Predicting Behavior of Instacart Shoppers

Second Capstone Project for Springboard's
Data Science Career Track Program

Author: Zack Kneupper

Date: February 25, 2018

Outline

- What is Instacart?
- The Business Problem
- Potential Business Impact
- The Tech Stack
- The Dataset
- Feature Engineering
- Exploratory Data Analysis
- Machine Learning
- Relative Feature Importance
- Results

What is Instacart?

- What is Instacart?
  - Instacart is an app for on-demand grocery shopping with same-day delivery service.
  - Instacart uses a crowdsourced marketplace model, where:
    1. App users place their orders through the app,
    2. A locally crowdsourced "shopper" is notified of the order, goes to a nearby store, buys the groceries, and delivers them to the user.

- Instacart's revenue model
  - There are three ways that Instacart generates revenue:
    1. Delivery fees,
    2. Membership fees, and
    3. Mark-up on in-store prices.

The Business Problem

- The Problem:
  - We want to build a model to predict which grocery items each Instacart user will likely reorder based on the user's purchase history.
  - Framing the problem:
    - We frame it as a binary classification problem.
    - There is one sample for each user-product pair.
    - The target variable: a binary variable for whether or not the user reordered the product in their most recent order.

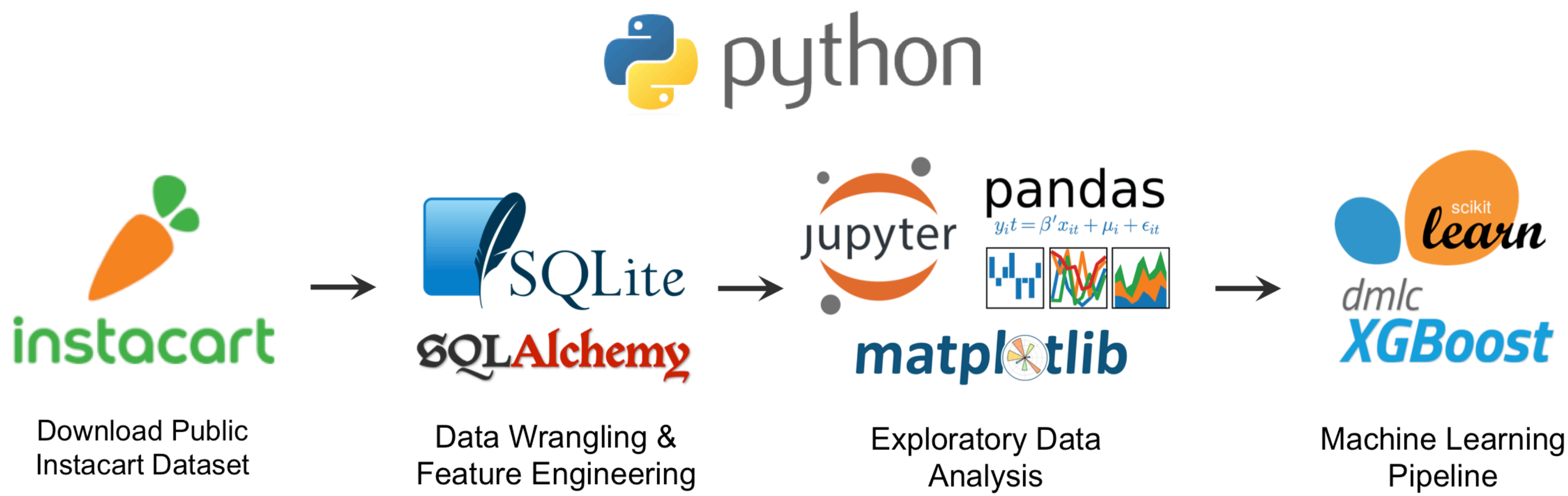
Springboard instacart

Potential Business Impact

- Business impact:
  - With the proposed predictive model, Instacart could provide its users with intelligently targeted purchase recommendations, discounts, or other promotions.
  - This could help Instacart:
    1. Improve the app's overall user experience,
    2. Retain current app users, and
    3. Increase the number of purchases made through the app, which could boost Instacart's revenues.

