

INTRODUCTION

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.

This is a Software Prototype on Demand Forecasting of perishable goods using machine learning solutions. The project aims to predict number of orders a meal delivery company would receive depending upon the past

LITERATURE SURVEY

Existing Problem: A food delivery service has to deal with a lot of perishable raw materials which makes it all. Too much inventory in the warehouse means more risk of wastage, and not enough could lead to out-of-stocks.

Proposed Solution: The machine learning Model solution has been prepared which should have the ability to predict accurately. And the accuracy can be achieved if more data is provided. Hence More data Gives More Accuracy in demand Forecasting

THEORETICAL ANALYSIS

Hardware and Software requirements of project:

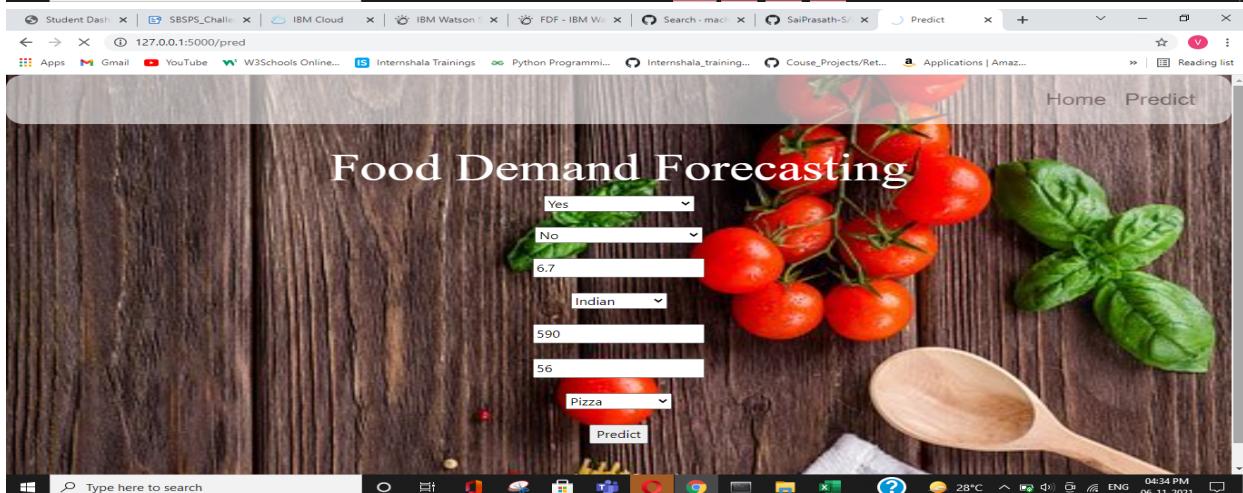
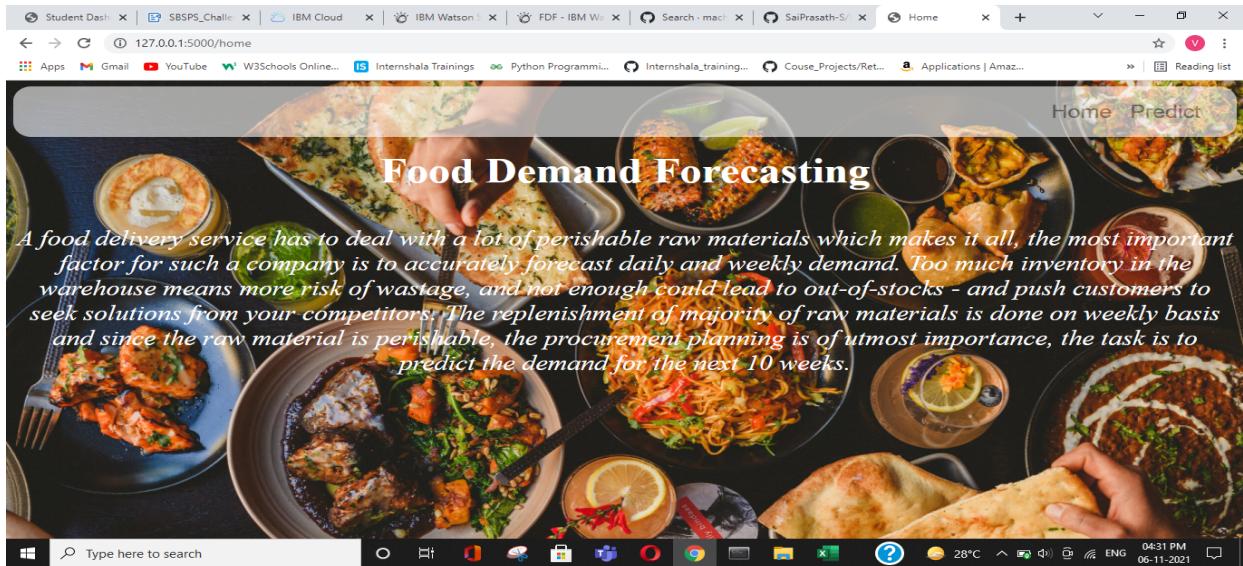
- IBM Cloud Storage
 - Watson studio
 - Anaconda Navigator

Libraries Used:

pandas, numpy, scikit learn, matplotlib, seaborn and flask

RESULT

End Point: <https://us-south.ml.cloud.ibm.com/ml/v4/deployments/11f7d5d6-bb8a-4e09-9a44-b27b8d63ef49/predictions?version=2021-11-05>



IBM Watson Studio All Search Buy Buy

Vidya Reddy's Account

File Edit View Insert Cell Kernel Help

In [2]: train.head()

```
Out[2]:
   id  week  center_id  meal_id  checkout_price  base_price  emailer_for_promotion  homepage_featured  num_orders
0  1379560      1      55    1885       136.83      152.90                  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 [3]: test.head()

```
Out[3]:
   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
```

In [4]: train.info()

Student Dashboard | Sbps_Challenge | IBM Cloud | IBM Watson Studio | FDF - IBM Watson | Search - machine | New Tab | + | - | X

← → ⌂ dataplatform.cloud.ibm.com/analytics/notebooks/v2/7d88b9b0-e96e-404d-8e9b-d214c554e238?projectId=17bf2f0b-d8a2-445d-b2f5-6524aaeff440&context=cpdaas

Apps Gmail YouTube W3Schools Online... Internshala Trainings Python Programming Internshala_training... Couse_Projects/Ret... Applications | Amaz... » | Reading list

IBM Watson Studio All Search Buy ⌂ 1 Vidya Reddy's Account VR

Projects / Custom Model Deployment / FDF

File Edit View Insert Cell Kernel Help Not Connected Not Trusted | Python 3.8 ⌂

`submit = pd.DataFrame({
 'id' : testfinal['id'],
 'num_orders' : pred
})`

In [31]: `submit.to_csv("submission.csv", index=False)`

In [32]: `submit.describe()`

Out[32]:

	id	num_orders
count	3.257300e+04	32573.000000
mean	1.248476e+06	262.725498
std	1.441580e+05	363.311343
min	1.000085e+06	15.573770
25%	1.123969e+06	64.666667
50%	1.247296e+06	147.652985
75%	1.372971e+06	324.250000
max	1.499996e+06	6087.555556

Type here to search 04:36 PM 28°C ENG 06-11-2021

Student Dashboard | Sbps_Challenge | IBM Cloud | IBM Watson Studio | FDF - IBM Watson | Search - machine | New Tab | + | - | X

← → ⌂ dataplatform.cloud.ibm.com/analytics/notebooks/v2/7d88b9b0-e96e-404d-8e9b-d214c554e238?projectId=17bf2f0b-d8a2-445d-b2f5-6524aaeff440&context=cpdaas

Apps Gmail YouTube W3Schools Online... Internshala Trainings Python Programming Internshala_training... Couse_Projects/Ret... Applications | Amaz... » | Reading list

IBM Watson Studio All Search Buy ⌂ 1 Vidya Reddy's Account VR

Projects / Custom Model Deployment / FDF

File Edit View Insert Cell Kernel Help Not Connected Not Trusted | Python 3.8 ⌂

`from ibm_watson_machine_learning import APIClient`

In [34]: `wml_credentials = {
 "url": "https://us-south.ml.cloud.ibm.com",
 "apikey": "spShCeZWxsuisv9G_yWMMv_nSBm4rTloU6RmlA9nq0v"
}`

`client = APIClient(wml_credentials)`

In [35]: `def guid_from_space_name(client, space_name):
 space = client.spaces.get_details()
 #print(space)
 return(next(item for item in space['resources'] if item['entity'][name] == space_name)[metadata]['id'])`

In [36]: `space_uid = guid_from_space_name(client, 'models')
print("Space UID = " + space_uid)`

Space UID = 8870350e-42c1-4b32-ab97-1bab7d97d89c

In [37]: `client.set.default_space(space_uid)`

Out[37]: 'SUCCESS'

In [38]: `client.software_specifications.list()`

Type here to search 04:37 PM 28°C ENG 06-11-2021

The screenshot shows the IBM Watson Studio interface with a Jupyter Notebook open. The code in the notebook is as follows:

```

In [41]: software_spec_uid = client.software_specifications.get_uid_by_name("default_py3.8")
software_spec_uid
Out[41]: 'ab9e1b80-f2ce-592c-a7d2-4f2344f77194'

In [48]: model_details = client.repository.store_model(model_DT,meta_props={
    client.repository.ModelMetaNames.NAME:"Food demand forecasting",
    client.repository.ModelMetaNames.TYPE:"scikit-learn_0.23",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid
})
model_id = client.repository.get_model_uid(model_details)

In [49]: model_id
Out[49]: '6712d188-ec32-4e39-b43e-c44053e90e1b'

In [52]: X_train[0]
Out[52]: array([ 0.,  0.,  3.6,  2., 599., 56.,  6.])

In [53]: DT.predict([[ 0.,  0.,  3.6,  2., 599., 56.,  6.]])
Out[53]: array([52.546875])

```

The screenshot shows the deployment details for the "Food demand forecasting" model. The deployment is labeled "newdeployment" and is currently "Deployed" and "Online". The deployment ID is 11f7d5d6-bb8a-4e09-9a44-b27b8d63ef49. The software specification used is "default_py3.8". The deployment has 1 copy and no serving name. There is no description provided.

APPLICATIONS

- Used in food industries to have a rough idea about the working
- Used in warehouse management to save raw materials.
- Used to employ and manage the working of employees
- Altogether gives the food delivery company a robustness and financial stability

CONCLUSION

Our project accurately predicts the number of orders from each area. So as a result, the food delivery company can estimate the total production required and number of employees to deployed to each area. Depending on the cuisine type the raw materials are ordered. Wastage can minimized further.

FUTURE SCOPE

Machine Learning (ML) has found it's impact in every part of our life. It is bound to bring about a new revolution in mankind. In this project ML model can predict and provide a better solution to the problem. It can also predict many other features like weather or any major change in lifestyle of people of a particular area and thus can map their effect on the number of orders. Hypothetically speaking their are countless benefits this project can draw from growing ML.

BIBLIOGRAPHY

- YouTube Vedios
- Google