Presented on Dec 2023

Instacart Purchase Predictor

Modeler: Ko-Jen Wang

Overview

Instacart Kaggle project: Predicting next purchase using data analysis for personalized shopping and optimized recommendations

01

Instacart is an American company that operates as a same-day grocery delivery and pick-up service in the U.S. and Canada. Customers can select products from various stores via the Instacart app and have them delivered by personal shoppers.

02

Our project aims to **use machine learning to predict which products a user will buy again in their next order** based on their previous orders.

03

Our project targets **data scientists** at Instacart and interacts with **product managers**, **marketing team leaders**, and possibly even **executives**.

WHY ARE WE SOLVING THIS PROBLEM?

Customer Centricity

Provide personalized recommendations. Make **enjoyable shopping experience** for customers and save their time.

Increased Sales

Accurate product predictions can lead to increased sales for Instacart. **Increase company's revenue** by suggesting products that the customer is likely to buy.

Business Strategy

This project will help Instacart make **data-driven** decisions and improve overall business strategy & operations by understanding customer behavior and preferences.

Datasets Overview

aisles.csv (134)

Contains product aisle information.
Key variables: 'aisle_id,' 'aisle.'
Importance: Enhances product categorization for better shopping experiences.

departments.csv (21)

Provides product department details.
Key variables: 'department_id,' 'department.'
Importance: Streamlines store organization and customer navigation.

orders.csv (3,421,083)

Identifies order sets and provides order-related information.
Key variables: 'order_id,' 'user_id,' 'eval_set,' 'order_number.'
Importance: Organizes orders and provides essential insights into user behavior.

order_products__*.csv (train:1,384,617 prior:32,434,489)

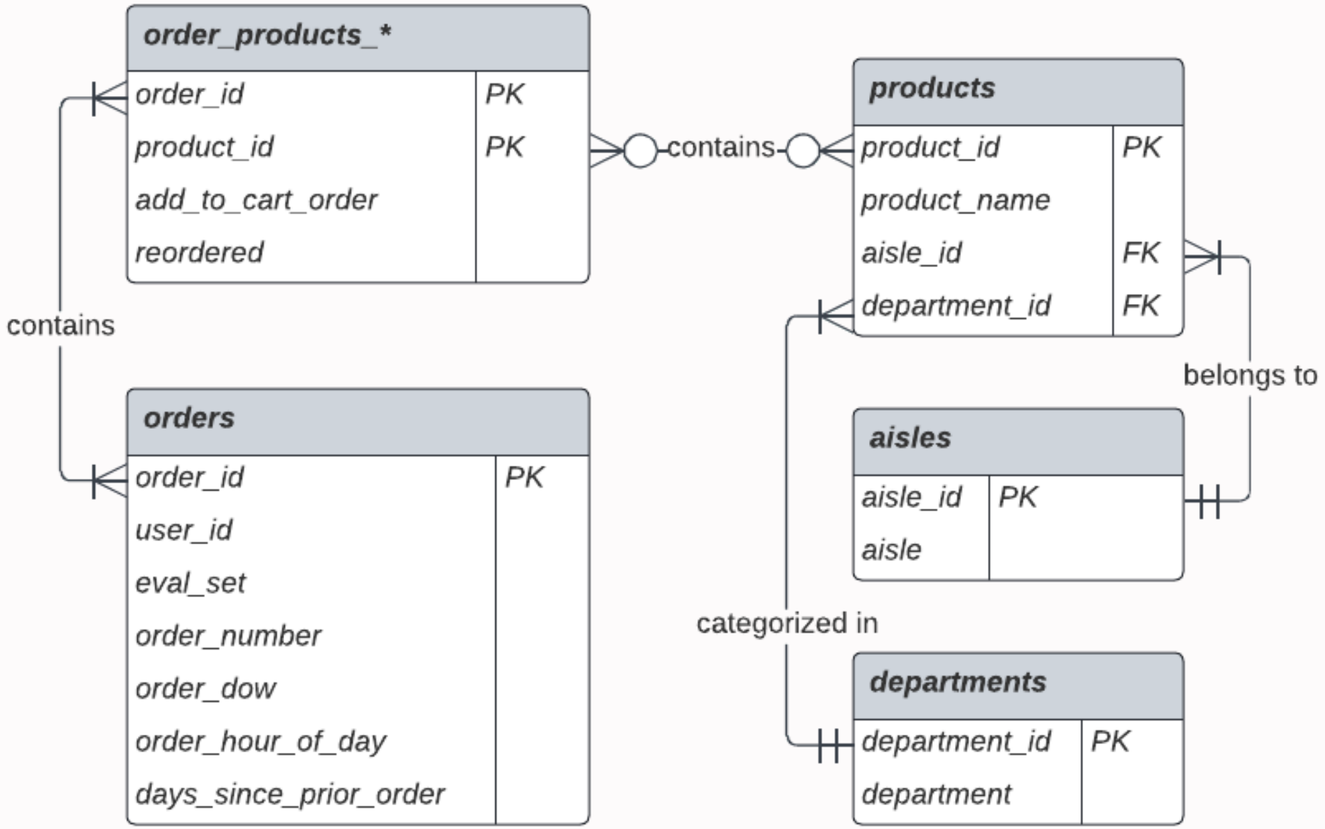
Specifies products in each order and reordering status.
Key variables: 'order_id,' 'product_id,' 'reordered.'
Importance: Critical for predicting reordering behavior, a key component of recommendations.

products.csv (49,688)

Describes products in each order, including aisle and department information.
Key variables: 'product_id,' 'product_name,' 'aisle_id,' 'department_id.'
Importance: Defines product characteristics and supports effective recommendations.

