



Food Delivery Estimation

Ignatius Evans Erlangga



The Contents of presentation

A. Introductions

- 1 Importing The Libraries
- 2 Load the data
- 3 Business problem
- 4 EDA (exploratory data analysis)
- 5 Data visualizations
- 6 Data Preprocessing
- 7 Feature Scaling



The Contents of presentation

B. Data Modelling

- 1 Prediction of estimation food delivery time using:
 - Linear Regressions
 - SVR (Support Vector Regression)
 - KNN Regression
 - XGB Regression

C. Expansions of the Linear Regressions



Permasalahan Bisnis

Porter adalah Pasar Logistik Dalam Kota Terbesar di India. Sebagai pemimpin di pasar logistik dalam kota senilai \$40 miliar di negara ini, Porter berupaya meningkatkan taraf hidup lebih dari 1.50.000 mitra pengemudi dengan memberi mereka penghasilan dan kemandirian yang konsisten.

Saat ini, perusahaan telah melayani 5+ juta pelanggan. Porter bekerja dengan berbagai restoran untuk mengantarkan barang-barang mereka langsung ke masyarakat.



Tujuan Pengolahan data

Project ini bertujuan untuk membuat prediksi estimasi waktu pengantaran.

Dataset yang akan digunakan berfokus pada pengantaran makanan di India.



Porter memiliki 1,5 juta mitra pengemudi dan telah melayani 5 juta pelanggan. Oleh karena itu, perlu untuk meningkatkan kualitas pelayanan, terutama terkait masalah waktu pengantaran.



Preparation

Pengolahan data dimulai dengan melakukan import library yang dibutuhkan

Library data of manipulations:

- ❖ `import pandas as pd`
- ❖ `import numpy as np`

Library data of Visualizations:

- ❖ `import matplotlib.pyplot as plt`
- ❖ `import seaborn as sns`

Library data of Data Preprocessing:

- ❖ `from sklearn.model_selection import train_test_split`
- ❖ `from sklearn.preprocessing import LabelEncoder, MinMaxScaler`



Library of data modelling:

- ❖ `from sklearn.linear_model import LinearRegression`
- ❖ `from sklearn.svm import SVR`
- ❖ `from xgboost import XGBRegressor`
- ❖ `import xgboost as xgb`
- ❖ `from sklearn.neighbors import KNeighborsRegressor`



Preparation

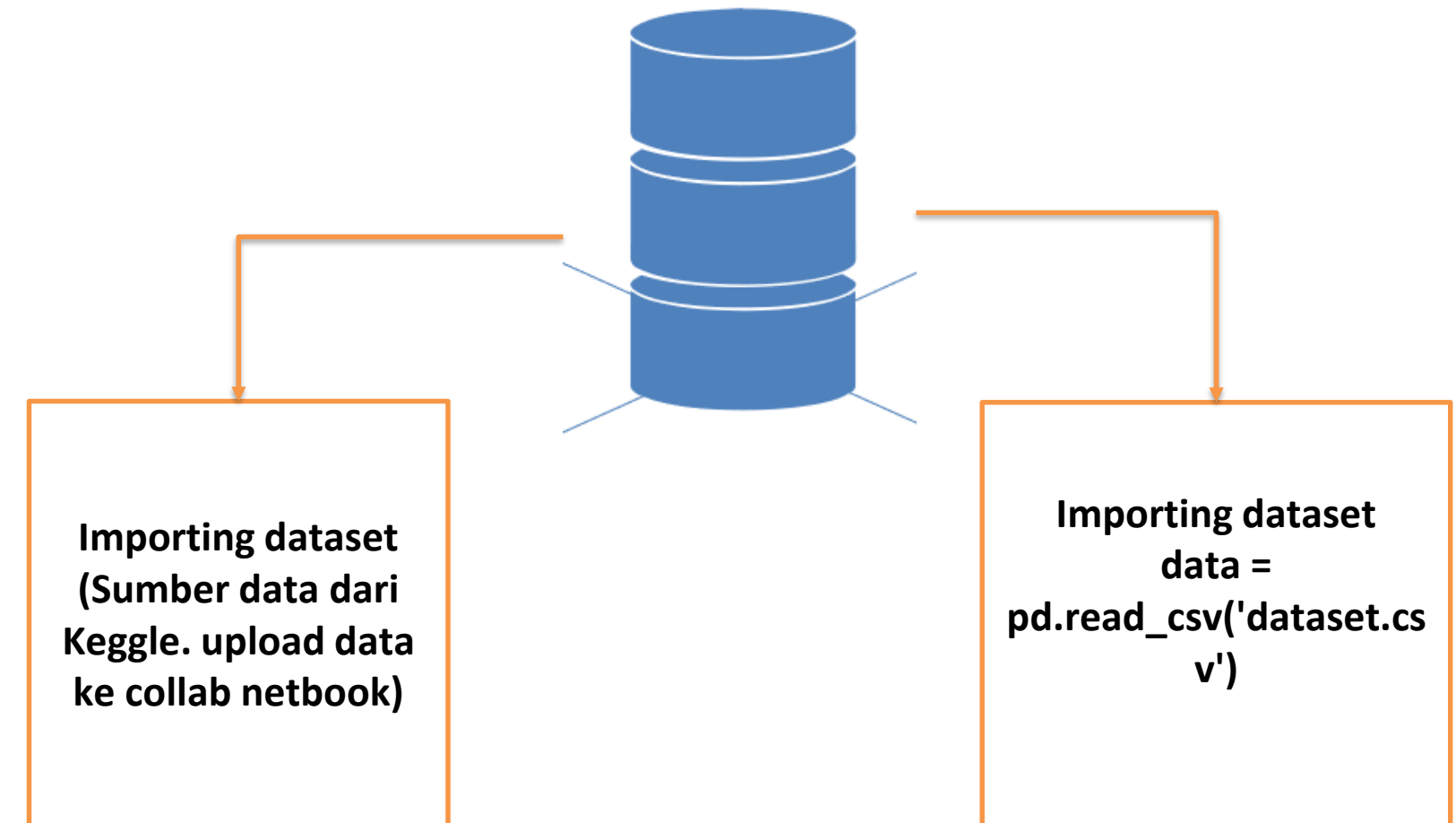
Library of data evaluation:

- ❖ `from sklearn.model_selection import train_test_split`
- ❖ `from sklearn.metrics import mean_squared_error`
- ❖ `from sklearn.metrics import mean_absolute_error`
- ❖ `from sklearn.metrics import r2_score`

Library data of export Modelling:

- ❖ Import pickle

Load CSV Format Data



Preparation

data.head()

	market_id	created_at	actual_delivery_time	store_id	store_primary_category	order_protocol	total_items	subtotal	num_distinct_items	min_item_price	max_item_price	total_onshift_partners	total_busy_partners	total_outstanding_orders
0	1.0	2015-02-06 22:24:17	2015-02-06 23:27:16	df263d996281d984952c07998dc54358	american	1.0	4	3441	4.0	557.0	1239.0	33.0	14.0	21.0
1	2.0	2015-02-10 21:49:25	2015-02-10 22:56:29	f0ade77b43923b38237db569b016ba25	mexican	2.0	1	1900	1.0	1400.0	1400.0	1.0	2.0	2.0
2	3.0	2015-01-22 20:39:28	2015-01-22 21:09:09	f0ade77b43923b38237db569b016ba25	NaN	1.0	1	1900	1.0	1900.0	1900.0	1.0	0.0	0.0
3	3.0	2015-02-03 21:21:45	2015-02-03 22:13:00	f0ade77b43923b38237db569b016ba25	NaN	1.0	6	6900	5.0	600.0	1800.0	1.0	1.0	2.0
4	3.0	2015-02-15 02:40:36	2015-02-15 03:20:26	f0ade77b43923b38237db569b016ba25	NaN	1.0	3	3900	3.0	1100.0	1600.0	6.0	6.0	9.0

Preparation

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 197428 entries, 0 to 197427
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   market_id                            196441 non-null  float64
1   created_at                           197428 non-null  object
2   actual_delivery_time                 197421 non-null  object
3   store_id                             197428 non-null  object
4   store_primary_category               192668 non-null  object
5   order_protocol                       196433 non-null  float64
6   total_items                          197428 non-null  int64
7   subtotal                             197428 non-null  int64
8   num_distinct_items                   197428 non-null  int64
9   min_item_price                       197428 non-null  int64
10  max_item_price                       197428 non-null  int64
11  total_onshift_partners                181166 non-null  float64
12  total_busy_partners                   181166 non-null  float64
13  total_outstanding_orders              181166 non-null  float64
dtypes: float64(5), int64(5), object(4)
memory usage: 21.1+ MB
```

Tipe –Tipe Data dalam Setiap Kolom

Preparation

data.describe()

	created_at	actual_delivery_time	total_items	subtotal	num_distinct_items	min_item_price	max_item_price
count	78474	78472	78474.000000	78474.000000	78473.000000	78473.000000	78473.000000
mean	2015-02-04 21:22:20.727208960	2015-02-04 22:12:03.785821696	3.246642	2726.084754	2.698979	689.728161	1180.300944
min	2014-10-19 05:24:15	2015-01-21 16:17:43	1.000000	0.000000	1.000000	-86.000000	0.000000
25%	2015-01-29 02:19:21.249999872	2015-01-29 03:09:36.249999872	2.000000	1448.000000	2.000000	299.000000	800.000000
50%	2015-02-05 03:17:07.500000	2015-02-05 04:25:30.500000	3.000000	2253.000000	2.000000	595.000000	1095.000000
75%	2015-02-12 01:30:09.249999872	2015-02-12 02:14:43.500000	4.000000	3455.000000	3.000000	945.000000	1400.000000
max	2015-02-18 06:00:44	2015-02-19 22:45:31	411.000000	24300.000000	20.000000	8999.000000	8999.000000
std	NaN	NaN	2.958795	1841.830626	1.646766	537.978962	583.636451

Descriptive statistics

Preparation

Porter delivery time estimation

Kolom (14):

- market_id = ID market (int) lokasi restoran
- created_at = Timestamp (tanggal dan waktu) pesanan dibuat
- actual_delivery_time = Timestamp (tanggal dan waktu) pesanan dikirimkan
- store_id = ID restoran
- store_primary_category = Kategori restoran
- order_protocol = Code value (int) untuk protokol pesanan (bagaimana pesanan dilakukan, misalnya dari Porter, menghubungi restoran langsung, pre-booked, pihak ketiga, dll.)
- total_items = Total jumlah barang yang dipesan
- subtotal = Harga akhir pesanan
- num_distinct_items = Jumlah barang yang berbeda (distinct) dalam pesanan
- min_item_price = Harga item terendah dalam pesanan
- max_item_price = Harga item tertinggi dalam pesanan
- total_onshift_partners = Jumlah mitra Porter yang sedang bertugas (stand by)
- total_busy_partners = Jumlah mitra Porter yang sedang menyelesaikan pesanan lain
- total_outstanding_orders = Jumlah pesanan yang harus diselesaikan at the moment

- Data terdiri dari **197.428 baris dan sekitar 14 kolom.**
- **Deskripsi tiap Fitur**
- Fitur Target(variabel dependen) = 'delivery_duration'.
- **Fitur Pendukung(variabel independen)**



Exploratory Data Analysis (EDA)

