# Predicting Product Reorders: A Machine Learning Approach for Instacart Market Basket Analysis


Chanyoung Park
Department of Data Analytics Engineering
George Mason University
Fairfax, VA, United States
cpark50@gmu.edu

Zijie He
Department of Data Analytics Engineering
George Mason University
Fairfax, VA, United States
zhe20@gmu.edu

Oluwasegun Adegoke
Department of Data Analytics Engineering
George Mason University
Fairfax, VA, United States
oadegoke@gmu.edu

Saketh Kallepalli
Department of Data Analytics Engineering
George Mason University
Fairfax, VA, United States
skallep@gmu.edu



***Abstract*—** Understanding customer buying habits using market basket analysis is at the core of recommendation system improvement. This project puts to work some machine learning models in predicting product reorderability using the Instacart Market Basket dataset. In this paper, we put together a broad predictive model to identify probable product reorders using logistic regression, random forest, gradient boosting, XGBoost, and deep learning models. The analysis done shows that the ensemble and deep learning models significantly outperform the traditional approaches, while the deep learning model achieves an accuracy of 90.4% and a recall of 63% for reordered products. These findings yield actionable insights for personalized recommendations and inventory optimization within online grocery platforms.



***Keywords—Machine Learning, Deep Learning, Ensemble Models, Recommendation Systems, Predictive Analytics, Feature Engineering.***


## 1. *Introduction*

Online grocery platforms, like Instacart, face significant challenges when it comes to predicting customer purchasing behaviors and generating personalized recommendations. Understanding the probability of product reorders helps enhance customer satisfaction, makes inventory management smoother, and boosts overall shopping experiences [1]. Machine learning techniques provide robust instruments for interpreting intricate purchasing patterns and anticipating future consumer actions.

### *1.1 Objectives*

The primary objectives of this research are:

- Develop accurate machine learning models for predicting product reorders
- Compare performance across different algorithmic approaches
- Identify key features influencing reorder probabilities
- Generate actionable recommendations for e-commerce grocery platforms

## *2. Literature Review*

In the last couple of years, machine learning has grown as an essential tool for any prediction of customer behavior in a retail company. Various machine learning models, such as logistic regression, random forest, and gradient boosting, were able to demonstrate high efficacy in predicting customer preferences, product recommendations, and shopping trends. The mentioned techniques are especially good in handling tricky features of retail data that usually consists of huge amounts of data with high dimensionality.

## *3. Data Overview and Preprocessing*

### *3.1 Dataset Description*

The dataset used in the present study, Instacart Market Basket, contains over 3 million orders from over 200,000 unique users. It is constituted of several parts, each representing order history, product details, user-specific information, and product-specific attributes. Each of these various kinds of information is enclosed in several CSV files:

- **User order histories**
- **Product details**
- **Temporal ordering patterns**
- **Product categorizations**

### *3.2 Data Cleaning and Preprocessing Steps*

The preprocessing step focused on data quality and feature engineering for variables that were deemed important for the analysis. Missing values in the 'days_since_prior_order' column, which represented first-time orders, were replaced with zeros. Key identifiers like product IDs and user IDs were checked for completeness to ensure that incomplete records would not be included in model training. Many datasets were combined into one, including orders, products, aisles, departments, and prior and training order-product datasets. It connected user purchase histories, product details, and ordering behaviors into one and provided a sound basis for more profound trend analysis. Furthermore, measures were implemented to prevent dataset bias by ensuring that both frequent and infrequent users, as well as products, were adequately represented. This balanced methodology sought to enhance the capability of generalization across different user segments by the models.

### *3.3 Feature Engineering and Scaling*

Feature engineering: played an important role in enhancing model performance by constructing features that captured user-product interactions and temporal purchasing patterns.

**Order Frequency**: Features like mean order frequency and days since the last order were indicative of user purchasing habits and shed light on the likelihood of reordering certain products.

**Reorder Rate**: The feature told about the tendency of users to reorder certain items; this helped the model recognize habitual or essential purchases.

**Temporal Patterns**: Features like average order hour and day of the week shone light on when users were most likely to place an order, thus enabling better predictions.

**Product Popularity**: It was computed from the total number of times a product had been ordered; this feature provided a broad view of which items were most likely to be reordered.

**User-Product Interaction**: Interaction features, such as the number of previous orders of a product by a particular user, made the model tailor to individual user tastes.

**Cart Position**: This included the position of an item in the cart, since often the first added items represented staples, and the latter represented impulse buys.

**Feature Scaling:** StandardScaler was used to scale the numerical features to a common range, which helps in improving the performance of machine learning models. Models like logistic regression and gradient boosting benefit from feature scaling as it leads to faster convergence and better accuracy. For tree-based models, such as random forest and XGBoost, scaling is not necessarily required; however, it was done here for consistency among all models.

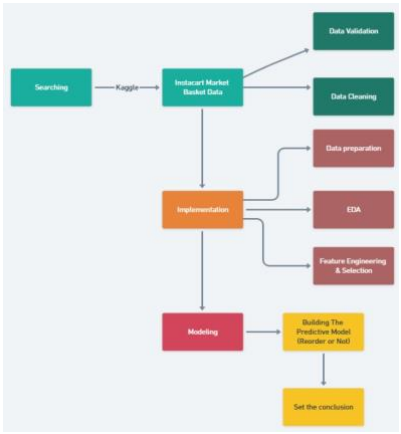


*Figure 1 Project Framework*

### *4. Exploratory Data Analysis (EDA)*

Exploratory data analysis has been performed to understand the behavior of customers and to identify key trends. Some of the key findings from EDA are:

- **The most reordered products** include organic bananas and avocados, which denotes their popularity and necessity within the user's shopping lists. The top ten most reordered products are visualized with a bar chart:

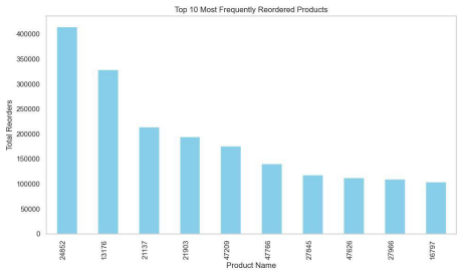


*Figure 2 Popularity of Products*

- **Time-Based Reordering Patterns**: Most orders were placed on Sundays, with a peak during the early afternoon (between 12 PM and 3 PM). This suggests that customers often do their grocery shopping during weekends and around lunchtime. Promotions could be optimized for these times to increase sales.

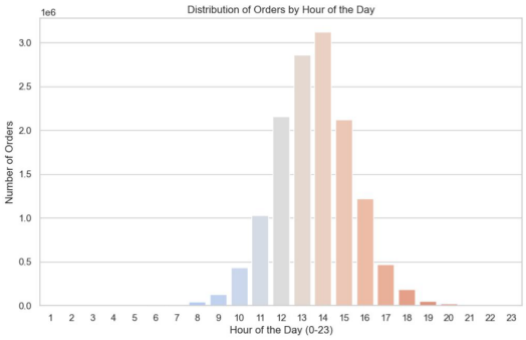


*Figure 3 Temporal Pattern*

- **Customer Reordering Behavior:** Users with higher reorder rates tend to exhibit consistent purchasing habits. Products added early to the cart were often high-priority items, suggesting that personalized recommendations should prioritize these products.

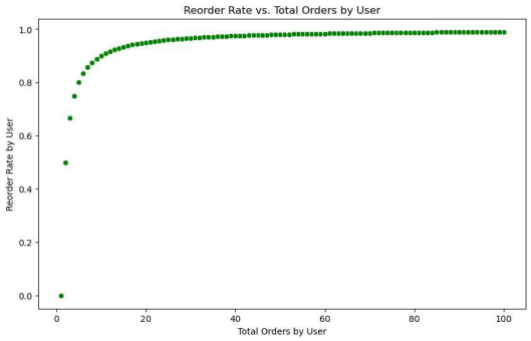


*Figure 4 Reorder Rate*

### *5. Machine Learning Models*

We trained five independent machine learning models:

1. **Logistic Regression:** Baseline linear classifier
2. **Random Forest:** Ensemble method capturing non-linear interactions
3. **Gradient Boosting:** Iterative error correction method
4. **XGBoost:** Extreme gradient boosting
5. **Deep Learning Neural Network:** Multilayered perception for complex patterns

#### *5.1 Logistic Regression Model*

Logistic regression was used to establish a baseline for the task of predicting product reordering. Its simplicity and interpretability made it a very powerful tool in understanding the interaction between features and reorder behavior.

