Food Demand Prediction

Importing dependencies

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

Importing the dataset

In [2]:

```
train = pd.read_csv('Analytics Vidya/train/train.csv')
test = pd.read_csv('Analytics Vidya/food_Demand_test.csv')
meal = pd.read_csv('Analytics Vidya/train/meal_info.csv')
centerinfo = pd.read_csv('Analytics Vidya/train/fulfilment_center_info.csv')
submission= pd.read_csv('Analytics Vidya/sample_submission.csv')
```

Analysing the Train data

The first 5 values of the data

In [3]:

```
train.head()
```

Out[3]:

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

The dimension of the data

In [4]:

```
train.shape
```

Out[4]:

```
(456548, 9)
```

The information of the columns in the data

In [5]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 456548 entries, 0 to 456547
Data columns (total 9 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   id                     456548 non-null  int64  
 1   week                   456548 non-null  int64  
 2   center_id              456548 non-null  int64  
 3   meal_id                456548 non-null  int64  
 4   checkout_price         456548 non-null  float64
 5   base_price             456548 non-null  float64
 6   emailer_for_promotion  456548 non-null  int64  
 7   homepage_featured      456548 non-null  int64  
 8   num_orders             456548 non-null  int64  
dtypes: float64(2), int64(7)
memory usage: 31.3 MB
```

Null value count

In [6]:

```
train.isnull().sum()
```

Out[6]:

```
id                       0
week                     0
center_id                0
meal_id                  0
checkout_price           0
base_price               0
emailer_for_promotion    0
homepage_featured        0
num_orders               0
dtype: int64
```

Unique value count

In [7]:

```
train.nunique()
```

Out[7]:

```
id                      456548
week                       145
center_id                   77
meal_id                     51
checkout_price            1992
base_price                1907
emailer_for_promotion        2
homepage_featured            2
num_orders                1250
dtype: int64
```

In [8]:

```
len(train[train['homepage_featured']==1])
```

Out[8]:

```
49855
```

In [9]:

```
len(train[train['emailer_for_promotion']==1])
```

Out[9]:

```
37050
```

In [10]:

```
train['checkout_price'].max()
```

Out[10]:

```
866.27
```

In [11]:

```
train['checkout_price'].min()
```

Out[11]:

```
2.97
```

In [12]:

```
train['base_price'].max()
```

Out[12]:

```
866.27
```

In [13]:

```
train['base_price'].min()
```

Out[13]:

```
55.35
```

In [14]:

```
train['num_orders'].max()
```

Out[14]:

```
24299
```

In [15]:

```
train['num_orders'].min()
```

Out[15]:

```
13
```

Analysing the Test data

The first 5 values of the data

In [16]:

```
test.head()
```

Out[16]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	home
0	1028232	146	55	1885	158.11	159.11	0	
1	1127204	146	55	1993	160.11	159.11	0	
2	1212707	146	55	2539	157.14	159.14	0	
3	1082698	146	55	2631	162.02	162.02	0	
4	1400926	146	55	1248	163.93	163.93	0	

The dimension of the data

In [17]:

```
test.shape
```

Out[17]:

```
(32573, 8)
```

The information of the columns in the data

In [18]:

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32573 entries, 0 to 32572
Data columns (total 8 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   id                     32573 non-null  int64  
 1   week                   32573 non-null  int64  
 2   center_id              32573 non-null  int64  
 3   meal_id                32573 non-null  int64  
 4   checkout_price         32573 non-null  float64
 5   base_price             32573 non-null  float64
 6   emailer_for_promotion  32573 non-null  int64  
 7   homepage_featured      32573 non-null  int64  
dtypes: float64(2), int64(6)
memory usage: 2.0 MB
```

Null value count

In [19]:

```
test.isnull().sum()
```

Out[19]:

```
id                       0
week                     0
center_id                0
meal_id                  0
checkout_price           0
base_price               0
emailer_for_promotion    0
homepage_featured        0
dtype: int64
```