Datasets ERD

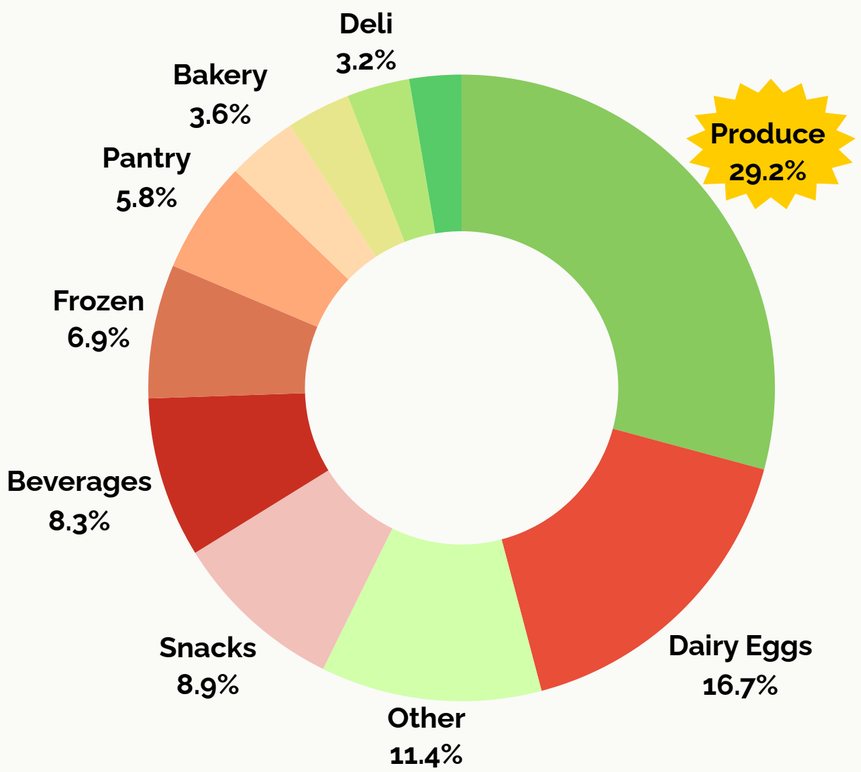
This illustrates the project's data structure, showing interconnections between various entities. It's a key tool for efficient data querying, manipulation, and analysis for the project.



Exploratory Data Analysis

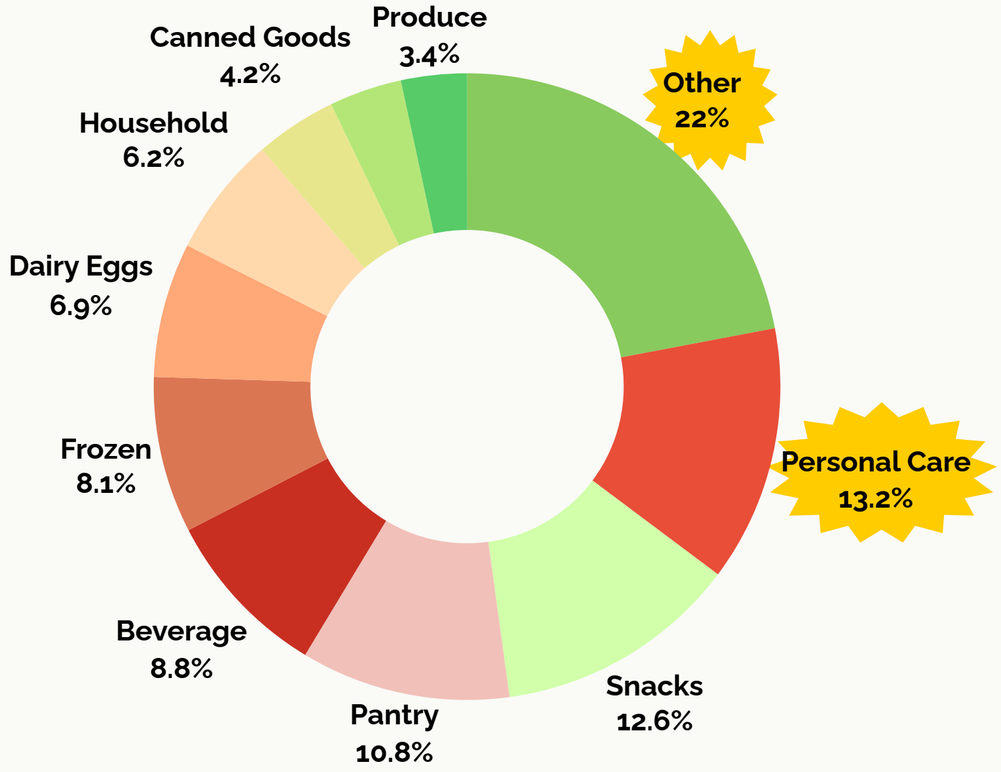
We want to understand underlying patterns in customer purchasing behavior, identify key predictors, and inform feature engineering for predictive modeling.

Total Purchased Products by Department



Across departments, **Produce** stands out as the top one purchased category, suggesting a strong demand for produce items among customers.

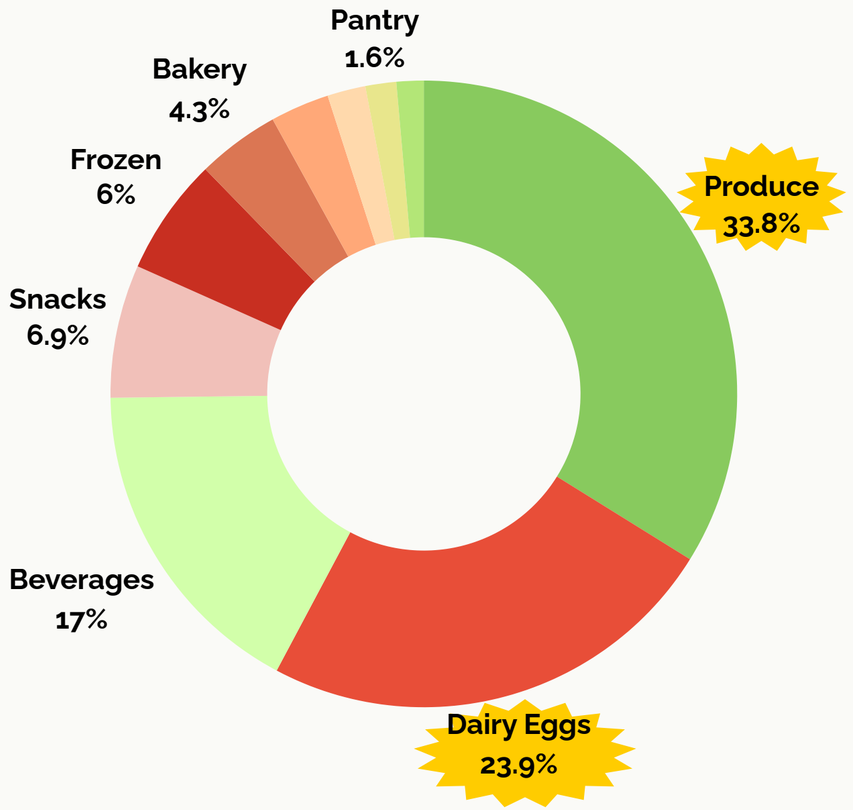
Products by Department



Other and Personal Care departments stand out for their extensive range of products, suggesting a wide array of choices catering to diverse consumer needs.

