Springboard Data Science Career Track

Capstone 2 Project

InstaCart Grocery App

Created By: Andrew Milne

InstaCart Data - Capstone 2 Milestone Report

Problem Definition

The kaggle competition behind this is to help InstaCart better predict items that customers will buy again. An additional component is to determine which items they might put in their cart relative to other items such as milk, bread, cheese however this was considered out of scope for the purposes of this capstone.

Source:
https://www.kaggle.com/c/instacart-market-basket-analysis

Client:

The client is the grocery getting app known as InstaCart. They are a luxury shopping app that specializes in fulfilling orders from various suppliers. For example, customers can buy products from a number of different vendors and InstaCart will retrieve them. In this way InstaCart is a multi storefront business with a big online presence.

Data Cleaning/Wrangling:

The data needed to be cleaned to remove the following:

- Removed items that had been purchased less than 40 times
- Removed items that had a 100% success rate of always being reordered
- Dictionary of weekday names to allow for legible plots
- Amalgamation of data frames to allow for modeling
- For purposes of competition, broke out the ‘train’ portion of the dataset

Feature engineering was performed to incorporate the following

- Conversion of timestamp data into day, week, month, hour, year, date
- Mapping of codified categorical values for interpretable plots
- Proportion reordered aggregate calculation

I used a variety of tool bags and libraries to assist in this analysis. These included the following:

- Python 3.6.1 run in Jupyter Notebook
- Plotting capability provided by matplotlib, seaborn, tableau
- Analysis capability from pandas, numpy, datetime
- Machine Learning provided by SciKit Learn: MiniBatchKMeans, Linear Regression, Decision Tree Regressor, Random Forest Regressor

Other Datasets:

- Within the competition a prior orders dataset was provided, allowed for determining proportion reordered

- Had timeseries data been provided, could have done analysis for seasonality of purchases
- Nutritional information would also be very valuable to understand healthy customer segments

Summary of Key Findings:

Through exploratory analysis and investigation we have determined the following:

- The busiest shopping days are Sunday and Monday
- People are shopping steadily throughout the core hours of the day 10am-4pm
- Surprisingly, people are shopping every 11 days rather than 1x a week
- Produce is the most popular even though it has far less variety than other departments
- Within the produce department people are buying fruits and vegetables, with the Banana leading the pack as the most purchased item
- 2% Lactose Free Milk is the most likely to be reordered

Exploratory Data Analysis & Inferential Statistics Discussion:

The following represents a few key features about the data.

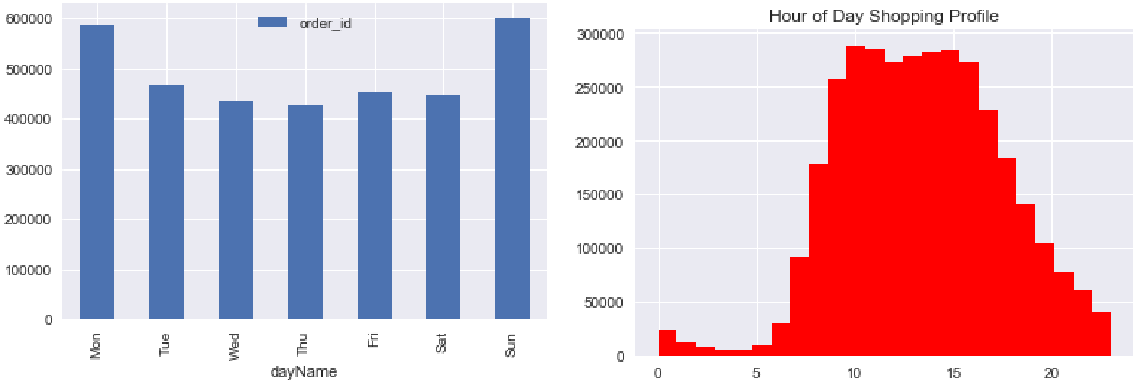


Fig. 1 - Shopping Profile

We investigate other features of the data for insightful plots:

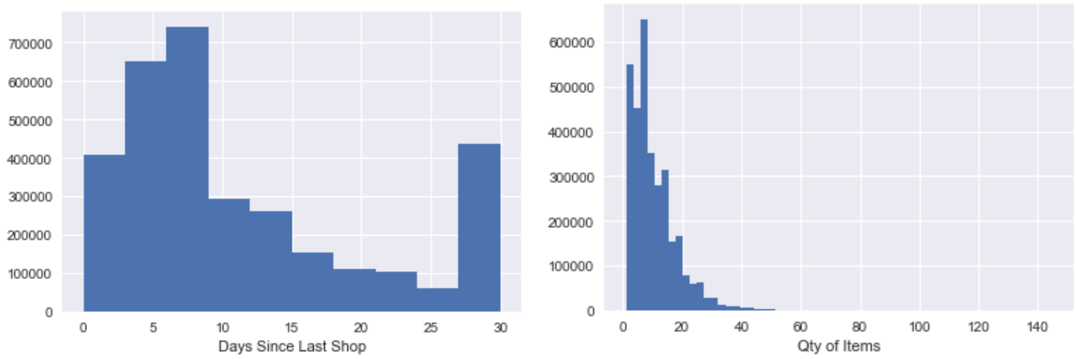
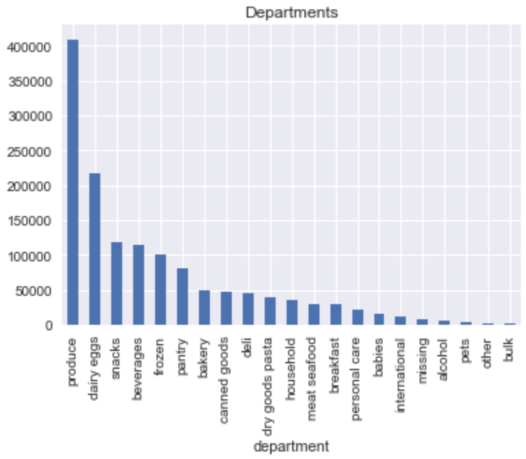
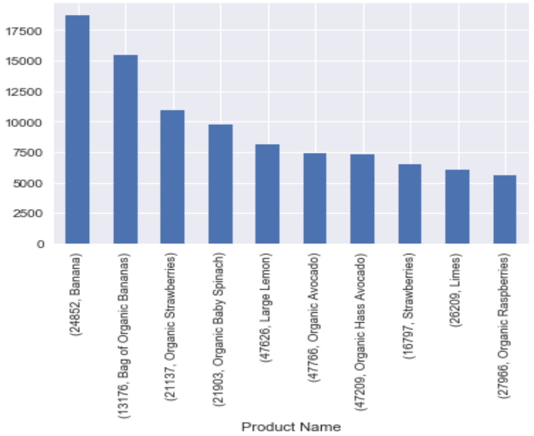


Fig. 2 - Customer Orders

Customers are typically shopping every 11 days and they buy 10 items on average each time.





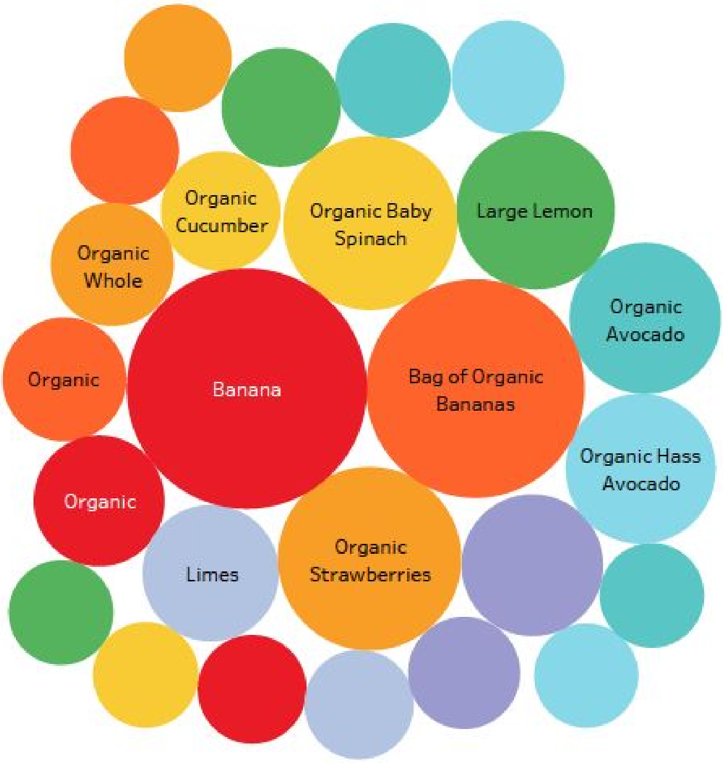


Fig. 3 - Most Popular Items

Here we see a breakdown of the most popular items on the app. Bananas are the clear favourite followed by strawberries, spinach, lemons and a host of other produce. I find it remarkable that apples don’t show up in the top 10 produce items. Also, if a customer were to purchase strictly these items, they would have a pretty imbalanced selection of food for the week, not to mention that bananas go bad so quickly.

