PROJECT TITLE:

WORLDWIDE FOODPANDA RESTAURANTS

LECTURE'S NAME:

ASSOC. PROF. DR. ROSLINAZAIRIMAH ZAKARIA

GROUP MEMBERS:

ID NO	NAME
SD20045	CHONG WEI HAN **(LEADER)**
SD21015	NURUL ATHIRAH BINTI RAMLI
SD21029	BANU SHREE A/P SHANMUGAM
SD21030	QISTINA ZAWANI BINTI RIDZUAN

GROUP PROJECT

BSD2223 Data Science Programming II

Semester II, 2022/2023

CONTENTS

1.0 INTRODUCTION

The COVID-19 pandemic has acted as a catalyst, propelling the rapid growth of online food delivery. With lockdown measures and social distancing regulations implemented worldwide, dining in restaurants became challenging, prompting people to prioritize their safety by turning to online food ordering and delivery services. In Malaysia, there was a staggering 90% increase in online sales between February and March, highlighting the growing preference for online food orders and grocery purchases as a means to maintain social distancing and minimize the risk of infection. This surge in demand has led to the widespread popularity of online food ordering platforms and applications. Moreover, it has created employment opportunities for delivery riders and provided restaurants with additional revenue streams. As a result, OFD services have witnessed unprecedented growth and have become the new normal for a larger segment of the population. In this highly competitive market, understanding and catering to consumer needs is crucial for OFD service providers to remain relevant and avoid being overshadowed.

Therefore, when it comes to the online food delivery marketplace, FoodPanda Group has emerged as a dominant player, particularly in emerging markets. The platform offers restaurants cutting-edge software and technology to enhance their online visibility and attract more customers to their offerings. On the consumer side, FoodPanda ensures a seamless experience, providing easy access to a wide range of culinary options. With just a few taps on their app or website, customers can indulge in their favourite meals from numerous restaurants. Founded in 2012, FoodPanda has experienced remarkable growth and expansion, operating in 22 countries across 3 continents and serving customers in 450 cities. With its headquarters based in Berlin, Germany, the company boasts a dedicated global team of over 2500 professionals. As a key player in the food delivery industry, FoodPanda serves as a vital link between customers and an extensive network of restaurants. In this increasingly competitive landscape, gaining insights into restaurant performance and ensuring customer satisfaction is pivotal.

This research project aims to conduct a comprehensive analysis of restaurants affiliated with FoodPanda. These objectives are focusing on:

1. Evaluate restaurant performance by assessing the overall ratings of restaurants in each country based on the types of cuisine followed by ratings.
2. Analyse customer preference by examining customer choices on vertical parent and types of vertical.
3. Identify the taste of each customer in a different country where restaurants can enhance their offerings in main cuisines.
4. Analyse the minimum delivery time and minimum pickup time for chosen delivery types of vertical based on the list of restaurants in each city.
5. Presenting analysis rating and reviews by country.

2.0 PROJECT DESCRIPTION

This project is motivated by an existing case study titled "Case Study on FoodPanda," available on Academia.edu. The case study highlights the need to analyze and optimize the performance of restaurants on the FoodPanda platform, a leading food delivery service. The chosen project aims to replicate and expand upon the case study's findings by conducting a comprehensive analysis of restaurant performance on the FoodPanda platform. This analysis utilizes real-world data to understand the factors influencing restaurant success and provide actionable insights for platform optimization. The project's importance lies in its potential to improve customer satisfaction and enhance operational efficiency. By examining the performance of restaurants on the FoodPanda platform, this project can identify areas for improvement, such as reducing delivery times, enhancing order accuracy, and increasing overall customer ratings.

This project is chosen due to its potential to analyze and optimize the performance of ASEAN Restaurant on FoodPanda, which is a prominent food delivery platform. The availability of a comprehensive dataset allows for a thorough examination of order patterns, delivery performance, and customer satisfaction, presenting an opportunity to identify areas that can be enhanced. The insights and recommendations resulting from this analysis can have a direct impact on the restaurant's operations, improving the overall customer experience and financial viability. Moreover, the project has the potential to provide valuable industry insights, contributing to a deeper understanding of the dynamics within the food delivery sector.

This project is incredibly important as it focuses on optimizing the performance of ASEAN Restaurant on FoodPanda, one of the leading food delivery platforms. By delving into the restaurant's order patterns, delivery performance, and customer satisfaction, we aim to improve their overall efficiency, enhance the customer experience, and ultimately boost their profitability. What makes this project truly innovative is the utilization of data-driven insights and recommendations to make informed decisions. This approach allows us to improve service quality, stay competitive in the ever-changing food delivery industry, and ensure ASEAN Restaurant's success. Additionally, the knowledge gained from this project can contribute to a deeper understanding of best practices and emerging trends within the food delivery ecosystem, benefiting not only ASEAN Restaurant but also the industry as a whole.

3.0 DATA DESCRIPTION

Attribute	Data Type	Description	Types data (Qualitative/ Quantitative)
budget	Integer	Unknown	Quantitative
is_new_until	DateTime	Date and time when the entity or business was started.	Qualitative
latitude	Float	Latitude coordinate of the entity or business location	Quantitative
longitude	Float	Longitude coordinate of the entity or business location	Quantitative
minimum_delivery_time	Integer	Minimum number of required for delivery of orders	Quantitative
minimum_order_time	Integer	Minimum number of required for orders	Quantitative
minimum_pickup_time	Integer	Minimum number of required for order pickup	Quantitative
name	String	Name of the entity or business	Qualitative
rating	Float	Average rating given to the entity or business	Qualitative
review_number	Integer	Total number of reviews received	Quantitative
review_with_comment_number	Integer	Total number of reviews with comments	Quantitative
vertical	String	Subcategory or industry vertical of the entity or business	Qualitative
vertical_parent	String	Category or industry vertical of the entity or business	Qualitative
delivery_provider	String	Delivery service provider associated with the entity	Qualitative
is_active	Boolean	Flag indicating if the entity or business is active	Qualitative
is_new	Boolean	Flag indicating if the entity or business is new	Qualitative
is_promoted	Boolean	Flag indicating if the entity or business is promoted	Qualitative
city	String	City in which the entity or business is located	Qualitative
timezone	String	Time zone associated with the entity or business	Qualitative
dine_in	Boolean	Flag indicating if the entity or business offers dine-in	Qualitative
main_cuisine	String	Main cuisine or food category offered by the entity	Qualitative
country	String	Country in which the entity or business is located	Qualitative

