Food Demand Forecasting for Food Delivery Company

Project Descreption:

The majority of the raw ingredients used by a food delivery service are perishable, thus precisely forecasting daily and weekly demand is crucial for this kind of business. A warehouse with too much inventory runs the danger of wastage, while one with too little could experience out-of-stocks, which would force clients to turn to your rivals for assistance. The challenge is to forecast demand because the majority of raw materials must be replenished on a weekly basis and are perishable, making procurement planning crucial.

The main goal of this project is to create an appropriate machine learning model to forecast the number of orders to gather raw materials for upcoming weeks. To achieve this, we should first gather data from HDFS. We should know the information about of fulfilment center like area, city etc., and meal information like category of food sub category of food price of the food or discount in particular week. By using this data, we can use any regression algorithm to forecast the quantity for upcoming weeks.

Data Pre-processing

Data Pre-processing includes the following main tasks

- Import the libraries
- Loading the dataset from HDFS
- Exploratory Data Analysis
- Checking for Null Values
- Reading and merging files
- Dropping the columns
- Label Encoding
- Data Visualization
- Splitting the Dataset into train and test
- Model Building

The train data is splitted into 70/30 % train/test and model is build using Linear Regressor and Random Forest Tree Regressor.

In [1]:
```
# Import Libraries
from pyspark.sql import SparkSession
from pyspark.sql.functions import col
from pyspark.ml.stat import Correlation
from pyspark.ml.feature import VectorAssembler
from pyspark.sql.functions import monotonically_increasing_id

from pyspark.ml.feature import StringIndexer
from pyspark.ml.regression import LinearRegression
from pyspark.ml.regression import RandomForestRegressor
from pyspark.ml.evaluation import RegressionEvaluator

# Import standard libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotnine as p9
```

```
%matplotlib inline
color = sns.color_palette()
```

In [2]:
```
# Create a SparkSession
spark = SparkSession.builder.appName("FoodDemand").getOrCreate()

# Load a fulfilmentcenterinfo table from HDFS
center_data = spark.read.format("csv").option("header", "false").load("fooddemand/fulfil

# Columns of the fulfilmentcenterinfo DataFrame
center_data = center_data.withColumnRenamed("_c0", "center_id").withColumnRenamed("_c1",
# Datatype of the fulfilmentcenterinfo DataFrame
center_data = center_data.withColumn("center_id", col("center_id").cast("integer")).with

# Show the contents of the fulfilmentcenterinfo DataFrame
center_data.show()
```

```
23/04/11 16:57:15 WARN Utils: Your hostname, cis6180 resolves to a loopback address: 12
7.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
23/04/11 16:57:15 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLev
el).
23/04/11 16:57:16 WARN NativeCodeLoader: Unable to load native-hadoop library for your p
latform... using builtin-java classes where applicable
+---------+---------+-----------+-----------+-------+
|center_id|city_code|region_code|center_type|op_area|
+---------+---------+-----------+-----------+-------+
|       11|      679|         56|     TYPE_A|    3.7|
|       13|      590|         56|     TYPE_B|    6.7|
|      124|      590|         56|     TYPE_C|    4.0|
|       66|      648|         34|     TYPE_A|    4.1|
|       94|      632|         34|     TYPE_C|    3.6|
|       64|      553|         77|     TYPE_A|    4.4|
|      129|      593|         77|     TYPE_A|    3.9|
|      139|      693|         34|     TYPE_C|    2.8|
|       88|      526|         34|     TYPE_A|    4.1|
|      143|      562|         77|     TYPE_B|    3.8|
|      101|      699|         85|     TYPE_C|    2.8|
|       86|      699|         85|     TYPE_C|    4.0|
|       32|      526|         34|     TYPE_A|    3.8|
|      149|      478|         77|     TYPE_A|    2.4|
|      152|      576|         34|     TYPE_B|    4.0|
|       92|      526|         34|     TYPE_C|    2.9|
|       27|      713|         85|     TYPE_A|    4.5|
|       14|      654|         56|     TYPE_C|    2.7|
|       26|      515|         77|     TYPE_C|    3.0|
|      104|      647|         56|     TYPE_A|    4.5|
+---------+---------+-----------+-----------+-------+
only showing top 20 rows
```

In [3]:
```
# Print the total number of rows of the fulfilmentcenterinfo DataFrame
print("Total number of rows: ", center_data.count())
```

```
Total number of rows:  77
```