The data was then split into a training and test set in a 70-30 ratio to make sure that the target variable was stratified to ensure class balance. The model was trained using scikit-learn

with a maximum iteration limit of 1000 to ensure proper convergence.

The accuracy on the test set was 77%. A confusion matrix showed the model was very good at predicting products that were not reordered but struggled when it came to predicting what products were reordered. There was a recall of only 10% for the reordered class, meaning that many of the reordered were missing. However, precision for the reordered class was 56%, showing that the model was relatively stingy in its predictions, not making many false-positive mistakes.

The classification report showed an imbalance in performance between the two classes, with much better metrics for the non-reordered class compared to the reordered class. This brings out the difficulty in a linear model in capturing effectively the intricacies of the reorder behavior.

ROC curve analysis provided an AUC score of 0.71, indicating moderate discriminatory ability between the reordered and non-reordered classes. This further provides evidence that logistic regression is a good starting point in understanding feature importance and baseline prediction but does not really handle non-linear interactions and complexities inherent within the data.

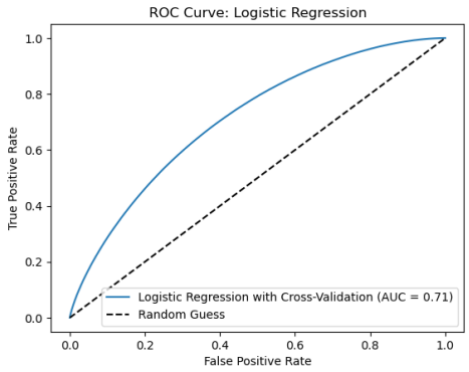


*Figure 5 Logistic ROC Curve*

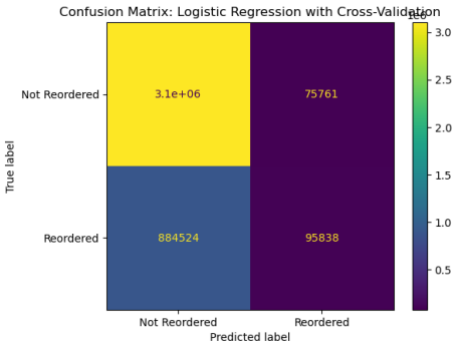


*Figure 6 Logistic Confusion Matrix*

### *5.2 Random Forest Model*

The random forest model was, therefore, used to offset these limitations of logistic regression by capitalizing on the former's ability to handle non-linear feature interactions and higher-dimensional data. Being an ensemble learning technique based on decision trees, random forests are particularly good at recognizing complex patterns in data while, at the same time, reducing the risk of overfitting by averaging many trees.

The model was trained using 100 estimators, with a maximum depth set at 10, and the minimum samples required for a split set at 2. These parameters were selected to balance computational efficiency and model performance. The feature importance measures produced by the random forest provided very interesting insights about the variables that contributed the most to reorder likelihood estimation.

The random forest model achieved 78% accuracy on the test set, an improvement over logistic regression. The classification report showed improved recall for the reordered class at 16%, which meant that the model had become better at identifying reordered products. Precision for the reordered class was also increased to 69%, showing a better balance between false positives and true positives. The F1-score for the reordered class also rose, thus showing an overall improvement in performance with respect to the minority class.

Feature importance analysis showed that variables such as reorder rate, mean days since the previous order, and user-specific purchase frequency were some of the most predictive features for the probability of reordering.


