



Machine Learning Models Predicting Future Purchase

Terrance Xia

Agenda

- Introduction
- Project Goal & Data Description
- Modeling Process
- Business Implementation & Next Step
- Q&A

Instacart Business Model



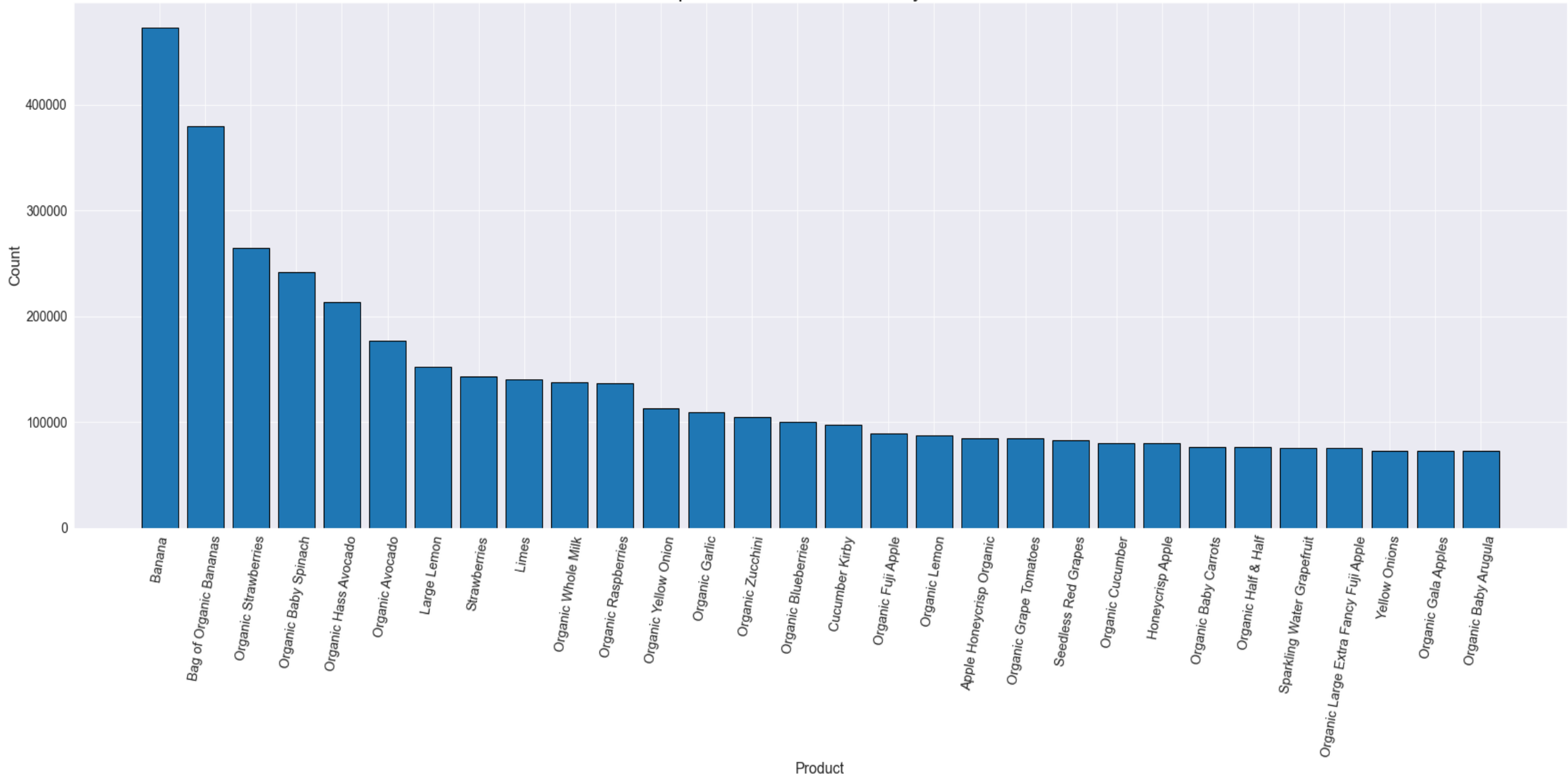
Project Incentive

- Predict which product will the user order based on past purchasing behaviors
- Remind people to add predicted orders to their shopping cart
- Increase the overall items in their shopping cart to increase the revenue for stores