```
[6] #Change dtype
    #to datetime
    data.created_at = pd.to_datetime(data.created_at)
    data.actual_delivery_time = pd.to_datetime(data.actual_delivery_time)

    #to object
    data.market_id = data.market_id.astype('object')
    data.store_primary_category = data.store_primary_category.astype('object')
    data.order_protocol = data.order_protocol.astype('object')
```

```
[7] #Duplicate and null values check

    print(f'Dataset dimensions\t: {data.shape}')
    print(f'Rows duplicated\t\t: {data.duplicated().sum()}')

    type_null = pd.DataFrame(data.dtypes).T.rename(index={0: 'Column Type'})
    type_null = pd.concat([type_null, pd.DataFrame(data.isnull().sum()).T.rename(index={0: 'Amount of Null Values'})])
    type_null = pd.concat([type_null, pd.DataFrame(round(data.isnull().sum()/data.shape[0]*100, 4)).T.rename(index={0: 'Percentage of Null Values'})])
    type_null = type_null.T
    type_null = type_null.reset_index().rename(columns={'index': 'feature'})
    type_null
```

- Mengubah tipe data
- Pengecekan data duplikat dan data null

Exploratory Data Analysis (EDA)

➡ The data consist of 69 rows of data with total transaction equal to zero.
About 0.09% of total data

The data consist of 1019 rows of data with min item price quantity less/equal than zero.
About 1.30% of total data

The data consist of 6 rows of data with max item price equal to zero.
About 0.01% of total data

The data consist of 8 rows of data with current online Porter Partners that is less than zero.
About 0.01% of total data

The data consist of 12 rows of data with total busy Porter Partners that is less than zero.
About 0.02% of total data

The data consist of 12 rows of data with current ongoing orders less than zero.
About 0.02% of total data

#Remove anomaly (copy it first)

```
dataset = data.copy()
```

#Drop unnecessary columns

```
dataset.drop(labels=['store_id'], axis=1, inplace=True)
```

```
dataset = data.copy()
```

```
dataset = dataset[
    (dataset['subtotal'] > 0) &
    (dataset['min_item_price'] > 0) &
    (dataset['max_item_price'] > 0) &
    (dataset['total_onshift_partners'] >= 0) &
    (dataset['total_busy_partners'] >= 0) &
    (dataset['total_outstanding_orders'] >= 0)
]
```

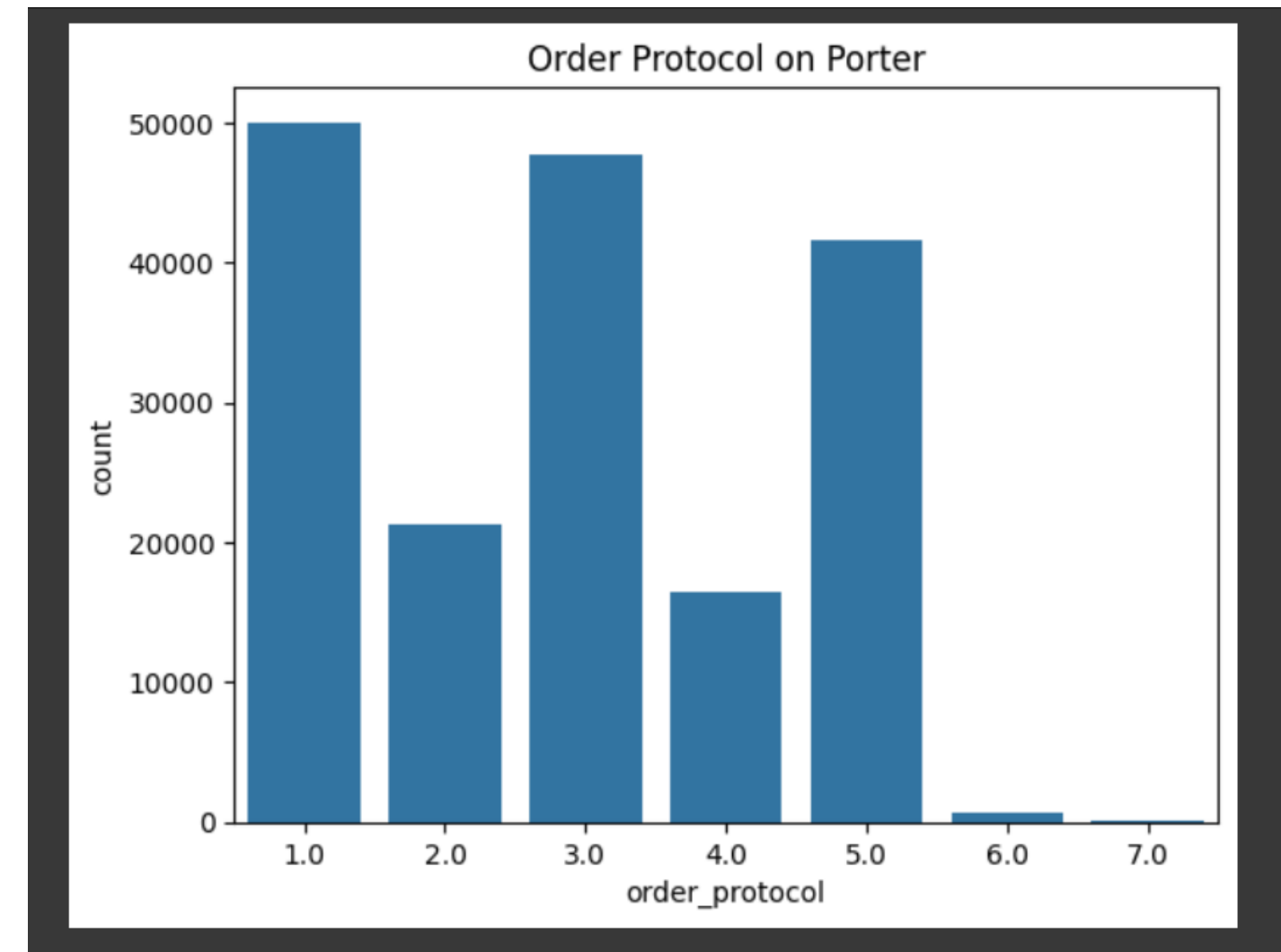
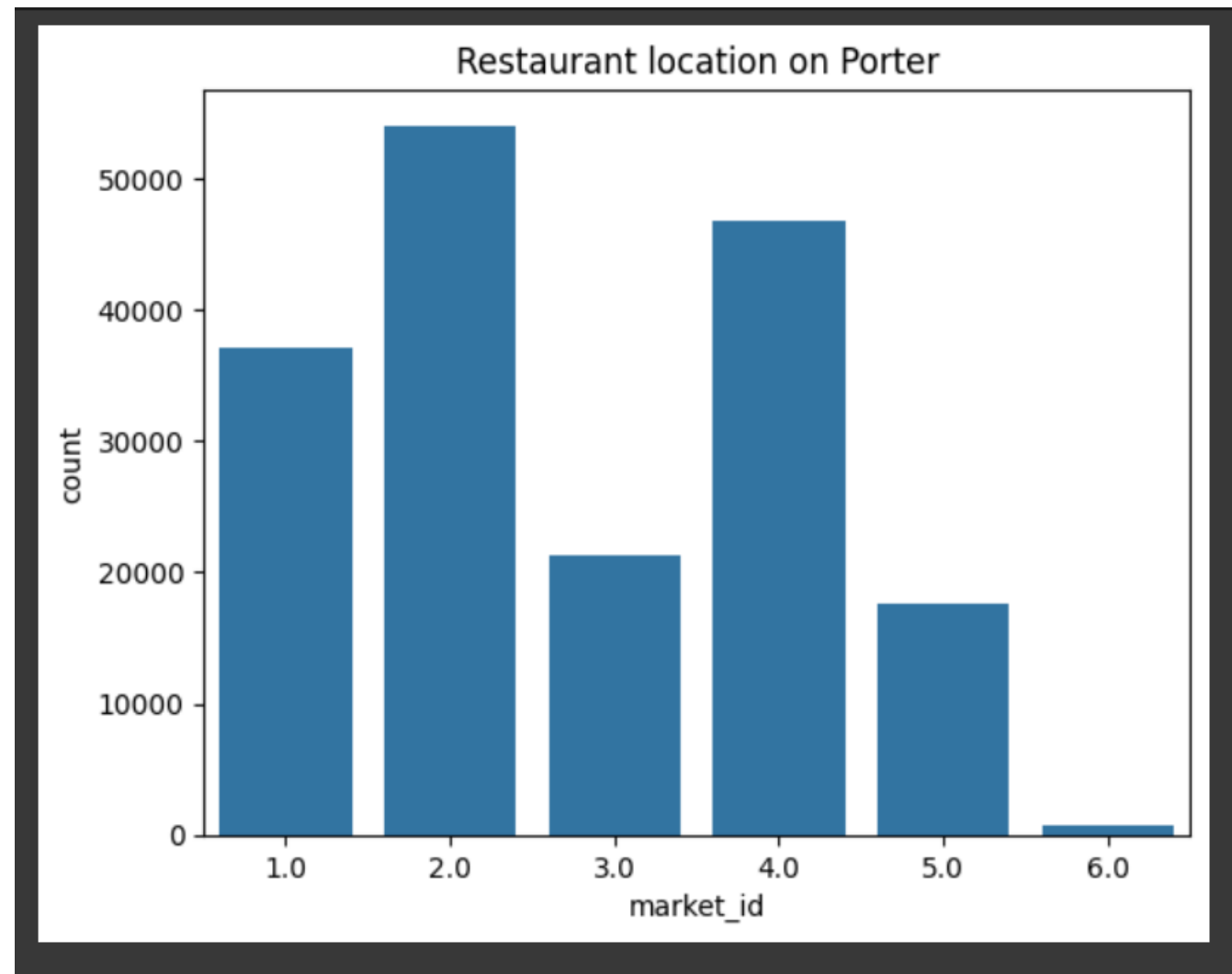
#Add column

