Sanwich EDA and Beyond..

AMAN SHARMA (B2021005)

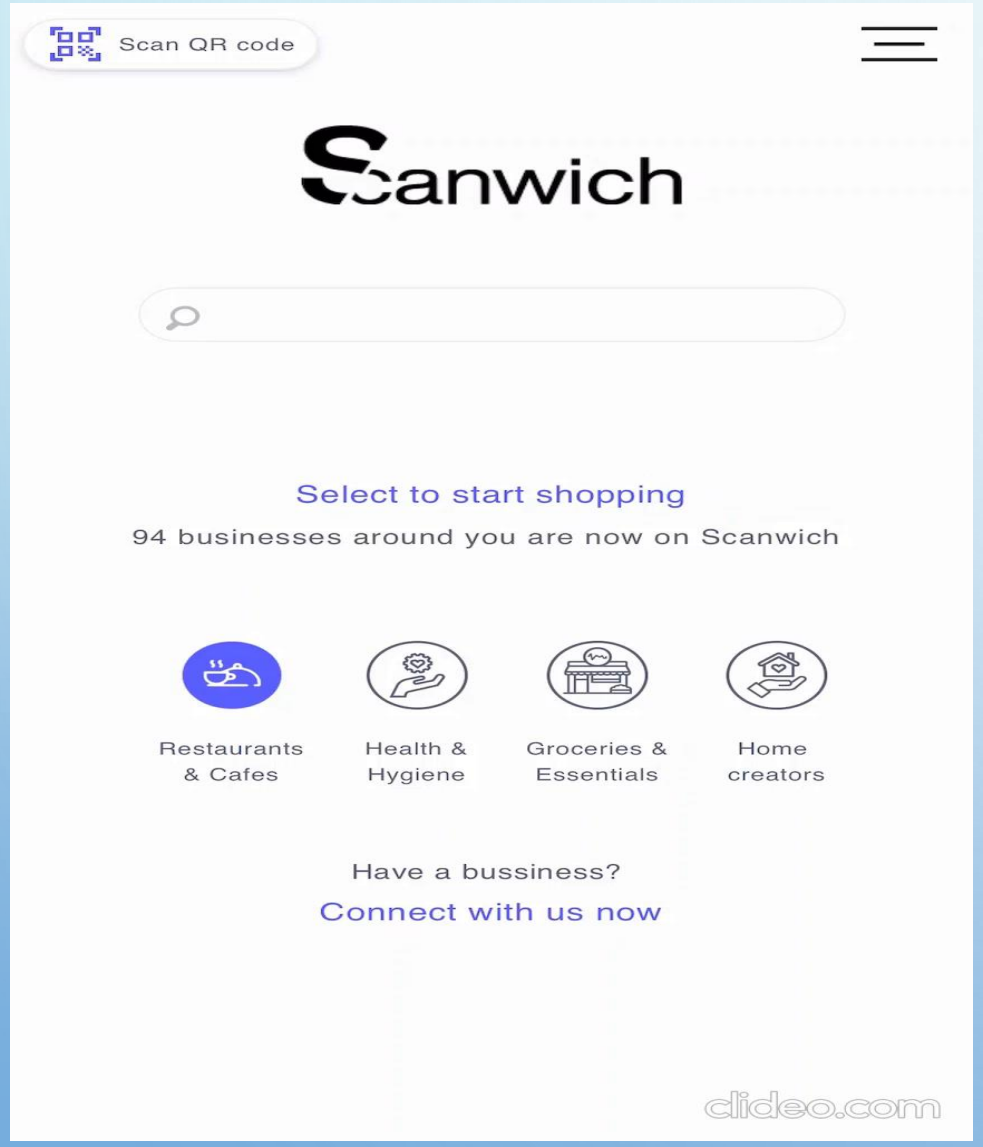
HAZIA FERNANDES (B2021021)

MANALI HEDAOO (B2021025)

SUMIT GROVER (B2021048)

TRISHIT BISWAS (B2021051)