Unique value count

In [20]:

```
test.nunique()
```

Out[20]:

```
id                       32573
week                        10
center_id                   77
meal_id                     51
checkout_price            1397
base_price                1179
emailer_for_promotion        2
homepage_featured            2
dtype: int64
```

In [21]:

```
len(test[test['homepage_featured']==1])
```

Out[21]:

```
2650
```

In [22]:

```
len(test[test['emailer_for_promotion']==1])
```

Out[22]:

```
2164
```

In [23]:

```
test['checkout_price'].max()
```

Out[23]:

```
1113.62
```

In [24]:

```
test['checkout_price'].min()
```

Out[24]:

```
67.9
```

In [25]:

```
test['base_price'].max()
```

Out[25]:

```
1112.62
```

In [26]:

```
test['base_price'].min()
```

Out[26]:

```
89.24
```

Analysing the meal_info data

The first 5 values of the data

In [27]:

```
meal.head()
```

Out[27]:

	meal_id	category	cuisine
0	1885	Beverages	Thai
1	1993	Beverages	Thai
2	2539	Beverages	Thai
3	1248	Beverages	Indian
4	2631	Beverages	Indian

The dimension of the dataset

In [28]:

```
meal.shape
```

Out[28]:

```
(51, 3)
```

The information of the columns in the data

In [29]:

```
meal.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   meal_id   51 non-null     int64 
 1   category  51 non-null     object
 2   cuisine   51 non-null     object
dtypes: int64(1), object(2)
memory usage: 1.3+ KB
```

Null value count

In [30]:

```
meal.isnull().sum()
```

Out[30]:

```
meal_id     0
category    0
cuisine     0
dtype: int64
```

Unique value count

In [31]:

```
meal.nunique()
```

Out[31]:

```
meal_id     51
category    14
cuisine      4
dtype: int64
```

In [32]:

```
meal['category'].unique()
```

Out[32]:

```
array(['Beverages', 'Extras', 'Soup', 'Other Snacks', 'Salad',
       'Rice Bowl', 'Starters', 'Sandwich', 'Pasta', 'Desert', 'Biryani',
       'Pizza', 'Fish', 'Seafood'], dtype=object)
```

In [33]:

```
meal['cuisine'].unique()
```

Out[33]:

```
array(['Thai', 'Indian', 'Italian', 'Continental'], dtype=object)
```

Analysing the fulfilment_center_info data

The first 5 values of the data

In [34]:

```
centerinfo.head()
```

Out[34]:

	center_id	city_code	region_code	center_type	op_area
0	11	679	56	TYPE_A	3.7
1	13	590	56	TYPE_B	6.7
2	124	590	56	TYPE_C	4.0
3	66	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.6

The dimension of the data

In [35]:

```
centerinfo.shape
```

Out[35]:

```
(77, 5)
```

The information of the columns in the data

In [36]:

```
centerinfo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 77 entries, 0 to 76
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   center_id    77 non-null     int64  
 1   city_code    77 non-null     int64  
 2   region_code  77 non-null     int64  
 3   center_type  77 non-null     object 
 4   op_area      77 non-null     float64
dtypes: float64(1), int64(3), object(1)
memory usage: 3.1+ KB
```

Null value count

In [37]:

```
centerinfo.isnull().sum()
```

Out[37]:

```
center_id      0
city_code      0
region_code    0
center_type    0
op_area        0
dtype: int64
```

Unique value count

In [38]:

```
centerinfo.nunique()
```

Out[38]:

```
center_id      77
city_code      51
region_code     8
center_type     3
op_area        30
dtype: int64
```

