IBM Hack Challange 2020: Warehouse Optimisation

Introduction

Overview

Any time, and especially during tumultuous times like now, the planning for stocking of warehouses is of the utmost importance to avoid any loss both in material and financial to the owner. By employing current trends like machine learning in the market, it can easily be statistically modeled to predict the demand and supply of raw materials required for their management.

Purpose

The requirement for such a system is a necessity for small businesses so that they can handle difficult situations without going into financial trouble. Hence we propose a few additional features which will better the project:
A feature that helps to predict the cost of the raw material required and how it will increase or decrease so that the warehouse management can act accordingly to minimize their losses.
The second greatest issue faced by warehouses is pests and the model can have a feature that will analyze the seasonal and local geographical data to provide the management with a pest warning so that they can be prepared well in advance so that they don't suffer a loss.

Literature survey

Existing problems and available publications

1. Intelligent algorithms for warehouse management. Eleonora Bottani, Roberto Montanari, Marta Rinaldi, Giuseppe Vignali.
2. Ensemble multi-objective biogeography-based optimization with application to automated warehouse scheduling
3. Batching orders in warehouses by minimizing travel distance with genetic algorithms
4. An overview of warehouse optimisation. (http://ijates.org/index.php/ijates/article/view/61)
5. Multi-commodity warehouse location and distribution planning with inventory consideration.- Ronald G. Askin, Ilaria Baffo & Mingjun Xia.

Proposed solution

We propose a solution that will help the owner to predict the demand for the raw material not only based on previous data but also the data generated every day so that the model keeps up with the current trends and the dataset does not become “outdated”.

The model will be trained to predict the demand and provide both visual and numerical representation of the prediction for the user to understand the trends and make decisions accordingly.

Theoretical Analysis

1. Gathering and reviewing the dataset
2. Cleaning and pre-processing the dataset
3. Data Analysis
4. Studying different ML models
5. Using IBM Cloud
6. Developing the application using Node-RED
7. Testing and Deploying the application

Task 1: Gathering and reviewing the dataset

Food Demand Forecast Challenge Dataset from Kaggle
This dataset consists of following columns:

- Week
- Fulfilment Centre
- Meal
- Checkout Price
- Base Price
- Promotion Email
- Homepage feature
- Number of orders (target)

Thus given the features above an ML model will be able to detect number of orders or food demand. These features are experimental and can in future be changed to set of features specific to the warehouse location to predict demand on them.

Train data - 457K examples
Test data - 32.6K examples

Task 2: Cleaning and pre-processing the Dataset

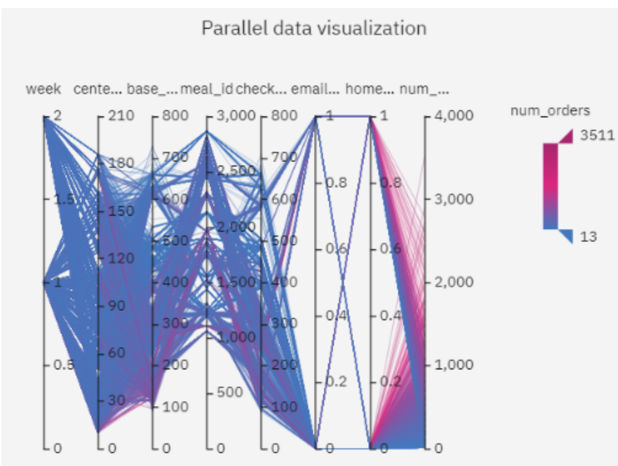
This dataset has separated `fulfilment-center` and `meal` information into separate csv and their unique IDs have been substituted in the train and test dataset.

