IBM HACK CHALLENGE 2020

Optimized Warehouse Management of Perishable Goods for a Food Delivery Company

Goods Demand Forecasting Calculator

Team Name – **Code 2k20**

Team Members - **Saloni Mittal**

Vaibhav Chaudhary

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 the majority of raw materials is done on a weekly basis and since the raw material is perishable, the procurement planning is of utmost importance. This project we are going to develop a machine learning model using IBM services (Watson studio and node-red) and dataset. We approached the task of predicting food demand as a supervised machine learning task. We trained a machine learning model to predict the demand for goods for the next 10 weeks or in the future with the help of Auto AI. This project is a dashboard where we can predict the demand for goods for the next 10 weeks or in the future.

1 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 solutions from your competitors. The replenishment of the majority of raw materials is done on a weekly basis and since the raw material is perishable, the procurement planning is of utmost importance.

Purpose - In this project, we are going to predict food demand for any company. For this, we have developed a user interface, which can be used by Food Delivery Company to optimize Warehouse Management of Perishable Goods. Develop a website to predict the demand for goods for the next 10 weeks or in the future. The aim of this project is to provide a simple interface by which we can easily and accurately predict the food demand.

2. LITERATURE SURVEY

2.1 Existing problem -

2.1.1 In the retail industry, demand forecasting is one of the main problems of supply chains to optimize stocks, reduce costs, and increase sales, profit.

2.1.2 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.

2.1.3 Too much inventory in the warehouse means more risk of wastage, and not enough could lead to out-of-stocks.

2.1.4 The replenishment of the majority of raw materials is done on a weekly basis and since the raw material is perishable, the procurement planning is of utmost importance.

2.2 Proposed solution - we proposed a machine learning model build using Auto AI and a user interface which is easy to use and more accurate build using Node-Red. In this project, we developed a model using IBM services (Watson studio and node-red) and available food dataset. Methods we approached the task of predicting food demand as a supervised machine learning task. We trained a machine learning model to predict food demand using Auto AI and provide a simple user interface(dashboard) by which anyone can easily predict the food demand for the future.

3. THEORETICAL ANALYSIS

Food demand prediction is a regression problem.

Stages of predictive modelling

- Problem Definition
- Hypothesis Generation
- Data Extraction / Collection
- Data Exploration and Transformation
- Predictive Modeling
- Model Deployment / Implementation

We have used Auto AI for model building. We have collected the dataset and visualized the data.

Fig - Hist. plot

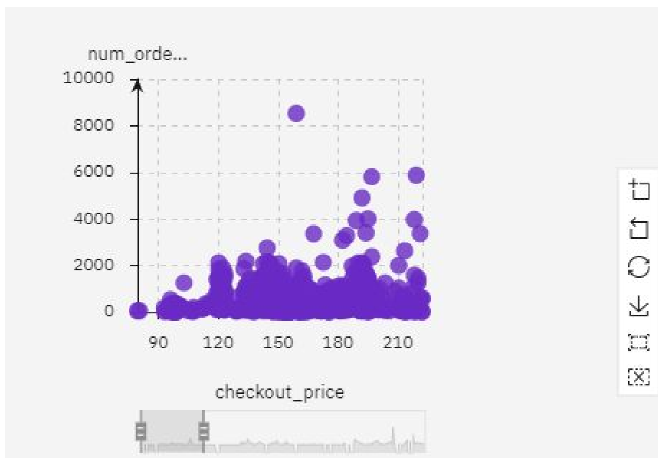

fig. - scatter plot

fig. - scatter plot

After this, we applied Auto AI in a dataset where the dataset is refined and it creates 8 pipelines and the pipeline 3 is on 1st rank.

