Project Report on

Optimized Warehouse Management of Perishable Goods for a Food Delivery Company

Prepared By

Team Name: Dexter
Team Members:
Neelam Somai (Team Leader)
Gayatri Patil
Gaurav Tirodkar
Yash Mate

Date: 15/07/2020

INDEX

1. INTRODUCTION

1.1 Overview

The food delivery system has to deal with lots of perishable food items. The most important task for such companies is to accurately forecast the weekly and daily demand. Too much inventory in the warehouse means more risk of wastage of food whereas less could lead to out-of-stock and push customers to seek solutions from the competitors. Since, the companies have to deal with perishable materials the procurement planning is utmost important .

1.2 Purpose

Your client is a meal delivery company which operates in multiple cities.They have various fulfillment centers in these cities for dispatching meal orders to their customers. The client wants you to help these centers with demand forecasting for upcoming weeks so that these centers will plan the stock of raw materials accordingly. 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.Secondly, staffing of the centers is also one area wherein accurate demand forecasts are really helpful.Given the following information,the task is to predict the demand for the next 10 weeks(Weeks: 146-155) for the center-meal combinations in the test set:

Historical data of demand for a product-center combination(Weeks:1 to 145) Product(Meal) features such as category,sub-category,current price and discount Information for fulfillment center like center area, city information etc.

2. LITERATURE SURVEY

2.1 Existing Problem

There are few apps available for optimized warehouse management with functionalities like inventory management, smooth workflow, scanning, shipping of goods, and tracking.

Snappi warehouse is one such application with key features as:

1. make the warehouse inventory quickly and easily, customizable
2. track goods shipped in and shipped out
3. create supply records
4. make quick picklist
5. barcode scanner

Another application with varied features is LoMag Warehouse Management

The features include:

1. import data from excel
2. create and restore a backup copy
3. The stock level at a given date and hour may be exported to Excel
4. Warehouse documents include goods received the note, goods issued note, and inventory
5. History of warehouse transfers for a chosen item

There is an application called Inventory Now with key features:

1. Get a snapshot of your inventory
2. Mark inventory as it is received, shipped and delivered
3. Get an overview of inventory at any time

Also, there are few more available apps like these but there is none with the main focus on perishable goods.

So, Our system will have unique features like:

1. Visualization on dashboard
2. Feedback for continuous improvement
3. Prediction of order to be placed depending on history, basically it will provide weekly or daily predictions.

2.2 Proposed Solution

XGBoost is an open source library providing a high-performance implementation of gradient boosted decision trees. An underlying C++ codebase combined with a Python interface sitting on top makes for an extremely powerful yet easy to implement package.

The performance of XGBoost is no joke — it's become the go-to library for winning many Kaggle competitions. Its gradient boosting implementation is second to none and there's only more to come as the library continues to garner praise.

Boosting Trees

With a regular machine learning model, like a decision tree, we'd simply train a single model on our data set and use that for prediction. We might play around with the parameters for a bit or augment the data, but in the end we are still using a single model. Even if we build an ensemble, all of the models are trained and applied to our data separately.

Boosting, on the other hand, takes a more iterative approach. It's still technically an ensemble technique in that many models are combined together to perform the final one, but takes a more clever approach.

Rather than training all of the models in isolation of one another, boosting trains models in succession, with each new model being trained to correct the errors made by the previous ones. Models are added sequentially until no further improvements can be made.

The advantage of this iterative approach is that the new models being added are focused on correcting the mistakes which were caused by other models. In a standard ensemble method where models are trained in isolation, all of the models might simply end up making the same mistakes!

Gradient Boosting specifically is an approach where new models are trained to predict the residuals (i.e errors) of prior models. I've outlined the approach in the Fig 1. shown below.

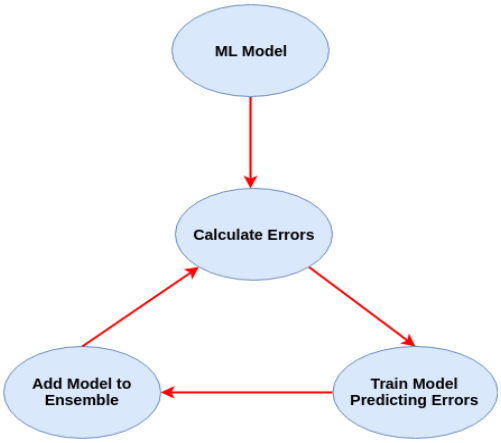

Fig 1. Gradient Boosting Approach

Tone Analyzer :

Watson Tone Analyzer

Conduct social listening

Analyze emotions and tones in what people write online, like tweets or reviews.

Enhance customer service

Monitor customer service and support conversations so you can respond to your customers appropriately and at scale. See if customers are satisfied or frustrated, and if agents are polite and sympathetic.

Integrate with chatbots

Enable your chat bot to detect customer tones so you can build dialog strategies to adjust the conversation accordingly.

The growth of web contributes a huge quantity of user created content such as customer feedback, opinions and reviews. Sentiment analysis in web embraces the problem of aggregating data in the web and extraction about opinions. Studying the opinions of customers helps to determine the people feeling about a product and how it is received in the market. Various commercial tools are available for sentiment analysis. In this paper, we propose a system which classifies the reviews on a scale of emotions like joy,anger,surprise, formal, confident based on the sentiments in the words. The groups of words used to make a decision to rate the reviews are displayed as word cloud.

Data Visualization:

Visualization of data on the dashboard will help the warehouse managers to analyze and understand the data easily and manage their products and orders accordingly. As a result we provide user friendly interface with data visualization for the convenience of the users in understanding the prediction and taking business decisions accordingly.

