# Leveraging Insights from "Buy-Online Pickup-in-Store" data to Improve On-Shelf Availability


Pranav Saboo
*Krannert School of Management*
*Purdue University*
West Lafayette, Indiana
psahoo@purdue.edu

Sachin U. Arakeri
*Krannert School of Management*
*Purdue University*
West Lafayette, Indiana
sarakeri@purdue.edu

Shantam D. Mogali
*Krannert School of Management*
*Purdue University*
West Lafayette, Indiana
smogali@purdue.edu

Sushree S. Patra
*Krannert School of Management*
*Purdue University*
West Lafayette, Indiana
patras@purdue.edu

Zaid Ahmed
*Krannert School of Management*
*Purdue University*
West Lafayette, Indiana
ahmedl52@purdue.edu

Matthew A. Lanham
*Krannert School of Management*
*Purdue University*
West Lafayette, Indiana
lanhamm@purdue.edu



***Abstract*—This research provides insights on how to leverage 'Buy-Online Pickup-in-Store' data to understand customer preferences, demand patterns and when products go out-of-stock (OOS) to improve replenishment decisions for grocery chains. The motivation for this study is to reduce lost sales opportunities by improving on-shelf availability (OSA), subsequently improving the overall revenue and profitability of the retail store. In collaboration with a national grocery chain having over 240 stores in the USA, our team developed and assessed different predictive models to improve on-shelf availability rate. The solution uses various product categories based on the grocer's business segments, and then specific predictive models are implemented to predict stockouts for each category. While some research has been performed in this area, our work is novel in how OOS data from brick-and-click is utilized to advise the grocery stores on timely replenishment of stock to reduce overall lost sales. This research aims to evaluate and compare multiple classification algorithms for predicting OOS at a store-product level. Subsequently, the study performed an in-depth analysis to ascertain which business segments rendered better prediction accuracy.**



***Index Terms*—feature engineering, out-of-stock, on-shelf availability, retail industry, supply chain management, scalability ,product replenishment cycle**


## I. Introduction

Customers demand a channel-agnostic, seamless shopping experience across physical stores, mobile, online and other platforms. One of the strong omnichannel trends is the 'Buy-Online Pickup-in-Store' model, which integrates online and offline operations by allowing customers to place orders online and collect them in their chosen stores. The retail store is the last mile of supply chain management and error-free store execution ensures that the collective efforts of the whole supply chain yield desired results. Store execution primarily involves moving goods from the backdoor to the shelves to make the products available to end consumers. However, in-store logistics are highly labor-intensive. Store managers oftentimes ask store employees to perform shelf audits and enter them to the database for future reference. According to a study conducted by the University of Colorado, the implications of stock-out suggested that retailers on average lose 4 percent of their annual sales due to OOS items. The study also lost highlighted that on an average, OOS items cost the manufacturers $23 million for every $1 billion in sales [1].

The data used in our study provided new opportunities for the use of analytics in improving the ease of business. To manage the scale of the problem a machine learning model will predict the probability of having a product in stock when a customer order arrives. The retail giant Walmart recently admitted to a shelf-OOS problem and predicted a $3 billion opportunity in filling up the empty shelves created due to ineffective auditing and re-shelving operations [2]. It is one of the key performance components of customer service in retail. The complement to On-Shelf Availability (OSA) is Out-of-Stock (OOS) which can be defined as: 'a product not found in the desired form, flavor or size, not found in saleable condition, or not shelved in the expected location'(ECR Europe 2003) [3]. Simply put, an OOS occurs if a product is not available when a customer order arrives.

In recent years many machine learning techniques have been implemented to forecast sales and predict out of stock products, but the variety of approaches makes it challenging and time-consuming to pick the optimal methodology due to unpredictable consumer behavior . The objective of our study is to use multiple classification algorithms to predict OOS and improve on-shelf availability for retail stores. The dataset was obtained from a national grocer and it contains transactional information for online orders at a store-product level. There are a lot of factors that can drive the OOS rate some of which are promotions, balance on hand, seasonality, etc. We developed a model which can incorporate some of these features to accurately predict OOS. We also created new features using the existing variables to capture the cyclicity and seasonality in the dataset. We tried to present important issues like data cleaning, feature engineering, feature selection, and model evaluation criteria for the classification model.

The paper is majorly divided into the following sections: Literature Review summarizes the prior research conducted in the area of OOS prediction and how our study is an extension to the prior research. The Data section summarizes the data used for our study. Data dictionary and the Entity-Relationship diagram provides an understanding of the variables used and how they are related to each other. The Methodology section elaborates on the steps taken to reach the end goal. This section summarizes the steps for data preprocessing, upsampling, feature engineering, target transformation, parallel computation, and deployment. The Model section summarizes the model evaluation criteria used for this study and various models that were used and implemented. Results section summarizes the result and compares with the baseline model and the conclusion section summarizes the application and some of the recommendations from our study. Finally, the Reference section contains all the journals and websites that we referred to during this research study.

