# FOODIE-TECTOR

GROUP 8

# CONTENTS


**03**
INTRODUCTION

**04**
THE DATA

**05**
THE MODEL

**06**
THE CODE

**07**
MARKETING STRATEGY

**08**
IMPORTANT QUESTIONS

**09**
THE TEAM

# INTRODUCTION

FOOD-TECTOR , like the name suggests, is an application created to simply make a customer's life more convenient. The application is focused on detecting restraints that match  the user's needs.

FOODIE-TECTOR is a simple easy to use text analyzer that takes customer reviews to recommend a restaurant based on your ideal description.

## WHY FOODIE-TECTOR?

There is a fundamental gap in the market between user wants and the information available online. FOODIE-TECTOR bridges that gap by connecting user's and their wants to restaurants that provide them.

*"Food items are experienced goods and not searchable goods."*

# THE DATA

We collected and scraped the data we need for this project from Yelp.
The dataset is mainly a collection of customers' reviews of all 78 restaurants across the university village, which is the area between W. Harrison Street, W. 16 th Street, S. Halsted Street, S. Ashland Avenue.


The following variables are included:

- **Restaurant Name**
- **Restaurant Address**
- **Restaurant Type:** The category of restaurants. For example, sandwiches, Italian, fast food,etc. 56 unique categories are included.
- **Restaurant Rating:** The overall rating of a restaurant. The restaurant rating can range from0.0 to 5.0 and the interval is 0.5. The maximum and minimum restaurant rating in this dataset are 4.5 and 1.0.
- **Restaurant Reviews:** Customers' reviews of the restaurant. 14,369 unique reviews are included.
- **Review Date:** The date that a review was written.
- **Review Rating:** The rating for a restaurant rated by the review writer. The review rating can range from 0.0 to 5.0 and the interval is 1.0. The maximum and minimum review rating in this dataset are 5.0 and 1.0.

# THE MODEL

As demonstrated during our presentation, when the user inputs a statement the model provides a restaurant result that matches the users request. This is achieved through "Random Forest"

Packages used to execute random forests:

- import numpy as np
- import matplotlib.pyplot as plt
- import scipy
- import sklearn
- import seaborn as sns
- import re
- from collections import Counter
- %matplotlib inline

- The entire process can be carried out in a series of simple systematic steps. First we import our newly extracted data set from Yelp. A preview of this data set gives us some ideas what information is available for further analysis. We will also take a look at the missing data.
- About the Restuarant: the name, physical address(address + postal code + city + province/state), Restaurant Type & restaurant Rating
- About the review: the content (review.text), review date, review rating

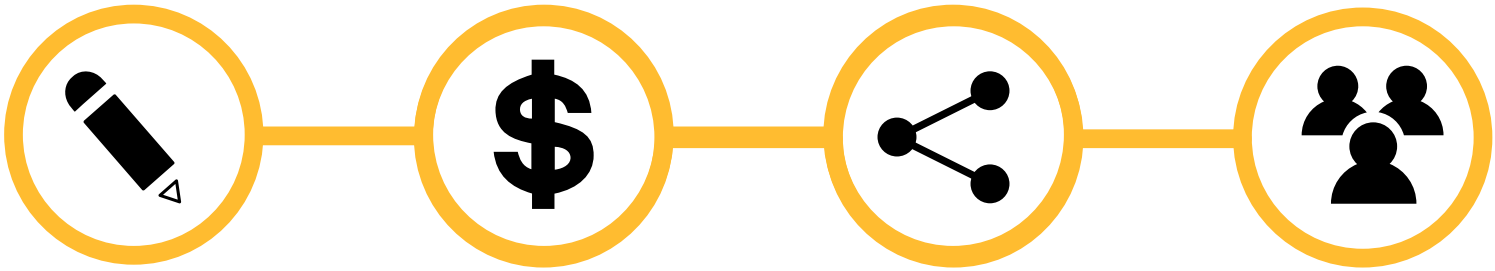
# BREAKING DOWN THE MODEL'S CODE

```
# Import the Data Set.
data = pd.read_csv('yelp data.csv')

# Make everything lower case.
data['review'] = data['review'].str.lower()

# Remove non-text characters.
data['review'] = data['review'].str.replace(r'\.|\!|\?|\'|,|-|\(|\)', "",)

# Fill in black reviews with '' rather than Null (which would give us errors).
data['review'] = data['review'].fillna('') <--

# Import and initiate a vectorizer.
from sklearn.feature_extraction.text import CountVectorizer

# The max features is how many words we want to allow us to create columns for.
vectorizer = CountVectorizer(max_features=5000)

# Vectorize our reviews to transform sentences into volumns.
X = vectorizer.fit_transform(data['review'])

# And then put all of that in a table.
bag_of_words = pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names())

# X is our words.
X = bag_of_words

# Y is our restaurant name (the outcome we care about).
Y_restaurant = data['restaurant name']

# Import a random forest model.
fromsklearn.ensemble importRandomForestClassifier
rfc = RandomForestClassifier()

# Fit that random forest model to our data.
rfc.fit(X,Y_restaurant)

# Write your own desired restaurant description here. -->
 test_review = ['''I would like to try some mexican food tonight!''']

# Convert your test review into a vector.
XX_test = vectorizer.transform(test_review).toarray()

# Match your review.
prediction = rfc.predict(X_test)[0]
```

**At this point we have successfully built our model. The next steps are executed when the user inputs his desired description of the restaurant.**

# FOODIE-TECTOR MARKETING STRATEGY

The main strategy for Foodie-Sector is use influencer marketing and collaborate with brands to spread the word about the application. The product will have a freemium model, as in basic features would be free and users will have to pay for premium features. Advertising will be based on a digital IMC campaign. Our Ad campaign for sales promotions: "Bye Bye Hunger, Get the Foodie-tector!"



**STRATEGY**
Influencer marketing, Brand integrations

**PRICING**
Freemium Model

**ADVERTISING**
Digital IMC campaign

**SALES**
Ad campaign

## ALTERNATE STRATEGIES

Built partnerships with APP such as WhatsApp, WeChat, and Line: By doing so, the user can get access to the recommending function even with their cellphone. For example, simply type in your dream review and send it to a certain phone number on WhatsApp, you can get the recommendation just in a few seconds.

Introduce the recommending function to websites such as Yelp: By simply adding a small message box or interactive box at the corner of the screen (webpage), the interactive feature of “typing in your dream review and get a recommendation of restaurant as an output” can be easier for the user to get access to and take advantage of it.

# SOME IMPORTANTS QUESTIONS

**why is your data limited?**

We scraped the data from Yelp, which takes time for the code to execute even after down scoping the area to the University Village. Additionally, Yelp detects and bans scraping, which makes the execution of the code even much more time-consuming as we have to add “time.sleep(2)” into loops or change our IP to prevent being blocked by Yelp. We also wanted to ensure high processing speeds for our users

**Why is data limited to University Village?**

The main goal and purpose for this project are to provide a handy tool for UIC students to find the best dining place near campus. Think of a situation that a user near UIC is finding a restaurant and the APP gives her a restaurant that is 10 miles aways from the campus. There is little chance that the user will follow the recommendation given by the APP.

# Talish Barmare

*UIN:662161711*

# Yi-Chen Chen

*UIN:665523926*

# Kavya Ravi Gowda

*UIN:670911760*


# GROUP EIGHT

MEET THE BRAINS BEHIND THE PROJECT