What do people typically put in their cart first?

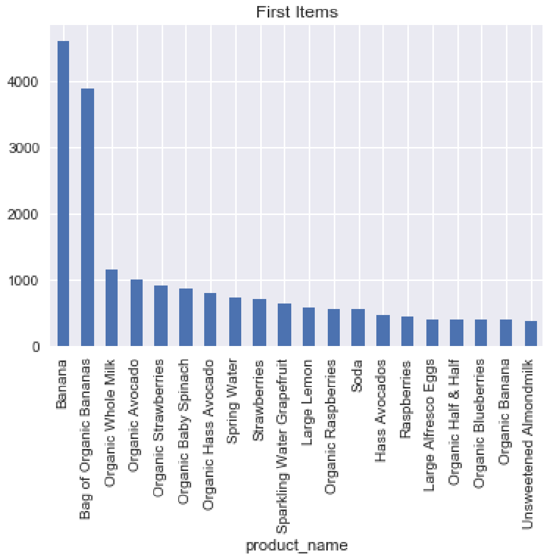


Fig. 4 - First Item in Cart

Unsurprisingly Bananas are first in the cart. However we also see some drinks and eggs make it into the first purchase group. Perhaps this is our first indication of a different customer segment.

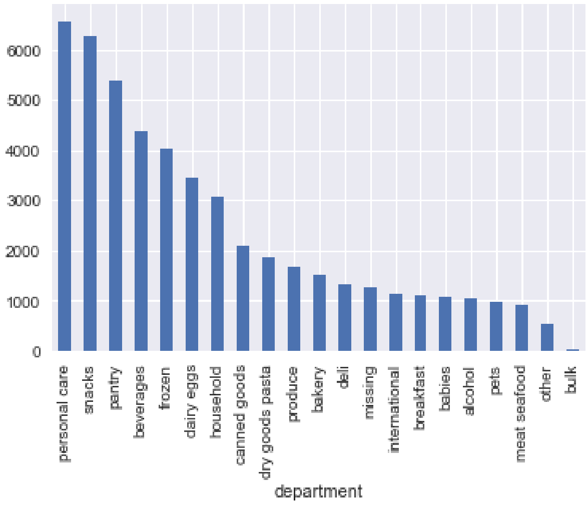


Fig. 5 - Department Variety

The bar chart above illustrates the stark reality of the 'grocery' store. It is evident that the sale of food is at best a parallel priority to the marketing of cosmetics, personal hygiene and fake eyelashes. While this may seem insurmountable produce surprisingly outpaces it's more diverse competitors by a wide margin, see Figure 3.

The data came with a field called 'reordered' which provided a binary classifier of 0 or 1 based on if the customer purchased the item again in another order. A new feature was developed to aggregate the mean of this value over all of the purchases to gain an understanding of how often the item was bought again. This also required some data cleaning because many items were bought a few times, leading to artificially high reordered percentages. Figure 5 displays the most likely to be reordered items.

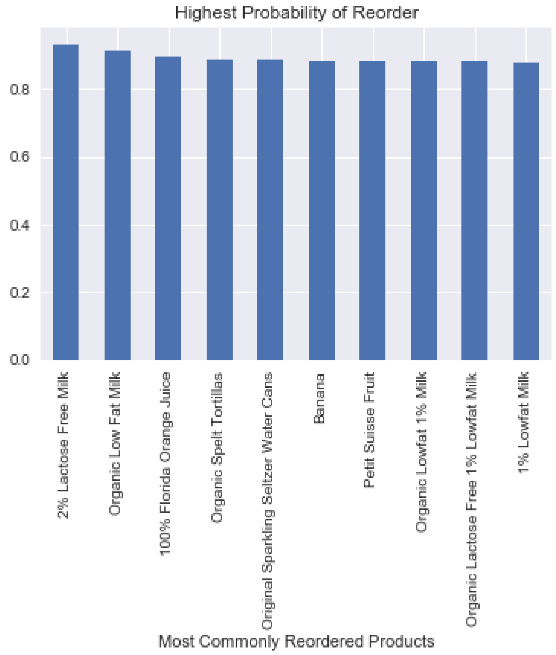


Fig. 6 - Probability of Reorder

Why do people buy so much milk? Are we baby cows?
Is it possible that there is a relationship between days since prior order and probability of an item being repurchased?