In [39]:

```
centerinfo['region_code'].unique()
```

Out[39]:

```
array([56, 34, 77, 85, 23, 71, 35, 93], dtype=int64)
```

In [40]:

```
centerinfo['center_type'].unique()
```

Out[40]:

```
array(['TYPE_A', 'TYPE_B', 'TYPE_C'], dtype=object)
```

Analysing the submission data

First 5 values in the data

In [41]:

```
submission.head()
```

Out[41]:

	id	num_orders
0	1028232	0
1	1127204	0
2	1212707	0
3	1082698	0
4	1400926	0

The dimension of the data

In [42]:

```
submission.shape
```

Out[42]:

```
(32573, 2)
```

The information of the columns in the data

In [43]:

```
submission.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32573 entries, 0 to 32572
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   id          32573 non-null  int64
 1   num_orders  32573 non-null  int64
dtypes: int64(2)
memory usage: 509.1 KB
```

Null value count

In [44]:

```
submission.isnull().sum()
```

Out[44]:

```
id            0
num_orders    0
dtype: int64
```

Unique value count

In [45]:

```
submission.nunique()
```

Out[45]:

```
id            32573
num_orders        1
dtype: int64
```

Combining to a single dataframe

Combining the training data

In [46]:

```
df = pd.merge(train, centerinfo, on='center_id')
df = pd.merge(df, meal, on='meal_id')
df.head()
```

Out[46]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	home
0	1379560	1	55	1885	136.83	152.29	0	
1	1018704	2	55	1885	135.83	152.29	0	
2	1196273	3	55	1885	132.92	133.92	0	
3	1116527	4	55	1885	135.86	134.86	0	
4	1343872	5	55	1885	146.50	147.50	0	

In [47]:

```
df.shape
```

Out[47]:

```
(456548, 15)
```

In [48]:

```
df.nunique()
```

Out[48]:

```
id                       456548
week                        145
center_id                    77
meal_id                      51
checkout_price             1992
base_price                 1907
emailer_for_promotion         2
homepage_featured             2
num_orders                 1250
city_code                    51
region_code                   8
center_type                   3
op_area                      30
category                     14
cuisine                       4
dtype: int64
```

Combinig the testing data

In [49]:

```
data= pd.merge(test, centerinfo, on='center_id')
data = pd.merge(data, meal, on='meal_id')
data.head()
```

Out[49]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	home
0	1028232	146	55	1885	158.11	159.11	0	
1	1262649	147	55	1885	159.11	159.11	0	
2	1453211	149	55	1885	157.14	158.14	0	
3	1262599	150	55	1885	159.14	157.14	0	
4	1495848	151	55	1885	160.11	159.11	0	