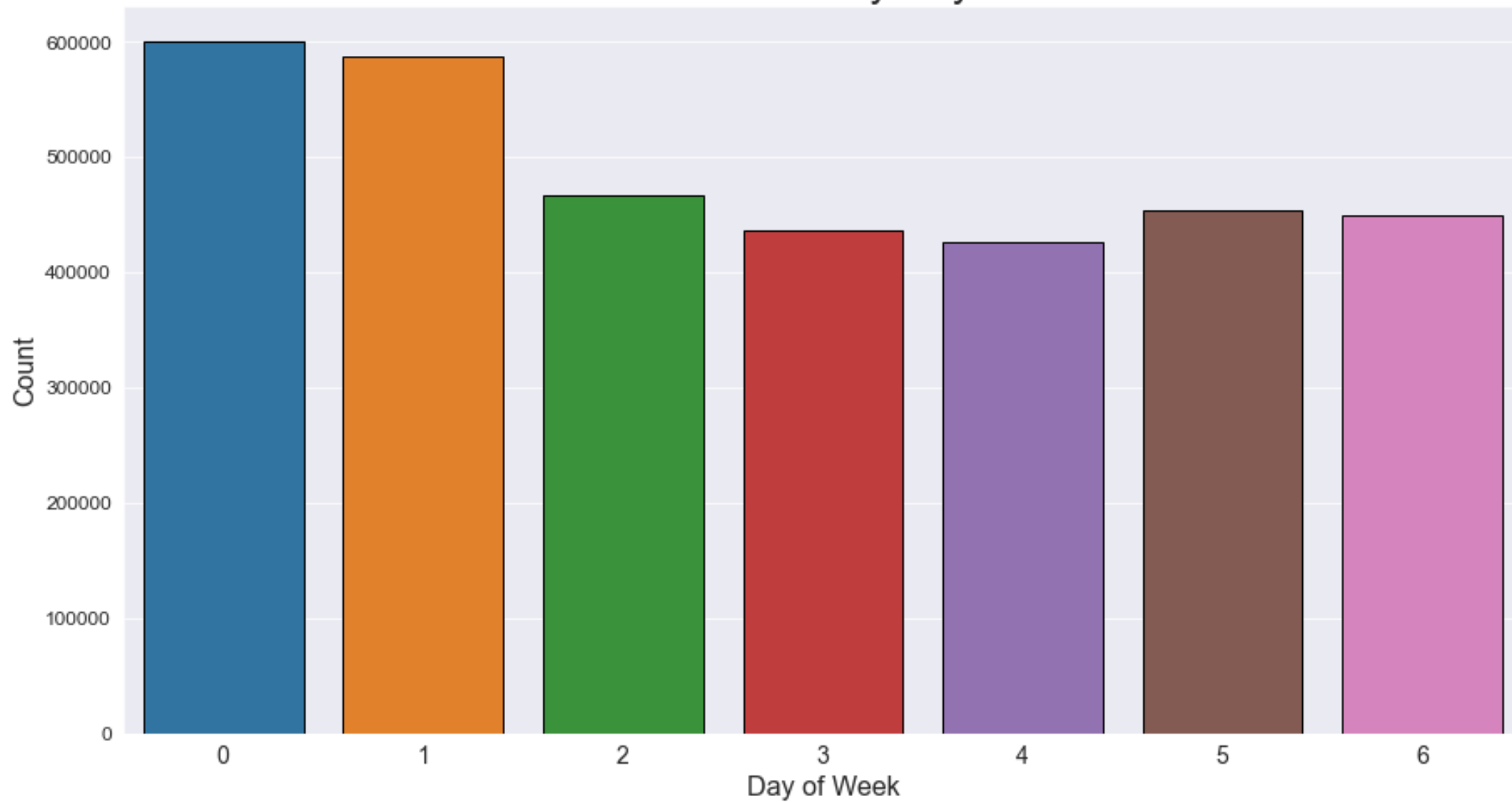
Data Overview

Tables	Size	Brief Description
Prior	32M obs.	Orders prior to that users most recent order
Train	1.3M obs.	One of partitions of the total orders
Order	3M obs.	Details of each orders includes (DOW, hr of a day, etc.)
Product	49K obs.	Details of each product
Department	21 obs.	The department info. related to products
Aisles	134 obs.	The department info. related to products

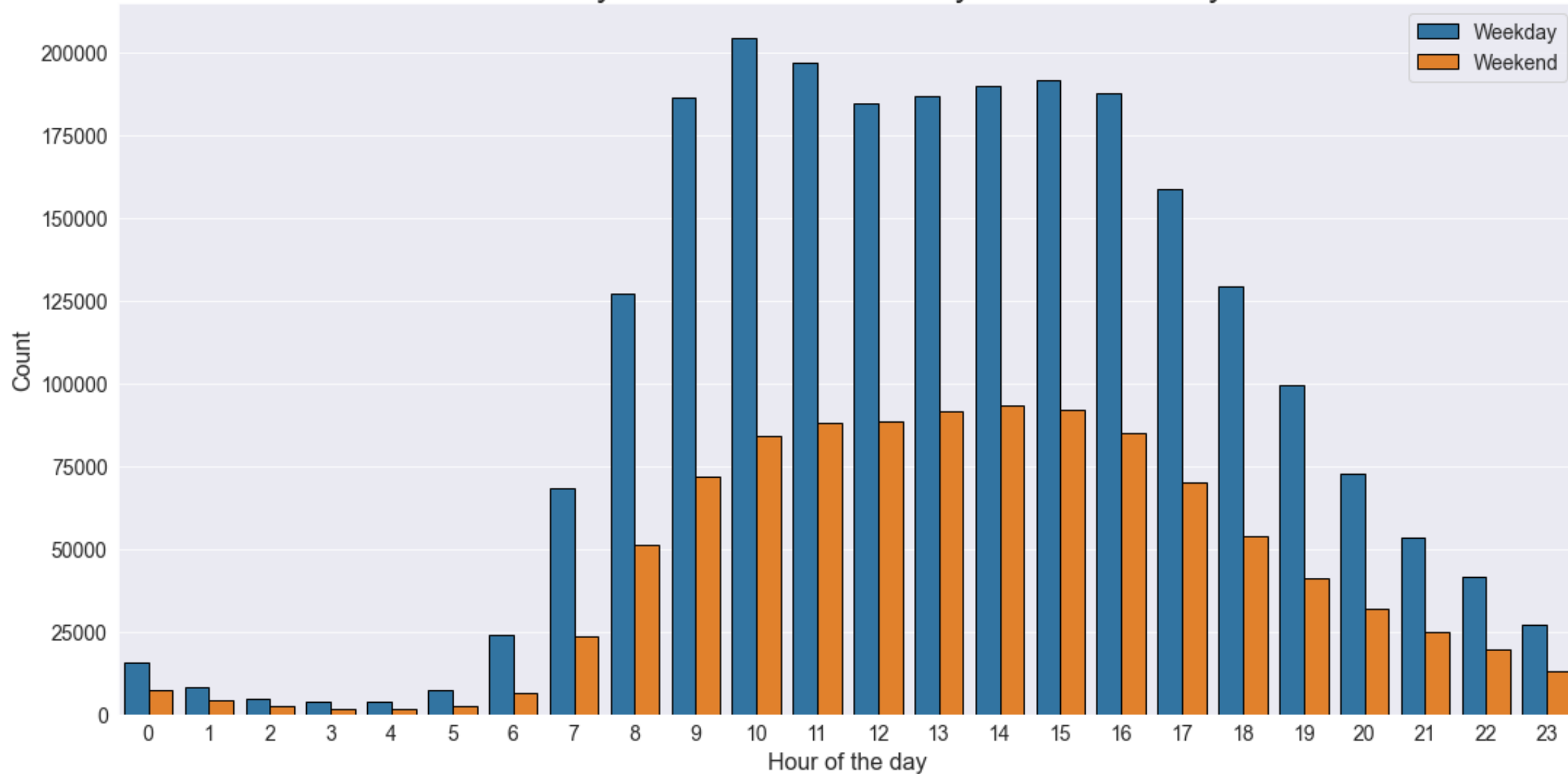
Top 30 Products Ordered by Users



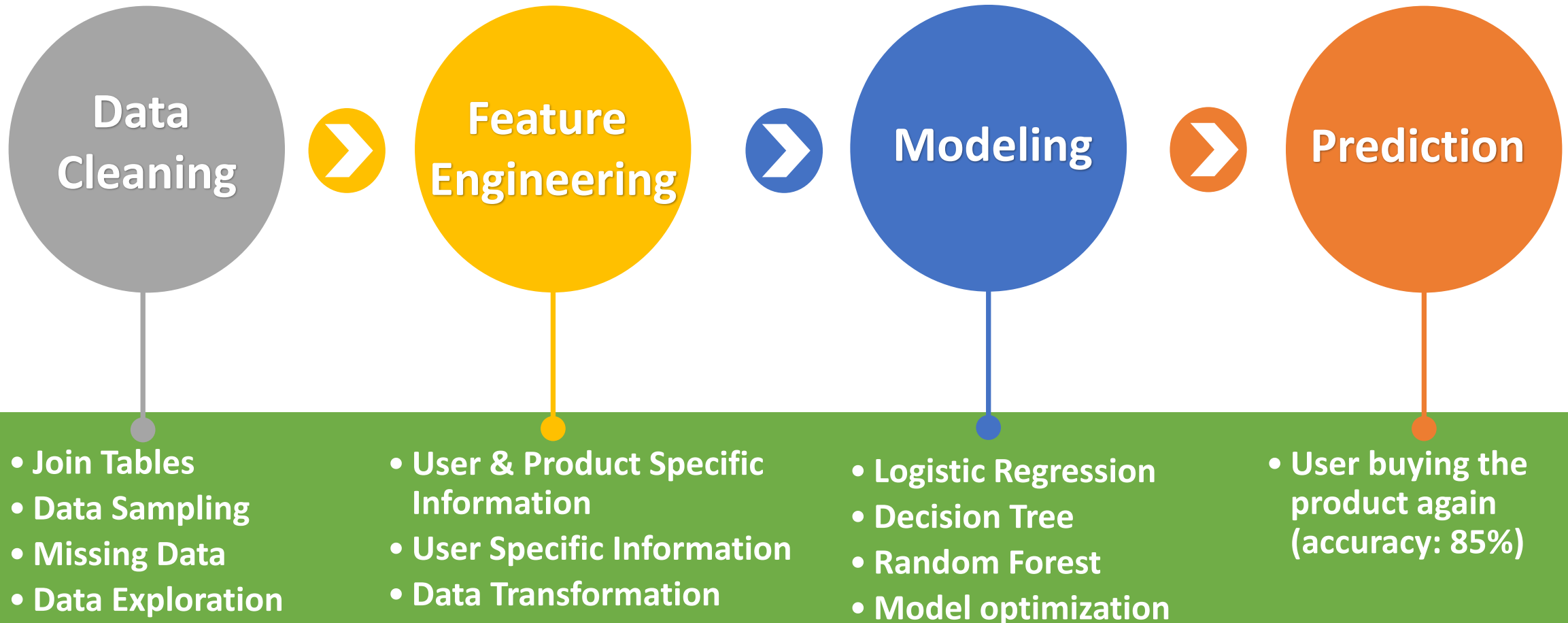
Number of Orders by Day of Week



Weekday VS Weekend Order by Hour of the Day



Modeling Process

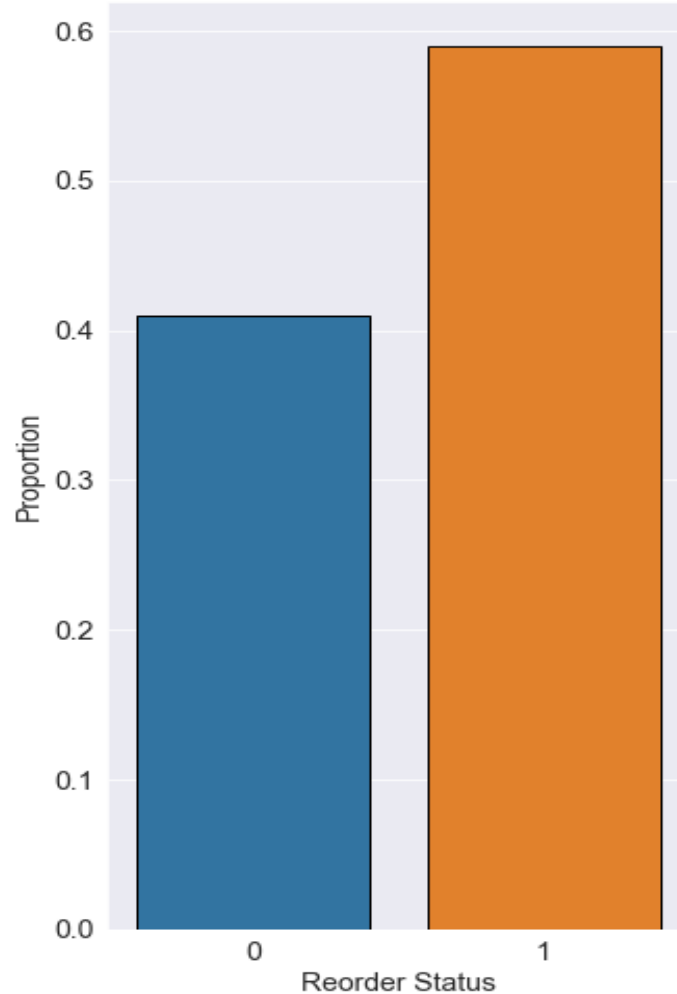


Modeling Process – Data Cleaning

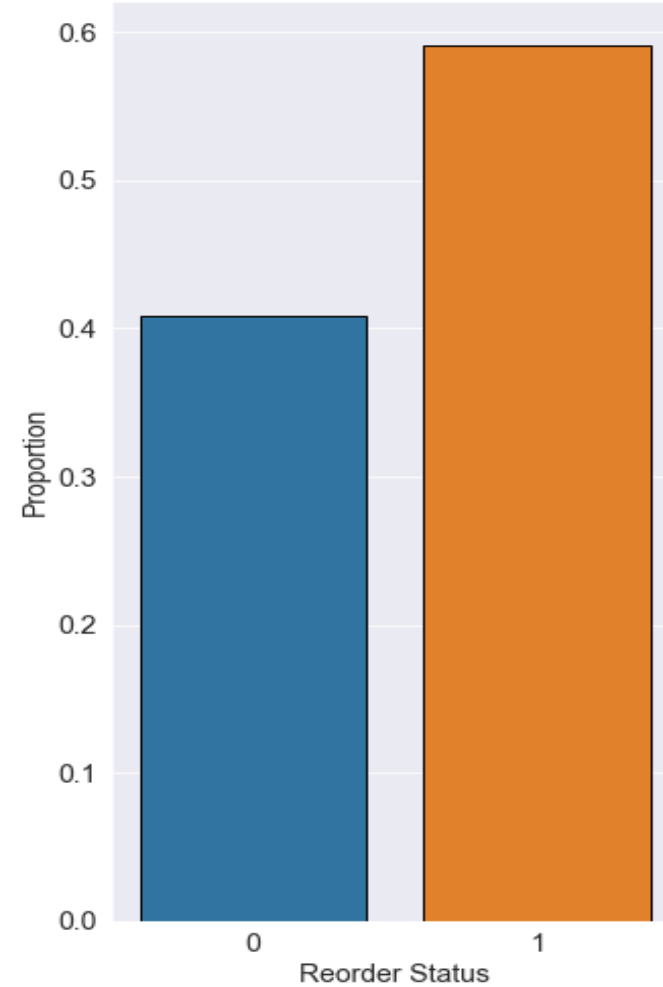
- Join Relational Data
 - 16 features
 - 32M of Transactional Data
- Data Sampling
 - 5% of Original Data
 - 10,000 Users / 1.5M of Transactional Data
- Data Manipulation
- Missing Data
 - Fill in with 0 for days since prior order if they are missing

Modeling Process – Data Cleaning

Reorder Status Proportion in Original Data



Reorder Status Proportion in Sample Data



Modeling Process – Feature Engineering

- User & Product Specific Features
 - Reordered ratio for each product of each user
 - Most frequent day of week for product placed by user
 - Most frequent hour of day for product placed by user
 - Overall reorder ratio for each product
- User Specific Features
 - Average products ordered by user
 - User reordered ratio
 - Most frequent order day of week for user
 - Most frequent order hour of week for user

Modeling Process – Feature Engineering

- Convert aisle_id to dummy variables
- Total 146 features
- Train / Test Split
 - 70% Train
 - 30% Test

Modeling Process – Modeling

- Logistic Regression
 - Optimize regularization parameter: C with cross validation
 - 78% accuracy rate on test set
- Decision Tree
 - Optimize hyperparameter max_depth, splitter with cross validation and GridSearchCV
 - 84.7% accuracy rate on test set
- Random Forest
 - Optimize hyperparameter: n_estimator, max_depth, min_samples_split with cross validation and GridSearchCV
 - 85% accuracy rate on test set