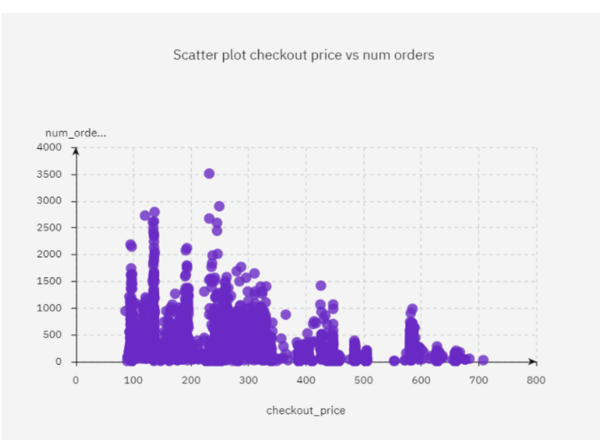
Pre-processing

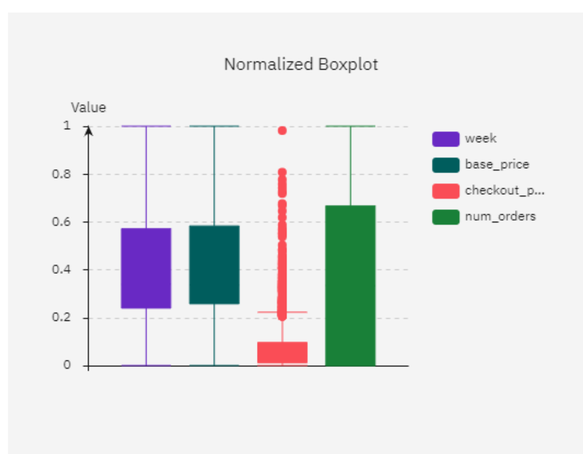
- Normalisation - Bringing all the values between 0 and 1

Task 3: Data Analysis

1. Plot the change in demands for a fulfilment center over weeks
2. Change in demands for a meal
3. Difference in base price and checkout price and its relation to demand
4. Email and Homepage Featuring relation to demand







Task 4: Studying different ML models

Since the dataset chosen is not a time-series dataset, thus we can not use models like LSTM, ARIMA and Prophet.

We have implemented a Random Forrest model using IBM AUTOAI which predicts number of orders for a particular product based on the week number, meal id, base price, checkout price, whether an email was sent for promotion or not and whether the product was featured on the homepage of the store.

The models trained by AUTO AI were :
Random Forrest Regressor - HPO1 + FE : 185

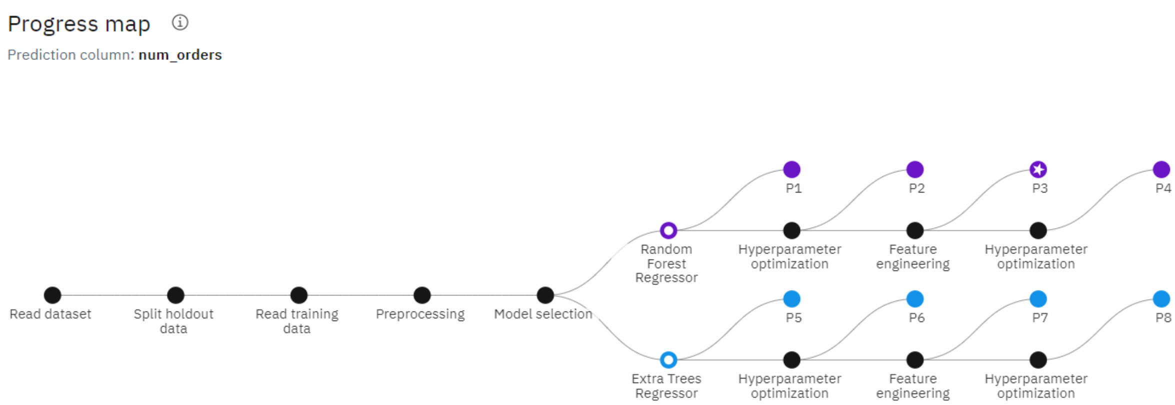


Figure - Chart of different models tested

Rank	Name	Algorithm	RMSE (Optimized)	Enhancements
★ 1	Pipeline 3	Random Forest Regressor	185.202	HPO-1 FE
2	Pipeline 4	Random Forest Regressor	185.202	HPO-1 FE HPO-2
3	Pipeline 7	Extra Trees Regressor	191.555	HPO-1 FE
4	Pipeline 8	Extra Trees Regressor	191.555	HPO-1 FE HPO-2
5	Pipeline 1	Random Forest Regressor	191.800	*None*
6	Pipeline 2	Random Forest Regressor	191.800	HPO-1
7	Pipeline 5	Extra Trees Regressor	201.305	*None*
8	Pipeline 6	Extra Trees Regressor	201.305	HPO-1

Figure - Different models trained by AUTOAI and their RMSE scores

Task 4: Model Optimisation

The model was built using IBM AutoAI which does hyperparameter optimisation and feature engineering while building the model itself hence does not require any external optimisation.

