

FOOD DEMAND FORECASTING FOR FOOD DELIVERY COMPANY USING IBM CLOUD

Industry Oriented mini–Project Report submitted to **Jawaharlal Nehru Technological University, Hyderabad**

In partial fulfillment for the requirement for the award of B-Tech Degree in Computer Science and Engineering.

SUBMITTED BY

NARAGONI SNEHA 19UK5A0510

SAI KUMAR KANAVERA 19UK5A0526

Under the esteemed guidance of

DR. VARUN REVURI

Associate Professor

Department of Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VAAGDEVI ENGINEERING COLLEGE WARANGAL



CERTIFICATE

This is to Certify that the Project entitled “**FOOD DEMAND FORECASTING FOR DELIVERING COMAPANY USING IBM CLOUD**”is submitted by below mentioned names submitted on November, 2021.In partial fulfillment of the requirements for the award of the Degree in Bachelor of Technology in CSE during the

Academic year 2018-2022

NARAGONI SNEHA 19UK5A0510

SAIKUMAR KANA VENA 19UK5A0526

Project Guide

DR.VARUN REVURI

Associate Professor

Head of the Department

Dr. R.NAVEEN KUMAR

Professor

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 mini project 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 mini project.

We express heartfelt thanks to the guide, **DR.VARUN REVURI** Assistant Professor, Department of CSE for her constant support and giving necessary guidance for completion of this mini project.

We express our sincere thanks and gratitude to **THE SMART BRIDGE**, for providing internship.

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

ABSTRACT

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.

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.

TABLE OF CONTENT

CONTENT	PAGE.NO
1. INTRODUCTION	8
1.1 Motivation	8
1.2 Problem definition	8-9
1.3 Objective of project	9
2. LITERATURE SURVEY	10
2.1 Machine Learning	11-12
3. ANALYSIS	13
3.1 Project Architecture	13
3.2 Software requirements	13-14
3.3 Hardware requirements	14
3.4 Project Flow	14
4. DESIGN	15
4.1 UML Diagrams	15
4.1.1 Class Diagram	15
4.1.2 Use Case Diagram	16
4.1.3 Sequence Diagram	16
5. IMPLEMENTATION AND RESULT	17
5.1 Introduction	17
5.2 Explanation of key functions	17
5.3 Method of implementation	17-18

6. TESTING AND VALIDATION	19
6.1 Introduction	19
6.2 Validation	19
7. CODE SNIPPETS	20
7.1 Model Code	20-22
7.2 HTML Code of Index	23-24
7.3 App.py Code	24
8. CONCLUSION	25
9. FUTURE ENHANCEMENT	25
10. BIBILOGRAPHY	25
REFERENCES	25-26

LIST OF FIGURES

Figure 1: Project Architecture	13
Figure 2: Class Diagram	15
Figure 3: Sequence Diagram	16
Figure 4: Use Case Diagram	16
Figure 5: .ipynb code describing importing libraries and datasets	20
Figure 6: .ipynb code describing Merging Data Frames	21
Figure 7: .ipynb code describing Label Encoding	21
Figure 8 : .ipynb code describing Splitting dataset into train and test	22
Figure 9: Predicting the result and Saving the model	22
Figure 10: Anaconda Navigator	22
Figure 11: HTML Code of index.html	23
Figure 12: HTML Code of index1.html	24
Figure 13: HTML code of result.html	24
Figure 14: Code for prediction in python file	24

1.INTRODUCTION

“Demand is an economic principle referring to a consumer's desire to purchase goods and services and willingness to pay a price for a specific good or service”.

Demand Forecasting is a process by which an individual or entity predicts the how much the consumer or customer would be willing to buy the product or use the service. Without Proper Demand forecasting it becomes impossible for any business to function. Improper Demand forecasting. would result in heavy loss. Different industry or company has different methods to predict the demands. In case of food industry, it is at most important that the demand needs to be on bulls’ eye since the food materials gets perished easily and has the fixed time frame to be used. So, the daily and weekly demand needs to be precise to avoid wastage which would otherwise increase the operating cost.

1.1 MOTIVATION:

The dataset, “Food Demand Forecasting” was released by an American professional services firm. The data set is related to a meal delivery company which operates in multiple cities. They have various fulfilment centers in these cities for dispatching meal orders to their customers. The dataset consists of historical data of demand for a product-center combination for weeks 1 to 145.

With the given data and information, the task is to predict the demand for the next 10 weeks (Weeks: 146-155) for the center-meal combinations, so that these fulfilment centers stock the necessary raw materials accordingly. The replenishment of raw materials is done only on weekly basis and since the raw material is perishable, the procurement planning is of utmost importance.

Therefore predicting the Demand helps in reducing the wastage of raw materials which would result in the reduced cost of operation. Increased customer satisfaction by timely fulfilling their expectations and requirements.