In [50]:

```
data.shape
```

Out[50]:

```
(32573, 14)
```

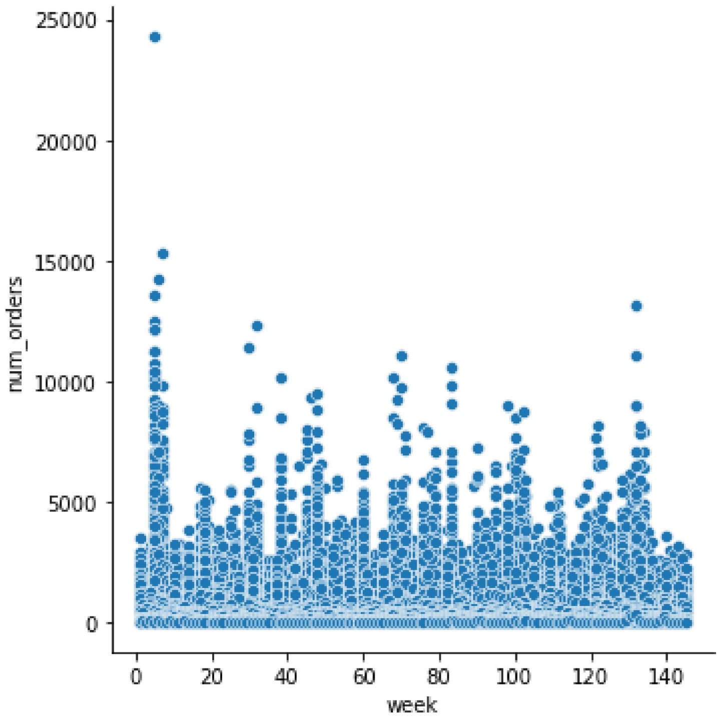
Visualization

In [51]:

```
sns.relplot(data=df,x='week',y='num_orders')
```

Out[51]:

```
<seaborn.axisgrid.FacetGrid at 0x1f112cfa400>
```

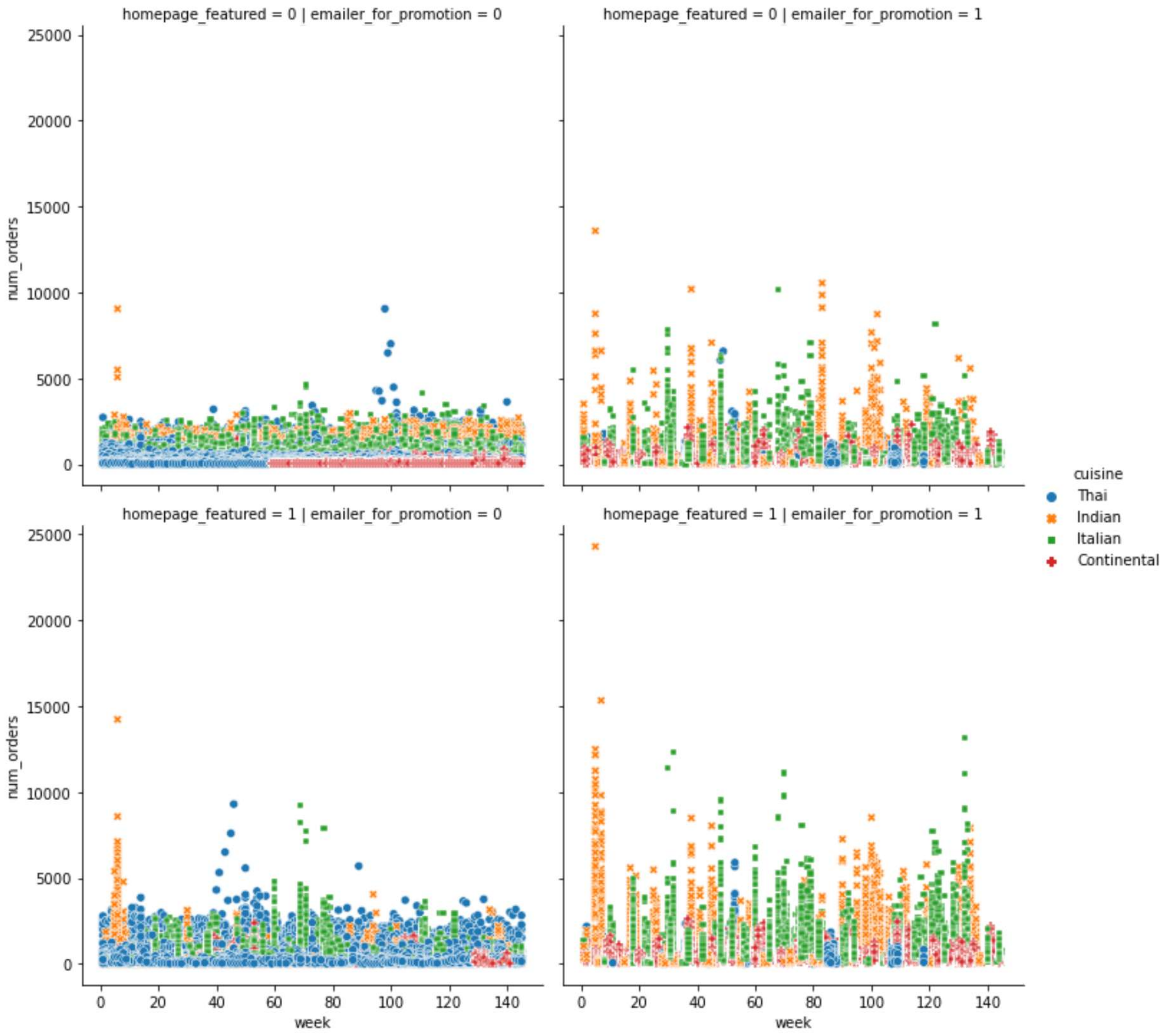


In [52]:

```
sns.relplot(data=df,x='week',y='num_orders',hue='cuisine',col='emailer_for_promotion',row='
```