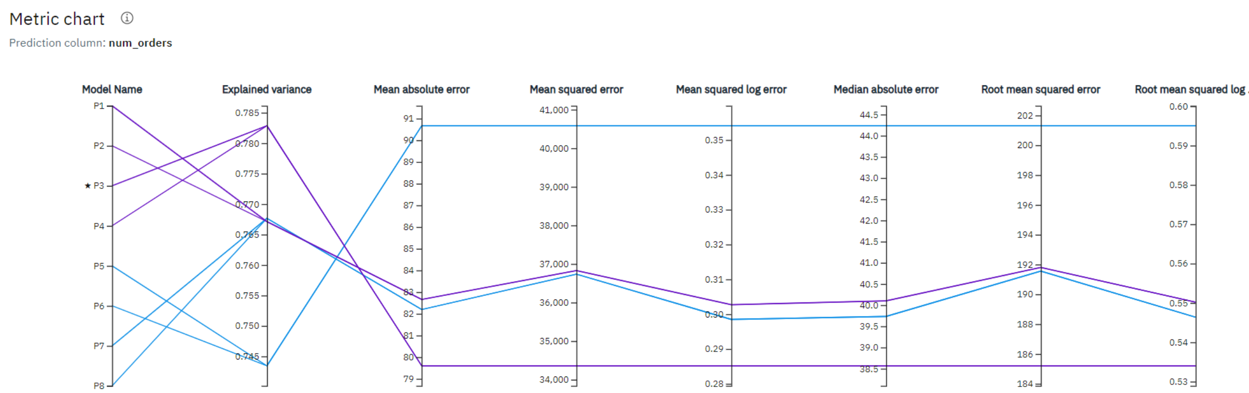
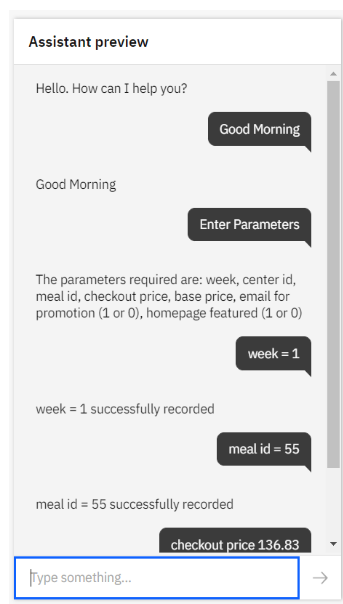


Figure - Model metrics comparassion

Task 5: Using IBM Cloud



The following IBM Cloud services were used in this project:

1. Watson Machine Learning
2. Watson Assistant
3. Watson Text to Speech
4. Watcon Speech to Text
5. Internet of Things
6. Node Red
7. Watson Studio
8. IBM CognosDashboard Embedded

Task 6: Developing the application using Node-RED

Node-RED provided a visual way to build applicaition workflows. It was used to build following flows for a warehouse user.

A warehouse can enter all the information regarding a food item by filling a form. This information is then passed to IBM AutoAI model by making an API call through http requests with the food data filled in the form. AutoAI then returns the prediction of number of such food item to be stocked. This is displayed as a prediction to the user. Once the actual food items are dispatched, the number is displayed in the gauge; this real number may not be same as the prediction. To simulate this, a function is used to randomly add a number between 0 and 10 to the prediction. This "real" data is then plotted in a graph to display the history of dispatched items. This real time data can be used to retrain the model and produce more accurate reaults over time.

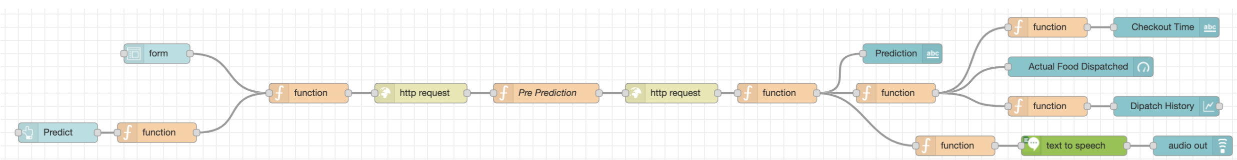


Figure - Main flow for adding food detail, predicting number of orders, and displaying the actual dispatched food items

IBM Watson Assistant was used to make entering food details more natural. As the user enters the detail, Assistant parsed the values and they where recorded from watsons output. These values were shown as preview for verification. A predict button when pressed takes all these values and uses the same flow as above to use AutoAI for prediction. Watson can be used for more natural communication with the application.