The Tech Stack

- The graphic below summarizes the main technologies and software libraries that we used at each step of the project.



The Dataset

- "The Instacart Online Grocery Shopping Dataset 2017"
  - A public dataset of anonymized Instacart grocery orders.
  - Contains data from:
    - more than 200,000 Instacart users, and
    - about 3.4 million individual orders.
  - Of the 3.4 million orders,
    - about 3.2 million are orders prior to that users most recent order, and
    - about 131 thousand are users' most recent orders.
  - The dataset contains six relational tables

Feature Engineering

- We needed to engineer a new data table from the existing tables.
- The index of the new table:
  - We created a list of unique pairs of users and products from the prior set of orders. We gave this list the label "up_pair" for "user-product pair". This list list was used as a unique index of our new data table.
- The target variable:
  - We created a binary target variable for whether or not a user reordered a product.
  - If a user bought a product in the prior set of orders and reordered the product in the train set of orders, then the target variable is assigned the value 1.
  - If a user bought a product in the prior set, but they didn't reorder the product in the train set, then the target variable takes the value 0.

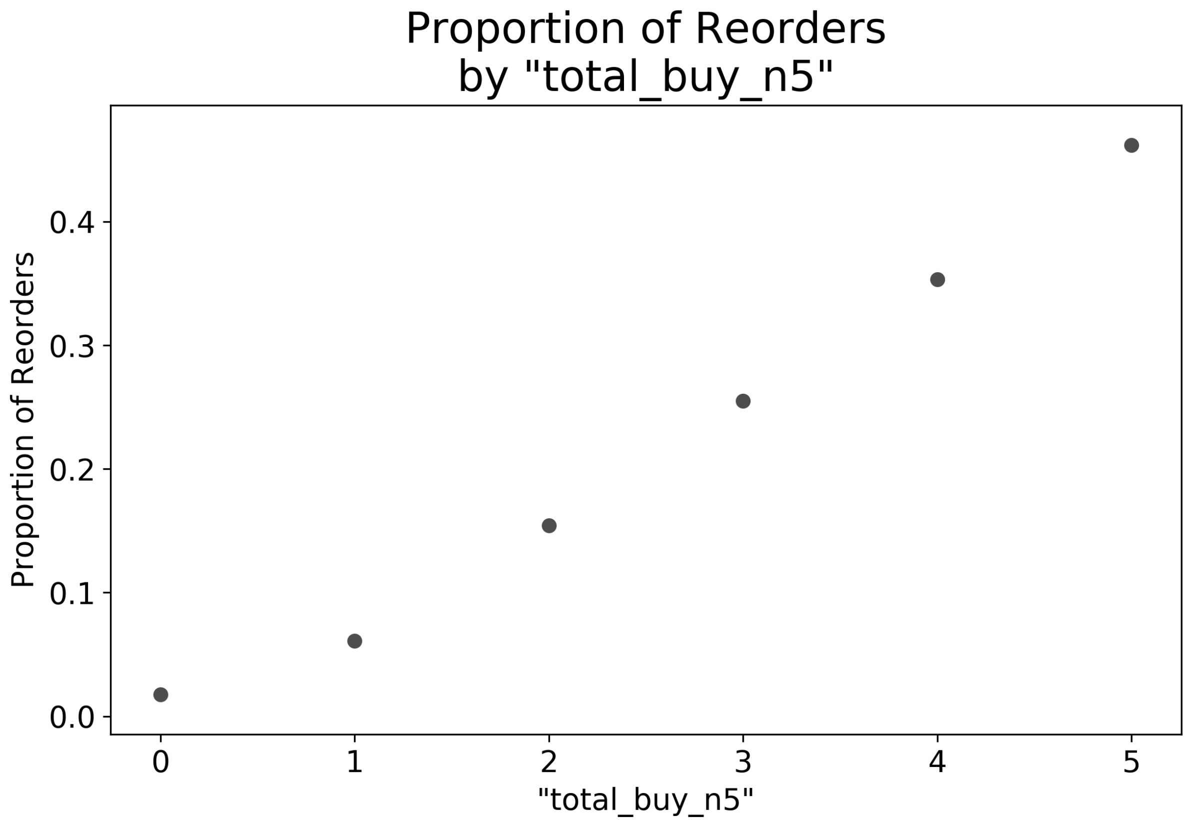
Feature Engineering

- We engineered four explanatory features. These features are as follows:
- Given the user-product pair of (User A, Product B),
  1. total_buy_n5: the total number of times User A bought Product B out of the 5 most recent orders.
  2. order_ratio_by_chance_n5: the proportion of User A's 5 most recent orders in which User A had the "chance" to buy B, and did indeed do so. Here, a "chance" refers to the number of opportunities the user had for buying the item after first encountering (viz., buying) it.
  3. useritem_order_days_max_n5: the longest number of days that User A has recently gone without buying Product B. We are only considering the 5 most recent orders.
  4. useritem_order_days_min_n5: the shortest number of days that User A has recently gone without buying Product B. Again, we are only considering the 5 most recent orders.
- The choice of these four features was inspired by Onodera's solution, which won 2nd place in the Instacart Kaggle competition.