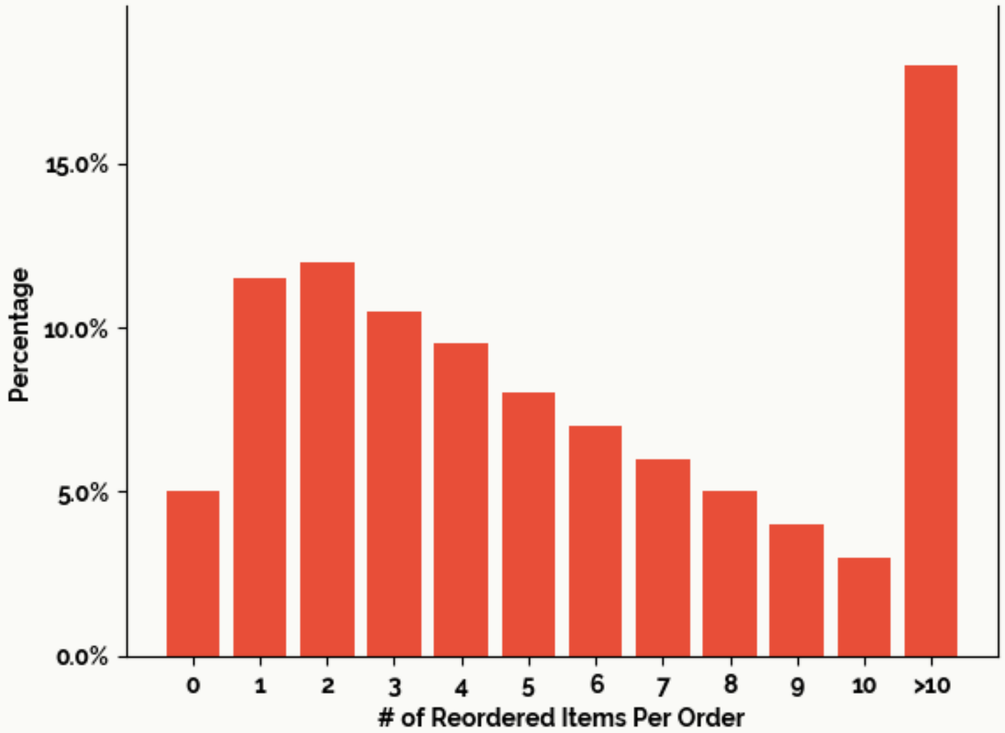
Sales Product Category Analysis

The Most Popular Reordered Department



Produce and **Dairy Eggs** departments exhibit exceptionally high reorder rates, emphasizing ongoing demand for fresh essentials.