4.0 DATA PREPARATION

In this section, there are a few steps to clean the raw data. The process including delete some columns, drop missing values, filtering some country and son. Below are the step-by-step process for data preparation.

```r
# import data

#file.choose()    # to find file location
rdata <- read.csv("D:\\UMP\\Sem 4 2022-23\\BSD2223 Data Sc Programming II\\Project\\Dataset\\FoodPanda Restaurants.csv", header=TRUE)
rdata
```

Description: df [267,378 × 23]

budget <int>	is_new_until <chr>	latitude <dbl>	longitude <dbl>
2	2023-01-08T07:58:20Z	11.57846	104.8940
3	2023-01-08T07:58:40Z	11.58970	104.8750
3	2023-01-08T08:02:26Z	11.54416	104.9160
3	2023-01-08T07:58:23Z	11.53443	104.8860
3	2023-01-08T07:59:17Z	11.55282	104.8838
3	2023-01-08T07:58:35Z	11.55018	104.9003
1	2021-07-21T00:00:00Z	11.56657	104.9301
3	2023-01-08T08:02:44Z	11.56946	104.9183
1	2021-09-18T00:00:00Z	11.54210	104.9152
1	2022-11-25T00:00:00Z	11.54978	104.9004

1-10 of 267,378 rows | 1-4 of 23 columns Previous 1 2 3 4 5 6 ... 100 Next

Figure 4.1 import dataset

First and foremost, we import the dataset for FoodPanda using read.csv() and rename the data as “rdata”. It shows the raw data in figure 4.1.

```r
# v1: delete column budget & post_code because:
# budget & post_code = MEANINGLESS to data analysis
# minimum_order_amount & review_with_comment_number are SAME VALUE
# is_active & is_promoted ONLY have OPTION option

v1 <- subset(rdata, select = -c(budget, minimum_order_amount, post_code,
review_with_comment_number, is_active, is_promoted))
v1
```

Description: df [267,378 × 17]

	is_new_until <chr>	latitude <dbl>	longitude <dbl>	minimum_delivery_time <int>
1	2023-01-08T07:58:20Z	11.57846	104.8940	0
2	2023-01-08T07:58:40Z	11.58970	104.8750	0
3	2023-01-08T08:02:26Z	11.54416	104.9160	29810
4	2023-01-08T07:58:23Z	11.53443	104.8860	29800
5	2023-01-08T07:59:17Z	11.55282	104.8838	29800
6	2023-01-08T07:58:35Z	11.55018	104.9003	29805
7	2021-07-21T00:00:00Z	11.56657	104.9301	29815
8	2023-01-08T08:02:44Z	11.56946	104.9183	29810
9	2021-09-18T00:00:00Z	11.54210	104.9152	29810
10	2022-11-25T00:00:00Z	11.54978	104.9004	0

1-10 of 267,378 rows | 1-5 of 17 columns Previous 1 2 3 4 5 6 ... 100 Next

Figure 4.2 delete some columns

In figure 4.2, we delete some columns. we delete the columns for “budget” and “post_code” because they are meaningless to the dataset. While the whole columns for “minimum_order_amount” and “review_with_comment_number” are same values. The columns for “is_active” and “is_promoted” show one option only in Boolean format which is TRUE and FALSE respectively.

```r
# v2: change date format

#install.packages("lubridate")
library(lubridate)
library(dplyr)

# Define the input date string
date_string <- v1$is_new_until

# Parse the date string into a POSIXct object
date <- ymd_hms(date_string)

# Format the date as "dd-mm-yyyy hh:mm:ss"
v1$is_new_until <- format(date, format = "%d-%m-%Y")

# Print the formatted date
v2 <- v1
v2
```

Description: df [267,378 × 17]

	is_new_until <chr>	latitude <dbl>	longitude <dbl>	minimum_delivery_time <int>
1	08-01-2023	11.57846	104.8940	0
2	08-01-2023	11.58970	104.8750	0
3	08-01-2023	11.54416	104.9160	29810
4	08-01-2023	11.53443	104.8860	29800
5	08-01-2023	11.55282	104.8838	29800
6	08-01-2023	11.55018	104.9003	29805
7	21-07-2021	11.56657	104.9301	29815
8	08-01-2023	11.56946	104.9183	29810
9	18-09-2021	11.54210	104.9152	29810
10	25-11-2022	11.54978	104.9004	0

Figure 4.3 change date format

The date “is_new_until” shows the date and time format. Since the time will not benefit to the analysis so we decided to filter the time and use the date only in figure 4.3. We need to install the library “lubridate” if we need to select the date only.

```r
# v3: sort by date

# Convert the 'is_new_until' column to a date format
v2$is_new_until <- as.Date(v2$is_new_until, format = "%d-%m-%Y")

# Sort the data by date
sorted_data <- v2 %>% arrange(is_new_until)

# View the sorted data
v3 <- sorted_data
v3
```

Description: df [267,378 × 17]

is_new_until <date>	latitude <dbl>	longitude <dbl>	minimum_delivery_time <int>
0001-01-01	11.555201	104.92402	29815
0001-01-01	24.868921	67.05936	50
0001-01-01	24.798868	67.04406	17015
0001-01-01	24.807753	67.03744	17015
0001-01-01	24.882514	67.05875	17025
0001-01-01	24.863486	67.05474	17020
0001-01-01	24.901055	67.11571	17035
0001-01-01	24.793147	67.06271	17020
0001-01-01	24.882122	67.06304	17020
0001-01-01	24.914044	67.10130	60

1-10 of 267,378 rows | 1-4 of 17 columns Previous 1 2 3 4 5 6 ... 100 Next

Figure 4.4 sort the date

We sort the date by ascending to check the earliest date is when in figure 4.4.

```r
# v4: filter certain date (01/01/2013 - 31/5/2023)
# because the date start from 0001-01-01

# Convert the 'is_new_until' column to a date format
v3$is_new_until <- as.Date(v3$is_new_until)


# Filter the data based on the date range
v4 <- v3 %>% filter(is_new_until >= as.Date("2013-01-01") & is_new_until <= as.Date("2023-05-31"))
v4
```

Description: df [264,910 × 17]

is_new_until <date>	latitude <dbl>	longitude <dbl>	minimum_delivery_time <int>
2013-01-03	47.528890	19.07938	7315
2013-01-17	46.077385	18.22837	7130
2013-03-05	47.532995	19.04076	7315
2013-03-06	47.572689	19.05539	7360
2013-03-20	46.064501	18.18965	60
2013-04-23	47.503737	19.05751	7315
2013-05-27	47.572689	19.05539	45
2013-05-28	47.529249	19.06011	7350
2013-07-03	24.990416	121.50093	23245
2013-07-15	47.502131	19.07049	15