CASE INTRODUCTION



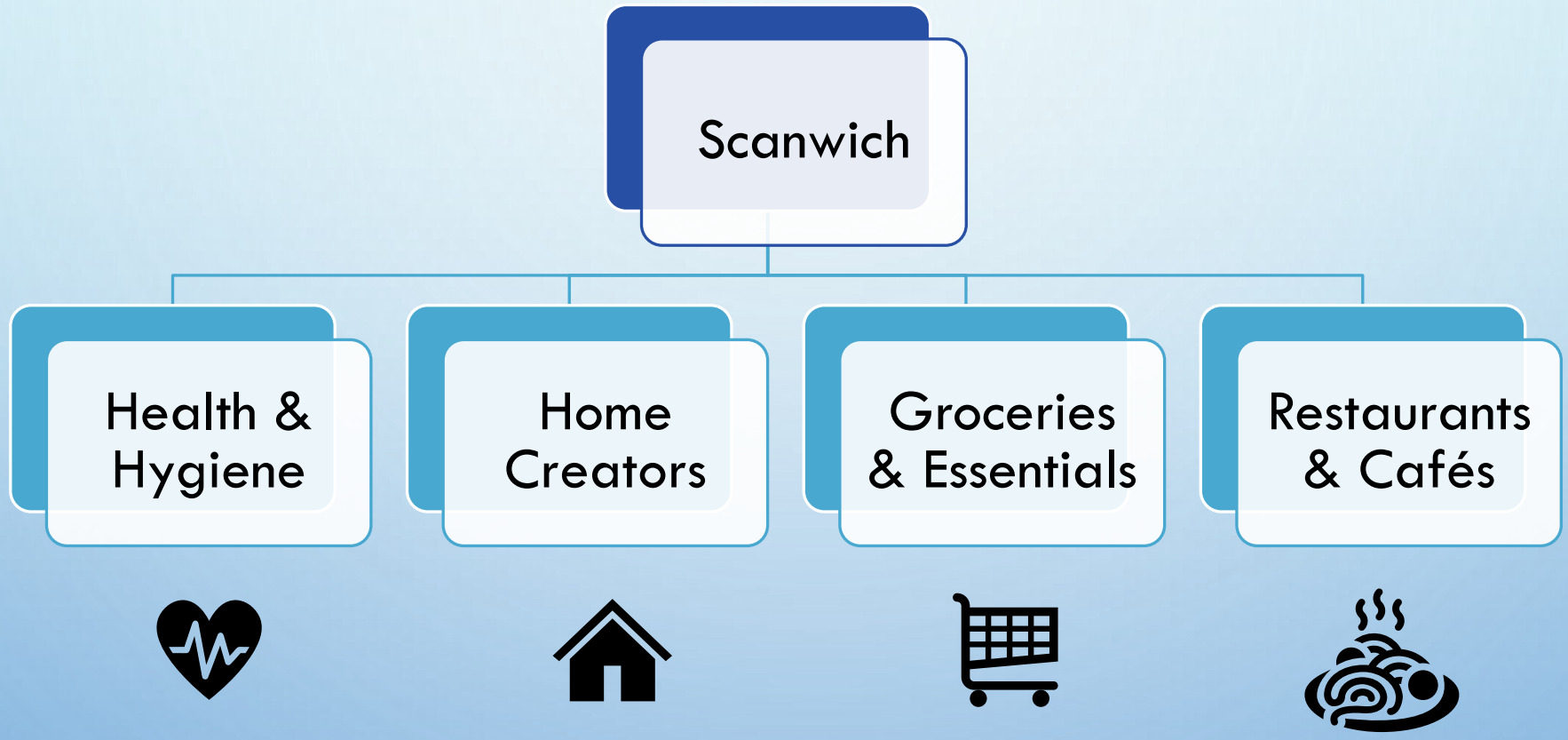
Scanwich is an online cataloguing service