In [4]:
```
# Print fulfilmentcenterinfo schema
center_data.printSchema()
```

```
root
 |-- center_id: integer (nullable = true)
 |-- city_code: integer (nullable = true)
 |-- region_code: integer (nullable = true)
 |-- center_type: string (nullable = true)
```

```
 |-- op_area: float (nullable = true)
```

In [5]:
```
# Load a mealinfo table from HDFS
meal_data = spark.read.format("csv").option("header", "false").load("fooddemand/mealinfo

# Columns of the mealinfo DataFrame
meal_data = meal_data.withColumnRenamed("_c0", "meal_id").withColumnRenamed("_c1", "cate
# Datatype of the mealinfo DataFrame
meal_data = meal_data.withColumn("meal_id", col("meal_id").cast("integer")).withColumn("

# Show the contents of the mealinfo DataFrame
meal_data.show()
```

```
+-------+------------+-------+
|meal_id|    category|cuisine|
+-------+------------+-------+
|   1885|   Beverages|   Thai|
|   1993|   Beverages|   Thai|
|   2539|   Beverages|   Thai|
|   1248|   Beverages| Indian|
|   2631|   Beverages| Indian|
|   1311|      Extras|   Thai|
|   1062|   Beverages|Italian|
|   1778|   Beverages|Italian|
|   1803|      Extras|   Thai|
|   1198|      Extras|   Thai|
|   2707|   Beverages|Italian|
|   1847|        Soup|   Thai|
|   1438|        Soup|   Thai|
|   2494|        Soup|   Thai|
|   2760|Other Snacks|   Thai|
|   2490|       Salad|Italian|
|   1109|   Rice Bowl| Indian|
|   2290|   Rice Bowl| Indian|
|   1525|Other Snacks|   Thai|
|   2704|Other Snacks|   Thai|
+-------+------------+-------+
only showing top 20 rows
```

In [6]:
```
# Print the total number of rows of the mealinfo DataFrame
print("Total number of rows: ", meal_data.count())
```

```
Total number of rows:  51
```

In [7]:
```
# Print mealinfo schema
meal_data.printSchema()
```

```
root
 |-- meal_id: integer (nullable = true)
 |-- category: string (nullable = true)
 |-- cuisine: string (nullable = true)
```

In [8]:
```
# Load a train table from HDFS
train_data = spark.read.format("csv").option("header", "false").load("fooddemand/train/p

# Columns of the train DataFrame
train_data = train_data.withColumnRenamed("_c0", "id").withColumnRenamed("_c1", "week").
# Datatype of the fulfilmentcenterinfo DataFrame
train_data = train_data.withColumn("id", col("id").cast("integer")).withColumn("week", c

# Show the contents of the train DataFrame
train_data.show()
```

```
+-------+----+---------+-------+--------------+----------+-------------------+----------
```

```
-------+----------+
|     id|week|center_id|meal_id|checkout_price|base_price|email_for_promotion|homepage_f
eatured|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|
|1469367|   1|       55|   1230|        323.01|     390.0|                  0|
      1|       823|
|1029333|   1|       55|   2322|        322.07|     388.0|                  0|
      1|       972|
|1446016|   1|       55|   2290|        311.43|    310.43|                  0|
      0|       162|
|1244647|   1|       55|   1727|        445.23|    446.23|                  0|
      0|       420|
|1378227|   1|       55|   1109|        264.84|    297.79|                  1|
      0|       756|
|1181556|   1|       55|   2640|        282.33|    281.33|                  0|
      0|       108|
|1313873|   1|       55|   2306|         243.5|    340.53|                  0|
      0|        28|
|1067069|   1|       55|   2126|         486.0|     485.0|                  0|
      0|        28|
|1058482|   1|       55|   2826|        306.58|    305.58|                  0|
      0|       188|
|1240935|   1|       55|   1754|        289.12|    289.12|                  0|
      0|       485|
+-------+----+---------+-------+--------------+----------+-------------------+----------
-------+----------+
only showing top 20 rows
```

In [9]:
```
# Print the total number of rows of the train DataFrame
print("Total number of rows: ", train_data.count())
```

```
Total number of rows:  456548
```