3. THEORITICAL ANALYSIS

3.1 Block Diagram

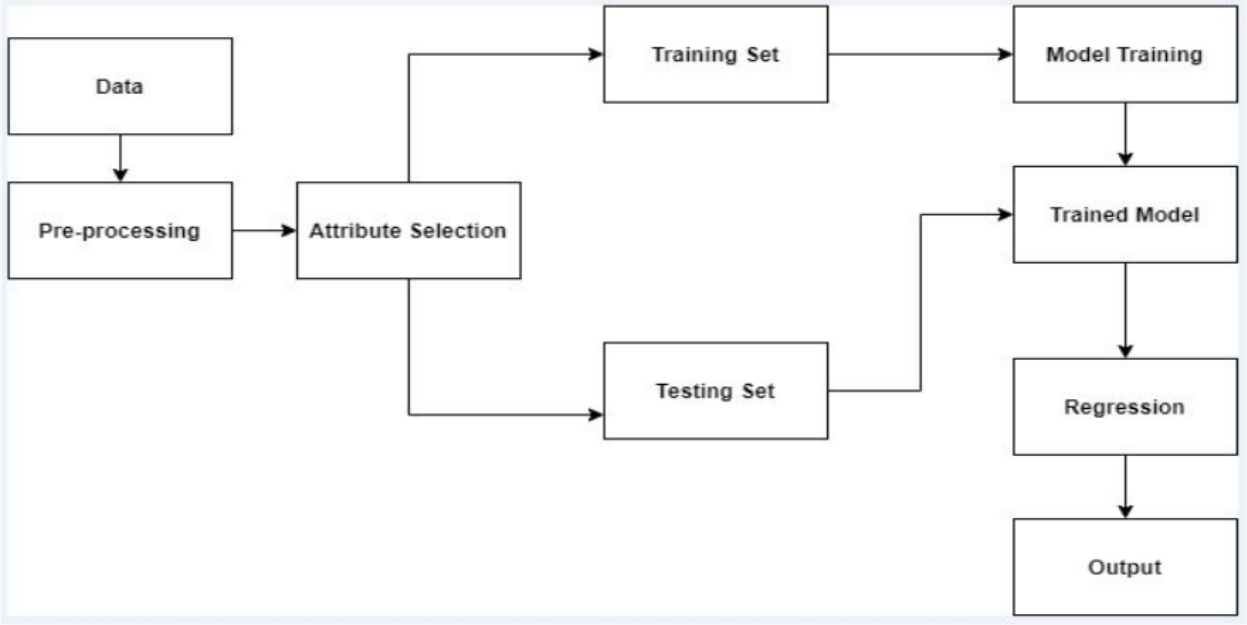

Fig. 2: Block Diagram Of ML Model

Download Data set :

Data set was provided by kaggle, this data set was in four files namely train.csv, test.csv, fulfillment.csv and meal_info.csv.

Train.csv : Train.csv is the original data set which will be used for training the model. This data set provides information about the meal sold,the price of each meal and the number of orders in that week.

Fulfillment_info and meal_info :

Fulfillment info talks about the area which is being serviced with these meals.This data set gives information about the region,city and the center where these warehouses are located.

Meal info talks about which meal is chosen i.e. the cuisine, whether it is Thai,Indian,etc and also what type of meal, is it a beverage,soup,extras,etc.

Model Implementation Using Different Algorithms :

After the top features were finalized, now is the time to pass the data through different Machine Learning algorithms and test it's accuracy. We will train out model with different algorithms like

XGB,Lasso,Decision Tree Regression and CatBoost.
Out of all these, CatBoost was found to be the most efficient and provided the highest accuracy.

3.2 Software designing

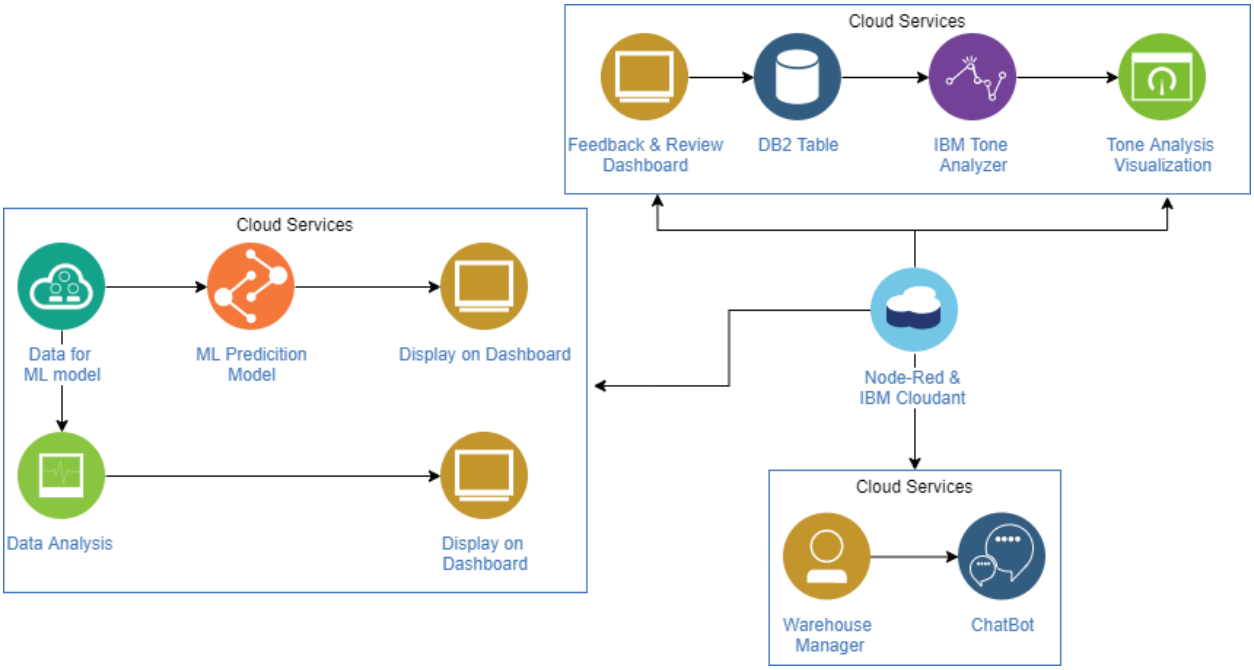

Fig. 3: Software Design

The designed software is operated using IBM Cloud Services like, IBM Watson Studio for Machine Learning Model, Node-Red for the UI, DB2 instance for the database connectivity, IBM Tone Analyzer for the tone analysis of the feedback.

4. EXPERIMENTAL INVESTIGATIONS

While implementing the project various IBM services were investigated and studied for their use. Also while implementing the machine learning model, the data available from the kaggle dataset was to be investigated and studied properly to make it suitable for prediction.