Propelled forward during the Covid pandemic

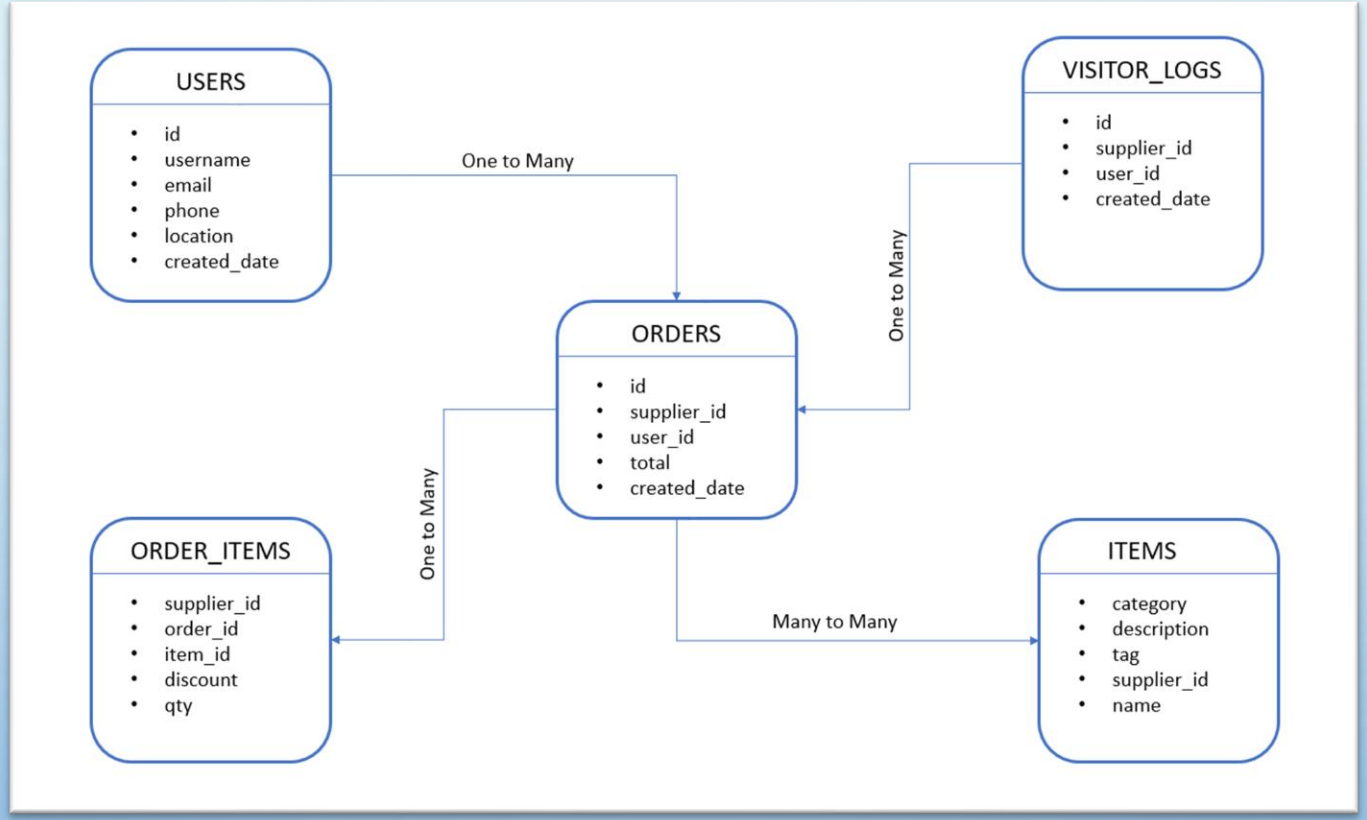
No contact ordering cataloguing

Presence in Goa and is currently expanding to other locations

DOMAINS CATERED BY SCANWICH



Database Schema



Python Setup and SQL Queries

Install Packages

```
# !pip install pymysql
# !pip install ipython-sql
# !pip install mysqlclient
# !pip install pandas
# !pip install matplotlib
# !pip install seaborn
# !pip install pmdarima
# !pip install mlxtend
# !pip install networkx
```

Setting up the mysql connection

```
import pymysql
import pandas as pd
import numpy as np

conn=pymysql.connect(host='localhost',port=int(3306),user=[redacted],passwd=[redacted],db=[redacted])

## Dummy Query to test
## df=pd.read_sql_query("SELECT count(*) FROM users;",conn)
```

```
--- 1. Daily Traffic
SELECT concat(SUBSTR(MONTHNAME(created_at),1,3),' ',YEAR(created_at)) as xaxis_label,YEAR(created_at) AS y, MONTH(created_at) AS
 m, MONTHNAME(created_at) AS month_name, COUNT(id) AS number_of_hits FROM visitor_logs  where supplier_id=1 GROUP BY y, m ,
month_name,xaxis_label ORDER BY y,m

--- 2. Best performing items - 3 or 5

--- Last Month
SELECT COUNT(item_id) AS value,item_name AS NAME, MONTHNAME(order_items.created_at) FROM order_items join orders on order_items.
order_id = orders.id WHERE MONTH(order_items.created_at) = MONTH(CURRENT_DATE()) - 1 AND YEAR(order_items.created_at) = YEAR(
CURRENT_DATE()) AND order_items.supplier_id = 1 group by item_id,item_name,MONTHNAME(order_items.created_at) order by value desc;

-- Overall
SELECT COUNT(item_id) AS value,item_name AS name FROM order_items join orders on order_items.order_id = orders.id WHERE orders.
supplier_id = 1 and orders.status != 'cancelled' group by item_id,item_name order by value desc limit 7;

--- 3. How many Orders & 4. Total sales

-- Daily Current MONTH
SELECT concat(DAY(created_at),' ', SUBSTR(MONTHNAME(created_at),1,3)) as xaxis_label,DAY(created_at) AS d, COUNT(id) AS
number_of_orders, SUM(total) AS total FROM orders where MONTH(created_at) = MONTH(CURRENT_DATE()) AND YEAR(created_at) = YEAR(
CURRENT_DATE()) and supplier_id = 1 GROUP BY d,xaxis_label  ORDER BY d;

-- Daily Last MONTH
SELECT concat(DAY(created_at),' ', SUBSTR(MONTHNAME(created_at),1,3)) as xaxis_label,DAY(created_at) AS d, COUNT(id) AS
number_of_orders, SUM(total) AS total FROM orders where MONTH(created_at) = MONTH(CURRENT_DATE()) - 1 AND YEAR(created_at) =
YEAR(CURRENT_DATE()) and supplier_id = 1 GROUP BY d,xaxis_label  ORDER BY d;
```

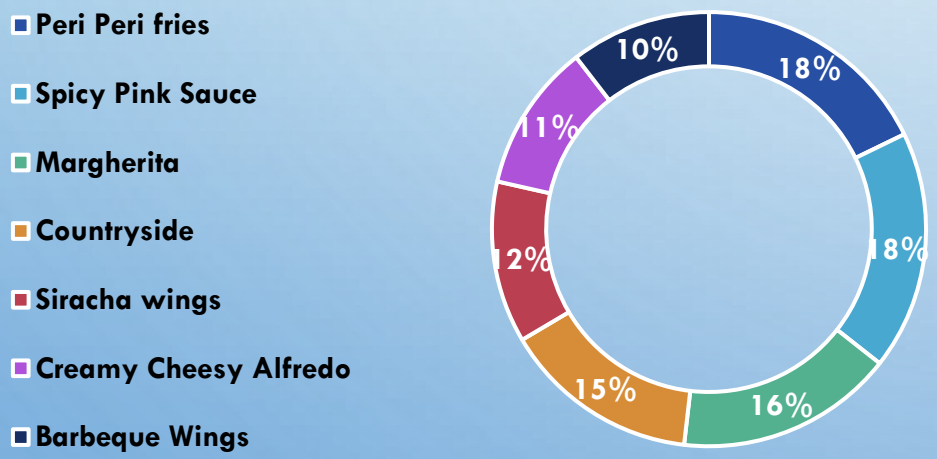
Descriptive Data Analysis - Supplier Side