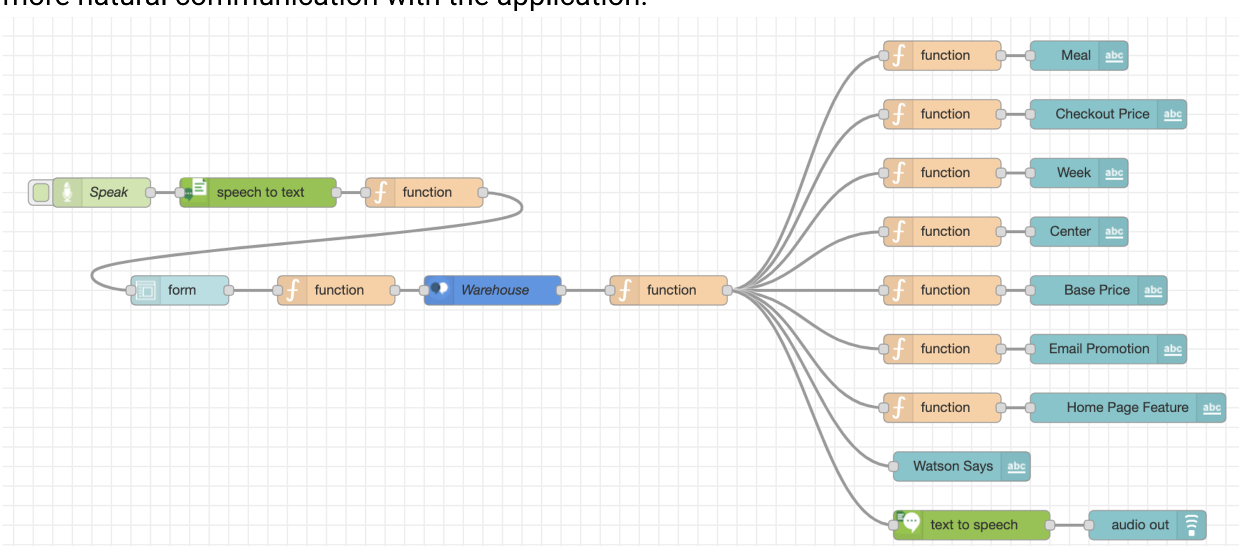


Figure - Flow for Watson Assistant interaction and data preview for prediction.

To monitor the actual warehouse conditions and thus the food health, IoT was used to sense temperature, humidity and object temperature. This information was then displayed to the warehouse dashboard under Health tab. Graphs for each were used to record and display the history. If either of these parameters go beyond the set normal, the status gets changed from Normal to Perishable. If the environment is perishable then the user gets a notification from anywhere within the app.

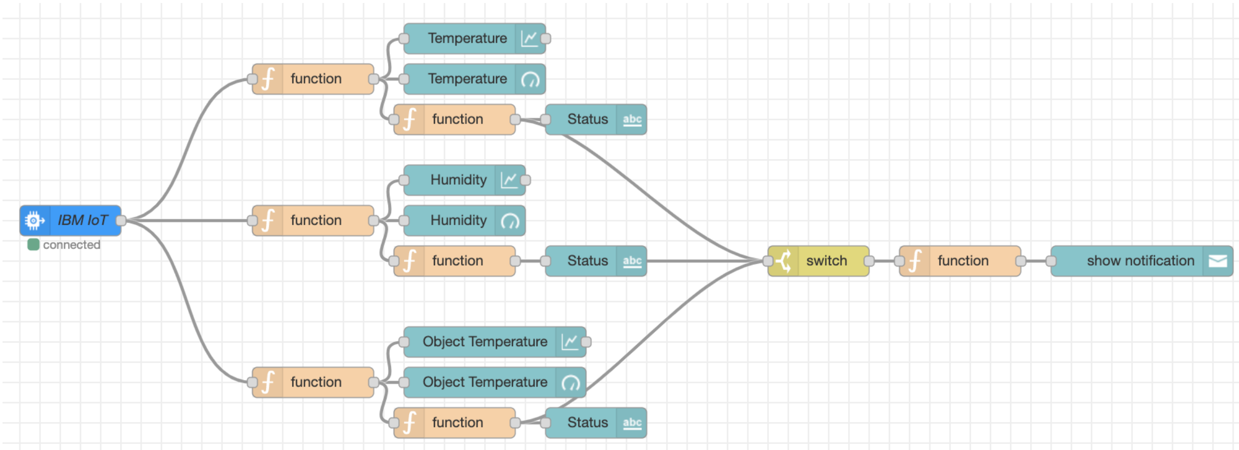


Figure - Flow for displaying IoT sensor data, recording the changes in graph and showing notification when the environment is abnormal