## II. Literature Review

A significant number of works of academic literature have been published regarding predicting out of stocks, demand forecasting, on-shelf availability, the causes, and the impact of stockouts and lost opportunity for retailers when a stockout occurs. Product availability is a measure of the service level a firm's supply chain offers to the end customer. High product availability means the consumers find and buy the products they want. The out-of-shelf measure is used to determine items that are not available on the store shelves. Out of stock cases are driving away revenue of multiple major retailers. Walmart recently admitted to a shelf-OOS problem and predicted a $3 billion opportunity in filling in empty shelves caused by ineffective auditing and re-shelving operations (Dudley, 2014)[5]. Not only that, Walmart even issued an urgent memo that demands store managers to improve grocery performance, which was seriously compromised by non-negligible shelf-OOS ratios (Greenhouse & Tabuchi, 2014) [6]. Predicting demand is a challenging task for FMCG products as there can be multiple contributors for a sudden surge or dip in demand in the market. However, it becomes all the more difficult when a retailer is present in various channels. Our study is limited to predicting out of stocks for online transactions only. Nevertheless, we needed to understand the ecosystem of multi-channel retailers as this can have a confounding effect on demand and thus affect the OOS rate.

In the paper "Towards a predictive Approach for Omni-Channel Supply Chains", (Enzo Morosini Frazzon et al. 2019) ) [7], The researcher aimed to find a predictive approach to deal with the complexities in demand forecasting of an Omni-present retailer by combining clustering with Artificial Neural Network. In this paper, the products were clustered depending on the channels using the K-means method. Inaccurate demand forecasting and thus, erroneous inventory levels can have a cascading effect on the bottom line of a retailer. Inaccurate store inventories hinder cross-channel fulfillment and also increase stockout possibilities for walk-in customers while overstaffing for replenishment by overestimating demands will burden the retailer with higher costs (Forrester, 2014)[8].

To accurately understand the client's expectations and model accordingly, it was essential for us to understand the definition of OOS for our study correctly. As in the retail industry, OOS cases also refer to the events wherein an item is in-store (e.g., misplaced or stored in the backroom), but it is unavailable to customers (Ton & Raman, 2010) [9]. Furthermore, we learned that it would be better to explore different models for each product category. The causes of the stockouts examined in various research studies indicate that the causes of retail stockouts are specific to the retailer, store, category, and product. No single solution works everywhere (Ehrenthal et al. 2013)[10] . Coming to the approach, a classification model was our first choice to model the problem. The study "Predict on-shelf product availability in grocery retailing with classification methods" (Dimitris Papakiriakopoulos 2011) [11], gave us a high-level understanding of algorithms which will potentially perform well to identify "out-of-shelf" products. We also used an ensemble learning method to increase the performance of the base classifiers. Another challenge we tackled while modeling was to handle imbalanced data. Our Random Forest classifier, which generally speaking, is robust to noise, also suffers from the curse of learning from an extremely imbalanced dataset. Because Random forest tends to focus more on the prediction accuracy of the majority class to minimize the overall error rate. Thus, resulting in poor accuracy for the minority class.

The paper (Chao Chen) [12] proposed two solutions: balanced random forest (BRF) and weighted random forest (WRF) to solve the issue of imbalance data in the Random Forest model. We got great insights from the paper, "Forty years of Out-of-Stock research – and shelves are still empty" (Jesper Aastrup et al. 2010) [13] for our feature selections. A good portion of this paper dealt with supply-side issues and analyzed the extent and root causes of situations resulting in out of stock cases. Another valuable insight from this paper which we experienced in our data as well was that the degree to which the OOS rate depends on the characteristics of the category i.e. OOS rates vary between categories, with the worst-performing at about 15 to 16%, and the best performing at OOS rates as low as 1%. One alternative approach for this project would have been Time series analysis, but we passed on that considering the granularity of the data provided. In Time Series, we would have required to aggregate the data either weekly and monthly, similar to (9 M.W.T. Gemmink 2017) paper, but this would have resulted in a naïve model with weak accuracy. Lastly, when modeling for the OOS rate, another important factor to incorporate is the timing of shelf audits for replenishment as it has a substantial influence on product availability, customer satisfaction, and sales performance (Aastrup et al. 2010)[13]. Unfortunately, in our research, the data regarding an audit for replenishment was not available, and this can have some effect on our model's prediction accuracy.

## III. Data

The data was collected from a national retailer which consists of online transactions from January 2019 to December 2019 for 3,547 products across 246 stores throughout the country amounting to 32.4 million records for this period. The transactional data is at the store-product level which contains information on ProductID, Store ID, Number of units sold, Number of digital transactions per day, Out-of-stock (OOS) transaction, and Sales forecasted for the next day (Table I).

The other data table used is the Product Hierarchy table which has information for a product with individual product level to the aggregated Business segment level. The other levels of hierarchy are business area, product category, product subcategory, and product class. The Entity-Relationship Diagram (ERD) shows the relationships between various attributes that explain the logical structure of all tables. The combination of Product ID, Day-Date, and Store ID makes the primary key for the OOS Prediction table.

## IV. Methodology

The methodology is executed in four consecutive steps. The first step describes the process of garnering raw data by physical store audit and retailer's database encompassing chronological data for 3,547 products dating back to January 2019. The second step illustrates the process of KDD [14] , EDA, and closely assess the next steps for data preparation, data selection, data cleaning, pre-defined business rules, the correct interpretation of results of information discovery and its business translation. The third step deals with feature engineering, the performance of various classification models, parallel computation using multi-processing capabilities of all server cores and narrow down on potential models. The third step shortlists the model to be proposed for deployment, evaluates the findings with the national retailer and involves multiple improvisations during the whole process. After evaluation and iterations, the last phase explains the steps to deploy the model and create a prototype to integrate with the retail chain's existing information system and be ready for consumption.