Modeling Process – Model Evaluation

Confusion Matrix for Logistic Regression

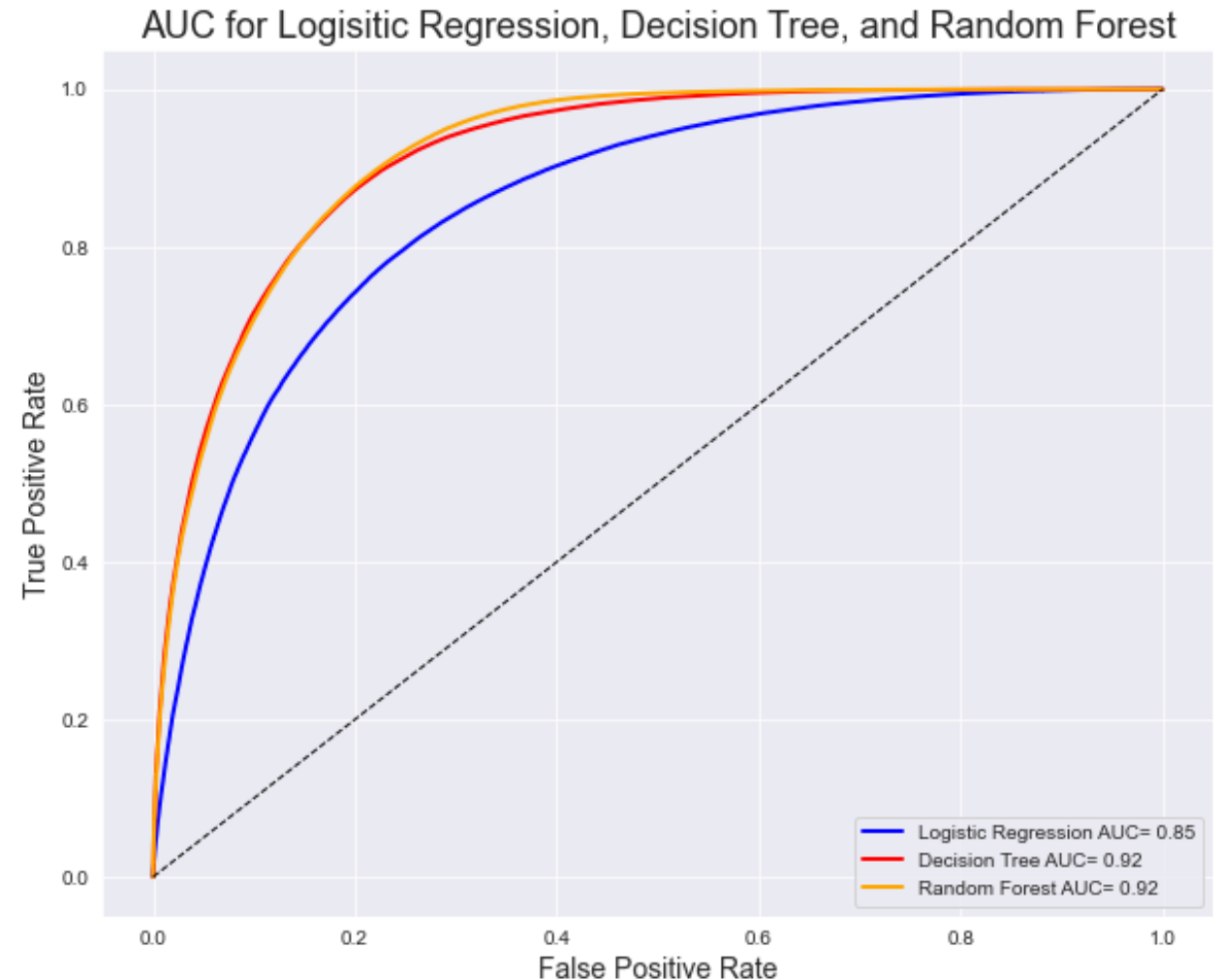
	Precision	Recall
0	0.75	0.71
1	0.80	0.84

Confusion Matrix for Decision Tree

	Precision	Recall
0	0.86	0.75
1	0.84	0.92

Confusion Matrix for Random Forest

	Precision	Recall
0	0.85	0.77
1	0.85	0.91



Modeling Process – Final Model

➤ Decision Tree

- 84.7% accuracy rate on test set
- Better AUC score compared to Logistic Regression
- Precision and recall scores are fairly close
- Easier and faster to build compare to Random Forest

Model Application



- Model: Decision Tree
- Accuracy: 84.7%



- Display the products when users shop
- Remind users to order the predicted products
- Set up easy reorder icon
- Promotion for those products
- Inventory management

The Next Step

- Scaled up the model with all of the data
- Product
 - Precise category assignment for each product
 - Pricing information for each product
 - Price bundling
- Customer
 - Customer demographic information
 - Customer lifetime value
 - Target marketing

Thank you

Questions?