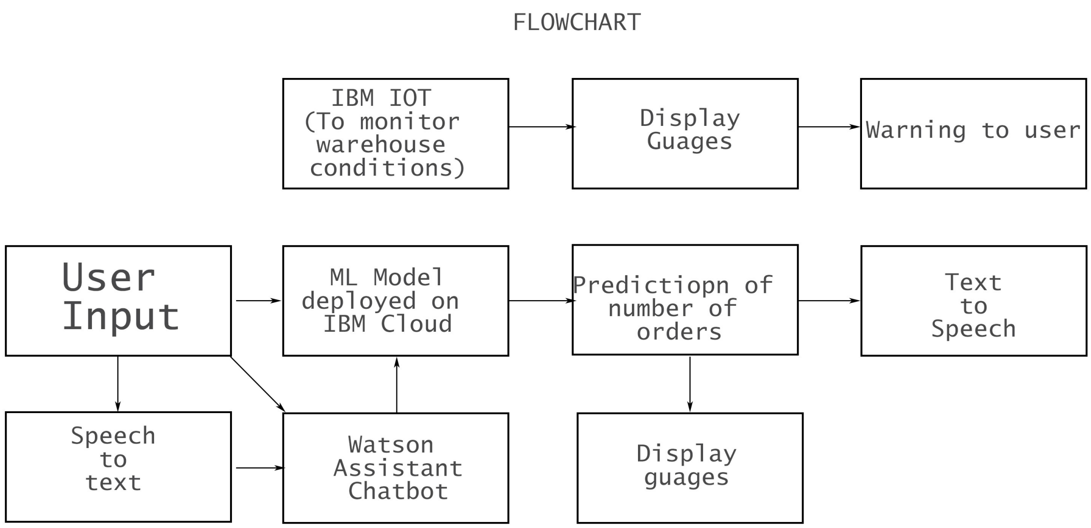
Experimental Investigantion

While doing the projects we worked with various new services offered by IBM Cloud like IBM CognosDashboard Embedded.
We also talked with various warehouse owners and shopkeepers about what they expect from such a product and the cost they are willing to pay for its benifits.
The responses can be summarized as:

1. Warehouse owners and shopkeepers are unaware of the technologica ladvantages in the field of Machine Learning and do not even the existance of such services being available.
2. The owners were ready to pay a moderate price even in the tumultuous times like now where the whole industry is in a disarray due to the COVID-19 crisis, if it means that it will help them turn a profit or reduce losses sometime in the near future.
3. The requirement of such a platform is that it should be such that even an illiterate person can easily undestand and operate it.

Keeping these points in our minds we went ahead with the project.

