FOOD DEMAND FORECASTING FOR FOOD DELIEVERY COMPANY

1.INTRODUCTION

1.1Overview:

It is a meal delivery company which operates in multiple cities. They have various fulfillment centers in these cities for dispatching meal orders to their customers. The client wants you to help these centers with demand forecasting for upcoming weeks so that these centers will plan the stock of raw materials accordingly.

1.2Purpose:

Demand forecasting is a key component to every growing online business. Without proper demand forecasting processes in place,it can be nearly impossible to have the right amount of stock on hand at any given time. A food delivery service has to dealwith a lot of perishable raw materials which makes it all the more important for such a company to accurately forecast daily and weekly demand.
Too much invertory in the warehouse means more risk of wastage,and not enough could lead to out-of-stocks - and push customers to seek solutions from your competitors. In this challenge, get a taste of demand forecasting challenge using a real datasets.

2.LITERATURE SURVEY

2.1Existing problem:

Your client is a meal delivery company which operates in multiple cities.They have various fulfillment centers in these cities for dispatching meal orders to their customers. The client wants you to help these centers with demand forecasting for upcoming weeks so that these centers will plan the stock of raw materials accordingly. The replenishment of majority of raw materials is done on weekly basis and since the raw material is perishable,the procurement planning is of utmost importance.Secondly, staffing of the centers is also one area wherein accurate demand forecasts are really helpful.Given the following information,the task is to predict the demand for the next 10 weeks(Weeks:146-155) for the center-meal combinations in the test set:

- **Historical data of demand for a product-center combination(Weeks:1 to145)**
- **Product(Meal) features such as category,sub-category,current price and discount**
- **Information for fulfillment center like center area, city information etc.**

2.2Proposed solution:

Data Dictionary:

1. Weekly demand data(train.csv): Contains the historicaal data for all centers,test.csv contains all the following features except the target variable

Variable	Definition
id	Unique ID
week	week no
center_id	Unique ID for fulfillment center
meal_id	Unique ID for meal
checkout_price	Final price including discount,taxes&delivery charges
base_price	Base price of the meal
emailer_for_promotion	Emailer sent for promotion of meal
homepage_featured	Meal featured at homepage
num_orders	(Target) Orders Count

2. fulfillment _center_info.csv: Contains information for each fulfillment center

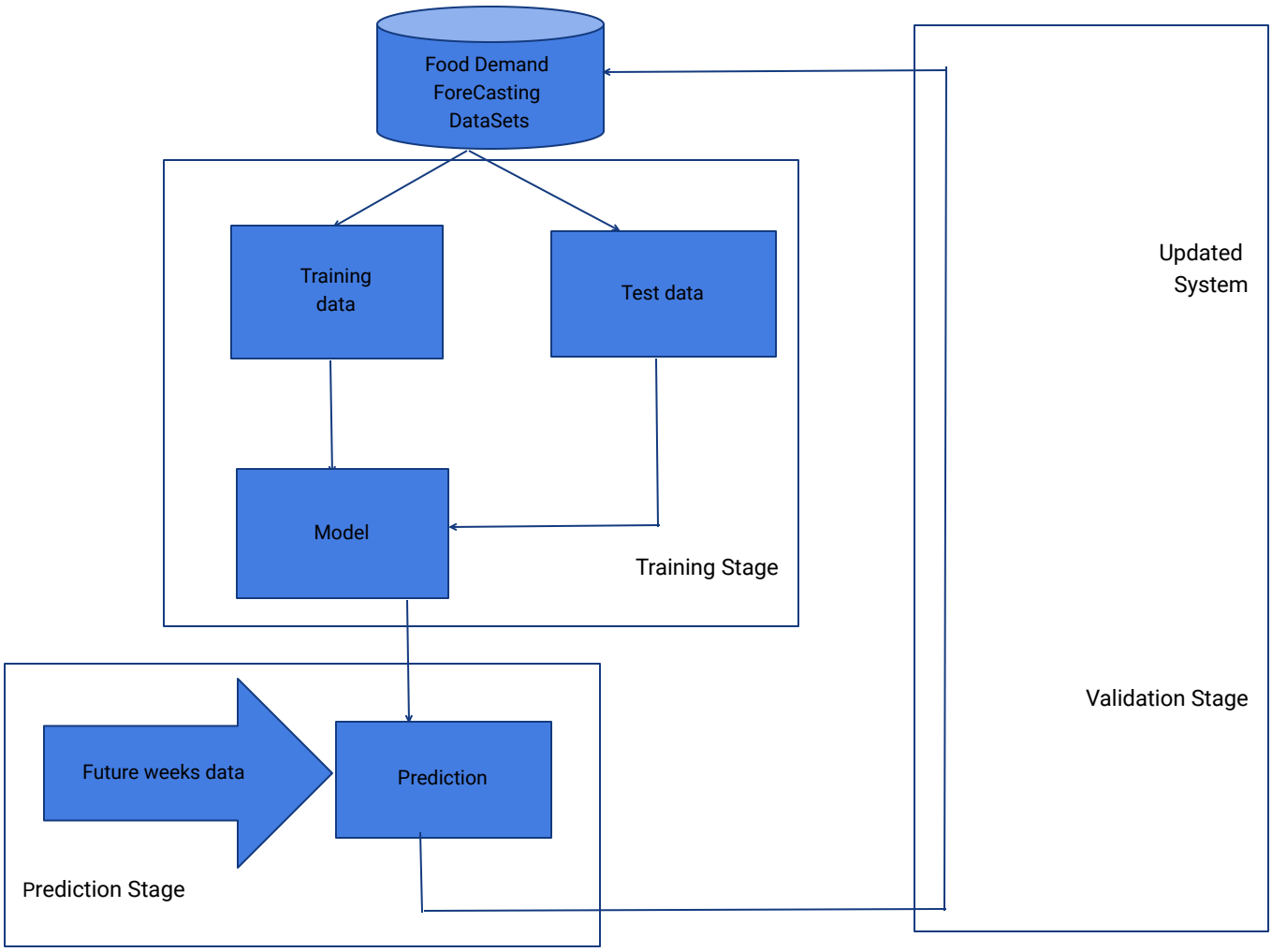
Variable	Description
center_id	Unique ID for fulfillment center
city_code	Unique code for city
region_code	Unique code for region
center _type	Anonymized center type
op_area	Area of operation (in km^2)

3. meal_info.csv: Contains information for each meal being served

Variable	Definition
meal_id	Unique ID for the meal
category	Type of meal (beverages/snacks/soups.....)
cuisine	Meal cuisine(Indian/Italian/.....)

