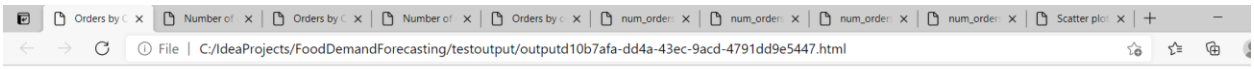
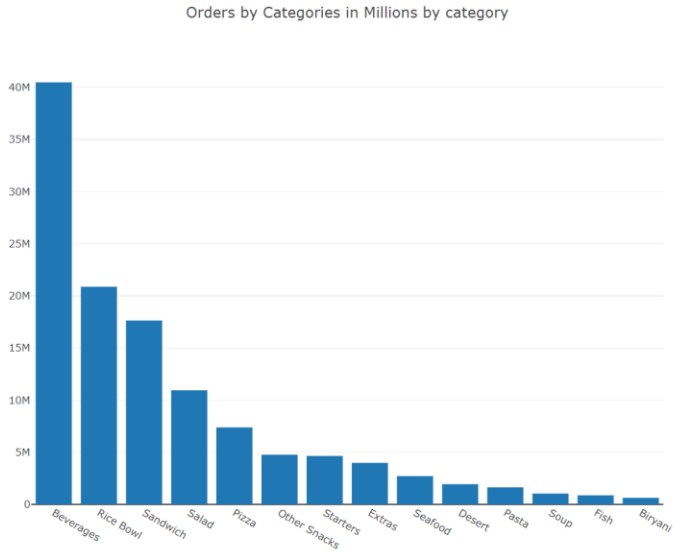
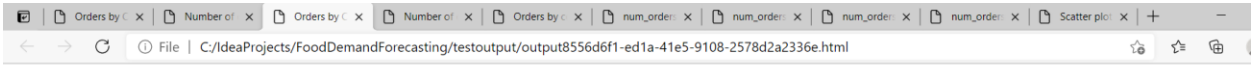