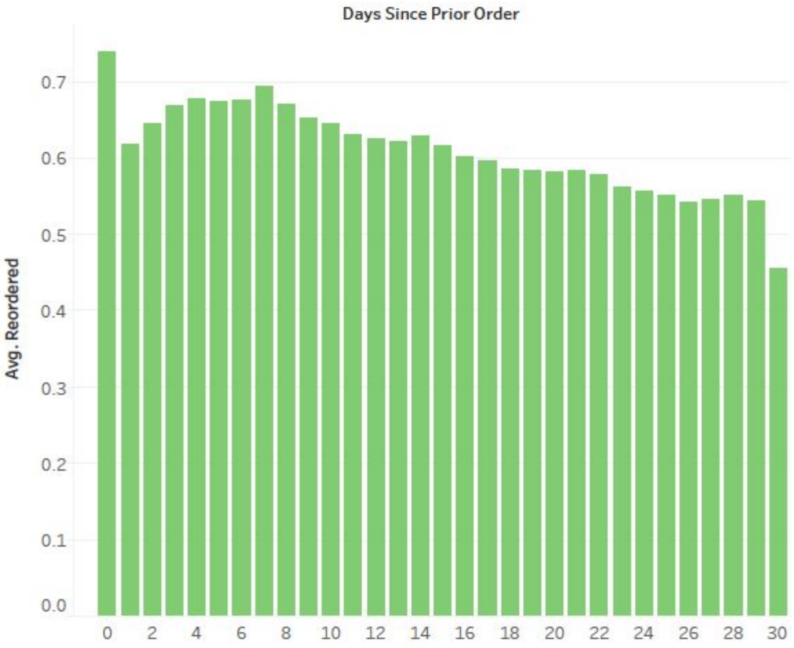
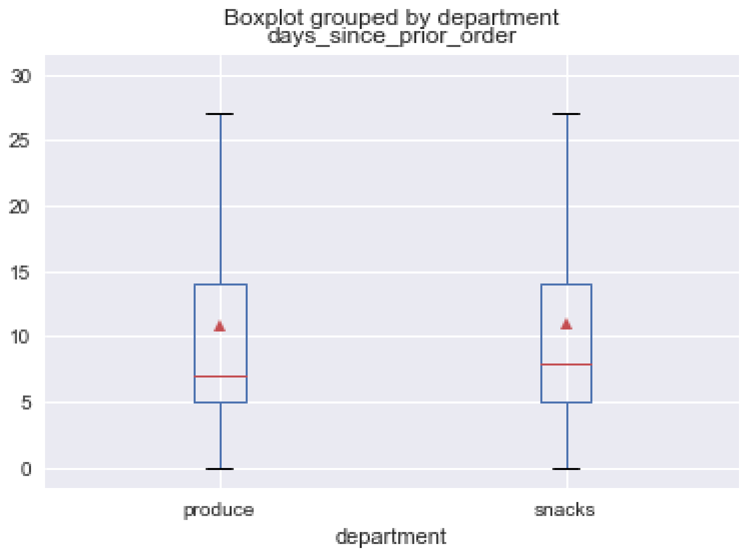


Fig. 7 - Elapsed Time & Reorder

The bar chart shows that if a customer purchases again within a week there is a higher likelihood of an item being reordered. This is likely an indication of customer loyalty. A customer who uses InstaCart for their sole supply of groceries is more likely to reorder the same items.

We now turn our attention to a statistical analysis surrounding days since prior order by department. This is a close proxy for product type. Imagine for a moment that people are receiving their week worth of groceries and deciding what to eat first? Are they more prone to the delicious snacks even though they buy more produce on average? Do they therefore return to the app sooner to replenish their snack supply? Unlike our childhoods, with an app like this, the cookie jar is never empty.

Split by department this makes for an interesting opportunity to test means using *hypothesis test for two samples*



These are huge samples at roughly 100k per vendor. As such we can determine the means and standard deviations to be normally distributed. The Central Limit Theorem holds that since sample size is >40.