1.2 PROBLEM DEFINITION:

The data set is related to a meal delivery company which operates in multiple cities. They have various fulfilment centers in these cities for dispatching meal orders to their

customers. The dataset consists of historical data of demand for a product-center combination for weeks 1 to 145. With the given data and information, the task is to predict the demand for the next 10 weeks (Weeks: 146-155) for the center-meal combinations, so that these fulfillment centers stock the necessary raw materials accordingly.

1.3 OBJECTIVE OF PROJECT:

The main aim of this project is to create an appropriate machine learning model to forecast then the numbers of orders for the next ten weeks.

By the end of this project:

- You'll be able to understand the problem to classify if it is a regression or a classification kind of problem.
- You will be able to know how to pre-process / clean the data using different data pre-processing techniques.
- You will be able to analyze or get insights of data through visualization.
- Applying different algorithms according to dataset and based on visualization.
- You will be able to know how to find accuracy of the model.
- You will be able to know how to build a web application using Flask framework.
- Applying different algorithms according to the dataset and based on visualization.
- Real Time Analysis of Project
- Building an ease of User Interface (UI)
- Navigation of ideas towards other projects (creativity)
- Knowledge on building ML Model.
- You will be able to know how to find the accuracy of the model.
- How to Build web applications using the Flask framework.
- To find correlations between each variable and the target variable.
- To build a random forest regression model.
- To calculate efficiency of the model and errors in predictions
- To build a supervised machine learning model for forecasting value of a vehicle based on multiple attributes.
- The system that is being built must be feature based i.e. feature wise prediction must be possible.
- Providing graphical comparisons to provide a better view.

2. LITERATURE SURVEY

2.1 MACHINE LEARNING

Machine Learning (ML) is that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do.

In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method. The main focus of ML is to allow computer systems learn from experience without being explicitly programmed or human intervention.

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Before discussing the machine learning model, we must need to understand the following formal definition of ML given by professor Mitchell –

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

The above definition is basically focusing on three parameters, also the main components of any learning algorithm, namely Task(T), Performance(P) and experience (E). In this context, we can simplify this definition as –

ML is a field of AI consisting of learning algorithms that –

- Improve their performance (P)
- At executing some task (T)
- Over time with experience (E)

Applications of Machines Learning:

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML –

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

LINEAR REGRESSION ALGORITHM:

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog). Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables.

It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y (output). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

3. ANALYSIS

3.1 PROJECT ARCHITECTURE:

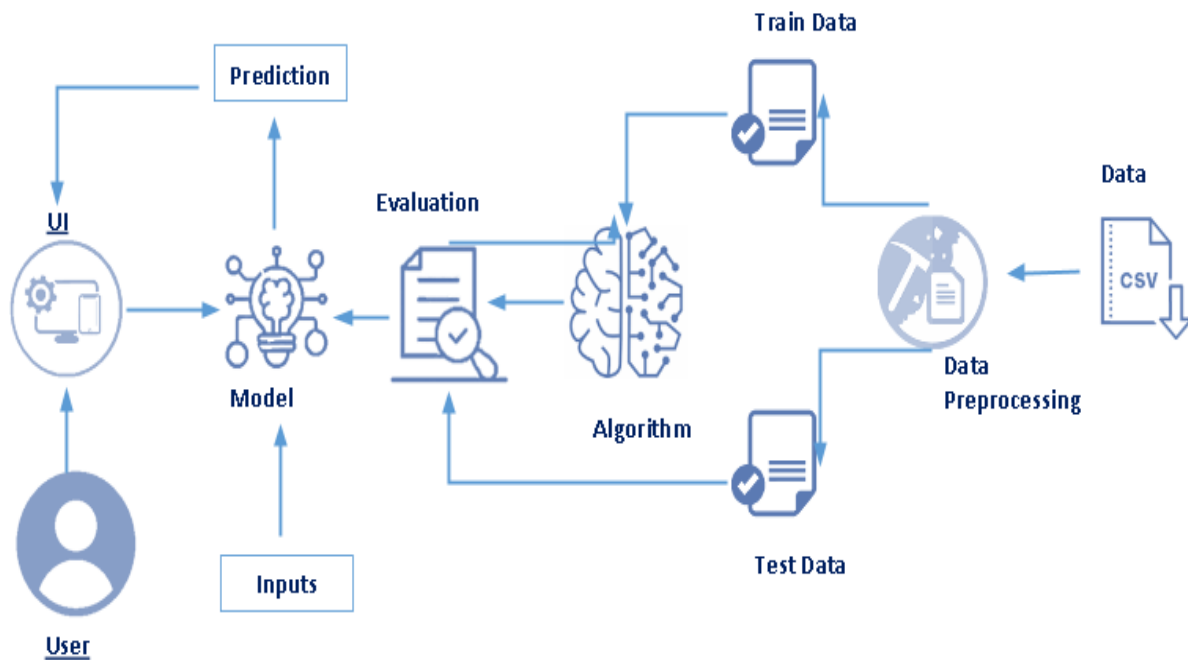


Figure 1: Project Architecture

3.2 SOFTWARE REQUIREMENTS:

➤ Python 3.8:

- Python is an interpreted high-level general-purpose programming language.
- Python can be used on a server to create web application.