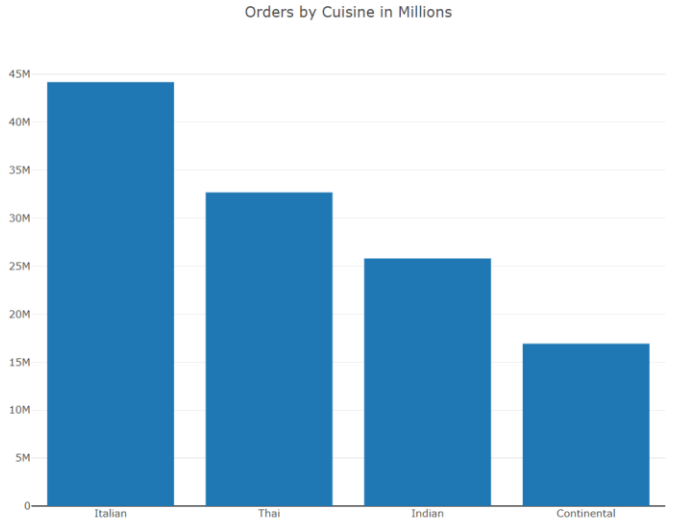
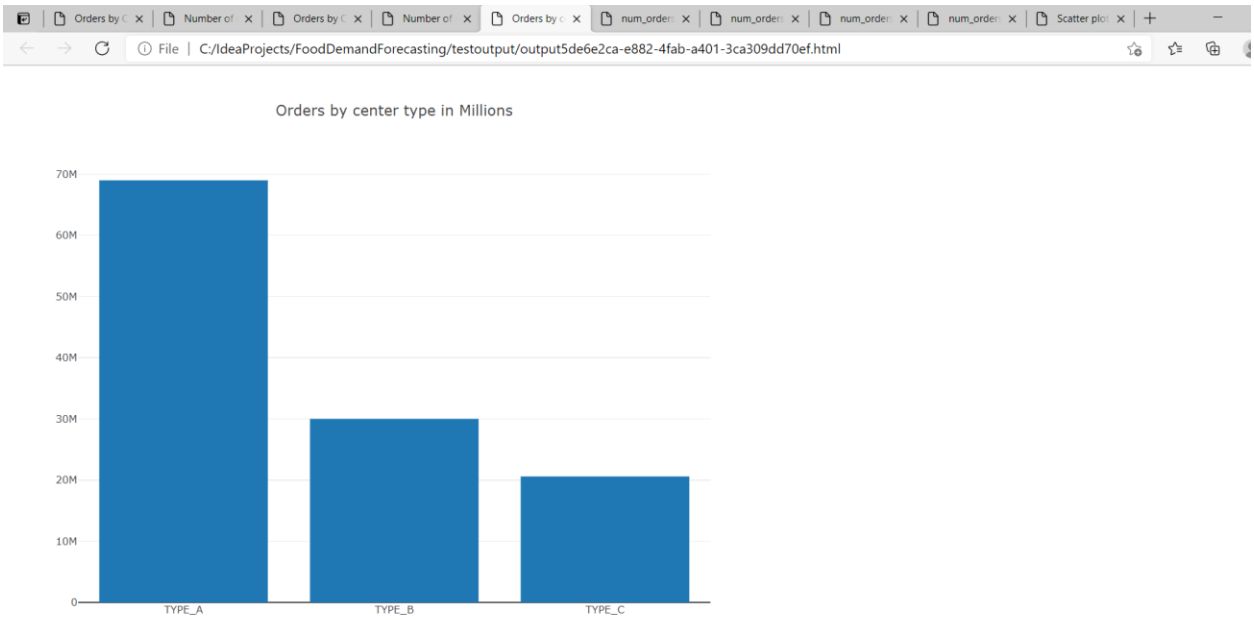
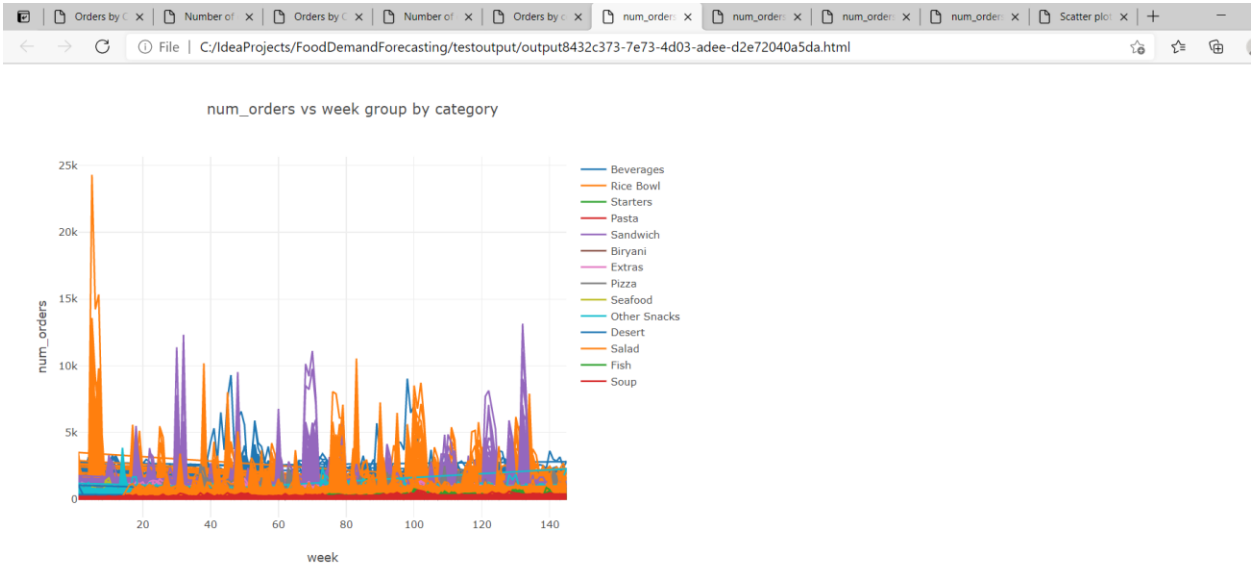