```
dataset['delivery_duration'] =
(dataset['actual_delivery_time'] -
dataset['created_at']).dt.total_seconds()/60
```

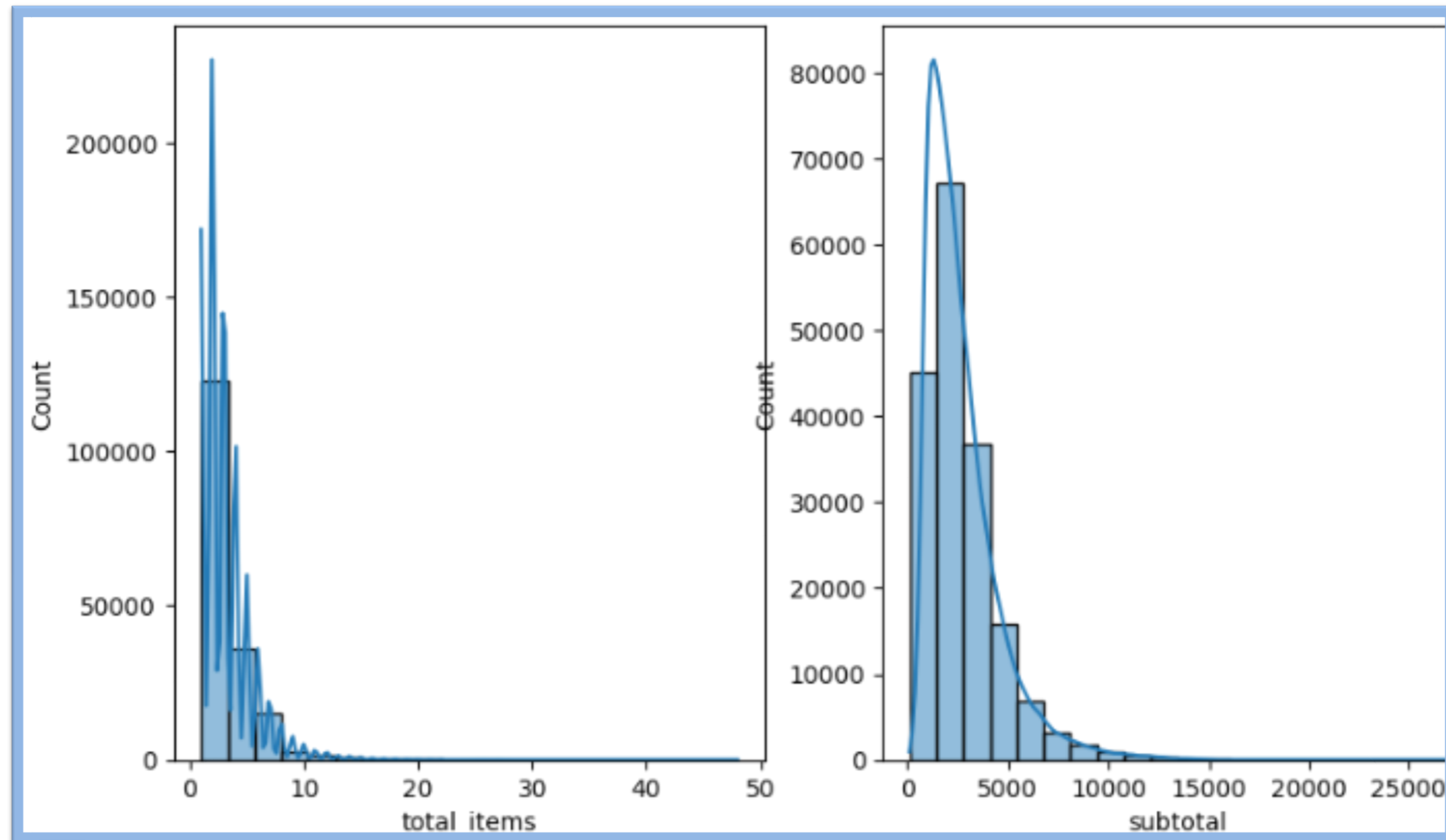
- Melihat presentasi data yang memiliki nilai Null.
- Menghilangkan data anomaly.
- Drop kolom yang kurang diperlukan
- Tambahkan nama kolom yang dibutuhkan

*hal ini perlu dilakukan untuk memastikan data bersih dan proper

Exploratory Data Analysis (EDA)



Exploratory Data Analysis (EDA)



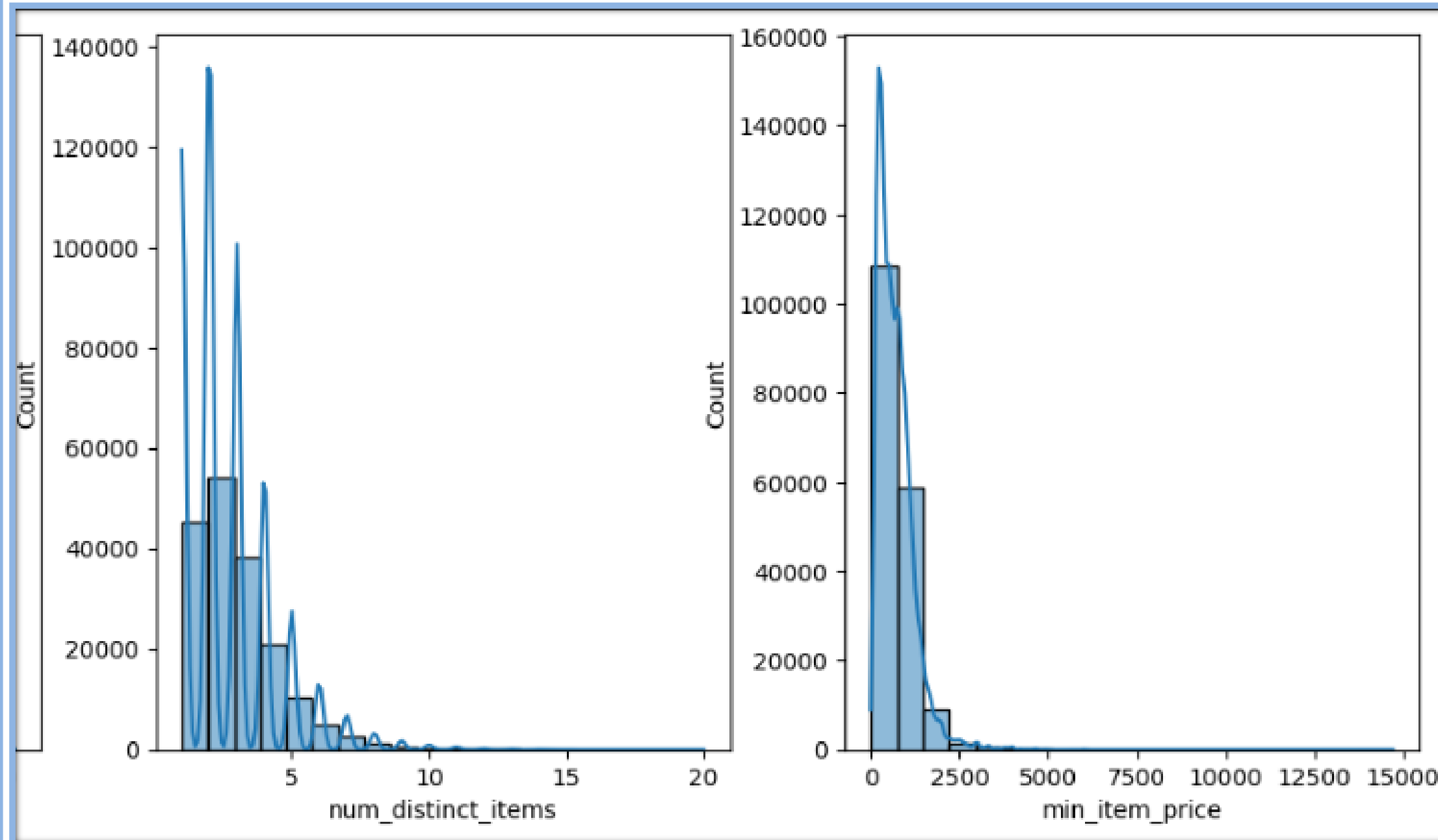
Total items:

Biasanya, Konsumen membeli

2 buah barang
dalam satu kali
transaksi

Sub total : Kebanyakan
transaksi memiliki total
biaya **Rs. 2000**

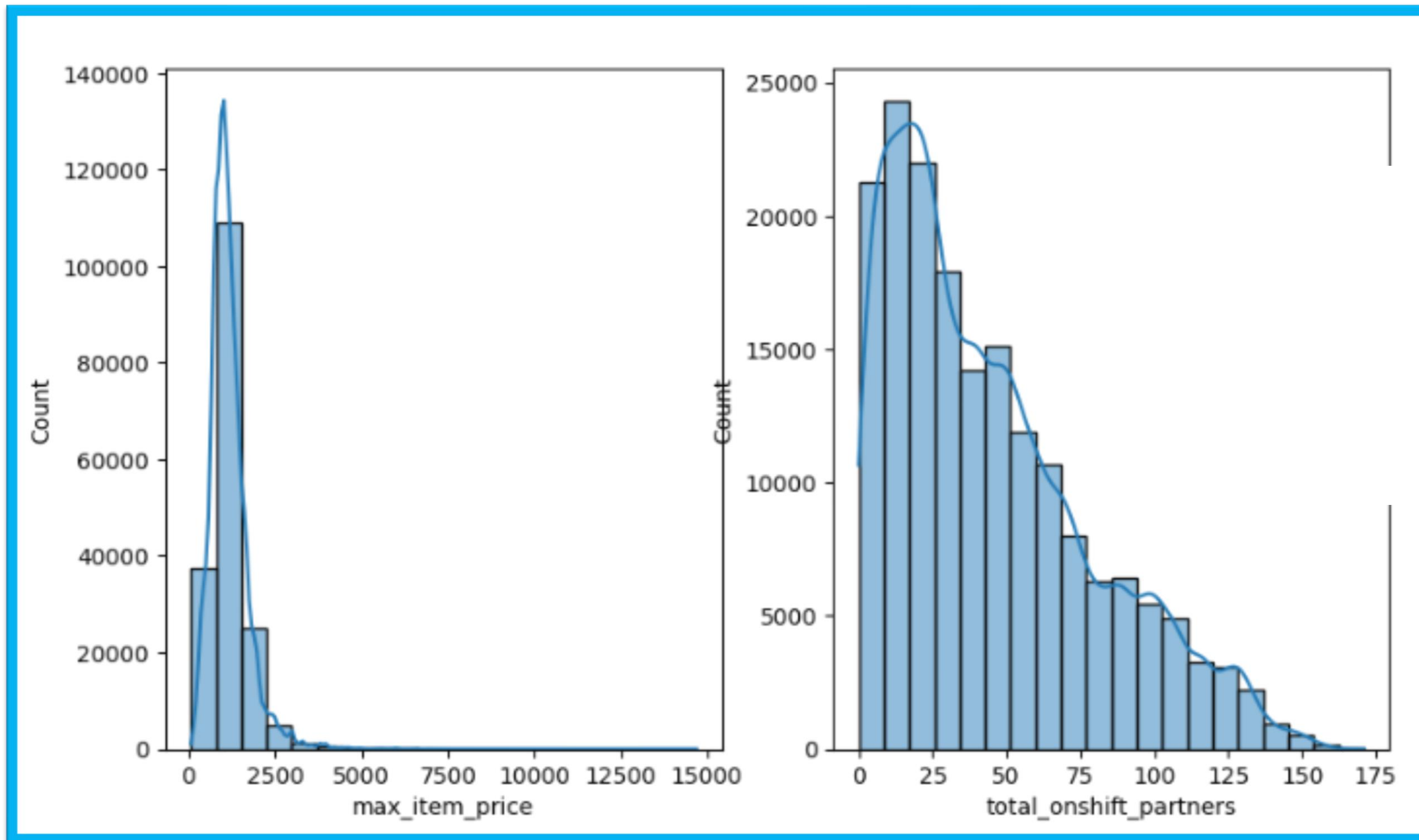
Exploratory Data Analysis (EDA)



Num_distinc_ items:
Konsumen biasanya memesan
2 Jenis
Barang per transaksi.

Min_items_price:
Umumnya harga
barang/makanan yang paling
murah adalah dibawah
Rs.1000.

Exploratory Data Analysis (EDA)

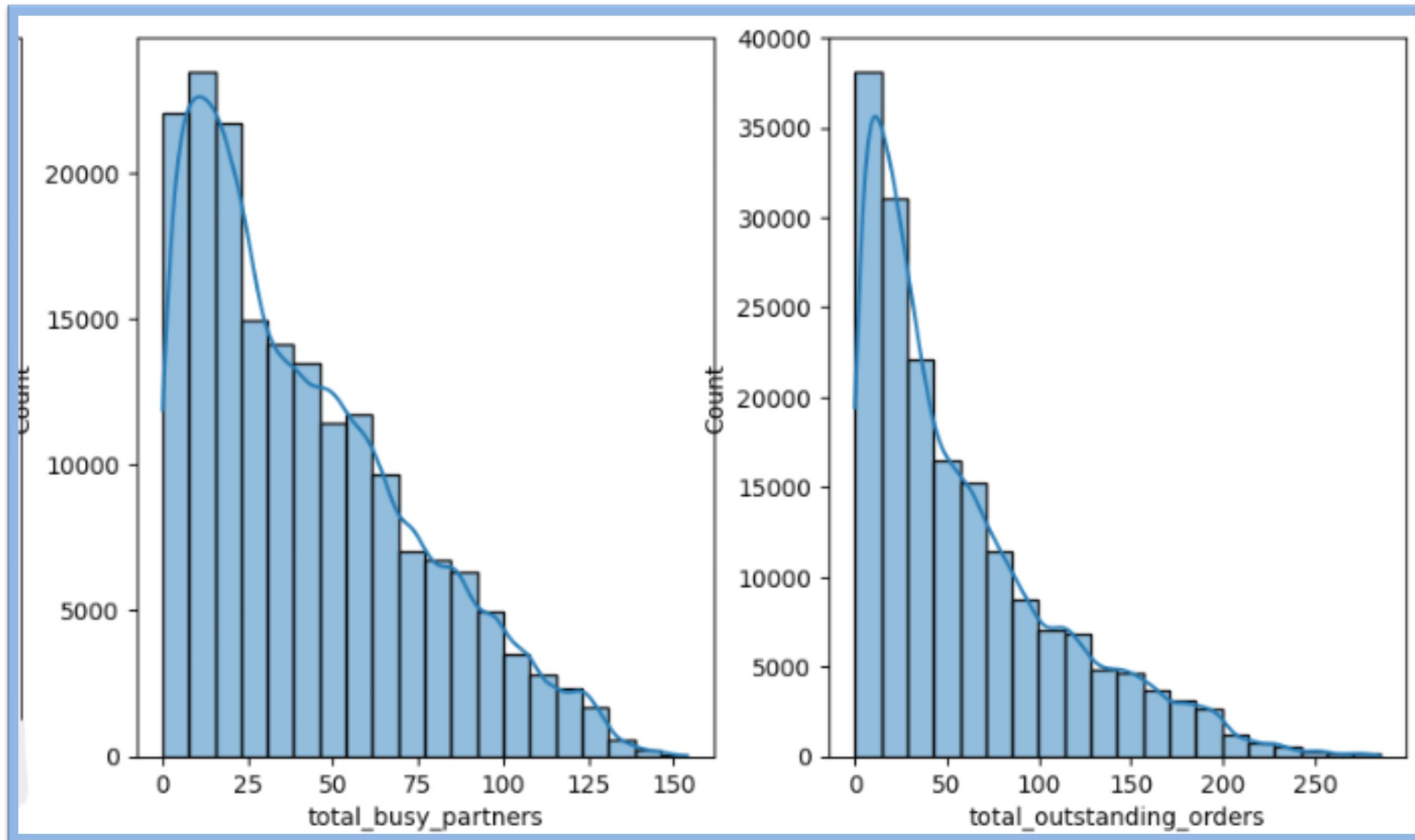


Max_items_price:

Harga termahal per item mayoritas ada pada rentang **Rs. 500 sampai Rs. 1000**

Total_onshift_Partners: Jumlah kurir yang available ketika pesanan sedang banyak sering kali **sedikit**, karena mayoritas dari mereka sedang **mengerjakan pesanannya**.

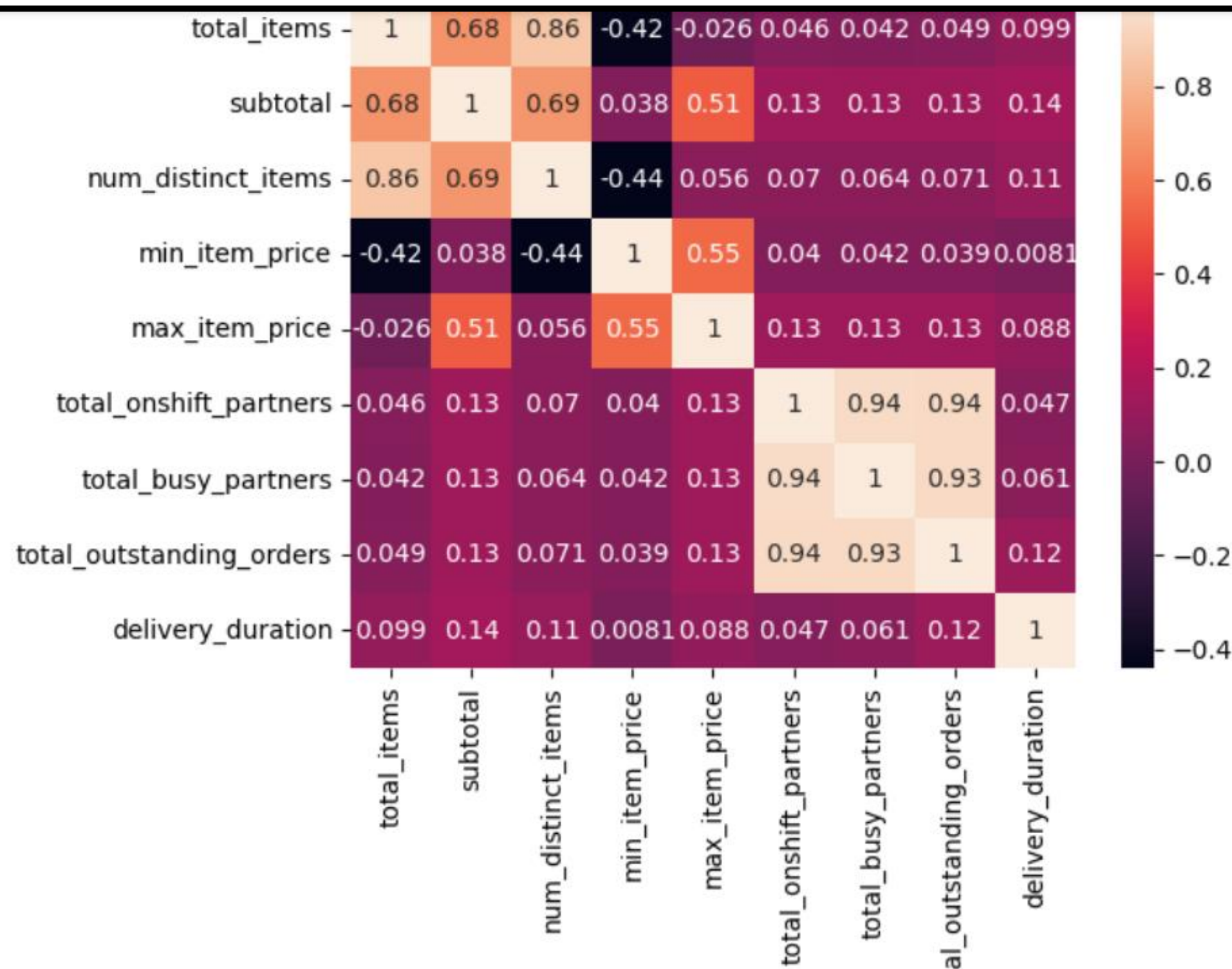
Exploratory Data Analysis (EDA)



Total Busy Partners:
Semakin **sedikit jumlah order**, maka **semakin banyak jumlah kurir** yang melakukan pengiriman.

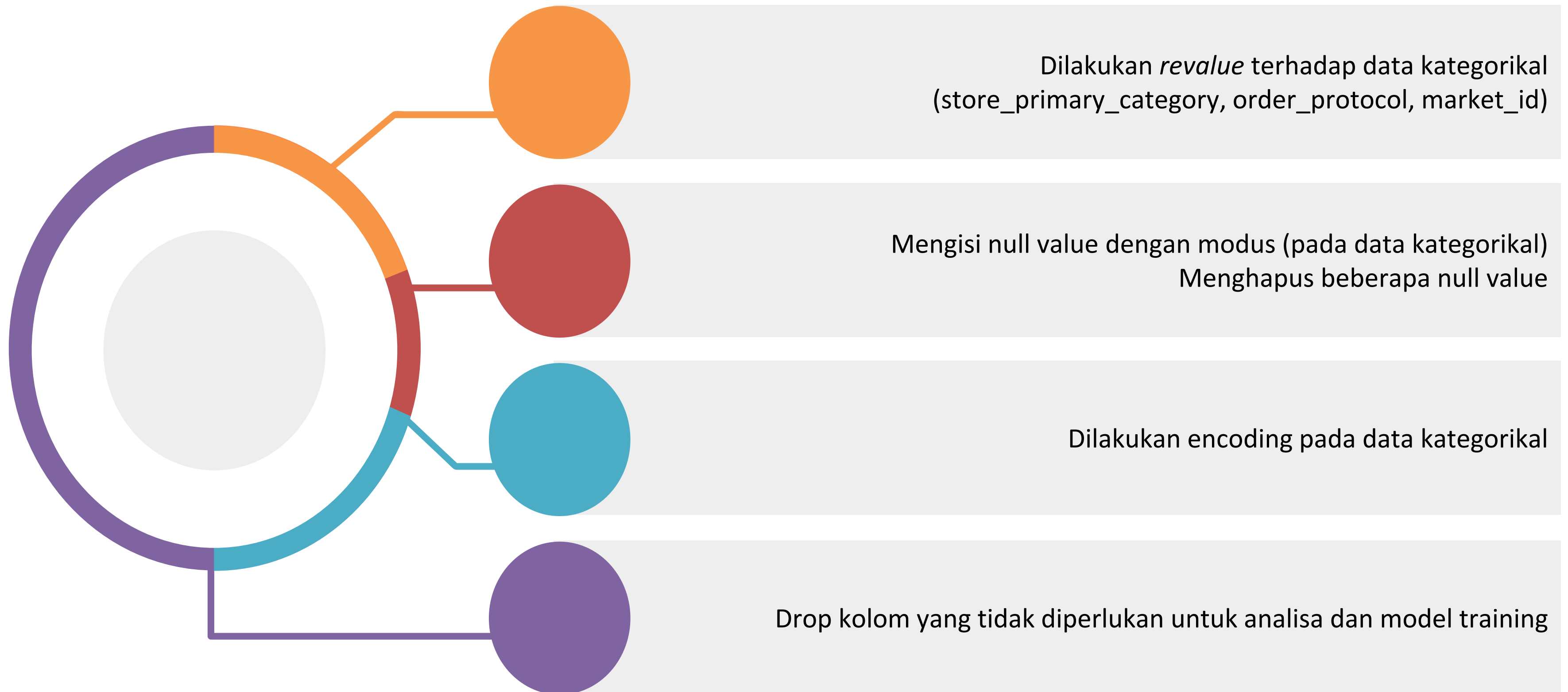
Total_outstanding_orders:
Semakin banyak **jumlah orderan**, semakin sedikit jumlah **orderan yang belum ditangani**.

Exploratory Data Analysis (EDA)



- **total_items ↔ subtotal**
Jumlah barang yang dipesan memiliki korelasi sebesar **68%** kepada total harga yang harus dibayarkan
- **subtotal ↔ num_distinct_item**
Total harga yang harus dibayarkan memiliki korelasi sebesar **69%** terhadap jumlah tipe barang yang dipesan oleh konsumen.

Data Preprocessing



Data Preprocessing

Tampilan data preprocessing pada Collab