Average Amount of Single Order
₹521

Average Number of Orders (Monthly)
24

Conversion Rate (Current Month)
8.6%

Best Performing Items



Brewed Black - Sales Trend

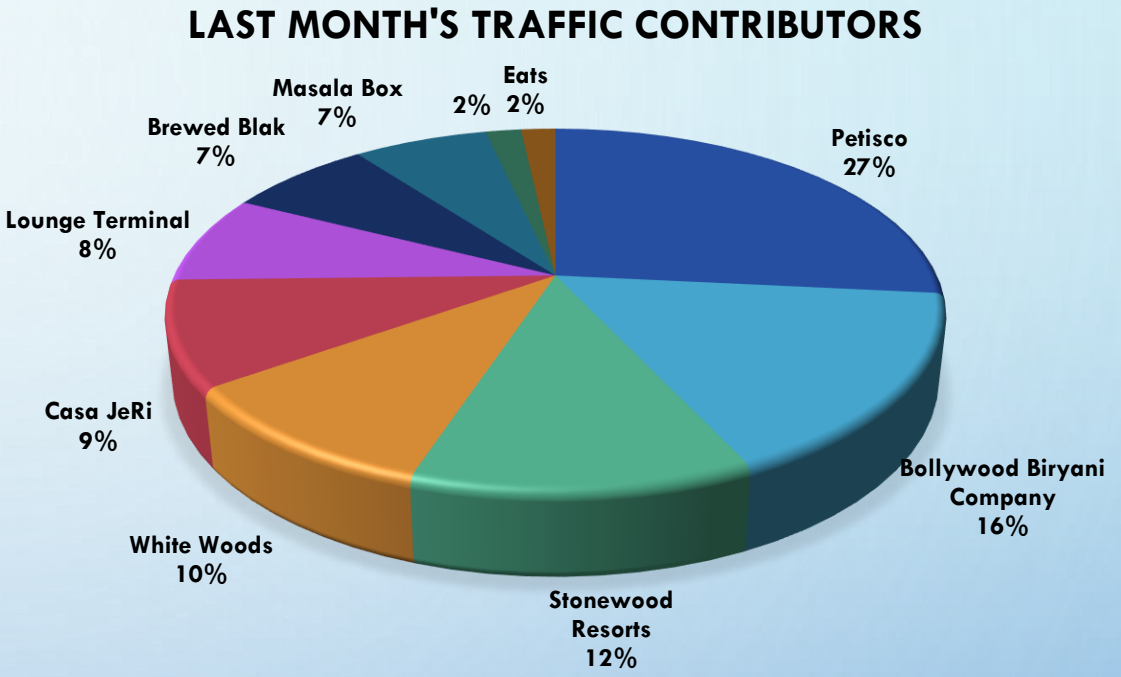
Descriptive Data Analysis - Admin Side

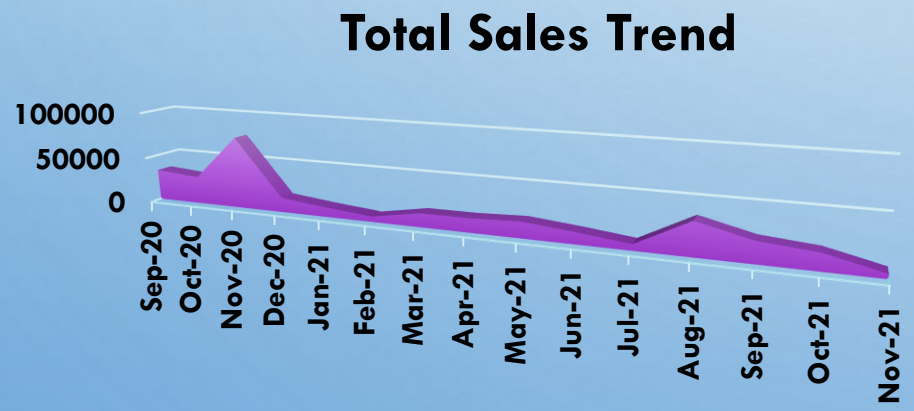