File | C:/IdeaProjects/FoodDemandForecasting/testoutput/outputd10b7afa-dd4a-43ec-9acd-4791dd9e5447.html

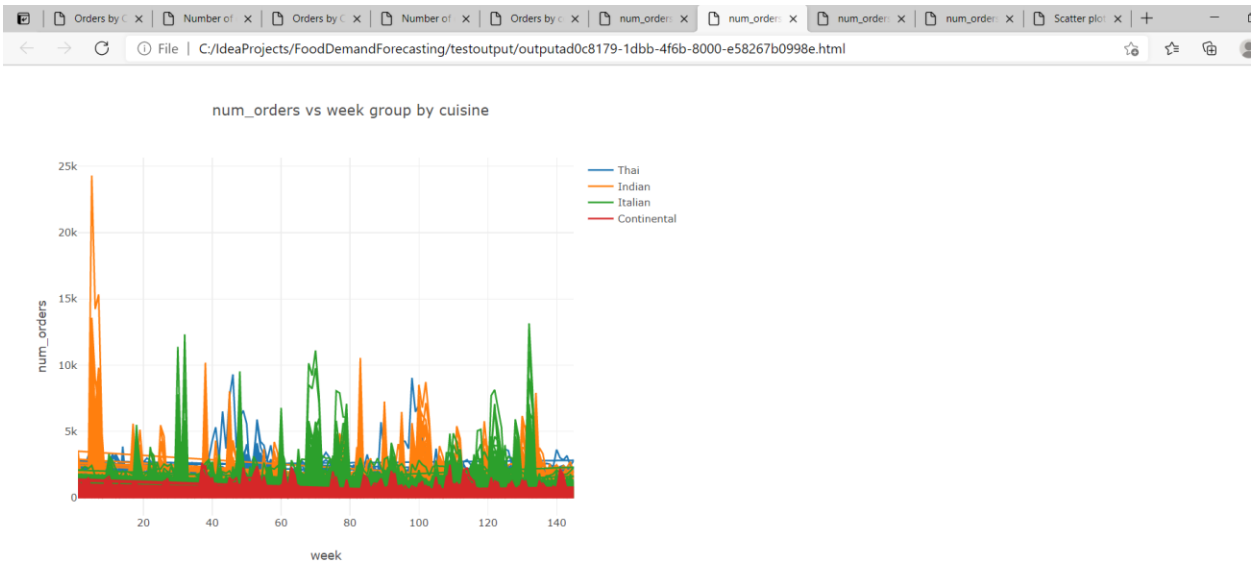

Orders by Categories in Millions by category
40M
35M
30M
25M
20M
15M
10M
5M
0
Beverages
Rice Bowl
Sandwich
Salad
Pizza
Other Snacks
Starters
Extras
Seafood
Desert
Pasta
Soup
Fish
Biryani


File | C:/IdeaProjects/FoodDemandForecasting/testoutput/output8556d6f1-ed1a-41e5-9108-2578d2a2336e.html

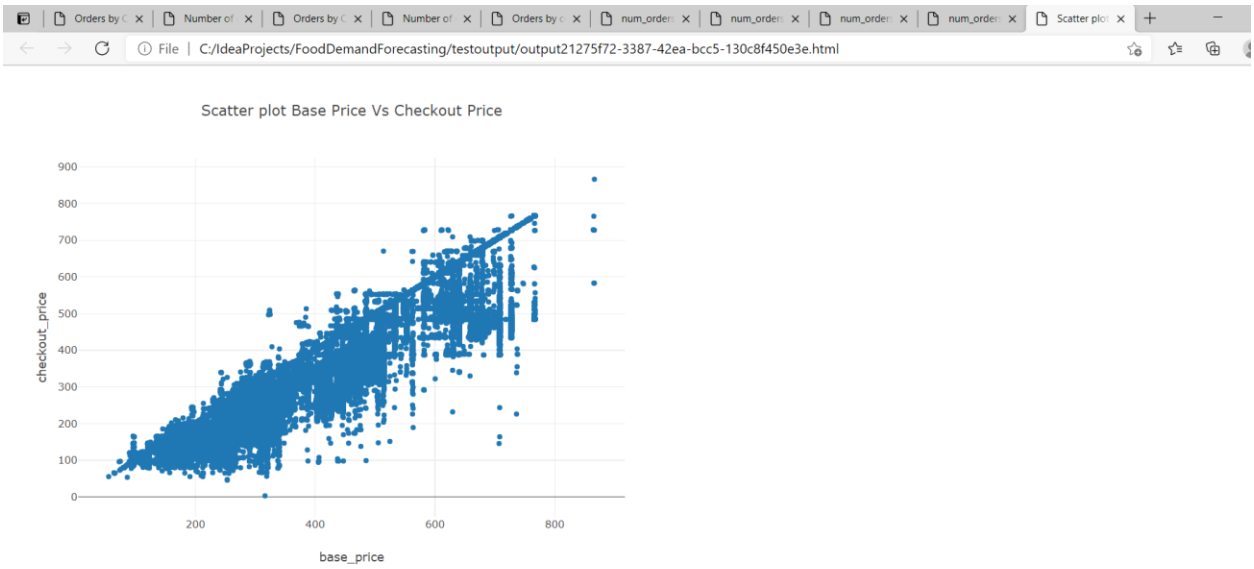

Orders by Cuisine in Millions
45M
40M
35M
30M
25M
20M
15M
10M
5M
0
Italian
Thai
Indian
Continental