Investigation of Data:

Remove Outliers :

The first step after downloading the data set is to pre-process it before one can pass it to a model for training. To clean this data set all the null values had to be removed so that the error in result can be avoided.

Another challenge is to make the data set numeric as that's what the computers can analyze and compute. Columns such as center_type,category and cuisine were categorical and were converted to numeric using a LabelEncoder.

```
lbe_center_type = LabelEncoder()
train['center_type'] = lbe_center_type.fit_transform(train['center_type'])

train.head()
```

week	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders	category	cuisine	city_code	region_code	center_type	op_area
1	136.83	152.29	0	0	177	0	3	647	56	2	2.0
2	135.83	152.29	0	0	323	0	3	647	56	2	2.0
3	132.92	133.92	0	0	96	0	3	647	56	2	2.0
4	135.86	134.86	0	0	163	0	3	647	56	2	2.0
5	146.50	147.50	0	0	215	0	3	647	56	2	2.0

Fig. 4: Converting data to numeric values

Often the dataset contains a few anomalies called Outliers. Outliers are those data points which are very far from the rest of the categories. Even this data set had a few outliers. Some of the num_orders (Number Of Orders column) had over 20000 values while the rest were less than 12000. Similarly the price for a few items were above 850 whereas as many of them were less than800.

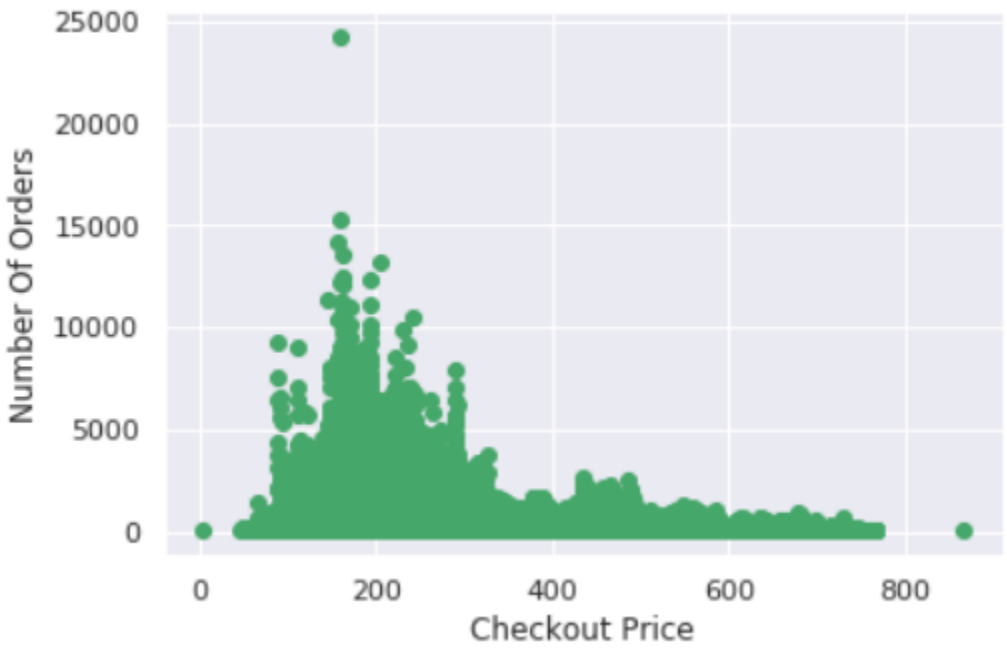

Fig. 5: Outliers

Even a correlation matrix was plotted to check the correlation between the eight highest features.

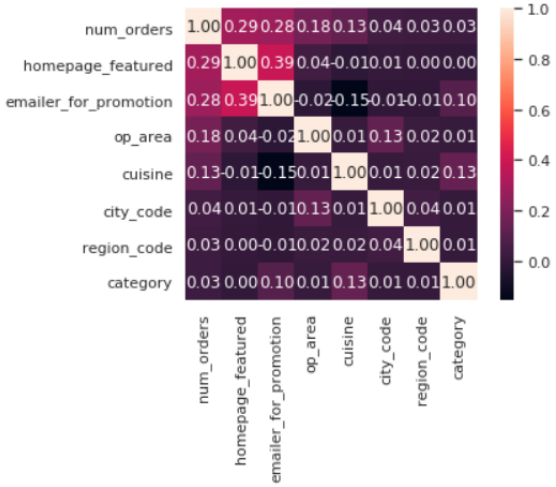

Fig. 6: Correlation Matrix

Investigation of IBM Services:

Various IBM Cloud Services were required to be used while implementing the project. The IBM services studied were:

- IBM DB2 Instance
- IBM Node-Red Editor

- Watson Studio
- ToneAnalyzer
- ChatBot

Apart from these IBM services various other concepts were needed to be understood for further completion of the project. The other fields explored were:

- ZOHO Writer
- Github
- Project Management Tools
- dash, Plotly, Flask
- Google Colab

Dash, Plotly and Flask were explored and investigated for trying to implement dynamic charts using chart.js library and integrating ML model with UI

Google Colab was investigated to implement Catboost Model for prediction.