Average Monthly Order Amount

₹ 21000

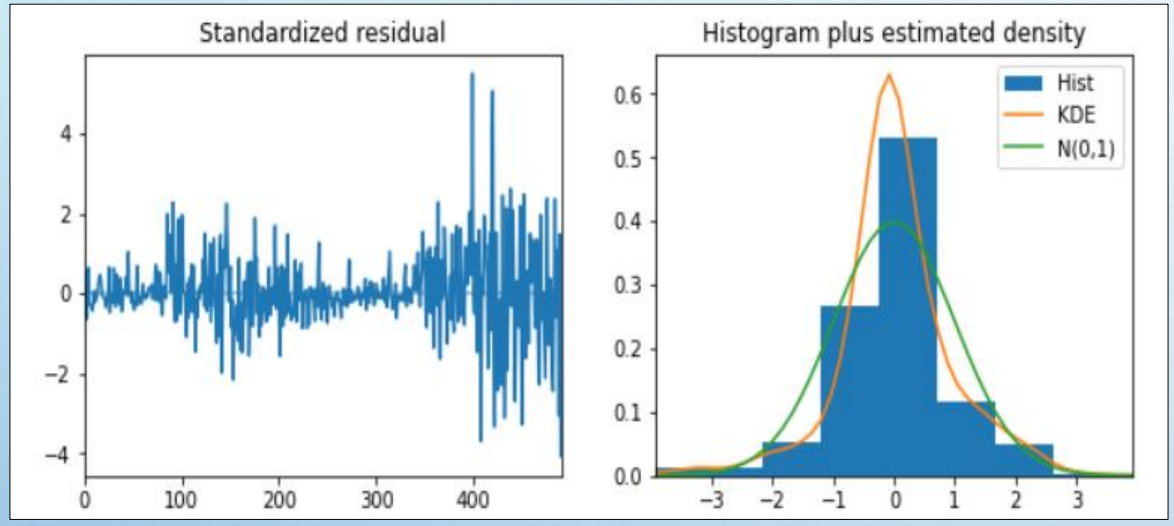
Average Monthly Orders Count

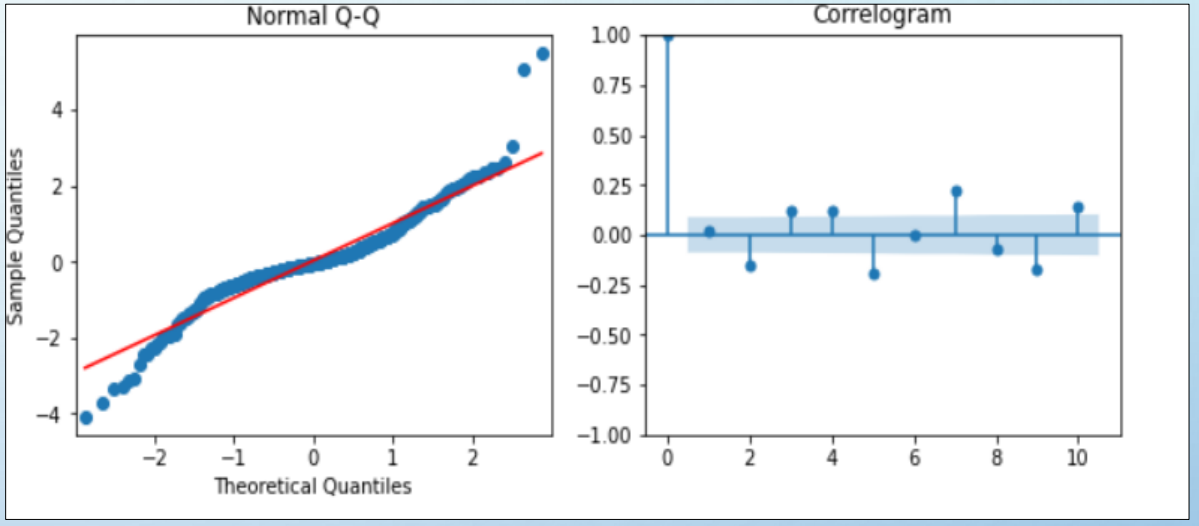
81

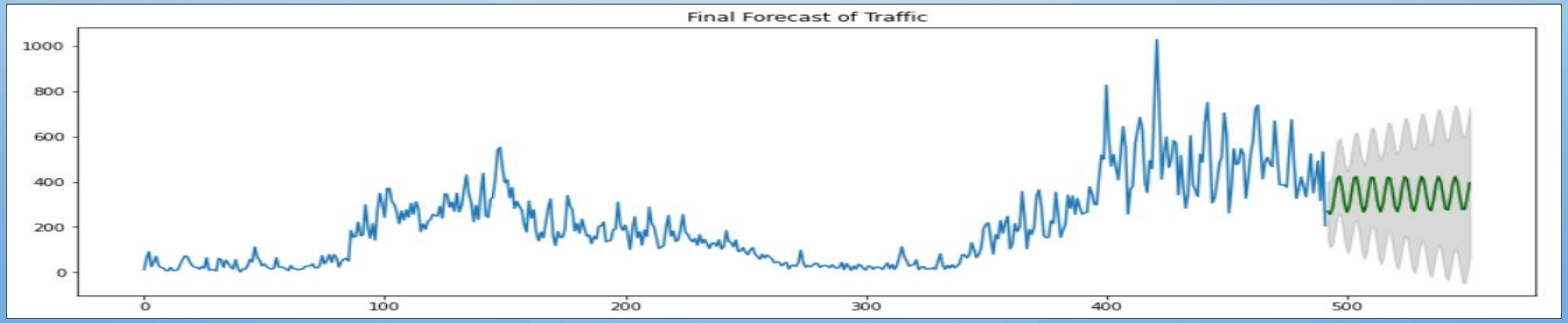




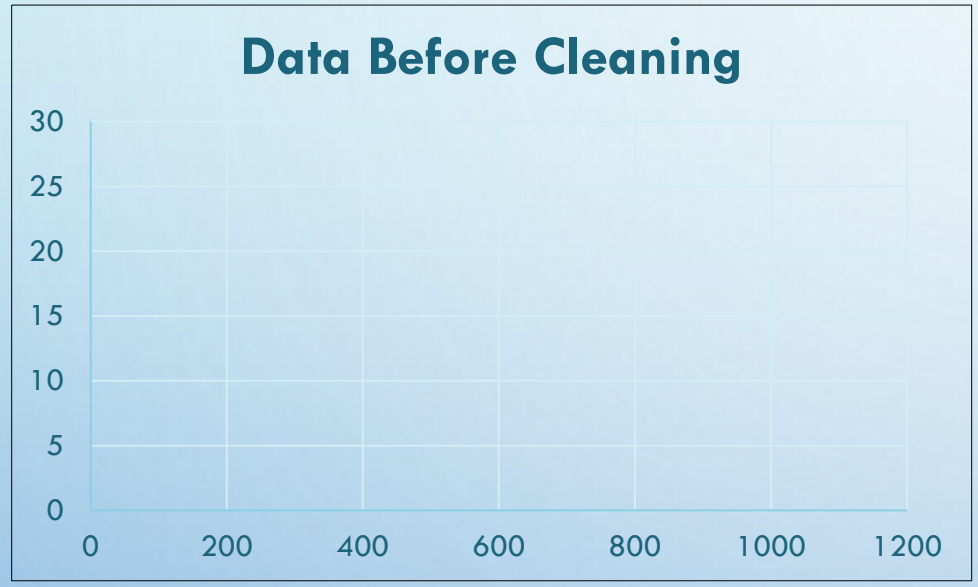
Time Series Analysis - Forecast Traffic (Auto ARIMA)

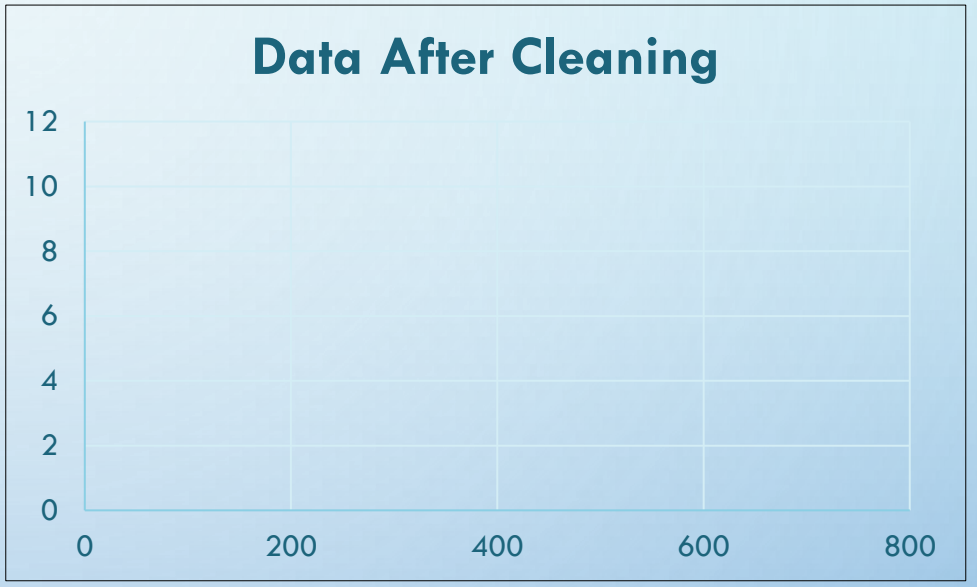


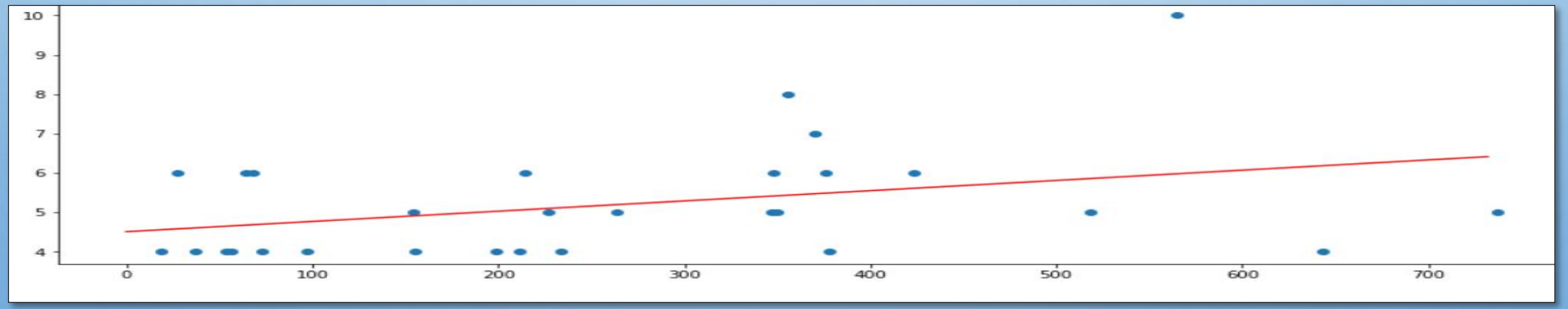




Linear Regression - Predict Daily Orders Based on Traffic



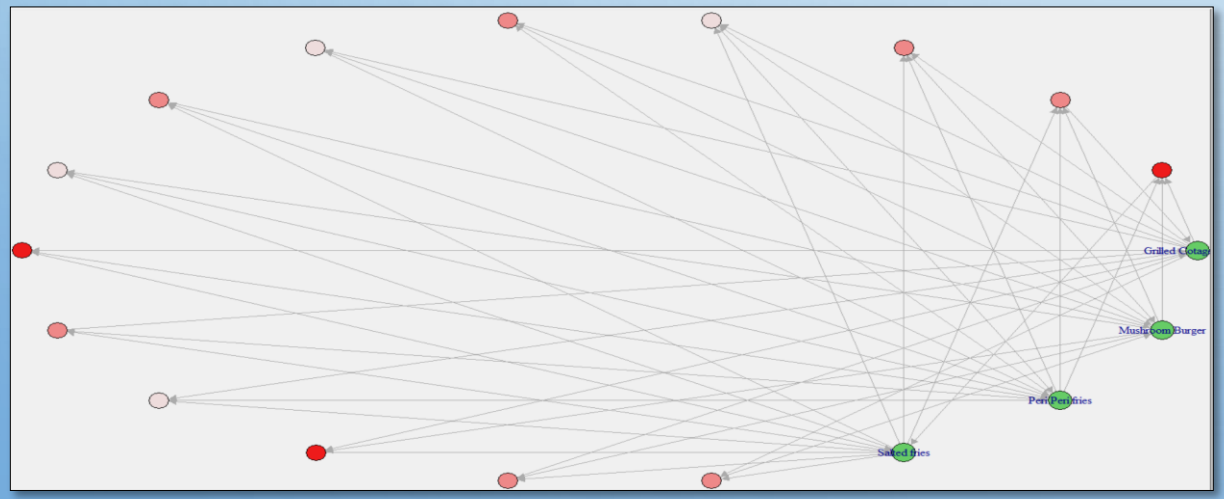


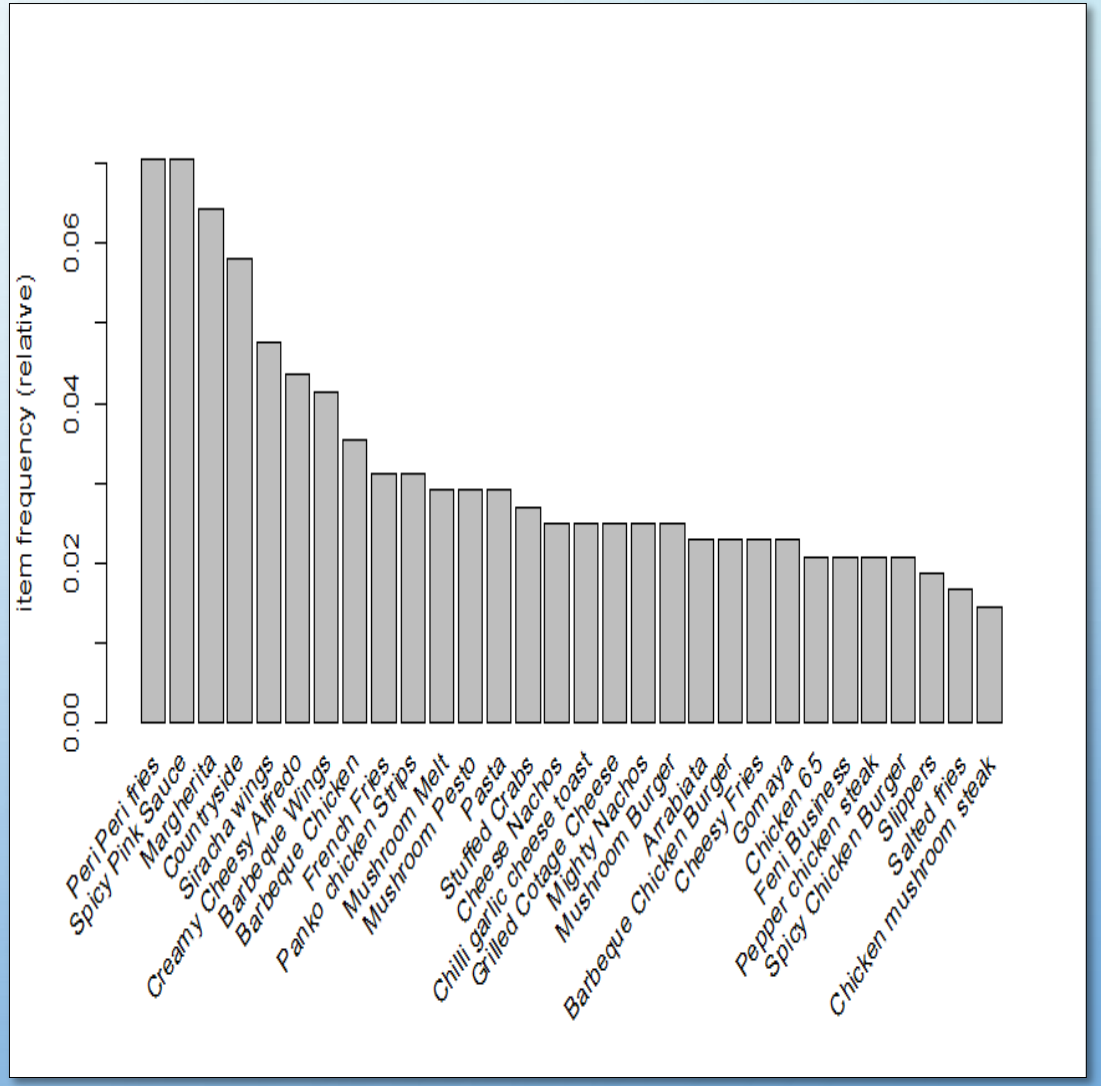


Product Recommendation – Business Specific

```
#Let's view our interpretation values using the Associan rule function.
df_ar = association_rules(df, metric = "confidence", min_threshold = 0.95)
df_ar
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Mushroom Burger, Grilled Cotage Cheese)	(Peri Peri fries)	0.012448	0.070539	0.012448	1.0	14.176471	0.011570	inf
1	(Peri Peri fries, Grilled Cotage Cheese)	(Mushroom Burger)	0.012448	0.024896	0.012448	1.0	40.166667	0.012138	inf