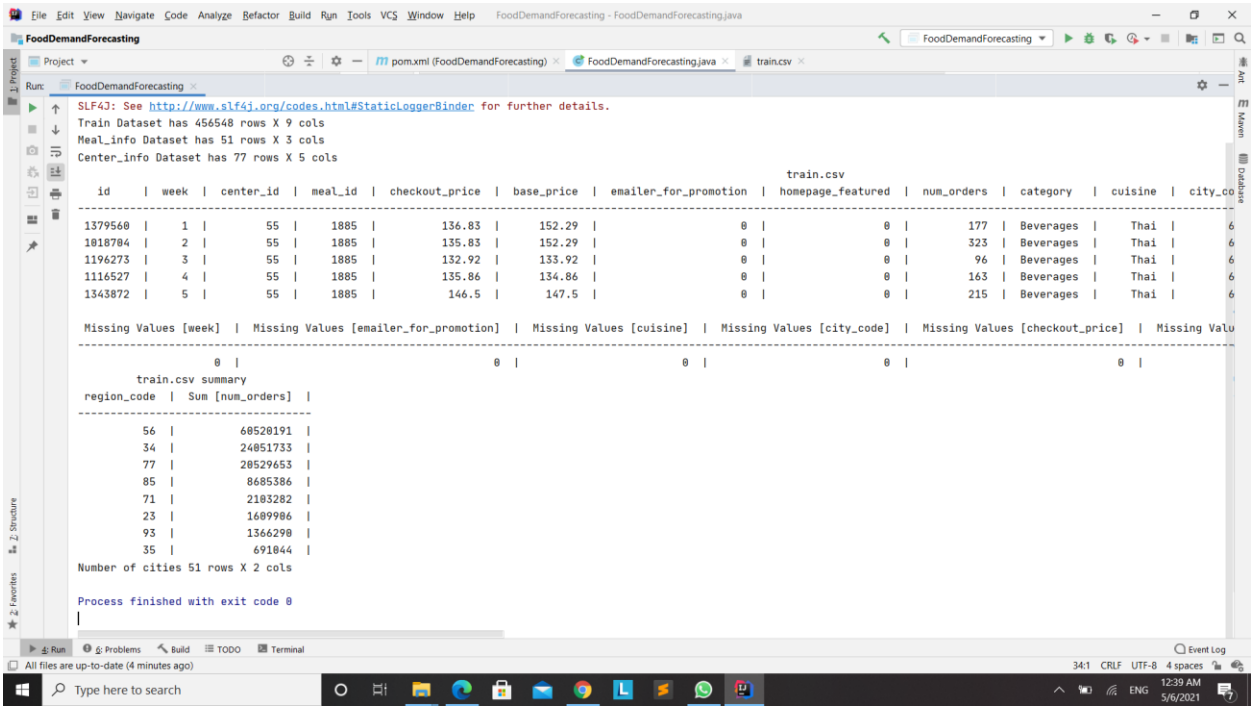

File | C:/IdeaProjects/FoodDemandForecasting/testoutput/output5de6e2ca-e882-4fab-a401-3ca309dd70ef.html
Orders by center type in Millions
70M
60M
50M
40M
30M
20M
10M
0
TYPE_A
TYPE_B
TYPE_C

