OPTIMIZED WAREHOUSE MANAGEMENT OF PERISHABLE GOODS FOR A FOOD DELIVERY COMPANY

A UG-PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY,

HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Submitted by

GUNDA ROHITH 18UK1A05A3

Under the esteemed guidance of

Mr.MANKALA SATISH

(Assistant Professor)

DEPARTMENT OF COMPUTER SCINECE&ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTU Hyderabad & Approved by AICTE,New Delhi) Bollikunta , Warangal – 506005

2018-2022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

WARANGAL

CERTIFICATE OF COMPLETION

UG PROJECT PHASE-1

This is to Certify that the Project entitled **"Optimized Warehouse Management of Perishable Goods For a Food Delivery Company "** is submitted by **GUNDA ROHITH(18UK1A05A3)** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science & Engineering** to **Jawaharlal Nehru Technological University, Hyderabad** during the academic year 2021- 2022.

Project Guide	**Head of the Department**
Mr. MANKALA SATISH	**Dr. R.NAVEEN KUMAR**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr. R. NAVEEN KUMAR,** Head, Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project phase-1.

We express heartfelt thanks to the guide, **Mr.M.SATISH** Assistant Professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project phase-1.

We express our sincere thanks and gratitude to **THE SMARTBRIDGE EDUCATIONAL SERVICES PRIVATE LIMITED,** for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experiencing throughout these.

GUNDA ROHITH **18UK1A05A3**

ABSTRACT

In recent years, food wastage becomes the major problem of the world and researchers indicate that 20-60% of the total production is lost in the food supply chain. Due to perishable nature and the products fresh food companies face more challenges throughout the supply chains. An order proposal is generated for all the products for a time period of a week by the integration of machine learning and loud and also taking into supply chain with some barriers such as supplier delivery times and also the maximum and minimum number of orders. The whole process of prediction is done using Decision tree regression algorithm. This paper focuses particularly on perishable goods and analyzed based on the accuracy of the training and testing data.

TABLE OF CONTENT

INTRODUCTION

- ## 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 solution 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, and this project optimized warehouse management of perishable goods for a food delivery company helps all the food delivery companies to forecast the raw materials for the next 10 weeks. Being computerized the system will be more helpful and provide companies at ease of stocking and keeping raw materials so that they are replenished enough according to needs and there is no shortage.

- ## 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 fulfillment 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.

LITERATURE SURVEY

- **EXISTING PROBLEM:**

1. The replenishment of majority of raw materials is done on weekly bias and since the raw material is perishable, the procurement planning is of utmost importance. Secondly, staffing of the centers is also one area where in accurate demand forecasts are really helpful. This is done through manual process.
2. The food and beverage industry is one of the most important sectors of any economy, with a significant participation in GDP index. The food delivery companies have been worried about investing in planning their operations, making use, mainly of forecasting methods in order to become more competitive in the market.
3. In the case of food industry, the seasonal and the short perish ability factors are a limitation to the maintenance of stocks, requiring a forecast with a high accuracy level

- **PROPOSED SOLUTION:**

The solution that we are proposing to this problem is a Machine Learning Model that analyzes the information as follows:-

1. To forecast the raw materials for next 10 weeks.
2. Assembling the data about fulfillment center like area, city etc., and meal information like category of food and should also have information about number of orders of a particular category in particular week.

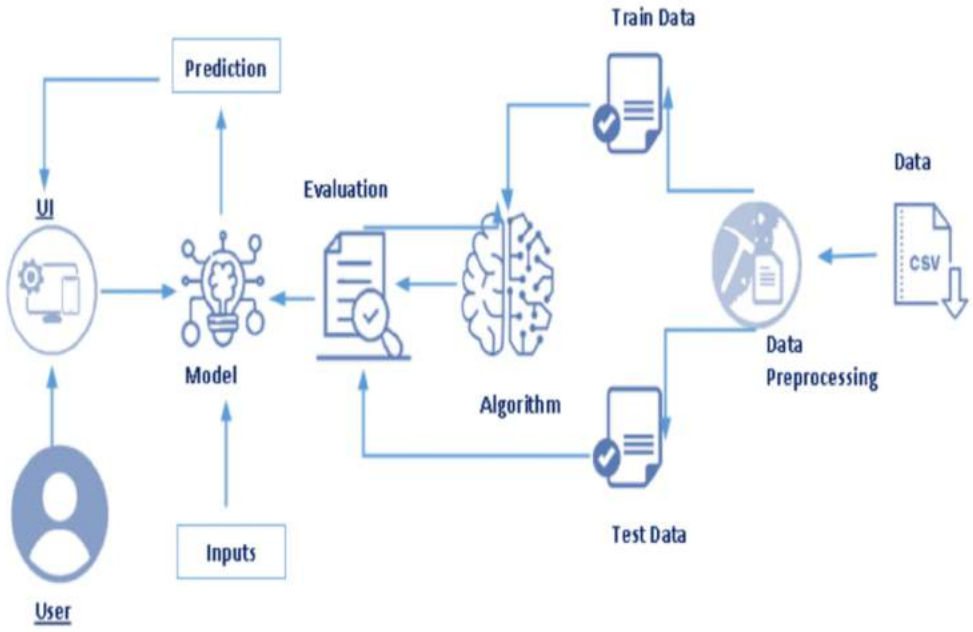
The algorithms used to train the Machine Learning Model are:-

1. Linear Regression
2. Decision Tree Regressor
3. K Neighbors Regressor
4. XGB Regressor
5. Gradient Boosting Regressor
6. Elasticity
7. Lasso

THEORETICAL ANALYSIS

The expansion in the volume of data and its diversity is the result of the bigger dataset. To manage and evaluate these new and potential valuable data sets, new methods and application have been developed in the form of predictive analytics. One of these methods is Machine Learning (ML). ML has helped us to predict the amount of goods and the retailer needs to store in his warehouse so that his perishable goods do not deteriorate. This was possible as a prediction is ideal for mobile application, websites and other application that use results interactively.

BLOCK DIAGRAM



SOFTWARE AND HARDWARE DESINGNING

- **Software used:-**
    1) Python
    2) Anaconda
    3) Jupyter notebook and spyder
    4) Windows 7 or higher
    5) Python flask

- **Hardware used:-**
    1) Processor-Dual core or higher
    2) Hard disk -50gb
    3) Memory – 4gb ram

CONCLUSION

This project not only concentrates for the benefit of retailers, but also customers. The retailers will know how much quantity of goods they must need to store in the warehouse and reduce the wastage of goods which in turn reduces their loss. At the same time, there will not be any shortage of goods to sell to the customers. The customers will also get good quality food products as the retailers buy the products as per the needs of the customers.

This will also improve the customer retailer relationship and will be a profit for both of them, in terms of money spent by on goods and quantity for food products. This model will enhance the sales thus this project will help and gives benefit both the retailers and customers. The prediction is highly reliable and it prevents wastage of valuable perishable food products.

FUTURE ENHANCEMENT

UG Project Phase -2 is the extension of UG Project Phase-1. UG Project Phase-2 involves all the coding and implementation of the design which we have retrieved from UG Project Phase-1. All the implementation is done and conclusion will be retrieved in this phase. We will also work on the applications, advantages, and disadvantages of the project in this phase. Future scope of the project will be also discussed in the UG Project Phase-2.

.

REFERENCES

- Alexandrov T., Bianconcini S., Dagum E. B., Maass P. and Mcelroy T. S. 2012 'A review of some modern approaches to the problem of trend extraction', Econometric Reviews, Vol. 31, pp. 593-624.
- Ballou R. H. 2007. 'The evolution and future of logistics and supply chain management', European business review, Vol. 19, pp. 332-48.
- Christo E., Ferreira M. and Alonso K. 2013. 'Use of Statistical Control for Improved Demand Forecasting', Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on. IEEE.
- De Oliveira Silva R., Da Silva Christo E. and Alonso Costa K. 2014. 'Analysis of Residual Autocorrelation in Forecasting Energy Consumption through a Java Program', Advanced Materials Research. Trans Tech Publ.
- Eksoz C., Mansouri S. A. and Bourlakis M. 2014 'Collaborative forecasting in the food supply chain: A conceptual framework', International journal of production economics, Vol. 158, pp. 120-35.

OPTIMIZED WAREHOUSE MANAGEMENT OF PERISHABLE GOODS FOR A FOOD DELIVERY COMPANY

A UG-PROJECT PHASE-2 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY,

HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Submitted by

GUNDA ROHITH 18UK1A05A3

Under the esteemed guidance of

Mr.MANKALA SATISH

(Assistant Professor)

DEPARTMENT OF COMPUTER SCINECE&ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTU Hyderabad & Approved by AICTE,New Delhi) Bollikunta , Warangal – 506005

2018-2022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

WARANGAL

CERTIFICATE OF COMPLETION

UG PROJECT PHASE-2

This is to Certify that the Project entitled **"Optimized Warehouse Management of Perishable Goods For a Food Delivery Company "** is submitted by **GUNDA ROHITH(18UK1A05A3)** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science & Engineering** to **Jawaharlal Nehru Technological University, Hyderabad** during the academic year 2021- 2022.

Project Guide **Head of the Department**

Mr. MANKALA SATISH **Dr. R.NAVEEN KUMAR**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-2 in the institute.

We extend our heartfelt thanks to **Dr. R. NAVEEN KUMAR,** Head, Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project phase-2.

We express heartfelt thanks to the guide, **Mr.M.SATISH** Assistant Professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project phase-2.

We express our sincere thanks and gratitude to **THE SMARTBRIDGE EDUCATIONAL SERVICES PRIVATE LIMITED,** for their constant supervision as well as for providing necessary information regarding the UG Project Phase-2 and for their support in completing the UG Project Phase-2

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experiencing throughout these.

GUNDA ROHITH **18UK1A05A3**

ABSTRACT

In recent years, food wastage becomes the major problem of the world and researchers indicate that 20-60% of the total production is lost in the food supply chain. Due to perishable nature and the products fresh food companies face more challenges throughout the supply chains. An order proposal is generated for all the products for a time period of a week by the integration of machine learning and loud and also taking into supply chain with some barriers such as supplier delivery times and also the maximum and minimum number of orders. The whole process of prediction is done using Decision tree regression algorithm. This paper focuses particularly on perishable goods and analyzed based on the accuracy of the training and testing data.

TABLE OF CONTENT

1. INTRODUCTION

As the population increases at a rapid rate the necessity to feed them, there is a requisite to reduce the tones of perishable waste. About one third of global fresh fruits and vegetables are thrown away due to their quality. The particular aspect that must be considered in the competitive management of fresh food supply chains is concern for the deterioration in product quality. Maintaining high food quality standards is important for product market value, customer happiness and in turn for the long-term reputation of the organization.

It has been a formidable task to handle the perishable food supply chain due to its short lifespan and the possibility of spoilage of the product due to its deterioration nature. All these components can cause a substantial amount of shortage of food items and retails loss.

With the use of machine learning and its algorithms we have created a way for the optimization of the goods so that the wastage of perishable goods can be minimized to maximum level. Machine learning and cloud both integrated together taking into account supply chain constraints, minimum or maximum order quantities can be predicted. This suggests for the required amount of perishable food products to be stored in the warehouse by the retailer for better sales.

UG Project Phase-2 involves all the coding and implementation of the design which we have retrieved from UG Project Phase-1. All the implementation is done and conclusion is retrieved in this phase. We will also work on the application, advantages, and disadvantages of the project in this phase. Future scope of the project will also be discussed in the UG Project Phase-2.

2. CODE SNIPPETS

- **MODEL CODE:**

Data preprocessing:-

Data preprocessing includes the following main tasks

1) Import the libraries
2) Importing the dataset
3) Checking for null values
4) Data visualization
5) Label encoding
6) Onehot encoding
7) Spitting data into train and test.

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

```
In [2]: train = pd.read_csv("train.csv")
        test = pd.read_csv("test.csv")
```

```
In [3]: train.head()
```

Out[3]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders
0	1379560	1	55	1885	136.83	152.29	0	0	177
1	1466964	1	55	1993	136.83	135.83	0	0	270
2	1346989	1	55	2539	134.86	135.86	0	0	189
3	1338232	1	55	2139	339.50	437.53	0	0	54
4	1448490	1	55	2631	243.50	242.50	0	0	40

```
In [4]: test.head()
```

Out[4]:

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

Figure 1:ipynb code describing importing libraries and displaying the few rows from the Dataset and loading the Dataset.

```
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
```

Figure 2: Here in the above figure it shows the info of the train dataset.

```
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
```

Figure 3: Here it shows the null values.

```
In [7]: train['num_orders'].describe()
```

```
Out[7]: count    456548.000000
        mean        261.872760
        std         395.922798
        min          13.000000
        25%          54.000000
        50%         136.000000
        75%         324.000000
        max       24299.000000
        Name: num_orders, dtype: float64
```

Figure 4: Describing the dataset.

```
In [8]: meal_info = pd.read_csv("meal_info.csv")
        center_info = pd.read_csv("fulfilment_center_info.csv")
```

```
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	regior
0	1379560	1	55	1885	136.83	152.29	0	0	177	Beverages	Thai	647	
1	1018704	2	55	1885	135.83	152.29	0	0	323	Beverages	Thai	647	
2	1196273	3	55	1885	132.92	133.92	0	0	96	Beverages	Thai	647	
3	1116527	4	55	1885	135.86	134.86	0	0	163	Beverages	Thai	647	
4	1343872	5	55	1885	146.50	147.50	0	0	215	Beverages	Thai	647	

Figure 5: Merging the columns and creating a dataset from it.

```
In [10]: trainfinal = trainfinal.drop(['center_id', 'meal_id'], axis=1)
         trainfinal.head()
```

Out[10]:

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

```
In [11]: cols = trainfinal.columns.tolist()
         print(cols)
```

```
['id', 'week', 'checkout_price', 'base_price', 'emailer_for_promotion', 'homepage_featured', 'num_orders', 'categor
y', 'cuisine', 'city_code', 'region_code', 'center_type', 'op_area']
```

```
In [12]: cols = cols[:2] + cols[9:] + cols[7:9] + cols[2:7]
         print(cols)
```

```
['id', 'week', 'city_code', 'region_code', 'center_type', 'op_area', 'category', 'cuisine', 'checkout_price', 'base_p
rice', 'emailer_for_promotion', 'homepage_featured', 'num_orders']
```

Figure 6: Dropping and printing columns.

```
In [13]: trainfinal = trainfinal[cols]
```

```
In [14]: trainfinal.dtypes
```

```
Out[14]: id                      int64
         week                    int64
         city_code               int64
         region_code             int64
         center_type            object
         op_area               float64
         category               object
         cuisine                object
         checkout_price        float64
         base_price            float64
         emailer_for_promotion   int64
         homepage_featured       int64
         num_orders              int64
         dtype: object
```

```
In [15]: from sklearn.preprocessing import LabelEncoder
```

```
In [16]: lb1 = LabelEncoder()
         trainfinal['center_type'] = lb1.fit_transform(trainfinal['center_type'])

         lb2 = LabelEncoder()
         trainfinal['category'] = lb1.fit_transform(trainfinal['category'])

         lb3 = LabelEncoder()
         trainfinal['cuisine'] = lb1.fit_transform(trainfinal['cuisine'])
```

Figure 7: Here from sklearn preprocessing we are importing the label encoder.

```
In [17]: trainfinal.head()
```

Out[17]:

	id	week	city_code	region_code	center_type	op_area	category	cuisine	checkout_price	base_price	emailer_for_promotion	homepage_featured	num
0	1379560	1	647	56	2	2.0	0	3	136.83	152.29	0	0	
1	1018704	2	647	56	2	2.0	0	3	135.83	152.29	0	0	
2	1196273	3	647	56	2	2.0	0	3	132.92	133.92	0	0	
3	1116527	4	647	56	2	2.0	0	3	135.86	134.86	0	0	
4	1343872	5	647	56	2	2.0	0	3	146.50	147.50	0	0	

```
In [18]: trainfinal.shape
Out[18]: (456548, 13)
```

Figure 8: Here we are previewing trainfinal dataset after importing the label encoder

```
In [19]: plt.style.use('fivethirtyeight')
         plt.figure(figsize=(12,7))
         sns.distplot(trainfinal.num_orders, bins = 25)
         plt.xlabel("num_orders")
         plt.ylabel("Number of Buyers")
         plt.title('num_orders Distribution')
```

```
/Users/gundarohith/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
Out[19]: Text(0.5, 1.0, 'num_orders Distribution')
```

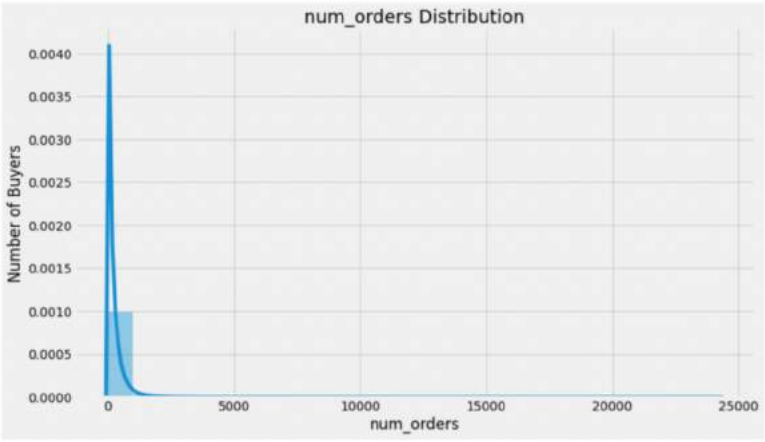


Figure 9: The above graph displaying num_orders Distribution

```
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')
```

Figure 10: Here we are dropping the id column and we are displaying the columns of trainfinal 2 dataset.

```
In [21]: plt.figure(figsize=(12,7))
         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()
```

Figure 11: This is the correlation map.

```
In [22]: features = columns.drop(['num_orders'])
         trainfinal3 = trainfinal[features]
         X = trainfinal3.values
         y = trainfinal['num_orders'].values

         from sklearn.model_selection import train_test_split
         X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.25)
```

```
In [23]: trainfinal3.head()
```

Out[23]:

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

Figure 12: Here we created the new dataset and we are displaying the values in the dataset.

```
In [24]: from sklearn.linear_model import LinearRegression
         from sklearn.linear_model import Lasso
         from sklearn.linear_model import ElasticNet
         from sklearn.tree import DecisionTreeRegressor
         from sklearn.neighbors import KNeighborsRegressor
         from sklearn.ensemble import GradientBoostingRegressor
```

```
In [25]: from xgboost import XGBRegressor
         XG = XGBRegressor()
         XG.fit(X_train, y_train)
         y_pred = XG.predict(X_val)
         y_pred[y_pred<0] = 0
         from sklearn import metrics
         print('RMSLE:', 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

         RMSLE: 69.39081014145997

In [26]: LR = LinearRegression()
         LR.fit(X_train, y_train)
         y_pred = LR.predict(X_val)
         y_pred[y_pred<0] = 0
         from sklearn import metrics
         print('RMSLE:', 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

         RMSLE: 130.1829172760739

In [27]: L = Lasso()
         L.fit(X_train, y_train)
         y_pred = L.predict(X_val)
         y_pred[y_pred<0] = 0
         from sklearn import metrics
         print('RMSLE:', 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

         RMSLE: 129.5878743345221

In [28]: EN = ElasticNet()
         EN.fit(X_train, y_train)
         y_pred = EN.predict(X_val)
         y_pred[y_pred<0] = 0
         from sklearn import metrics
         print('RMSLE:', 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

         RMSLE: 131.03199025037327

In [29]: DT = DecisionTreeRegressor()
         DT.fit(X_train, y_train)
         y_pred = DT.predict(X_val)
         y_pred[y_pred<0] = 0
         from sklearn import metrics
         print('RMSLE:', 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

         RMSLE: 62.84528220833866

In [30]: DT.get_params

Out[30]: <bound method BaseEstimator.get_params of DecisionTreeRegressor()>

In [31]: KNN = KNeighborsRegressor()
         KNN.fit(X_train, y_train)
         y_pred = KNN.predict(X_val)
         y_pred[y_pred<0] = 0
         from sklearn import metrics
         print('RMSLE:', 100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

         RMSLE: 66.8546384062411

In [44]: GB = GradientBoostingRegressor()
         GB.fit(X_train, y_train)
         y_pred = GB.predict(X_val)
         y_pred[y_pred<0] = 0
         from sklearn import metrics
         print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

         RMSLE: 94.98938089649886
```

Figure 13: Here we are importing the algorithms and also we are checking the RMSLE(Root Mean Square Log Error) value.

```
In [33]: testfinal = pd.merge(test, meal_info, on="meal_id", how="outer")
         testfinal = pd.merge(testfinal, center_info, on="center_id", how="outer")
         testfinal = testfinal.drop(['meal_id', 'center_id'], axis=1)

         tcols = testfinal.columns.tolist()
         print(tcols)

         ['id', 'week', 'checkout_price', 'base_price', 'emailer_for_promotion', 'homepage_featured', 'category', 'cuisine',
         'city_code', 'region_code', 'center_type', 'op_area']

In [34]: tcols = tcols[:2] + tcols[8:] + tcols[6:8] + tcols[2:6]
         testfinal = testfinal[tcols]

         lb1 = LabelEncoder()
         testfinal['center_type'] = lb1.fit_transform(testfinal['center_type'])

         lb2 = LabelEncoder()
         testfinal['category'] = lb1.fit_transform(testfinal['category'])

         lb3 = LabelEncoder()
         testfinal['cuisine'] = lb1.fit_transform(testfinal['cuisine'])

         testfinal.head()
```

Out[34]:

	id	week	city_code	region_code	center_type	op_area	category	cuisine	checkout_price	base_price	emailer_for_promotion	homepage_featured
0	1028232	146	647	56	2	2.0	0	3	158.11	159.11	0	0
1	1262649	147	647	56	2	2.0	0	3	159.11	159.11	0	0
2	1453211	149	647	56	2	2.0	0	3	157.14	158.14	0	0
3	1262599	150	647	56	2	2.0	0	3	159.14	157.14	0	0
4	1495848	151	647	56	2	2.0	0	3	160.11	159.11	0	0

```
In [35]: X_test = testfinal[features].values

In [36]: features

Out[36]: Index(['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
                'city_code', 'region_code', 'category'],
               dtype='object')
```

Figure 14: We are testing our test dataset.

```
In [37]: pred = DT.predict(X_test)
         pred[pred<0] = 0
         submit = pd.DataFrame({
             'id' : testfinal['id'],
             'num_orders' : pred
         })

In [38]: submit.to_csv("submission.csv", index=False)

In [39]: submit.describe()
```

Out[39]:

	id	num_orders
count	3.257300e+04	32573.000000
mean	1.248476e+06	262.951486
std	1.441580e+05	365.173579
min	1.000085e+06	15.350877
25%	1.123969e+06	64.055336
50%	1.247296e+06	149.465385
75%	1.372971e+06	322.640152
max	1.499996e+06	6029.090909

Figure 15: By using decision tree we are predicting.

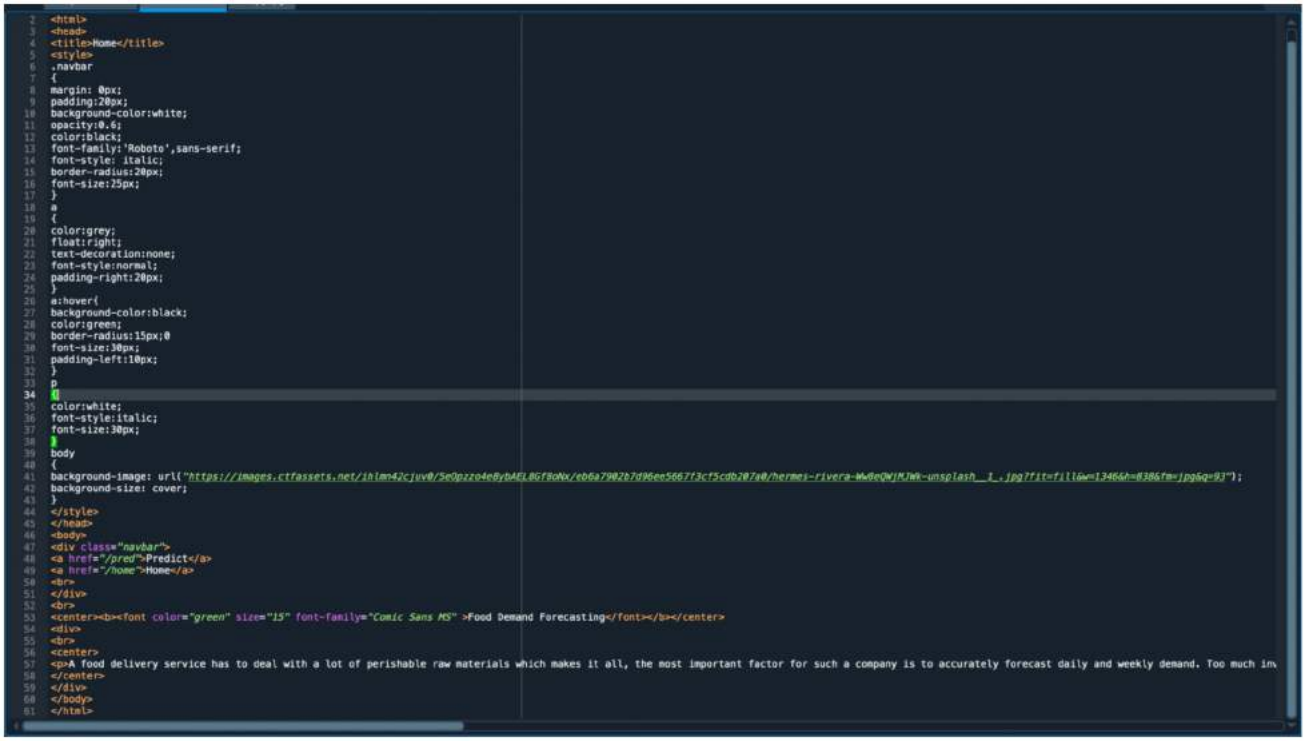
```
In [40]: import pickle

In [41]: pickle.dump(DT,open('fdemand.pkl','wb'))
```

Figure 16: Here we are creating the pickle file and dumping it into pkl file.

- **HTML CODE AND PYTHON CODE:**

1. Home.html:



```
<html>
<head>
<title>Home</title>
<style>
.navbar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:green;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
p
{
color:white;
font-style:italic;
font-size:30px;
}
body
{
background-image: url("https://images.ctfassets.net/inlmn42cjuv0/5eOpzzo4eBybAELBGf8oNx/eb6a7902b7d96ee5667f3cf5cdb207a0/hermesi-rivera-Ww8eQWjMJWk-unsplash__1_.jpg?fit=fill&w=1346&h=838&fm=jpg&q=93");
background-size: cover;
}
</style>
</head>
<body>
<div class="navbar">
<a href="/pred">Predict</a>
<a href="/home">Home</a>
<br>
</div>
<br>
<center><b><font color="green" size="15" font-family="Comic Sans MS" >Food Demand Forecasting</font></b></center>
<div>
<br>
<center>
<p>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 in
</center>
</div>
</body>
</html>
```

Figure 17: The html code used for Home page.

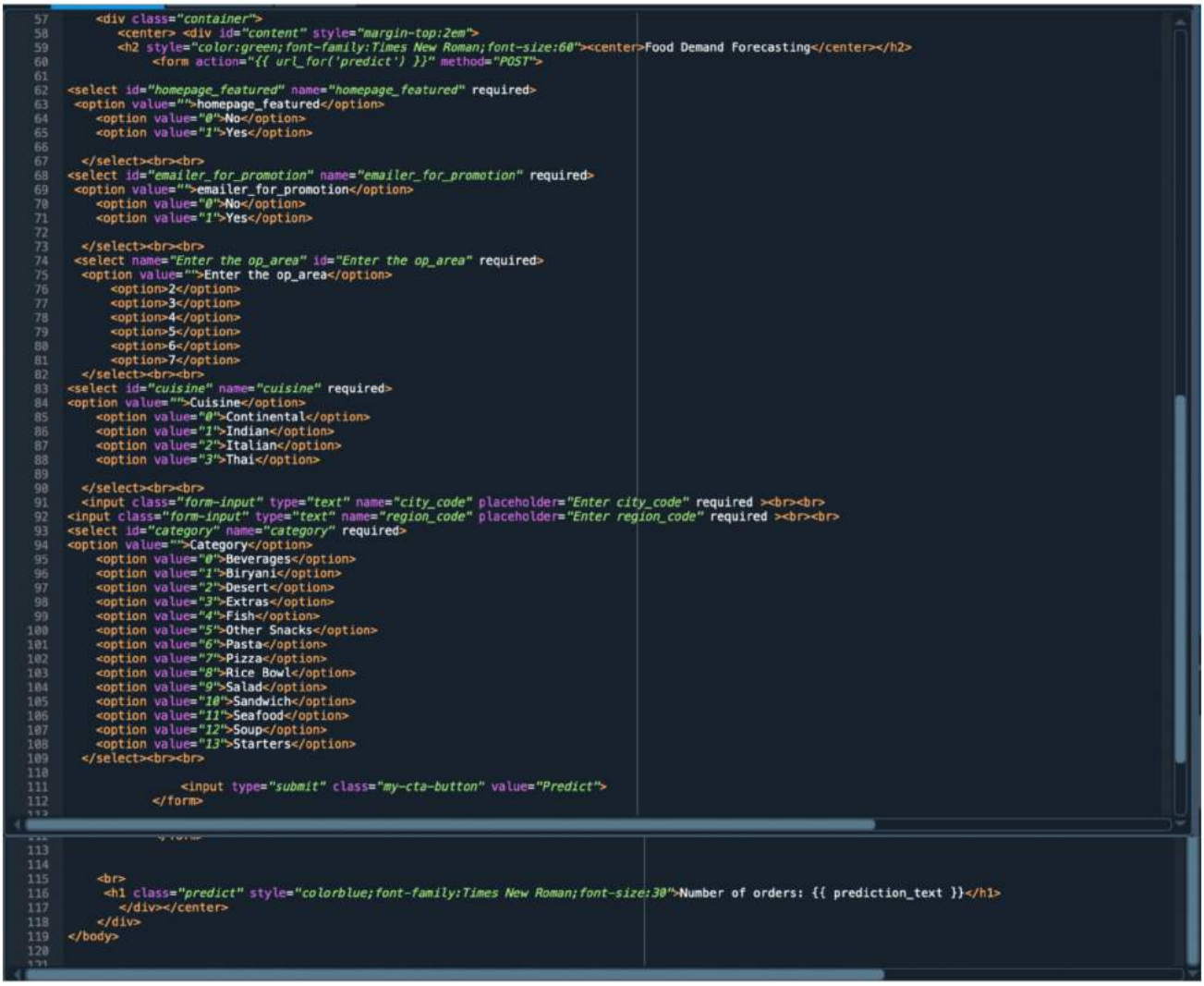
2. Predict.html:

```
<html lang="en">

<head>
    <title>Predict</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
<style>
.bar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:green;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
body
{
    background-image: url("https://images.ctfassets.net/ihlmn42cjuv0/5eOpzzo4e8ybAEL8Gf8oNx/eb6a7902b7d96ee5667f3cf5cdb207a0/hermes-rivera-Ww8eQWjMJWk-u
    background-size: cover;
}
p
{
color:red;
font-style:italic;
font-size:30px;
}
</style>
</head>

<body>

<div class="bar">
<a href="/pred">Predict</a>
<a href="/home">Home</a>
<br>
</div>
```



```
    <div class="container">
       <center> <div id="content" style="margin-top:2em">
       <h2 style="color:green;font-family:Times New Roman;font-size:60"><center>Food Demand Forecasting</center></h2>
            <form action="{{ url_for('predict') }}" method="POST">

<select id="homepage_featured" name="homepage_featured" required>
 <option value="">homepage_featured</option>
    <option value="0">No</option>
    <option value="1">Yes</option>

  </select><br><br>
<select id="emailer_for_promotion" name="emailer_for_promotion" required>
 <option value="">emailer_for_promotion</option>
    <option value="0">No</option>
    <option value="1">Yes</option>

  </select><br><br>
 <select name="Enter the op_area" id="Enter the op_area" required>
  <option value="">Enter the op_area</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
      <option>5</option>
      <option>6</option>
      <option>7</option>
  </select><br><br>
<select id="cuisine" name="cuisine" required>
<option value="">Cuisine</option>
    <option value="0">Continental</option>
    <option value="1">Indian</option>
    <option value="2">Italian</option>
    <option value="3">Thai</option>

  </select><br><br>
  <input class="form-input" type="text" name="city_code" placeholder="Enter city_code" required ><br><br>
<input class="form-input" type="text" name="region_code" placeholder="Enter region_code" required ><br><br>
<select id="category" name="category" required>
<option value="">Category</option>
    <option value="0">Beverages</option>
    <option value="1">Biryani</option>
    <option value="2">Desert</option>
    <option value="3">Extras</option>
    <option value="4">Fish</option>
    <option value="5">Other Snacks</option>
    <option value="6">Pasta</option>
    <option value="7">Pizza</option>
    <option value="8">Rice Bowl</option>
    <option value="9">Salad</option>
    <option value="10">Sandwich</option>
    <option value="11">Seafood</option>
    <option value="12">Soup</option>
    <option value="13">Starters</option>
  </select><br><br>

              <input type="submit" class="my-cta-button" value="Predict">
           </form>

    <br>
     <h1 class="predict" style="colorblue;font-family:Times New Roman;font-size:30">Number of orders: {{ prediction_text }}</h1>
      </div></center>
    </div>
</body>
```

Figure 18: The html code used for predict page**.**

3. app.py code:

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Jan 27 08:57:55 2021

@author: rincy
"""

# import the necessary packages
import pandas as pd
import numpy as np
import pickle
import os
from flask import Flask,request, render_template
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']
    prediction = model.predict(features_value)
    output=prediction[0]
    print(output)
    return render_template('upload.html', prediction_text=output)


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

Figure 19: Python code used for rendering all the HTML pages.

4. Execution of python code:

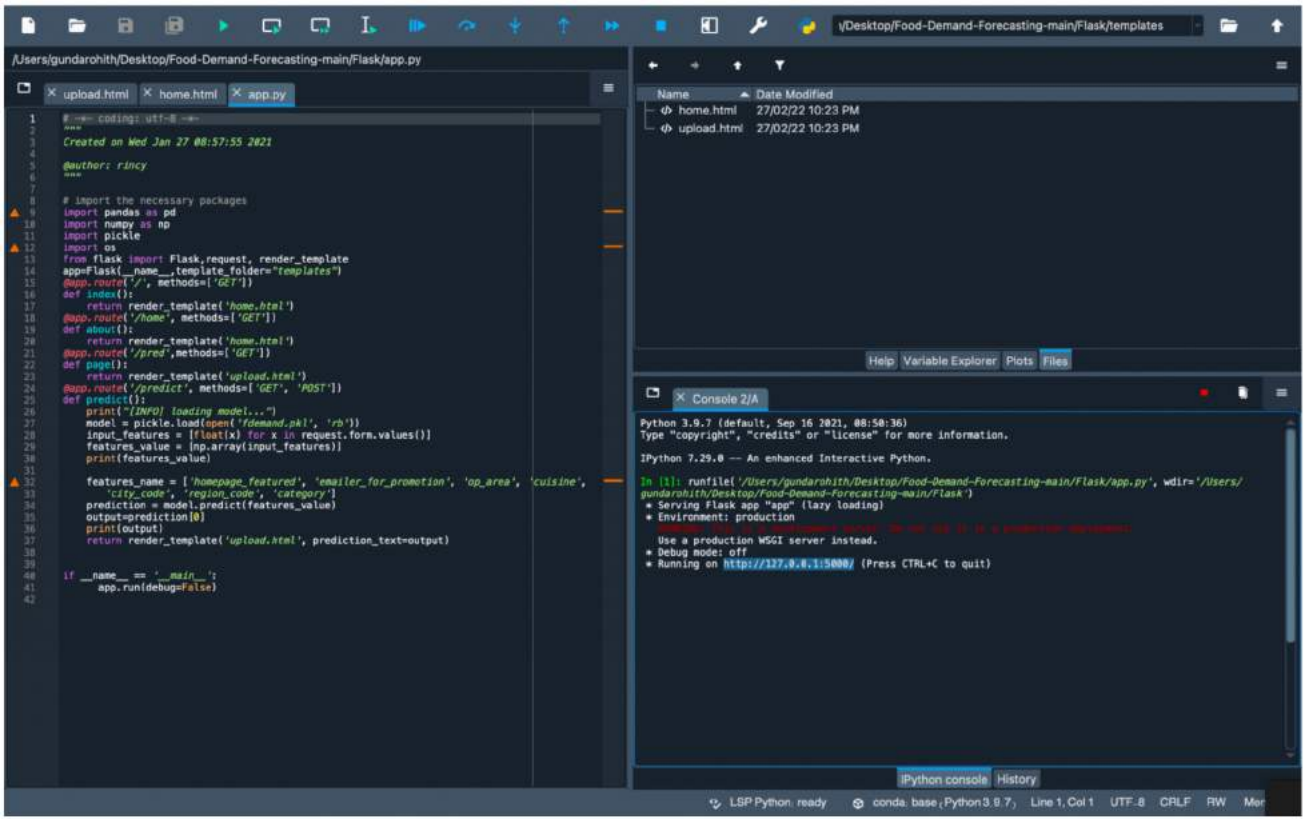


Figure 20: Executing the python file in the console.

3.OUTPUT

Figure 21: HOME PAGE

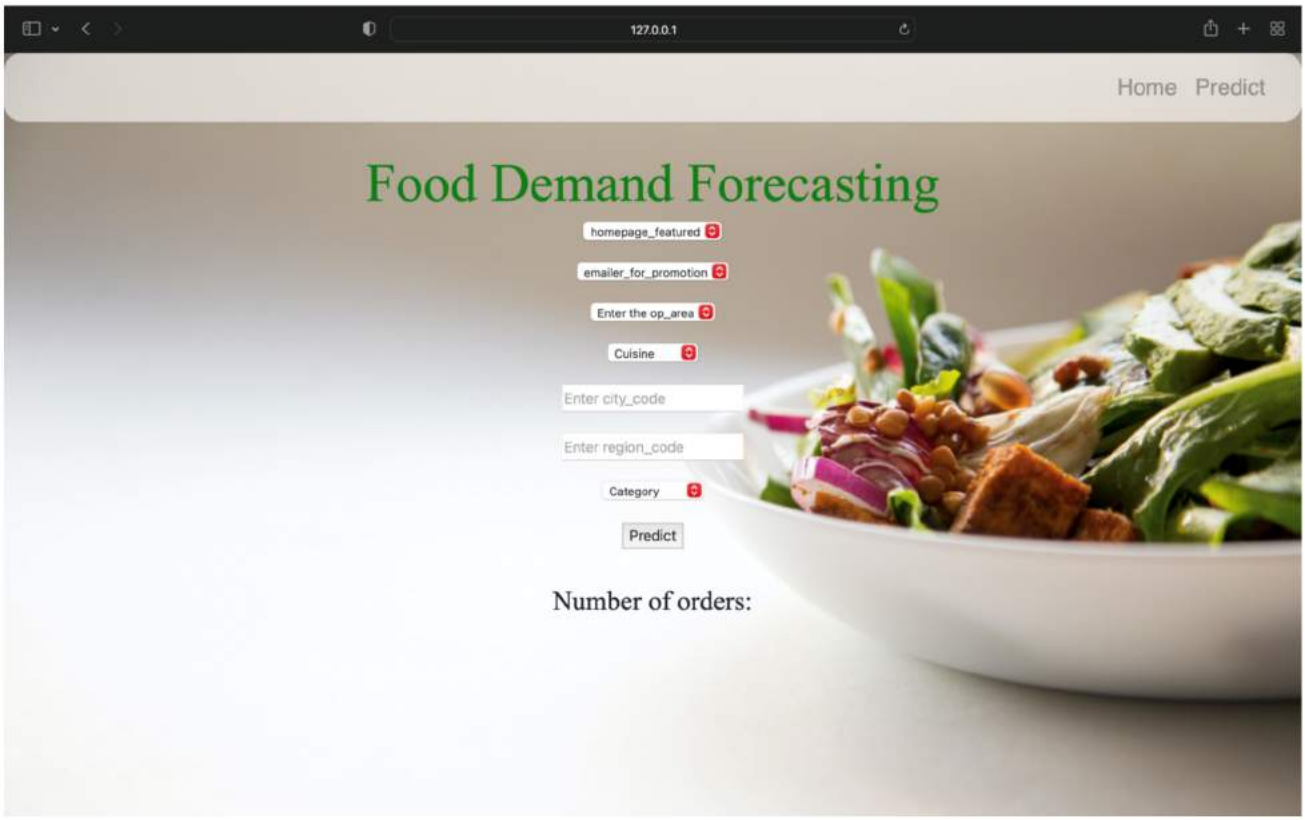


Figure 22: PREDICT PAGE

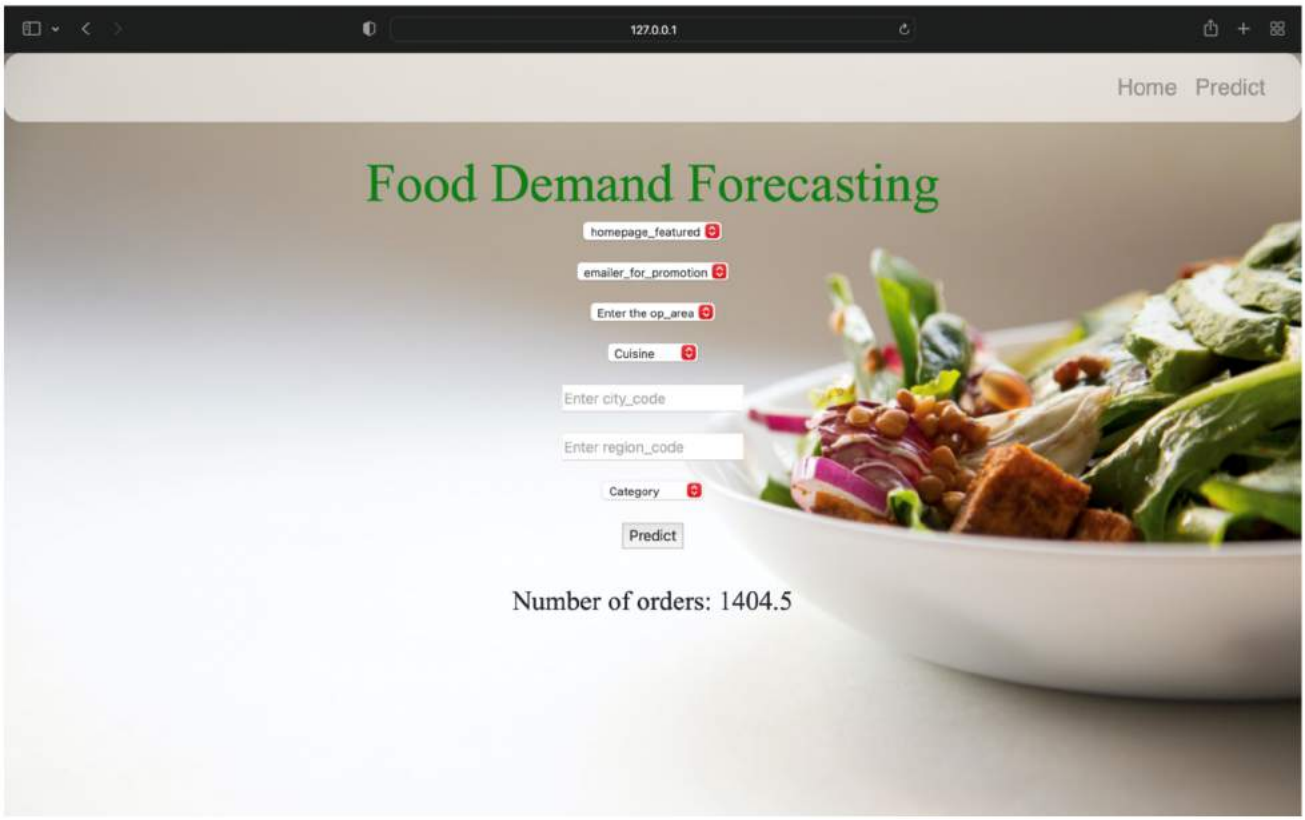


Figure 23: PREDICTION DONE

APPLICATIONS

1. This helps reduce risks and make efficient financial decisions that impact profit margins, cash flow, allocation of resources, opportunities for expansion, inventory accounting, operating costs, staffing and overall spend.
2. This also helps food industry business for forecasting the correct food from becoming perishable.

ADVANTAGES

1. Makes better financial decisions.
2. Increase profit margins.
3. Decrease wastage of food.
4. Maintain stock levels.
5. Improve business strategy.

DISADVANTAGES

- It requires maintaining dataset and regular update and testing.

CONCLUSION

This project not only concentrates for the benefit of retailers, but also customers. The retailers will know how much quantity of goods they must need to store in the warehouse and reduce the wastage of goods which in turn reduces their loss. At the same time, there will not be any shortage of goods to sell to the customers. The customers will also get good quality food products as the retailers buy the products as per the needs of the customers.

This will also improve the customer-retailer relationship and will be a profit for both of them. In terms of money spent by on goods and quantity of food products. This model will enhance the sales thus this project will help and gives benefit both the retailers and customers. The prediction is highly reliable and it prevents wastage of valuable perishable food products. Similarly the UI is also user friendly.

FUTURE ENHANCEMENT

- The steps that can be taken in due future to enhance the project can be collecting the real-time data of every day demand and retrain the model everyday with that data to increase the accuracy of predictions.
- Also we can start working on actual sensors and connect them to node red via a gateway for monitoring the warehouse environment.
- This also raises questions about the accuracy of the forecasts and how it can be improved. This can be done by refining the data more and some more data about the seasonality of the products.

BIBILOGRAPHY

- Alexandrov T., Bianconcini S., Dagum E. B., Maass P. and Mcelroy T. S. 2012 'A review of some modern approaches to the problem of trend extraction', Econometric Reviews, Vol. 31, pp. 593-624.
- Ballou R. H. 2007. 'The evolution and future of logistics and supply chain management', European business review, Vol. 19, pp. 332-48.
- Christo E., Ferreira M. and Alonso K. 2013. 'Use of Statistical Control for Improved Demand Forecasting', Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on. IEEE.
- De Oliveira Silva R., Da Silva Christo E. and Alonso Costa K. 2014. 'Analysis of Residual Autocorrelation in Forecasting Energy Consumption through a Java Program', Advanced Materials Research. Trans Tech Publ.
- Eksoz C., Mansouri S. A. and Bourlakis M. 2014 'Collaborative forecasting in the food supply chain: A conceptual framework', International journal of production economics, Vol. 158, pp. 120-35.

HELP FILE

PROJECT EXECUTION:

STEPS:

1. Go to **Start**, search and launch **ANACONDA NAVIGATOR**.
2. After launching of **ANACONDA NAVIGATOR**, launch **JUPYTER NOTEBOOK**.
3. Open "Major Project code" **IPYNB** file.
4. Then run all the cells.
5. All the **DATA PREPROCESSING**, **TRAINING AND TESTING**, **MODEL BUILDING**, **ACCURACY** of the model with different algorithms can be showcased
6. And at last a **PICKLE** file will be generated
7. Create a folder named **FLASK** on the **DESKTOP**. Extract the pickle file into this flask Folder.
8. Extract all the html files(home.html, upload.html) and python file (app.py) into the **FLASK** folder.
9. Then go back to **ANACONDA NAVIGATOR** and the launch the **SPYDER**.
10. After launching **SPYDER**, give the path of **FLASK** folder which you have created on the desktop.
11. Open all the **app.py** and **HTML** files present in the flask folder
12. After running of the app.py, open **ANACONDA PROMPT** and follow the below steps:
    a. cd File path -> click enter
    b. Python app.py -> click enter(we could see running of files).
13. Then open browser, at the **URL** area type "**localhost: 5000"**.
14. Home page of the project will be displayed.
15. Click on "**PREDICT**". Directly it will be navigated to upload page.
16. The upload page will be displayed where the user needs to enter the input and the click on the **PREDICT** and the output will be generated.

THANK YOU