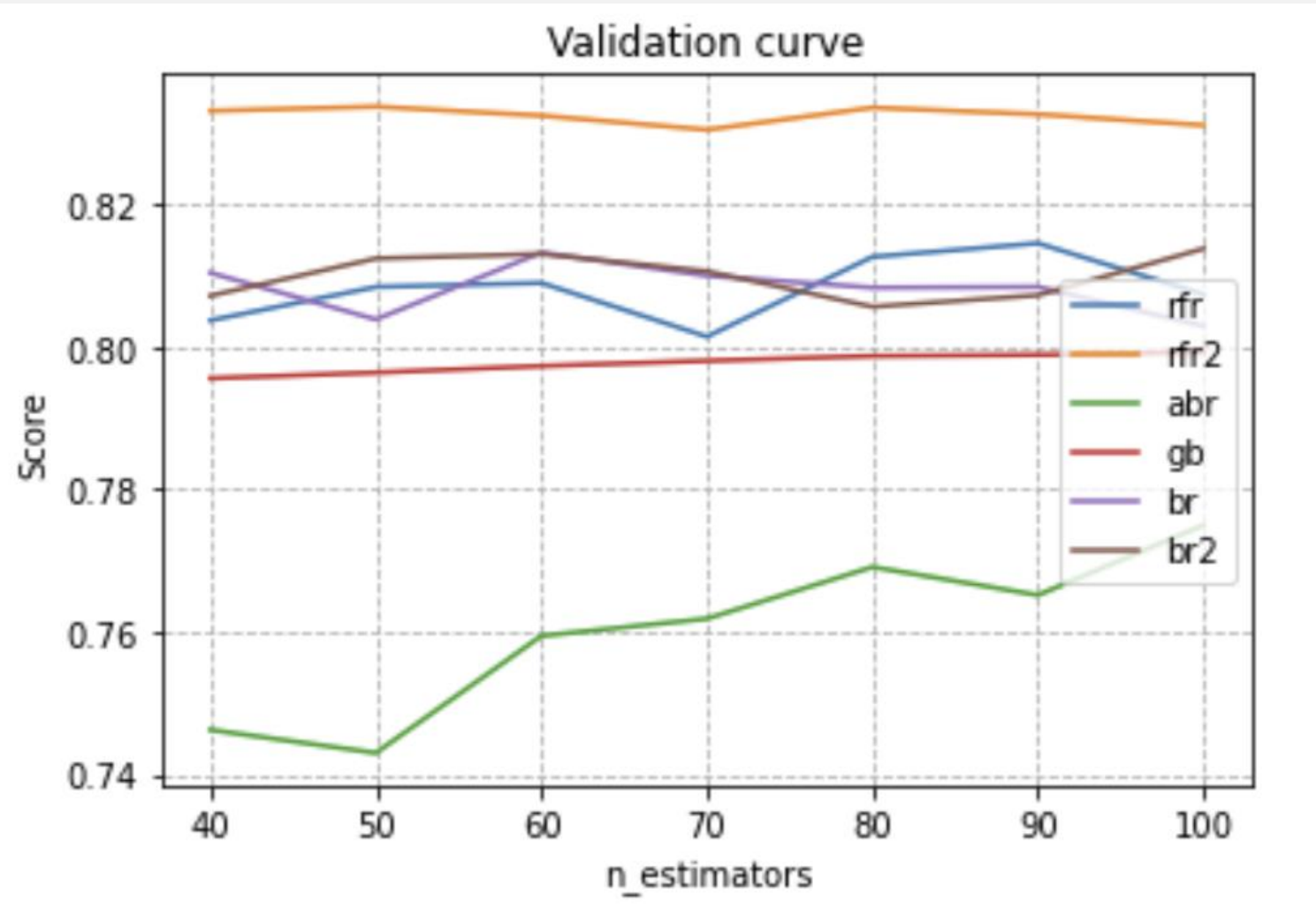


# Model Predictions

- For our first model, we performed a **linear regression**. We obtained a score of **0.71**
- For our second model, we used a **Bagging** model. We obtained a score of **0.82**
- For our third model, we used a **Random Forest** model. We obtained a score of **0.823**
- For our fourth model, we used a **AdaBoosting** model. We obtained a score of **0.80**
- For our fifth model, we used a **GradientBoosting** model. We obtained a score of **0.81**

We then performed a cross validation on our two best models. We changed the parameters and used a RandomizedSearch grid to get the best parameters for each model. The models that we obtained had indeed a better score. For our final model, we used the one with the best performance which is the Random Forest after cross validation.

# Model Predictions



# Flask API

Now that we have our model, it is time to display it in a pretty way !

We used a Flask app and created some pages using both HTML and CSS. We created a Form where you can select the parameters to get the average cost for two of a restaurant in India.