```
[ ] # Copy data
    data_prepro = dataset.copy()
```

```
[ ] # Mendeteksi kembali Missing Values
    data_prepro.isna().sum()
```

```
market_id          911
created_at          0
actual_delivery_time 7
store_primary_category 4182
order_protocol      911
total_items         0
subtotal            0
num_distinct_items  0
min_item_price      0
max_item_price      0
total_onshift_partners 0
total_busy_partners  0
total_outstanding_orders 0
delivery_duration    7
dtype: int64
```

Melakukan pengecekan kembali atas data yang bernilai null

Data Preprocessing

Tampilan data preprocessing pada Collab

```
# Mengelompokkan data dalam kolom store_primary_category menjadi berbagai tipe restoran

def restaurant_category (store_primary_category):
    ethnic_based_restaurant = ['american', 'mexican', 'indian', 'italian', 'thai',
                               'chinese', 'singaporean', 'japanese', 'greek', 'filipino',
                               'asian', 'vietnamese', 'middle-eastern', 'persian',
                               'korean', 'latin-american', 'burmese', 'hawaiian',
                               'british', 'nepalese', 'peruvian', 'turkish', 'ethiopian',
                               'german', 'french', 'caribbean', 'afghan', 'pakistani',
                               'moroccan', 'malaysian', 'brazilian', 'european', 'african',
                               'argentine', 'irish', 'spanish', 'russian', 'southern',
                               'lebanese', 'belgian', 'mediterranean', 'cajun']

    specialize_food_restaurant = ['sandwich', 'salad', 'pizza', 'burger', 'barbecue',
                                   'dessert', 'smoothie', 'seafood', 'steak', 'sushi',
                                   'chocolate', 'pasta', 'alcohol', 'dim-sum', 'bubble-tea',
                                   'tapas', 'soup', 'cheese']

    dietary_based_restaurant = ['vegan', 'vegetarian', 'gluten-free', 'kosher']

    other = ['cafe', 'catering', 'convenience-store', 'other', 'fast', 'breakfast',
             'comfort-food', 'gastropub', 'alcohol-plus-food']

    if store_primary_category in ethnic_based_restaurant:
        return 'Ethnic Based Food'
    elif store_primary_category in specialize_food_restaurant:
        return 'Specialize Food'
    elif store_primary_category in dietary_based_restaurant:
        return 'Dietary Based Food'
    elif store_primary_category in other:
        return 'Others'