➤ IBM Watson Studio

- IBM Watson® Studio empowers data scientists, developers and analysts to build, run and manage AI models, and optimize decisions anywhere on IBM Cloud Pak® for Data.

➤ Anaconda Environment

- The default environment base (path) is used because it consists of multiple libraries and modules.

➤ Flask:

- Flask is the module used for web framework.
- Flask provides you with tools, libraries and technologies that allow you to build a web application.
- And other python libraries like NumPy, pandas, etc.

3.3 HARDWARE REQUIREMENTS:

- Processor: Minimum 1 GHz; Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above.

1.3 PROJECT FLOW:

- The user interacts with the UI (User Interface) to upload the input features.
- Uploaded features/input is analysed by the model which is integrated.
- Once the model analyses the uploaded inputs, the prediction is showcased on the UI.

4.DESIGN

4.1 UML DIAGRAMS:

4.1.1 Class Diagram:

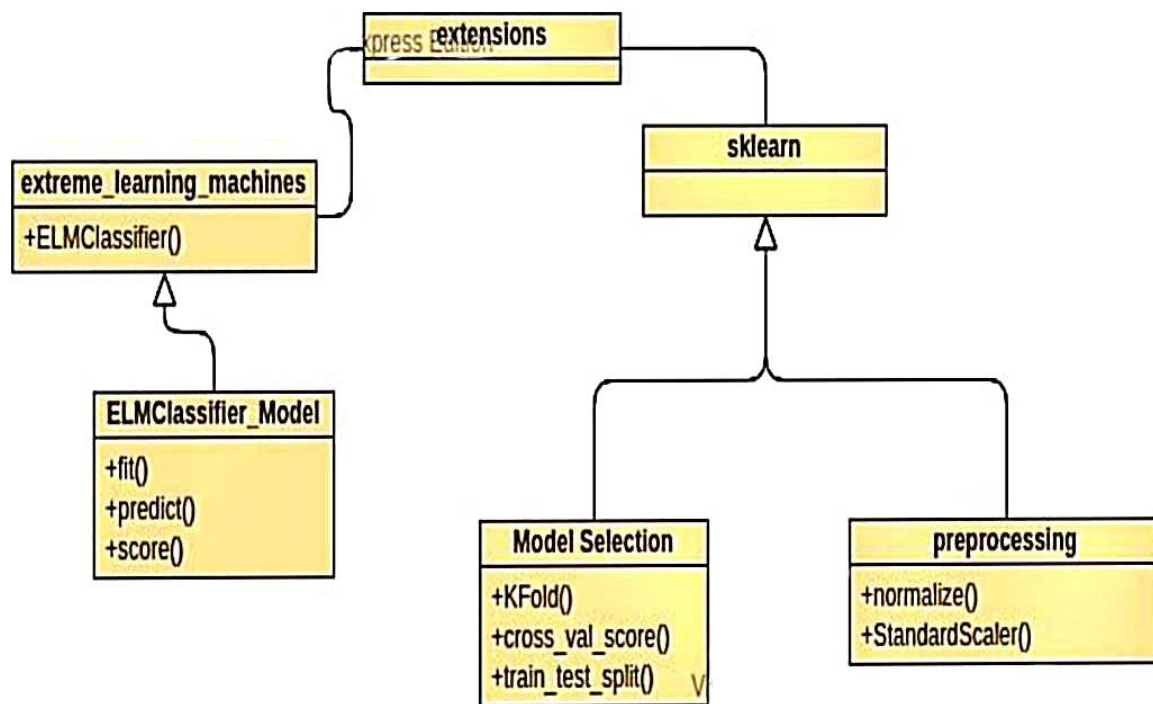


Figure 2: Class Diagram

4.1.2 Sequence Diagram

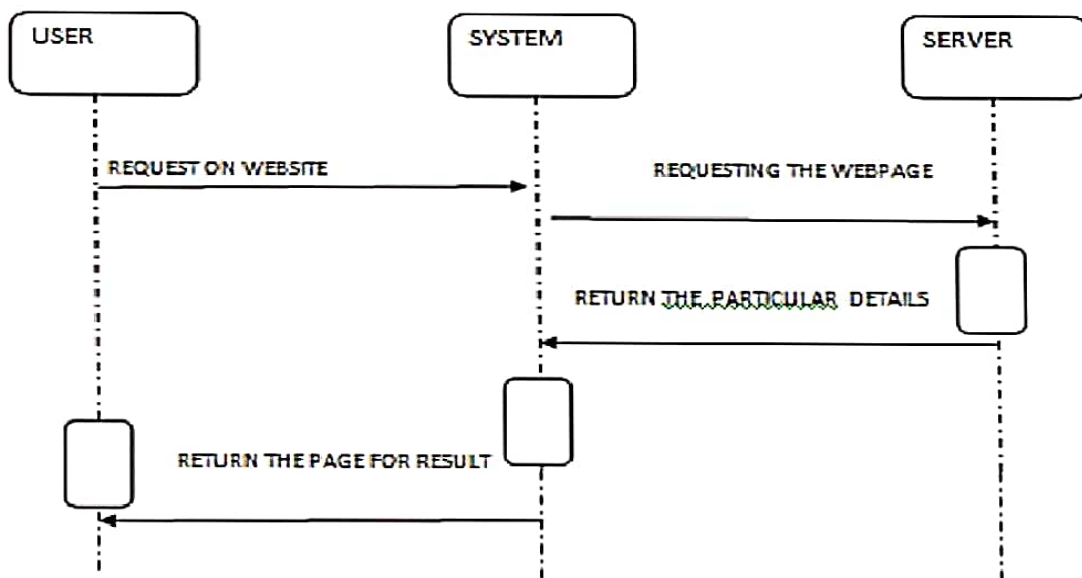


Figure 3: Sequence Diagram

4.1.3 Use Case Diagram:

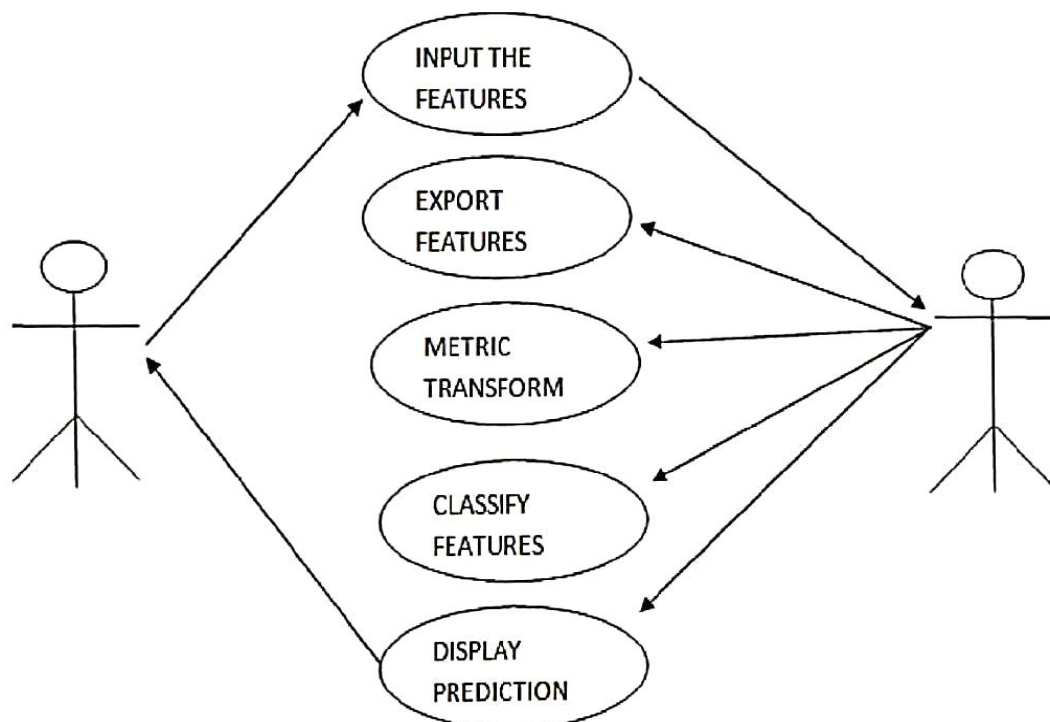


Figure 4: Use Case Diagram

5. IMPLEMENTATION AND RESULT:

5.1 INTRODUCTION:

Implementation is the process of defining how the system should be built, ensuring that it is operational and meets quality standards. It is a systematic and structured approach for effectively integrating a software-based service or component into the requirements of end users.

Implementation steps:

- 1) Data Collection
- 2) Data Pre-processing
- 3) Model Buliding
- 4) Application Buliding
- 5) Evaluation of the built model on ibm

5.2 EXPLANATION OF KEY FUNCTIONS:

Anaconda is a free and open sources distribution of the Python and R programming languages for data science and learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify management and deployment.

Anaconda3 includes Python 3.8. Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage anaconda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

5.3 METHOD OF IMPLEMENTATION:

License: Free use and redistribution under the terms of the Anaconda End User License Agreement.

Operating system: Windows Vista or newer, 64-bit macOS 10.10+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others. Windows XP supported on Anaconda versions 2.2 and earlier. See lists. Download it from our archive System architecture: 64-bit x86, 32-bit x86 with Windows or Linux, Power8 or Power9. Minimum 3 GB disk space to download and install.