Out[52]:

```
<seaborn.axisgrid.FacetGrid at 0x1f113cd7e80>
```

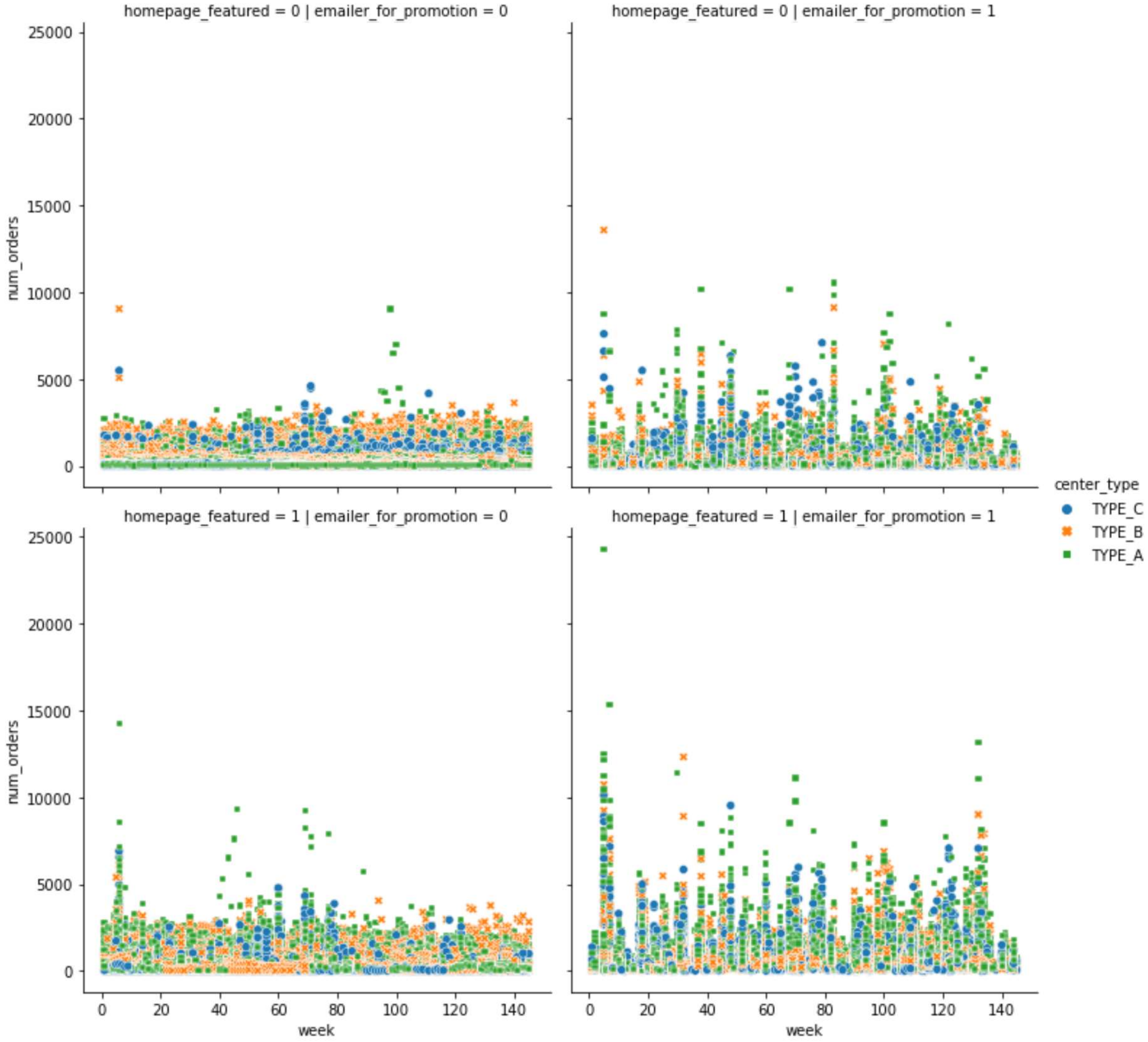


In [53]:

```
sns.relplot(data=df,x='week',y='num_orders',hue='center_type