Ho: $\bar{x} - \bar{y} = \Delta_o$
Ha: $\bar{x} - \bar{y} < \Delta_o$

This is to say that for Ha, the mean days since prior order for produce is less than that of snacks. This inequality indicates a lower tailed test where we reject the null hypothesis if $z \leq -z_\alpha$.

For an alpha value of 0.99 the $z_{critical}$ value is 2.33 (Probability & Statistics for Engineers - Table A.3) meaning that in order to reject the null hypothesis our calculated statistic has to be less than or equal to -2.33.

$$z = \frac{\bar{x} - \bar{y} - \Delta_o}{\sqrt{\left(\frac{s_x^2}{n_x} - \frac{s_y^2}{n_y}\right)}}$$

Equation of Z Statistic

In this case our z statistic is -12.7 which is less than -1.65 and so we must reject the null hypothesis that the mean days since prior order for each department are equal and accept that produce has a lower mean elapsed time since order than snacks.

The P-value is also 0 meaning Ho should be rejected for any reasonable significance level.

Congratulations to the users, they are making healthy choices!

Key Finding: Hypothesis Test
Produce is clearly the most purchased with the least elapsed time between purchases.

Machine Learning Regressions

Here we are trying to model the response of ‘reordered’. The problem is an ordinal classification problem and will benefit from a few techniques. The data is quite large and requires some dimensionality reduction before we proceed with regressions. The modelling routine will follow the iterative approach outlined below:

- Transform data to show UserID by Aisle, associates products with users
- performance of PCA vs Truncated SVD
- K-Means clustering on reduced data
- Logistic Regression
- Random Forest Regression
- XG Boost

We will then select the best model based on performance and then test it on the unseen data.

PCA vs Truncated SVD:

After pivoting on the data to associate UserID with Aisle we were left with a data frame of 130,000 rows by 134 columns. The size of this data frame was too large for clustering analysis on a local machine and had to be put through some dimensionality reduction using either PCA or Truncated SVD.

PCA did not perform as well as truncated SVD. PCA produced an explained variance of roughly 24% over 20 dimensions. Truncated SVD was able to explain 73% over 24 dimensions. Because of this large difference in explained variance we elected to use Truncated SVD as the method for reducing the original matrix for UserID by Aisle.

The next step was to apply a K-Means clustering routine in order to differentiate customers with unique buying behaviours.

Clustering with K-Means Outcomes:

Key Finding: Cluster Optimization

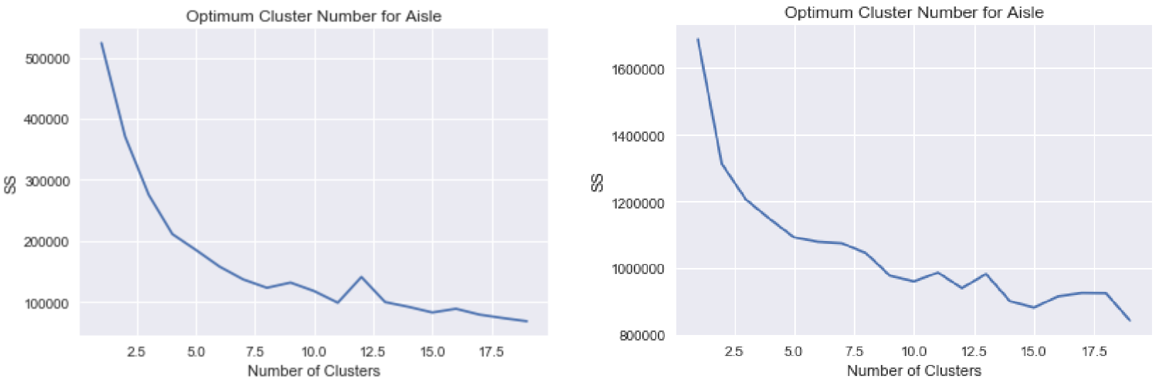


Fig. 8 - PCA & Truncated SVD Clustering

After reducing the transformed data a clustering optimization was performed using the elbow method. The two methods of dimensionality reduction produced similar clustering results.

Key Finding: Cluster Number

The cluster numbers of each dimensionality reduction were applied to the original data such that future modelling could take them into consideration as engineered features. Below, the figure demonstrates how each of the truncated SVD clusters reflect the aisle preferences of each.

