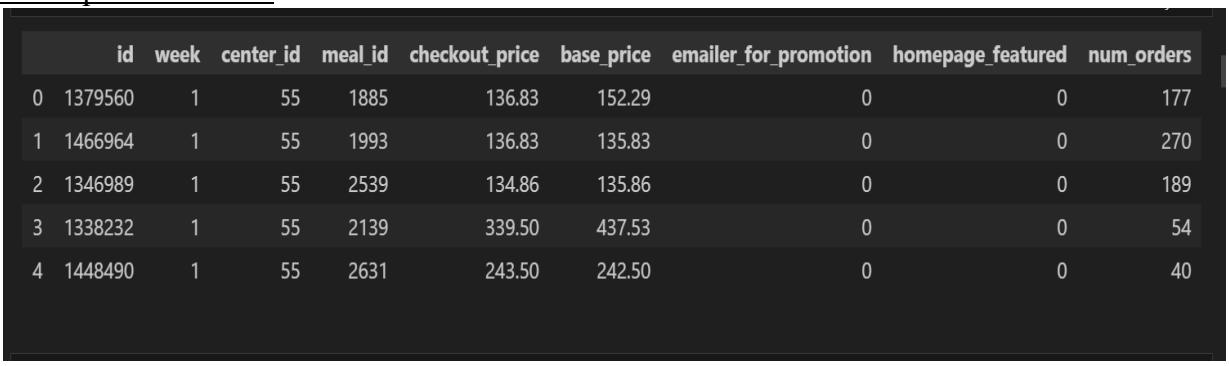
Data Collection and Preprocessing Phase

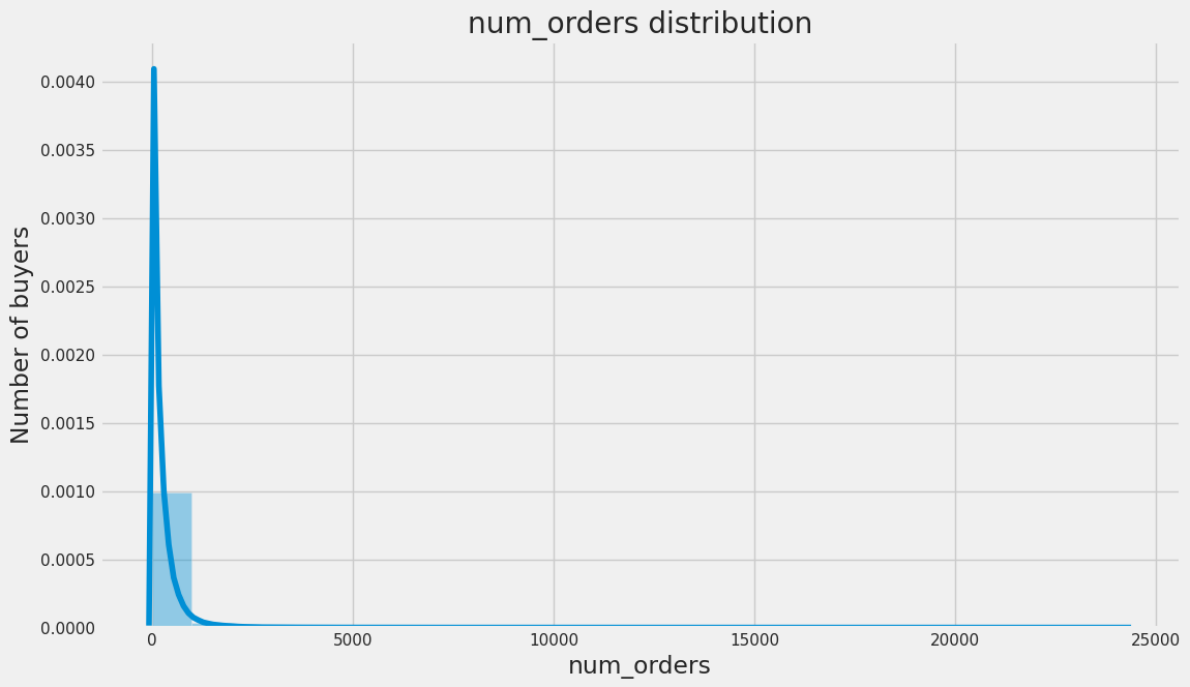
Date	4th July 2024
Team ID	739754
Project Title	Food demand forecasting for food delivery company
Maximum Marks	6 Marks