',col='emailer_for_promotion',r
            style='center_type')
```

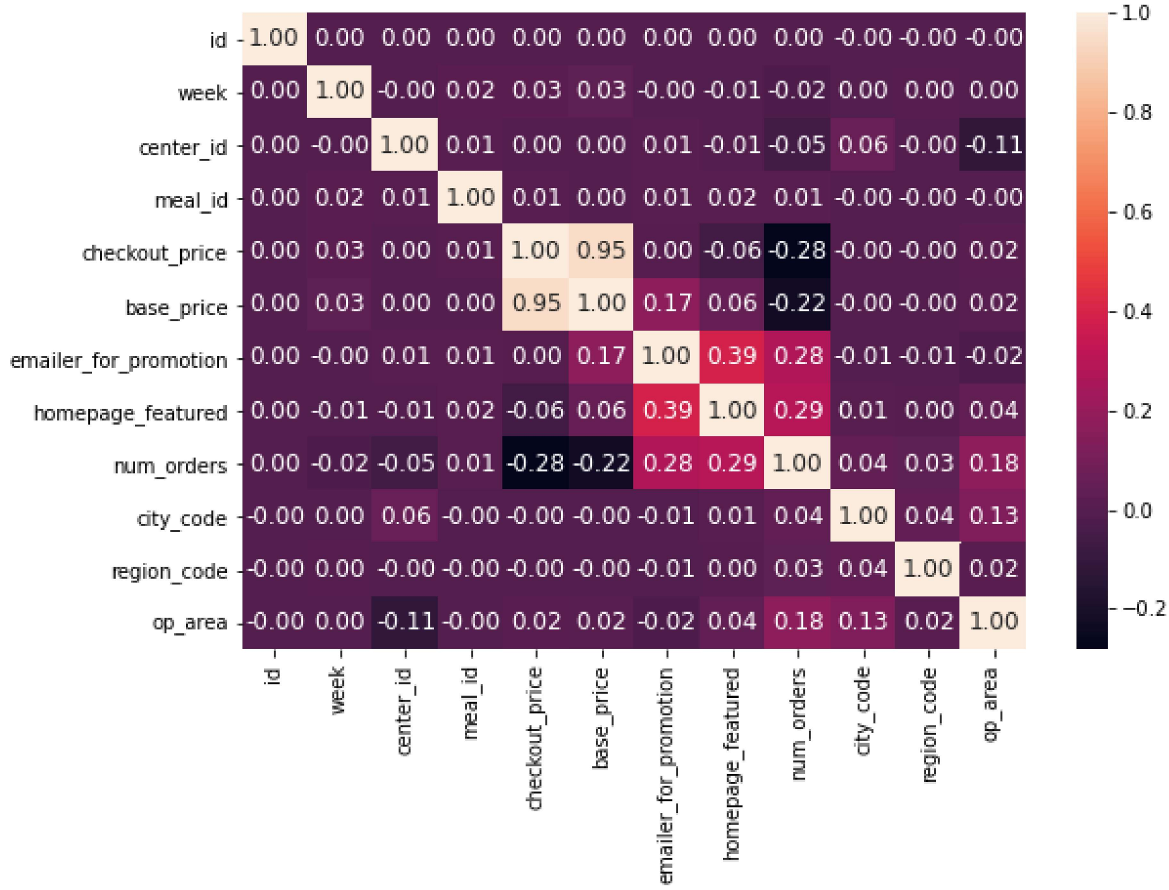
Out[53]:

```
<seaborn.axisgrid.FacetGrid at 0x1f113ff2640>
```