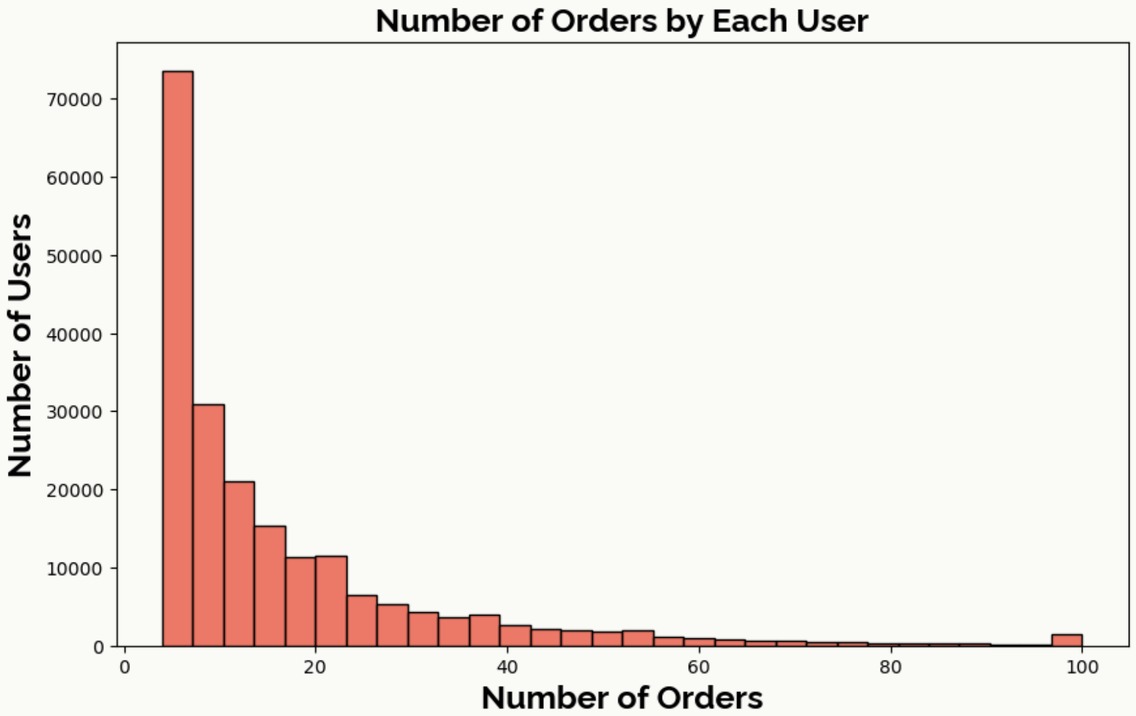
Reordered Items Distribution Per Order

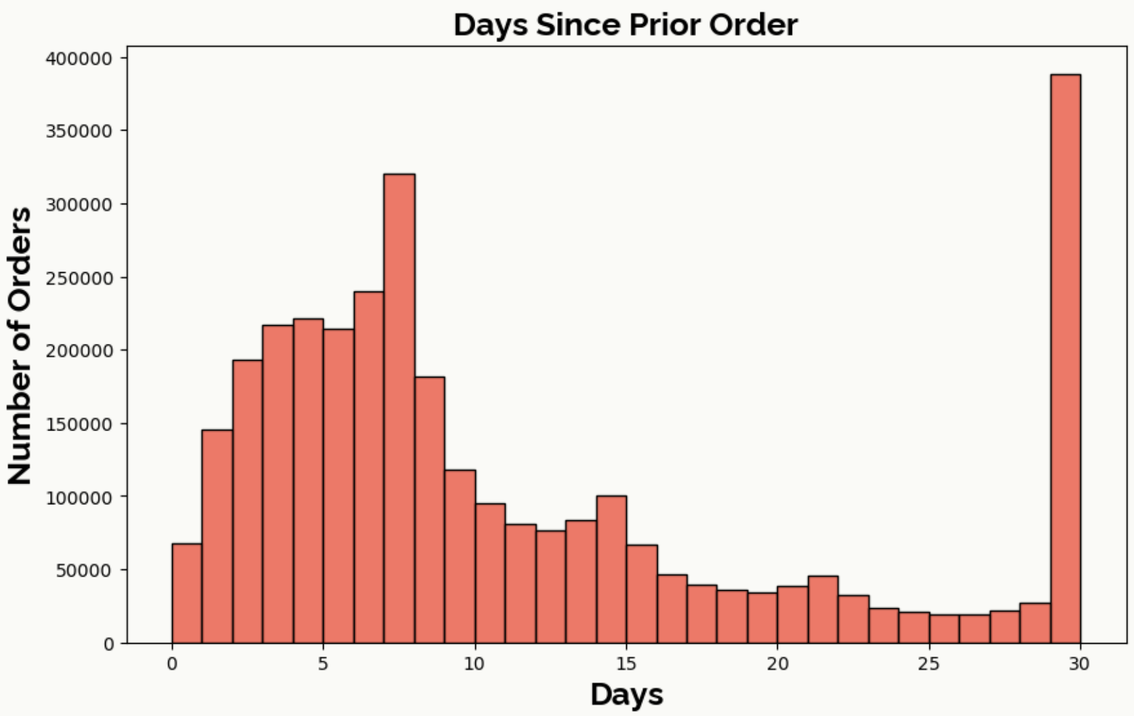


A significant portion of customers (**15%**) tend to reorder 10+ products, suggesting a strong brand loyalty or satisfaction with Instacart's service.

Customer Reorder Analysis

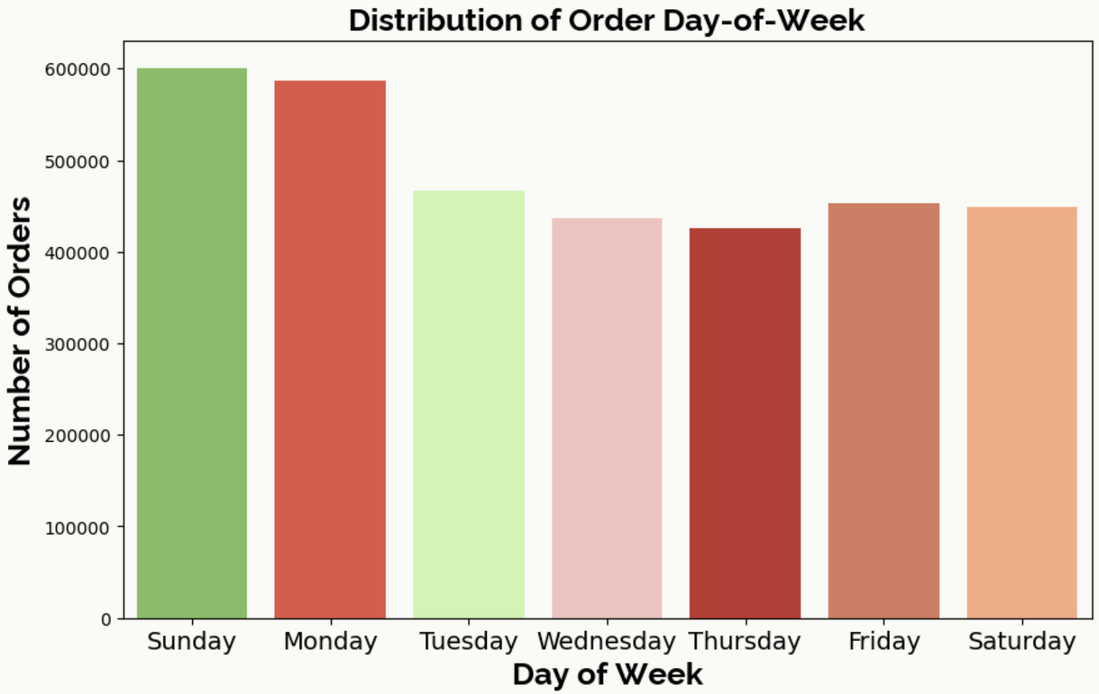
- The majority of users only place a single order, yet a small segment of users exhibits extremely high loyalty (>100 orders).
- Two peak reorder days are 7 days (weekly replenishment) and 30 days (monthly replenishment) from the prior order.

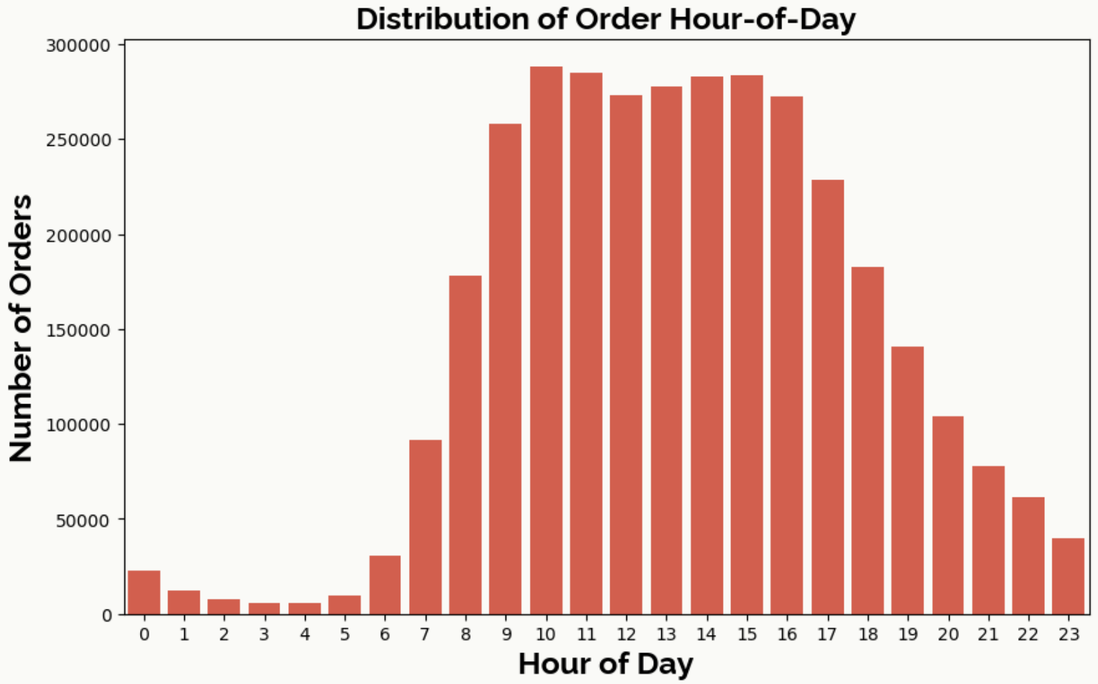




Customer Purchase Behavior Analysis

- Order day-of-week shows an even distribution pattern, with slightly more orders placed on Sunday and Monday.
- Most orders were placed during 9 AM to 4 PM of the day.

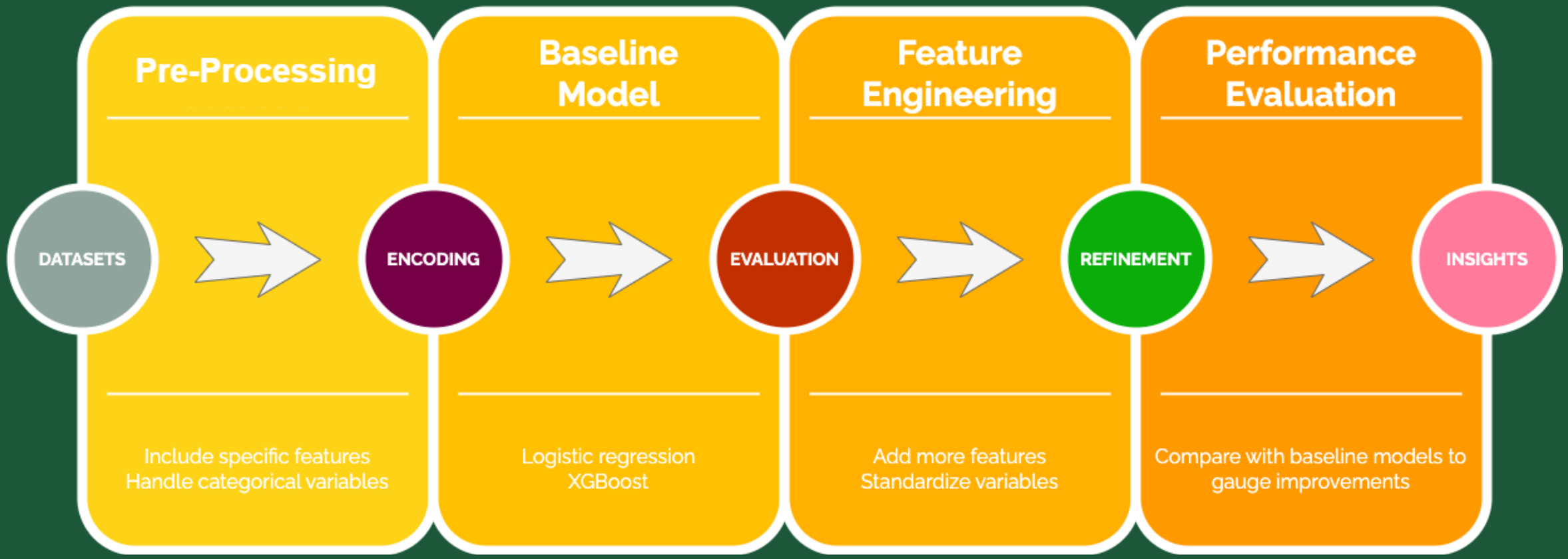




Customer Purchase Behavior Analysis


Will customers buy products again?

Machine Learning Pipeline



Data Preprocessing

Data Setting

	Order_number									
	1	2	3	4	5	6	7	8	9	10
User A	p	p	p	p	p	tr				
User B	p	p	p	p	p	p	p	p	tr	
User C	p	p	p	p	p	p	te			
User D	p	p	p	p	p	p	p	p	p	tr

Strategy

Use the last order as the reference `tr` , and include all the prior purchased items `p`

- If the product from the last order is repurchased, then “reordered” is 1. Otherwise, it is set to 0.
- Train-test Split (80-20) based on user_id

Final Dataset

Total 8,474,661 rows

Unique
131,209 users

Unique
92,148 products

Feature Selection - Baseline v.s. Final

Baseline model

Order Level	order_dow
Order Level	order_hour_of_day
Order-Product	days_since_prior_order
Product Level	department_id

Reduce underfitting
+
Add more complexity

Final model

Baseline model features

aisle_id	Product Level
reorder rate	User-Order
scale_since_prior_order	User-order

User-specific reorder rates

=Item Appearance/Total Orders

Jamie bought apples in 3 previous orders and she has had 10 orders so far. The reorder rate for that product (apple) is 0.3

Scaled day since prior order

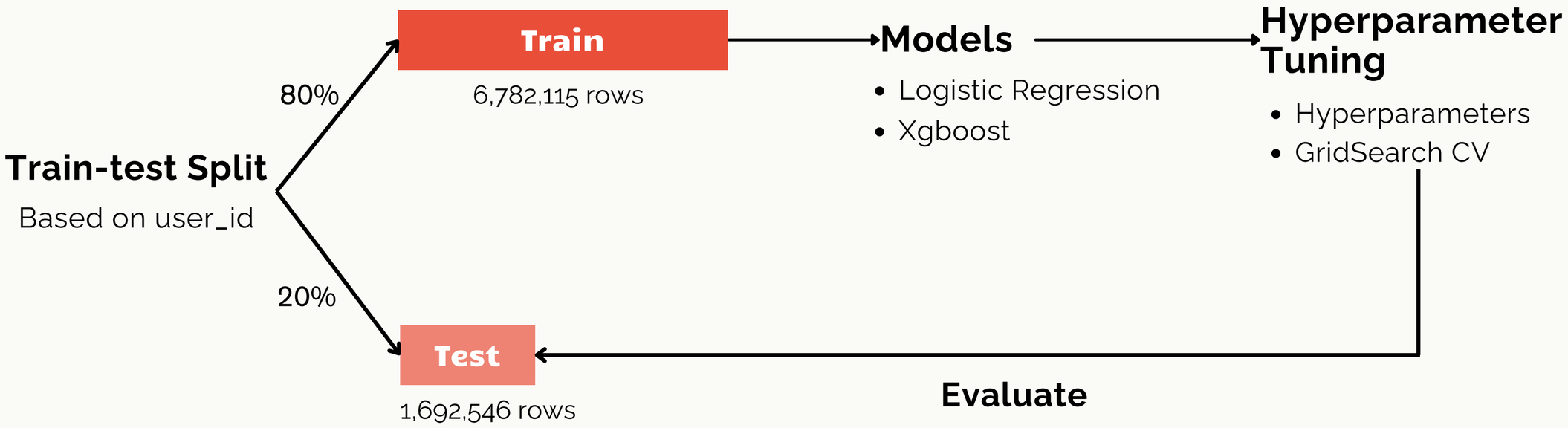
=(Order Gaps - mean) /std. dev

Order Number	Days Since Prior Order	Scaled Days Since Prior Order
1	10	0.92
2	7	-1.38
3	8	-0.62
4	9	0.15
5	10	0.92

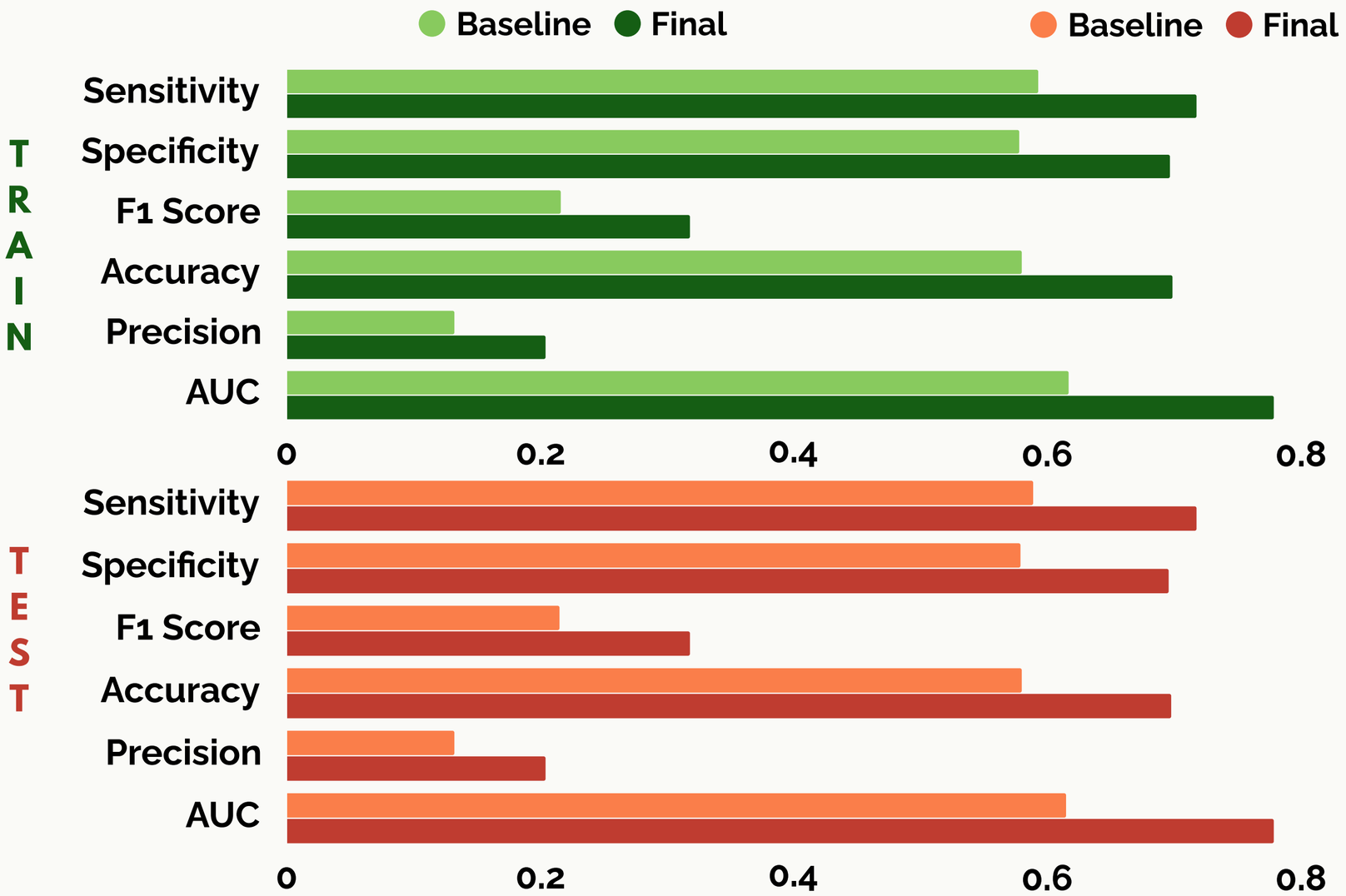
A positive value signifies that an order was anticipated but not made, while a negative value suggests an order is likely to be placed soon.

Feature Engineering Formulas

Model Training Process



Model Improvement: Logistic Regression

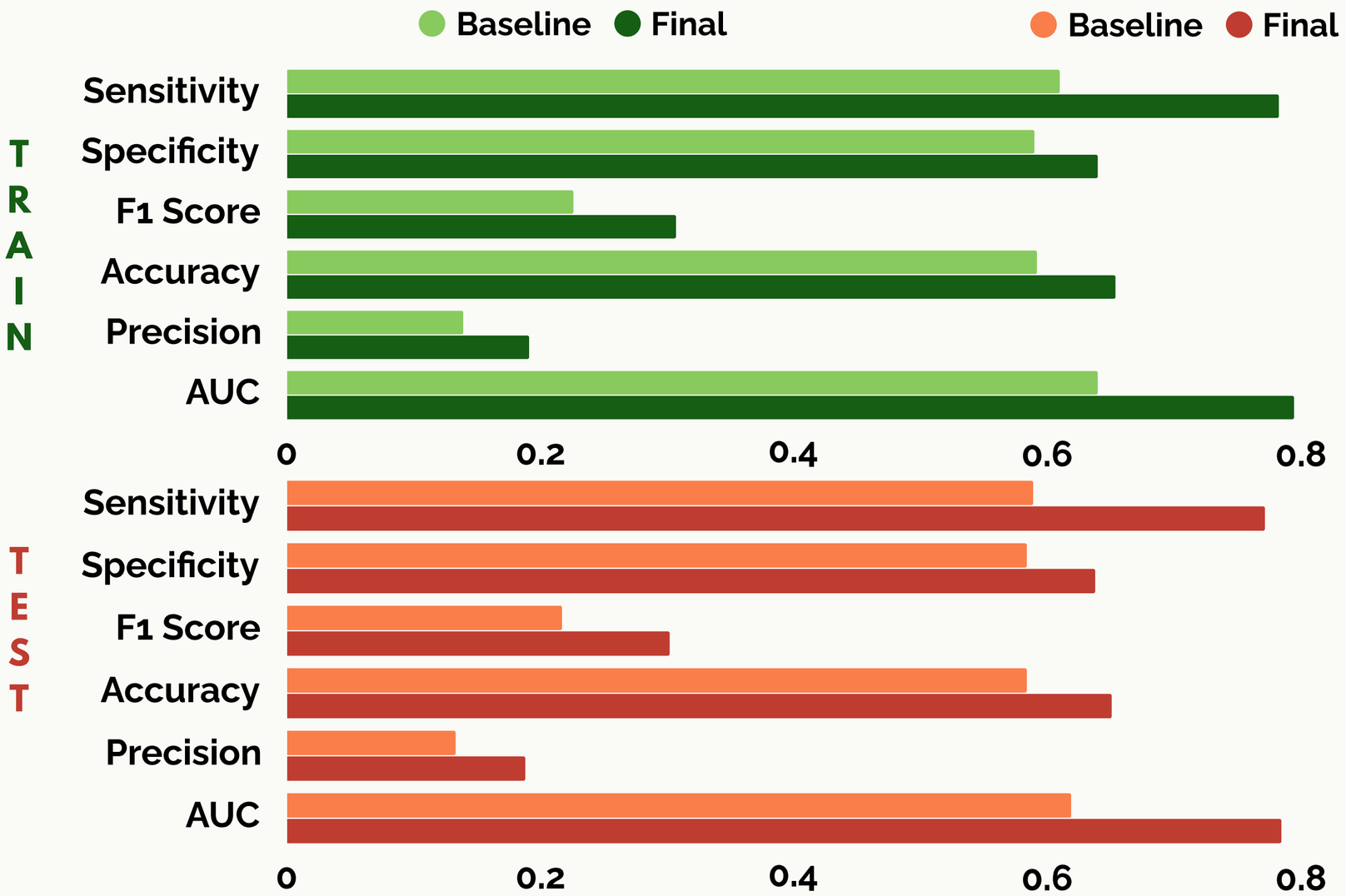


Takeaways

We can compare the baseline and final models across these metrics. Overall, we achieve around 70% in sensitivity for the final model. Sensitivity refers to the capture rate of the reordered cases. In our context, the model can identify 70% of the reordered items. Which is not bad in terms of model performance.

We can also see that by adding more user specific information, the final model can increase 12% in sensitivity, 12% in specificity, 10% of F-1 score, 10% of accuracy, 8% of precision, and 16% of AUC score.

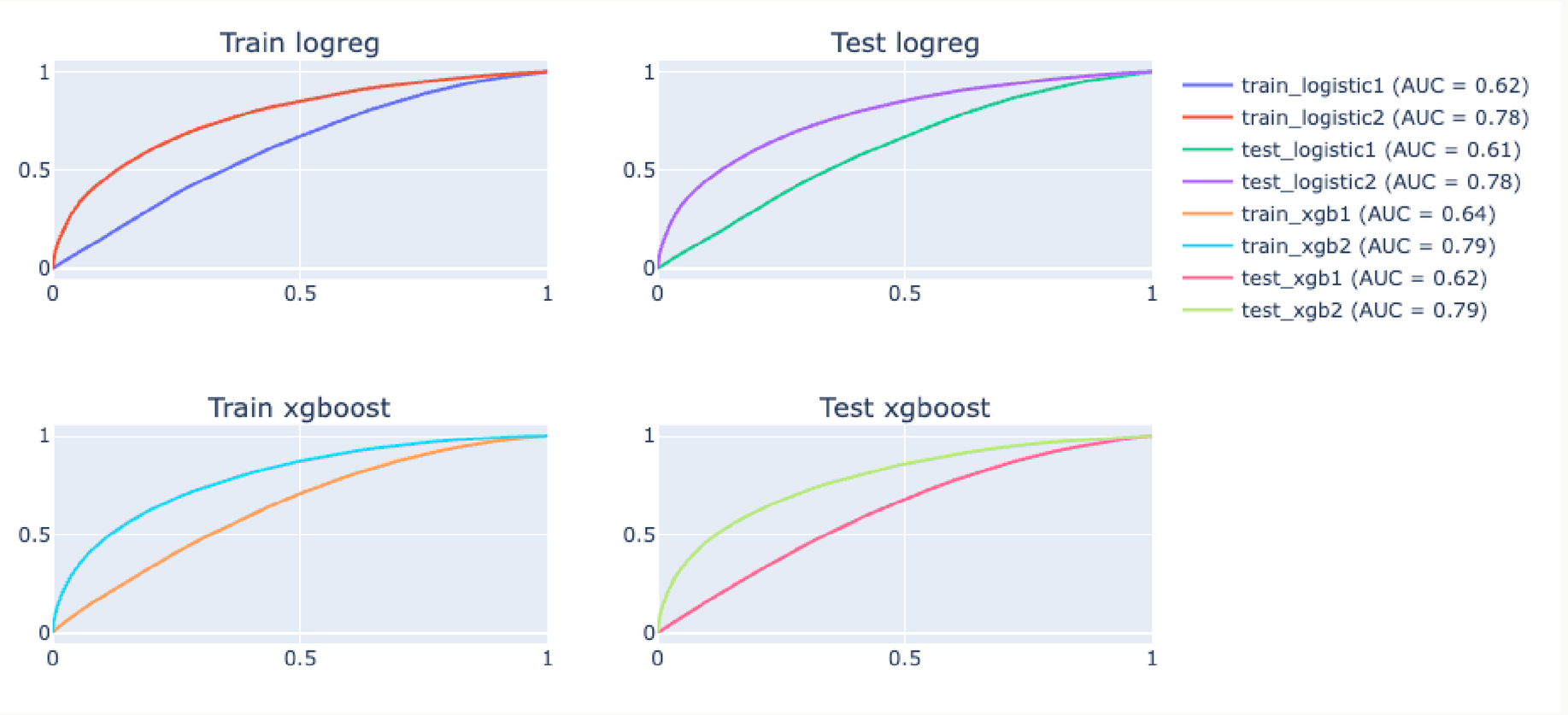
Model Improvement: XGBoost



Takeaways

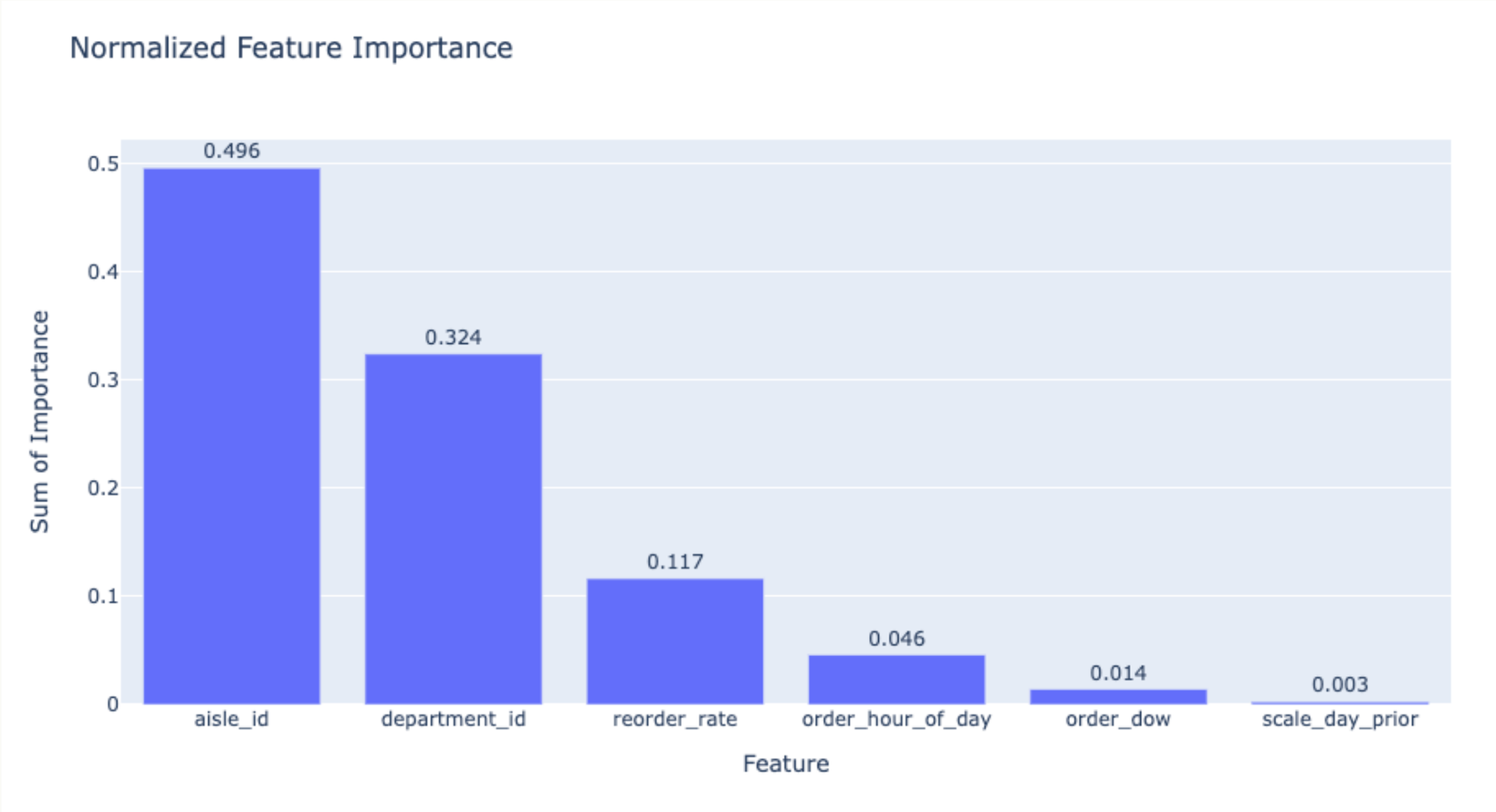
For XGBoost, we can see that it performed better than Logistic Regression in terms of sensitivity and AUC. XGBoost increased the sensitivity by 17%, which is a larger improvement than Logistic Regression. Also the AUC is about 78%.

Model Improvement (AUC)



AUC score increases → Great improvement in the model

Final XGB Model Feature Importance



The aisle_id and departments_id are crucial for predictions, while reorder_rate greatly influences predictions on its own

Takeaways

We can see that ‘aisle_id’ has the highest importance score of about 50%, followed by ‘department_id’. This sounds intuitive, but the aisle_id and department_id are categorical variables. The reorder_rate column seems to greatly influence its predictions on its own.

To put it simply, the aisle_id is a categorical variable that represents over 112 different aisles from one hot encoding process, so the sum of these features is around 50%. Each specific aisle_id variable weighs only about 0.4%. The reorder_rate alone weighs over 11%.