Flowchart



Result

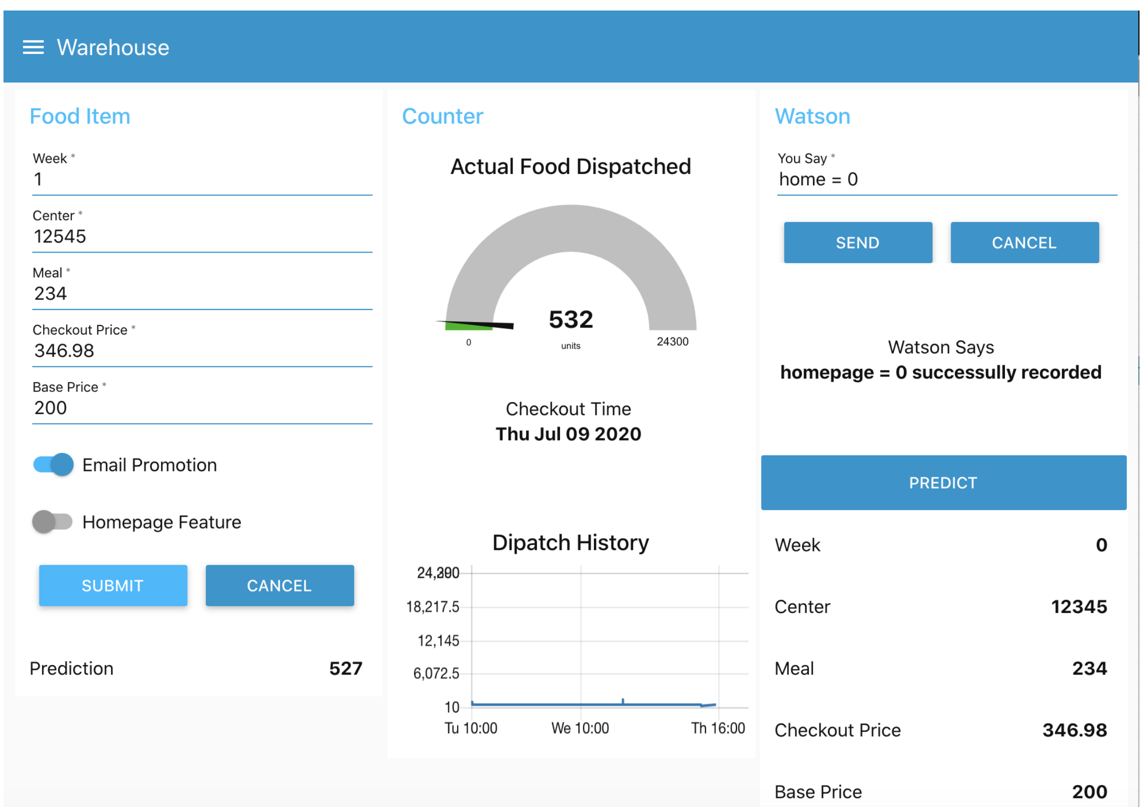
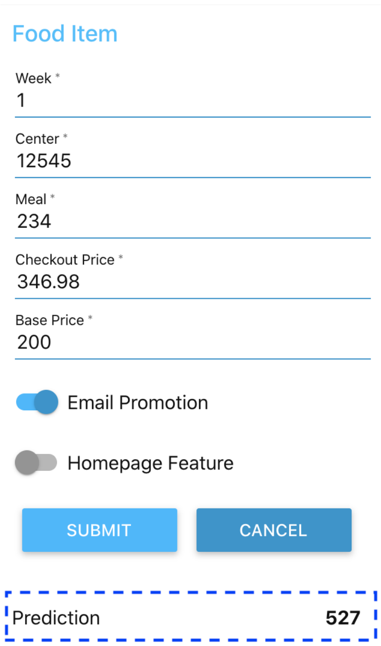
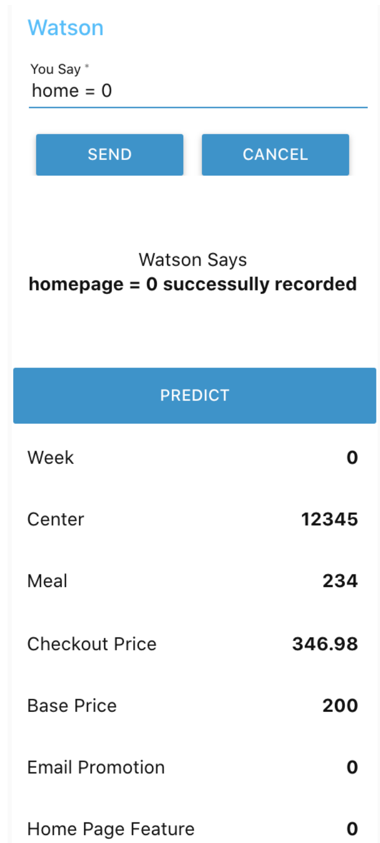


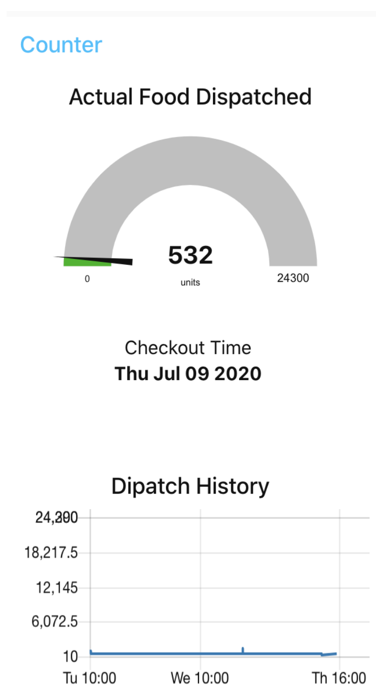
Figure - Warehouse main page



The warehouse dashboard allows a user to enter item details into application. These details then return prediction of the number of such food items that the warehouse must stock to cater to the demand in near future.



If user chooses to, the information can be entered more naturally by conversing with Watson assistant. All the responses from the assistant appear below the user entered text.
For user's convineance, all the information recorded by the assistant is previewed in the bottom.
Once the user has confirmed the food item details, user clicks the Predict button to get the prediction in the same area below the food form.



When the actual food items are dispatched, the data is represented in Actual Food Dispatched gauge with the date of recording. This information may not be same as predicted one, so, it can be sent to the model as a new data point. This way the model can get more accurate and is up to date with the warehouse.
All the actual dispatch history for a week is represented as a graph to show the trend.