3.THEORETICAL ANALYSIS

3.1Block diagram:



In this we have :

- food demand forecasting datasets
- training data
- test data
- model
- future weeks data
- prediction

3.2Hardware/Software designing

i) Hardware Requirement:

- windows 7 and above(64bit)
- RAM : 4GB
- Processor : Minimum pentinum 2 266MHz processor
- Browsers : Chrome

ii) Software Requirements:

- Weka
- Eclipse IDE
- Java JDK 10

4.EXPERIMENTAL INVESTIGATIONS

After analysing the correlation between every continuous variable with each other, below are the findings:

● The checkout price and base price has high positive correlation with each other

● Both prices also have negative correlation with number of orders, which make sense

● The discount, which was derived from both prices, has low positive correlation with number of order

● Weekly Trend

It was found that week 62 had lowest orders while week 5 and week 48 had highest orders. ○ After further analysis, there was hugh difference in the promotional activity by emails for week 62 compared to week 48 and week 5.

● Yearly Trend ○ Data is not sufficient to analyse the yearly trend in number of orders.

● Monthly Trend ○ It was found that month 2 had highest orders and month 9 had the lowest orders.

● Week in Month Trend ○ It was found that start and end of the month has highest orders as compared to the mid of month. Trend in number of orders with respect to center

Trend in number of orders were analysed with respect to center's metadata like city code, region code, center type and operation area.

Below are the findings from the same:

● Centers with center type TYPE_B get more orders than centers with center type TYPE_A and TYPE_C

● Centers with region code 35 has lowest orders

● There are fluctuations in the number of orders for almost all regions

● Orders increased with increase in operating areas

Trend in number of orders were analysed with respect to meal's metadata like cuisine and category.

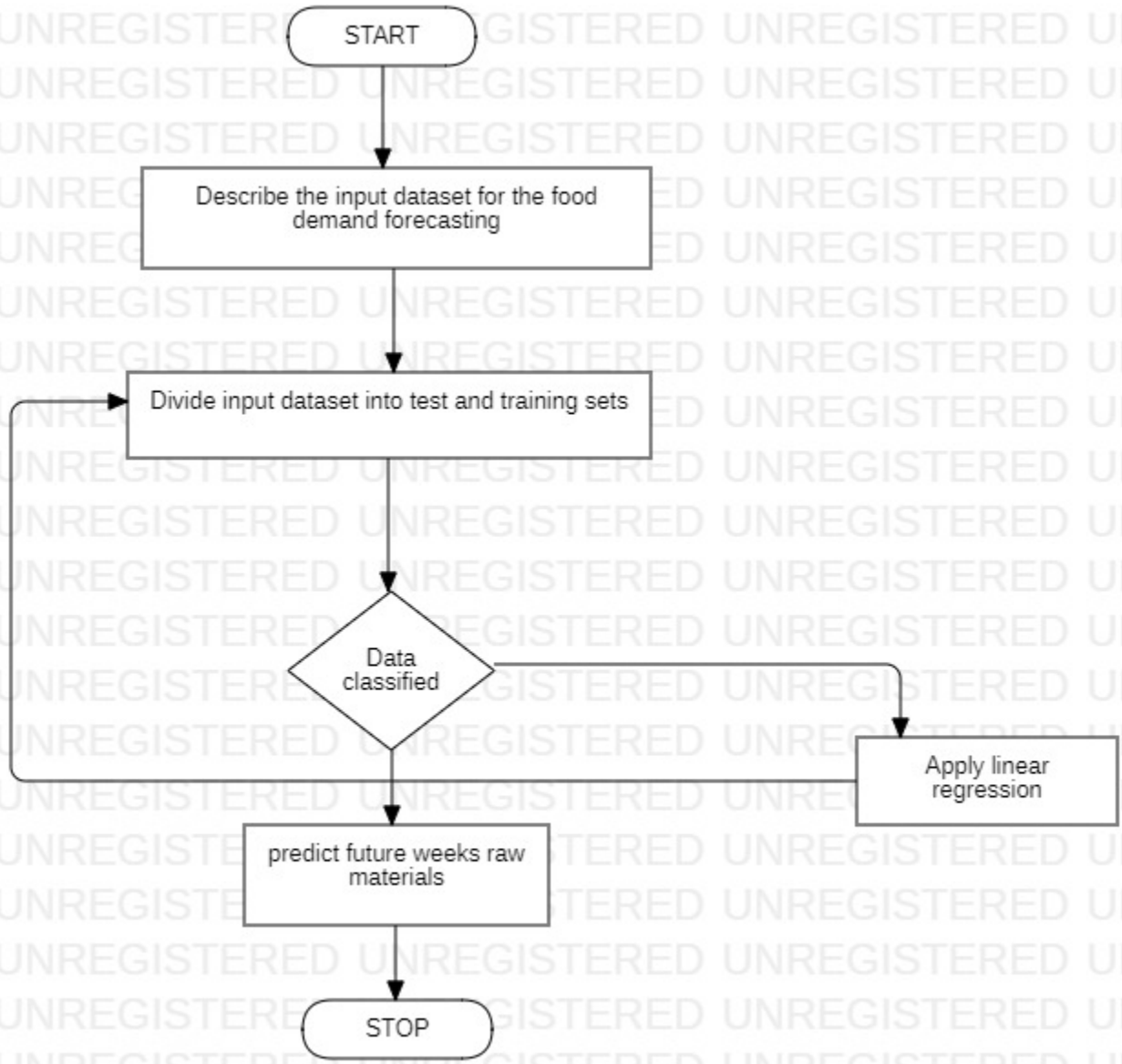
Below are the findings from the same:

● Orders for Italian meals and Beverages are always high

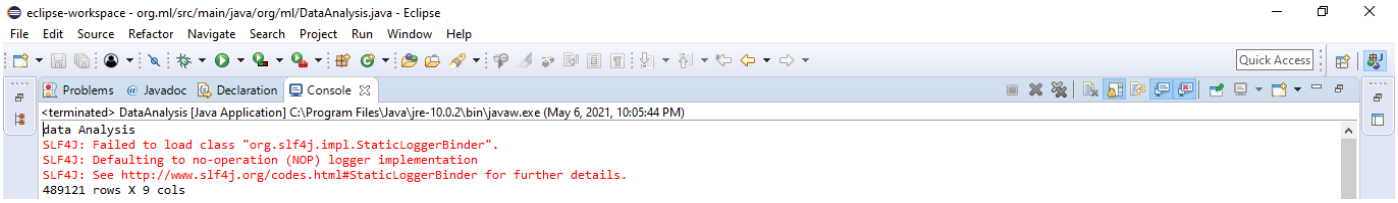
● There are fluctuations in the number of orders for Indian meals, Rice Bowl and Sandwich.