# Mengaplikasikan kategori restoran tersebut kedalam fitur 'store_primary_category'
data_prepro.loc[:, 'restaurant_category'] = data_prepro['store_primary_category'].apply(restaurant_category)
```

Mengelompokkan data dalam kolom store_primary_category menjadi berbagai Tipe restoran.

Data Preprocessing

Tampilan data preprocessing pada Collab

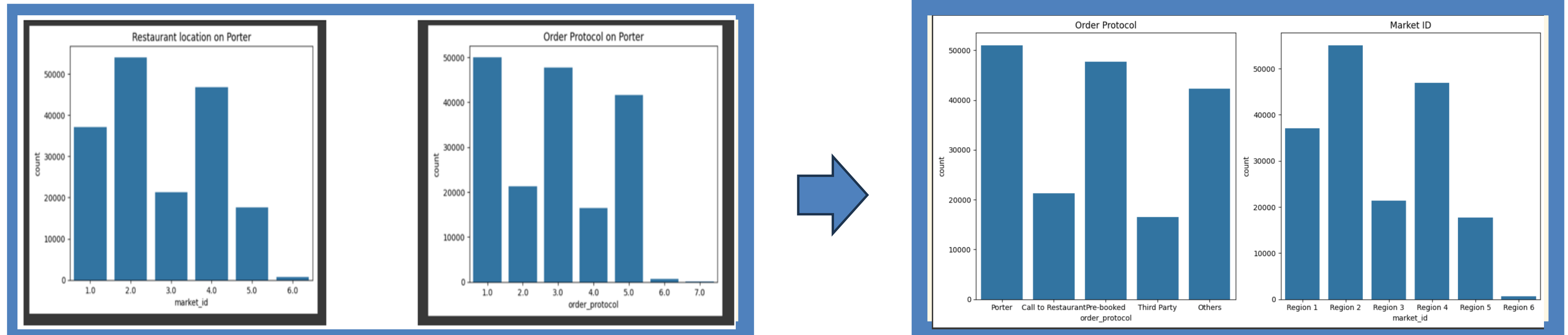
```
[ ] # Memberikan penamaan ulang terhadap values dalam fitur order_protocol & market_id

# Dari sumber data, penamaan valus dalam fitur 'order_protocol' adalah sebagai berikut:
data_prepro.loc[:, 'order_protocol'] = data_prepro.loc[:, 'order_protocol'].replace({
    1.0 : 'Porter',
    2.0 : 'Call to Restaurant',
    3.0 : 'Pre-booked',
    4.0 : 'Third Party',
    5.0 : 'Others',
    6.0 : 'Others',
    7.0 : 'Others'
})

# Dari sumber data, penamaan valus dalam fitur 'market_id' adalah sebagai berikut:
data_prepro.loc[:, 'market_id'] = data_prepro.loc[:, 'market_id'].replace({
    1.0 : 'Region 1',
    2.0 : 'Region 2',
    3.0 : 'Region 3',
    4.0 : 'Region 4',
    5.0 : 'Region 5',
    6.0 : 'Region 6'
})
```

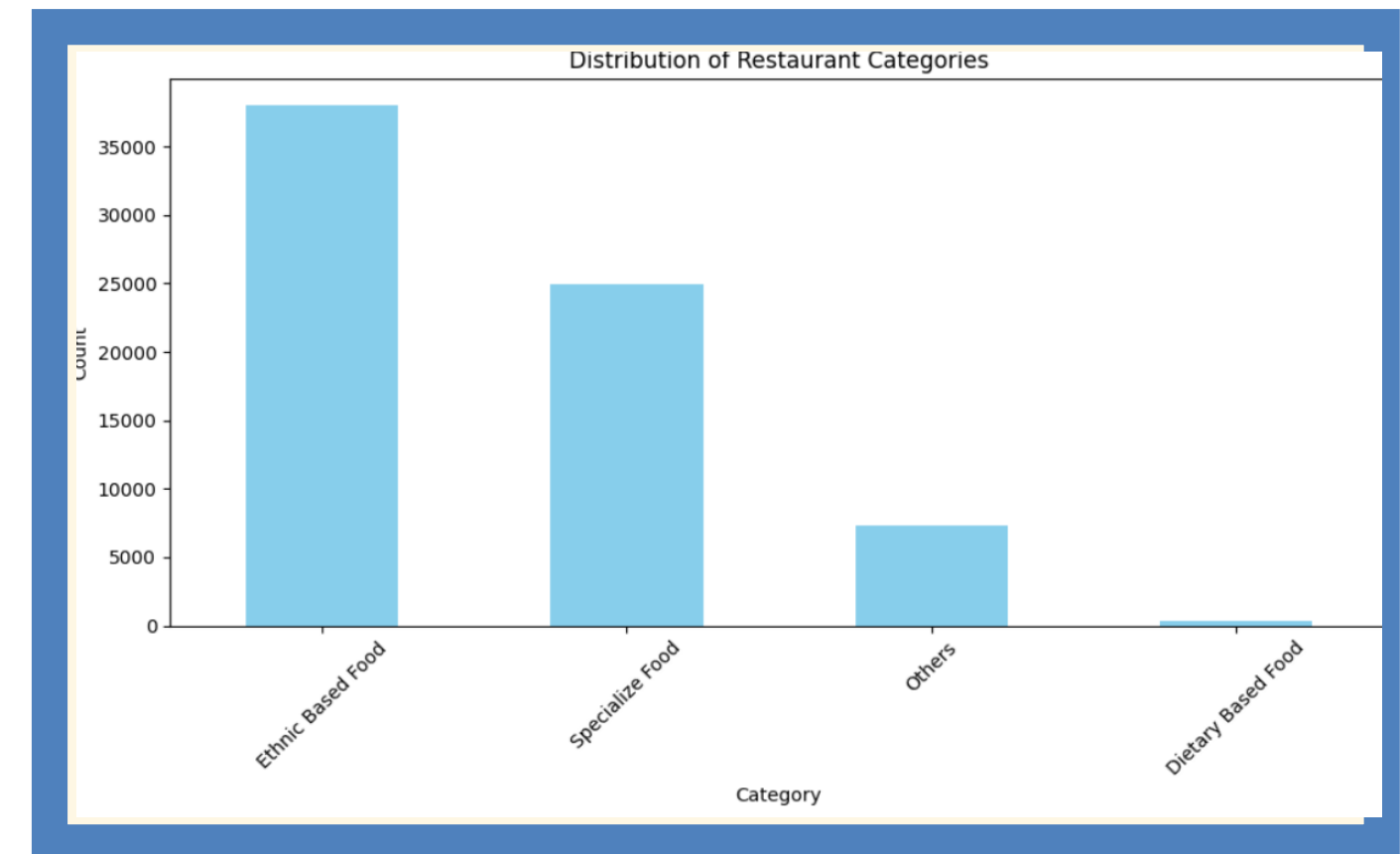
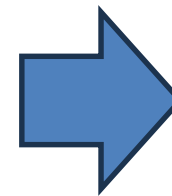
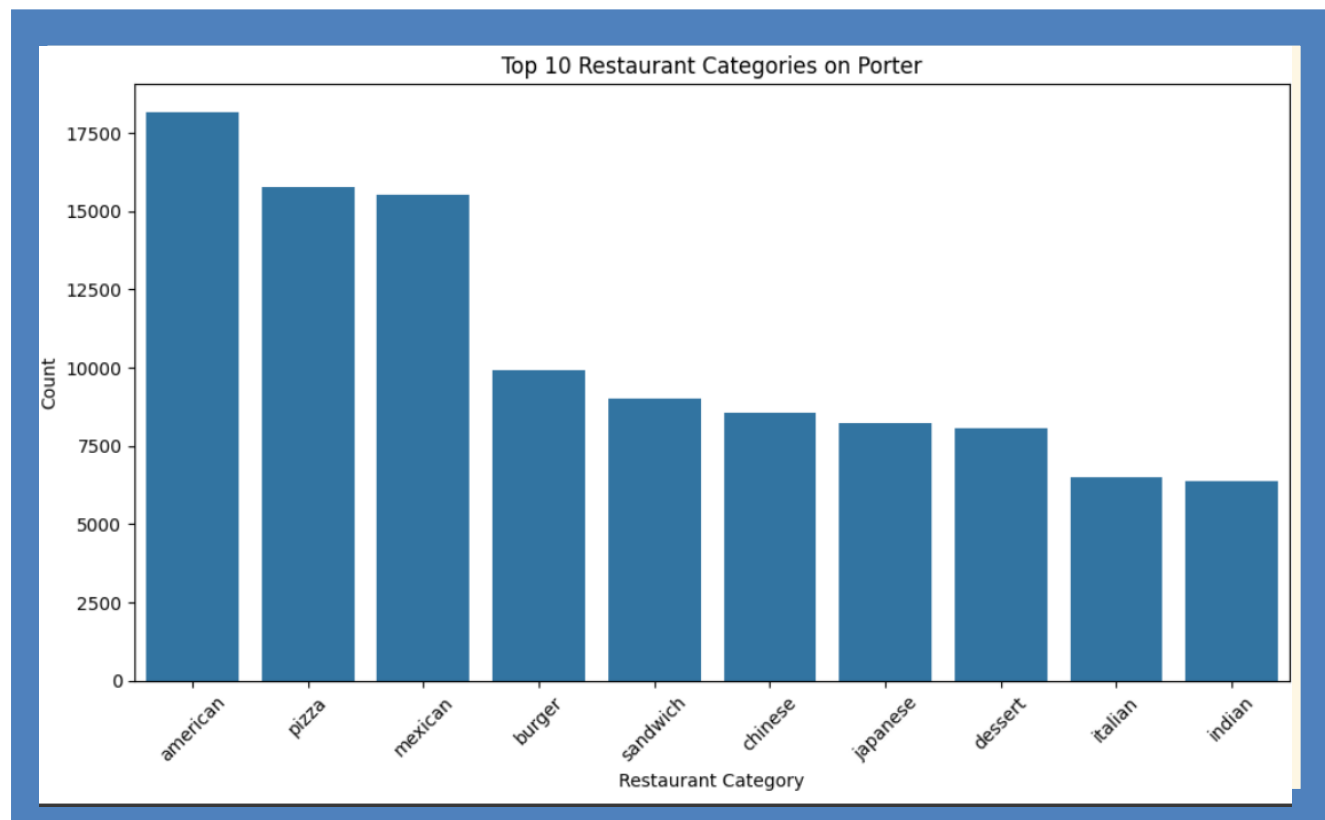
Melakukan penamaan ulang
terhadap values dalam
fitur order_protocol dan market_id

Data Preprocessing



Perbedaan sebelum dan sesudah dilakukan re-value

Data Preprocessing



Perbedaan sebelum dan sesudah dilakukan re-value

Data Preprocessing