In [10]:
```
# Print train data schema
train_data.printSchema()
```

```
root
 |-- id: integer (nullable = true)
 |-- week: integer (nullable = true)
 |-- center_id: integer (nullable = true)
 |-- meal_id: integer (nullable = true)
 |-- checkout_price: float (nullable = true)
 |-- base_price: float (nullable = true)
 |-- email_for_promotion: integer (nullable = true)
```

```
 |-- homepage_featured: integer (nullable = true)
 |-- num_orders: integer (nullable = true)
```

In [11]:

```
# Load a test table from HDFS
test_data = spark.read.format("csv").option("header", "false").load("fooddemand/test/par

# Columns of the test DataFrame
test_data = test_data.withColumnRenamed("_c0", "id").withColumnRenamed("_c1", "week").wi
# Datatype of the fulfilmentcenterinfo DataFrame
test_data = test_data.withColumn("id", col("id").cast("integer")).withColumn("week", col

# Show the contents of the test DataFrame
test_data.show()
```

```
+-------+----+---------+-------+--------------+----------+------------------+----------
-------+
|     id|week|center_id|meal_id|checkout_price|base_price|email_for_promotion|homepage_f
eatured|
+-------+----+---------+-------+--------------+----------+------------------+----------
-------+
|1028232| 146|       55|   1885|        158.11|    159.11|                 0|
      0|
|1127204| 146|       55|   1993|        160.11|    159.11|                 0|
      0|
|1212707| 146|       55|   2539|        157.14|    159.14|                 0|
      0|
|1082698| 146|       55|   2631|        162.02|    162.02|                 0|
      0|
|1400926| 146|       55|   1248|        163.93|    163.93|                 0|
      0|
|1284113| 146|       55|   1778|        190.15|    190.15|                 0|
      0|
|1197966| 146|       55|   1062|        191.09|    192.09|                 0|
      0|
|1132739| 146|       55|   2707|        242.56|    240.56|                 0|
      0|
|1057981| 146|       55|   1207|         360.9|     360.9|                 0|
      0|
|1095932| 146|       55|   1230|        383.18|    384.18|                 0|
      0|
|1454421| 146|       55|   2322|         389.0|     390.0|                 0|
      0|
|1149765| 146|       55|   2290|        302.64|    303.64|                 0|
      0|
|1166964| 146|       55|   1727|        466.63|    465.63|                 0|
      0|
|1010949| 146|       55|   1109|        309.43|    310.43|                 0|
      0|
|1409575| 146|       55|   2577|        320.13|    320.13|                 0|
      1|
|1181616| 146|       55|   2640|        319.13|    319.13|                 0|
      0|
|1456020| 146|       55|   2826|        377.33|    379.33|                 0|
      0|
|1264395| 146|       55|   1754|        329.86|    329.86|                 0|
      0|
|1472931| 146|       55|   1971|        344.35|    344.35|                 0|
      0|
|1412015| 146|       55|   1198|         135.8|    190.18|                 0|
      0|
+-------+----+---------+-------+--------------+----------+------------------+----------
-------+
only showing top 20 rows
```

In [12]:
```
# Print the total number of rows of the train DataFrame
print("Total number of rows: ", test_data.count())
```

```
Total number of rows:  32573
```

In [13]:
```
# Print test data schema
test_data.printSchema()
```

```
root
 |-- id: integer (nullable = true)
 |-- week: integer (nullable = true)
 |-- center_id: integer (nullable = true)
 |-- meal_id: integer (nullable = true)
 |-- checkout_price: float (nullable = true)
 |-- base_price: float (nullable = true)
 |-- email_for_promotion: integer (nullable = true)
 |-- homepage_featured: integer (nullable = true)
```

Merging_data train and meal_data dataset by using common key id:

The meal_id column in train_data is similar to meal_id in meal_data dataset. Let us merge these two datasets using common key meal_id and name the table as data_train.

In [14]:
```
data_train = train_data.join(meal_data, on='meal_id', how='outer')
data_test = test_data.join(meal_data, on='meal_id', how='outer')
```

Merging data_train and center_data dataset by using common key center_id:

The center_id column in data_train is similar to center_id in center_data dataset. Let us merge these two datasets, using common key center_id and store it back in data_train.