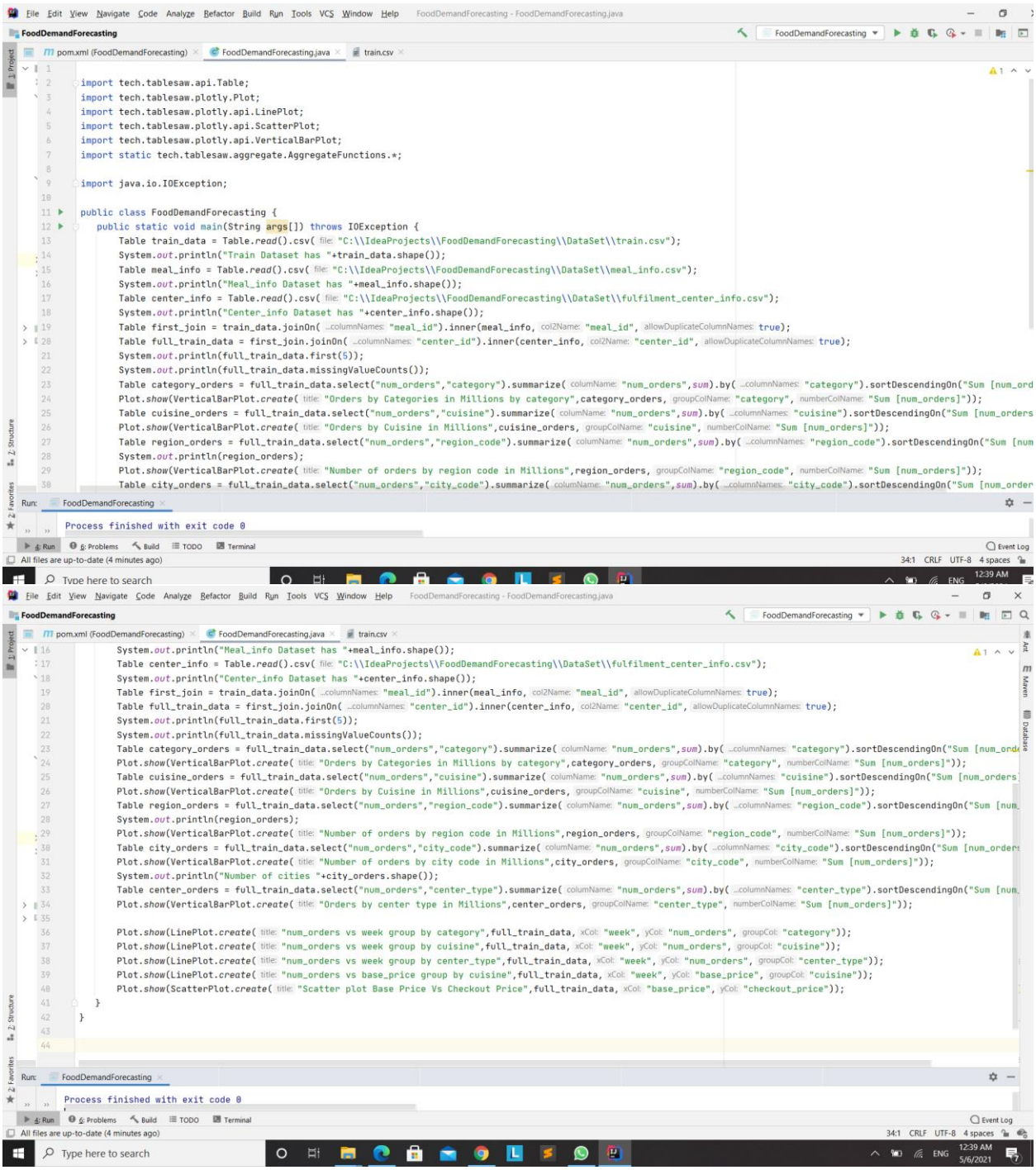

File | C:/IdeaProjects/FoodDemandForecasting/testoutput/output8432c373-7e73-4d03-adee-d2e72040a5da.html
num_orders vs week group by category
num_orders
25k
20k
15k
10k
5k
0
20
40
60
80
100
120
140
week
Beverages
Rice Bowl
Starters
Pasta
Sandwich
Biryani
Extras
Pizza
Seafood
Other Snacks
Desert
Salad
Fish
Soup


File | C:/IdeaProjects/FoodDemandForecasting/testoutput/outputad0c8179-1dbb-4f6b-8000-e58267b0998e.html
num_orders vs week group by cuisine
num_orders
25k
20k
15k
10k
5k
0
20
40
60
80
100
120
140
week
Thai
Indian
Italian
Continental


File | C:/IdeaProjects/FoodDemandForecasting/testoutput/output0ff04d32-5054-4e07-bdce-111104105735.html
num_orders vs base_price group by cuisine
base_price
900
800
700
600
500
400
300
200
100
20
40
60
80
100
120
140
week
Thai
Indian
Italian
Continental