Time for you to test it !

Launch the app in your shell and let us guide you !

# Flask API

Estimate the price of your restaurants based on your preferences !

Has Online Delivery ☒ Yes ☐ No

Has Booking Table ☒ Yes ☐ No

City

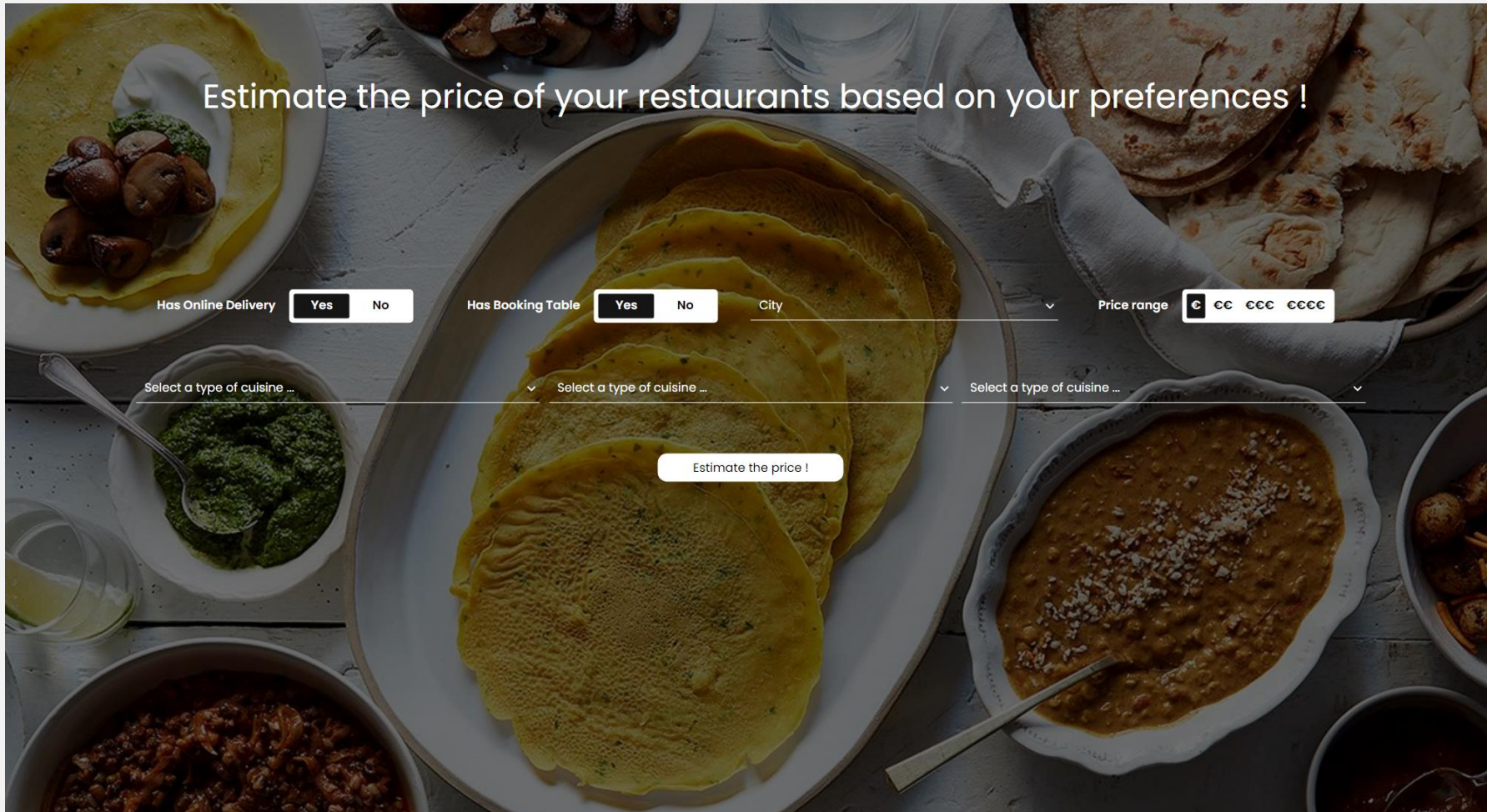
Price range ☐ € ☐ €€ ☐ €€€ ☐ €€€€

Select a type of cuisine ...

Select a type of cuisine ...

Select a type of cuisine ...

Estimate the price !



# Conclusion

As you can see, we encountered a problem to display the final prediction on the page, because of a problem of *Radio Button* in the HTML file.  
To avoid this problem, we created a *HowMuch* function to let you use our API and see how it works.

We had a lot of fun with this project !

**THANK YOU FOR  
YOUR INTEREST**

Laury FYOT-TAMADON

Sohana HASSANALY