5. FLOWCHART

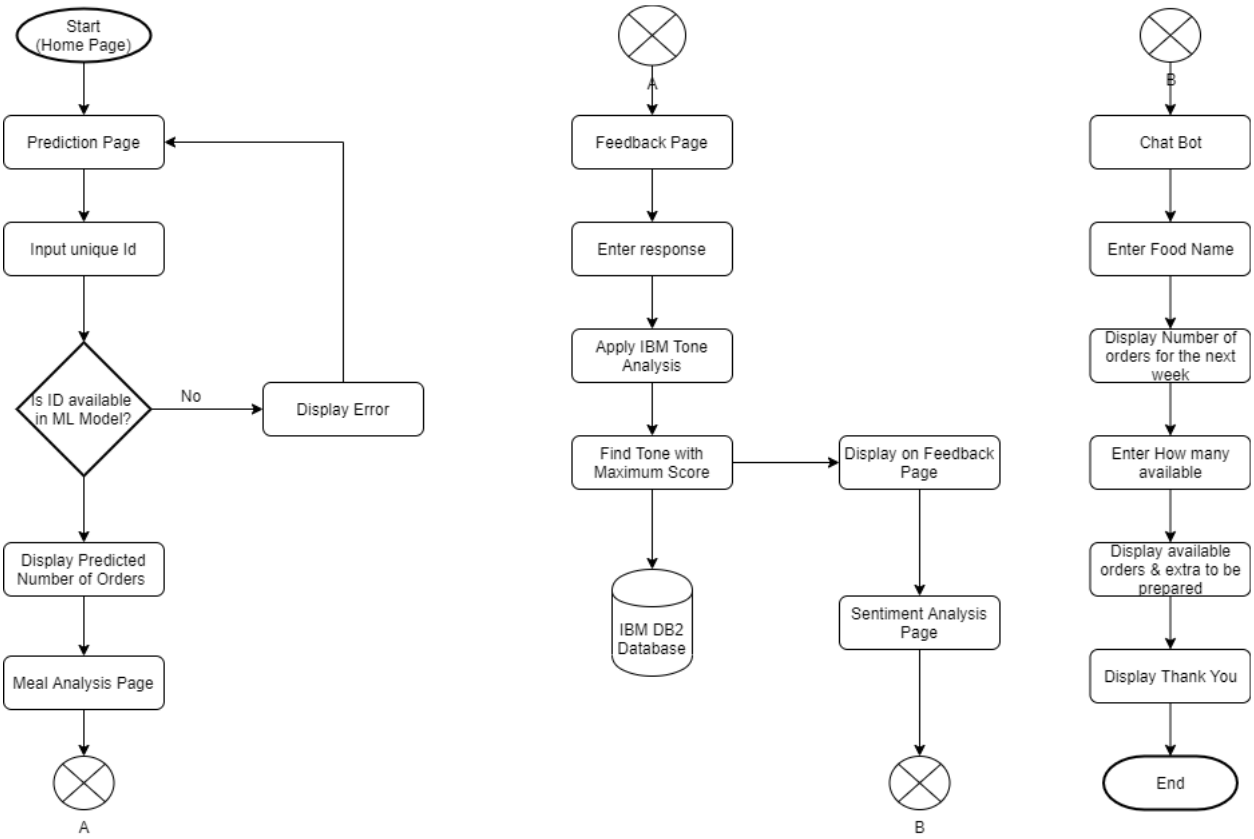

Fig. 7: Flow Chart

6. RESULT

ML Model Prediction:

The various metrics used to evaluate the results of the prediction are :

1.Mean Squared Error(MSE)

2.Root-Mean-Squared-Error(RMSE).

3.Mean-Absolute-Error(MAE).

Mean Squared Error: MSE or Mean Squared Error is one of the most preferred metrics for regression tasks. It is simply the average of the squared difference between the target value and the value predicted by the regression model. As it squares the differences, it penalizes even a small error which leads to over-estimation of how bad the model is. It is preferred more than other metrics because it is differentiable and hence can be optimized better.

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)}_{\text{The square of the difference between actual and predicted}}^{2}$$

.....Equation 1

Root Mean Squared Error: RMSE is the most widely used metric for regression tasks and is the square root of the averaged squared difference between the target value and the value predicted by the model. It is preferred more in some cases because the errors are first squared before averaging which poses a high penalty on large errors. This implies that RMSE is useful when large errors are undesired.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

.....Equation 2

Mean Absolute Error: MAE is the absolute difference between the target value and the value predicted by the model. The MAE is more robust to outliers and does not penalize the errors as extremely as mse. MAE is a linear score which means all the individual differences are weighted equally. It is not suitable for applications where you want to pay more attention to the outliers.

$$MAE = \underbrace{\frac{1}{n}}_{\text{Divide by the total number of data points}}\underbrace{\Sigma}_{\text{Sum of}}\left|\underbrace{y}_{\text{Actual output value}} - \underbrace{\hat{y}}_{\text{Predicted output value}}\right|$$

The absolute value of the residual

.....Equation 3

```
[ ] from sklearn import metrics
    print('Mean Absolute Error:', metrics.mean_absolute_error(y_valid, y_pred))
    print('Mean Squared Error:', metrics.mean_squared_error(y_valid, y_pred))
    print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_valid, y_pred)))
```

```
Mean Absolute Error: 141.06354668228207
Mean Squared Error: 73071.1911701629
Root Mean Squared Error: 270.3168347886659
```

Fig. 8: Prediction Accuracy

The predicted output was stored in a .csv file along with the other parameters which were considered for testing.

```
[ ] pred = model.predict(test_data[features])
    pred = (np.exp(pred) - 1)
    submission = pd.DataFrame({'id':test['id'], 'week':test['week'], 'center_id':test['center_id'], 'meal_id':test['meal_id'], 'checkout_price':test['che
    submission = submission[['id','week', 'center_id', 'meal_id', 'checkout_price', 'base_price', 'emailer_for_promotion', 'homepage_featured', 'num_orde
    submission.head()
```

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders
0	1028232	146	55	1885	158.11	159.11	0	0	1043.790425
1	1262649	147	55	1885	159.11	159.11	0	0	1064.439018
2	1453211	149	55	1885	157.14	158.14	0	0	1040.641858
3	1262599	150	55	1885	159.14	157.14	0	0	896.750876
4	1495848	151	55	1885	160.11	159.11	0	0	1052.958971

```
[ ] submission.to_csv('catboost_1.csv', index=False)
```

Fig. 9: Conversion to CSV File

User Interface:

The user interface is distributed into six pages:

1. Home Page
2. Prediction
3. Meal Analysis
4. FeedBack (of Warehouse Manager)
5. Sentiment Analysis (of user reviews)
6. Chat Bot

1. Home Page:

The home page displays the dynamic chart which is plotted from the csv file of the training dataset. It allows the user to select the parameters which he wants and view the relation between them graphically.

The second part of the home page includes the table which displays the amount spent in

week 146 with respect to the order of meal and the number of orders.