In [15]:
```
data_train = data_train.join(center_data, on='center_id', how='outer')
data_test = data_test.join(center_data, on='center_id', how='outer')
data_train.show()
```

```
[Stage 25:=============================================>             (3 + 1) / 4]
+---------+-------+-------+----+--------------+----------+-------------------+----------
-------+----------+---------+-------+---------+-----------+-----------+-------+
|center_id|meal_id|     id|week|checkout_price|base_price|email_for_promotion|homepage_f
eatured|num_orders| category|cuisine|city_code|region_code|center_type|op_area|
+---------+-------+-------+----+--------------+----------+-------------------+----------
-------+----------+---------+-------+---------+-----------+-----------+-------+
|       26|   1062|1215035|   1|        161.08|    161.08|                  0|
      0|       324|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|       26|   1062|1004523|   2|        174.66|    174.66|                  0|
      0|       418|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|       26|   1062|1315636|   3|        177.54|    178.54|                  0|
      0|       337|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|       26|   1062|1001462|   4|        178.54|    177.54|                  0|
      0|       324|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|       26|   1062|1381001|   5|        165.93|    164.93|                  0|
      0|       338|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|       26|   1062|1183853|   6|        155.26|    153.26|                  0|
      0|       554|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|       26|   1062|1245045|   7|        159.14|    159.14|                  0|
      0|       460|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|       26|   1062|1046050|   8|        149.41|    149.41|                  0|
      0|       447|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|       26|   1062|1084268|   9|        154.23|    154.23|                  0|
      0|       474|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|       26|   1062|1256757|  10|         156.2|     155.2|                  0|
```

```
      0|       676|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1457239|  11|        155.26|    153.26|                  0|
      0|       620|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1002295|  12|        168.84|    168.84|                  0|
      0|       420|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1372744|  13|        172.69|    173.69|                  0|
      0|       176|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1428509|  14|        178.54|    178.54|                  0|
      0|       324|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1336073|  15|        172.69|    173.69|                  0|
      0|       364|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1369877|  16|        173.63|    174.63|                  0|
      0|       257|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1198059|  17|        153.26|    173.63|                  0|
      0|       595|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1034336|  18|        149.38|    173.63|                  1|
      1|       566|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1222578|  19|        151.32|    175.63|                  0|
      1|       379|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
|       26|   1062|1153637|  20|        151.32|    175.63|                  0|
      1|       284|Beverages|Italian|      515|          77|     TYPE_C|    3.0|
+---------+-------+-------+----+--------------+----------+-------------------+----------
-------+----------+---------+-------+---------+-----------+-----------+-------+
only showing top 20 rows
```

In [16]:

```
# Drop columns “center_id” and “meal_id” as they are not required for the further proces
data_train = data_train.drop('meal_id', 'center_id')
data_test = data_test.drop('meal_id', 'center_id')
data_train.show(5)
```

```
[Stage 36:==============================>                          (2 + 2) / 4]
+-------+----+--------------+----------+-------------------+-----------------+----------
+---------+-------+---------+-----------+-----------+-------+
|     id|week|checkout_price|base_price|email_for_promotion|homepage_featured|num_orders
| category|cuisine|city_code|region_code|center_type|op_area|
+-------+----+--------------+----------+-------------------+-----------------+----------
+---------+-------+---------+-----------+-----------+-------+
|1215035|   1|        161.08|    161.08|                  0|                0|       324
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|1004523|   2|        174.66|    174.66|                  0|                0|       418
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|1315636|   3|        177.54|    178.54|                  0|                0|       337
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|1001462|   4|        178.54|    177.54|                  0|                0|       324
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
|1381001|   5|        165.93|    164.93|                  0|                0|       338
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|
+-------+----+--------------+----------+-------------------+-----------------+----------
+---------+-------+---------+-----------+-----------+-------+
only showing top 5 rows
```

In [17]:

```
data_train.printSchema()
```

```
root
 |-- id: integer (nullable = true)
 |-- week: integer (nullable = true)
 |-- checkout_price: float (nullable = true)
 |-- base_price: float (nullable = true)
 |-- email_for_promotion: integer (nullable = true)
 |-- homepage_featured: integer (nullable = true)
 |-- num_orders: integer (nullable = true)
```

```
 |-- category: string (nullable = true)
 |-- cuisine: string (nullable = true)
 |-- city_code: integer (nullable = true)
 |-- region_code: integer (nullable = true)
 |-- center_type: string (nullable = true)
 |-- op_area: float (nullable = true)
```