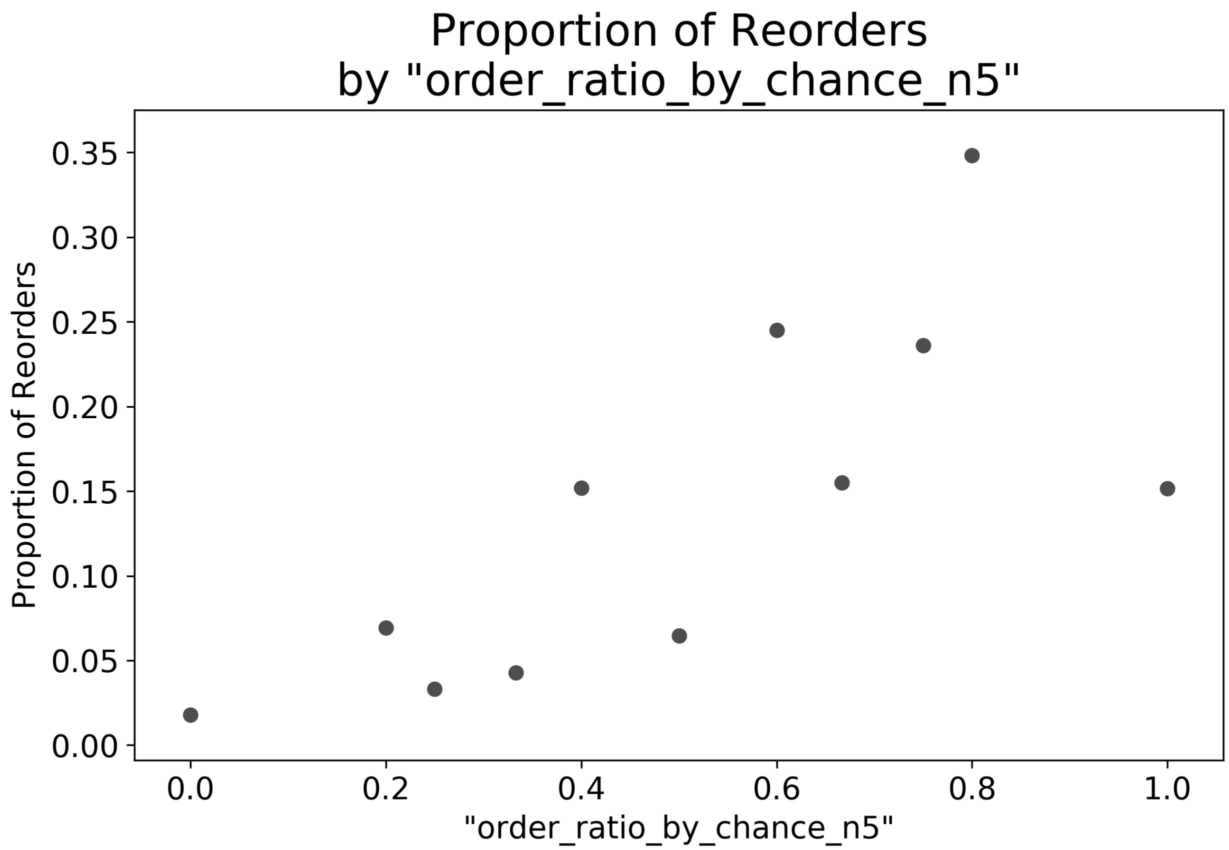
Exploratory Data Analysis

- Bivariate Visualizations
  - As a part of our EDA, we visualized how the proportion of reorders varies with each feature.
  - These visualizations can help us get an intuition about how the probability of reordering a product is affected by each of the feature variables.