Scatter plot Base Price Vs Checkout Price
checkout_price
base_price
900
800
700
600
500
400
300
200
100
0
200
400
600
800


FoodDemandForecasting - FoodDemandForecasting.java
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Train Dataset has 456548 rows X 9 cols
Meal_info Dataset has 51 rows X 3 cols
Center_info Dataset has 77 rows X 5 cols
train.csv
id | week | center_id | meal_id | checkout_price | base_price | emailer_for_promotion | homepage_featured | num_orders | category | cuisine | city_co
1379560 | 1 | 55 | 1885 | 136.83 | 152.29 | 0 | 0 | 177 | Beverages | Thai | 6
1018704 | 2 | 55 | 1885 | 135.83 | 152.29 | 0 | 0 | 323 | Beverages | Thai | 6
1196273 | 3 | 55 | 1885 | 132.92 | 133.92 | 0 | 0 | 96 | Beverages | Thai | 6
1116527 | 4 | 55 | 1885 | 135.86 | 134.86 | 0 | 0 | 163 | Beverages | Thai | 6
1343872 | 5 | 55 | 1885 | 146.5 | 147.5 | 0 | 0 | 215 | Beverages | Thai | 6
Missing Values [week] | Missing Values [emailer_for_promotion] | Missing Values [cuisine] | Missing Values [city_code] | Missing Values [checkout_price] | Missing Valu
0 | 0 | 0 | 0 | 0 |
train.csv summary
region_code | Sum [num_orders] |
56 | 60520191 |
34 | 24051733 |
77 | 20529653 |
85 | 8685386 |
71 | 2103282 |
23 | 1689986 |
93 | 1366298 |
35 | 691044 |
Number of cities 51 rows X 2 cols
Process finished with exit code 0