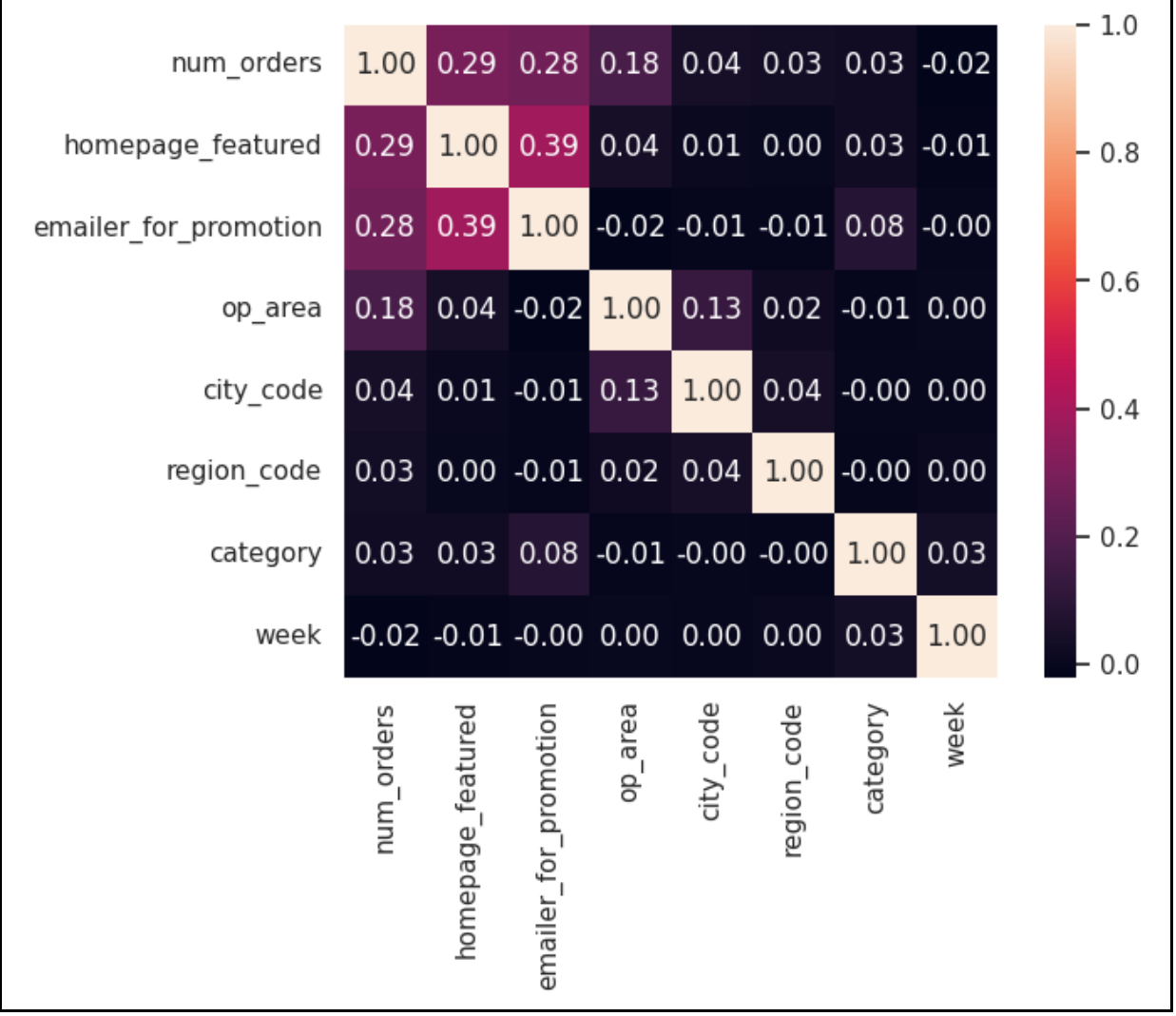
Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description
Data Overview	Dimension: 32574 rows × 10 columns Descriptive statistics: 
Univariate Analysis	It is the simplest form of data analysis where the data being analyzed contains only o

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders
0	1379560	1	55	1885	136.83	152.29	0	0	177
1	1466964	1	55	1993	136.83	135.83	0	0	270
2	1346989	1	55	2539	134.86	135.86	0	0	189
3	1338232	1	55	2139	339.50	437.53	0	0	54
4	1448490	1	55	2631	243.50	242.50	0	0	40

Bivariate Analysis	

Multivariate Analysis	

Outliers and Anomalies	-

Data Preprocessing Code Screenshots

Loading Data

```
# Convert the encoded 'category' column to numeric type
trainfinal['category'] = trainfinal['category'].astype(int)
```

```
trainfinal.head()
```

	id	week	city_code	region_code	center_type	op_area	category	cuisine	checkout_price	base_price	emailer_f
0	1379560	1	647	56	2	2.0	0	Thai	136.83	152.29	
1	1018704	2	647	56	2	2.0	0	Thai	135.83	152.29	
2	1196273	3	647	56	2	2.0	0	Thai	132.92	133.92	
3	1116527	4	647	56	2	2.0	0	Thai	135.86	134.86	
4	1343872	5	647	56	2	2.0	0	Thai	146.50	147.50	

Handling Missing Data

```
trainfinal = pd.merge(train, meal_info, on="meal_id",how="outer")
trainfinal = pd.merge(trainfinal, center_info, on="center_id",how="outer")
trainfinal.head()
```

Data Transformation

```
from sklearn.preprocessing import LabelEncoder
lb1 = LabelEncoder()
trainfinal['center_type'] = lb1.fit_transform(trainfinal['center_type'])
lb2 = LabelEncoder()
trainfinal['category'] = lb2.fit_transform(trainfinal['category'])
```

```
# Convert the encoded 'category' column to numeric type
trainfinal['category'] = trainfinal['category'].astype(int)
```

Feature Engineering	`features = columns.drop(['num_orders'])`
	`trainfinal3 = trainfinal[features]`
	`x = trainfinal3.values`
	`y = trainfinal['num_orders'].values`

Save Processed Data	`import pickle`
	`pickle.dump(DT,open('fdemand.pkl','wb'))`
	
	`testfinal = pd.merge(test, meal_info, on="meal_id",how="outer")`
	`testfinal = pd.merge(testfinal, center_info, on="center_id",how="outer")`
	`testfinal = testfinal.drop(['center_id','meal_id'], axis=1)`