1-10 of 264,910 rows | 1-4 of 17 columns Previous 1 2 3 4 5 6 ... 100 Next

Figure 4.5 filter date

After sorting the date, we found that the date starts from “0001-01-01”. So, we decided to filter the date and choose between “2013-01-01” and “2023-05-31” in figure 4.5.

```r
# v5: filter the ASEAN country only (Cambodia, Laos, Malaysia, Myanmar, Philippines, Thailand, Singapore) because aim to focus on ASEAN country only

library(dplyr)

v5 <- filter(v4, country %in% c("Cambodia", "Laos", "Malaysia", "Myanmar", "Philippines", "Thailand", "Singapore"))
v5
```

Description: df [177,677 × 17]

is_new_until <date>	latitude <dbl>	longitude <dbl>	minimum_delivery_time <int>
2017-09-02	3.116778	101.67670	0
2017-09-03	13.706919	100.60477	60
2017-09-03	13.722605	100.57417	14380
2017-09-03	14.642841	121.05910	0
2017-09-04	12.911923	100.87059	14430
2017-09-04	14.555004	121.04935	26780
2017-09-04	14.587476	121.04475	0
2017-09-06	1.309029	103.85843	4930
2017-09-06	1.391374	103.89399	4930
2017-09-06	13.730762	100.58521	14380

1-10 of 177,677 rows | 1-4 of 17 columns Previous 1 2 3 4 5 6 ... 100 Next

Figure 4.6 filter ASEAN Country

In figure 4.6, we decided to filter ASEAN country because our project is only for ASEAN countries only which are "Cambodia", "Laos", "Malaysia", "Myanmar", "Philippines", "Thailand", and "Singapore".

```r
# v6: delete the missing value in the "main_cuisine" column because those values can't simply replace with other terms

missing_rows <- is.na(v5$main_cuisine) | v5$main_cuisine == ""

v6 <- v5[!missing_rows, ]
v6
```

Description: df [167,644 × 17]

	is_new_until <date>	latitude <dbl>	longitude <dbl>	minimum_delivery_time <int>
1	2017-09-02	3.116778	101.67670	0
2	2017-09-03	13.706919	100.60477	60
3	2017-09-03	13.722605	100.57417	14380
4	2017-09-03	14.642841	121.05910	0
5	2017-09-04	12.911923	100.87059	14430
6	2017-09-04	14.555004	121.04935	26780
7	2017-09-04	14.587476	121.04475	0
8	2017-09-06	1.309029	103.85843	4930
9	2017-09-06	1.391374	103.89399	4930
10	2017-09-06	13.730762	100.58521	14380

1-10 of 167,644 rows | 1-5 of 17 columns Previous 1 2 3 4 5 6 ... 100 Next

Figure 4.7 delete missing value

We delete the missing value in “main_cuisine” column because the value in that column can replace with other terms.

```r
# v7: filter the minimum_delivery_time <= 1000 because non logic to the analysis

v7 <- v6[v6$minimum_delivery_time <= 1000, ]
v7
```

Description: df [111,152 × 17]

	is_new_until <date>	latitude <dbl>	longitude <dbl>	minimum_delivery_time <int>
1	2017-09-02	3.116778	101.6767	0
2	2017-09-03	13.706919	100.6048	60
4	2017-09-03	14.642841	121.0591	0
7	2017-09-04	14.587476	121.0447	0
12	2017-09-09	3.079577	101.6862	0
14	2017-09-10	3.072610	101.6730	0
18	2017-09-13	3.082310	101.5270	0
19	2017-09-13	1.354506	103.8773	30
23	2017-09-14	1.350720	103.8489	30
24	2017-09-15	3.166484	101.6966	0

1-10 of 111,152 rows | 1-5 of 17 columns Previous 1 2 3 4 5 6 ... 100 Next

Figure 4.8 filter min_delivery_time

We found that the value for “min_delivery_time” more than thousand, it will impact our analysis. So, we decided to select the “min_delivery_time” less than and equal to 1000 only.

```r
# save the file as csv

write.csv(v7, "D:\\UMP\\Sem 4 2022-23\\BSD2223 Data Sc Programming II\\Project\\Dataset\\data.csv", row.names=FALSE)
```

Figure 4.9 Save cleaned dataset

After cleaned the dataset, we save the file as “data.csv” by using “write.csv”.

5.0 DATA ANALYSIS, RESULTS AND DISCUSSION

5.1 Sidebar Dashboard

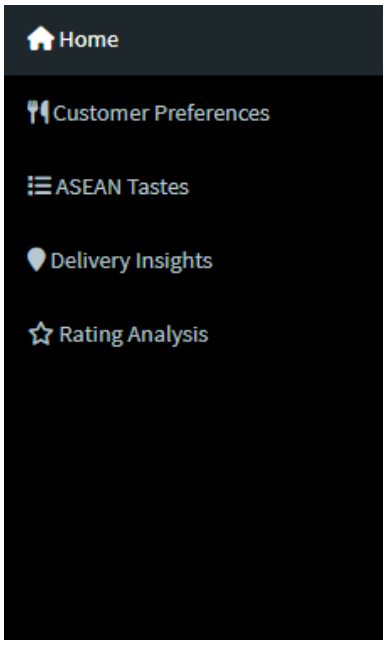

Figure 5.1.1 Sidebar Dashboard

Based on Figure 5.1.1, the dashboard sidebar was created on the left side that performs five dashboards which are Home, Customer Preferences, ASEAN Tastes, Delivery Insights, and Rating Analysis.