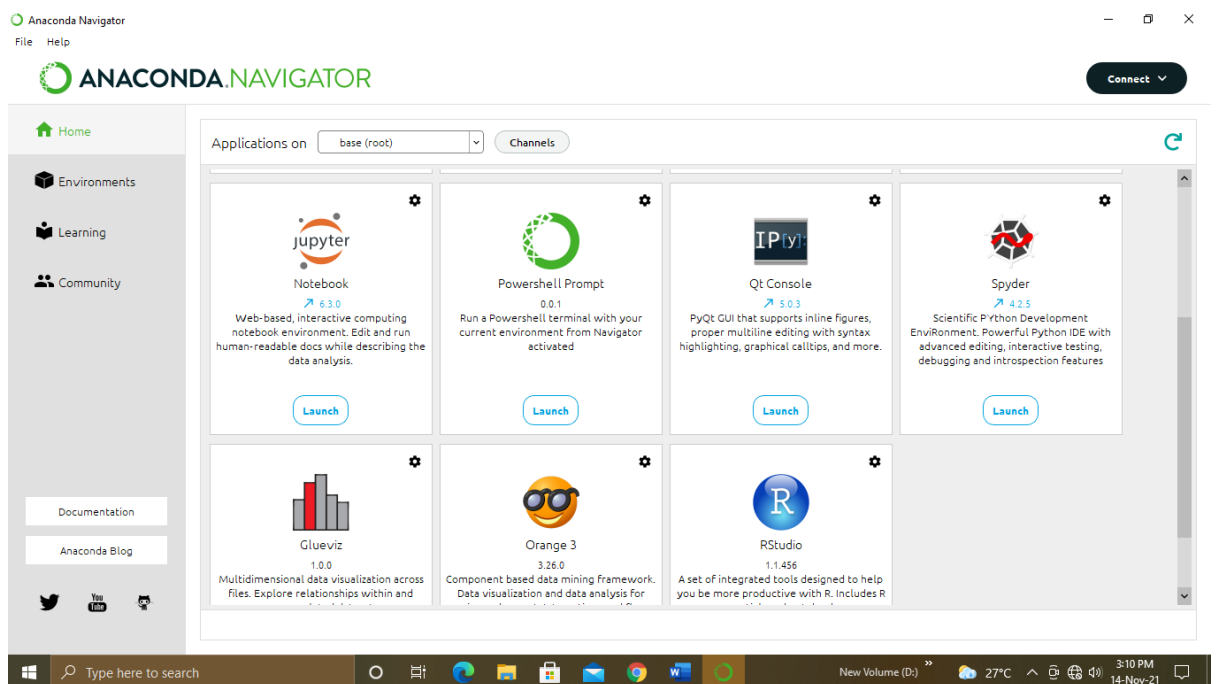


Figure 6: Anaconda Navigator

After the installation of anaconda navigator, we were taught python programming. We were taught how to use Pandas, how to include data frames, finding and replacing missing data with useful information, group-by functions, merging, joining We were taught various inclusion of python libraries such as Numpy i.e introduction to NumPy, NumPy arrays, few notes on array indexing, NumPy array indexing, NumPy operations and few exercises to recall it. and concatenating and other data input and output operations. We were also taught python for data visualization that is matplotlib, seaborn. Matplotlib is a plotting library for python and its extension NumPy. It makes use of general-purpose GUI kits and provides an object-oriented API for embedding the plots. In seaborn we were taught distribution plots, categorical plots, matrix plots, grids, regression plots etc.

6. TESTING AND VALIDATION:

6.1 INTRODUCTION:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 VALIDATION:

Validation is a series of techniques to avoid overfitting and to test the robustness of a method. The concept of cross-validation is about testing and training on different subset of the data available. The most naive way of training a neural network would be just to split the data in two sets and train on the whole training set until the loss stops to improve and then check the performance on the test set, this method however is very prone to overfitting, since there is no way of knowing when the optimization process goes from learning the relations in the data to learning exactly the points in the training set. One of the most simple and common variants of cross-validation is to split the data into three sets, training-set, validation-set, and test-set. The network is trained on the training set for as long as the loss of the validation set keep improving. Then to make the performance test unbiased the network is tested on the test-set which is essential to have since the network model will be based on both the training and validation sets as they have been used to determine the network model. A problem with the three-set split is that it might give good results purely by chance. If the data is not large enough or if the test set is 'easy' for the network to predict then one will be led to believe that the model is better than it really is. A common cross-validation technique to avoid this is k-fold cross-validation. The principle is to divide the data set into k subsets, and for k iterations, train the network using every set except k, and the validate the results with subset k. Then take the average of the results from all the k iterations as the real network performance. This method with a proper size of k will eliminate the problem with 'easy' sets and give a more-fair estimate of how well the network will actually perform on unseen data.