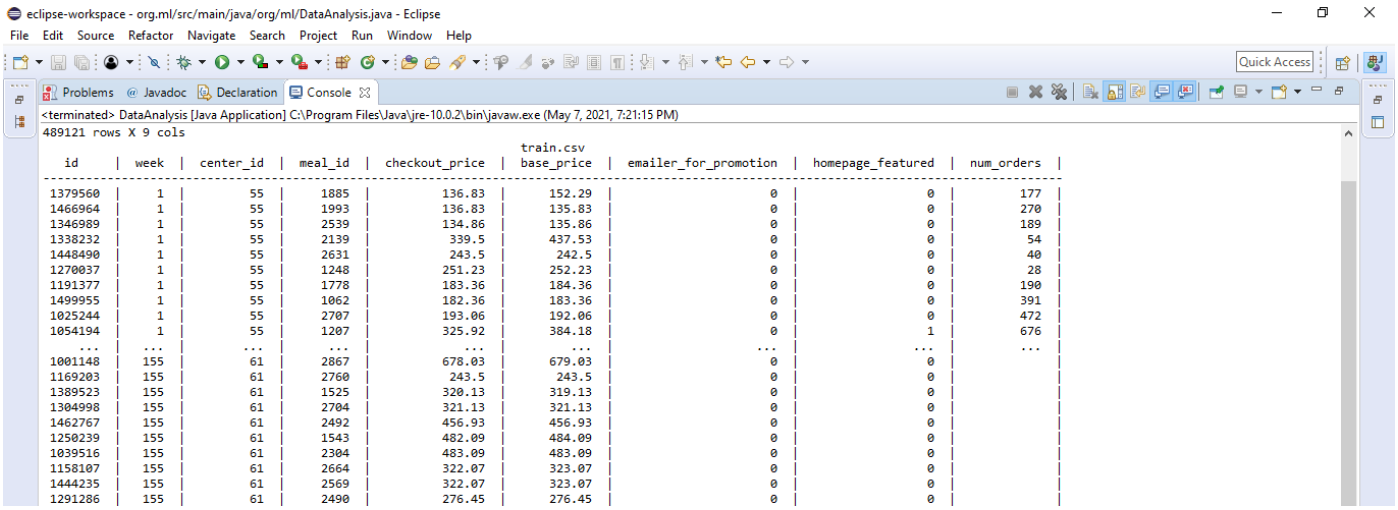
● Orders for Salad increased after week 18

5.FLOWCHART

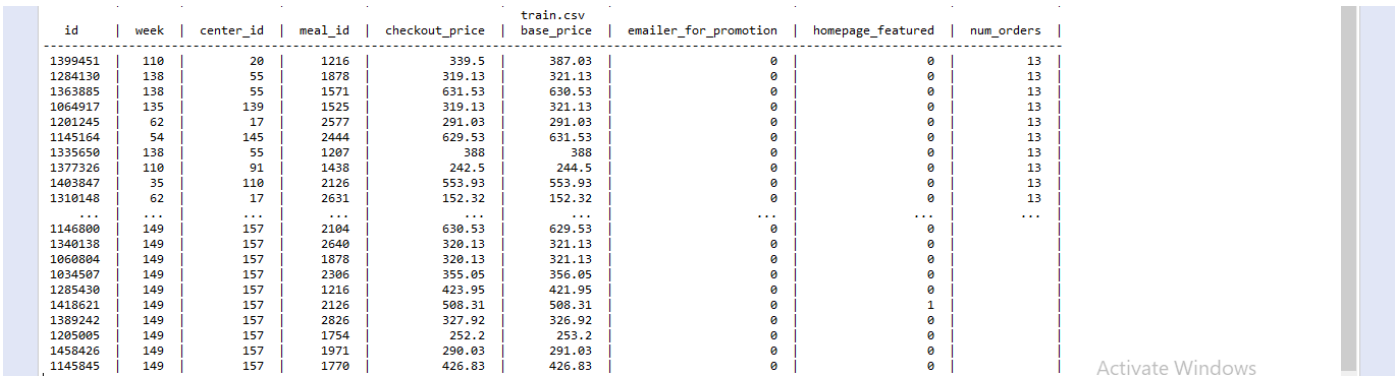

START
Describe the input dataset for the food demand forecasting
Divide input dataset into test and training sets
Data classified
Apply linear regression
predict future weeks raw materials
STOP