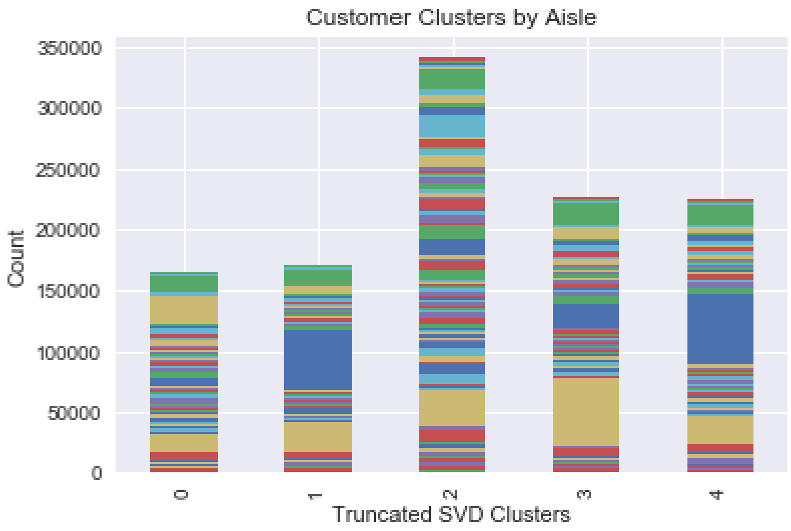


Fig. 9 - Customer Clusters by Aisle

While this figure doesn't provide a detailed explanation of the customer segments it confirms for us that there are unique values of customers in each one and that some aisles are more popular amongst the different segments.

Classification Models:

We performed three iterations of regression making use of SciKit Learn's Train, Test, Split methodology. The performance results are as follows:

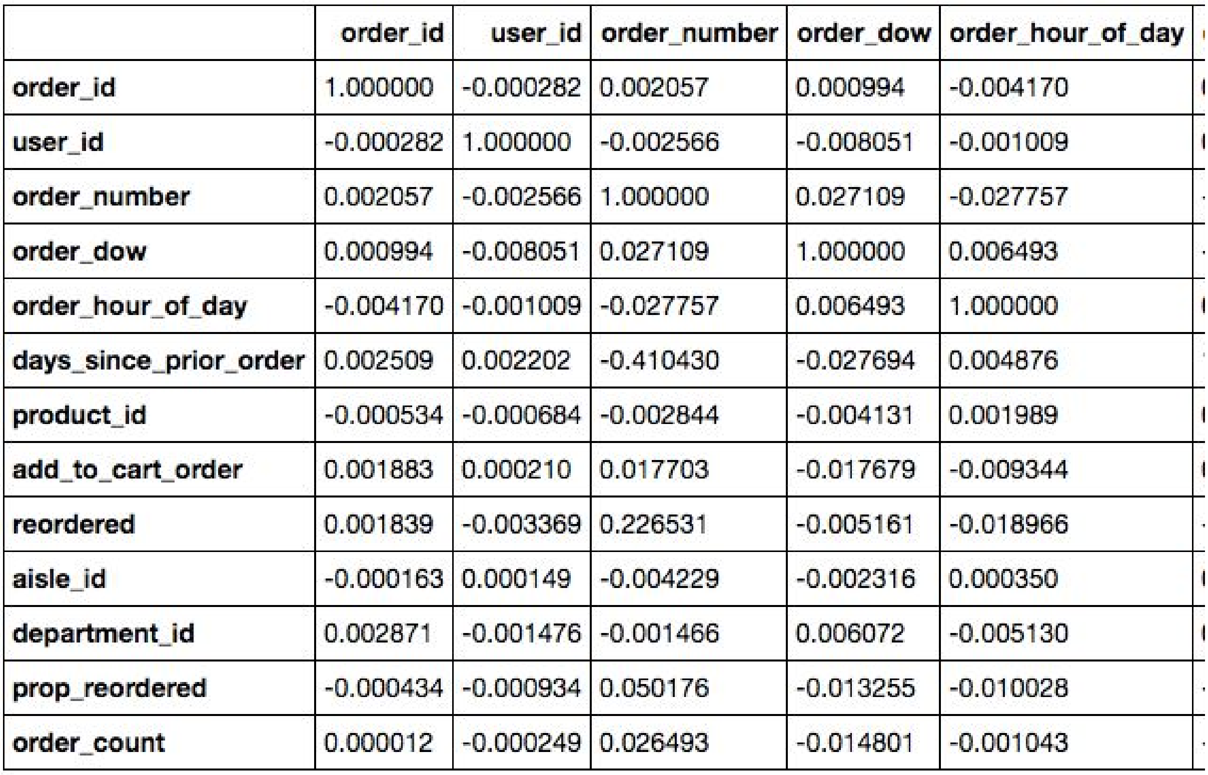
Table 1 - Classification Performance Metrics

	Logistic	Random Forest	XG Boost
Precision	0.40	0.67	0.70
Recall	0.63	0.67	0.71
F1 Score	0.49	0.67	0.69

The 'reordered' field was used as the response variable in all three models as we are trying to predict the likelihood that something will be reordered.