5. CODE SNIPPETS:

7.1 MODEL CODE:

```
In [3]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

train = pd.read_csv(r"C:\Users\sneha\OneDrive\mini project\Food-Demand-Forecasting-main\Dataset\train.csv")
test = pd.read_csv(r"C:\Users\sneha\OneDrive\mini project\Food-Demand-Forecasting-main\Dataset\test.csv")
```

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

```
Out[4]:
```

	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

Figure 7: .ipynb code describing importing libraries and datasets

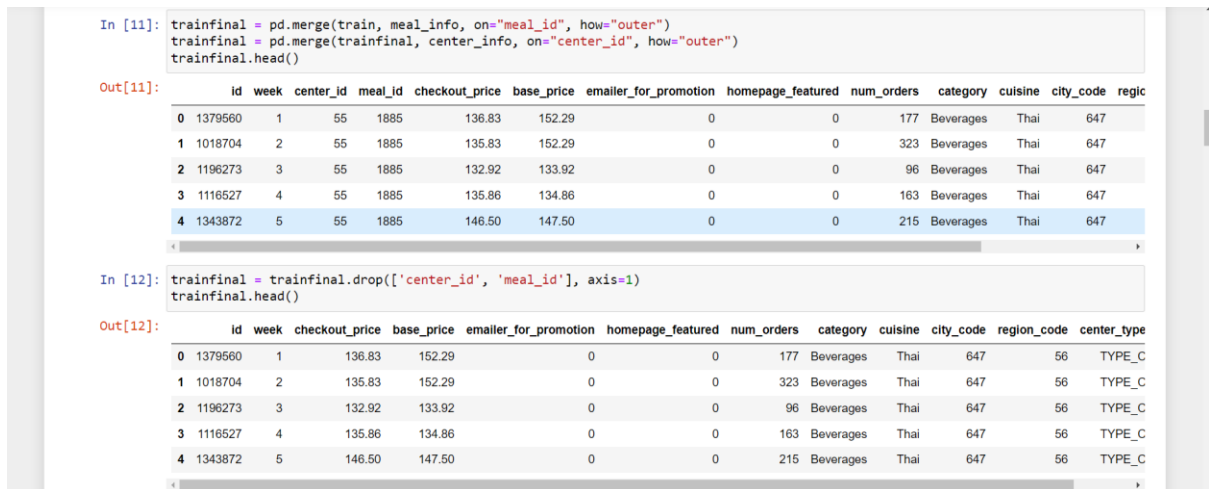


Figure 8: .ipynb code describing Merging Data Frames

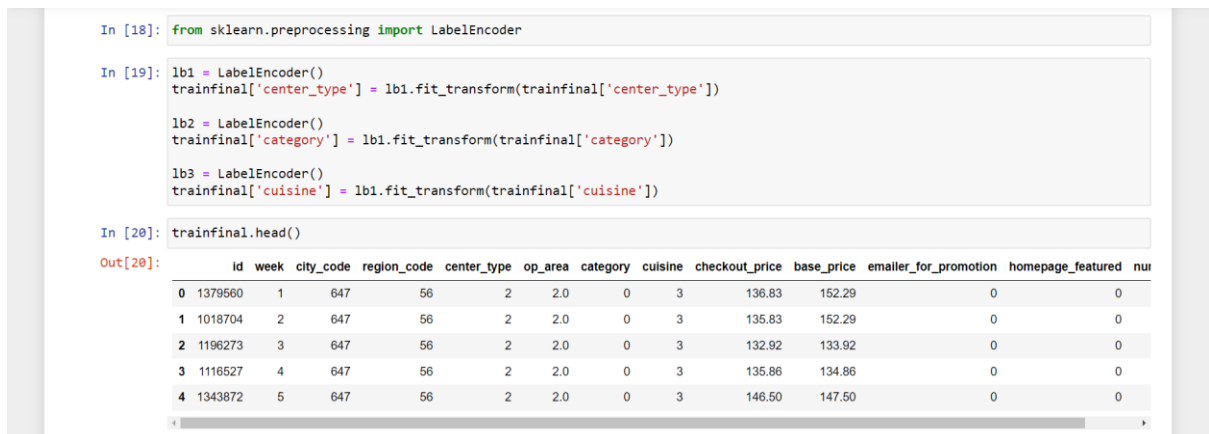


Figure 9: .ipynb code describing Label Encoding

```
In [25]: 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 [26]: trainfinal3.head()

Out[26]:
```

	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

```
In [27]: 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
```

Figure 10: .ipynb code describing Splitting dataset into train and test

```
In [42]: pred = DT.predict(X_test)
pred[pred<0] = 0
submit = pd.DataFrame({
    'id' : testfinal['id'],
    'num_orders' : pred
})

In [43]: submit.to_csv("submission.csv", index=False)

In [44]: submit.describe()

Out[44]:
```