6.RESULT



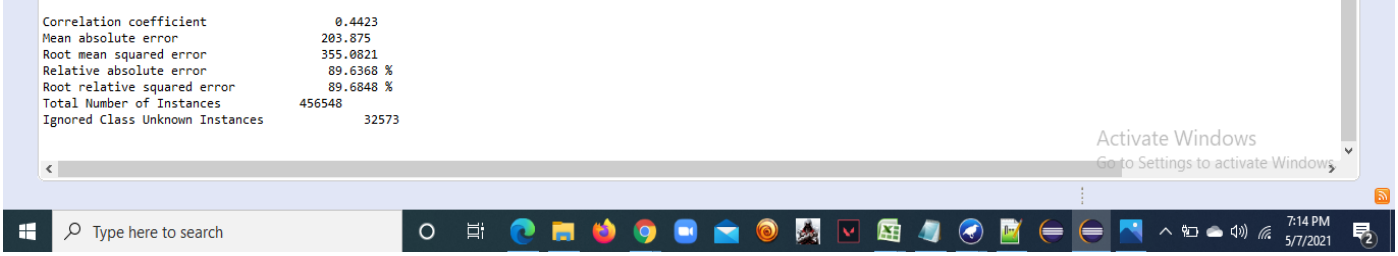


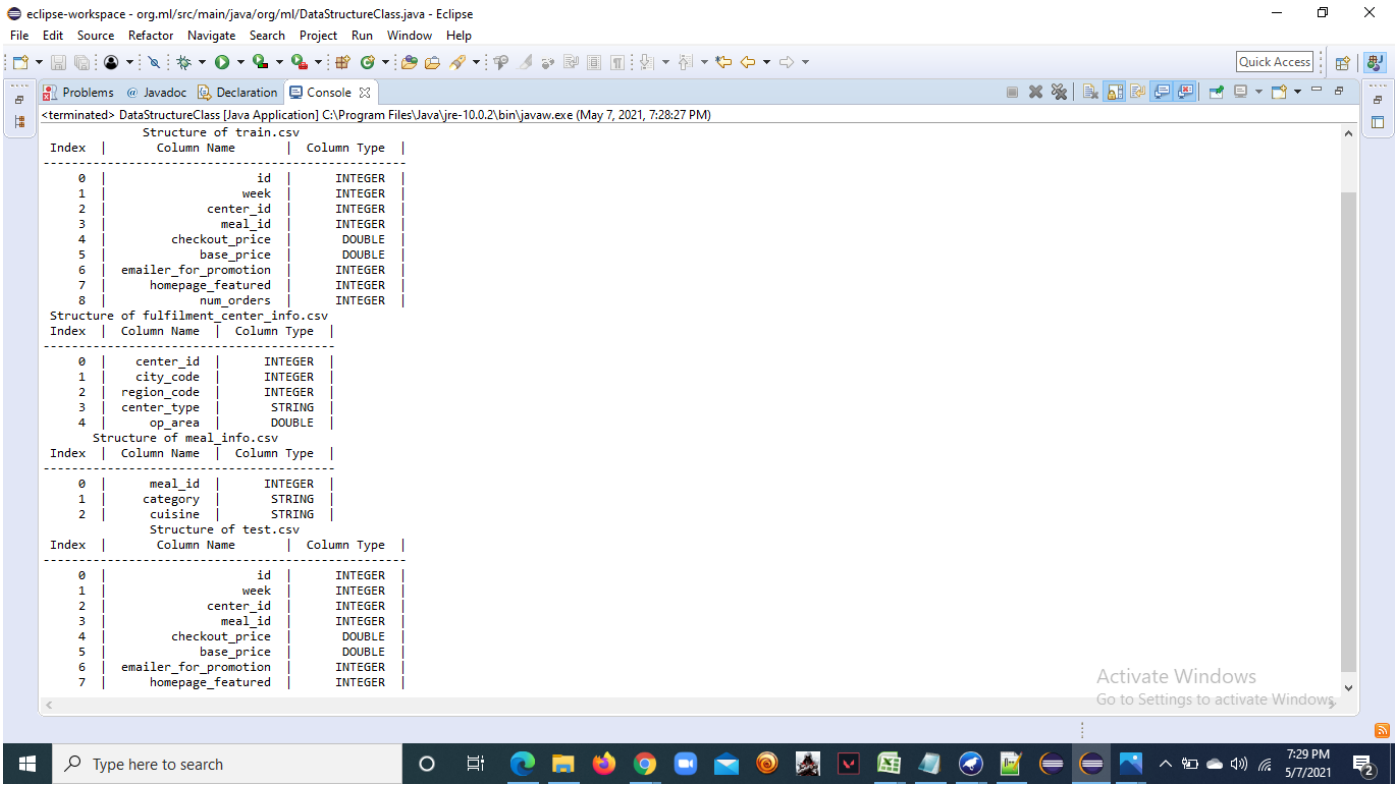
id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders
1379560	1	55	1885	136.83	152.29	0	0	177
1466964	1	55	1993	136.83	135.83	0	0	270
1346989	1	55	2539	134.86	135.86	0	0	189
1338232	1	55	2139	339.5	437.53	0	0	54
1448490	1	55	2631	243.5	242.5	0	0	40
1270037	1	55	1248	251.23	252.23	0	0	28
1191377	1	55	1778	183.36	184.36	0	0	190
1499955	1	55	1062	182.36	183.36	0	0	391
1025244	1	55	2707	193.06	192.06	0	0	472
1054194	1	55	1207	325.92	384.18	0	1	676
...	...	...	...	...	...	...	...	...
1001148	155	61	2867	678.03	679.03	0	0	
1169203	155	61	2760	243.5	243.5	0	0	
1389523	155	61	1525	320.13	319.13	0	0	
1304998	155	61	2704	321.13	321.13	0	0	
1462767	155	61	2492	456.93	456.93	0	0	
1250239	155	61	1543	482.09	484.09	0	0	
1039516	155	61	2304	483.09	483.09	0	0	
1158107	155	61	2664	322.07	323.07	0	0	
1444235	155	61	2569	322.07	323.07	0	0	
1291286	155	61	2490	276.45	276.45	0	0	



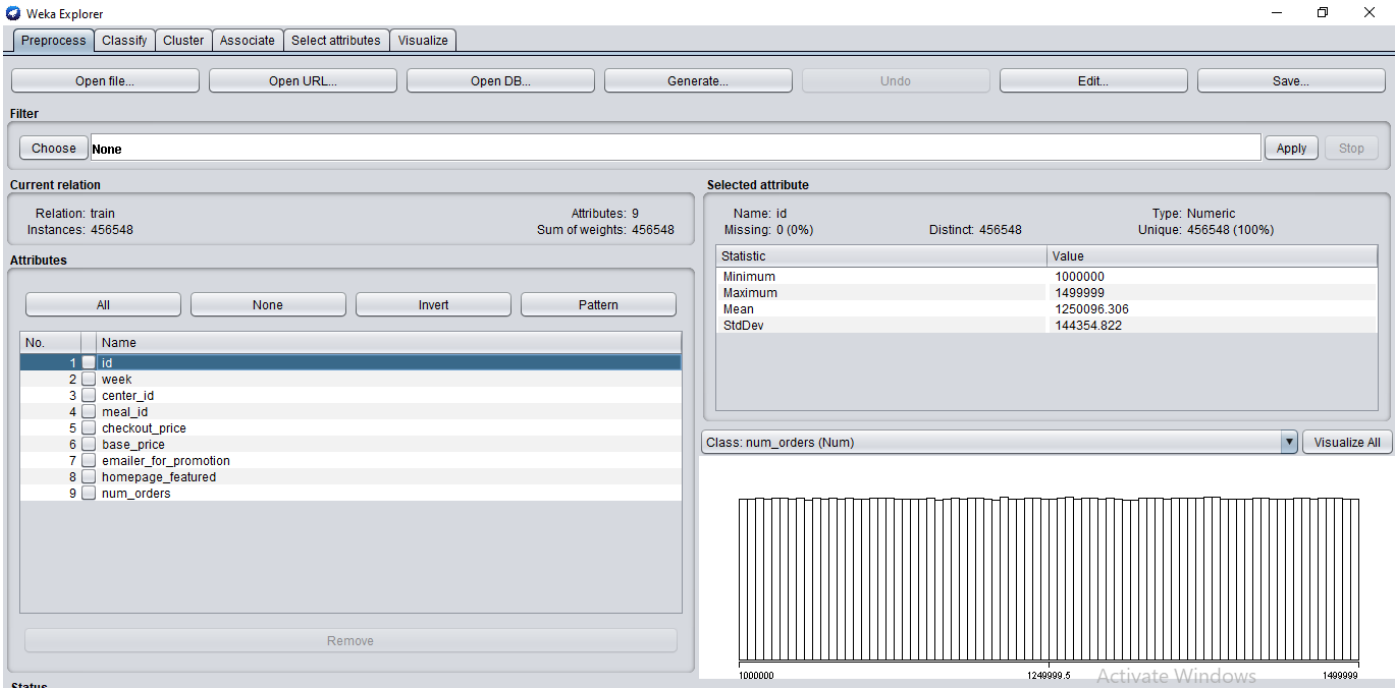
id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders
1399451	110	20	1216	339.5	387.03	0	0	13
1284130	138	55	1878	319.13	321.13	0	0	13
1363885	138	55	1571	631.53	630.53	0	0	13
1064917	135	139	1525	319.13	321.13	0	0	13
1201245	62	17	2577	291.03	291.03	0	0	13
1145164	54	145	2444	629.53	631.53	0	0	13
1335650	138	55	1207	388	388	0	0	13
1377326	110	91	1438	242.5	244.5	0	0	13
1403847	35	110	2126	553.93	553.93	0	0	13
1310148	62	17	2631	152.32	152.32	0	0	13
...	...	...	...	...	...	...	...	...
1146800	149	157	2104	630.53	629.53	0	0	
1340138	149	157	2640	320.13	321.13	0	0	
1060804	149	157	1878	320.13	321.13	0	0	
1034507	149	157	2306	355.05	356.05	0	0	
1285430	149	157	1216	423.95	421.95	0	0	
1418621	149	157	2126	508.31	508.31	0	1	
1389242	149	157	2826	327.92	326.92	0	0	
1205005	149	157	1754	252.2	253.2	0	0	
1458426	149	157	1971	290.03	291.03	0	0	
1145845	149	157	1770	426.83	426.83	0	0	