```
# Import ML Data Pre-Processing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, MinMaxScaler

# Melakukan Encoding untuk : market_id, store_primary_category, order_protocol
data_prepro_new = data_prepro.copy()

columns = data_prepro_new.select_dtypes(include=['object']).columns.to_list()

label_encoding = LabelEncoder()

# encode the data into a label
for i in columns:
    data_prepro_new[i] = label_encoding.fit_transform(data_prepro_new[i])
```

Proses encoding untuk kolom:

- market_id
- store_primary_category
- order_protocol

Menggunakan label encoding untuk data type kategorikal menjadi kolom numerik sehingga dapat disesuaikan dengan machine learning.

Modeling & Prediction

Dalam tahapan ini, kita melakukan **MinMax Scaling terlebih dahulu pada fitur-fitur (X_{train} dan X_{test})** untuk memastikan bahwa rentang nilai dari setiap fitur adalah antara 0 dan 1. Setelah itu, kita melakukan **pelatihan model-model regresi menggunakan data pelatihan yang sudah di-scaling ($X_{\text{train_scaled}}$ dan y_{train})**.



Modeling & Prediction

Model-model regresi yang digunakan dan definisinya:

Linear Regression

pada linear regression, ditujukan untuk melakukan prediksi pada variabel terikat (y) berdasarkan variabel bebas yang diberikan (x)

SVR (Support Vector Regression)

algoritma supervised learning yang digunakan untuk memprediksi nilai variabel kontinu. Bertujuan menemukan garis keputusan yang paling sesuai (hyperplan bernilai maksimum).

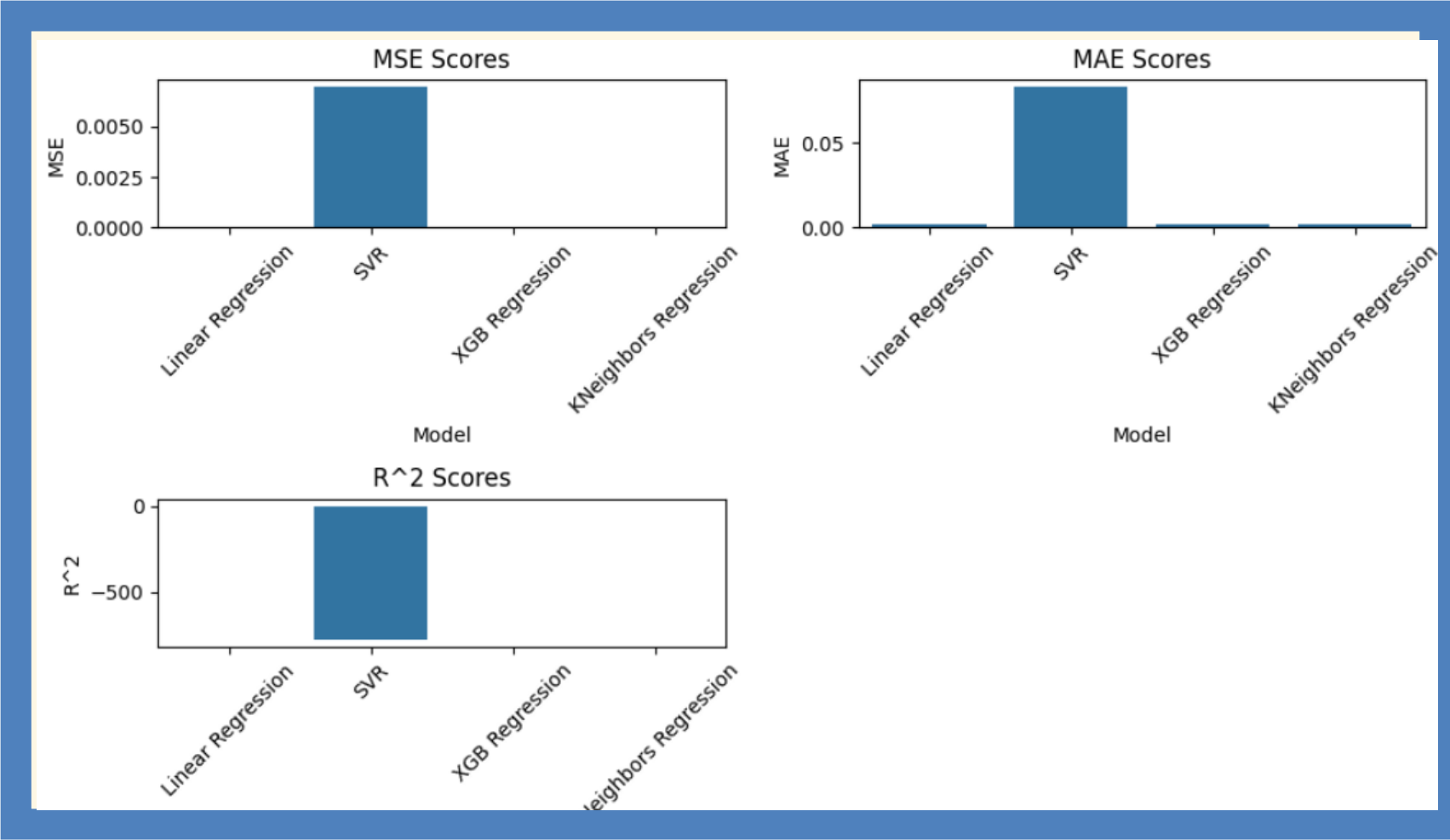
XGBRegressor

jenis algoritma machine learning yang dapat digunakan untuk membuat prediksi pada data numerik berkelanjutan.

KNN (K-Nearest Neighbor Regressor)

sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut

Model Evaluation



Machine Learning	Model Evaluation		
	MSE	MAE	R ² Score
Linear Regression	7.5254002807021515e-06	0.0019662559631801945	0.15993579025755322
SVR	0.006953028183821053	0.08326233454240951	-775.1700253389848
XGB Regression	9.234656659889137e-06	0.0018679308905265566	-0.030869357092692118
KNeighbors Regression	1.1174144691669817e-05	0.002043885450047738	-0.24737537935716492

****fokus pada performa secara keseluruhan, terutama dalam hal MSE yang merupakan metrik yang paling umum digunakan untuk mengevaluasi model regresi, model Linear Regression memiliki MSE yang paling rendah dari semua model yang ada. kita dapat memilih model Linear Regression sebagai model yang akan digunakan berikutnya.**

Model Evaluation

- Mean Squared Error (MSE) dan Mean Absolute Error (MAE):

Metrik ini mengukur seberapa dekat prediksi dengan nilai aktual. Semakin kecil nilainya, semakin baik.

- R^2 Score:

Metrik ini menunjukkan seberapa baik model menjelaskan variabilitas data. Nilai R^2 yang mendekati 1 menunjukkan model yang baik, sementara nilai negatif menunjukkan model yang buruk.

Berdasarkan metrik ini, berikut adalah penilaiannya:

- Linear Regression memiliki MSE dan MAE yang rendah serta R^2 positif, yang menunjukkan performa relatif baik.
- SVR memiliki MSE dan MAE yang tinggi serta R^2 yang sangat negatif, yang menunjukkan performa yang buruk.
- XGBoost Regression memiliki MSE dan MAE yang rendah serta R^2 yang positif, meskipun lebih rendah dari Linear Regression.