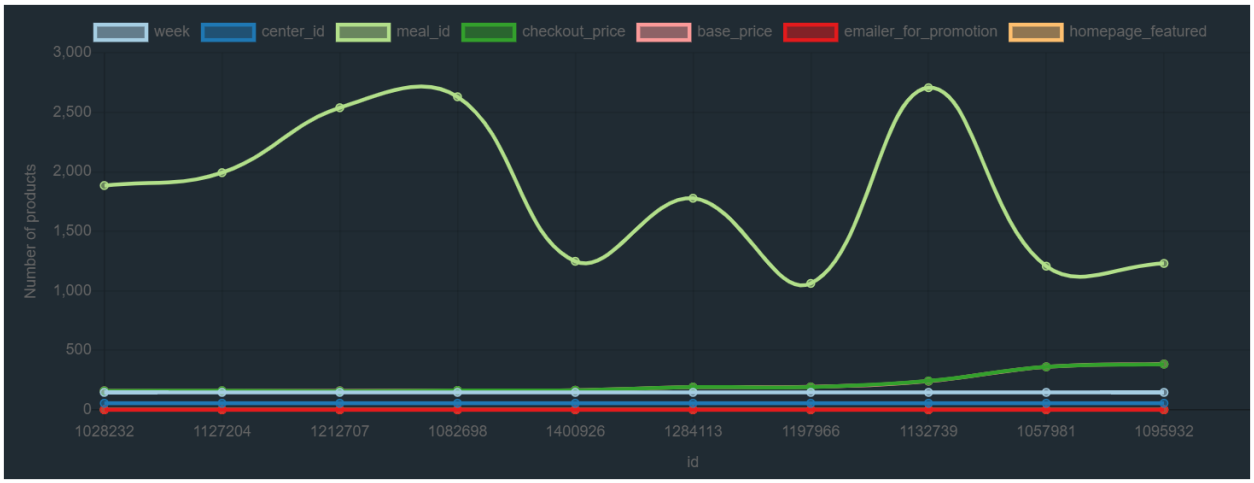

Fig. 10(a): Dynamic Chart

WEEK 146 SALES	QUANTITY	TOTAL
Thai Cuisine Full Meal	20	$1,342
Indian Cuisine Full Meal	18	$1,550
Italian Cuisine Full Meal	15	$4,170
Beverages and Sandwich	25	$2,974

Fig. 10(b): Week 146 Sales Table

2. Prediction:

The prediction page is connected to the csv file which was generated from the ML model. This page displays the predicted number of orders and the other parameters related to the id given as input to the form. Figure 11 displays the prediction page.

3. Meal Analysis:

Along with the prediction it is important for the warehouse manager to compare the amount of raw material available in the warehouse and the number of orders which can be made from the existing material. On the basis of the existing number of orders and the predicted number of orders, the manager is supposed to order the raw material to satisfy the extra needs only. Thus the comparison of the existing material and the expected number of orders is displayed on the meal analysis page. Figure 12 displays the Meal Analysis page.

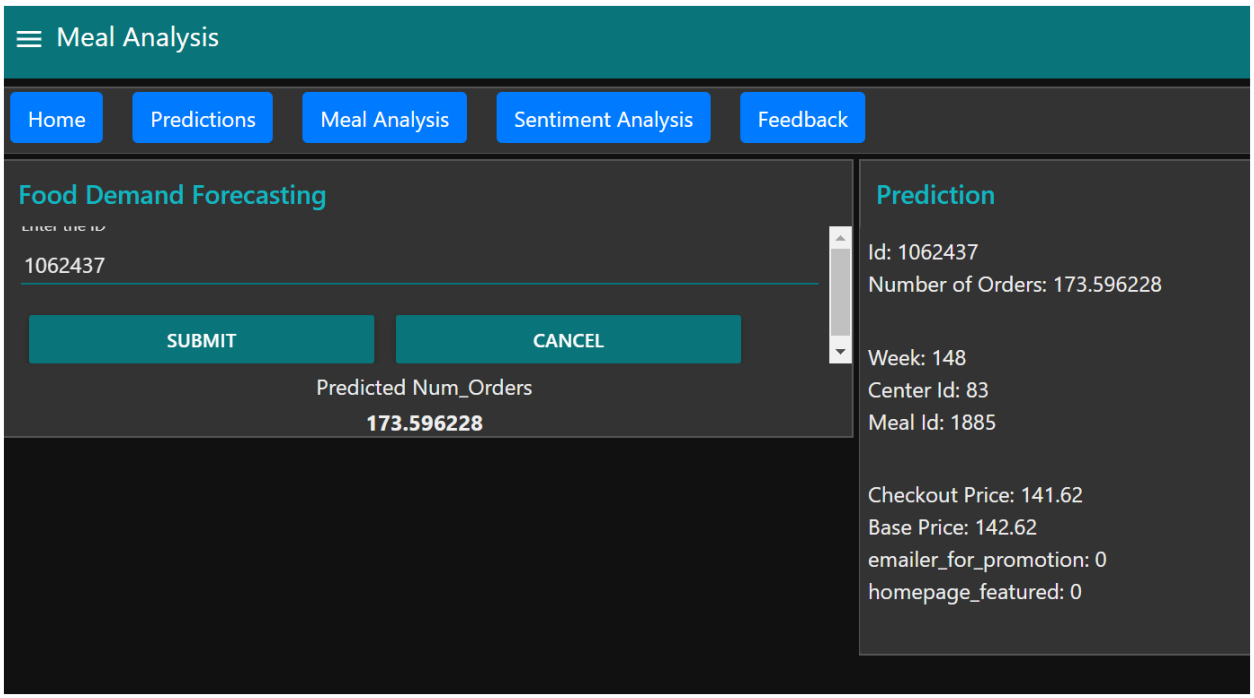

Fig. 11: Prediction Page

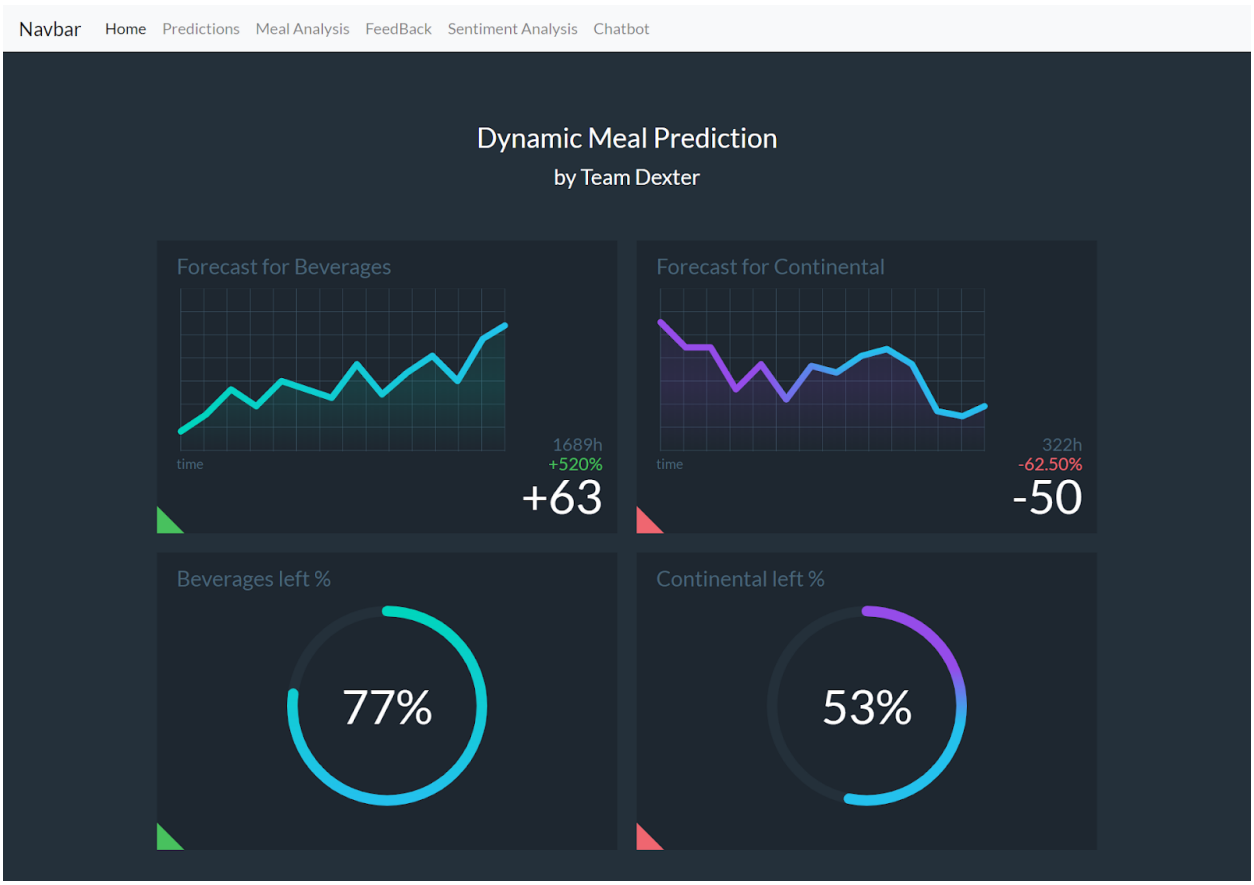

Fig. 12: Meal Analysis Page

4. FeedBack (of Warehouse Manager):

In order to understand the satisfaction of the warehouse manager with the predicted output and the system, for improving the prediction model and system, obtaining feedback is a crucial task. The feedback form obtains the feedback of the manager and applies IBM Tone Analyzer on it to understand the tone of the feedback. The Output of the tone analyzer which gives a series of array objects and the scores of the tone is then passed through a function to find out the maximum score of the types of tones and the tone with the maximum score is stored in the IBM DB2 Schema in the Feedback table along with the information of the Manager. The figure 13 displays the Feedback form and the personalized result of the tone analyzer.

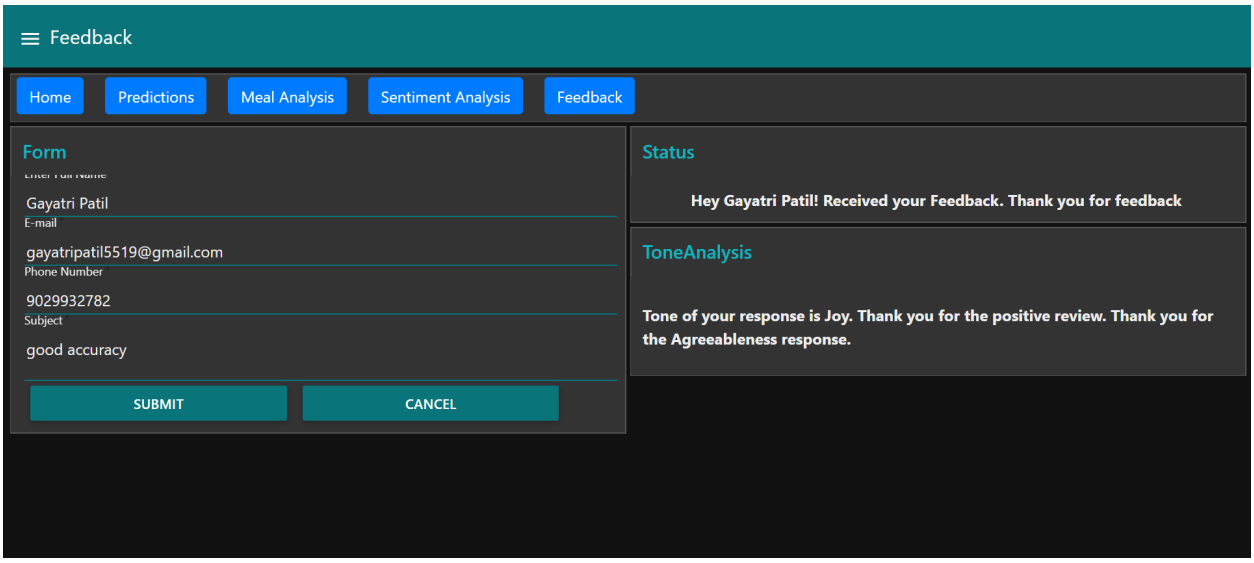

Fig. 13: Feedback Page

5. Sentiment Analysis (for User Reviews):

As the feedback of the manager is important for improving the accuracy and the system, feedback of the users to whom the warehouse supplies orders is also important for the maintenance of the warehouse. The sentiment Analysis page displays the output of the opinions of the users on the three different cuisines served by the warehouse and it's analysis. Figure 14 displays the sentiment analysis page.

6. Chat Bot:

Chat bot is an additional feature added in the system for improved user experience and convenience. The chat bot displays the extra number of orders to be prepared to satisfy the predicted number of orders, taking into consideration the cuisines given as input. Figure 15 displays the output of the chat bot.

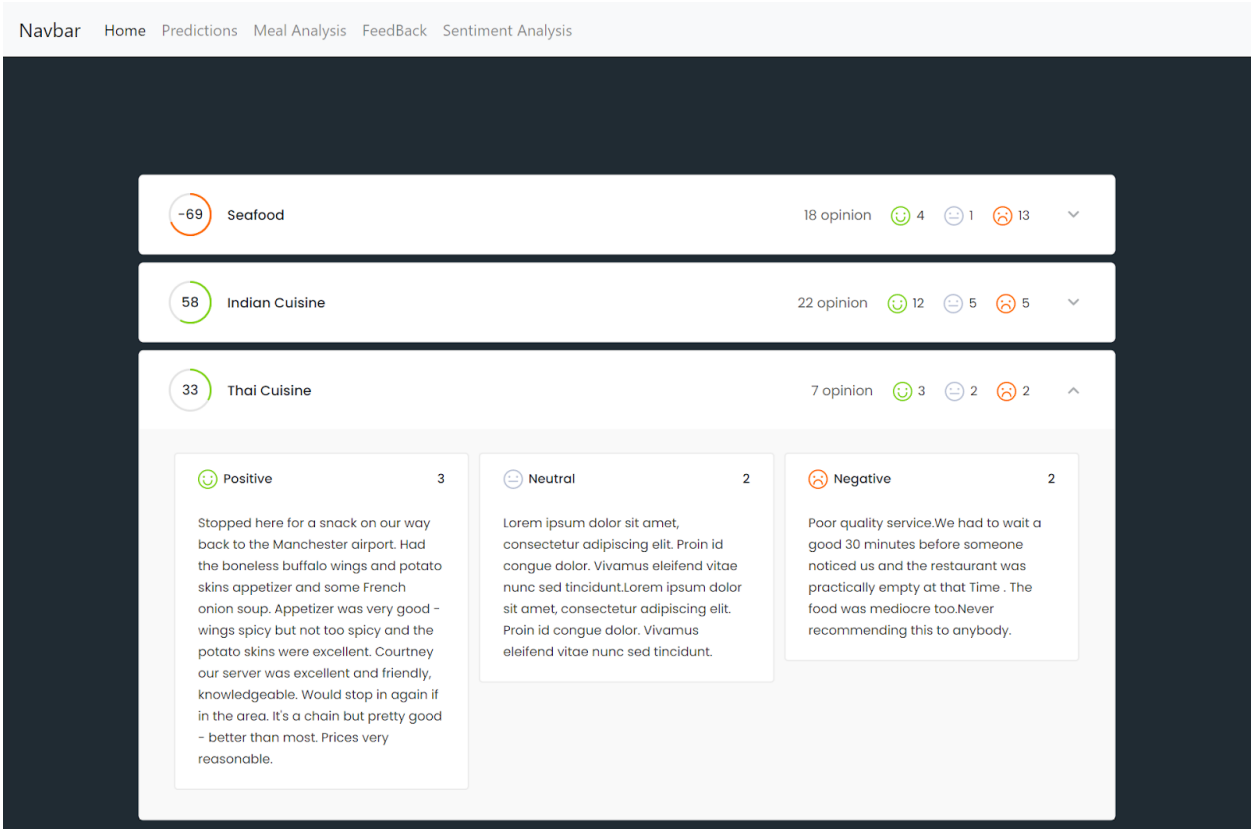

Fig. 14: Sentiment Analysis Page

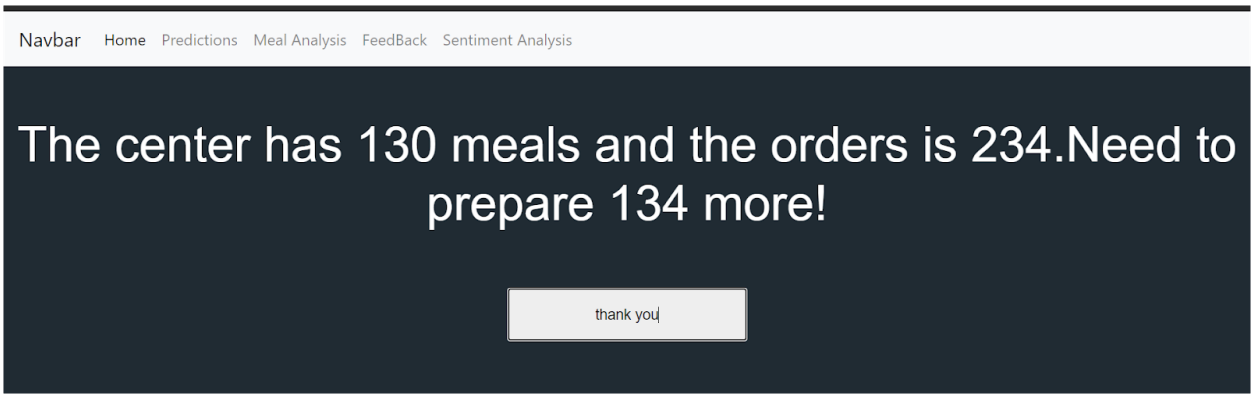

Fig. 15: Chat Bot

7. ADVANTAGES & DISADVANTAGES

Advantages:

1. Less wastage of food and quickly predict orders per week
2. Flexible, can be used by any food company.
3. Accurate predictions for Number of Orders
4. Easy visualization for any naive person.
5. Tone analysis for further improvement in the model according to the feedback

Disadvantages:

1. Can not create & scan QR codes according to items for easy functioning
2. Data available is per week, thus does not give day to day analysis which would be useful for increased saving of food.

8. APPLICATIONS

1. Warehouse Management:
    a. The prediction of expected number of orders of specific food types for upcoming weeks, and as a result the amount of required raw material can be analyzed, making it useful in warehouse management.
2. The System can be used to keep track of reviews of the users and can be used in the up gradation and improvement of the existing warehouses.

9. CONCLUSION

Demand forecasting is a key component to every growing online business. Without proper demand forecasting processes in place, it can be nearly impossible to have the right amount of stock on hand at any given time. A food delivery service has to deal with a lot of perishable raw materials which makes it all the more important for such a company 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. Thus with the help of prediction of the expected number of orders of specific meal types to be delivered from the

warehouse will be helpful in proper management of the inventory and avoid wastage of food. The implementation of sentiment analysis will help in improving the system and the warehouse as well.

10. FUTURE SCOPE

1. The system can be extended to implement prediction of everyday number of orders.
2. The project can be extended for food transportation and tracking of deliveries.

11. BIBILOGRAPHY

Team Name: Dexter
Team Member Names:

1. Neelam Somai (Team Leader)
2. Gayatri Patil
3. Gaurav Tirodkar
4. Yash Mate

College: Vivekanand Education Society's Institute of Technology, Mumbai-74

Project Title: Optimized Warehouse Management of Perishable Goods for a Food Delivery Company

References:

- Research and Scope
    - Research for existing projects
        - https://www.softwareadvice.com/scm/warehouse-management-system-comparison/
        - https://www.selecthub.com/warehouse-management-software/
    - Define Technology Stack
        - https://www.ibm.com/in-en/cloud
- Design UI and UX
    - Design Prototype
        - https://www.interaction-design.org/literature/article/design-thinking-get-st

arted-with-prototyping#:~:text=A%20prototype%20is%20a%20simple,or%20possible%20changes%20in%20direction.

        - https://www.toptal.com/designers/prototyping/guide-to-prototype-design
    - Create NodeRed Project
        - https://www.youtube.com/watch?v=WUB6gySRjQg
- Download Dataset and Preprocess it
    - Download Dataset
        - https://www.kaggle.com/kannanaikkal/food-demand-forecasting
    - Remove outliers
        - https://www.youtube.com/watch?v=2Qrost474lQ
        - https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba
- Try different ML models and choose the best one
    - Feature Selection
        - https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e
        - https://www.datacamp.com/community/tutorials/feature-selection-python
    - Model Implementation using different algorithms
        - https://cognitiveclass.ai/courses/machine-learning-with-python
        - https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/autoai-overview.html
    - Selection of the best model
        - https://towardsdatascience.com/a-short-introduction-to-model-selection-bb1bb9c73376
- Prediction
    - Test the model for various cases
        - https://developers.google.com/machine-learning/testing-debugging/pipeline/deploying
    - Predict the final output
        - https://machinelearningmastery.com/how-to-connect-model-input-data-with-predictions-for-machine-learning/
- Dashboard development

    - Create the UI according to the prototype
        - https://www.youtube.com/watch?v=WUB6gySRjQg
    - Connect the model with UI
        - https://youtu.be/O5wqjk_GeJo
- Deployment and Documentation
    - Deploy the AI model on IBM cloud
        - https://youtu.be/O5wqjk_GeJo
        - https://developer.ibm.com/depmodels/cloud/tutorials/deploy-your-first-app-to-ibm-cloud/
    - Finalize the documentation and Report
        - https://www.youtube.com/embed/wn-1yI9t4WQ
    - Upload the final code on GitHub
        - https://guides.github.com/
        - https://guides.github.com/activities/hello-world/

APPENDIX

Link to Node-Red WorkSpace:
Gayatri Patil: https://node-red-aftnm.eu-gb.mybluemix.net/red/#flow/2596ff4d.ba01b
Gaurav Tirodkar: https://node-red-gaurav.eu-gb.mybluemix.net/red/#flow/3f83af3c.5ee94

Link to Project UI: https://node-red-aftnm.eu-gb.mybluemix.net/uibuilder/home.html

Source Code:

ML Prediction Model Code:

```
#Merging the datasets into single train and test files
train = pd.merge(train,fulfilment_center, on='center_id')
test = pd.merge(test,fulfilment_center, on='center_id')
train = pd.merge(train,meal_info, on='meal_id')
test = pd.merge(test,meal_info, on='meal_id')

#finding outliers and deleting them
outlier_index = train[(train['num_orders']>15000)].index
train.drop(outlier_index,inplace = True)

#creating catboost model
model = CatBoostRegressor(
    iterations=2000,
    learning_rate=0.02,
    max_depth=8,
    l2_leaf_reg=10,
    loss_function='RMSE',
    random_seed=2019,
    od_type='Iter',
    od_wait=25,
    verbose=100,
    use_best_model=True
    )

#training the model
errcb=[]
y_pred_test=[]
```

```
fold = KFold(n_splits=5,shuffle=True,random_state=2019)

for train_index, test_index in fold.split(train_data[features],train_data['num_orders']):
    X_train, X_valid =train_data[features].iloc[train_index], train_data[features].iloc[test_index]
    y_train, y_valid = train_data['num_orders'][train_index],train_data['num_orders'][test_index]
    model.fit(X_train,y_train,
              cat_features = categorical_features_indices,
              eval_set=(X_valid,y_valid),
              early_stopping_rounds=300,
              verbose=100)
    preds = model.predict(X_valid)
    print("err: ",np.sqrt(mean_squared_error(y_valid,preds)))
    errcb.append(np.sqrt(mean_squared_error(y_valid,preds)))
    p = model.predict(test_data[features])
    y_pred_test.append(p)

#testing the model
model.fit(X=train_data[features], y=train_data['num_orders'],
cat_features=categorical_features_indices, eval_set=(train_data[features],
train_data['num_orders']), verbose=100)

#converting the predictions into CSV files
pred = model.predict(test_data[features])
pred = (np.exp(pred) - 1)
submission = pd.DataFrame({'id':test['id'], 'week':test['week'], 'center_id':test['center_id'],
'meal_id':test['meal_id'], 'checkout_price':test['checkout_price'], 'base_price':test['base_price'],
'emailer_for_promotion':test['emailer_for_promotion'],
'homepage_featured':test['homepage_featured'], 'num_orders':pred})
submission = submission[['id','week', 'center_id', 'meal_id', 'checkout_price', 'base_price',
'emailer_for_promotion', 'homepage_featured', 'num_orders']]
submission.head()
submission.to_csv('catboost_1.csv', index=False)
```