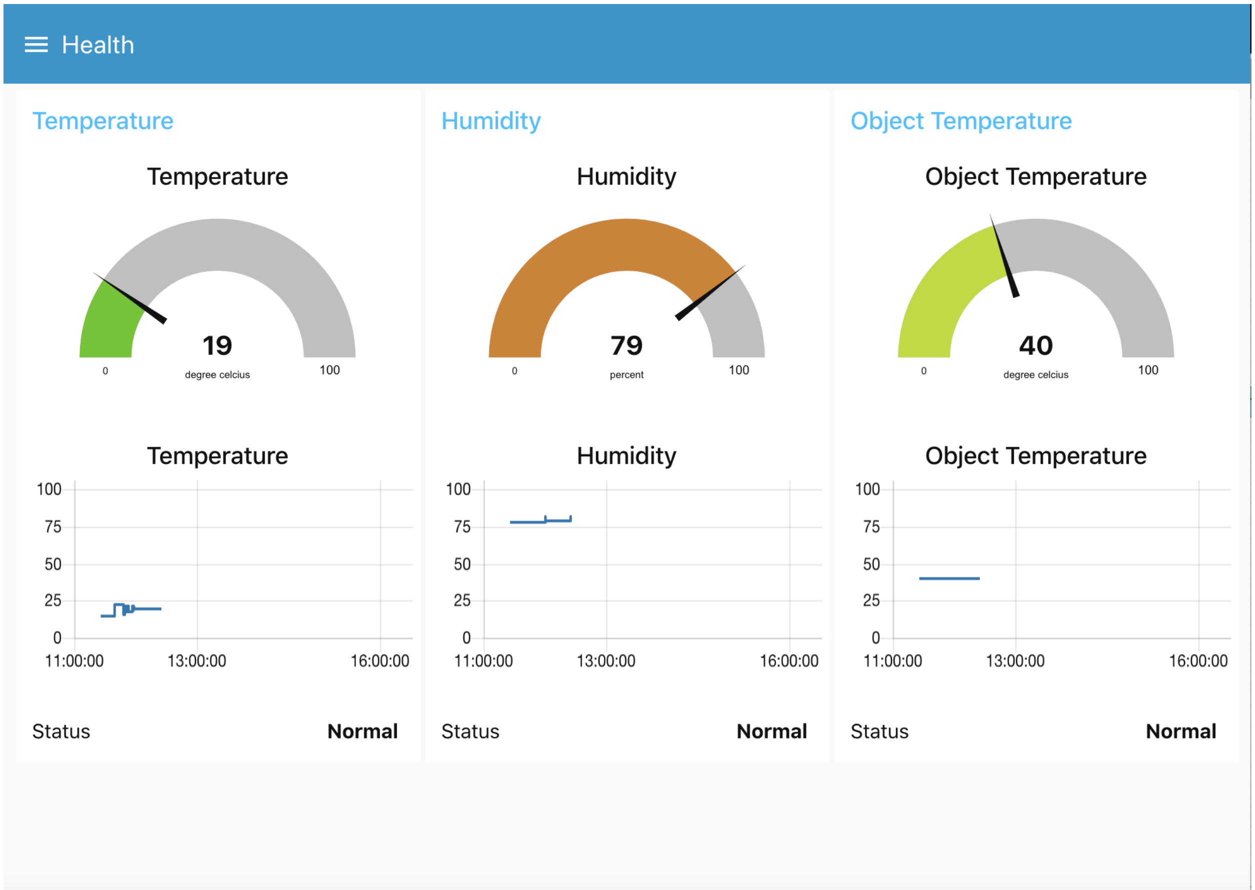


Figure - Warehouse health page

The Health tab displays the environmental condition of the warehouse in real time. Each graph represents the changes throughout the day. When the conditions are away from pre-set normal, status of that particular parameter changes and the user is notified from anywhere within the app.

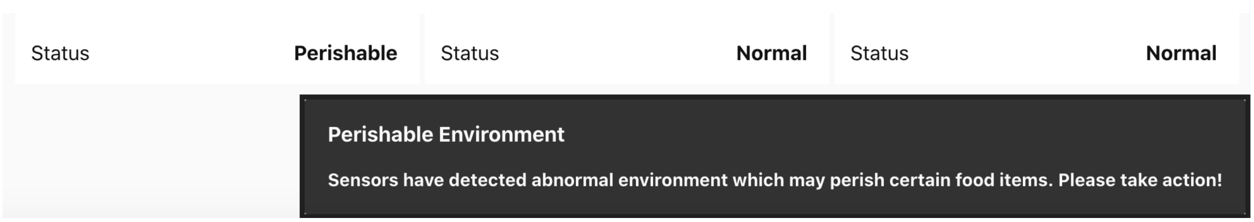


Figure - Perishable Environment Notification

Advantages

1. Help warehouse to be more efficient by preparing for the demand
2. Increase the profits made due to reduced losses and predicted surge
3. Reduce wastage of perishable food items
4. Reduce other costs and efforts for making the items
5. Share the predictions with the entire supply chain