In [18]:
```
data_test.printSchema()
```

```
root
 |-- id: integer (nullable = true)
 |-- week: integer (nullable = true)
 |-- checkout_price: float (nullable = true)
 |-- base_price: float (nullable = true)
 |-- email_for_promotion: integer (nullable = true)
 |-- homepage_featured: integer (nullable = true)
 |-- category: string (nullable = true)
 |-- cuisine: string (nullable = true)
 |-- city_code: integer (nullable = true)
 |-- region_code: integer (nullable = true)
 |-- center_type: string (nullable = true)
 |-- op_area: float (nullable = true)
```

In [19]:
```
# Label coding for to convert each text category to numbers in order for the machine to

# Define a list of input columns and output columns
input_cols = ['category', 'cuisine', 'center_type']
output_cols = ['category_index', 'cuisine_index', 'center_type_index']

# Create a list of StringIndexer objects
indexers = [StringIndexer(inputCol=input_col, outputCol=output_col) for input_col, outpu

# Fit and transform the data using the list of StringIndexer objects
for indexer in indexers:
    data_train = indexer.fit(data_train).transform(data_train)

data_train.show()
```

```
[Stage 93:=============================================>             (3 + 1) / 4]
+-------+----+--------------+----------+------------------+-----------------+----------
+---------+-------+---------+-----------+-----------+-------+--------------+-----------
-+-----------------+
|     id|week|checkout_price|base_price|email_for_promotion|homepage_featured|num_orders
| category|cuisine|city_code|region_code|center_type|op_area|category_index|cuisine_inde
x|center_type_index|
+-------+----+--------------+----------+------------------+-----------------+----------
+---------+-------+---------+-----------+-----------+-------+--------------+-----------
-+-----------------+
|1215035|   1|        161.08|    161.08|                 0|                0|       324
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|           0.0|          0.
0|              1.0|
|1004523|   2|        174.66|    174.66|                 0|                0|       418
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|           0.0|          0.
0|              1.0|
|1315636|   3|        177.54|    178.54|                 0|                0|       337
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|           0.0|          0.
0|              1.0|
|1001462|   4|        178.54|    177.54|                 0|                0|       324
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|           0.0|          0.
0|              1.0|
|1381001|   5|        165.93|    164.93|                 0|                0|       338
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|           0.0|          0.
0|              1.0|
|1183853|   6|        155.26|    153.26|                 0|                0|       554
```

```
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1245045|   7|        159.14|    159.14|                  0|                0|       460
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1046050|   8|        149.41|    149.41|                  0|                0|       447
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1084268|   9|        154.23|    154.23|                  0|                0|       474
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1256757|  10|         156.2|     155.2|                  0|                0|       676
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1457239|  11|        155.26|    153.26|                  0|                0|       620
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1002295|  12|        168.84|    168.84|                  0|                0|       420
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1372744|  13|        172.69|    173.69|                  0|                0|       176
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1428509|  14|        178.54|    178.54|                  0|                0|       324
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1336073|  15|        172.69|    173.69|                  0|                0|       364
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1369877|  16|        173.63|    174.63|                  0|                0|       257
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1198059|  17|        153.26|    173.63|                  0|                0|       595
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1034336|  18|        149.38|    173.63|                  1|                1|       566
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1222578|  19|        151.32|    175.63|                  0|                1|       379
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
|1153637|  20|        151.32|    175.63|                  0|                1|       284
|Beverages|Italian|      515|         77|     TYPE_C|    3.0|            0.0|          0.
0|             1.0|
+-------+----+--------------+----------+------------------+-----------------+----------
+---------+-------+---------+-----------+-----------+-------+--------------+------------
-+-----------------+
only showing top 20 rows
```