Linear Regression

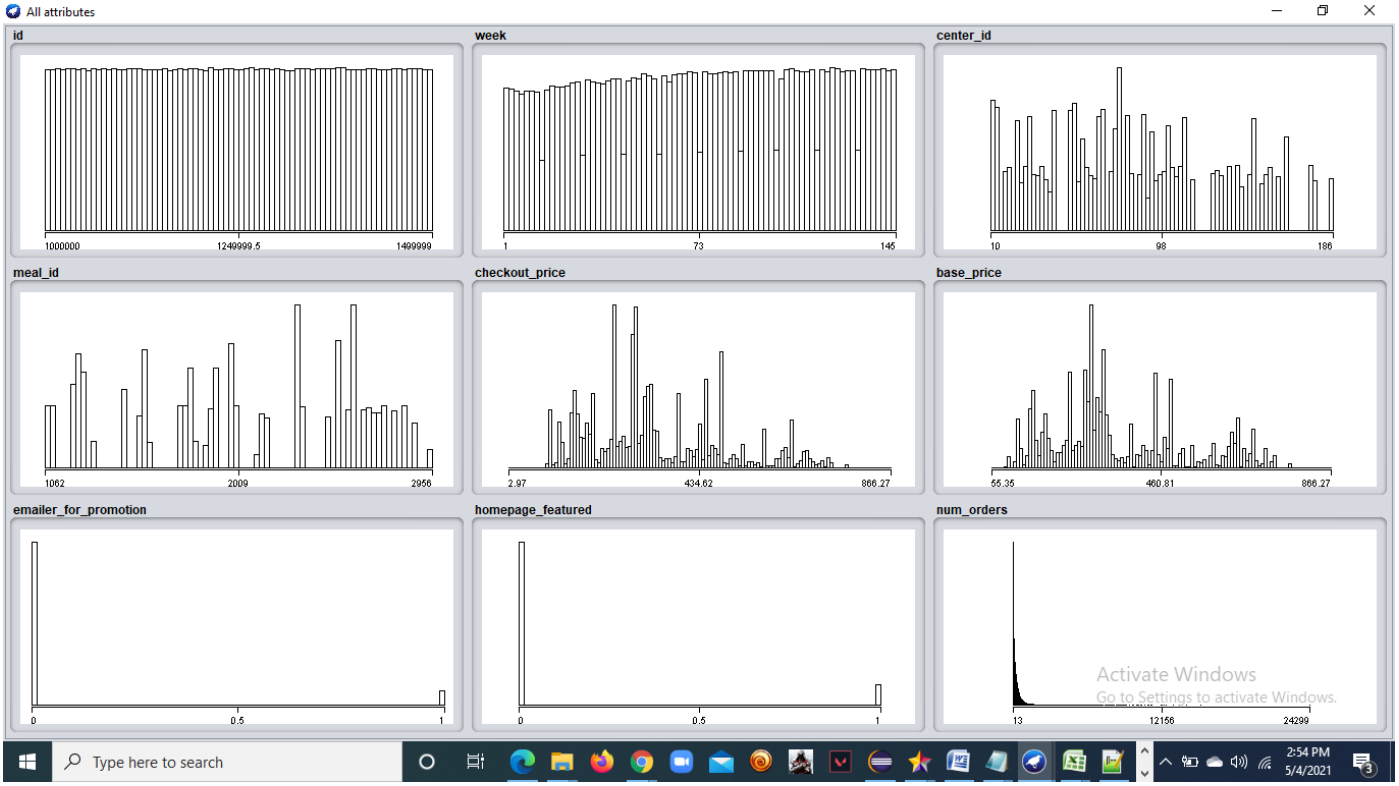




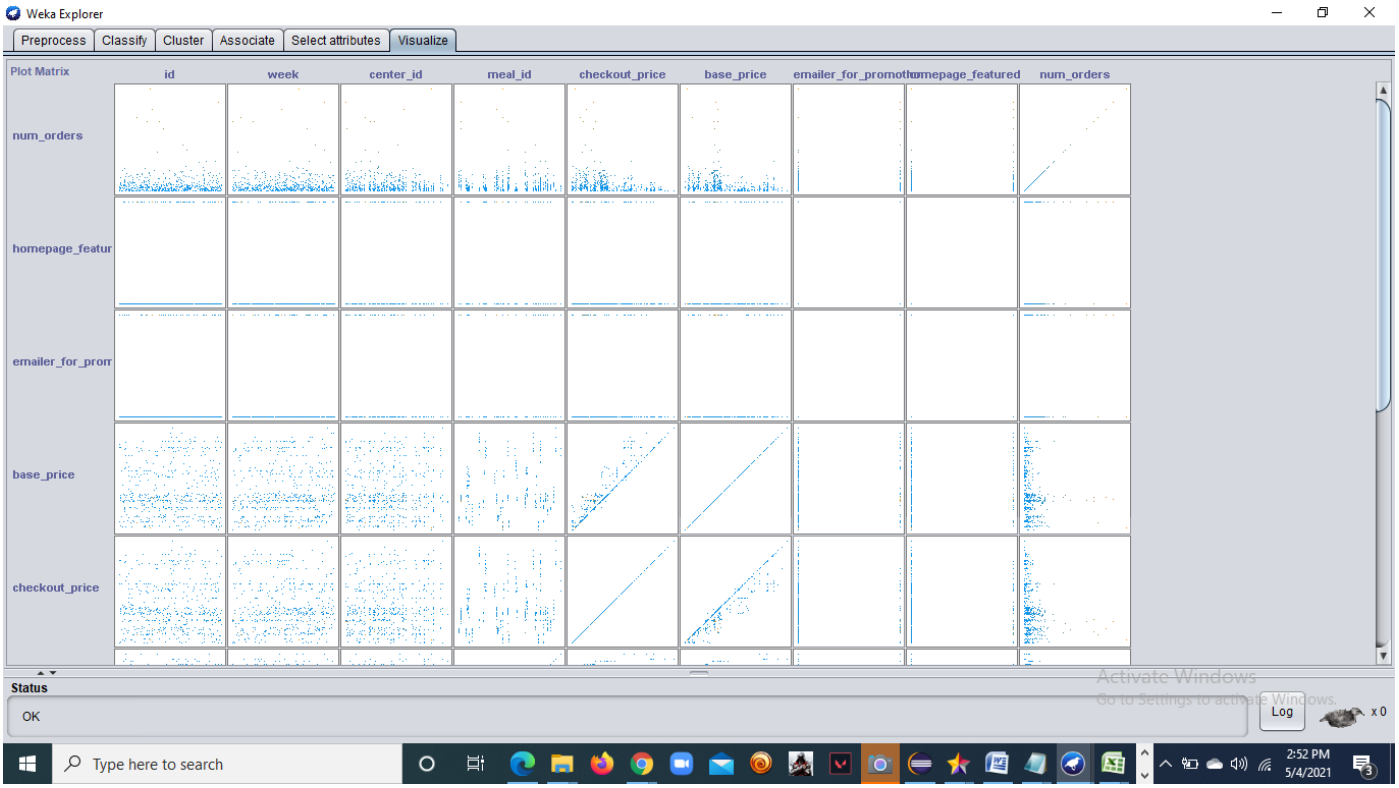
Data Set Attributes

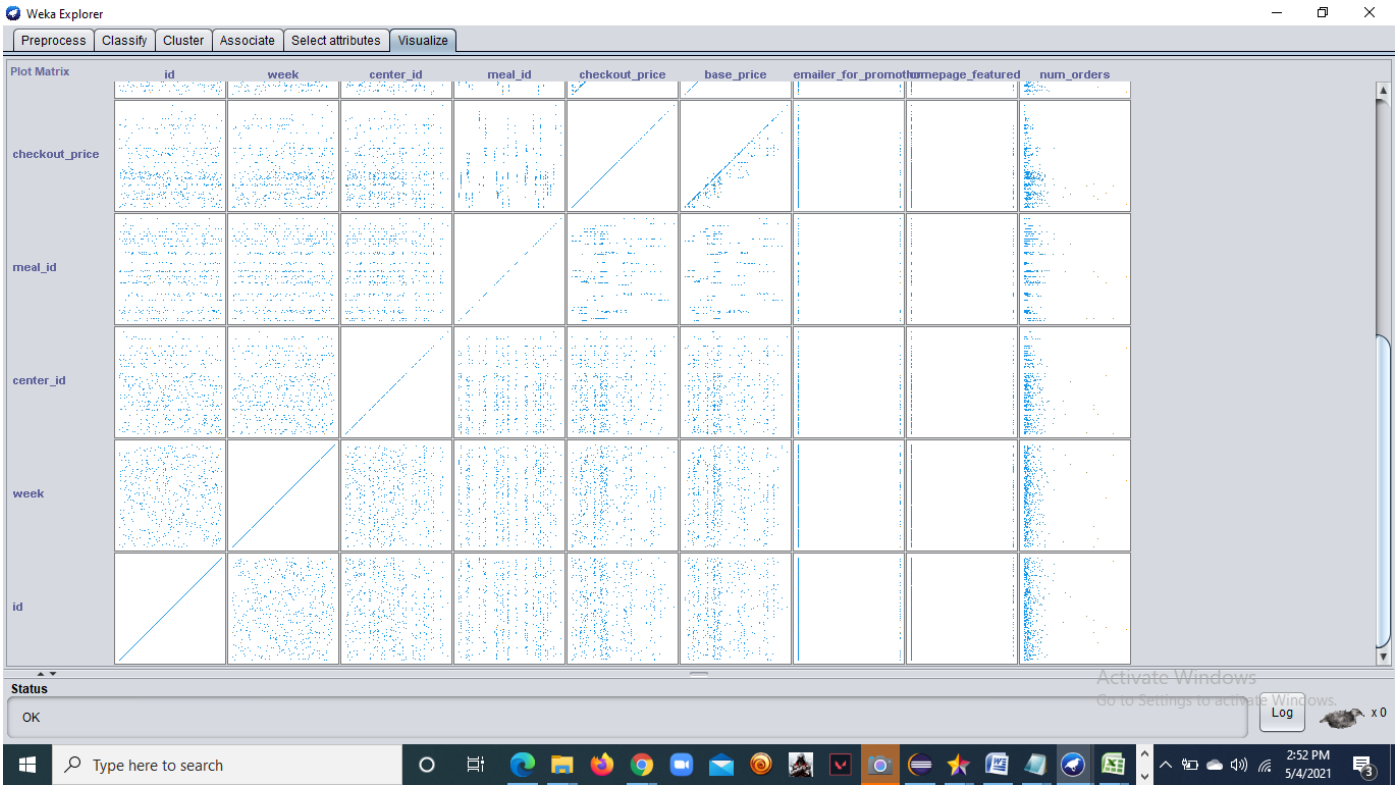


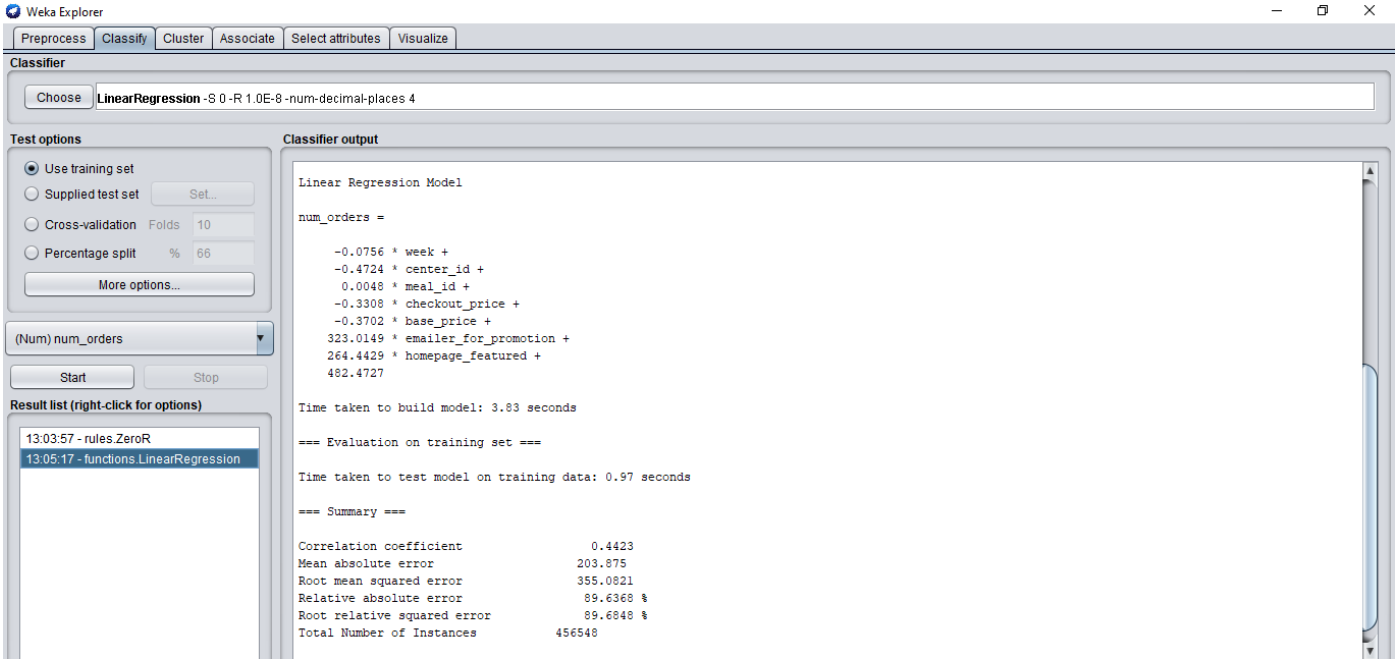
Visualization



Weka




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Plot Matrix
id
week
center_id
meal_id
checkout_price
base_price
emailer_for_promotion
homepage_featured
num_orders
Status
OK
Log
x 0
Activate Windows
Go to Settings to activate Windows.
Type here to search
2:52 PM
5/4/2021


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Num) num_orders
Start
Stop
Result list (right-click for options)
13:03:57 - rules.ZeroR
13:05:17 - functions.LinearRegression
Classifier output
Linear Regression Model
num_orders =
-0.0756 * week +
-0.4724 * center_id +
0.0048 * meal_id +
-0.3308 * checkout_price +
-0.3702 * base_price +
323.0149 * emailer_for_promotion +
264.4429 * homepage_featured +
482.4727
Time taken to build model: 3.83 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0.97 seconds
=== Summary ===
Correlation coefficient 0.4423
Mean absolute error 203.875
Root mean squared error 355.0821
Relative absolute error 89.6368 %
Root relative squared error 89.6848 %
Total Number of Instances 456548

There should be 77*51 = 3927 center-meal pair, but we have 3597 pairs in train data, that means some centers did not sell some of the meals.

There should be 3597*145 = 521565 records in past 145 week data, but we have 456548 records. which means some centers did not sell some meal for some week or they stared selling some new type of meal after some weeks. Same with test data.