### A. Data Pre-Processing

Multiple ad-hoc data cleaning methods were applied to fix incorrect datatypes, remove extreme outliers due to typo entries, and inconsistencies from the data.

$$OOSRate = \frac{OOS_{txn}}{OOS_{txn} + \ Dgt\_Occ_{txn}}$$

The below table provides an overview of the issues found during the data cleaning process. Post-ad-hoc data cleaning, to prioritize focus on business impact and prepare computationally feasible dataset, below business segments were removed based on an insignificant amount of sales & out-of-stock transactions. See Table II

### B. Feature Engineering

To predict if a stock will go out of stock or not, certain exogenous variables could not be used such as unit, sales, etc. for the day to be predicted due to their unavailability on the day itself. The features were created to incorporate the cyclic nature of demand and on-shelf availability. For example, sales for the day to be predicted are not possible, therefore, to inculcate the information of sales for that day we take a standard deviation and mean by aggregating day, week, month, and quarterly basis. The engineered features provided a significant increase in overall accuracy. Further, dimensionality reduction was performed to remove features that were highly correlated to avoid redundancy, risk of overfitting, and optimize computational performance. Business segments that were extremely skewed such as Alcoholic Beverages, Grocery, Dairy, Dry Grocery, etc. observed increment in F1 score on average, by 60%. However, business segments that had OOS Rate greater than 20%, observed a 5-7% increment in F1 score.

Table III provides the information to identify features' contributing to the model's accuracy. We experimented with multiple features and found features listed in Table IV as the most significant ones.

### C. Upsampling

OOS data skewness is not uncommon in the retail industry (T. Gruen, D. Corsten, and S. Bharadwaj) [15] with most figures ranging between 3.3% to 12%. The skewness intrinsically makes the data less informational. Our approach was to segment the data granular enough to obtain good accuracy and simultaneously aggregate the data to avoid computational challenges. Table V provides a brief overview of the OOS Rate for different segments. Classification using skewed data is biased in favor of the majority class. The situation deteriorates when combined with high dimensional data. To tackle the imbalance of classes, we used an oversampling technique [16] which creates new minority class examples by extrapolating between existing examples.

TABLE I
DATA DICTIONARY

| Variable | Type | Description |
|---|---|---|
| P_ID | Categorical | Product ID |
| Day_dt | Date | Day date |
| ut_id | Categorical | Store ID |
| promo_flg | Categorical | Item on promotion |
| DP_fcst | Numeric | Day prior forecast |
| dgt_txn_occ | Numeric | Digital transaction count |
| DP_Units | Numeric | Day Prior Units sold |
| oos_tn | Numeric | Out-of-Stock transaction count |
| DP_OOS | Numeric | Day Prior OOS |
| DP_dgt_tn | Numeric | Day Prior Digital Transaction Count |
| dp_oos_chain | Categorical | Day Prior Total OOS for all Stores for a specific product |
| sell_thru | Numeric | dp_units dp_fcst |
| Boh | Numeric | Balance on Hand |
| Mdq | Numeric | Minimum Display Quantity |

TABLE II
ISSUES DISCOVERED IN DATASET

| Issue discovered | Root Cause | Cleaning Rule | Records affected |
|---|---|---|---|
| Products containing minimum display quantities (MDQ) over 100,000 units | Manual Entry for the value | Removed records having MDQ > physical possible space on the shelf | 0.07% |
| The Balance On Hand (BOH) containing negative values | Correction for products returned or replaced | Replaced with MDQ | 0.01% |
| Sell Through containing extreme high values | NA | Removed all values beyond four standard deviations | 0.04% |
| True Duplicate | Data extraction and loading repetition | Removed true duplicates | 0.01% |

TABLE III
FEATURES IMPORTANCE

| Feature | Importance |
|---|---|
| week_mean_dgt_txn_occ | 0.099476 |
| month_mean_dgt_txn_occ | 0.076283 |
| quarter_mean_dgt_txn_occ | 0.056487 |
| day_mean_dgt_txn_occ | 0.050176 |
| month_sd_dgt_txn_occ | 0.044761 |
| week_num_sd_dgt_txn_occ | 0.038701 |
| week_num_sd_chain_oos | 0.030058 |
| day_mean_chain_oos | 0.029998 |
| day_sd_dgt_txn_occ | 0.025305 |
| quarter_sd_dgt_txn_occ | 0.020243 |
| day_mean_sales | 0.006902 |
| week_num_sd_units | 0.005063 |
| DAY_DT_Ordinal | 0.004835 |
| month_sd_units | 0.004508 |
| day_sd_units | 0.004335 |

### *D. Target Transformation*

The dependent variable 'oos_tn' ranged between 0 to 35, which represented how many times the product was reported not on-shelf. Although this provides the intensity of demand for a product and could translate to sales, it becomes cumbersome and unintuitive to classify if a product is out of stock or not. Therefore, we transformed the dependent variable which can be expressed as follows:

$$OutOfStock_{(t,s)} = 0, OOS = 0$$

$$OutOfStock_{(t,s)} > 0, OOS = 1$$

### *E. Parallel Computation*

With roughly 11 billion data points (32 Million rows x 343 columns) to model, the computational system is bound to break at some point. During experimentation with various models, the kernel broke frequently despite segmenting and reducing data points substantially. The first solution to address this was to downsize the dataset until the kernel can parse, process and return the results successfully. Although the kernel completed the task, the time taken to model was as high as 20,000 seconds. ( 6 hours). An alternative to run the models was to perform out-of-core learning methods processes data in chunks. However, only a few data models support partial learning (out-of-core) and models such as Adaboost, Logistic Regression, etc. require the entire dataset to be fed to the model. To overcome the computational challenge, we leveraged all the cores of the server by exploiting multiprocessing capabilities. Each dataset was mapped to one core (total of 10) and executed in parallel. With this, we received results in 1/10th the time when ran 10 datasets in parallel. Table VI shows the comparison for two models executed with and without features and the time saved.