5.2 Home Dashboard

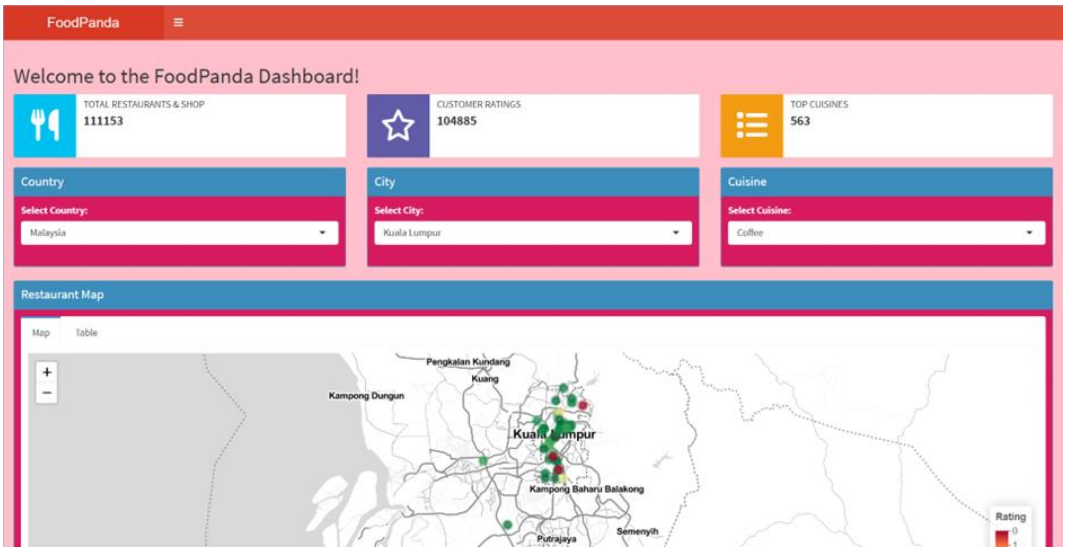

Figure 5.2.1 Home Dashboard

Based on the figure 5.2.1, the data visualization map shows the distribution of restaurants across ASEAN countries, with the drop-down representing each country. By analyzing the map, it is apparent that the highest ratings and lowest ratings of restaurants are found in urban areas, such as Bangkok, Singapore, Quezon City, Vientiane Capital, Yangon, Phnom Penh and Kuala Lumpur. The main cuisine that we will be focusing on is Vegetarian.

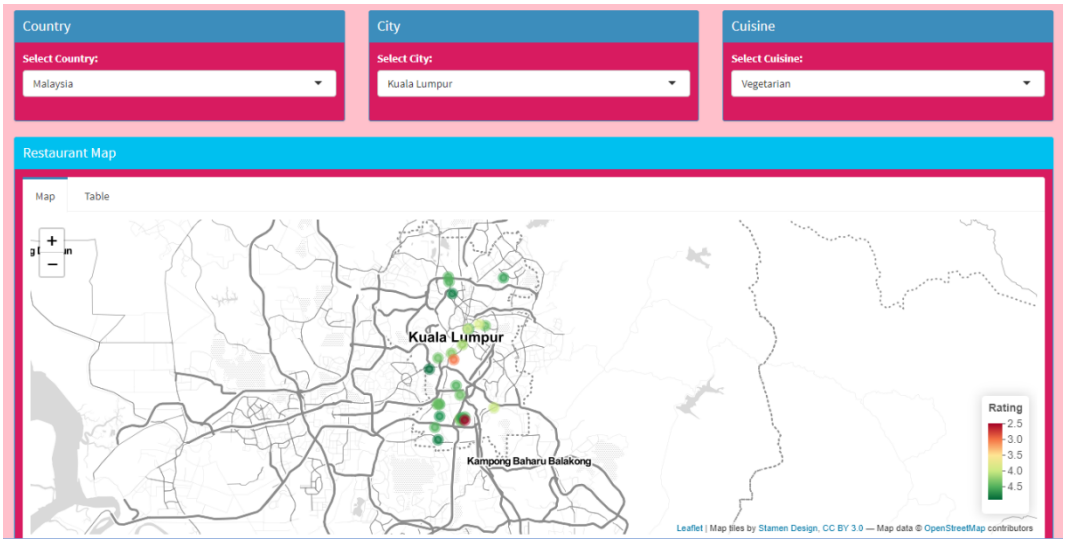

Figure 5.2.2 Kuala Lumpur Restaurant Vegetarian

One of the key features of the dashboard is the ability to filter the data based on specific criteria. Users can choose the country, city, and cuisine they are interested in, enabling them to focus on a particular subset of restaurants. This filtering functionality allows users to customize their analysis and obtain insights that are relevant to their specific needs and preferences. The project also features a user-friendly graphical interface that allows users to input text and view sentiment ratings. These ratings are represented by three colours: red for ratings between 2.5 and less than 3.5, yellow for ratings between 3.5 and less than 4.0, and green for ratings between 4.0 and 5.0. This visual representation helps users understand the sentiment behind the analyzed text.

The main visualization in the dashboard is the map visualization, which displays the top-rated restaurants based on the selected filters. The leaflet map represents a restaurant, and its height represents the rating. By colour-coding based on the ratings, it becomes effortless to identify how highly rated restaurants are distributed across different

cities. This visual representation enables users to compare restaurant ratings and identify top performers easily.

Based on Figure 5.2.2, the first country in ASEAN that we choose is Kuala Lumpur, Malaysia. There is a restaurant that provides vegetarian food which got the lowest rating which is FOLOSO Vegetarian Cuisine. Apart from that, SRI PETALING VEGETARIAN HEALTH FOOD got the highest rating for vegetarian cuisine. By referring to the map, it shows that vegetarian restaurants are more prevalent in certain cities known for their emphasis on plant-based or vegetarian diets.

5.3 Customer Preferences

Figure 5.3.1 Customer Preferences Dashboard

The dashboard above shows the customer preferences dashboard with a vertical bar plot, filtered by vertical parent and country. It provides businesses with valuable insights into customer preferences across different categories and locations.

By selecting a specific vertical parent category and country, businesses can analyze customer preferences in depth. The vertical bar plot visualizes the data, allowing for easy comparison and identification of popular preferences within each category and country.

For example, after filtering by the country “Singapore”, the vertical bar plot reveals customer preferences in various categories, including “bakery”, “fruits and vegetables”, “pastry supplies”, and “snacks and sweets”. The analysis indicates that

"pastry supplies" have the highest level of preference among customers, while "bakery" and "snacks and sweets" have the lowest level of preference. This information can guide the shop in making strategic decisions, such as focusing more on party supplies and potentially adjusting their offerings in the bakery and snack and sweet categories to better align with customer preferences.