Test set has only 3548 center-meal pair, that means some of the centers did not sell some type of meals in this 10 week.

Here in the test set (future 10 week), center 73 started selling meal 2956 & 1571, center 92 started selling meal 2104, which they have never sold in last 145 weeks. There are only 13 records with unknown center-meal pair in test set.

7.ADVANTAGES & DISADVANTAGES

Advantages:

1. You'll gain valuable insight

Forecasting gets you into the habit of looking at past and real-time data to predict future demand. And in doing so, you'll be able to anticipate demand fluctuations more effectively.

2. You'll learn from past mistakes

You don't start from scratch after each forecast. Even if your prediction was nowhere close to what ended up coming to pass, it gives you a starting point.

3. It can decrease costs

When done right, anticipating demand will help you tweak your processes to increase efficiency all along the supply chain.

Disadvantages:

1. Forecasts are never 100% accurate

Let's face it: it's hard to predict the future. Even if you have a great process in place and forecasting experts on your payroll, your forecasts will never be spot on.

2. It can be time-consuming and resource-intensive

Forecasting involves a lot of data gathering, data organizing, and coordination. Companies typically employ a team of demand planners who are responsible for coming up with the forecast.

3. It can also be cost

On a related note, hiring a team of demand planners is a significant investment.

8.APPLICATIONS

- Customer demand planning
- Economic forecasting
- Earthquake prediction
- Egain forecasting
- Finance against risk of default via credit ratings and credit scores
- Land use forecasting
- Player and team performance in sports
- Political forecasting
- Product forecasting
- Sales forecasting
- Technology forecasting
- Telecommunications forecasting
- Transport planning and Transportation forecasting
- Weather forecasting, Flood forecasting and Meteorology

9.CONCLUSION

we forecasted raw materials for next few weeks,using datasets in machine learning model.Here we have given only for 145weeks data so for next few weeks the data is forecasted.

10.FUTURE SCOPE

By this analysis we can forecast the raw materials for next few weeks without any wastage of food, so,this machine learning model forecast can help in analysize the raw material .
A food delivery service has to deal with a lot of perishable raw materials which makes it all, the most important factor for such a company is to accurately forecast daily and weekly demand. Too much inventory in the warehouse means more risk of wastage, and not enough could lead to out-of-stocks and push customers to seek solutions from your competitors. The replenishment of the majority of raw materials is done on weekly basis and since the raw material is perishable, the procurement planning is of utmost importance.