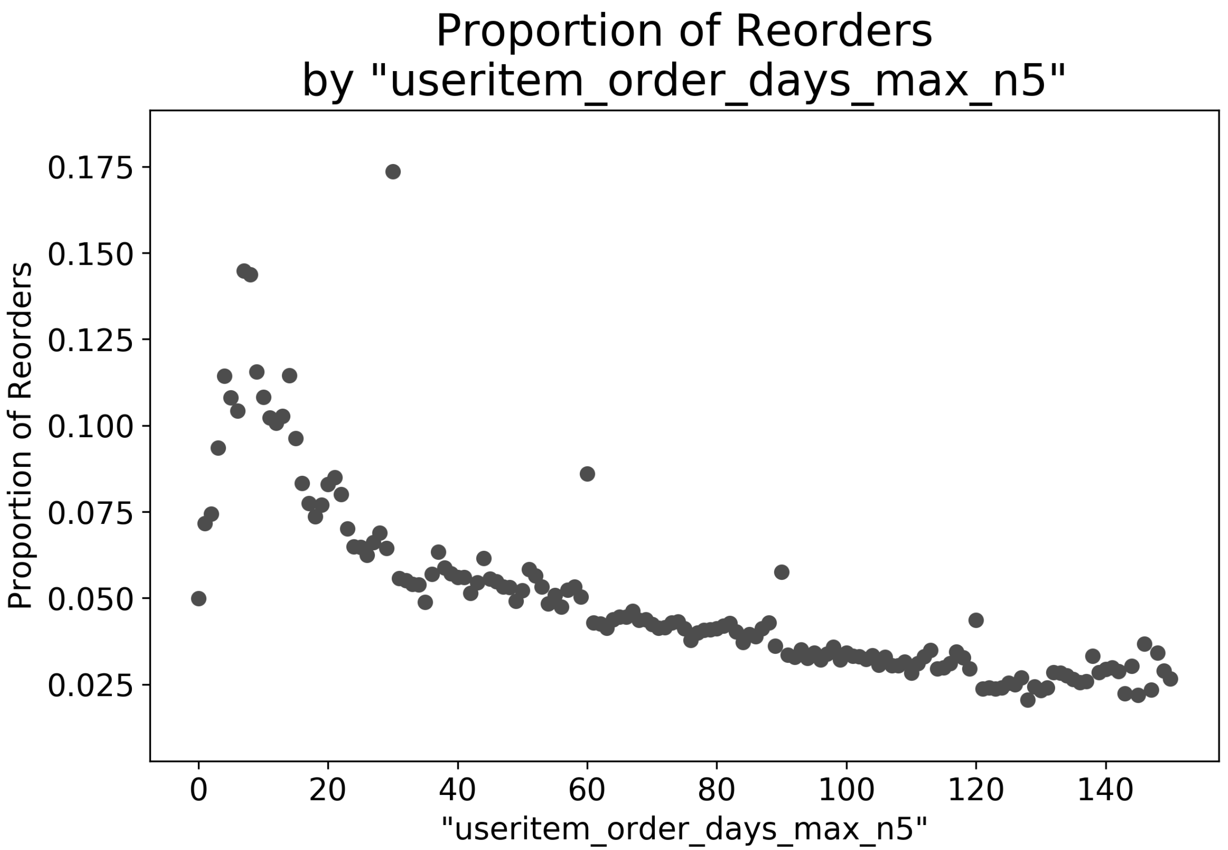
EDA: Bivariate Visualization 1



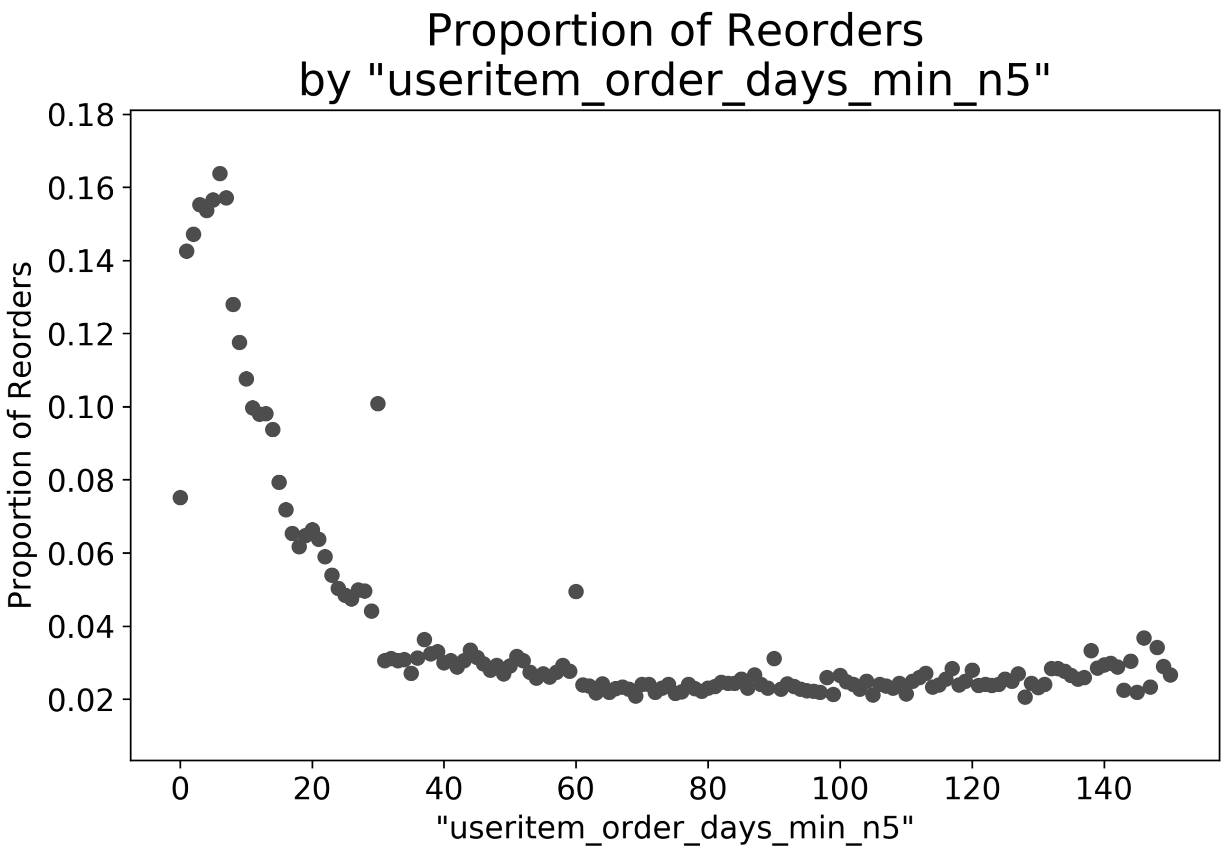
EDA: Bivariate Visualization 2



EDA: Bivariate Visualization 3



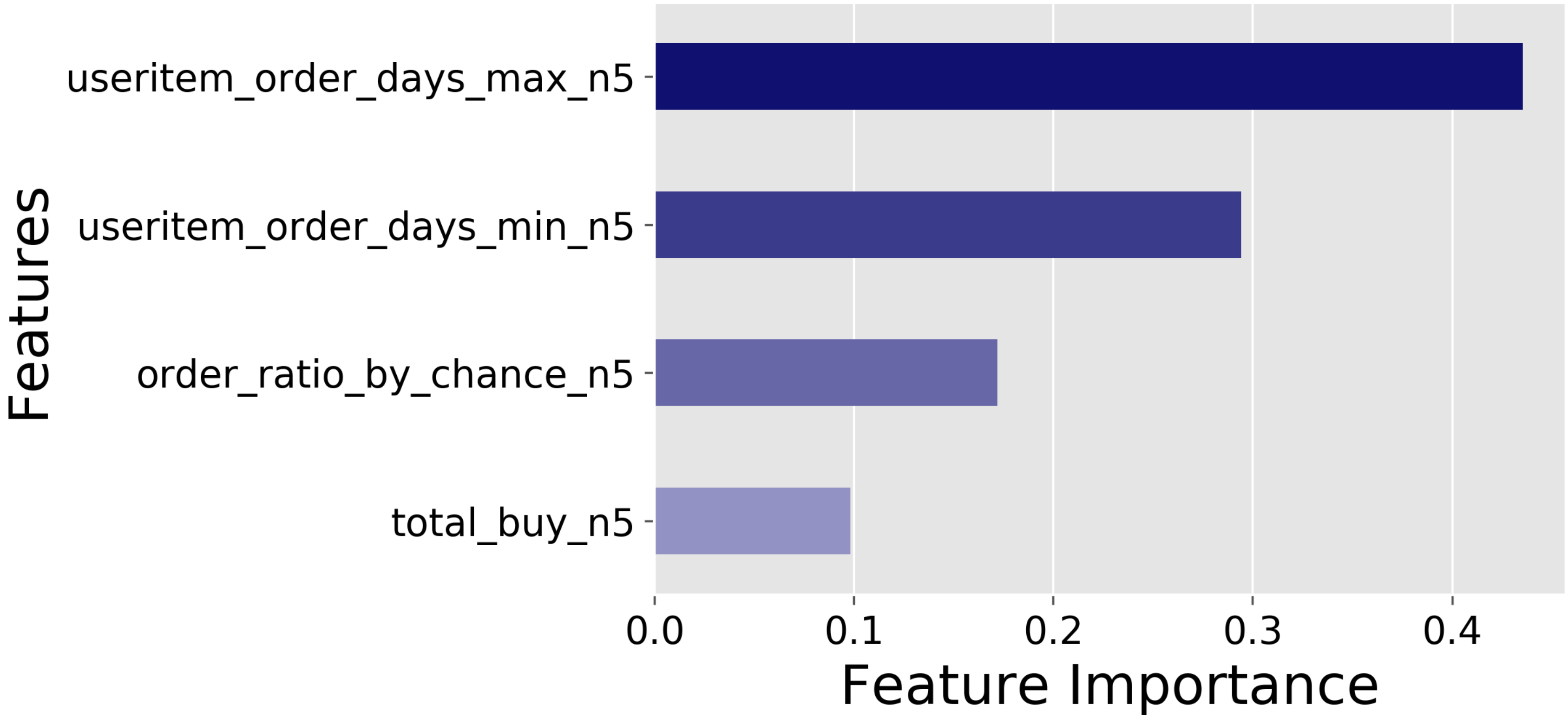
EDA: Bivariate Visualization 4



Machine Learning

- We built a ML pipeline and ran a grid search to find an optimal set of parameters.
- The pipeline has two steps:
  - 1. A sampler step to rebalance the training data
  - 2. A gradient boosting classifier
- We used the xgboost implementation of the gradient boosting classifier algorithm.
- Our grid search found that:
  - Random undersampling did not improve model performance.
  - Increasing the number of estimators in the gradient boosting classifier beyond 100 made no impact on model performance.

Relative Feature Importance



Springboard

instacart

Results

- Our performance metric:
  - Area under the receiver operating curve (aka, the ROC AUC score)
- The model achieved:
  - A mean cross-validated ROC AUC score of **0.79** on the train set
  - A ROC AUC score of **0.78** on the test set