## V. MODEL

### *A. Model Evaluation Criteria*

The dataset is highly imbalanced with 94% majority class (Not OOS) and 6% minority class (OOS). As our area of interest is to predict OOS accurately which is the minority class, we focus on Precision and Recall. Recall summarizes the fraction of examples assigned as OOS that belong to the

TABLE IV
FEATURES CREATED & SELECTED FOR RESEARCH STUDY

| Type | Feature Name | Definition |
|---|---|---|
| Product-Time | DAY_DT_Ordinal | Converts the given date to proleptic Gregorian ordinal of the date |
| | day_mean_sales | Aggregate sales of product into mean for each day |
| | day_mean_chain_oos | Aggregate chain oos of product into mean for each day |
| | day_mean_dgt_txn_occ | Aggregate digital transaction of product into mean for each day |
| | day_sd_units | Standard Deviation of units for each day |
| | day_sd_dgt_txn_occ | Standard Deviation of digital transaction for each day |
| | week_mean_dgt_txn_occ | Aggregate digital transaction of product into mean for each week |
| | week_num_sd_units | Standard Deviation of units for each week |
| | week_num_sd_chain_oos | Standard Deviation of chain oos for each week |
| | week_num_sd_dgt_txn_occ | Standard Deviation of digital transaction for each week |
| | month_mean_dgt_txn_occ | Aggregate digital transaction of product into mean for each month |
| | month_sd_units | Standard Deviation of units for every month |
| | month_sd_dgt_txn_occ | Standard Deviation of digital transaction for every month |
| | quarter_mean_dgt_txn_occ | Aggregate digital transaction of product into mean for each quarter |
| | quarter_sd_dgt_txn_occ | Standard Deviation of digital transaction for every quarter |
| Product-OOS | not_dp_oos_since | Number of days since a product is not out of stock |
| Context Features | quarter | The quarter of the year |
| | month | The month of the year |
| | week_of_year | The week of the year |

TABLE V
BUSINESS SEGMENT CONSIDERED

| Rules | Business Segment | OOS Rate % | % of records | % of sales |
|---|---|---|---|---|
| Included | PRODUCE | 6.10% | 19.14% | 38.71% |
| | DAIRY | 3.70% | 20.91% | 19.66% |
| | GROCERY DSD | 2.90% | 18.62% | 16.79% |
| | DRY GROCERY | 3.30% | 20.60% | 8.53% |
| | DELI | 37.40% | 2.76% | 3.68% |
| | PACKAGED MEAT | 4.90% | 4.38% | 3.40% |
| | FROZEN FOODS | 4.50% | 6.92% | 2.66% |
| | MEAT | 36.00% | 1.32% | 2.43% |
| | CONSUMABLES | 3.60% | 3.16% | 2.17% |
| | ALCOHOLIC BEVERAGES | 1.60% | 0.24% | 0.62% |
| | BAKERY | 7.90% | 1.01% | 0.60% |
| | SEAFOOD | 22.60% | 0.37% | 0.58% |
| Excluded | PETS | 4.20% | 0.31% | 0.08% |
| | OTC HEALTH CARE | 2.00% | 0.14% | 0.05% |
| | BABY CONSUMABLES | 1.50% | 0.11% | 0.05% |
| | BULK WATER AND COFFEE | 21.70% | 0.02% | 0.01% |
| | BEAUTY CARE | 25.00% | 0.00% | 0.00% |

TABLE VI
PROCESSING TIME COMPARISONS FOR SERIES & PARALLEL FOR DELI & PACKAGED MEAT

| Dataset | Time Taken in Series | Time Taken in Parallel |
|---|---|---|
| Deli | 2,468.63 secs | 1,455.28 secs |
| Packaged Meat | 7,619.28 secs | 3,709.63 secs |

OOS class and precision summarizes how well the OOS class was predicted. Precision and recall can be combined into a single score that seeks to balance both concerns, called the F-score or the F-measure. The F-Measure is a popular metric for imbalanced classification. (See Table VII)

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$FMeasure = \frac{2 * Precision * Recall}{Precision + Recall}$$

## B. Model Experiments

As there are very few research studies which classify out of stock to improve shelf availability, we tried to examine Linear Models, Neural networks, and Decision Tree-based models for classification on Seafood and Alcoholic Beverages data to narrow down on promising models Utilizing each classification algorithm, we evaluated the f1 score, AUC score, Precision, Recall and time taken for each model. Based on these parameters we found that Random Forest, Logistic Regression and Ada boost performed relatively well over others. Because of the ease of interpretability of the Logistic Regression was one of the potential candidate models. The tables VIII and IX below show the results for multiple models run on the Seafood

TABLE VII
CONFUSION MATRIX DEFINITION

| | | Prediction n | Prediction p |
|---|---|---|---|
| Actual | n′ | True Negative | False Positive |
| Actual | p′ | False Negative | True Positive |

and Alcoholic Beverages dataset.

Because of data skewness, stratified 5-fold cross-validation was performed to observe the consistency of model performance to avoid overfitting of the trained dataset. Cross-validation also results in less biased models over models without cross-validation. RF consistently performed better than Logistic Regression and Ada Boost regardless of the skewness in the data

### C. Logistic Regression

Logistics regression was our first choice for this project because of the elegance, simplicity, and interpretability of the model. Since in our data cleaning process, we categorized all the not OOS cases as 0 and all the OOS cases as 1. This makes our logistics regression model as a Binary logistics Regression. Similar to linear Regression, Logistic Regression uses an equation as the representation. But the difference here is it uses a logistic function to fit the output of the equation between 0 and 1 instead of fitting a straight line or hyperplane. Below is the logistic function equation:

$$H(x) = \frac{1}{1 + exp(-x)}$$

The advantage of using Logistics regression is it is very efficient to train, easy to implement and uses less computational resources. Moreover, it doesn't require input features to be scaled, doesn't require much tuning and easy to regularize. Also, the logistic regression model gives us probabilities instead of just classifying; this makes decision making easy and straightforward. Few drawbacks of logistics regression are that we can't solve non-linear problems. Also, it doesn't perform well when feature space is too large, doesn't handle a large number of categorical features/variables well, and are vulnerable to overfitting. Since feature Engineering plays an essential role concerning the performance of Logistic Regression, the model does work better when we remove attributes that are unrelated to the output variable. The choice we made while selecting the features and synthesizing new ones resulted in good outcomes from the model. Also, being conservative in selecting features with multicollinearity helped us in getting good results.

### D. Random Forest

Random forest is a method similar to the Decision trees. In Decision trees used for classification, we used recursive binary splitting to divide the predictor space into various regions. The classification error rate is used as the criterion for each binary split. The better approaches that can be used individually for the binary splits are the Gini index and entropy. This process of decision trees is further improved by using Bootstrap aggregation, also called Bagging. The intuition behind Bagging is that averaging a set of observations reduces variance. One of the advantages of the Random Forest is that it is one of the most accurate decision models and works great on large datasets. Random Forest also assists in extracting variable importance. Coming to the drawbacks of Random Forests, it is vulnerable to overfitting in case of noisy data, and it's results are difficult to interpret. The reason why Random forest gave us fantastic results is its ability to handle the high dimensionality in our data. Since we have (Insert num of variables) variables and thus potentially high multicollinearity, Random forest is an apt method for classification of Out of stock (OOS) and non-OOS cases. Another reason is possibly a large quantity of data set availability for training the model as Random forest works great for large data sets.

### E. Ada Boost

Another classifier that gave us a great result is the Ada boost classifier. Ada-boost or Adaptive Boosting combines multiple classifiers to increase the accuracy of classifiers and is an iterative ensemble method. AdaBoost classifier builds a robust classifier by combining multiple poorly performing classifiers, which are generally a stump (a tree with just two leaves) to ultimately get a strong classifier. The intuition behind Adaboost is to set the weights of classifiers and to train the sample data sample in each iteration in such a way that it makes accurate predictions of unusual observations conducive. Now any statistical learning algorithm can be used as a base classifier if it accepts weights on the training set. In our model, the base classifier used was a default option that is a decision tree classifier.

## VI. RESULTS

By applying the classification algorithm on the dataset, the research concludes that the data without any enhanced features shows low F1 Score for business segments with high OOS Rate. Features engineered during the research, establish the possibility of developing a novel method for capturing information on cyclicity and seasonality of product's OOS behavior and utilize them to predict significantly better. The empirical results derived from the research show an improvement of 50% to 68% (Fig 3) and Table X in F1 Score for products with high OOS Rate. The study also provides marginal increment (1%-3%) in F1 Score for products that are already being predicted with accuracy as high as 90%. The results found from the study provides an innovative solution to the low on-shelf availability issues faced across retail industry especially for fast moving goods. By alleviating the key hurdle

TABLE VIII
MODEL RESULTS ON SEAFOOD DATASET (WITH & WITHOUT FEATURES)

| SEAFOOD | F score | | AUC Score | | Precision | | Recall | | Time Taken (s) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | W/O Feat | W Feat | W/O Feat | W Feat | W/O Feat | W Feat | W/O Feat | W Feat | W/O Feat | W Feat |
| Random Forest Classifier | 86% | 90% | 92% | 95% | 83% | 85% | 90% | 95% | 49 | 54 |
| Ada Boost Classifier | 85% | 88% | 92% | 94% | 81% | 84% | 90% | 92% | 39 | 53 |
| Logistic Regression | 84% | 89% | 92% | 95% | 78% | 83% | 92% | 95% | 15 | 13 |
| Complement NB | 84% | 85% | 92% | 91% | 78% | 82% | 91% | 87% | 0 | 0 |
| Multinomial NB | 84% | 85% | 92% | 91% | 78% | 82% | 91% | 87% | 0 | 0 |
| Bernoulli NB | 84% | 78% | 90% | 89% | 82% | 69% | 86% | 90% | 1 | 0 |
| Passive Aggressive Classifier | 84% | 89% | 92% | 95% | 77% | 83% | 92% | 96% | 1 | 2 |
| Liner SVC | 84% | 89% | 92% | 95% | 77% | 83% | 92% | 95% | 9 | 18 |
| Linear Discriminant Analysis | 84% | 88% | 92% | 94% | 77% | 83% | 92% | 94% | 11 | 11 |
| Ridge Classifier | 84% | 88% | 92% | 94% | 77% | 83% | 92% | 94% | 2 | 2 |
| Nearest Centroid | 82% | 83% | 88% | 88% | 84% | 85% | 80% | 81% | 0 | 0 |
| Decision Tree Classifier | 77% | 83% | 85% | 90% | 77% | 83% | 77% | 84% | 8 | 14 |
| MLP Classifier | 77% | 82% | 85% | 89% | 77% | 82% | 77% | 82% | 713 | 692 |
| Gaussian NB | 67% | 72% | 82% | 86% | 57% | 61% | 82% | 88% | 1 | 1 |
| Quadratic Discriminant Analysis | 58% | 71% | 77% | 86% | 44% | 59% | 86% | 90% | 9 | 6 |

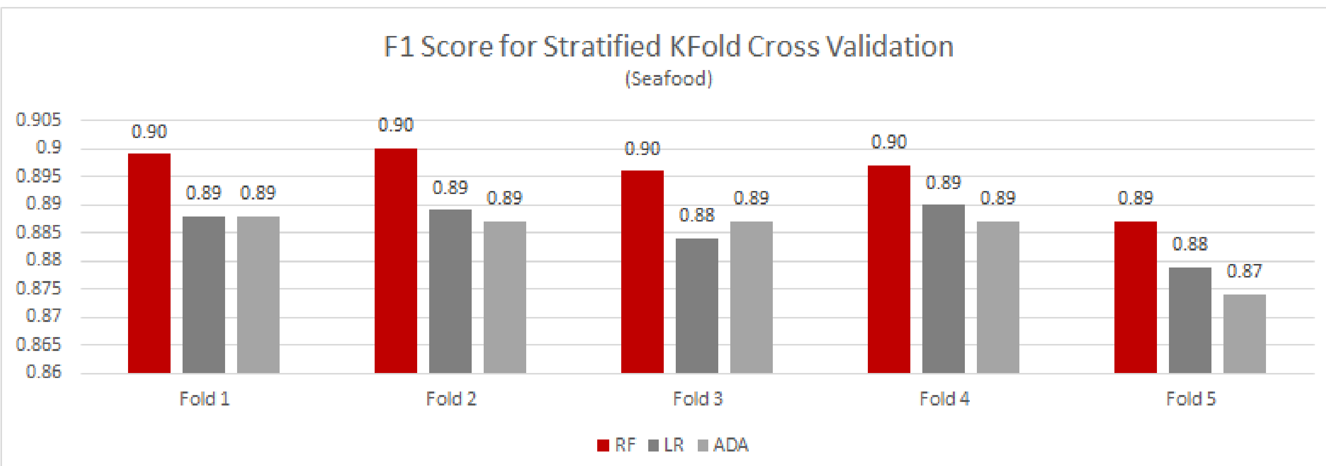


Fig. 1. F1 Score for Stratified K-fold Cross-Validation (OOS Rate high)

of low OSA, retailers can reduce significant source of loss of revenue, provide better customer experience, and maintain tighter supply chain management.

The figure below shows the business categories and the sub-categories within them where the model was able to perform relatively better than compared to other sub-categories. The figure summarizes the categories with f scores greater than 80%. Therefore, drilling down on each segment, the below sub-categories can be utilized to predict OOS with high confidence.

## VII. CONCLUSIONS

Engineered features showed significant improvement on model performance. Features like weekly mean, monthly mean, quarterly mean, monthly SD and weekly SD for digital transactions proved to have the highest importance in identifying OOS rate at product-store level. We incorporated the costs of incorrect classifications to compare models based on business measures and found Random forest to be the best performer. Based on the cost matrix, the estimated potential savings from our predictions was $50M per year Although the study provides significant benefits from both retailers' and customers' perspective, it still holds certain drawbacks.

The study could possibly perform better if provided with more data per store per product. Empirical results show that if dataset contains information of product that goes out of stock in at least 6 stores everyday (roughly 2,000 OOS counts),