*Figure 7 Radom Forrest Feature Importance*

### *5.3 Gradient Boosting Model (basic)*

Gradient boosting was used to increase predictive performance by sequentially building models that correct errors produced by previous iterations. This allows exploration of complex, non-linear associations of the data while still maintaining a relatively high level of flexibility.

The model was trained using 100 estimators, a learning rate of 0.1, and a maximum depth of 3 for each tree. These hyperparameters were chosen to balance the trade-off between model complexity and overfitting effectively. Gradient boosting achieved an accuracy of 87% on the test set, outperforming logistic regression and random forest in terms of overall predictive power by a large margin.

The classification report shows that the precision of this reordered class was 95%, with recall at 49%, which gave the F1-score of 64%. The result has shown a substantial improvement of the model's ability in identifying reordered products while the precision remains at a very high level. The larger recall indicates the model might include a much wider variety of reordered instances compared with the previous models.

Results showed that gradient boosting was particularly successful at improving recall for the reordered class, making it a good choice in scenarios where the capture of reordered products is vital. However, some further hyperparameter tuning and class imbalance mitigation are expected to improve results.

### *5.4 Gradient Boosting Model (XGBoost)*

XGBoost is an advanced gradient boosting implementation for high performance. Its regularization capabilities, in combination with its optimization techniques, make it especially suitable for dealing with large datasets and reducing overfitting.

The model was trained with 100 estimators and a learning rate of 0.1; all other regularization parameters were left at their defaults. The model had used cross-validation at training time to estimate how it performed given the varying data splits—mean accuracy scores 87.8%. It showed an 88% accurate class prediction on the test set, outdoing the base gradient boosting model.

The classification report showed that the reordered class had a precision of 92% and a recall of 52%, resulting in an F1-score of 66%. The increase in recall, in comparison with gradient boosting, flags the fact that XGBoost finds a greater number of reordered instances and thus is very powerful in dealing with imbalanced datasets.

XGBoost has demonstrated better predictive performance than earlier models, mainly because of its ability in handling imbalance data and capturing complex patterns. Its computational efficiency, along with its flexibility, makes it one of the best choices for large-scale predictive tasks; still, further tuning may further improve its performance.

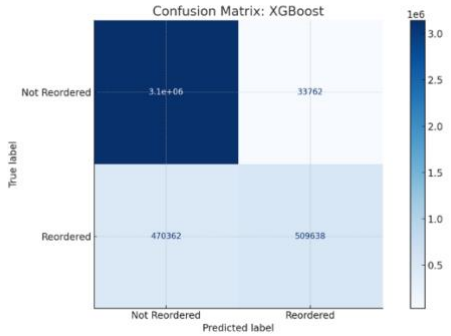


*Figure 8 XGBoost Confusion Matrix*

### *5.5 Deep Learning Model*

The deep learning model is implemented to test its ability in representing complex, non-linear relations in data. The architecture used is that of a Sequential neural network, which contains many fully connected layers with ReLU activation functions. The last output layer uses a sigmoid activation function for mapping the outputs to probability for the binary classification task.

The architecture of the model contained three hidden layers with 256, 128, and 64 neurons, respectively. The network was trained for 20 epochs using the Adam optimizer with a binary cross-entropy loss function. The batch size was 128 to help with the training process. Accuracy and recall were used as metrics during training to assess performance.

The deep learning model achieved an accuracy of 90.4% on the test set, which was the best among all models implemented earlier. The classification report highlighted a recall of 63% of the reordered class, showing an increase in the power of this model in classifying reordered products compared with gradient boosting and XGBoost. The precision over the reordered class was 61%, with an F1-score of 62%, which illustrates good balance between recall and precision.

During training, both training and validation accuracy showed gradual improvement, and the loss decreased; hence, the model was able to learn from the data without considerable overfitting.

While the deep learning model showed superior performance, it also required significantly larger computational resources and training time compared to traditional machine learning models. Results show that it was good at capturing complex patterns, but there is every indication that this could be improved further through advanced techniques, such as using dropout for regularization, hyperparameter optimization, or training on an even larger dataset.

Deep learning has proved to be the best model in this study, with the highest levels of accuracy and recall for the reordered class. Performances of this kind make the model very apt for large-scale applications where predictive accuracy is of utmost concern.

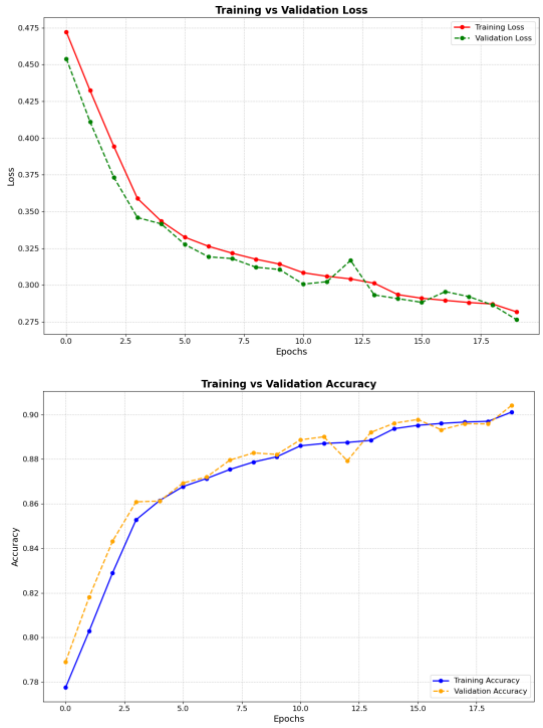


*Figure 9 DL Training and Validation*

### *6. Results and Discussion*

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Logistic Regression | 0.77 | 0.56 | 0.10 | 0.17 |
| Random Forest | 0.78 | 0.69 | 0.16 | 0.26 |
| Gradient Boosting | 0.87 | 0.95 | 0.49 | 0.64 |
| XGBoost | 0.88 | 0.92 | 0.52 | 0.66 |
| Deep Learning | 0.90 | 0.61 | 0.63 | 0.62 |

*Figure 10 Model Comparison*

### *6.1 Summary of Model Performances and Feature Importance*

The performance of each model varied quite a lot, therefore underlining their respective strengths and weaknesses. Logistic regression served as a baseline model, providing interpretability, but it struggled with capturing the complex interactions in the data. In contrast, Random Forest considerably improved by fittingly handling interactions of features and ranking features importances, hence tagging variables such as reorder rate and user purchase patterns as critical. The Gradient Boosting and XGBoost algorithms once again presented greater predictive power, and XGBoost took the highest accuracy among the traditional machine learning models. The deep learning model outperformed all the other models, excelling in recall and discriminatory ability.

Key feature importance rankings across models consistently highlighted variables such as reorder rate, mean days since prior order, and user-specific purchase frequency as the most impactful predictors of reorder likelihood.

### *6.2 Accuracy, Precision, Recall, and F1 Score Comparisons*

The models showed significant trade-offs on the metrics of accuracy, precision, recall, and F1 score. Logistic regression scored an accuracy of 77% with strong precision for the non-reordered class but poor recall for reordered items. Random Forest improved the recall for the reordered class to 16% while maintaining an accuracy of 78%.

Gradient Boosting significantly increased recall to 49%, with accuracy at 87%, whereas XGBoost reached the highest accuracy of the traditional models at 88% and somewhat raised recall to 52%.

With an accuracy of 90.4%, a recall of 63% for the reordered class, and an F1-score of 62%, the deep learning model outperformed all traditional approaches. The large increase in recall points out its ability to catch complex patterns, therefore being more effective, most particularly in datasets skewed at overcoming the problems.

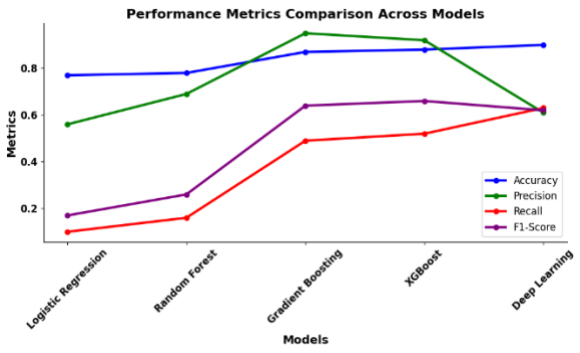


*Figure 11 Performance Metrics*

### *6.3 Strengths and Limitations of Each Model*

- **Logistic Regression:** The main advantage of logistic regression is interpretability and simplicity, which make it an excellent baseline model. However, it has a hard time modeling non-linear relationship, leading to suboptimal performance for the reordered class.
- **Random Forest:** The Random Forest algorithm effectively captures non-linear feature interactions and offers valuable insights through feature importance rankings. Nevertheless, its performance improvements are relatively limited when compared to boosting methods, especially in terms of recall.
- **Gradient Boosting:** performed with a significant increase in recall and overall accuracy; hence, it is an effective approach for reordered product prediction. Its iterative learning process allowed it to rectify the mistakes from previous iterations; however, it also needed very cautious hyperparameter tuning to avoid overfitting.
- **XGBoost**: is the most well-balanced among the compared traditional models in terms of performance versus computational efficiency. Its regularization features are effective at mitigating overfitting, and its scalability makes it appropriate for large datasets. However, like Gradient Boosting, it requires comprehensive tuning to achieve optimal performance.
- **Deep learning model:** had the best recall and overall accuracy, utilizing the advantages of its ability to learn complex patterns in data. However, it entailed much larger computational resources and was prone to overfitting without careful regularization and monitoring.

## *7. Conclusion and Recommendations*

Analysis of the different machine-learning models for product reorder prediction on Instacart's dataset reveals strengths and weaknesses of these methods. The logistic regression model created a simple, interpretable baseline, while the more complex models like Gradient Boosting, XGBoost, and Deep Learning significantly outperformed it in terms of predictive accuracy and recall for the reordered class.

### *7.1 Recommendations for Instacart*

1. Use Gradient Boosting or XGBoost in Production: These models have very good accuracy, recall, and precision; thus, they are quite robust in practical applications. XGBoost offers high computational efficiency and scalability, which is very important when working on large datasets like Instacart.

2. Focus on Feature Engineering: Features such as reorder rate, user-specific purchase frequency, and the average days since the last order have been reported to be highly influential. Other features like price sensitivity of the products, seasonality of demand, and customer categorization may improve the prediction accuracy.

3. Implement the Deep Learning Model for Critical Applications: The deep learning model achieved the highest recall and generally performed best; thus, it would be most applicable to high-stakes settings where predicting reordered items with few false negatives is crucial. Its computational cost, however, may make it less applicable to all operations.

## *References*

[1] S. D. Kalkar and P. M. Chawan, “Recommendation System using Machine Learning Techniques,” Sep. 28, 2022. https://www.researchgate.net/publication/363891251_Recommendation_System_using_Machine_Learning_Techniques

[2] Nurhayati Buslim, “Ensemble learning techniques to improve the accuracy of predictive model performance in the scholarship selection process,” Journal of Applied Data Sciences, vol. 4, no. 3, pp. 264–275, Sep. 2023, doi: https://doi.org/10.47738/jads.v4i3.112.

[3] Ekaterina Katya, “Exploring Feature Engineering Strategies for Improving Predictive Models in Data Science,” Deleted Journal, vol. 4, no. 2, pp. 201–215, Dec. 2023, doi: https://doi.org/10.52710/rjcse.88.

[4] R. Kumar and L. K. Shrivastav, “Gradient Boosting Machine and Deep Learning Approach in Big Data Analysis,” Journal of Information Technology Research, vol. 15, no. 1, pp. 1–20, Jan. 2022, doi: https://doi.org/10.4018/jitr.2022010101.

[5] Mishra, Ranjan & Reddy, G Y Sandesh & Pathak, Himanshu. (2021). The Understanding of Deep Learning: A Comprehensive Review. Mathematical Problems in Engineering. 2021. 1-15. 10.1155/2021/5548884.