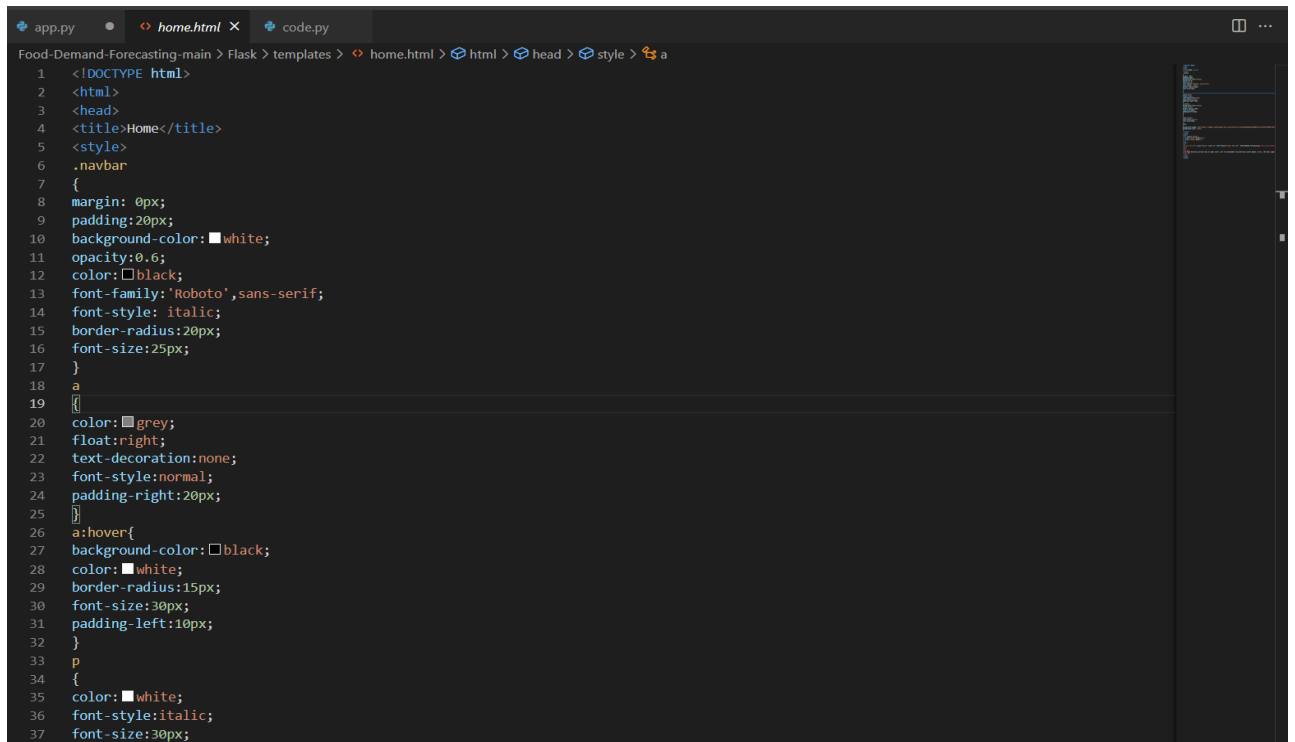
	id	num_orders
count	3.257300e+04	32573.000000
mean	1.248476e+06	262.834008
std	1.441580e+05	364.386375
min	1.000085e+06	14.500000
25%	1.123969e+06	64.456989
50%	1.247296e+06	148.623616
75%	1.372971e+06	318.149020
max	1.499990e+06	6123.894737

```
In [47]: import pickle
pickle.dump(DT, open('fdemand.pkl', 'wb'))
```

Figure 11: Predicting the result and Saving the model

7.2 HTML CODE OF INDEX

- INDEX.HTML



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Home</title>
5 <style>
6 .navbar
7 {
8 margin: 0px;
9 padding: 20px;
10 background-color: #white;
11 opacity: 0.6;
12 color: #black;
13 font-family: 'Roboto', sans-serif;
14 font-style: italic;
15 border-radius: 20px;
16 font-size: 25px;
17 }
18 a
19 [
20 color: #grey;
21 float: right;
22 text-decoration: none;
23 font-style: normal;
24 padding-right: 20px;
25 ]
26 a:hover{
27 background-color: #black;
28 color: #white;
29 border-radius: 15px;
30 font-size: 30px;
31 padding-left: 10px;
32 }
33 p
34 {
35 color: #white;
36 font-style: italic;
37 font-size: 30px;
```

Figure 12: HTML Code of index.html

- **INDEX2.HTML**