- KNeighbors Regression memiliki MSE dan MAE yang lebih tinggi dari Linear Regression dan XGBoost Regression, serta R^2 yang negatif.

Kesimpulan:

Berdasarkan hasil evaluasi di atas, Linear Regression dan XGBoost Regression adalah kandidat terbaik karena memiliki MSE dan MAE yang rendah serta R^2 yang positif. Namun, Linear Regression memiliki R^2 yang lebih tinggi dibandingkan dengan XGBoost Regression. Oleh karena itu, Linear Regression tampaknya menjadi model terbaik untuk digunakan dalam prediksi waktu berdasarkan hasil evaluasi yang kita miliki.

Model Tuning

Machine Learning	Tuning Model
	Hasil Tuning Model
Linear Regression	Parameter terbaik Linear Regression {}
SVR	Parameter terbaik SVR: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
XGB Regression	Parameter terbaik XGB Regression: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 300}
KNeighbors Regression	Parameter terbaik KNeighbor Regression: {'algorithm': 'brute', 'n_neighbors': 7, 'weights': 'uniform'}

****Tuning model** adalah proses eksperimental untuk menemukan nilai optimal dari hiperparameter untuk memaksimalkan kinerja model. Hasil tuning masing-masing model, membantu kita menentukan model terbaik yang akan dipilih.

Model Deployment (Streamlit)

✕


Menu

Home


▼

Share ☆ ↻ ⋮

Porter Delivery Time Estimation



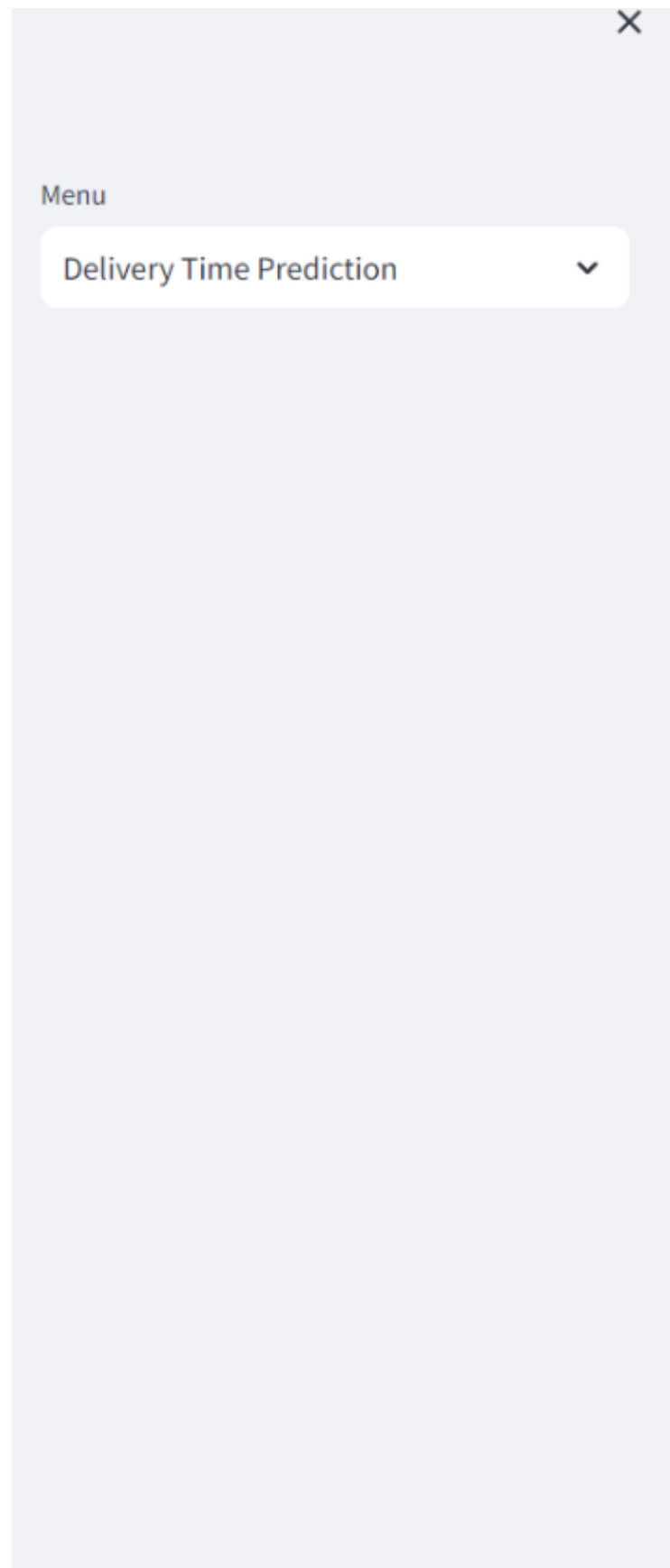
Delivery Hai?
Ho Jayega!°

 MENU

[▶ About Porter](#)

[▶ About Us](#)

Model Deployment (Streamlit)



Your Food Delivery Time

Please input these information for predict

What do I need to input?

How many food did you order?

1

How many types of food did you order?

1

How much does your food cost? (in Rupee)

1

Where the restaurant located at?

Region 1

How did you order the Food?

Through Porter

What kind of food the restaurant sell?

Ethnic Based Food

Your Foods Are Arriving in

Predict Delivery Time

Model Deployment (Streamlit)

How many food did you order?

4 - +

How many types of food did you order?

4 - +

How much does your food cost? (in Rupee)

4444 - +

Where the restaurant located at?

Region 2 ▼

How did you order the Food?

Call to Restaurant ▼

What kind of food the restaurant sell?

Others ▼

Your Foods Are Arriving in

Predict Delivery Time

Estimated Delivery Time: [86.19688216] minutes

How many food did you order?

50 - +

How many types of food did you order?

8 - +

How much does your food cost? (in Rupee)

17865 - +

Where the restaurant located at?

Region 6 ▼

How did you order the Food?

Pre-booked ▼

What kind of food the restaurant sell?

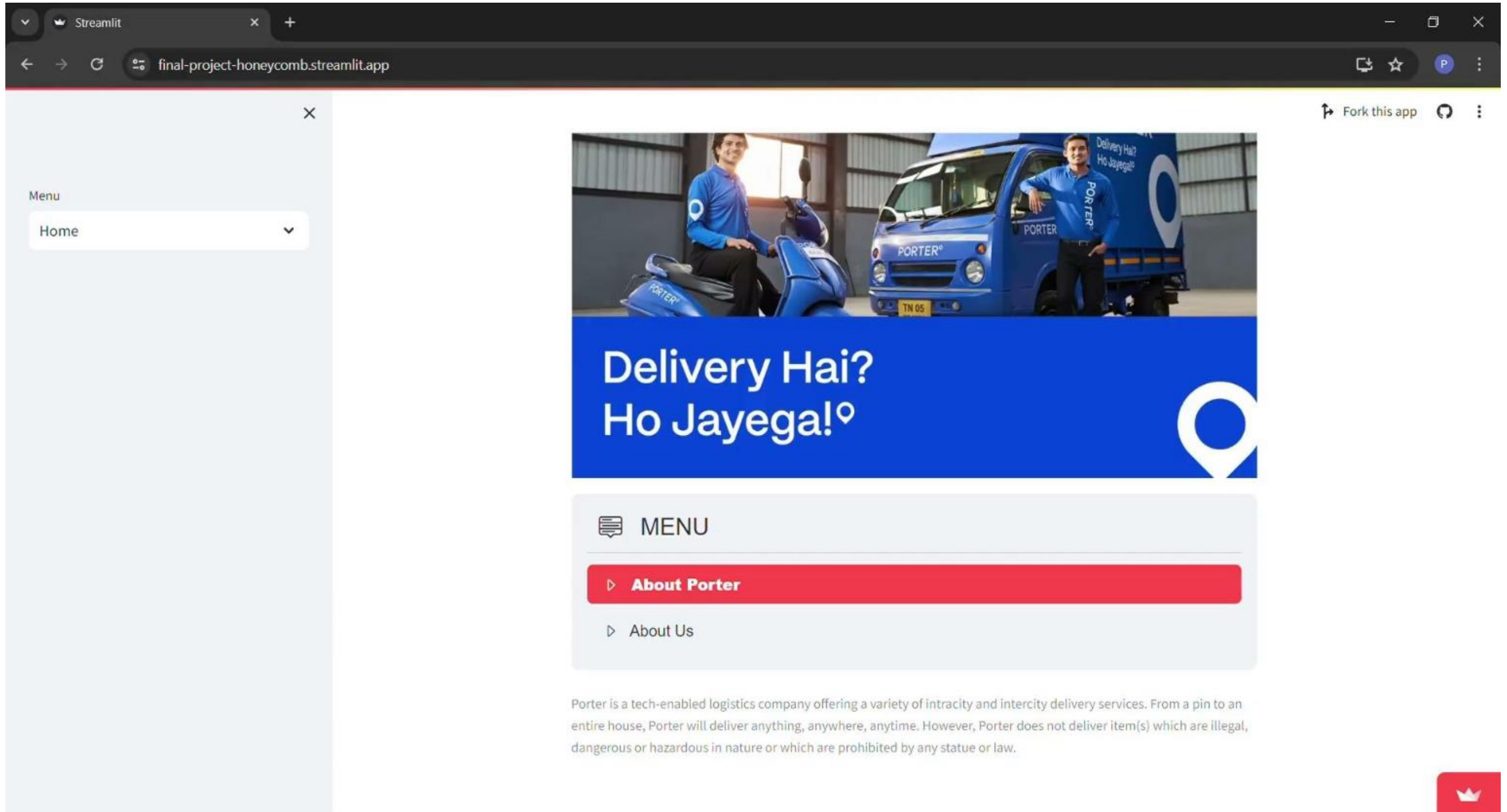
Specialized Food ▼

Your Foods Are Arriving in

Predict Delivery Time

Estimated Delivery Time: [317.31148688] minutes

Model Deployment (Streamlit)



Link

Google Drive

<https://drive.google.com/drive/folders/1aeCh154Veg2-9bNYiLoLeFcokb8P4isl?usp=sharing>

Streamlit

<https://final-project-honeycomb.streamlit.app/>

Google Colaboratory

[https://colab.research.google.com/drive/1N85nsxxAqvgG41sa94qwuhF9M29bwYN-
?usp=sharing](https://colab.research.google.com/drive/1N85nsxxAqvgG41sa94qwuhF9M29bwYN-?usp=sharing)



Thank you for listening!

Don't hesitate to ask any questions!