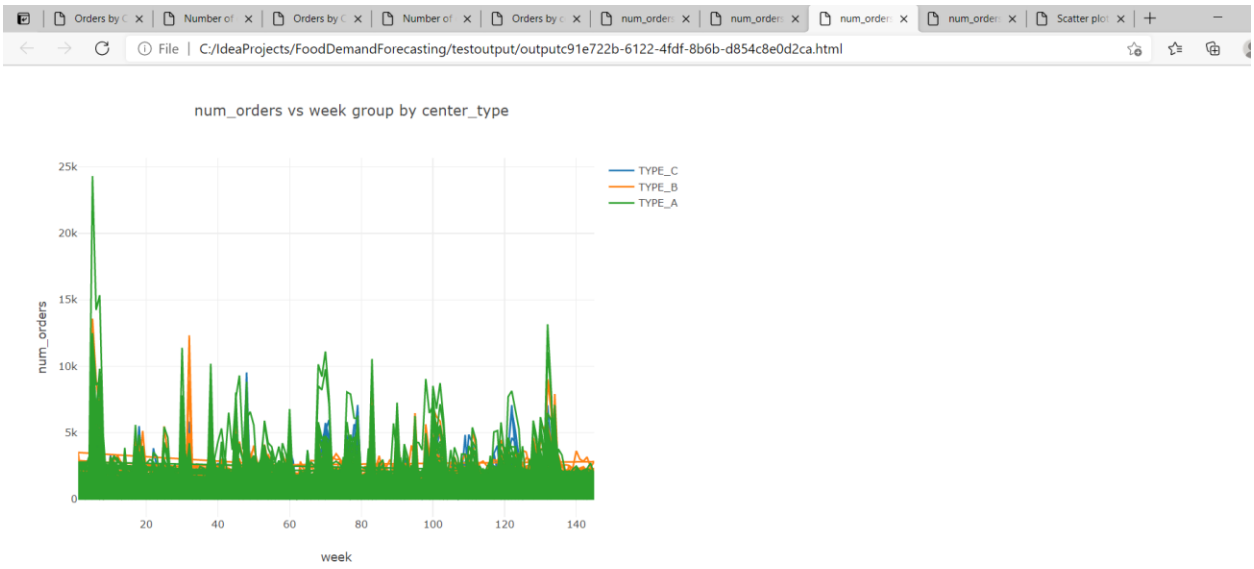
```
import tech.tablesaw.api.Table;
import tech.tablesaw.plotly.Plot;
import tech.tablesaw.plotly.api.LinePlot;
import tech.tablesaw.plotly.api.ScatterPlot;
import tech.tablesaw.plotly.api.VerticalBarPlot;
import static tech.tablesaw.aggregate.AggregateFunctions.*;

import java.io.IOException;

public class FoodDemandForecasting {
    public static void main(String args[]) throws IOException {
        Table train_data = Table.read().csv( file: "C:\\IdeaProjects\\FoodDemandForecasting\\DataSet\\train.csv");
        System.out.println("Train Dataset has "+train_data.shape());
        Table meal_info = Table.read().csv( file: "C:\\IdeaProjects\\FoodDemandForecasting\\DataSet\\meal_info.csv");
        System.out.println("Meal_info Dataset has "+meal_info.shape());
        Table center_info = Table.read().csv( file: "C:\\IdeaProjects\\FoodDemandForecasting\\DataSet\\fulfilment_center_info.csv");
        System.out.println("Center_info Dataset has "+center_info.shape());
        Table first_join = train_data.joinOn( ...columnNames: "meal_id").inner(meal_info, col2Name: "meal_id", allowDuplicateColumnNames: true);
        Table full_train_data = first_join.joinOn( ...columnNames: "center_id").inner(center_info, col2Name: "center_id", allowDuplicateColumnNames: true);
        System.out.println(full_train_data.first(5));
        System.out.println(full_train_data.missingValueCounts());
        Table category_orders = full_train_data.select("num_orders","category").summarize( columnName: "num_orders",sum).by( ...columnNames: "category").sortDescendingOn("Sum [num_ord
        Plot.show(VerticalBarPlot.create( title: "Orders by Categories in Millions by category",category_orders, groupColName: "category", numberColName: "Sum [num_orders]"));
        Table cuisine_orders = full_train_data.select("num_orders","cuisine").summarize( columnName: "num_orders",sum).by( ...columnNames: "cuisine").sortDescendingOn("Sum [num_orders
        Plot.show(VerticalBarPlot.create( title: "Orders by Cuisine in Millions",cuisine_orders, groupColName: "cuisine", numberColName: "Sum [num_orders]"));
        Table region_orders = full_train_data.select("num_orders","region_code").summarize( columnName: "num_orders",sum).by( ...columnNames: "region_code").sortDescendingOn("Sum [num
        System.out.println(region_orders);
        Plot.show(VerticalBarPlot.create( title: "Number of orders by region code in Millions",region_orders, groupColName: "region_code", numberColName: "Sum [num_orders]"));
        Table city_orders = full_train_data.select("num_orders","city_code").summarize( columnName: "num_orders",sum).by( ...columnNames: "city_code").sortDescendingOn("Sum [num_order
        Plot.show(VerticalBarPlot.create( title: "Number of orders by city code in Millions",city_orders, groupColName: "city_code", numberColName: "Sum [num_orders]"));
        System.out.println("Number of cities "+city_orders.shape());
        Table center_orders = full_train_data.select("num_orders","center_type").summarize( columnName: "num_orders",sum).by( ...columnNames: "center_type").sortDescendingOn("Sum [num_
        Plot.show(VerticalBarPlot.create( title: "Orders by center type in Millions",center_orders, groupColName: "center_type", numberColName: "Sum [num_orders]"));

        Plot.show(LinePlot.create( title: "num_orders vs week group by category",full_train_data, xCol: "week", yCol: "num_orders", groupCol: "category"));
        Plot.show(LinePlot.create( title: "num_orders vs week group by cuisine",full_train_data, xCol: "week", yCol: "num_orders", groupCol: "cuisine"));
        Plot.show(LinePlot.create( title: "num_orders vs week group by center_type",full_train_data, xCol: "week", yCol: "num_orders", groupCol: "center_type"));
        Plot.show(LinePlot.create( title: "num_orders vs base_price group by cuisine",full_train_data, xCol: "week", yCol: "base_price", groupCol: "cuisine"));
        Plot.show(ScatterPlot.create( title: "Scatter plot Base Price Vs Checkout Price",full_train_data, xCol: "base_price", yCol: "checkout_price"));
    }
}
```

```
Process finished with exit code 0
```


File | C:/IdeaProjects/FoodDemandForecasting/testoutput/outputc91e722b-6122-4fdf-8b6b-d854c8e0d2ca.html
num_orders vs week group by center_type
TYPE_C
TYPE_B
TYPE_A
num_orders
25k
20k
15k
10k
5k
0
20
40
60
80
100
120
140
week