In [54]:

```
plt.subplots(figsize=(9, 6))
ax = sns.heatmap(df.corr(), annot=True,fmt='.2f',annot_kws={'size':'12'})
```



Seperating the data

Seperating the data for training

In [55]:

```
X=df.drop(['id','week','center_id','meal_id','checkout_price','base_price','center_type','n
X.head()
```

Out[55]:

	emailer_for_promotion	homepage_featured	city_code	region_code	op_area	category	cuis
0	0	0	647	56	2.0	Beverages	
1	0	0	647	56	2.0	Beverages	
2	0	0	647	56	2.0	Beverages	
3	0	0	647	56	2.0	Beverages	
4	0	0	647	56	2.0	Beverages	

In [56]:

```
y=df['num_orders']
y.head()
```

Out[56]:

```
0    177
1    323
2     96
3    163
4    215
Name: num_orders, dtype: int64
```

Seperating the data for testing

In [57]:

```
X_test=data.drop(['id','week','center_id','meal_id','checkout_price','base_price','center_t
X_test.head()
```

Out[57]:

	emailer_for_promotion	homepage_featured	city_code	region_code	op_area	category	cuis
0	0	0	647	56	2.0	Beverages	
1	0	0	647	56	2.0	Beverages	
2	0	0	647	56	2.0	Beverages	
3	0	0	647	56	2.0	Beverages	
4	0	0	647	56	2.0	Beverages	

Encoding the catergorical values

Encoding the training values

In [58]:

```
le = LabelEncoder()
```

In [59]:

```
X.category = le.fit_transform(X.category)
X.cuisine=le.fit_transform(X.cuisine)
```

In [60]:

```
X.head()
```

Out[60]:

	emailer_for_promotion	homepage_featured	city_code	region_code	op_area	category	cuisi
0	0	0	647	56	2.0	0	
1	0	0	647	56	2.0	0	
2	0	0	647	56	2.0	0	
3	0	0	647	56	2.0	0	
4	0	0	647	56	2.0	0	

Encoding the testing values

In [61]:

```
X_test.category = le.fit_transform(X_test.category)
X_test.cuisine=le.fit_transform(X_test.cuisine)
```

In [62]:

```
X_test.head()
```

Out[62]:

	emailer_for_promotion	homepage_featured	city_code	region_code	op_area	category	cuisi
0	0	0	647	56	2.0	0	
1	0	0	647	56	2.0	0	
2	0	0	647	56	2.0	0	
3	0	0	647	56	2.0	0	
4	0	0	647	56	2.0	0	

Training the model

Linear regression

In [63]:

```
lin_model=LinearRegression()
```

In [64]:

```
lin_model.fit(X,y)
```

Out[64]:

```
LinearRegression()
```

Accuracy of the training data

In [65]:

```
training_data_prediction=lin_model.predict(X)
```

In [66]:

```
rmse = mean_squared_error(y, training_data_prediction, squared=False)
rmse
```

Out[66]:

```
359.7591796243089
```

Testing the model

In [67]:

```
test_data_prediction=lin_model.predict(X_test)
```

In [68]:

```
test_data_prediction.shape
```

Out[68]:

```
(32573,)
```

In [69]:

```
submission.num_orders=test_data_prediction
```

In [70]:

```
submission.head()
```

Out[70]:

	id	num_orders
0	1028232	178.481579
1	1127204	178.481579
2	1212707	178.481579
3	1082698	178.481579
4	1400926	178.481579

In [71]:

```
submission['num_orders'].nunique()
```

Out[71]:

```
2113
```

In [ ]: