FOOD DEMAND FORECASTING – SUPPLY CHAIN ANALYTICS

Nitanshu Joshi, Aditi Soni, Melissa Rochelle Mathias

Luddy of Informatics, Computing, and Engineering
Indiana University Bloomington, Indiana

ABSTRACT

The field of supply chain management is concerned with the management of the transportation of commodities and services from raw materials to finished goods. The Food Industry is one of the most important areas in which Supply Chain Management plays a significant role. The organization manages the flow of raw resources in order to maximize customer value and acquire a competitive advantage over its competitors. To ensure that the proper amount of product is conceived and produced, food and beverage businesses have always had to balance the insights of sales and marketing teams with production and supply chain plans. In the lack of sufficient planning, production schedules are disturbed, transfers and shipments must be hastened or slowed, customer service is inconsistent, and stock changes are unforeseeable.

Index Terms— Machine Learning, Statistical Analysis, Supply Chain Management, Demand Forecasting

1. INTRODUCTION

The data is for a food delivery business with many locations; these cities have a variety of fulfillment centers where customers can pick up their meal orders. A warehouse with too much inventory risks wastage, while a warehouse with too little inventory risks out-of-stocks, forcing customers to seek solutions from their competitors. Our goal is to estimate demand in these centers for the following weeks so that raw material inventory can be planned properly. The majority of raw materials are replenished on a weekly basis, and because the raw materials are perishable, procurement planning is critical. Second, precise demand estimates are quite beneficial when it comes to center staffing.

Despite there being a predictable seasonal pattern, the characteristics that forecast these seasons are not readily visible. As a result, it's difficult to estimate how orders will alter as the season's change. We are investigating how to estimate demand in order to tackle such challenges. We are looking at methods for forecasting food demand based on internal data like the number of orders. This paper describes the approach for forecasting methods using statistical analytics and the implementation of machine learning models.

The main Research Question that we aim to answer through our project is:

'What are top features that would impact standardized demand forecasting for a meal delivery kit company, facilitating their supply chain?'

There are also other sub-questions that we will work on, some of these are:

- How did the choice of encoding Categorical variables impact the performance of the models? (Choice between Label and One-Hot Encoding)
- How did hyperparameter tuning impact the performance of various regression models?
- Which regression model performed best for predicting the demand of orders for various Restaurants?

2. DEMAND FORECASTING RELATED WORK

Demand forecasting is a crucial activity since it can indicate market trends and aid in the company's strategic planning. According to our data, it is a critical tool for making decisions faster and safer. Analysts can use a variety of strategies to help them forecast demand. Although these techniques differ significantly, they share some common characteristics: generally assume that the factors that have influenced demand in the past will continue to influence demand in the future.

The proposed approach to predict demand forecasting without market research taken by Institute number 8 “Information Technology and Applied Mathematics”, Moscow Aviation Institute (National Research University), 4 Volokolamskoe Hhighway, 125993, Moscow, Russia 2 Department “Problems of mathematical modeling and high-performance computing”,Keldysh Institute of Applied Mathematics (Russian Academy of Sciences), 4 Miusskaya Square, 125047, Moscow, Russia, wherein they used implementations of gradient boosting machine learning algorithms like XGBoost, Light GBM and CatBoost. The dataset worked upon was from the Ozon Online store. Input parameters being price, name, category and description text, giving them highest accuracy in LightGBR model. As this proposed system can be amalgamated into complex systems as well, could help analysts for strategizing sales.

Another work proposed by S. M. Taslim Uddin Raju,[1] Amlan Sarker,[2] Apurba Das,[3] Md. Milon Islam,[1] **Mabrook S. Al-Rakhami**,[4] Atif M. Al-Amri,[4,5] Tasniah Mohiuddin,[6] and Fahad R. Albogamy[7] talks about steel demand prediction and _uses Random Forest Regressor and various other ensemble models like GBR, XGBR Boosting and STACK(ensemble of single reference models like ELM+GBR+XGBR-SVR). The dataset is obtained from a steel industry in Bangladesh, after performing PCA, used a grid search algorithm to find the optimal hyper-parameters, with an aim to reduce RMSE. STACK model giving them the highest accuracy of 97%, helping them predict demand in the steel industry almost a month prior. However in our approach we did even use the CatBoost ensemble method with Randomized Search CV, which gave us the highest R-Squared.

3. DATA UNDERSTANDING

As the aim is to predict demand for the different products of   the restaurant we would be considering the number of orders for the different meals as our target variable. While all the other factors in the data would act as predictor variables, these would include checkout price, base price, type of cuisine, location specific details etc.

Variable
id
week
center_id
meal_id
checkout_price
base_price
emailer_for_promotion
homepage_featured
num_orders

Variable
meal_id
category
cuisine

Variable
center_id
city_code
region_code
center_type
op_area

```
Data columns (total 15 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   week                   456548 non-null  int64
 1   center_id              456548 non-null  int64
 2   meal_id                456548 non-null  int64
 3   checkout_price         456548 non-null  float64
 4   base_price             456548 non-null  float64
 5   emailer_for_promotion  456548 non-null  object
 6   homepage_featured      456548 non-null  object
 7   num_orders             456548 non-null  int64
 8   category               456548 non-null  object
 9   cuisine                456548 non-null  object
 10  city_code              456548 non-null  int64
 11  region_code            456548 non-null  int64
 12  center_type            456548 non-null  object
 13  op_area                456548 non-null  float64
 14  weeks_of_20            456548 non-null  object
dtypes: float64(3), int64(6), object(6)
```

Fig 1. Data Description

Initially we had three csv files with 456548 rows, shown as in Fig above Fig 1, which contained train_set.csv, meal_info.csv and meal_fulfillment.csv. However we merged the entire dataset into a single one, the final dataset having a total of 15 columns, whose detailed description is provided in Fig 1 above and in the later stage performed Feature Engineering on the final merged dataset to derive new attributes. Each row in the final merged dataset represents a center with a unique meal ID. The columns provide information regarding the region, city, category, cuisine, price, promotional index and week number of the delivery.

4. EXPLORATORY DATA ANALYSIS

KEY QUESTIONS:

- What is the week-wise demand variation?

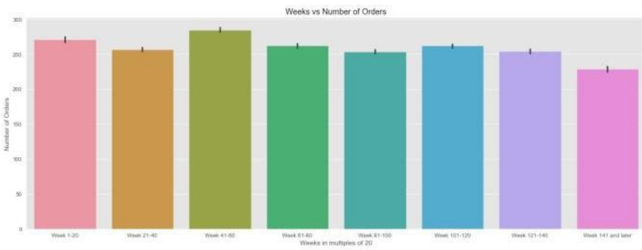

Fig 2. – 20 Week Demand Chart

Here we have accumulated the number of weeks in bins of 20 weeks each as we had data worth of 140 weeks, to show the average number of orders which were received over the weeks, the chart interprets that there is almost a constant flow of orders from the time data has been collected, which not much of crests and troughs.

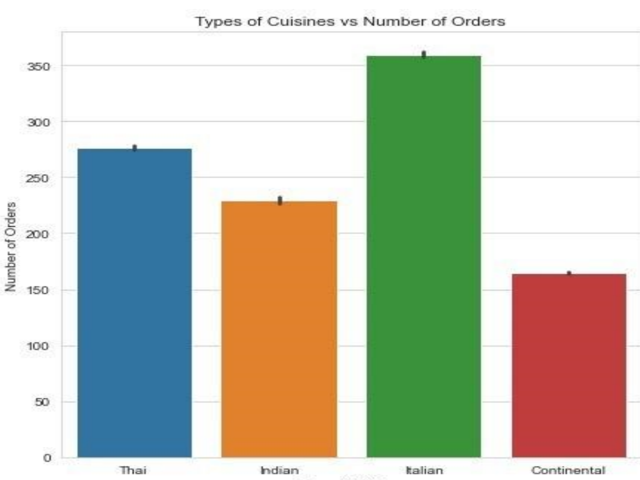

Fig 3. – Cuisine and Demand

- What is the effect of cuisines on the demand?

It can be seen that "Italian" cuisines have the most demand among customers whereas 'continental' has the least. By the above presented chart, we can interpret that Italian cuisine emerged as the most popular amongst customers, with average number of orders coming around 370, both Thai and Indian food were not too far behind, with, respectively, 275 and 225 number of orders and Continental being the least favorite amongst customers, which rakes in around only 170 orders.

- What is the effect of prices on the demand – both in terms of a base price and checkout price

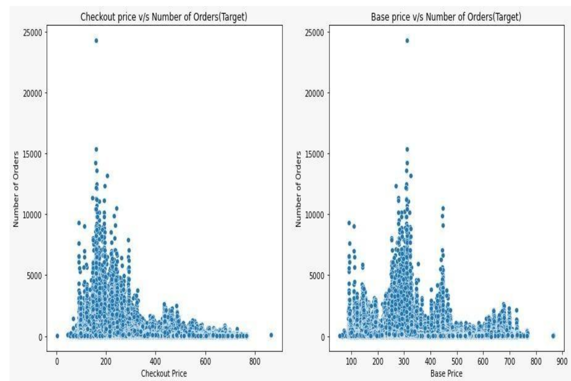

Fig 4. – Demand Price analysis

Through these scatter plots, we highlight the average number of orders that go out with the base prices and also with respect to prices that come after a discount has been applied to them. Observations are as below: the number of meals that go out with the discounted prices is around \$180-\$190, while most of the meals that go out with the base prices are in the range of \$290-\$300.

- What is the demand for different promotions?

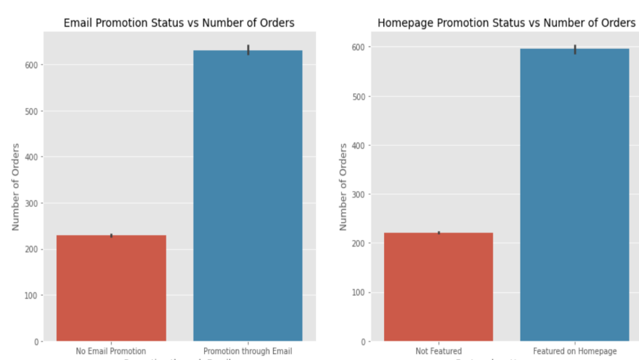

Fig 5 – Promotional Activity and Demand

It can be observed that the more featured and promoted the restaurants are, the more the number of orders. These above bar- plots showcase the impact of the number of orders with and without promotional status either on emails or homepages. It can be clearly projected that in both cases, the number of orders with a promotion status exhibits a higher numerical count.

5. FEATURE ENGINEERING

After importing the dataset, we performed data cleaning steps on it to transform raw data into a more understandable, useful format to extract accurate results and build a useful model on it. The first steps in this was to identify missing values in the data and outlier detection.

Refactoring some of the old features to a new format so that more information can be derived and they are easier to work with, such as promotions, homepage and putting the weeks into separate bins.

Deriving additional details from our current predictor variables:

1) Percentage discounts that customers got for meals in particular centers.
2) Profit earned on meal items by the different centers
3) Deriving year, month, quarter, year in month in order to showcase temporal trends.
4) Compare W eek Price - Changes between checkout prices on a weekly basis and relative percentages.

- Correlation Analysis

A statistical method for determining the relationship between two numerical variables is correlation analysis. It shows how the features correspond to the output from the point of view of machine learning. However, determining how characteristics are related is difficult. Data visualization can aid in determining how specific characteristics may influence the outcome. According to the definition, a value in the range of +1 to 1 implies that there is no correlation at all, +1 indicates that there is a perfect positive correlation, and 1 indicates that there is a perfect negative correlation.

As per the Fig 6, base price with checkout price are (+ve) strongly correlated with each other (+0.95). There is also a moderate correlation between the discounts offered and the base price(+0.31). If we were to check in context with the target variable which indicated demand no_of_orders, there is a high (+0.18) correlation with op_id whereas the most (-ve) correlation is observed with the attribute checkout price (-0.28).

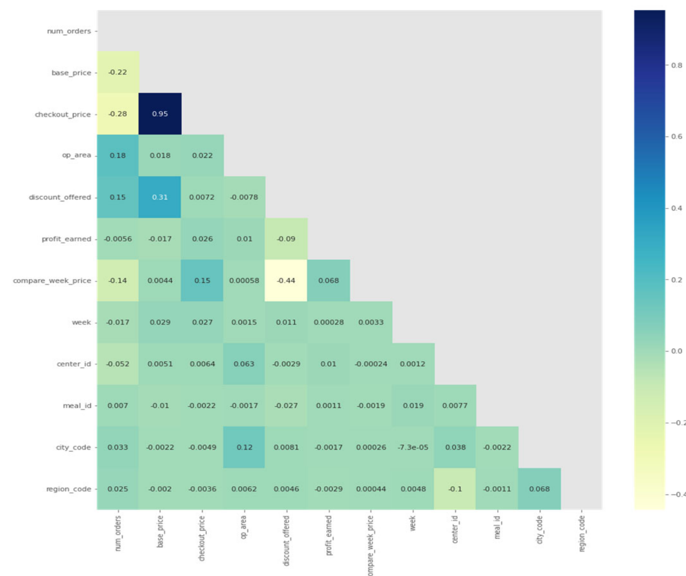

Fig 6 – Heat Map: Correlation amongst Predictor and Target Variables

6. MACHINE LEARNING MODELS

6.1. Linear Regression

Our first proposed model is a Linear Regression. It attempts to model a relationship between two variables by fitting a linear equation to observed data.

Linear regression line has an equation of the form:

$y = a + bx + \text{Error},$

where x is a predictor variable and y is the target variable that has to be predicted. The slope of the line is b, and a is the intercept (the value of y when x = 0).

For this project, we used Multiple Linear Regression, where there are multiple predictors (independent variables). It can be mathematical represented as:

y = a + b1x1 + b2x2 + b3x3 +... bnxn + Error

Here b1, b2, b3 ,.., bn are coefficients corresponding to the various predictor variables x1, x2, x3, …., xn.

Both simple linear regression and multi-linear regression have residual error terms related to it.

The basic model that we used did not have any regularization. We made the use of K-Fold Cross Validation techniques while running the Simple Regression Model.

Apart from the basic mode, we also made use of regularization techniques in order to enhance the performance of the linear regression model. Regularization is a type of regression in which the coefficient estimates are regularized i.e constrained towards zero. In other words, to prevent the danger of overfitting, this strategy inhibits learning a more flexible model.

A simple relation for linear regression looks like this. Here Y represents the learned relation and 'β' represents the coefficient estimates for different variables or predictors(X).

The fitting procedure involves a loss function, known as residual sum of squares or RSS. The coefficients are chosen, such that they minimize this loss function.

Now, based on training data, the coefficients will be adjusted. The computed coefficients will not generalize well to new data if the training data contains noise. This is where regularization is used to regularize these learnt estimations towards zero.

For our project we used two different regularization techniques:

1) Ridge Regularization (Regression)
2) Lasso Regularization (Regression)

6.1.1 Ridge Regression

This is denoted by the following mathematical equation.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

Here the RSS is modified by adding the shrinkage quantity**.** Now, the coefficients are estimated by minimizing this function. Here, λ is the tuning parameter that decides how much we want to penalize the flexibility of our model. The increase in flexibility of a model is represented by an increase in its coefficients, and if we want to minimize the above function, then these coefficients need to be small. This is how the Ridge regression technique prevents coefficients from rising too high.

6.1.2 Lasso Regression

This is denoted by the following mathematical equation.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

Lasso is another variation, in which the above function is minimized. It is clear that this variation differs from ridge regression only in penalizing the high coefficients. It uses |βj|(modulus)instead of squares of β, as its penalty. In statistics, this is known as the L1 norm.

6.2. Random Forest Regressor

Our second proposed model is Random Forest Regressor, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest gives us a class prediction and the class with the majority votes becomes our model's prediction.

Random forest builds entire decision trees in parallel using random bootstrap samples of the data set using a process known as bagging. The final forecast is a weighted average of all decision tree predictions.

The basic idea behind this is to combine multiple decision trees in determining the final output instead of relying on individual decision trees. First we pass the features(X) and the dependent(y) variable values of the data set, to the method created for the random forest regression model. We then use the grid search cross validation method to determine the optimal values to be used for the hyperparameters of our model from a specified range of values.

6.3. Decision Tree

Our third proposed model is Decision tree. It builds regression or classification models in the form of a tree structure. It gradually cuts down a dataset into smaller and smaller sections while also developing an associated decision tree. The end result is a tree containing leaf nodes and decision nodes.

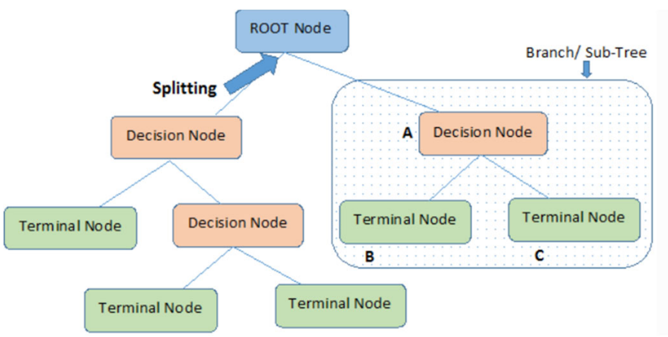

Figure 7 - Decision Tree

(Source:https://gdcoder.com/decision-tree-regressor-explained-in-depth/)

While training the model, it is fitted with historical data relevant to the problem domain and the true value we want the model to learn to predict. The model learns relationships between the data and the target variable.

After the training phase, the decision tree produces a tree similar to the one shown above, calculating the best questions as well as their order to ask in order to make the most accurate estimates possible. When we want to make a prediction the same data format should be provided to the model in order to make a prediction. The prediction will be an estimate based on the data that the model is trained on.

Strategic splits in decision trees heavily affects a tree's accuracy. Decision trees regression uses mean squared error -MSE to decide to split a node in two or more sub-nodes.

For example, if we consider a binary tree the algorithm first will pick a value and split the data into two subsets. For each subset, it will calculate the MSE separately. The tree chooses the value which results in the smallest MSE value.

This is the entirety of creating a decision tree regressor and will stop when some stopping condition (defined by hyperparameters) is met:

- When you hit a limit that was requested (e.g max_depth)
- When your leaf nodes only have one thing in them (no further split is possible, MSE for the train will be zero but will overfit for any other set -not a useful model)

6.4. Gradient Boosted Tree - Ensemble Learning

Boosting is a sequential ensemble strategy in which difficult to classify examples are given higher weights, causing succeeding learners to place greater focus on learning mis-classified data instances. The final model would be a weighted average of n poor students.

The GBDT (Gradient Boosting Decision Tree) model is an ensemble model comprising decision trees that are trained sequentially (i.e. an ensemble model of boosting). GBDT learns the decision tree in each iteration by fitting the residual errors (errors upto the current iteration). This means that each succeeding learner attempts to learn the difference between actual output and the weighted total of predictions up to the previous iteration. The gradient approach is used to reduce mistakes.

The most expensive operation in GBDT is training the decision tree, while the most time-consuming task is finding the optimal decision tree.

6.4.1. Light GBM:

Light GBM is a gradient boosting framework, using tree based learning algorithms, works with a higher efficiency and has faster training speeds. It is also associated with using lesser memory, although at the same time capable of handling large volumes of data, while providing more accurate results.

LightGBM performs leaf-wise (vertical) development, which results in more loss reduction and, as a result, improved accuracy while being faster. In contrast to previous boosting methods, it splits the tree leaf-wise. It selects the leaf with the greatest delta loss to grow. The leaf-wise algorithm has a lower loss after the leaf is fixed than the level-wise algorithm. Leaf-wise tree growth increases the model's complexity and, in the case of small datasets, may result in overfitting.

GOSS (Gradient Based One Side Sampling) is an unique sampling approach that uses gradients to down sample instances. As we know, instances with modest gradients are well trained (low training error), but examples with big gradients are undertrained. A basic approach to downsampling would be to eliminate cases with small gradients in favor of instances with big gradients, but this would change the data distribution. GOSS, in a nutshell, keeps instances with big gradients while randomly sampling examples with tiny gradients.

6.4.2 XG Boost

Our next model is XGBoost (Extreme Gradient Boosting), which implements a gradient boosting decision tree algorithm; this uses more accurate approximations to find the best tree model.

It is the top machine learning library for regression, classification, and ranking tasks, and it supports parallel tree boosting.To understand XGBoost, you must first understand the machine learning ideas and methods on which it is based: supervised machine learning, decision trees, ensemble learning, and gradient boosting.

Supervised machine learning use algorithms to train a model to detect patterns in a dataset with labels and features, and then employs the trained model to predict the labels on the features of a new dataset.

6.4.3 CAT Boost

Our final model is CatBoost. CatBoost builds symmetric (balanced) trees unlike XGBoost and Light GBM. CatBoost, uses the concept of ordered boosting, a permutation-driven approach to train model on a subset of data while calculating residuals on another subset, thus preventing target leakage and overfitting.

7. SUMMARY OF APPROACH AND MODEL IMPROVEMENTS USING EXPERIMENTATION

We followed a complete data analysis process for our project. We started with data reading and understanding, followed by data cleaning, feature engineering, and Exploratory Data Analysis. Then finally we proceeded with Model Building and predicting on the dataset for our target variable – “Number of Food orders (num_orders)”. The metrics we used here were the R-Squared Score and the Root Mean Squared Error (RMSE). We used 14 different combinations of models with and without hyperparameter tuning. The main models we used were – Linear Regression, Ridge Regression, Lasso Regression, Random Forest, Decision Tree, Boosting Ensembles (XGBoost, CAT Boost, Light GBM).

Model Improvement: Handling Categorical Features

For all the machine learning models that we would be implementing in our project, we plan to check outputs with both Label encoded and One Hot coded, in order to understand which classifier works better with respect to which encoded data format.

Finally after checking performance of the models with different ways of encoding the categorical data, we narrowed down to keeping some of the categorical columns encoded through label encoding while the others through One Hot Encoding. This was decided on the basis of getting data handling and achieving optimum model accuracy.

Label Encoding: 'center_id' , 'meal_id' , 'city_code' ,'region_code'

OneHot Encoding: 'category', 'cuisine', 'center_type'

Some of the other categorical features were changed by creating numerical bins.

Model Improvement: Hyperparameter Tuning Techniques

For improving the performance of our models we used the following hyperparameter tuning techniques.

Why hyperparameter tuning?

We know that every model has a specific set of parameters with a varying set of values. There are infinite combinations of parameters that a model can take to predict a particular set of data points. Selecting these manually is a big issue and can be time-consuming. This is where the concept of hyperparameter tuning comes in. A hyperparameter tuning is performed usually with a cross-validation technique to determine the hyperparameter value set which provides the best results for a particular metric. Metrics differ for Regression and Classification.

Grid Search CV:

This is one of the ways to tune the hyperparameters. It is the Scikit Learn brute-force-based hyperparameter tuning technique. Here we specify a specific set of hyperparameter values for a model and GridSearch CV tries all possible combinations of the hyperparameters and selects the best model/predictor with the best set of parameters. This is a time-consuming technique that increases the model running time significantly.

For our project, we used this only for the Decision Tree Regressor Model, since other models were taking multiple hours for the same. Randomized Search CV was a much better approach for the remaining model since it decreases the running time of the model significantly.

Randomized Search CV:

Like Grid Search CV, this is also used as a way to tune the hyperparameters. Here too we specify a specific set of hyperparameter values for a model. The only difference from Grid Search CV is that this technique relies on the number of iterations of running this function and it randomly selects a set of hyperparameters every iteration. This produces better results than Grid Search CV and is a lot faster than Grid Search CV.

Here for every model except the linear regression, we made the use of Randomized Search CV for hyperparameter tuning. It was seen that the model performance increased significantly for most of the models.

8. RESULTS AND CONCLUSIONS

	Model Name	Hyperparameter Tuning	Train R-squared	Test R-squared	Test RMSE
1	Linear Regression	No	0.412	0.419	299.334
2	Ridge Regression	No	0.412	0.312	325.808
3	Lasso Regression	No	0.400	0.314	325.248
4	Random Forest Regressor	No	0.850	0.849	152.459
5	Random Forest Regressor	Randomized Search CV	0.957	0.856	149.143
6	Decision Tree Regressor	No	0.712	0.689	219.125
7	Decision Tree Regressor	Grid Search CV	0.770	0.753	195.131
8	Decision Tree Regressor	Randomized Search CV	0.778	0.754	194.876
9	XG Boost	No	0.861	0.870	141.848
10	XG Boost	Randomized Search CV	0.979	0.863	145.157
11	Light GBM	No	0.847	0.852	150.840
12	Light GBM	Randomized Search CV	0.978	0.899	124.800
13	CAT Boost	No	0.955	0.898	140.250
14	CAT Boost	Randomized Search CV	0.975	0.900	123.920

Fig 8 – Results from all model implementations

Ensemble Learning Results:

For XG Boost without hyper parameter tuning, with the constraints set as max_depth=9 and the learning_rate=0.5. Using Randomized Search CV for hyper parameter tuning in order to achieve a more reliable estimate of what each candidate model's out of sample performance will be, the best parameters identified were.

For Light GBM without hyper parameter tuning, with the constraints set as max_depth=9, learning_rate=0.5, n_estimators=100. Using Randomized Search CV for hyper parameter tuning in order to achieve a more reliable estimate of what each candidate model's out of sample performance will be, the best parameters identified were.Learning Rate = 0.25

- Max Depth = 12
- Min Child Weight = 5
- Number of leaves = 122
- Estimators = 1244

For CatBoost without hyper parameter tuning, with the constraints set as max_depth=9, learning_rate=0.3. Using Randomized Search CV for hyper parameter tuning in order to achieve a more reliable estimate of what each candidate model's out of sample performance will be, the best parameters identified were.

- Learning Rate = 0.25
- Max Depth = 12

9. CONCLUSION AND BUSINESS INSIGHTS

Once we were able to figure out the best performing model, we made the use of a python library called - "Eli5". This library is used to display the best predictor variables by weight based on the prediction by a particular mode. Since the CAT Boost model performed best on the data, we used Eli5 with this and found the following result for the top predictor variables.

Weight	Feature
0.1204	category_Rice Bowl
0.1171	checkout_price
0.1140	op_area
0.1138	meal_id
0.0548	region_code
0.0524	emailer_for_promotion
0.0522	category_Sandwich
0.0499	week
0.0474	city_code
0.0467	cuisine_Italian
0.0464	center_id
0.0433	homepage_featured
0.0283	discount_offered
0.0268	base_price
0.0166	cuisine_Indian
0.0159	compare_week_price
0.0151	center_type_TYPE_C
0.0134	center_type_TYPE_B
0.0071	category_Salad
0.0051	category_Pasta

Fig 9 - Most Relevant Features - Eli5

We saw that the number of orders of the increased demand for food for the restaurants were based on the following factors.

1) Establishments selling Rice Bowl.
2) Checkout price of an item.
3) Area of operation of the establishment.
4) Promotion done by the establishment
5) Establishments selling sandwiches.

As a business decision, we can inform the establishments that the above features play a major role, if they want to increase their food demand.

10. FUTURE WORK

For forecasting financial data series, neural networks have been employed successfully. Classical approaches for time series prediction, such as Box-Jenkins or ARIMA, presuppose a linear relationship between inputs and outputs. The advantage of neural networks is that they can approximate nonlinear functions.

To achieve this, the model architecture would need to be changed, which is always the most difficult component. Several papers emphasize the significance of getting the model architecture correct. This may be considered as the 'new' feature engineering in many aspects. Leslie Smith's study proposes an intriguing strategy to selecting hyperparameters in a more disciplined manner.

It is necessary to determine the appropriate learning rate, weight decay, and embedding dropout.To determine the best hyperparameters, we need to run a learning rate finder with a few different weight decay and dropout values. The greatest combination with the lowest loss, highest learning rate (before rapidly growing), and maximum weight decay is then chosen.

We may also create models for predicting food demand based on the Nonlinear Autoregressive Exogenous Neural Network. To the best of our knowledge, such a model has not yet been developed; however, a type of RNN (Recurrent Neural Networks), i.e. many-to-one RNN, usually provides better predictions because it uses the additional information contained in the series of interest that has already been output prior to a given period.

11. PROJECT TAKEAWAYS

This project dealt with the forecasting of demand and supply of food orders. Our approach for Food Demand Forecasting tackles attempts at forecasting food demand by trying to find a trend in the features that tends to create an increase in the number of orders. From this project, we were able to learn how Machine Learning can be used in the case of Supply Chain Management. Since this project dealt with predicting a continuous value, we were able to experiment with various regression models. We also saw how creation of new features can also give an edge predicting the target variable. We also saw that various models performed differently on the data that we used for our project. It was observed that Boosting Ensemble models sometimes perform better than Random Forest on the data. One of the most important facts that we learned about Data Analysis is how tuning various hyperparameters impacts the performance of various models.

We did face some challenges during the project implementation. One of the major challenges was the scarcity of data. We had to work really hard to get a dataset that fulfilled our requirement of the data. Also the dataset was small in size. Another issue that we faced with the data was the unavailability of a proper data dictionary or a proper description of the variables in the dataset. We did not face any major issue while analyzing the data or while implementing the models.

Overall this project introduced us to a new field where machine learning and data science predictive techniques can play a vital role.

12. REFERENCES

[1] Patrick Meulstee and Mykola Pechenizkiy," Food Sales Prediction: "If Only It Knew What We Know"" 2008 IEEE International Conference on Data Mining Workshop.

[2] Eksoz, Can, S. Afshin Mansouri, and Michael Bourlakis. "Collaborative forecasting in the food supply chain: a conceptual framework." International journal of production economics 158 (2014): 120-135.

[3]Smirnov, P & Sudakov, Vladimir. (2021). Forecasting new product demand using machine learning. Journal of Physics: Conference Series. 1925. 012033. 10.1088/1742-6596/1925/1/012033.

[4] S. M. Taslim Uddin Raju, Amlan Sarker, Apurba Das, Md. Milon Islam, Mabrook S. Al-Rakhami, Atif M. Al-Amri, Tasniah Mohiuddin, Fahad R. Albogamy, "An Approach for Demand Forecasting in Steel Industries Using Ensemble Learning", *Complexity*, vol. 2022, Article ID 9928836, 19 pages, 2022. https://doi.org/10.1155/2022/9928836

[5] Silva, Juliana C., Manuel C. Figueiredo, and Ana C. Braga. "Demand forecasting: A case study in the food industry." International Conference on Computational Science and Its Applications. Springer, Cham, 2019.

[6] Pujara, Ansh, V. Pattabiraman, and R. Parvathi. "Food Demand Forecast for Online Food Delivery Service Using CatBoost Model." 3rd EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing. Springer, Cham, 2022.

[7] D. Adebanjo and R. Mann. Identifying problems in forecast-ing consumer demand within the fast paced commodity sector. Benchmarking: An International Journal, 7(3):223– 230, 2000.

[8] Bohdan M. Pavlyshenko," Machine-Learning Models for Sales Time Series Forecasting", 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 21–25 August 2018.