Food Demand Forecasting For Food Delivery Company Using IBM Cloud

<u>Food Demand Forecasting For Food Delivery Company Using IBM Cloud</u>

1. INTRODUCTION

1.1 Overview

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 majority of raw materials is done on weekly basis and since the raw material is perishable, the procurement planning is of utmost importance, the task is to predict the demand for the next 10 weeks.

1.2 Purpose

The main aim of this project is to create an appropriate machine learning model to forecast then number of orders to gather raw materials for next ten weeks. To achieve this, 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 classification algorithm to forecast the quantity for 10 weeks. A web application is built which is integrated with the model built.

.2. LITERATURE SURVEY

2.1 Existing problem

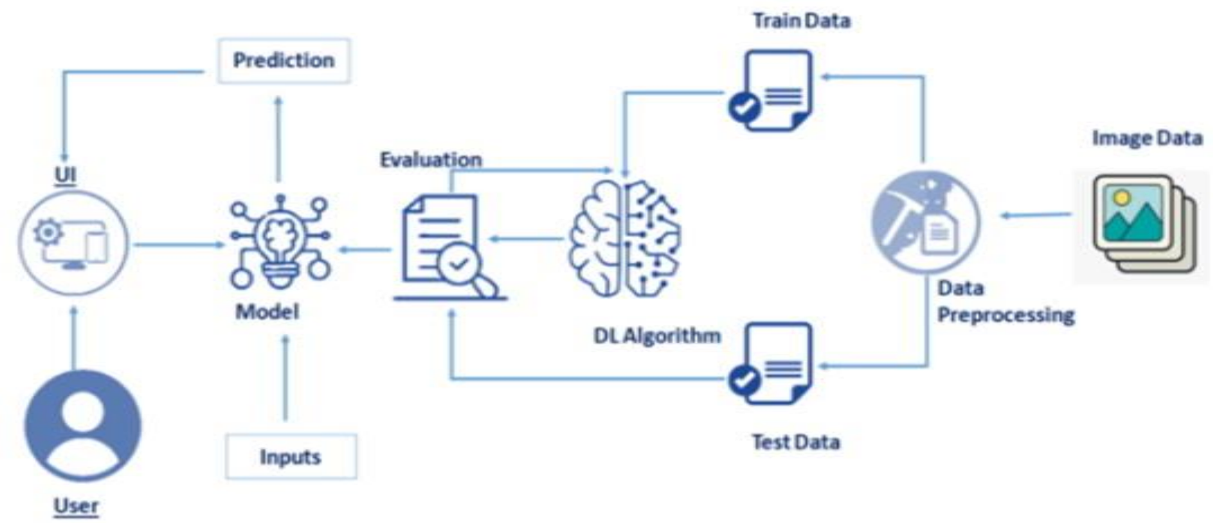
Food forecasting for food demand is an important role in world.Foodservice planning necessarily begins with a forecast of demand. Menu item demand forecasts are needed to make food item production decisions, work force and facility acquisition plans, and resource allocation and scheduling decisions. As these forecasts become more accurate, the tasks of adjusting original plans are minimized. Forecasting menu item demand need no longer be the tedious and inaccurate chore which is so prevalent in hospital food management systems today.

2.2 Proposed solution

Our aim is to map weather data to energy production. We wish to show that even data that is publicaly available for weather stations close to wind farms can be used to give a good prediction of the energy output . Furthermore, we examine the impact of different weather conditions on the energy output of technique to predict the energy output of wind farms. We are building an IBM Watson Auto AI Machine Learning technique to predict the energy output of wind turbine.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware / Software designing

Software Requirements:

- Anaconda Navigator
- Keras
- Flask

Hardware Requirements:

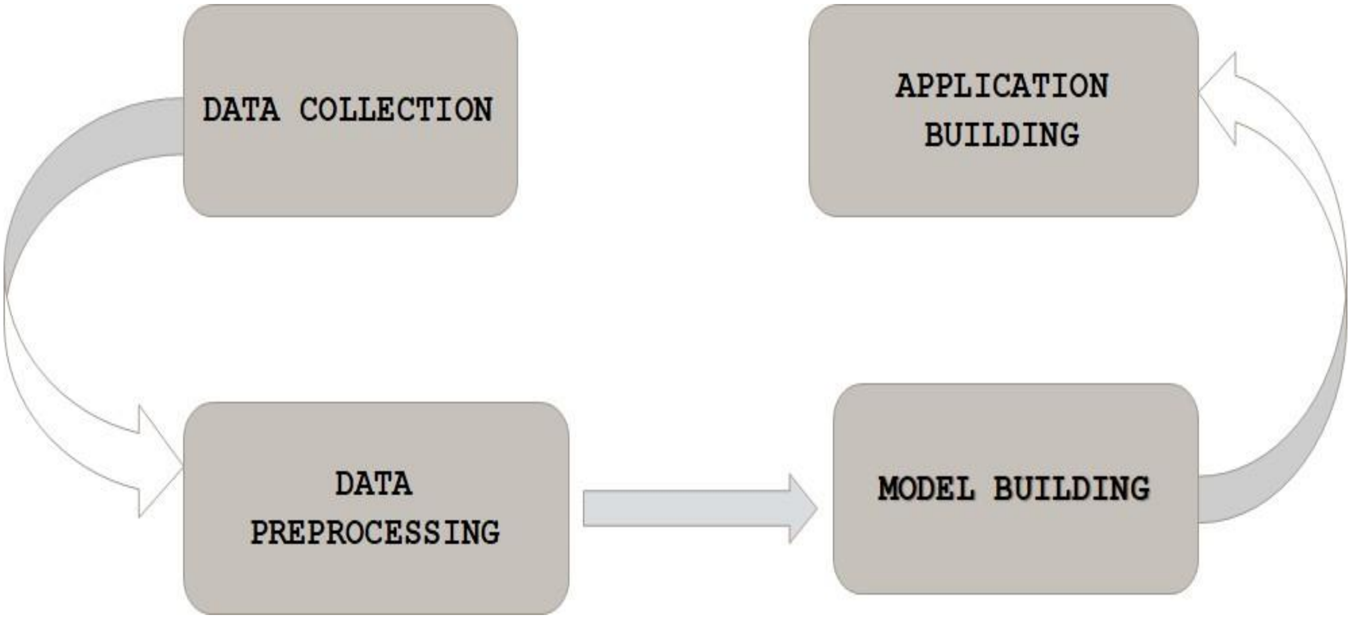
- Processor : Intel Core i3
- Hard Disk Space : Min 100 GB
- Ram : 8 GB
- Display : 14.1 “Color Monitor(LCD, CRT or LED)