5.4 ASEAN Tastes

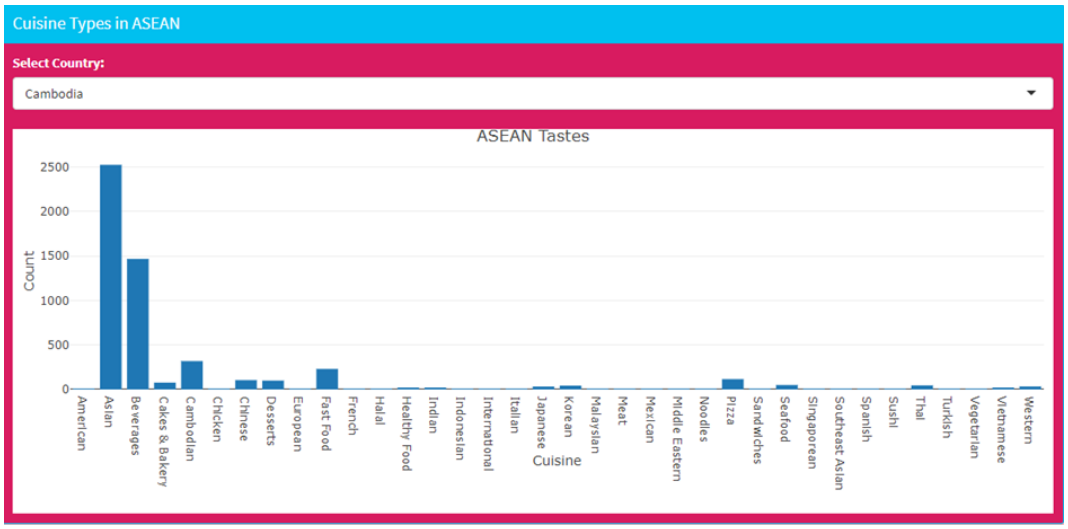

Figure 5.4.1 ASEAN Tastes

The ASEAN Taste dashboard is a powerful tool designed to explore the rich culinary traditions of ASEAN countries. With the ability to filter by country, the dashboard generates an informative bar plot that reveals the main cuisines associated with each selected country. The interactive nature of the bar plot enables users to make comparisons between the main cuisines across different ASEAN countries. This feature proves invaluable for individuals interested in exploring specific cuisines or planning culinary adventures within the region.

Figure 5.4.1 shows that Asian cuisine has the highest number of restaurants in Cambodia. Hence, we can conclude that Cambodia residents preferred "Asian" cuisine as their meal since there are many restaurants there. However, cuisines like "Halal", "Indonesian", "International", "Italian", "Malaysian", "Meat", "Mexican", "Middle Eastern", "Noodles", "Sandwiches", "Singaporean", "Southeast Asian", "Spanish", "Sushi", "Vegetarian" and "Vietnamese" are not really common there since there are not many restaurants and shops for that cuisines. After examining the cuisine types in

ASEAN countries, it becomes apparent that each country has its own unique culinary identity. The most frequently occurring cuisine types in ASEAN can vary based on cultural influences, local ingredients, historical background, and preferences of the local population.

By analyzing the bar plot within the ASEAN Taste dashboard, users gain deeper insights into the culinary preferences and cultural heritage of each country. The dashboard assists in making informed decisions about culinary exploration, offering a glimpse into the dominant flavours and cuisines one can expect to encounter in their chosen ASEAN destination.

5.5 Delivery Insights

Figure 5.5.1 Delivery Insights Dashboard

The delivery insight dashboard allows users to gain valuable insights into delivery options based on filters such as delivery option, country, city, and main cuisine. By

selecting these filters, the dashboard generates a vertical bar chart that showcases the minimum delivery time and minimum pick-up time for each combination of filters.

The vertical bar chart visualizes the data, with each bar representing a specific restaurant or combination of filters. The height of each bar represents the minimum delivery time or minimum pick-up time. This allows users to quickly compare the delivery and pick-up efficiency of different restaurants or filter combinations.

For example, after selecting “vendor delivery” as the delivery option, “Singapore” as the country, “Singapore " as the city and “pizza” as the main cuisine. The highest delivery time is “Overly Cheezy Pizzeria - Restaurant Delivery” and “FOUR PIZZA (Havelock Rd) - Restaurant Delivery” which is 45 times in the bar plot for minimum delivery time. However, “Overly Cheezy Pizzeria - Restaurant Delivery” and “FOUR PIZZA (Havelock Rd) - Restaurant Delivery” have the lowest pickup time which is 10 times in the bar plot minimum pickup time. Hence, we can conclude that the pizza lover more prefers using delivery service over pickup service. This is because it can save a lot of time without wasting time.

By analyzing the bar chart, users can identify restaurants or filter combinations that offer the shortest delivery and pick-up times. This information helps users make informed decisions when choosing a restaurant for their specific preferences and needs.

5.6 Rating Analysis Dashboard

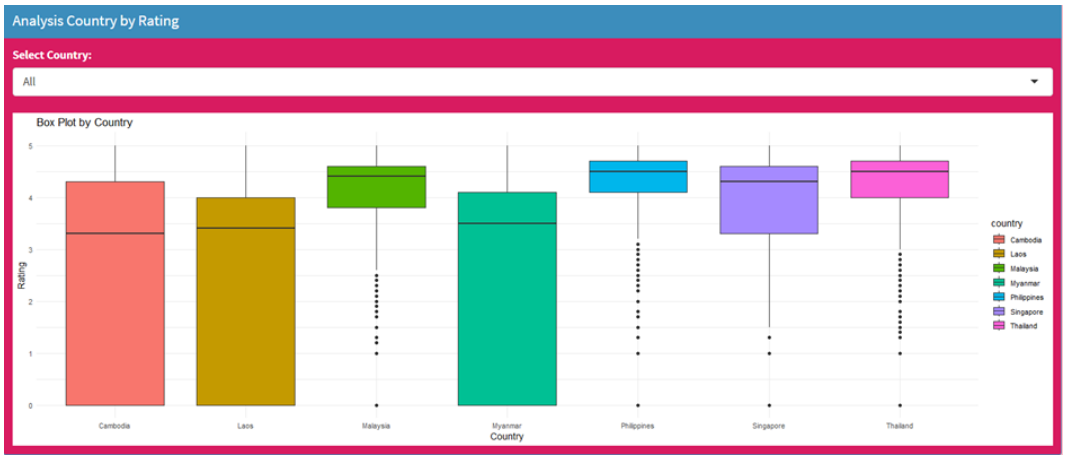

Figure 5.6.1 Dashboard for Rating Analysis

The dashboard for rating by country provides a visual representation of restaurant ratings using boxplots. It allows users to explore and compare the ratings of restaurants

across different countries in an intuitive and informative way. When using the dashboard, users can select a specific country of interest from a dropdown menu.

Upon selecting a country, the dashboard dynamically generates a boxplot that displays the distribution of ratings for that particular country. By examining the boxplots, users can compare the central tendency, spread, and potential outliers of the ratings. They can identify countries with higher or lower median ratings, variations in the spread of ratings, and any exceptional ratings that stand out from the majority.

Figure 5.6.1 shows the boxplot after selecting all countries. Among the countries examined, “Malaysia”, “Philippines”, “Singapore”, and “Thailand” exhibit a wider spread of restaurant ratings, as indicated by the presence of outliers in their respective boxplots. These outliers represent restaurants with exceptionally high or low ratings, suggesting a diverse range of dining experiences in these countries. Customers in these countries may encounter restaurants with outstanding ratings as well as establishments that receive lower ratings.

In contrast, “Cambodia”, “Laos”, and “Myanmar” display boxplots without any outliers, indicating a more consistent distribution of restaurant ratings. This implies that the dining experiences in these countries are relatively more uniform, with a narrower range of customer opinions. Restaurants in these countries may generally receive ratings within a specific range, indicating a consistent level of quality or a more homogeneous dining scene.