The logistic classification was used to establish a baseline. Further tweaking was not pursued as dramatic improvements were made using the second and third models.

It is worthwhile to provide a snapshot of the correlation matrix as a test of independence. Target correlation of 0.4 was used as a way to ward off multicollinearity. Luckily the features did not demonstrate multicollinearity in any significant way.



	order_id	user_id	order_number	order_dow	order_hour_of_day
order_id	1.000000	-0.000282	0.002057	0.000994	-0.004170
user_id	-0.000282	1.000000	-0.002566	-0.008051	-0.001009
order_number	0.002057	-0.002566	1.000000	0.027109	-0.027757
order_dow	0.000994	-0.008051	0.027109	1.000000	0.006493
order_hour_of_day	-0.004170	-0.001009	-0.027757	0.006493	1.000000
days_since_prior_order	0.002509	0.002202	-0.410430	-0.027694	0.004876
product_id	-0.000534	-0.000684	-0.002844	-0.004131	0.001989
add_to_cart_order	0.001883	0.000210	0.017703	-0.017679	-0.009344
reordered	0.001839	-0.003369	0.226531	-0.005161	-0.018966
aisle_id	-0.000163	0.000149	-0.004229	-0.002316	0.000350
department_id	0.002871	-0.001476	-0.001466	0.006072	-0.005130
prop_reordered	-0.000434	-0.000934	0.050176	-0.013255	-0.010028
order_count	0.000012	-0.000249	0.026493	-0.014801	-0.001043

Fig. 10 - Correlation Matrix as Test for Multicollinearity

A discussion of the Random Forest and XGBoost Classifier performance follows.

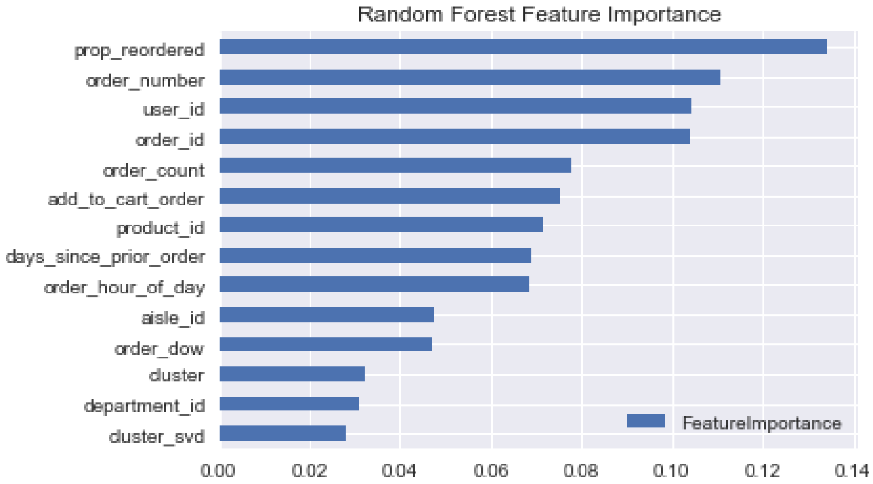


Fig. 11 - Random Forest Model Feature Importance

Intriguing that the Random Forest placed the more contextual features further down the list of importance. It appears to have provided a more literal approach to the model based on user_id and order_id.

The confusion matrix of Random Forest:

	Predicted No	Predicted Yes
Actual No	TN=77,791	FP=60,295
Actual Yes	FN=61,142	TP=173,641

The confusion matrix provides an idea of how successful the model is at accurately predicting positive and negative results. The table is meant to be read along the diagonal. We can see here that far more matches occurred than incorrect predictions.