Disadvantages

1. Initial setup cost for compute power and integration with the system
2. Inability to detect sudden market changes

Applications

1. Warehouse stocking perishable items like food can be most benefited
2. Other warehouses
3. Any other organisation requiring to predict the demands for a product

Conclusion

The project is aimed at predicting the demand, and monitoring the situation at a warehouse to reduce the stacking problem and, importantly, wastage especially in perishable industry like food. This further increases operation efficiency like logistic eases few of the processes.

Warehouses are ridden with various issues and the majority portion of the loss is the direct result of lack of planning. Predicting demand helps warehouse prepare for it beforehand and efficiently. This also leads to increse in profits and reduces waste and efforts.

As more and more businesses enter the digital age, the necessity of Machine Learning to maintain a competitive edge in the market has become obvious. The project design is such that it will greatly improve the planning aspects in the business strategy of small businesses.

In current times the COVID-19 pandemic has left the world in disarray and local businesses are struggling to keep up with the trends due to stringent lockdown conditions. Thus by using the current data to analyze the situation, the warehouse owner can stock products according to deamnd thus increase profits, reducing hasel and reducing wastage.

In future, this solution can be integrated with the existing system to provide more integrated feel. Every new data point can be used for more accurate and adaptive prediction as the time progresses. Watson assistant can be trained on more intents and information.

Future Scope

The requirement for such a system is a necessity for small businesses so that they can handle difficult situations without going into financial trouble. Hence we propose a few additional features which will better the project:

A feature that helps to predict the cost of the raw material required and how it will increase or decrease so that the warehouse management can act accordingly to minimize their losses.

The second greatest issue faced by warehouses is pests and the model can have a feature that will analyze the seasonal and local geographical data to provide the management with a pest warning so that they can be prepared well in advance so that they don't suffer a loss.
Thus, we provide a one stop solution for optimized warehouse management of perishable goods for a food delivery company.

Acknowledgement

During the start of the project, we challenged ourselves to learn new technologies under IBM Cloud. We targeted our entire project to be made using just technologies in the Watson. Using the knowledge we gained during the bootcamp, we're successfully able to complete the project. For this we thank IBM for providing the opportunity and SmartInternz for supporting along the way.

References

1. Getting started with IBM Cloud
https://cloud.ibm.com/docs/runtimes/php/getting-started.html

2. Using IBM AUTOAI:
https://www.ibm.com/in-en/cloud/watson-studio/autoai

3. Using IBM Watson Assistant:
https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started

4. Using Watson text to speech
https://www.ibm.com/in-en/cloud/watson-text-to-speech#:~:text=What%20is%20Watson%20Text%20to,in%20multiple%20languages%20and%20tones.

5. Deploying ML models on IBM cloud:
https://cloud.ibm.com/docs/tutorials?topic=solution-tutorials-create-deploy-retrain-machine-learning-model

6. Getting started with NODE-RED:
https://nodered.org/docs/getting-started/

7. Deemand and forecasting models:
https://www.altexsoft.com/blog/demand-forecasting-methods-using-machine-learning/

8. LSTMS
https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

9. ARIMA:
https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/#:

Source Code

Link to GitHub
https://github.com/SmartPracticeschool/SBSPS-Challenge-633-Warehouse-optimisation

Link to Watson Assistant
https://web-chat.global.assistant.watson.cloud.ibm.com/preview.html?region=eu-gb&integrationID=5b4bd37d-e362-4765-9236-3cd33770c28e&serviceInstanceID=2ca66b95-b92d-4104-bc2c-19c6bd4772a1

Link to NodeRED application
https://node-red-ybzuf.eu-gb.mybluemix.net/ui/#!/0?socketid=f_BLHQuUbza86NG9AAAE

Link to the ML Model
https://dataplatform.cloud.ibm.com/ml/deployments/c3bcd08a-0c76-4b60-babe-6ed25346c482/test?projectid=c23bb30a-2e9a-41fb-ac91-b42679310df4&mlInstanceGuid=539da7e6-f6e6-4801-8fb3-9ae3e1cd4b38&context=wdp&flush=true&wmlv4=true

Link to IoT
http://watson-iot-sensor-simulator.mybluemix.net/