By examining the boxplots, users can compare the central tendency, spread, and potential outliers of the ratings. They can identify countries with higher or lower median ratings, variations in the spread of ratings, and any exceptional ratings that stand out from the majority.

6.0 CONCLUSION

FoodPanda's success can be attributed to its innovative use of technology. By leveraging mobile apps and online platforms, FoodPanda revolutionized the way people order food. This entrepreneurial approach to embracing technological advancements allowed the company to create a convenient and efficient solution for customers, while also providing a new avenue for restaurants to reach a wider audience.

The case study highlights the importance of market analysis in entrepreneurship. FoodPanda conducted thorough research to understand customer preferences, market gaps, and competitors. This analytical approach helped them tailor their services and provide value to both restaurants and customers.

The analysis reveals a significant presence of FoodPanda restaurants across various cities in ASEAN countries, reflecting the popularity and demand for food delivery services in the region. The distribution of restaurants is concentrated in urban areas, particularly in major cities such as Bangkok, Singapore, and Kuala Lumpur. This indicates that FoodPanda has successfully established a strong foothold in urban markets and has effectively capitalized on the growing trend of food delivery. Furthermore, the analysis highlights the diversity of cuisines available through FoodPanda, showcasing a wide range of options including local specialities and international cuisines. This demonstrates FoodPanda's ability to cater to the diverse culinary preferences within ASEAN countries. In terms of entrepreneurship, the significance of these findings lies in the potential opportunities it presents for aspiring entrepreneurs. The presence of FoodPanda restaurants in ASEAN cities indicates a thriving market for food delivery services. Entrepreneurs can leverage this information to identify untapped areas or niche markets, allowing them to establish their own restaurants or food delivery businesses.

Additionally, the analysis underscores the importance of adapting to consumer preferences. By observing the distribution of highly rated restaurants and identifying gaps in specific cuisines or locations, entrepreneurs can tailor their offerings to meet the demands of customers who prioritize quality and variety.

Overall, the findings emphasize the significance of the proposed solution for entrepreneurship. Entrepreneurs can leverage the success and popularity of FoodPanda in ASEAN countries to develop innovative business models, focusing on unique cuisines, exceptional customer experiences, or specialized services that differentiate them from existing food delivery platforms. By aligning their entrepreneurial endeavours with the

insights gained from the analysis, aspiring entrepreneurs can tap into the growing market of food delivery services and contribute to the culinary landscape of ASEAN countries, ultimately offering consumers more choices and enhancing the overall entrepreneurship ecosystem.

7.0 LIMITATIONS OF STUDY

One limitation of this study is the quality of the dataset. The accuracy and reliability of the findings can be affected if the data contains errors, missing values, or biases. It is important to ensure that the dataset used for analysis is of high quality and represents the population of interest accurately.

Another limitation is the limited set of variables considered in the analysis. This study only focuses on restaurant ratings and country. However, there are several other important factors that can influence restaurant performance. Variables such as cuisine type, pricing, customer reviews, and demographic characteristics of the target audience can provide valuable insights into understanding the factors that contribute to restaurant success or failure. Incorporating these variables into the analysis would provide a more comprehensive understanding of the factors influencing restaurant performance.

Considering these limitations, it is important to interpret the results of the analysis with caution and acknowledge the potential impact of data quality issues and the absence of other influential variables. Expanding the scope of the analysis by incorporating additional variables and ensuring data quality would enhance the depth and reliability of the findings.

REFERENCES

FoodPanda Restaurants. (n.d.). Www.kaggle.com. Retrieved June 20, 2023, from https://www.kaggle.com/datasets/hashiromer/all-FoodPanda-restaurants

End-to-End Predictive Analysis on Zomato. (2021, October 25). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/end-to-end-predictive-analysis-on-zomato/

Shiny Dashboard Structure. (n.d.). Rstudio.github.io. https://rstudio.github.io/shinydashboard/structure.html

APPENDIX

RShiny Dashboard (coding)

Refer to:

https://drive.google.com/file/d/1b-i23LqDGrcHt5MWQD2A5DQRaLaerdPk/view?usp=sharing

```
writeLines("
/* Custom styles for FoodPanda Dashboard */
.main-header {
  background-color: pink !important;
}

.main-sidebar {
  background-color: black !important;
}

.content-wrapper,
.right-side {
  background-color: pink !important;
}
", "custom.css")




library(shiny)
library(shinydashboard)
library(dplyr)
library(plotly)
library(leaflet)

# Load the dataset
rdata <- read.csv("D:\\UMP\\Sem 4 2022-23\\BSD2223 Data Sc Programming II\\Project\\Dataset\\data.csv")

# dashboard header
header <- dashboardHeader(
  title = span("FoodPanda Dashboard", style = "color: white;")
)

# dashboard sidebar
sidebar <- dashboardSidebar(
  sidebarMenu(
    menuItem("Home", tabName = "home", icon = icon("home")),
    menuItem("Customer Preferences", tabName = "customer_pref", icon = icon("utensils")),
    menuItem("ASEAN Tastes", tabName = "asean_tastes", icon = icon("list")),
    menuItem("Delivery Insights", tabName = "delivery_insights", icon = icon("map-marker")),
    menuItem("Rating Analysis", tabName = "boxplot", icon = icon("star"))
  )
)

# dashboard body
body <- dashboardBody(
  # Add this line to include the custom CSS file
  includeCSS("custom.css"),
  tabItems(
    ############################################################################################ HOME
    tabItem(
      tabName = "home",
      h2("Welcome to the FoodPanda Dashboard!"),
      fluidRow(
        # Info Boxes
        infoBox("Total restaurants & Shop", "111153", icon = icon("utensils")),
        # Dynamic Info Boxes
        infoBoxOutput("ratingBox"),
        infoBoxOutput("cuisineBox"),
        # Add a styled box around the select inputs
        box(
          title = "Country",
          width = 4,
          solidHeader = TRUE,
          status = "primary",
          background = "maroon",
          selectInput("countryInput", "Select Country:",
                      choices = unique(rdata$country),
                      selected = NULL),
        ),
        box(
          title = "City",
          width = 4,
          solidHeader = TRUE,
          status = "primary",
          background = "maroon",
          selectInput("cityInput", "Select City:",
                      choices = NULL,
                      selected = NULL),
        ),
        box(
          title = "Cuisine",
          width = 4,
          solidHeader = TRUE,
          status = "primary",
          background = "maroon",
          selectInput("cuisineInput", "Select Cuisine:",
                      choices = NULL,
                      selected = NULL),
        ),
      ),
      fluidRow(
        # Display the map
        box(
          title = "Restaurant Map",
          width = 12,
          solidHeader = TRUE,
          status = "primary",
          background = "maroon",
          tabBox(id = "List of restaurant",
                 width = "100%",
                 tabPanel("Map ",
                          leafletOutput("restaurant_map")),
                 tabPanel("Table",
                          DT::dataTableOutput("top_restaurants_table"))
          )
        )
      )
    ),
```

```
    ############################################################################################ CUST. PREFERENCES
    tabItem(
      tabName = "customer_pref",
      h2("Customer Preferences"),
      fluidRow(
        box(
          title = "Vertical Parent",
          solidHeader = TRUE,
          status = "primary",
          background = "maroon",
          selectInput("verticalparent5Input", "Select Vertical Parent:",
                      choices = unique(rdata$vertical_parent),
                      selected = NULL)
        ),
        box(
          title = "Country",
          #width = 6,
          solidHeader = TRUE,
          status = "primary",
          background = "maroon",
          selectInput("country5Input", "Select Country: ",
                      choices = unique(rdata$country),
                      selected = NULL)
        )
      ),
      fluidRow(
        box(
          title = "Vertical",
          width = 12,
          solidHeader = TRUE,
          status = "primary",
          background = "maroon",
          plotOutput("verticalparent_plot")
        )
      )
    ),
    ############################################################################################ ASEAN TASTES
    tabItem(
      tabName = "asean_tastes",
      h2("Cuisine Types For Asean"),
      fluidRow(
        box(
          title = "Cuisine Types in ASEAN",
          width = 12,
          solidHeader = TRUE,
          status = "info",
          background = "maroon",
          selectInput("countryInput_asean", "Select Country:",
                      choices = c("All", "Singapore", "Malaysia", "Thailand", "Laos", "Philippines", "Myanmar", "Cambodia"),
                      selected = "All"),
          plotlyOutput("asean_tastes_plot")
        )
      )
    ),
    ############################################################################################ DELIVERY INSIGHTS
    tabItem(
      tabName = "delivery_insights",
      h2("Delivery Insights"),
      fluidRow(
        box(title = "Delivery Option",
            status = "primary",
            solidHeader = TRUE,
            width = 3,
            background = "maroon",
            selectInput("deliveryprovider1Input", "Select Delivery Provider",
                        choices = unique(rdata$delivery_provider),
                        selected = NULL)
        ),
        box(title = "Country",
            status = "primary",
            solidHeader = TRUE,
            width = 3,
            background = "maroon",
            selectInput("country1Input", "Select Country",
                        choices = NULL,
                        selected = NULL),
        ),
        box(title = "City",
            status = "primary",
            solidHeader = TRUE,
            width = 3,
            background = "maroon",
            selectInput("city1Input", "Select City",
                        choices = NULL,
                        selected = NULL),
        ),
        box(title = "Main Cuisine",
            status = "primary",
            solidHeader = TRUE,
            width = 3,
            background = "maroon",
            selectInput("maincuisine1Input", "Select Main Cuisine",
                        choices = NULL,
                        selected = NULL)
        )
      ),
      fluidRow(
        box(title = "Minimum of Delivery Time",
            status = "primary",
            width = 12,
            background = "maroon",
            plotlyOutput("mindelivery_plot"),
            solidHeader = TRUE),
      ),
```

```
      fluidRow(
        box(title = "Minimum of Pickup Time",
            status = "primary",
            width = 12,
            background = "maroon",
            plotlyOutput("minpickup_plot"),
            solidHeader = TRUE)
      )
    ),
    ############################################################################################ RATING ANALYSIS
    tabItem(
      tabName = "boxplot",
      fluidRow(
        box(
          title = "Rating Analysis",
          width = 12,
          status = "primary",
          background = "maroon",
          selectInput("country2Input", "Select Country:",
                      choices = c("All", unique(rdata$country)),
                      selected = "All"),
          plotOutput("boxplot_output")
        )
      )
    )
  )
)


# UI: Combine the header, sidebar, body & skin color
ui <- dashboardPage(header, sidebar, body, skin = "red")


# Server logic
server <- function(input, output, session) {

  ############################################################################################ HOME

  # Render Rating Box
  output$ratingBox <- renderInfoBox({
    infoBox(
      "Customer Ratings",
      "104885",
      icon = icon("star"),
      color = "purple"
    )
  })

  # Render cuisineBox
  output$cuisineBox <- renderInfoBox({
    infoBox(
      "Top Cuisines",
      "563",
      icon = icon("list"),
      color = "yellow"
    )
  })

  ### RESTAURANT MAP
  # Update city options based on selected country
  observeEvent(input$countryInput, {
    cities <- unique(rdata$city[rdata$country == input$countryInput])
    updateSelectInput(session, "cityInput", choices = cities, selected = NULL)
  })

  # Update cuisine options based on selected city
  observeEvent(input$cityInput, {
    cuisines <- unique(rdata$main_cuisine[rdata$city == input$cityInput])
    updateSelectInput(session, "cuisineInput", choices = cuisines, selected = NULL)
  })

  # Filter the data based on user inputs
  filtered_data <- reactive({
    data <- rdata

    # Filter by country
    if (!is.null(input$countryInput) && input$countryInput != "All") {
      data <- data %>% filter(country == input$countryInput)
    }
    # Filter by city
    if (!is.null(input$cityInput) && input$cityInput != "All") {
      data <- data %>% filter(city == input$cityInput)
    }
    # Filter by cuisine
    if (!is.null(input$cuisineInput) && input$cuisineInput != "All") {
      data <- data %>% filter(main_cuisine == input$cuisineInput)
    }
    return(data)
  })

  ### Create  MAP
  output$restaurant_map <- renderLeaflet({
    data <- filtered_data()

    # Filter for top 10 restaurants by rating
    top_restaurants <- data %>%
      filter(!is.na(rating)) %>%
      group_by(city, name) %>%
      summarise(rating = ifelse(all(is.na(rating)), NA, max(rating, na.rm = TRUE)), .groups = "drop_last") %>%
      filter(!is.na(rating)) %>%
      arrange(city, desc(rating)) %>%
      group_by(city) %>%
      top_n(10, wt = rating)

    # Create a color palette for ratings
    rating_palette <- colorNumeric(palette = "RdYlGn", domain = data$rating)