11.BIBILOGRAPY

https://www.kaggle.com/kannanaikkal/food-demand-forecasting

APPENDIX

A. Source code: pom.xml

```
<?xml version="1.0"?>
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd" xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com</groupId>
  <artifactId>org.ml</artifactId>
  <version>110.0.1-SNAPSHOT</version>
  <properties>
    <maven.compiler.target>1.8</maven.compiler.target>
    <maven.compiler.source>1.8</maven.compiler.source>
  </properties>
  <dependencies>
    <dependency>
      <groupId>nz.ac.waikato.cms.weka</groupId>
      <artifactId>weka-stable</artifactId>
      <version>3.8.0</version>
      <scope>compile</scope>
    </dependency>
    <dependency>
      <groupId>tech.tablesaw</groupId>
      <artifactId>tablesaw-core</artifactId>
      <version>0.38.1</version>
      <scope>compile</scope>
    </dependency>
    <dependency>
      <groupId>tech.tablesaw</groupId>
      <artifactId>tablesaw-jsplot</artifactId>
      <version>0.38.1</version>
      <scope>compile</scope>
    </dependency>
  </dependencies>
  <repositories>
    <repository>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
      <id>central</id>
```

```
            </reportPlugins>
          </configuration>
        </execution>
        <execution>
          <id>default-deploy</id>
          <phase>site-deploy</phase>
          <goals>
            <goal>deploy</goal>
          </goals>
          <configuration>
            <outputDirectory>C:\Users\Dhanpal Pravalika\eclipse-workspace\org.ml\target\site</outputDirectory>
            <reportPlugins>
              <reportPlugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-project-info-reports-plugin</artifactId>
              </reportPlugin>
            </reportPlugins>
          </configuration>
        </execution>
      </executions>
      <configuration>
        <outputDirectory>C:\Users\Dhanpal Pravalika\eclipse-workspace\org.ml\target\site</outputDirectory>
        <reportPlugins>
          <reportPlugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-project-info-reports-plugin</artifactId>
          </reportPlugin>
        </reportPlugins>
      </configuration>
    </plugin>
  </plugins>
 </build>
 <reporting>
   <outputDirectory>C:\Users\Dhanpal Pravalika\eclipse-workspace\org.ml\target\site</outputDirectory>
 </reporting>
</project>
```

Data analysis

```
package org.ml;

import java.io.IOException;

public class DataAnalysis
{
public static void main(String arg[])
{
System.out.println("data Analysis");
try
{
     Table fulfilment_center_info_data=Table.read().csv("E:\\data project\\fulfilment_center_info.csv");
     Table meal_info_data=Table.read().csv("E:\\data project\\meal_info.csv");
     Table test_data=Table.read().csv("E:\\data project\\test.csv");
     Table train_data=Table.read().csv("E:\\data project\\train.csv");
     Table meal_info=train_data.sortAscendingOn("num_orders");
     Table week_info=train_data.sortAscendingOn("week");
            System.out.println(meal_info.shape());
            System.out.println(week_info);
            System.out.println(meal_info);
                   Layout layout1 = Layout.builder().title("Distribution of meal_id").build();
                   HistogramTrace trace1 = HistogramTrace.builder(train_data.nCol("meal_id")).build();
                   Plot.show(new Figure(layout1, trace1));
                       Layout layout3 = Layout.builder().title(" train.csv").build();
                       BoxTrace trace3 =BoxTrace.builder(train_data.categoricalColumn("num_orders"), train_data.nCol("num_orders")).build();
                       Plot.show(new Figure(layout3, trace3));
}
catch(IOException e)
{
e.printStackTrace();
}
}
}
```

Data structure analysis

```
package org.ml;

import java.io.IOException;

public class DataStructureClass {

    public static void main(String arg[])
    {
    System.out.println("data Analysis");
    try
    {
             Table fulfilment_center_info_data=Table.read().csv("E:\\data project\\fulfilment_center_info.csv");
             Table meal_info_data=Table.read().csv("E:\\data project\\meal_info.csv");
             Table test_data=Table.read().csv("E:\\data project\\test.csv");
             Table train_data=Table.read().csv("E:\\data project\\train.csv");

    System.out.println(train_data.structure());
    System.out.println(fulfilment_center_info_data.structure());
    System.out.println(meal_info_data.structure());
    System.out.println(test_data.structure());

    }
    catch(IOException e)
    {
    e.printStackTrace();
    }
    }
}
```

Linear Regression

```
package org.ml;

import weka.classifiers.Evaluation;
import weka.classifiers.functions.LinearRegression;
import weka.core.Instances;
import weka.core.converters.ConverterUtils.DataSource;
public class regression
{
    public static void main(String[] args)

            throws Exception
    {
        DataSource source =new DataSource("E:\\data project\\train.csv");
        Instances dataset=source.getDataSet();
        dataset.setClassIndex(dataset.numAttributes()-1);
        //linear Regression
        LinearRegression lr=new LinearRegression();
        lr.buildClassifier(dataset);

        Evaluation lreval =new Evaluation(dataset);
        lreval.evaluateModel(lr,dataset);
        System.out.println(lreval.toSummaryString());


    }

}
```

Data calculation

```
package org.ml;

import java.io.IOException;
import tech.tablesaw.api.Table;
import tech.tablesaw.joining.DataFrameJoiner;

public class DataCalculation {
    public static void main(String arg[]) throws IOException
    {
        Table fulfilment_center_info_data=Table.read().csv("E:\\data project\\fulfilment_center_info.csv");
        Table meal_info_data=Table.read().csv("E:\\data project\\meal_info.csv");
        Table test_data=Table.read().csv("E:\\data project\\test.csv");
        Table train_data=Table.read().csv("E:\\data project\\train.csv");
        DataFrameJoiner train = train_data.joinOn("center_id");
        DataFrameJoiner test = test_data.joinOn("center_id");
        train_data=train.inner(fulfilment_center_info_data, "center_id");
        test_data=test.inner(fulfilment_center_info_data, "center_id");
        DataFrameJoiner train1 = train_data.joinOn("meal_id");
        DataFrameJoiner test1 = test_data.joinOn("meal_id");
        train_data=train1.inner(meal_info_data, "meal_id");
        test_data=test1.inner(meal_info_data, "meal_id");
        System.out.println(test_data.structure());
        System.out.println(train_data.structure());
        System.out.println(train_data.shape());
        System.out.println(train_data.summary());
        System.out.println(test_data.summary());
        String centerId="center_id";
        String center_meal_pair=train_data.columnArray()[train_data.columnIndex(centerId)]+"_"+ train_data.columnArray()[train_data.columnIndex("meal_id")];
        for(int r=0;r<train_data.columnCount();r++) {
            System.out.println(test_data.columnArray()[r].summary());
        }
    }
}
```