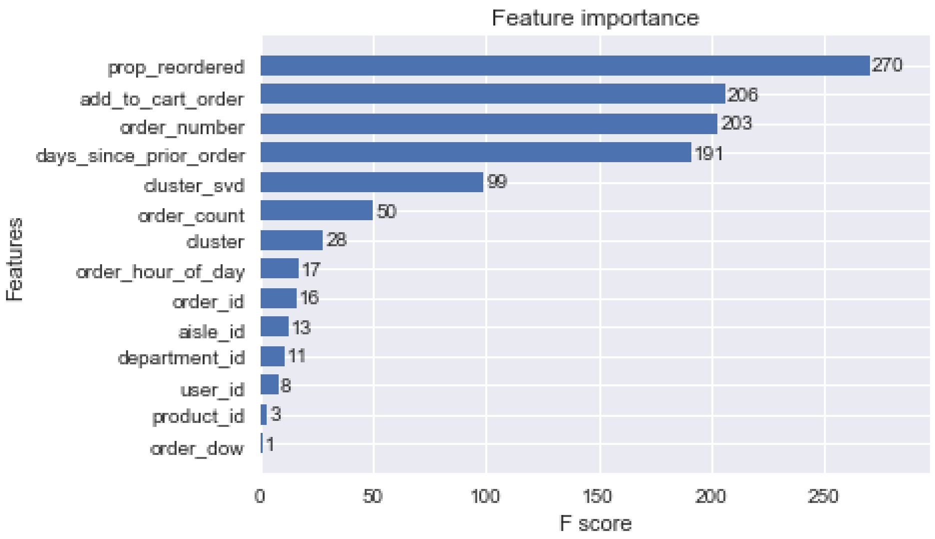


Fig. 12 - XGBoost Feature Importance, F Score

The XGBoost classifier has done a better job of producing a model that considers context such as sequence of item in order, days since previous order, clusters, size of order and time of day. The more literal features are left to the lower end of the feature importance series.

The Confusion Matrix of XGBoost:

	Predicted No	Predicted Yes
Actual No	TN=63,720	FP=74,366
Actual Yes	FN=35,348	TP=199,435

The XGBoost model produces a greater number of correct predictions and also fewer false predictions.

Improving the Model

A number of attempts were made at improving upon the initial logistic regression. The ensemble method of random forests worked well because they are capable of handling binary categorical features, features do not have to be linear and they do well with large datasets.

The XGBoost model created an added benefit because of the hyperparameter fitting.

Support Vector Machine was not explored due to it’s limitations on large datasets and high computing requirements.

The following parameter ranges were tested:
N_estimators [25,50,100]

Max_depth [5,10]
The greatest performance occured at n_estimators of 100 and max_depth of 5.

Results of Testing the Model

The model successfully predicted whether or not an item would be repurchased with a 70% accuracy.

Conclusions & Recommendations:
EDA & Inferential Statistics

The key findings behind the customer behaviours were quite revealing. It is clear that even though the app is handheld and entirely independent of a physical storefront the user demonstrates regular and consistent shopping patterns. For instance, while the shopping could be done at any time, the users are generally shopping in the early afternoon, perhaps when their minds are wandering to the age old question ‘what is for dinner?’

Performing clustering on the dimensionality reduction proved most interesting. Customers of a different type were identified by the algorithm. The differentiating features of these customers were not immediately obvious, partly because so many of us buy the same things. For example, the prevalence of dairy in our diets makes it highly likely that most people are buying milk products. I would not have expected the algorithm to pick out the handful of vegans shopping in the app but rather the similarities in features such as ‘add to cart order’ which identified the position of the item in the user’s cart.

Future studies could involve more feature engineering such as

- how many times an item was reordered by a user before they stopped
- Neural net feature learning
- Relative position in cart to other items
- Proportion of new items in each order

Additional data sets may include nutritional information and product studies such as organic vs non-organic.

Machine Learning Classifications

The opportunity to perform multiple classifications on the response variable ‘reordered’ proved highly educational and insightful. While the standard logistic regression did not perform very well it served as a great baseline for understanding the model.

The Random Forest classifier shone brightly as an improvement upon the logistic regression. This was most likely due to the ensemble method of voting from the individual decision trees.

XGBoost provided an even more impressive accuracy rating once tweaked using randomized search across a breadth of parameters. Ultimately, the XGBoost model was selected as the best performer.

Even further tweaks could be performed to enhance the generalizability of the model. In particular I would pursue some strategy of regularization in dropout or weight decay after enhancing with some new features.

Appendix:

Calculations

https://github.com/andrewcmilne/capstone1_instaCart

References

https://www.instacart.com/