Clock Speed : 1.67 GHz

4. EXPERIMENTAL INVESTIGATIONS

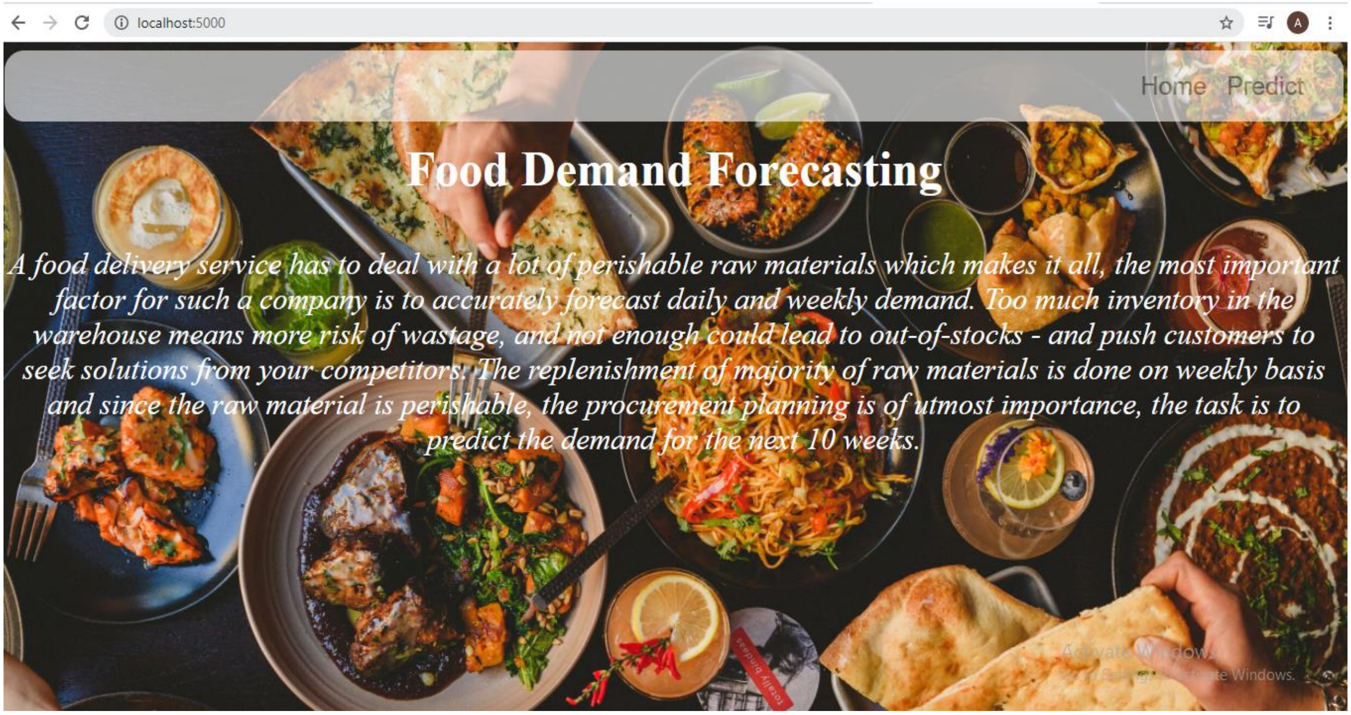
Demand forecasting is a key component to every growing online business. ... A food delivery service has to deal with a lot of perishable raw materials which makes it all the more important for such a company to accurately forecast daily and weekly demand..This project shows the accurate forecast the daily and weekly demand for a company.It is implemented by gathering information from various city,area 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 classification algorithm to forecast the quantity for 10 weeks.

5. FLOWCHART


DATA COLLECTION
APPLICATION BUILDING
DATA PREPROCESSING
MODEL BUILDING

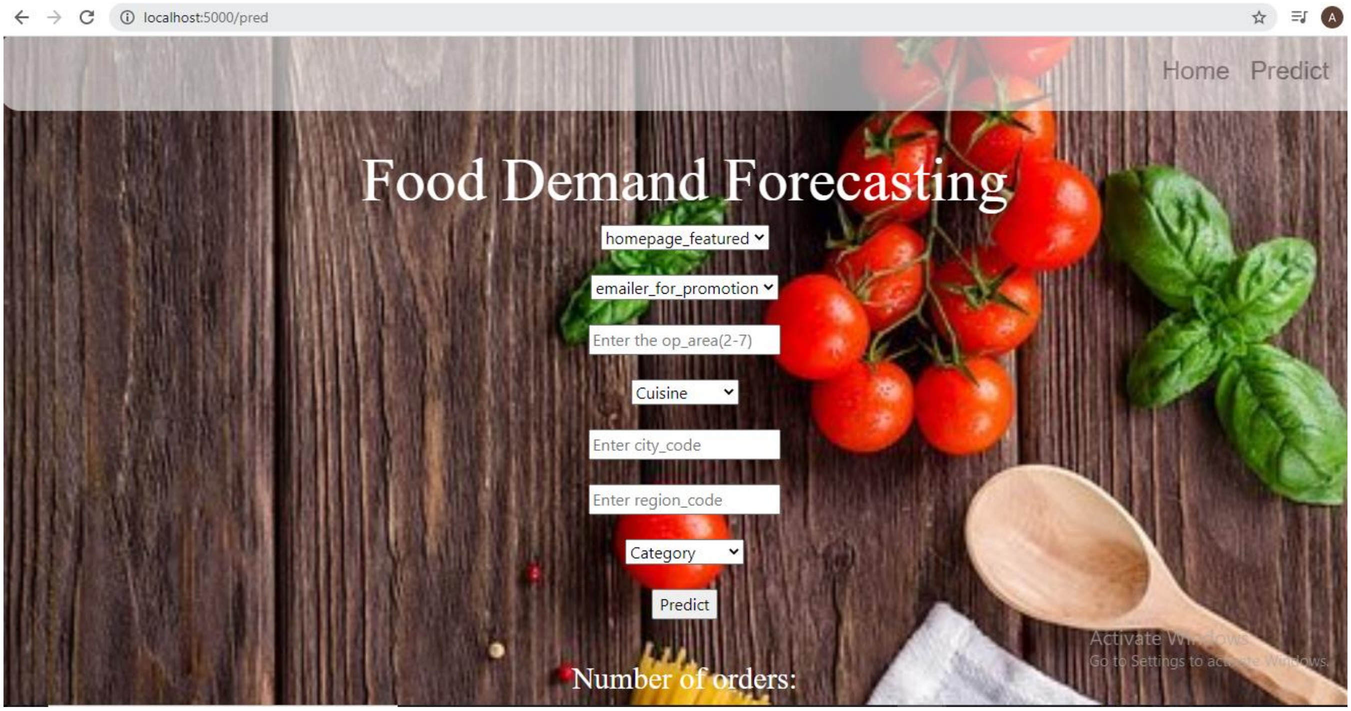
Food Demand Forecasting For Food Delivery Company Using IBM Cloud

6. RESULT



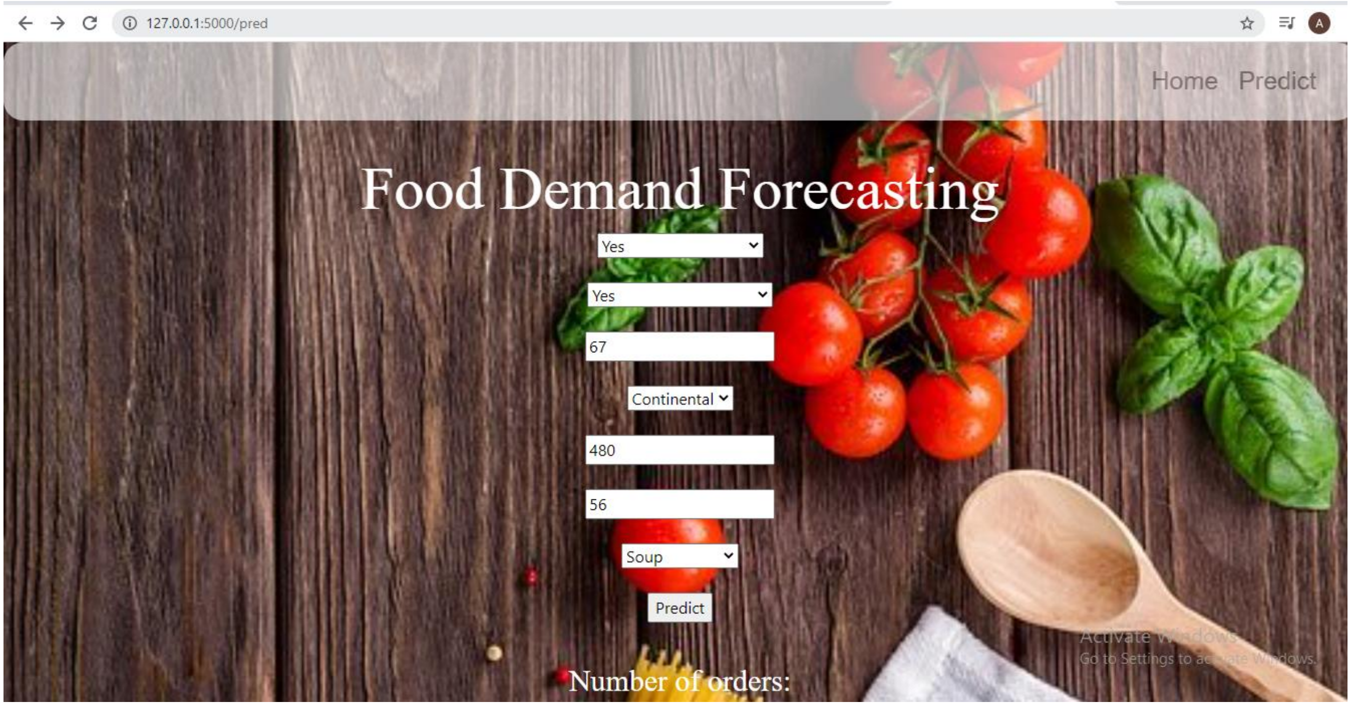
Food Demand Forecasting

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 majority of raw materials is done on weekly basis and since the raw material is perishable, the procurement planning is of utmost importance, the task is to predict the demand for the next 10 weeks.



Food Demand Forecasting

Number of orders:



127.0.0.1:5000/pred

Food Demand Forecasting

Yes

Yes

67

Continental

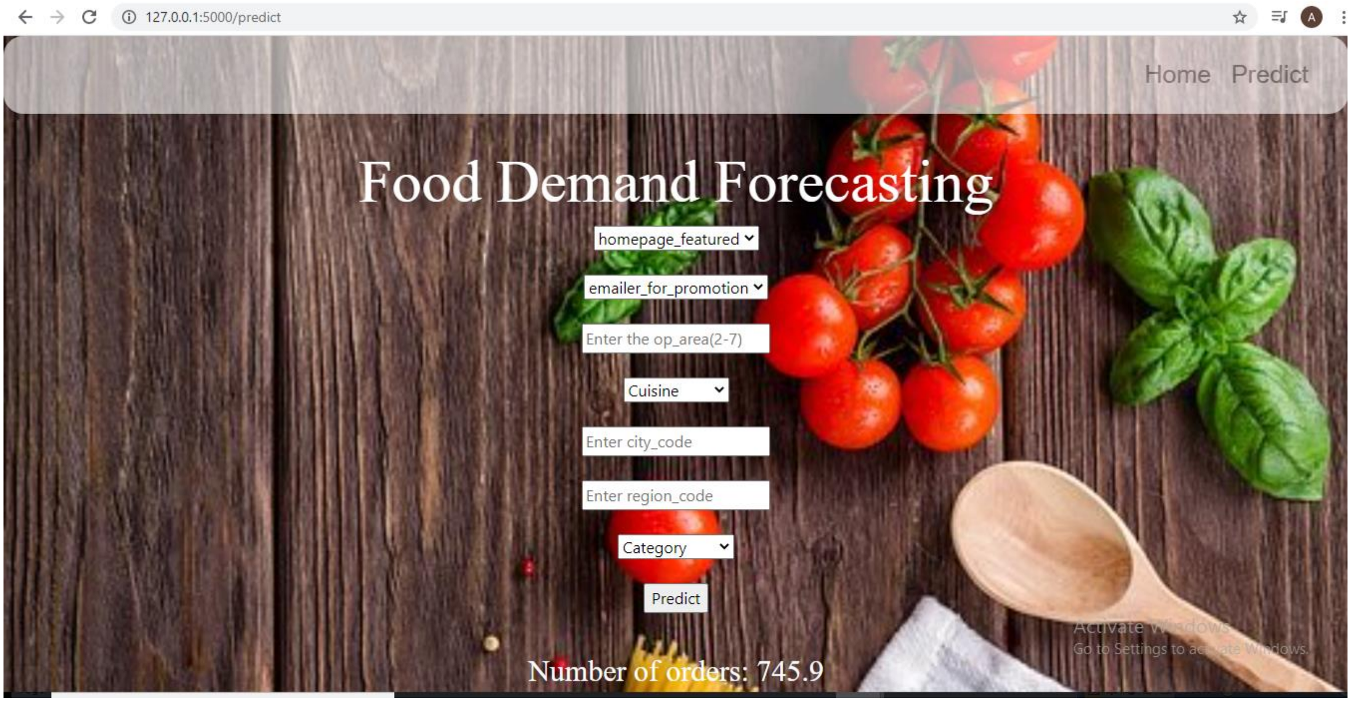
480

56

Soup

Predict

Number of orders:



127.0.0.1:5000/predict

Food Demand Forecasting

homepage_featured

emailer_for_promotion

Enter the op_area(2-7)

Cuisine

Enter city_code

Enter region_code

Category

Predict

Number of orders: 745.9

7. ADVANTAGES & DISADVANTAGES

Advantages:

- It limits the wastage of food and raw-materials .
- It will predict the consumed food orders in last 10 weeks sso that we can determines thats how much rawmatierials are kept to store .
- Simplicity improved accuracy,ease of use,identify etc.

Disadvantages:

- The challenges faced is this it can’t determine the quantity of food consumed it only know the number of orders.

8. APPLICATIONS

- Better Power Output Wind power forecasts are important in efficiently using wind turbines for generating power output.
- Efficient Predicting features like wind speed and wind direction can greatly help one to make decisions on when to switch on the wind turbine and when to switch it off(when it is assumed to not get the suitable conditions for generating power)
- Environment friendly If we are able to achieve predicting the wind power output, then it will open up more avenues for efficient power production in this field. This will lower the dependence on conventional sources of energy like coal which can cause harm to our environment.

9. CONCLUSION

In this project, we have established the application to predict number of orders of last 10 weeks in a place.So that orders of each food in a aea can be easily identifies and can be store raw-matierials correctly. Deploying it to the cloud makes it more scalable.

10. FUTURE SCOPE

Our attempt would be to further improve the predicting in quantity of food,How much quantity of food is delivered also provide page to customers to show the review about the project.so that we can provide the highest choosing food and their quality through that achieve the goal.

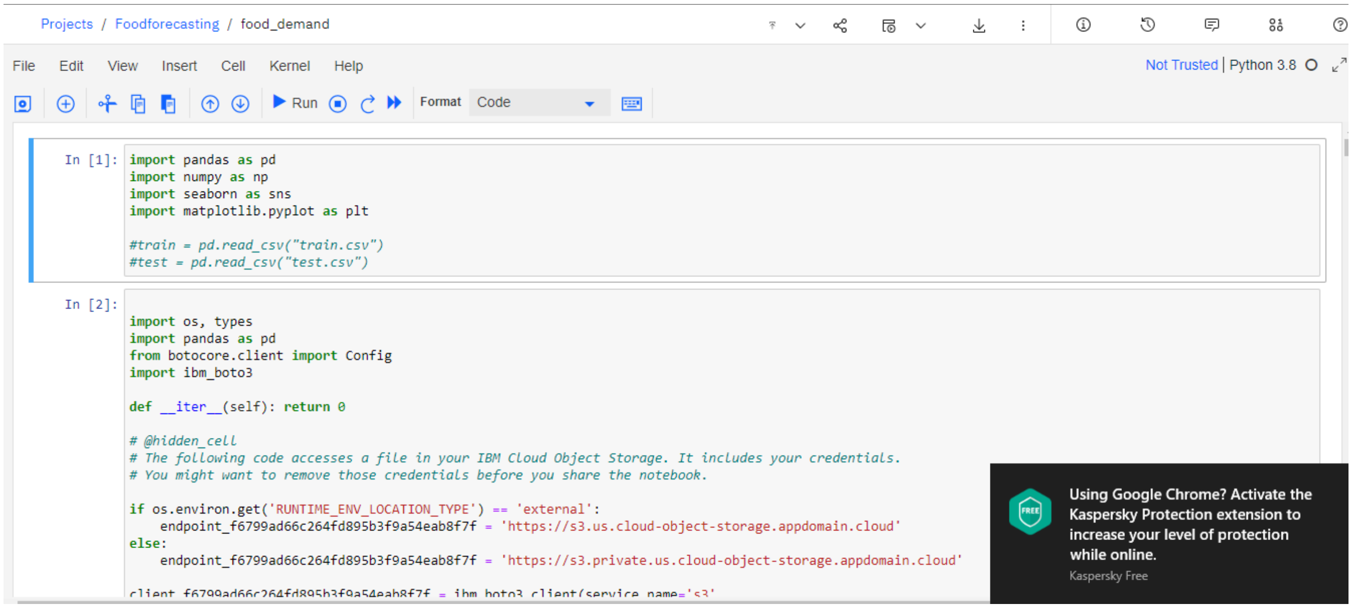
11. BIBILOGRAPHY

- Food demanding and forecasting survey sa flancis adora.
- Brower M 2012foodResource Assessment: A Practical Guide to Developing a demanding and forecasting Project (New York: Wiley).

Food Demand Forecasting For Food Delivery Company Using IBM Cloud

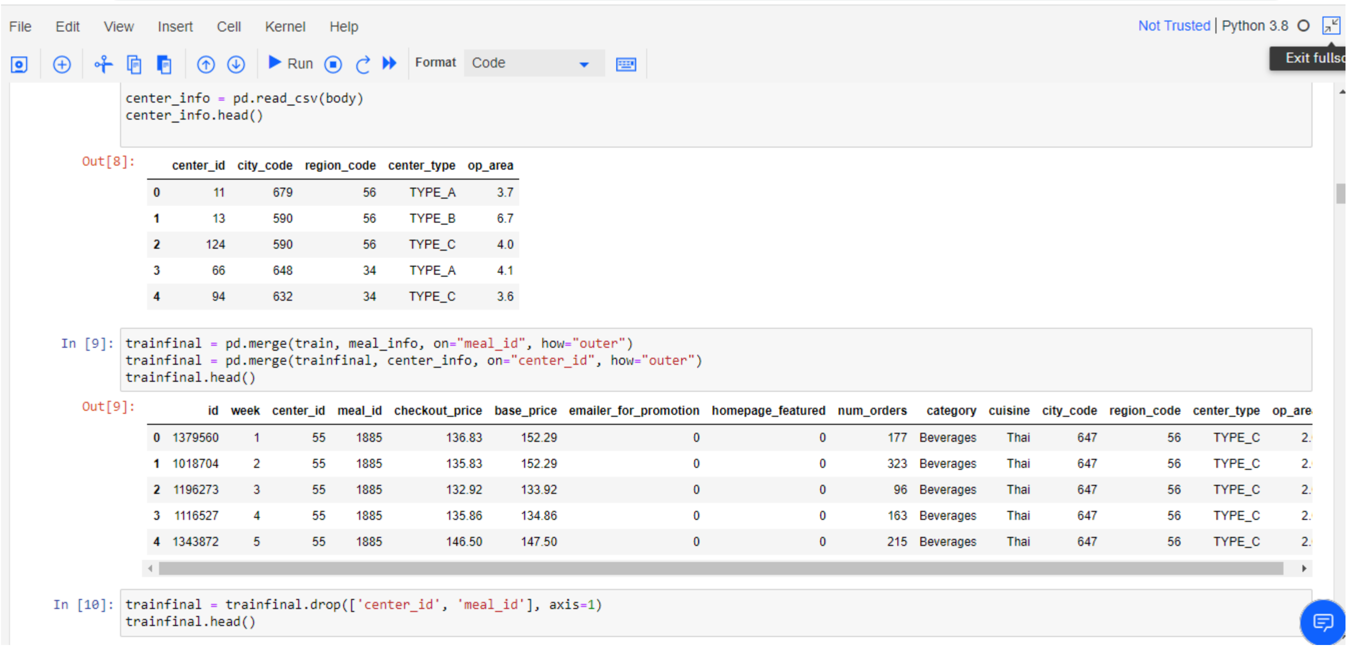
APPENDIX

Source Code



```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt

        #train = pd.read_csv("train.csv")
        #test = pd.read_csv("test.csv")
```

```
In [2]:
        import os, types
        import pandas as pd
        from botocore.client import Config
        import ibm_boto3

        def __iter__(self): return 0

        # @hidden_cell
        # The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
        # You might want to remove those credentials before you share the notebook.

        if os.environ.get('RUNTIME_ENV_LOCATION_TYPE') == 'external':
            endpoint_f6799ad66c264fd895b3f9a54eab8f7f = 'https://s3.us.cloud-object-storage.appdomain.cloud'
        else:
            endpoint_f6799ad66c264fd895b3f9a54eab8f7f = 'https://s3.private.us.cloud-object-storage.appdomain.cloud'

        client_f6799ad66c264fd895b3f9a54eab8f7f = ibm_boto3.client(service_name='s3'
```



```
center_info = pd.read_csv(body)
center_info.head()
```

Out[8]:

	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

In [9]:

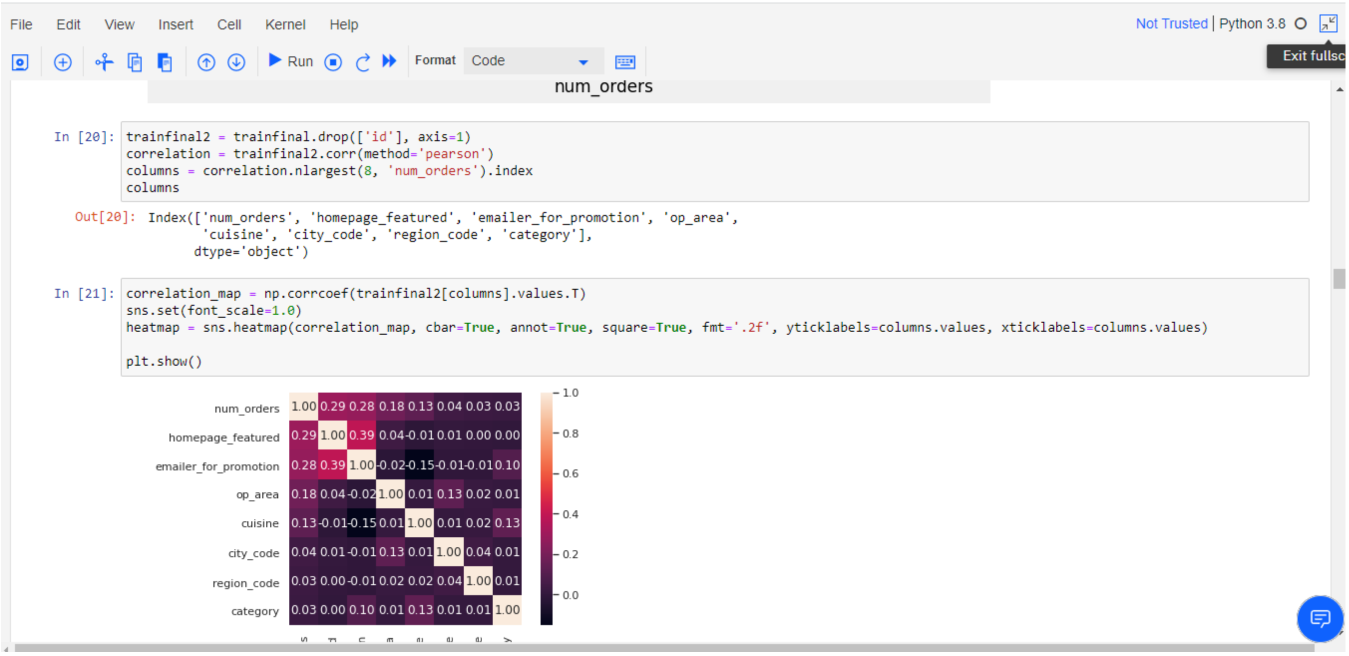
```
trainfinal = pd.merge(train, meal_info, on="meal_id", how="outer")
trainfinal = pd.merge(trainfinal, center_info, on="center_id", how="outer")
trainfinal.head()
```

Out[9]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders	category	cuisine	city_code	region_code	center_type	op_are
0	1379560	1	55	1885	136.83	152.29	0	0	177	Beverages	Thai	647	56	TYPE_C	2.
1	1018704	2	55	1885	135.83	152.29	0	0	323	Beverages	Thai	647	56	TYPE_C	2.
2	1196273	3	55	1885	132.92	133.92	0	0	96	Beverages	Thai	647	56	TYPE_C	2.
3	1116527	4	55	1885	135.86	134.86	0	0	163	Beverages	Thai	647	56	TYPE_C	2.
4	1343872	5	55	1885	146.50	147.50	0	0	215	Beverages	Thai	647	56	TYPE_C	2.

In [10]:

```
trainfinal = trainfinal.drop(['center_id', 'meal_id'], axis=1)
trainfinal.head()
```



In [20]:

```
trainfinal2 = trainfinal.drop(['id'], axis=1)
correlation = trainfinal2.corr(method='pearson')
columns = correlation.nlargest(8, 'num_orders').index
columns
```

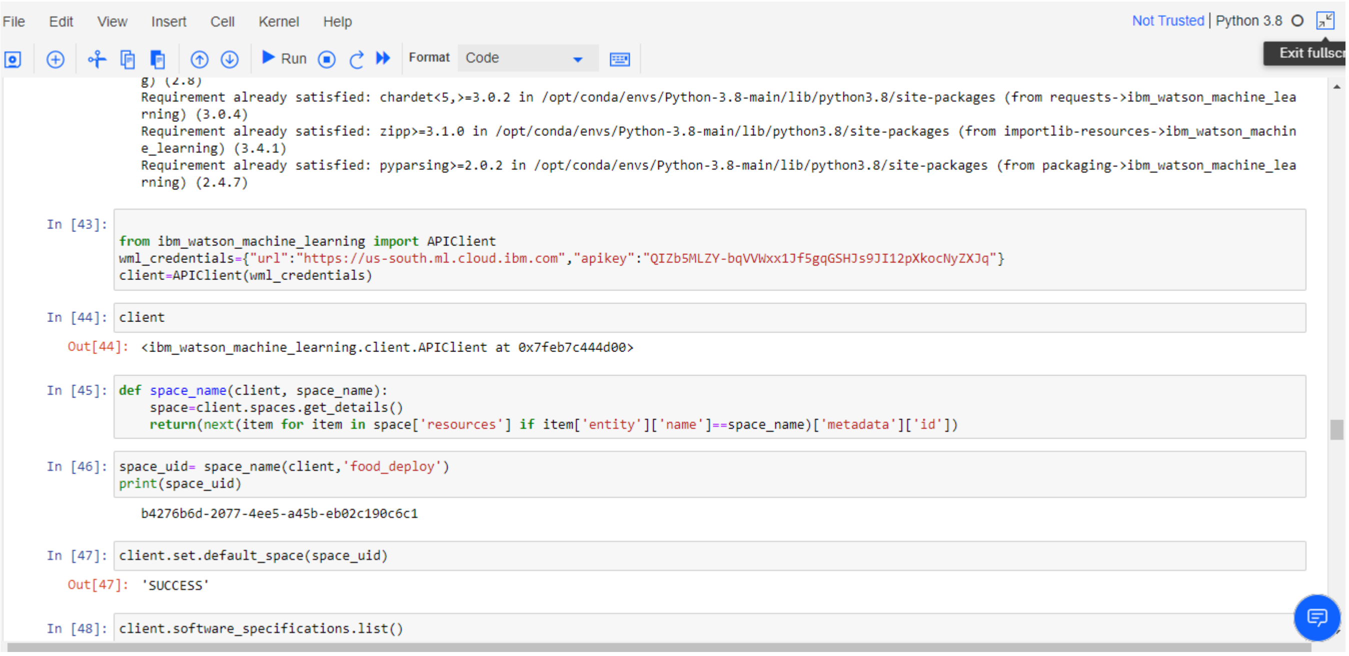
Out[20]:

```
Index(['num_orders', 'homepage_featured', 'emailer_for_promotion', 'op_area',
       'cuisine', 'city_code', 'region_code', 'category'],
      dtype='object')
```

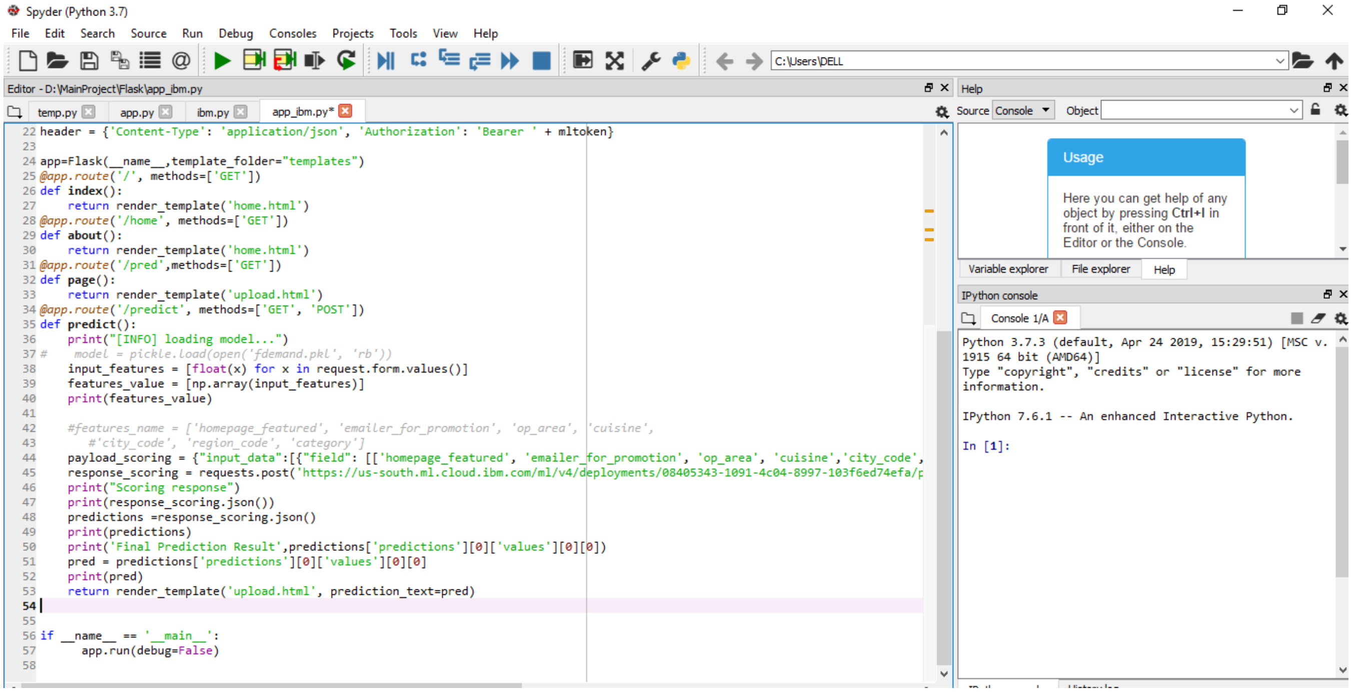
In [21]:

```
correlation_map = np.corrcoef(trainfinal2[columns].values.T)
sns.set(font_scale=1.0)
heatmap = sns.heatmap(correlation_map, cbar=True, annot=True, square=True, fmt='.2f', yticklabels=columns.values, xticklabels=columns.values)

plt.show()
```

num_orders 1.00 0.29 0.28 0.18 0.13 0.04 0.03 0.03
homepage_featured 0.29 1.00 0.39 0.04 -0.01 0.01 0.00 0.00
emailer_for_promotion 0.28 0.39 1.00 -0.02 -0.15 -0.01 -0.01 0.10
op_area 0.18 0.04 -0.02 1.00 0.01 0.13 0.02 0.01
cuisine 0.13 -0.01 -0.15 0.01 1.00 0.01 0.02 0.13
city_code 0.04 0.01 -0.01 0.13 0.01 1.00 0.04 0.01
region_code 0.03 0.00 -0.01 0.02 0.02 0.04 1.00 0.01
category 0.03 0.00 0.10 0.01 0.13 0.01 0.01 1.00
1.0
0.8
0.6
0.4
0.2
0.0



```
g) (2.8)
Requirement already satisfied: chardet<5,>=3.0.2 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from requests->ibm_watson_machine_lea
rning) (3.0.4)
Requirement already satisfied: zipp>=3.1.0 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from importlib-resources->ibm_watson_machin
e_learning) (3.4.1)
Requirement already satisfied: pyparsing>=2.0.2 in /opt/conda/envs/Python-3.8-main/lib/python3.8/site-packages (from packaging->ibm_watson_machine_lea
rning) (2.4.7)

In [43]:
from ibm_watson_machine_learning import APIClient
wml_credentials={"url":"https://us-south.ml.cloud.ibm.com","apikey":"QIZb5MLZY-bqVVWxx1Jf5gqGSHJs9JI12pXkocNyZXJq"}
client=APIClient(wml_credentials)

In [44]: client

Out[44]: <ibm_watson_machine_learning.client.APIClient at 0x7feb7c444d00>

In [45]: def space_name(client, space_name):
             space=client.spaces.get_details()
             return(next(item for item in space['resources'] if item['entity']['name']==space_name)['metadata']['id'])

In [46]: space_uid= space_name(client,'food_deploy')
         print(space_uid)

         b4276b6d-2077-4ee5-a45b-eb02c190c6c1

In [47]: client.set.default_space(space_uid)

Out[47]: 'SUCCESS'

In [48]: client.software_specifications.list()
```



```
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def about():
    return render_template('home.html')
@app.route('/pred',methods=['GET'])
def page():
    return render_template('upload.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    print("[INFO] loading model...")
#    model = pickle.load(open('fdemand.pkl', 'rb'))
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]
    print(features_value)

    #features_name = ['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
       #'city_code', 'region_code', 'category']
    payload_scoring = {"input_data":[{"field": [['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine','city_code',
    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/08405343-1091-4c04-8997-103f6ed74efa/p
    print("Scoring response")
    print(response_scoring.json())
    predictions =response_scoring.json()
    print(predictions)
    print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])
    pred = predictions['predictions'][0]['values'][0][0]
    print(pred)
    return render_template('upload.html', prediction_text=pred)


if __name__ == '__main__':
     app.run(debug=False)

```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

```
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.
1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more
information.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]:
```

Food Demand Forecasting For Food Delivery Company Using IBM Cloud