```
Food-Demand-Forecasting-main > Flask > templates > upload.html > html > head > style > a: hover
50
51 <div class="bar">
52 <a href="/pred">Predict</a>
53 <a href="/home">Home</a>
54 <br>
55 </div>
56
57 <div class="container">
58 <center> <div id="content" style="margin-top:2em">
59 <h2 style="color: ■ white;font-family:Times New Roman;font-size:60"><center>Food Demand Forecasting</center></h2>
60 <form action="{{ url_for('predict') }}" method="POST">
61
62 <select id="homepage_featured" name="homepage_featured">
63 <option value="">homepage_featured</option>
64 <option value="0">No</option>
65 <option value="1">Yes</option>
66 </select><br><br>
67 <select id="emailer_for_promotion" name="emailer_for_promotion">
68 <option value="">emailer_for_promotion</option>
69 <option value="0">No</option>
70 <option value="1">Yes</option>
71 </select><br><br>
72
73 </div>
74
75
76 <input class="form-input" type="text" name="op_area" placeholder="Enter the op_area(2-7)"><br><br>
77 <select id="cuisine" name="cuisine">
78 <option value="">Cuisine</option>
79 <option value="0">Continental</option>
80 <option value="1">Indian</option>
81 <option value="2">Italian</option>
82 <option value="3">Thai</option>
83 </select><br><br>
84 <input class="form-input" type="text" name="city_code" placeholder="Enter city_code"><br><br>
85 <input class="form-input" type="text" name="region_code" placeholder="Enter region_code"><br><br>
86 <select id="category" name="category">
87
```

Figure 13: HTML Code of index1.html

7.3 APP.PY CODE

```
Food-Demand-Forecasting-main > Flask > app.py > ...
1 import pandas as pd
2 import numpy as np
3 import pickle
4 import os
5 import pickle
6 from flask import Flask,request, render_template
7 app=Flask(__name__,template_folder="templates")
8 @app.route('/', methods=['GET'])
9 def index():
10     return render_template('home.html')
11 @app.route('/home', methods=['GET'])
12 def about():
13     return render_template('home.html')
14 @app.route('/pred',methods=['GET'])
15 def page():
16     return render_template('upload.html')
17 @app.route('/predict', methods=['GET', 'POST'])
18 def predict():
19     print("[INFO] loading model...")
20     model = pickle.loads(open('fdemand.pkl','rb').read())
21     input_features = [float(x) for x in request.form.values()]
22     features_value = [np.array(input_features)]
23     print(features_value)
24
25     features_name = ['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
26                     'city_code', 'region_code', 'category']
27     prediction = model.predict(features_value)
28     output=prediction[0]
29     print(output)
30     return render_template('upload.html', prediction_text=output)
31
32
33 if __name__ == '__main__':
34     app.run( debug=True, threaded=False)
```

Figure 15: Code for prediction in python file

8.CONCLUSION

Demand forecasts is, with no doubt, the basis for developing an efficient supply chain. The supply chain planning and control depends of accurate estimates of the volumes of products and services to be processed to satisfy customer's needs.

9. FUTURE ENHANCEMENT

As time progresses and more data are collected, the supervised learning will produce more accurate results and will be helpful in determining fare optimizer and adjustments and continuously improve future recommendations.

10.BIBILOGRAPHY

- <https://youtu.be/COoNk61K6HQ>
- https://smartinternz.com/Student/guided_project_workspace/5778

REFERENCES:

- [1] 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. [2] Ballou R. H. 2007. 'The evolution and future of logistics and supply chain management', *European business review*, Vol. 19, pp. 332-48.
- [3] 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.
- [4] 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.*
- [5] 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.
- [6] Giacon E. and Mesquita M. a. D. 2011. "A survey on detailed production scheduling in manufacturing plants in São Paulo, Brazil", *Gestão & Produção*, Vol. 18, pp. 487-98.

- [7] Hyndman R. J., Akram M. and Archibald B. C. 2008. "The admissible parameter space for exponential smoothing models", *Annals of the Institute of Statistical Mathematics*, Vol. 60, pp. 407-26.
- [8] Hyndman R. J., Koehler A. B., Ord J. K. and Snyder R. D. 2005. 'Prediction intervals for exponential smoothing using two new classes of state space models', *Journal of Forecasting*, Vol. 24, pp. 17-37. [9] Junior M. L. and Filho M. G. 2012. 'Production planning and control for remanufacturing: literature review and analysis', *Production Planning & Control*, Vol. 23, pp. 419-35.
- [10] Krajewski L. J., Ritzman L. P. and Malhotra M. K. 2012. *Operations management*, Pearson Education Limited.
- [11] Lewis C. 1997. "Demand forecasting and inventory control", *Control-Coventry-Institute Of Operations Management*, Vol. 23, pp. 20-23.
- [12] Schmit T. M. and Kaiser H. M. 2006. "Forecasting fluid milk and cheese demands for the next decade", *Journal of dairy science*, Vol. 89, pp. 4924-36.
- [13] Shang Y. 2009. 'Exponential random geometric graph process models for mobile wireless networks', *CyberEnabled Distributed Computing and Knowledge Discovery*, 2009. CyberC'09. International Conference on. IEEE.
- [14] Tratar, L. F. (2015) 'Forecasting method for noisy demand', *International Journal of Production Economics*, 161, 64-73.

HELP FILE

1. Turn on the computer
2. Connect internet
3. Open chrome
4. Search for
5. <http://127.0.0.1:5000>
6. You will be able to see home page of food demand forecasting
7. Click on predict button
8. Select the homepage featured
9. Select emailer for promotion
10. Enter the op_area
11. Enter the city code
12. Enter the region code
13. Enter the Category
14. Click on predict