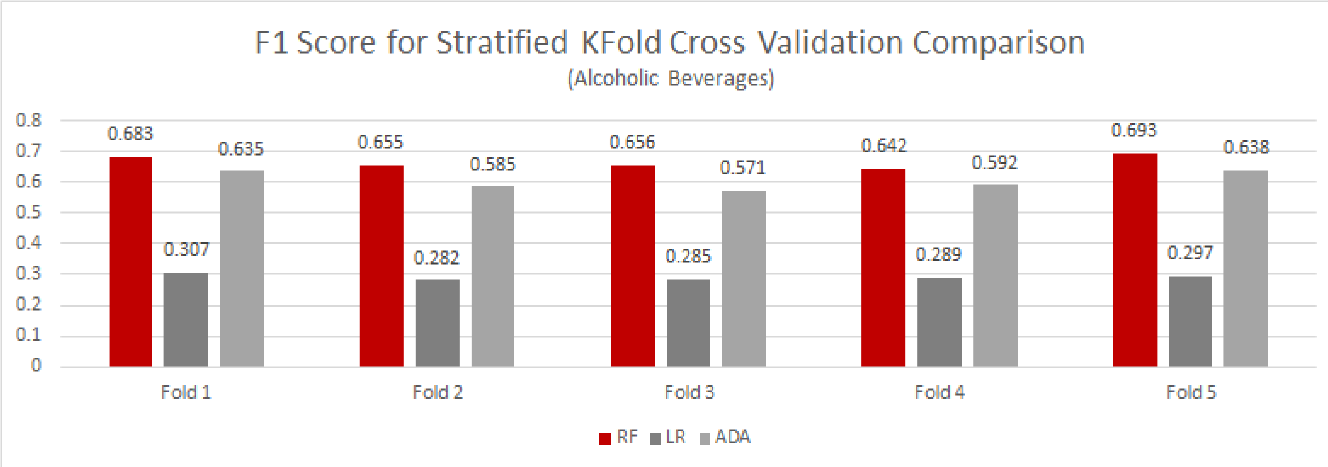


Fig. 2. F1 Score for Stratified K-fold Cross-Validation (OOS Rate low)

TABLE IX
MODEL RESULTS ON ALCOHOLIC BEVERAGES DATASET (WITH & WITHOUT FEATURES)

| ALCOHOLIC BEVERAGES | F score | | AUC Score | | Precision | | Recall | | Time Taken (s) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | W/O Feat | W Feat | W/O Feat | W Feat | W/O Feat | W Feat | W/O Feat | W Feat | W/O Feat | W Feat |
| Random Forest Classifier | 1% | 67% | 50% | 77% | 67% | 85% | 1% | 55% | 51 | 71 |
| Ada Boost Classifier | 19% | 61% | 62% | 84% | 14% | 56% | 27% | 69% | 59 | 49 |
| Decision Tree Classifier | 12% | 53% | 55% | 78% | 11% | 50% | 12% | 56% | 10 | 49 |
| MLP Classifier | 6% | 51% | 52% | 72% | 9% | 60% | 4% | 44% | 771 | 355 |
| Extra Trees Classifier | 3% | 48% | 51% | 67% | 33% | 81% | 1% | 34% | 109 | 110 |
| Linear Discriminant Analysis | 9% | 31% | 73% | 87% | 5% | 19% | 67% | 79% | 29 | 12 |
| Ridge Classifier | 9% | 31% | 73% | 87% | 5% | 19% | 67% | 79% | 2 | 3 |
| Passive Aggressive Classifier | 10% | 30% | 77% | 87% | 6% | 18% | 75% | 80% | 4 | 3 |
| Liner SVC | 9% | 29% | 73% | 87% | 5% | 18% | 66% | 81% | 12 | 13 |
| Logistic Regression | 10% | 28% | 74% | 87% | 5% | 17% | 67% | 81% | 14 | 16 |
| Bernoulli NB | 13% | 18% | 76% | 86% | 7% | 10% | 66% | 85% | 1 | 1 |
| Complement NB | 7% | 8% | 69% | 71% | 4% | 4% | 66% | 66% | 0 | 0 |
| Multinomial NB | 7% | 8% | 69% | 71% | 4% | 4% | 66% | 66% | 1 | 0 |
| Nearest Centroid | 6% | 7% | 68% | 68% | 3% | 4% | 70% | 63% | 1 | 0 |
| Gaussian NB | 3% | 3% | 51% | 52% | 2% | 2% | 92% | 94% | 1 | 1 |
| Quadratic Discriminant Analysis | 3% | 3% | 50% | 51% | 2% | 2% | 92% | 97% | 26 | 7 |

the predicting capabilities give higher accuracy. Moreover, the data also lacks the information about how often a product is replenished at each store which generally affects the whole process of knowledge discovery and modeling training. Lastly, even though class imbalance was handled by creating synthetic records, the imbalance still affects the overall performance and possibly hinders the learning process. The next steps for the research entails garnering granular data, increasing data points containing OOS, and information about replenishment rate. Such value addition will help engineer better features and might provide more accurate results, ultimately tackling industry-wide problem and possibly extending into other sectors as well.

## REFERENCES

[1] Gruen, Thomas W., and Daniel S. Corsten. A Comprehensive Guide to Retail out-of-Stock Reduction in the Fast-Moving Consumer Goods Industry. Grocery Manufacturers of America, 2007. (P.11)
[2] Aastrup, Jesper, and Herbert Kotzab. “Forty Years of Out-of-Stock Research – and Shelves Are Still Empty.” The International Review of Retail, Distribution and Consumer Research, vol. 20, no. 1, 2010, pp. 147–164., doi:10.1080/09593960903498284.
[3] Europe, E. C. R. (2003). ECR-optimal shelf availability: increasing shopper satisfaction at the moment of truth. ECR Europe.
[4] Gemmink, M. (2017). A Model to Prevent Stockouts in Retail using Time Series Sales Forecasting.
[5] (n.d.). Retrieved from https://www.bloomberg.com/news/articles/2014-03-28/wal-mart-says-refilling-empty-shelves-is-3-billion-opportunity

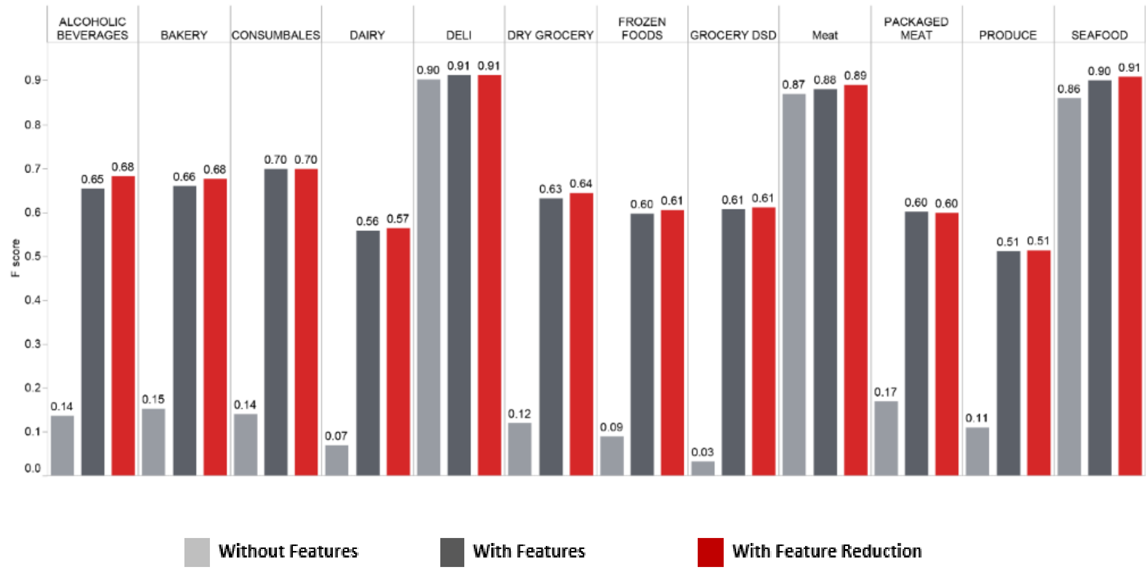


Fig. 3. Results comparing F1 Score obtained using Without Features, With Feature, and With Reduced Features

TABLE X
IMPROVEMENT ON BASE MODEL USING FEATURES

| Business Segment | Base F1 | Improvement using features |
|---|---|---|
| Deli | 0.90 | 1% |
| Meat | 0.87 | 1.5% |
| Packaged Meat | 0.04 | 56% |
| Dry Groceries | 0.04 | 61% |
| Grocery DSD | 0.04 | 58% |
| Seafood | 0.86 | 5.2% |
| Frozen Food | 0.04 | 57% |
| Alcoholic Beverages | 0.17 | 66% |

[6] Greenhouse, S., & Tabuchi, H. (2014, November 12). Walmart Memo Orders Stores to Improve Grocery Performance. Retrieved from https://www.nytimes.com/2014/11/12/business/walmart-memo-orders-stores-to-improve-grocery-performance.html
[7] Pereira, M. M., & Frazzon, E. M. (2019, December 25). Towards a Predictive Approach for Omni-channel Retailing Supply Chains. Retrieved from https://www.sciencedirect.com/science/article/pii/S2405896319311863
[8] Customer Desires Vs. Retailer Capabilities: Minding The ... (n.d.). Retrieved from https://www.accenture.com/us-en/ /media/accenture/conversion-assets/dotcom/documents/global/pdf/technology_7/accenture-customer-desires-vs-retailer-capabilities.pdf
[9] Ton, Z., & Raman, A. (2010, August 19). The Effect of Product Variety and Inventory Levels on Retail Store Sales: A Longitudinal Study. Retrieved from http://onlinelibrary.wiley.com/doi/10.1111/j.1937-5956.2010.01120.x/abstract
[10] A Theoretical Overview of the Stockout Problem in Retail ... (n.d.). Retrieved from https://www.researchgate.net/publication/328337965_A_Theoretical_Overview_of_the_Stockout_Problem_in_Retail_from_Causes_to_Consequences
[11] Papakiriakopoulos, D. (2011, October 13). Predict on-shelf product availability in grocery retailing with classification methods. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417411014643
[12] Chen, Chao. Using Random Forest to Learn Imbalanced Data. Aastrup, J., & Kotzab, H. (2010). Forty years of Out-of-Stock research – and shelves are still empty. The International Review of Retail, Distribution and Consumer Research, 20(1), 147–164. doi: 10.1080/09593960903498284
[13] Aastrup, J., & Kotzab, H. (2010). Forty years of Out-of-Stock research – and shelves are still empty. The International Review of Retail, Distribution and Consumer Research, 20(1), 147–164. doi: 10.1080/09593960903498284
[14] Fayyad, U. (2001). Knowledge Discovery in Databases: An Overview. Relational Data Mining, 28–47. doi: 10.1007/978-3-662-04599-2_2
[15] Corsten, D., & Gruen, T. (n.d.). On Shelf Availability: An Examination of the Extent, the Causes, and the Efforts to Address Retail Out-of-Stocks. Consumer Driven Electronic Transformation, 11. doi: 10.1007/3-540-27059-0_9
[16] Chawla, N. V., et al. "SMOTE: Synthetic Minority Over-Sampling Technique." Journal of Artificial Intelligence Research, vol. 16, Jan. 2002, pp. 321–357., doi:10.1613/jair.953.