2	(Mushroom Burger, Salted fries)	(Grilled Cotage Cheese)	0.012448	0.024896	0.012448	1.0	40.166667	0.012138	inf
3	(Mushroom Burger, Grilled Cotage Cheese)	(Salted fries)	0.012448	0.016598	0.012448	1.0	60.250000	0.012242	inf
4	(Grilled Cotage Cheese, Salted fries)	(Mushroom Burger)	0.012448	0.024896	0.012448	1.0	40.166667	0.012138	inf
5	(Peri Peri fries, Salted fries)	(Grilled Cotage Cheese)	0.012448	0.024896	0.012448	1.0	40.166667	0.012138	inf
6	(Peri Peri fries, Grilled Cotage Cheese)	(Salted fries)	0.012448	0.016598	0.012448	1.0	60.250000	0.012242	inf
7	(Grilled Cotage Cheese, Salted fries)	(Peri Peri fries)	0.012448	0.070539	0.012448	1.0	14.176471	0.011570	inf
8	(Mushroom Burger, Salted fries)	(Peri Peri fries)	0.012448	0.070539	0.012448	1.0	14.176471	0.011570	inf
9	(Peri Peri fries, Salted fries)	(Mushroom Burger)	0.012448	0.024896	0.012448	1.0	40.166667	0.012138	inf



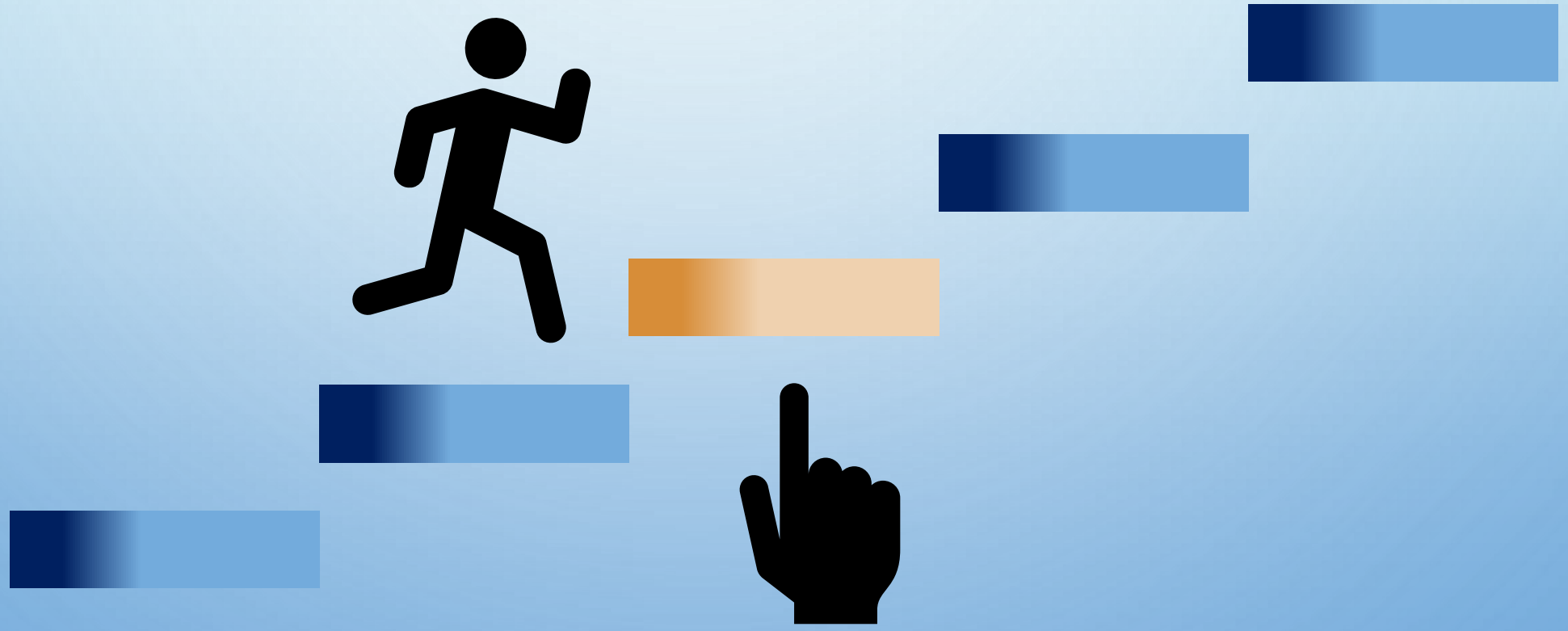


Logistic Regression – Browsing to Order

- Data currently Not available
- Have suggested the business to add the logic to record the data required.
- General Idea is to record the amount of time the user stays on a certain product before adding it to the cart and subsequently if it gets converted to order

Future Steps

- We are planning to add clustering algorithm for customer segmentation and focused recommendations.
- Need to improve the traffic and orders forecast – time series analysis
- Cluster based regression for improving performance of the predicting orders based on traffic.



THANK YOU