```

```
    # Create the leaflet map
    leaflet() %>%
      addProviderTiles(providers$Stamen.TonerLite,
                       options = providerTileOptions(noWrap = TRUE)) %>%
      addCircleMarkers(data = data,
                       lat = ~latitude, lng = ~longitude,
                       color = ~rating_palette(rating),
                       fillColor = ~rating_palette(rating),
                       fillOpacity = 0.7,
                       radius = 5,
                       popup = ~name) %>%
      addLegend("bottomright", pal = rating_palette,
                values = data$rating,
                title = "Rating", opacity = 1)
  })
  # Create the table
  output$top_restaurants_table <- DT::renderDataTable({
    filtered_data()
  })


  ################################################################################ CUST. PREF.

  ### Filter Vertical Parent by Country
  output$verticalparent_plot <- renderPlot({
    filtered_data5 <- rdata %>%
      filter(country == input$country5Input, vertical_parent == input$verticalparent5Input)

    ggplot(filtered_data5, aes(x = vertical)) +
      geom_bar() +
      labs(x = "Vertical Parent", y = "Count") +
      facet_wrap(vars(country)) +
      theme_minimal()
  })


  ################################################################################ ASEAN TASTE

  # Render ASEAN Tastes plot
  output$asean_tastes_plot <- renderPlotly({
    # Filter for cuisine counts in the selected ASEAN country
    if (input$countryInput_asean == "All") {
      asean_cuisines <- rdata %>%
        filter(country %in% c("All","Singapore", "Malaysia", "Thailand", "Laos", "Philippines","Myanmar","Cambodia")) %>%
        count(main_cuisine)
    } else {
      asean_cuisines <- rdata %>%
        filter(country == input$countryInput_asean) %>%
        count(main_cuisine)
    }

    # Create the bar plot for ASEAN tastes
    asean_tastes_plot <- plot_ly(asean_cuisines, x = ~main_cuisine, y = ~n, type = "bar", colors = "purple") %>%
      layout(title = "ASEAN Tastes", xaxis = list(title = "Cuisine"), yaxis = list(title = "Count"))

    return(asean_tastes_plot)
  })


  ################################################################################ DELIVERY INSIGHTS

  observeEvent(input$deliveryprovider1Input,{
    country1 <- unique(rdata$country[rdata$delivery_provider == input$deliveryprovider1Input])
    updateSelectInput(session, "country1Input", choices = country1, selected = NULL)
  })

  observeEvent(input$country1Input,{
    city1 <- unique(rdata$city[rdata$country == input$country1Input])
    updateSelectInput(session, "city1Input", choices = city1, selected = NULL)
  })

  observeEvent(input$city1Input,{
    maincuisine1 <- unique(rdata$main_cuisine[rdata$city == input$city1Input])
    updateSelectInput(session, "maincuisine1Input", choices = maincuisine1, selected = NULL)
  })

  # Filter the data based on user inputs
  filtered_data1 <- reactive({
    data1 <- rdata
    # Filter by delivery provider
    if (!is.null(input$deliveryprovider1Input) && input$deliveryprovider1Input != "All") {
      data1 <- data1 %>% filter(delivery_provider == input$deliveryprovider1Input)
    }
    # Filter by country
    if (!is.null(input$country1Input) && input$country1Input != "All") {
      data1 <- data1 %>% filter(country == input$country1Input)
    }
    # Filter by city
    if (!is.null(input$city1Input) && input$city1Input != "All") {
      data1 <- data1 %>% filter(city == input$city1Input)
    }
    # Filter by main cuisine
    if (!is.null(input$maincuisine1Input) && input$maincuisine1Input != "All") {
      data1 <- data1 %>% filter(main_cuisine == input$maincuisine1Input)
    }
    return(data1)
  })


  #### minimum delivery time
  output$mindelivery_plot <- renderPlotly({
    data <- filtered_data1()

```

```r
    mindelivery <- data %>%
      filter(!is.na(minimum_delivery_time)) %>%
      group_by(city, name) %>%
      summarise(minimum_delivery_time = ifelse(all(is.na(minimum_delivery_time)), NA, max(minimum_delivery_time, na.rm = TRUE)), .groups = "drop_last") %>%
      filter(!is.na(minimum_delivery_time)) %>%
      arrange(city, desc(minimum_delivery_time)) %>%
      group_by(city) %>%
      top_n(10, wt = minimum_delivery_time)

    # Create the bar plot
    mindelivery_plot <- plot_ly(mindelivery, y = ~name, x = ~minimum_delivery_time, color = ~city, type = "bar", colors = "blue") %>%
      layout(title = "Minimum Delivery Time", yaxis = list(title = "Restaurant"), xaxis = list(title = "Minimum Delivery Time"), bargap = 0.2)

    return(mindelivery_plot)
  })


  #### minimum pickup time
  output$minpickup_plot <- renderPlotly({
    data2 <- filtered_data1()

    minpickup <- data2 %>%
      filter(!is.na(minimum_pickup_time)) %>%
      group_by(city, name) %>%
      summarise(minimum_pickup_time = ifelse(all(is.na(minimum_pickup_time)), NA,
                                             max(minimum_pickup_time, na.rm = TRUE)), .groups = "drop_last") %>%
      filter(!is.na(minimum_pickup_time)) %>%
      arrange(city, desc(minimum_pickup_time)) %>%
      group_by(city) %>%
      top_n(10, wt = minimum_pickup_time)

    # Create the bar plot
    minpickup_plot <- plot_ly(minpickup, x = ~name, y = ~minimum_pickup_time, color = ~city, type = "bar", colors = "yellow") %>%
      layout(title = "Minimum Pickup Time", xaxis = list(title = "Restaurant"), yaxis = list(title = "Minimum Pickup Time"), bargap = 0.2)

    return(minpickup_plot)
  })


  ################################################################################ RATING ANALYSIS (BOXPLOT)

  # Filter data based on selected country
  filtered_data2 <- reactive({
    if (input$country2Input == "All") {
      rdata
    } else {
      rdata[rdata$country == input$country2Input, ]
    }
  })

  # Generate box plot
  output$boxplot_output <- renderPlot({
    ggplot(filtered_data2(), aes(x = country, y = rating, fill = country)) +
      geom_boxplot() +
      labs(title = "Box Plot by Country",
           x = "Country",
           y = "Rating") +
      theme_minimal()
  })
}

# Run the application
shinyApp(ui, server)
```

Data Cleaning

Refer to:

https://drive.google.com/file/d/1e5blbagC83d6EtsOdsKYYDRdzKhG2EJp/view?usp=sharing (.Rmd file)

https://drive.google.com/file/d/1mvdJB9KSU2PwtnR9szozAO1t-Ou3kN8y/view?usp=sharing (pdf file)

File (All Files)

Refer to:

https://drive.google.com/drive/folders/1p_2oPzlc4rZNH8PUuiGYXmK_PqVm_HBB?usp=sharing