In [20]:

```
# Drop columns 'category', 'cuisine', 'center_type' as they are not required for the fur
data_train = data_train.drop('category', 'cuisine', 'center_type')
data_train.show(5)
```

```
[Stage 104:==============>                                          (1 + 3) / 4]
+-------+----+--------------+----------+------------------+-----------------+----------
+---------+-----------+-------+--------------+-------------+-----------------+
|     id|week|checkout_price|base_price|email_for_promotion|homepage_featured|num_orders
|city_code|region_code|op_area|category_index|cuisine_index|center_type_index|
+-------+----+--------------+----------+------------------+-----------------+----------
+---------+-----------+-------+--------------+-------------+-----------------+
|1215035|   1|        161.08|    161.08|                  0|                0|       324
|      515|         77|    3.0|           0.0|          0.0|              1.0|
|1004523|   2|        174.66|    174.66|                  0|                0|       418
```

```
|      515|         77|    3.0|           0.0|          0.0|                1.0|                |
|1315636|   3|        177.54|    178.54|                   0|                  0|       337
|      515|         77|    3.0|           0.0|          0.0|                1.0|                |
|1001462|   4|        178.54|    177.54|                   0|                  0|       324
|      515|         77|    3.0|           0.0|          0.0|                1.0|                |
|1381001|   5|        165.93|    164.93|                   0|                  0|       338
|      515|         77|    3.0|           0.0|          0.0|                1.0|                |
+-------+----+-------------+----------+--------------------+-------------------+----------
+---------+-----------+-------+-------------+-------------+-----------------+
only showing top 5 rows
```

Splitting the Dataset into Train set and Test set

In [21]:
```
# assemble the feature columns into a single feature vector column named 'features'
assembler = VectorAssembler(inputCols=['week', 'checkout_price', 'base_price', 'email_fo
data_train = assembler.transform(data_train)
```

In [22]:
```
# split the data into training and test sets
train, test = data_train.randomSplit([0.7, 0.3], seed=42)
```

Model Building

- Train and test model algorithms
- Evaluation of Model
- Predicting the output using the model

Linear Regression

In [23]:
```
# train the linear regressor on the training data
lr = LinearRegression(featuresCol='features', labelCol='num_orders')
lr_model = lr.fit(train)

# Make predictions on the test set
predictions_lr = lr_model.transform(test)
```

```
[Stage 126:>                                                        (0 + 4) / 4]
23/04/11 16:57:55 WARN Instrumentation: [161d0946] regParam is zero, which might cause n
umerical instability and overfitting.
[Stage 131:>                                                        (0 + 4) / 4]
23/04/11 16:57:57 WARN InstanceBuilder$NativeBLAS: Failed to load implementation from:de
v.ludovic.netlib.blas.JNIBLAS
23/04/11 16:57:57 WARN InstanceBuilder$NativeBLAS: Failed to load implementation from:de
v.ludovic.netlib.blas.ForeignLinkerBLAS

23/04/11 16:58:01 WARN InstanceBuilder$NativeLAPACK: Failed to load implementation from:
dev.ludovic.netlib.lapack.JNILAPACK
```

In [24]:
```
# Evaluate the lr_model's performance
evaluator_lr = RegressionEvaluator(labelCol='num_orders', predictionCol='prediction', me
rmse_lr = evaluator_lr.evaluate(predictions_lr)
print('RMSE for Linear Regressor:', rmse_lr)
```

```
[Stage 153:============================>                            (2 + 2) / 4]
RMSE for Linear Regressor: 344.9834455470793
```

Random Forest Regressor

In [25]:
```
# A RandomForestRegressor model is created with 20 trees, a maximum depth of 5, and a ra

# train the random forest regressor on the training data
rf = RandomForestRegressor(featuresCol='features', labelCol='num_orders', numTrees=20, m
rf_model = rf.fit(train)

# Make predictions on the test set
predictions_rf = rf_model.transform(test)
```

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.util.SizeEstimator$ (file:/home/c
is6180/anaconda3/lib/python3.9/site-packages/pyspark/jars/spark-core_2.12-3.3.1.jar) to
field java.nio.charset.Charset.name
WARNING: Please consider reporting this to the maintainers of org.apache.spark.util.Size
Estimator$
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective acce
ss operations
WARNING: All illegal access operations will be denied in a future release
```

In [26]:
```
# The RegressionEvaluator is used to evaluate the predictions by calculating the root me
# between the predicted values and the actual values.
# The RMSE is printed to the console.

# Evaluate the rf_model's predictions
evaluator_rf = RegressionEvaluator(labelCol='num_orders', predictionCol='prediction', me
rmse_rf = evaluator_rf.evaluate(predictions_rf)

print(f"Root Mean Squared Error (RMSE) for Random Forest Regressor: {rmse_rf}")
```

```
[Stage 216:>                                                        (0 + 4) / 4]
Root Mean Squared Error (RMSE) for Random Forest Regressor: 281.85328071613617
```

In [ ]:
```

```