Pipeline 3 -

Holdout RMSE *(Optimized)*

151.792

Algorithm

Random Forest Regressor

Enhancements

1st hyperparameter optimization

HPO-1

Feature engineering

FE

Build time

01:52:01s

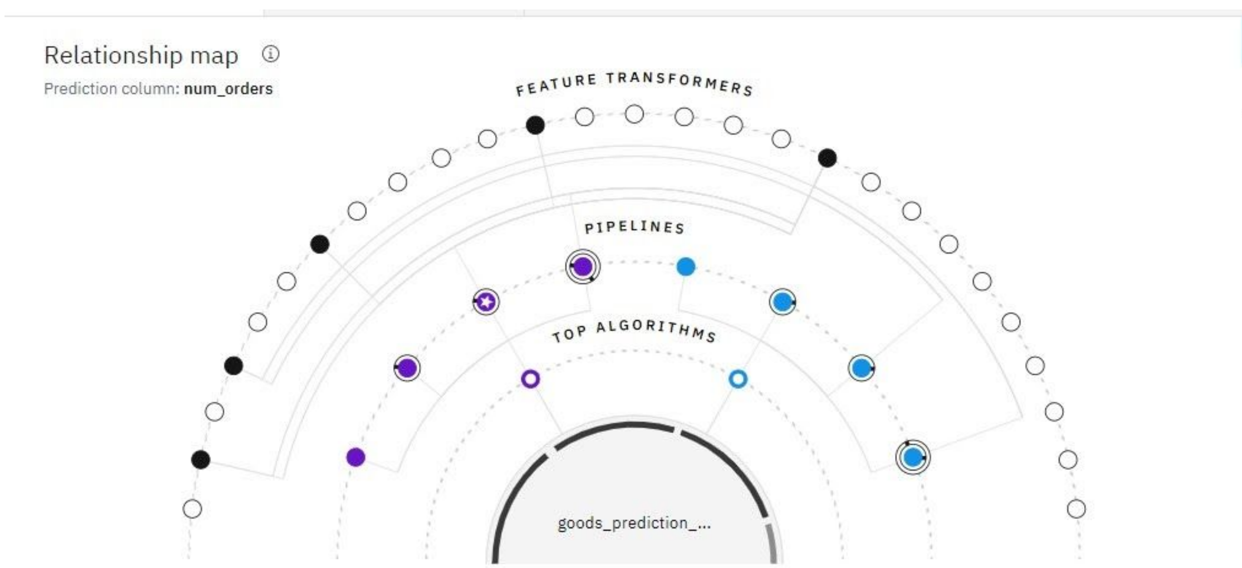

Fig-Pipelines and algorithms

fig-Auto AI Model generation process

Rank	Name	Algorithm	RMSLE	Enhancements	Build time
★ 1	Pipeline 3	Random Forest Regressor	**0.510**	HPO-1 FE	01:52:01
2	Pipeline 4	Random Forest Regressor	**0.510**	HPO-1 FE HPO-2	01:04:22
3	Pipeline 1	Random Forest Regressor	**0.510**	*None*	00:01:28
4	Pipeline 2	Random Forest Regressor	**0.510**	HPO-1	00:18:07
5	Pipeline 7	Extra Trees Regressor	**0.515**	HPO-1 FE	00:41:21
6	Pipeline 8	Extra Trees Regressor	**0.515**	HPO-1 FE HPO-2	00:25:34
7	Pipeline 5	Extra Trees Regressor	**0.517**	*None*	00:00:43
8	Pipeline 6	Extra Trees Regressor	**0.517**	HPO-1	00:07:30

Fig-All models and pipelines

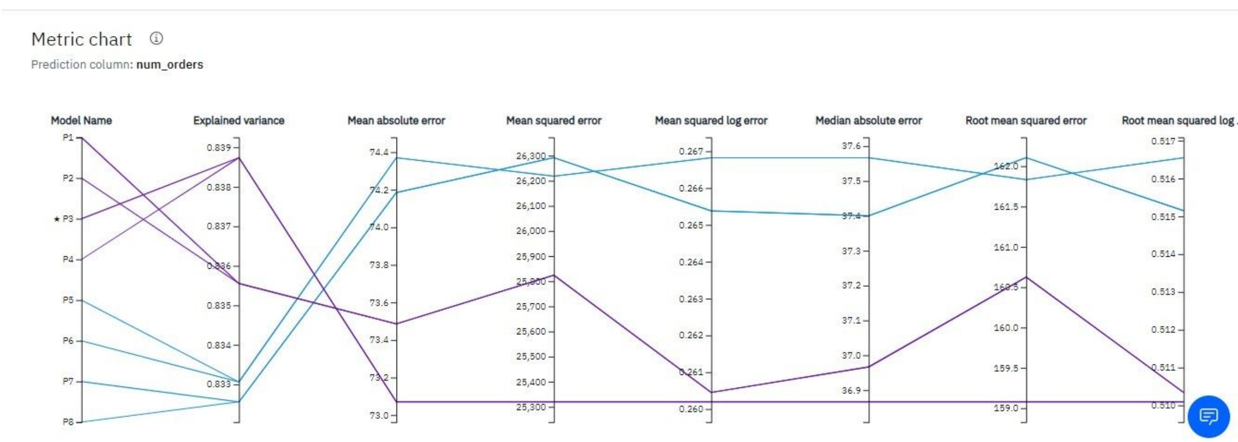

Fig-Merit Chart(Cross Validation)

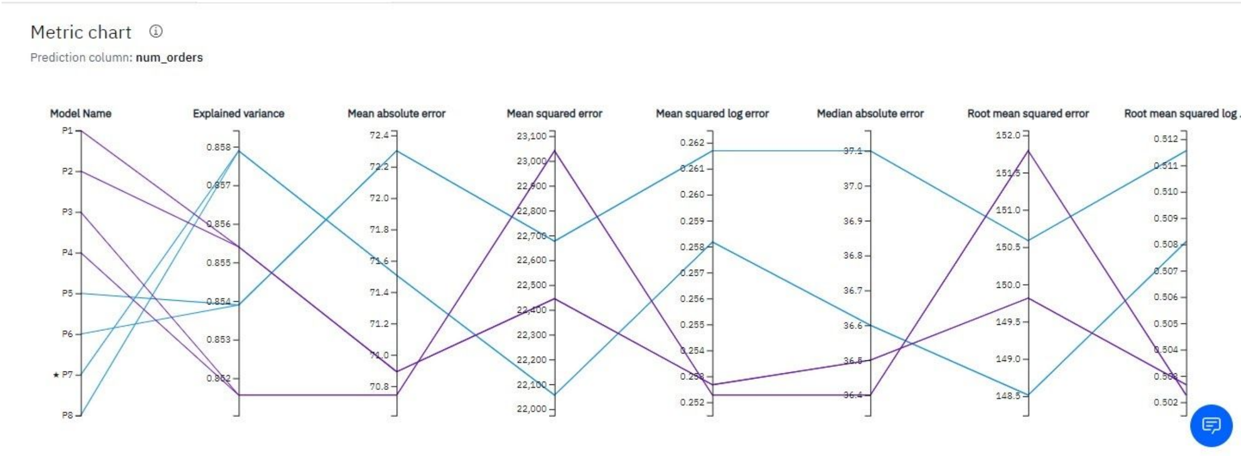

Fig- Merit Chart(Holdout)

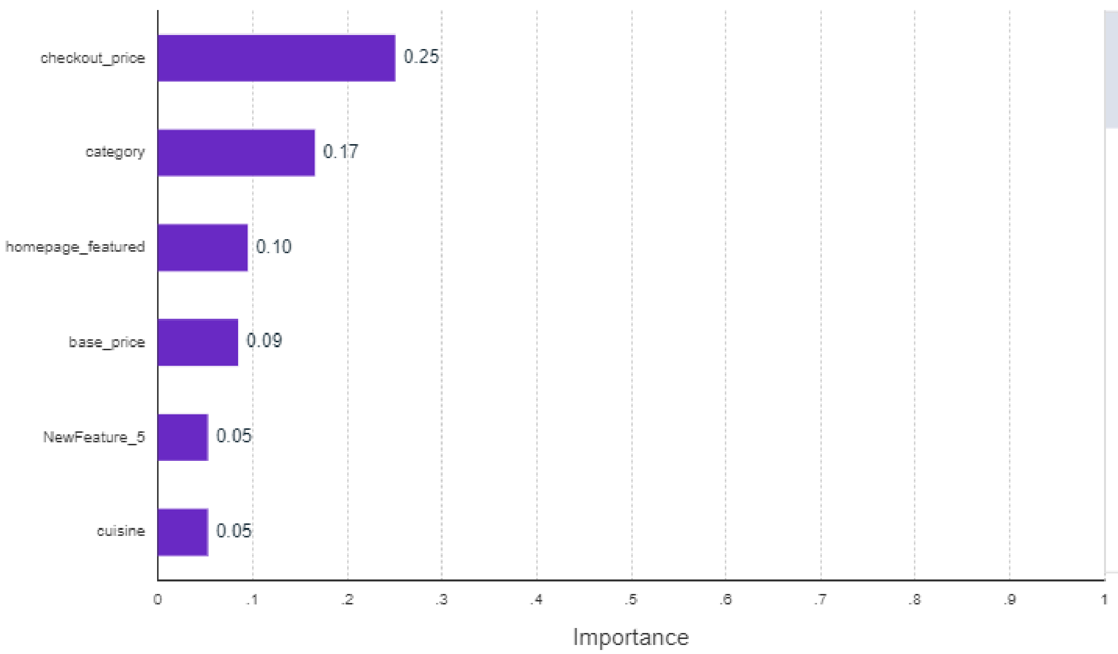

Fig- Columns

3.1 Block diagram - This is the block diagram of the workflow for building and deploying the model.

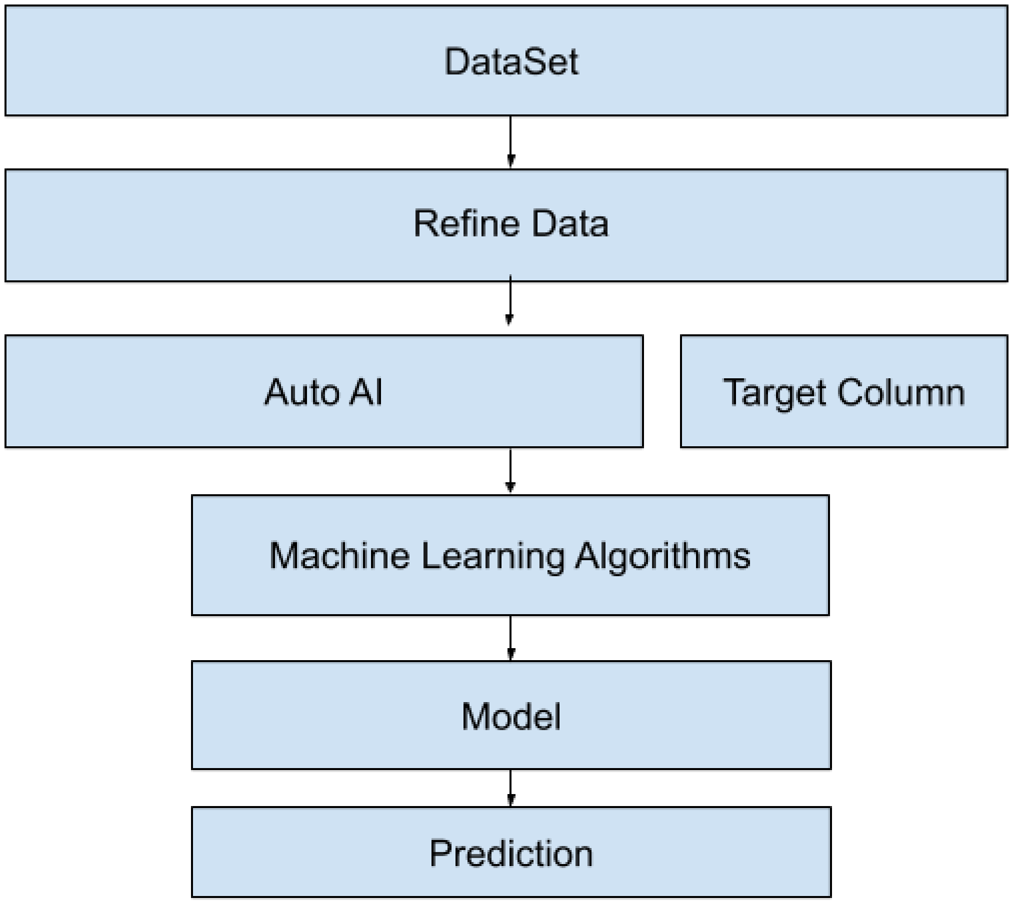

Fig. - Block Diagram

3.2 Hardware / Software designing - For designing models, we have used Auto AI After building the model, save and deploy the model. After deploying the model, connecting this model to Node-red for the user interface and deploying the flow, we built a dashboard for food demand prediction.

Node-Red Flow -

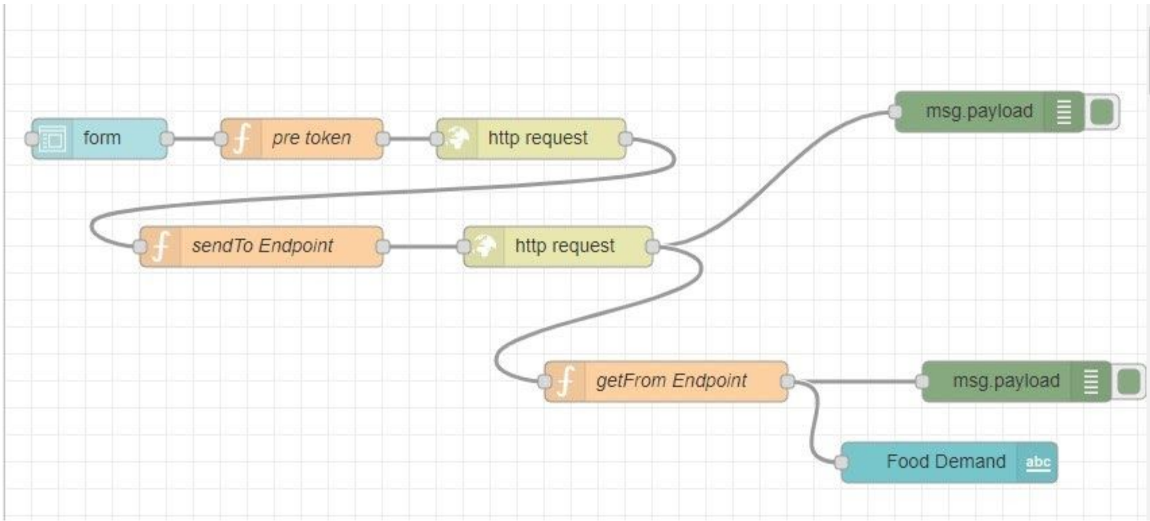

Fig - Node-Red Flow

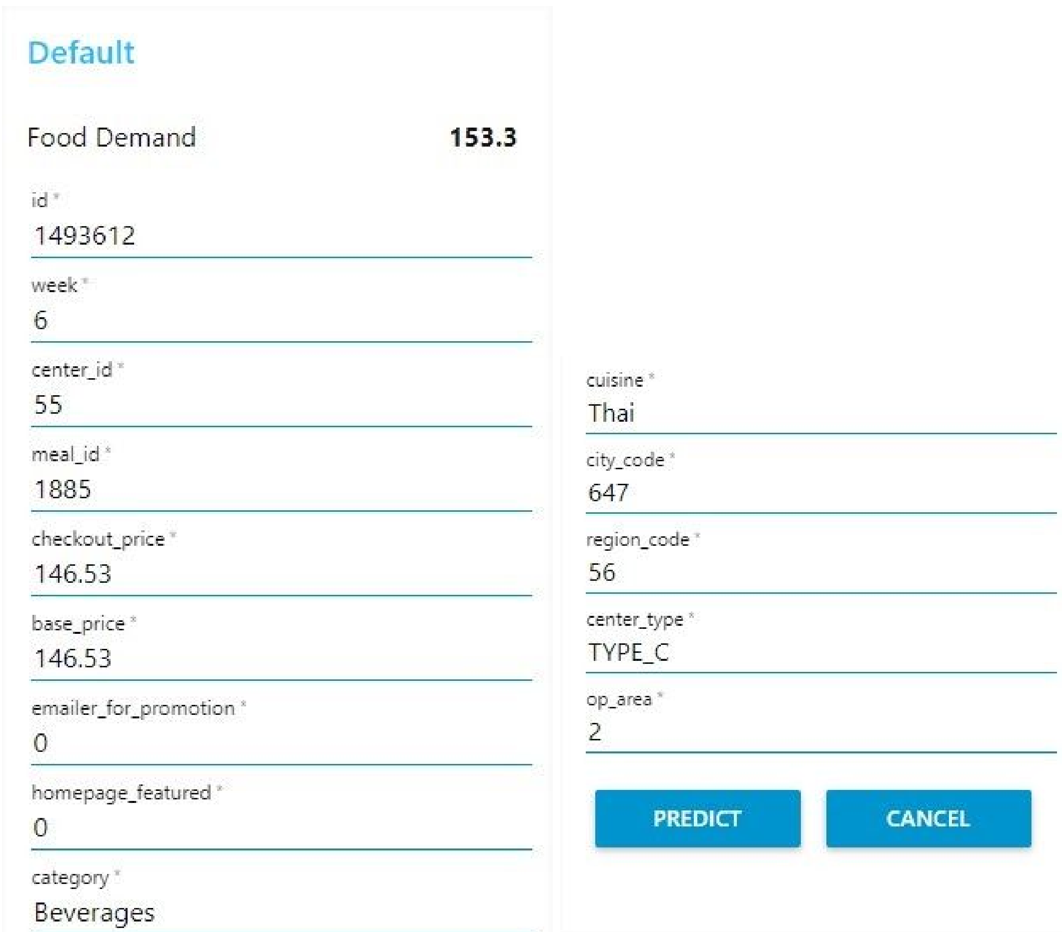

Fig- Node-Red Dashboard

4. EXPERIMENTAL INVESTIGATIONS

In this part we are going to investigate our model and look at our model performance. For any company having values as following "id":1493612, "week":6, "center_id":55, "meal_id":1885, "checkout_price":146.53, "base_price":146.53, "emailer_for_promotion":0, "homepage_featured":0, "category":Beverages, "cuisine":Thai, "city_code":647, "region_code":56, "center_type":TYPE_C, "op_area":2 gives us the result of food demands num_order value as **153.3.**

The Holdout RMSE *(Optimized) is* 151.792

5. FLOWCHART

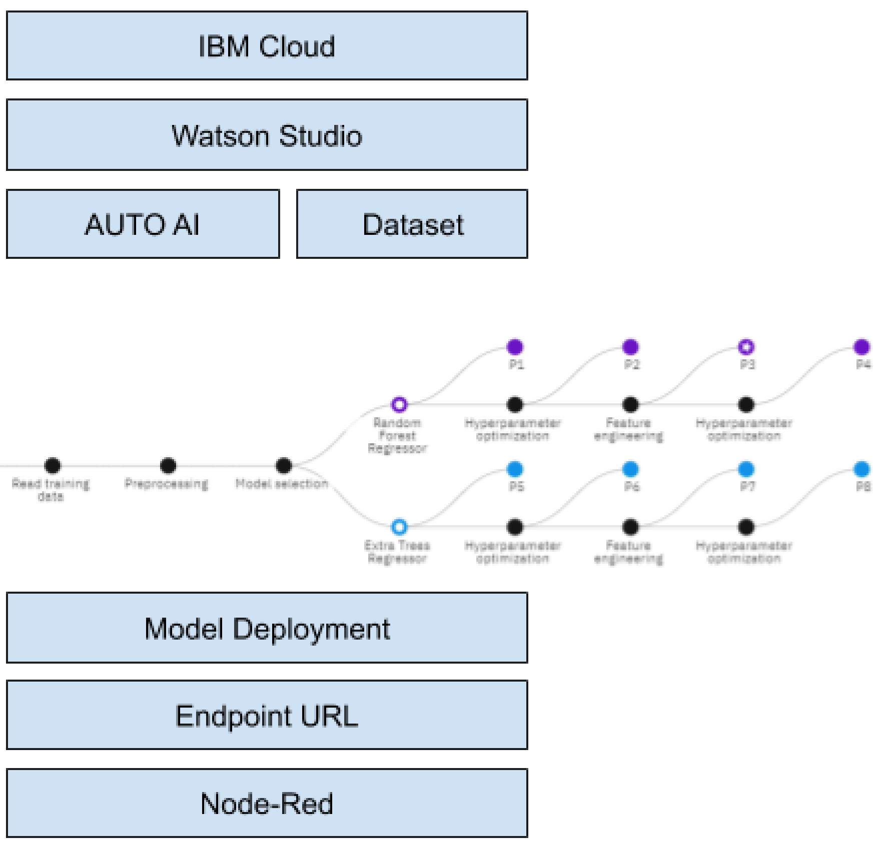

Fig-Flowchart

This is the workflow of the process and services used for building and deploying the model.

6. Novelty and Uniqueness

6.1 Business intelligence tool integrations.
6.2 Increase forecast accuracy.
6.3 No-cost system customizations.
6.4 Increase customer fill rates / Prevented stock-outs.
6.5 View forecasts by units, price, cost, weight, volume, pallets, profit, or any other user-definable measure.
6.6 Reduce overall forecasting time.

7. ADVANTAGES & DISADVANTAGES

7.1 Advantage

7.1.1 It is very beneficial for business purposes and by using these companies they increase their outcomes and get more benefit and efficiently use the available resources.
7.1.2 In the retail industry, demand forecasting is one of the main problems of supply chains to optimize stocks, reduce costs, and increase sales, profit, and customer loyalty. By this, they can overcome these problems.
7.1.3 This issue, there are several methods such as time series.
7.1.4 Analysis and machine learning approach to analyze and.
Learn complex interactions and patterns from historical data.

7.2 Disadvantage

Keep it simple.

8.APPLICATIONS

The applications are

8.1 Daily food demand analysis

8.2 Economy analysis

8.3 food demand affecting factors analysis

8.4 food price fac factors analysis

9. CONCLUSION

Finally,we are able to predict food demand using Auto AI and check what factors affect the food demand. It will be easier for a company to determine the demand for food in the market. This will help in suggesting a company which type of food people like. The outcome of this project is a dashboard which is easy to use.

10. FUTURE SCOPE

This will help in suggesting a company which food should be more demanding for more profit and for launching any new product in the market.

11. BIBLIOGRAPHY

References

11.1 IBM services

11.1.1 DashBoard - https://cloud.ibm.com/

11.1.2 Resources - https://cloud.ibm.com/resources

11.1.3 Watson Studio -

https:/bookdown.org/caoying4work/watsonstudio-workshop/setup.html

https://cloud.ibm.com/services/data-science-experience/crn%3Av1%3Abluemix%3Apublic%3Adata-science-experience%3Aeu-gb%3Aa%2F122c5402e42945b4851be75e266abe3a%3A48e73ece-1405-414d-bee3-120421f9ba99%3A%3A?paneId=manage

11.1.4 Node-Red- https://node-red-dqwgr.eu-gb.mybluemix.net/red/#flow/fe53c88d.b0d238

11.1.5 Node-Red Dashboard -
https://node-red-dqwgr.eu-gb.mybluemix.net/ui/#!/0?socketid=_7zSab04BpIdCOX5AAAj

11.1.6 Watson Machine Learning -

https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html

https://cloud.ibm.com/services/pm-20/crn%3Av1%3Abluemix%3Apublic%3Apm-20%3Aeu-gb%3Aa%2F122c5402e42945b4851be75e266abe3a%3A9a65746c-b0ae-47e9-9856-b1a8764d5bf1%3A%3A?paneId=manage

11.1.7 Notebook -

https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#create-notebook

https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/f6344e28-1fdc-4011-9d56-c021d48d2fbd/view?access_token=22dbc4704ffbabcce895767f65eb82a2d0ba66cb58e2dcf3878e53d47294ef4d

11.2 Dataset - https://www.kaggle.com/kumarajarshi/life-expectancy-who/data

APPENDIX

A. Source code

AUTO AI Notebook

https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/c28635aa-8f1c-4d44-8759-1e4b84e6c3be/view?access_token=bc3c08daa52350d76f33e37903a9bc190ff05232bf256611f4de5230372e339c

IBM AutoAI-SDK Auto-Generated Notebook v1.12.4

In [ ]:

```
!pip uninstall watson-machine-learning-client -y
```

In [ ]:

```
!pip install -U watson-machine-learning-client-V4
```

In [ ]:

```
!pip install -U autoai-libs
```

In [ ]:

```
# @hidden_cell
from watson_machine_learning_client.helpers import DataConnection, S3Connection, S3Location

training_data_reference = [DataConnection(
    connection=S3Connection(
        api_key='3YkBVtM-rCMx4RTe6bBfoXEOB--yX3UkfHXGXHOkUoet',
        auth_endpoint='https://iam.bluemix.net/oidc/token/',
        endpoint_url='https://s3.eu-geo.objectstorage.softlayer.net'
    ),
        location=S3Location(
        bucket='demandforecasting-donotdelete-pr-b9dkqslp3mmby9',
        path='data_asset/goods_prediction_shaped_4425a164.csv'
    ))
]
training_result_reference = DataConnection(
    connection=S3Connection(
        api_key='3YkBVtM-rCMx4RTe6bBfoXEOB--yX3UkfHXGXHOkUoet',
        auth_endpoint='https://iam.bluemix.net/oidc/token/',
        endpoint_url='https://s3.eu-geo.objectstorage.softlayer.net'
    ),
    location=S3Location(
        bucket='demandforecasting-donotdelete-pr-b9dkqslp3mmby9',

path='auto_ml/88560f76-3736-43dd-ba8c-c30e1012c3f8/wml_data/be9ff830-09de-4e9f-84e1-dfd5bc6d81c8/data/automl',

model_location='auto_ml/88560f76-3736-43dd-ba8c-c30e1012c3f8/wml_data/be9ff830-09de-4e9f-84e1-dfd5bc6d81c8/data/automl/cognito_output/Pipeline1/model.pickle',

training_status='auto_ml/88560f76-3736-43dd-ba8c-c30e1012c3f8/wml_data/be9ff830-09de-4e9f-84e1-dfd5bc6d81c8/training-status.json'
```

```
))
```

In [ ]:

```
experiment_metadata = dict(
   prediction_type='regression',
   prediction_column='num_orders',
   test_size=0.1,
   scoring='neg_root_mean_squared_error',
   csv_separator=',',
   excel_sheet=0,
   max_number_of_estimators=2,
   training_data_reference = training_data_reference,
   training_result_reference = training_result_reference)

pipeline_name='Pipeline_3'
```

In [ ]:

```
from watson_machine_learning_client.experiment import AutoAI

optimizer = AutoAI().runs.get_optimizer(metadata=experiment_metadata)
```

In [ ]:

```
pipeline_model = optimizer.get_pipeline(pipeline_name=pipeline_name)
```

In [ ]:

```
pipeline_model.pretty_print(combinators=False, ipython_display=True)
```

In [ ]:

```
pipeline_model.visualize()
```

In [ ]:

```
training_df, holdout_df = optimizer.get_data_connections()[0].read(with_holdout_split=True)

train_X = training_df.drop([experiment_metadata['prediction_column']], axis=1).values
train_y = training_df[experiment_metadata['prediction_column']].values

test_X = holdout_df.drop([experiment_metadata['prediction_column']], axis=1).values
y_true = holdout_df[experiment_metadata['prediction_column']].values
```

In [ ]:

```
from sklearn.metrics import r2_score

predictions = pipeline_model.predict(test_X)
score = r2_score(y_true=y_true, y_pred=predictions)
```

```
print('r2_score: ', score)
```

In [ ]:

```
wml_credentials = {
  "apikey": "",
  "iam_apikey_description": "",
  "iam_apikey_name": "",
  "iam_role_crn": "r",
  "iam_serviceid_crn": "",
  "instance_id": "",
  "url": ""
}
```

In [ ]:

```
from watson_machine_learning_client.deployment import WebService

service = WebService(wml_credentials)

service.create(
    model=pipeline_model,
    metadata=experiment_metadata,
    deployment_name=f'{pipeline_name}_webservice'
)
```

Deployment object could be printed to show basic information:

In [ ]:

```
print(service)
```

To be able to show all available information about deployment use .get_params() method:

In [ ]:

```
service.get_params()
```

Score webservice

You can make scoring request by calling score() on deployed pipeline.

In [ ]:

```
predictions = service.score(payload=holdout_df.drop([experiment_metadata['prediction_column']], axis=1).iloc[:10])
predictions
```

Node - Red

[{"id":"fe53c88d.b0d238","type":"tab","label":"Flow
1","disabled":false,"info":""},{"id":"49e21a49.b26a04","type":"function","z":"fe53c88d.b0d238","name":"pre token","func":"//make user given
values as global
variables\nglobal.set(\"u\",msg.payload.u);\nglobal.set(\"b\",msg.payload.b);\nglobal.set(\"c\",msg.payload.c);\nglobal.set(\"d\",msg.payload.d);\n
global.set(\"e\",msg.payload.e);\nglobal.set(\"f\",msg.payload.f);\nglobal.set(\"g\",msg.payload.g);\nglobal.set(\"h\",msg.payload.h);\nglobal.set(\
"i\",msg.payload.i);\nglobal.set(\"j\",msg.payload.j);\nglobal.set(\"k\",msg.payload.k);\nglobal.set(\"l\",msg.payload.l);\nglobal.set(\"m\",msg.pay
load.m);\n//following are required to receive a token\nvar
apikey=\"OhkeaxPMV3pC9KifvPg5Ho8_YF05KSSQNLfWrqJ-8ymh\";\nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\
"};\nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey};\nreturn
msg;\n","outputs":1,"noerr":0,"x":220,"y":100,"wires":[["d3470fff.db57c"]]},{"id":"e0917cae.6159c","type":"http
request","z":"fe53c88d.b0d238","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://eu-gb.ml.cloud.ibm.com/v4/deployments/4
50441d5-d18b-4150-92b9-75a4d2ac9a26/predictions","tls":"","persist":false,"proxy":"","authType":"","x":410,"y":180,"wires":[["29bfff20.fd8de
","d26a52b2.a80e"]]},{"id":"6bc65785.ff0568","type":"debug","z":"fe53c88d.b0d238","name":"","active":true,"tosidebar":true,"console":false,"t
ostatus":false,"complete":"payload","targetType":"msg","x":750,"y":280,"wires":[]},{"id":"d26a52b2.a80e","type":"function","z":"fe53c88d.b0d
238","name":"getFrom Endpoint","func":"msg.payload=msg.payload.predictions[0].values[0][0]\nreturn
msg;","outputs":1,"noerr":0,"x":490,"y":280,"wires":[["6bc65785.ff0568","875d1aa2.e45228"]]},{"id":"29bfff20.fd8de","type":"debug","z":"fe5
3c88d.b0d238","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":730,"y":80,
"wires":[]},{"id":"52fa9fe.efc366","type":"function","z":"fe53c88d.b0d238","name":"sendTo Endpoint","func":"//get token and make
headers\nvar token=msg.payload.access_token;\nvar instance_id=\"9ee41e1f-a5a4-41eb-b7be-d343fc0fc4d9\"\nmsg.headers={'Content-Type':
'application/json',\"Authorization\":\"Bearer \"+token,\"ML-Instance-ID\":instance_id}\n\n//get variables that are set earlier\nvar u =
global.get(\"u\");\nvar a = global.get(\"a\");\nvar b = global.get(\"b\");\nvar c = global.get(\"c\");\nvar d = global.get(\"d\");\nvar e =
global.get(\"e\");\nvar f = global.get(\"f\");\nvar g = global.get(\"g\");\nvar h = global.get(\"h\");\nvar i = global.get(\"i\");\nvar j =
global.get(\"j\");\nvar k = global.get(\"k\");\nvar l = global.get(\"l\");\nvar m = global.get(\"m\");\n//send the user values to service
endpoint\nmsg.payload=\n{\"input_data\": [{\"fields\": [\"id\", \"week\", \"center_id\", \"meal_id\", \"checkout_price\", \"base_price\",
\"emailer_for_promotion\", \"homepage_featured\", \"category\", \"cuisine\", \"city_code\", \"region_code\", \"center_type\",
\"op_area\"],\n\"values\":[[u,a,b,c,d,e,f,g,h,i,j,k,l,m]]}]};\n\nreturn
msg;\n","outputs":1,"noerr":0,"x":190,"y":180,"wires":[["e0917cae.6159c"]]},{"id":"d3470fff.db57c","type":"http
request","z":"fe53c88d.b0d238","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://iam.cloud.ibm.com/identity/token","tls":""
,"persist":false,"proxy":"","authType":"basic","x":390,"y":100,"wires":[["52fa9fe.efc366"]]},{"id":"875d1aa2.e45228","type":"ui_text","z":"fe53
c88d.b0d238","group":"837bad64.cc16c","order":1,"width":0,"height":0,"name":"","label":"Food
Demand","format":"{{msg.payload}}","layout":"row-spread","x":700,"y":340,"wires":[]},{"id":"2cb98ae.a084176","type":"ui_form","z":"fe53c
88d.b0d238","name":"","label":"","group":"837bad64.cc16c","order":0,"width":0,"height":0,"options":[{"label":"id","value":"u","type":"number"
,"required":true,"rows":null},{"label":"week","value":"a","type":"number","required":true,"rows":null},{"label":"center_id","value":"b","type":"
number","required":true,"rows":null},{"label":"meal_id","value":"c","type":"number","required":true,"rows":null},{"label":"checkout_price","va
lue":"d","type":"number","required":true,"rows":null},{"label":"base_price","value":"e","type":"number","required":true,"rows":null},{"label":"e
mailer_for_promotion","value":"f","type":"number","required":true,"rows":null},{"label":"homepage_featured","value":"g","type":"number","re
quired":true,"rows":null},{"label":"category","value":"h","type":"text","required":true,"rows":null},{"label":"cuisine","value":"i","type":"text","r
equired":true,"rows":null},{"label":"city_code","value":"j","type":"number","required":true,"rows":null},{"label":"region_code","value":"k","typ
e":"number","required":true,"rows":null},{"label":"center_type","value":"l","type":"text","required":true,"rows":null},{"label":"op_area","value"
:"m","type":"number","required":true,"rows":null}],"formValue":{"u":"","a":"","b":"","c":"","d":"","e":"","f":"","g":"","h":"","i":"","j":"","k":"","
l":"","m":""},"payload":"","submit":"Predict","cancel":"cancel","topic":"","x":70,"y":100,"wires":[["49e21a49.b26a04"]]},{"id":"837bad64.cc16
c","type":"ui_group","z":"","name":"Default","tab":"d5d09625.e59c28","order":1,"disp":true,"width":"6","collapse":false},{"id":"d5d09625.e59c
28","type":"ui_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]