DS 3001: Foundations of Machine Learning

December 10, 2024

Instacart Project: Results of Analysis

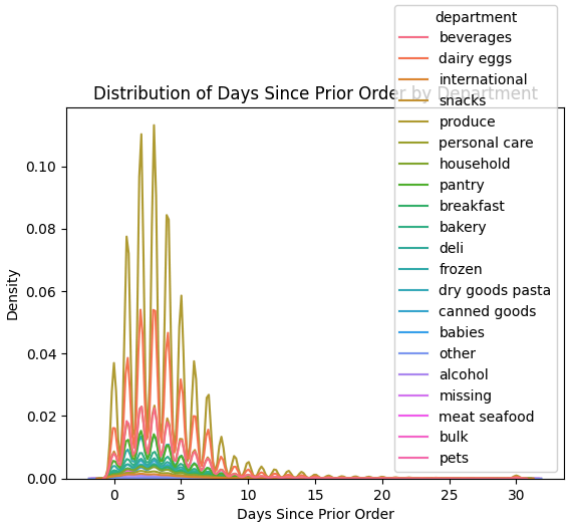
In order to seek to identify the relationships between variables in this Instacart dataset, we used a variety of methods to explore customer behavior. We chose customers as the main observation as we were interested in centering the data around behavior we can relate to. Our main prediction question guiding our analysis is: Which key variables most strongly influence customer purchasing frequency and preferences? We specifically looked at when and how often customers place orders, and from which departments. Additionally, we observed how the order of product selection communicates high-need items. To explore these prediction questions, we focused our analysis on the reordered, add_to_cart_order, total_order_count, and days_since_prior_order variables through EDA, single and multiple linear regression, PCA, Lasso, *k*-Means Clustering, and decision trees. By using many different techniques, we were able to investigate how the relationships between variables changed. Despite finding a range of results, which are described below, Instacart engineers may be interested in using these models to understand customers' grocery shopping habits and the factors that influence their behavior.

Regression and Classification (Pre-PCA and Lasso)

To begin our exploration of the data, we used linear regression models. We chose this model as it seeks to explain which variables are related to each other. Since our data has the potential for many unique explanations it was important to try out a variety of combinations. Our first step was doing more EDA. We ran various cross-tabulations, group-by statements, and visualizations to understand the relationships between different variables. These early results

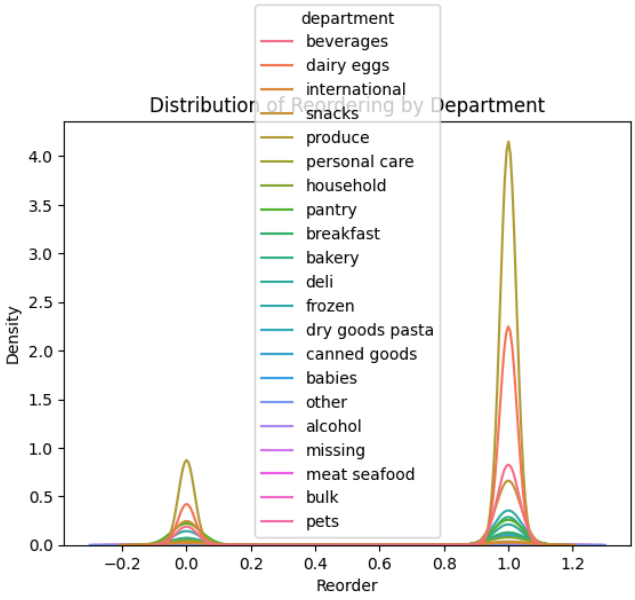
gave us ideas about what we wanted to further explore. Then we created kernel density plots. This type of plot is organized by peaks, where tall peaks communicate concentrations of the dependent variable and wide peaks communicate variability in the dependent variable with regard to the independent one.

The plot to the right displays how days_since_prior order is distributed across different departments. Since most of the peaks are centered over 0 to 8 days_since_prior order, there is low variability in the range of possible ordering times. In other words, customers shop on a bi-weekly to weekly basis. However, the height of the produce and dairy/eggs peaks indicate that a smaller ordering interval is concentrated in those departments. These departments may be especially high because they house products that go bad more quickly. Next, we ran a single linear regression model regressing days_since_prior order on department. The r^2 value was 0.001 indicating 1% of the variance can be explained by the model–essentially there is no correlation between the two variables. The coefficients of the model ranged from 3.98 for babies to 3.23 for alcohol. In general, department-specific behavior around reordering is not significantly varied. However, it is interesting to note that alcohol is associated with a lower value, as it is a pastime for many. These results were expected because it makes sense that more than just the department of a product would impact how often a customer makes a purchase.



Next, we looked at the distribution of departments in relation to whether or not products were reordered. Again, the height of the produce and dairy/eggs departments indicates more

concentrated reordering. The width is irrelevant as the variable is binary. For the model regressing reorders on departments, the r^2 value was 0.045, indicating 4.5% of the model can be explained. While this value is slightly higher, it still reveals a lack of correlation between the two variables. The coefficients ranged from 0.83 for produce and 0.49 for personal care products, showing produce is concentrated is more likely to be reordered and there is less of a pattern in consumer behavior for personal care products. The last single linear regression model we ran was regressing days single prior order on order day of the week. The r^2 was 0.007 indicating no correlation. However, it was interesting to see that the coefficients ranged from 3.41 to 4.09, with the highest being for Sunday. This shows that customers who placed Instacart orders on Sundays had the longest periods of time in between orders, possibly indicating that Sunday orders are more likely to be planned or larger orders, compared to more frequent orders placed throughout the week.



R-squared: 0.007362401986344 08

	variable	coefficient
0	friday	3.888866
1	monday	3.802638
2	saturday	3.995510
3	sunday	4.092941
4	thursday	3.466639
5	tuesday	3.453142
6	wednesday	3.409978

It is clear that these models are under-fitted because one variable can not capture complex relationships. Thus, it was time to explore multiple linear regression. We shifted to looking at classification and chose reorder as the dependent variable because we are most interested in understanding reordering behavior in customers. After regressing reorder on department (one hot

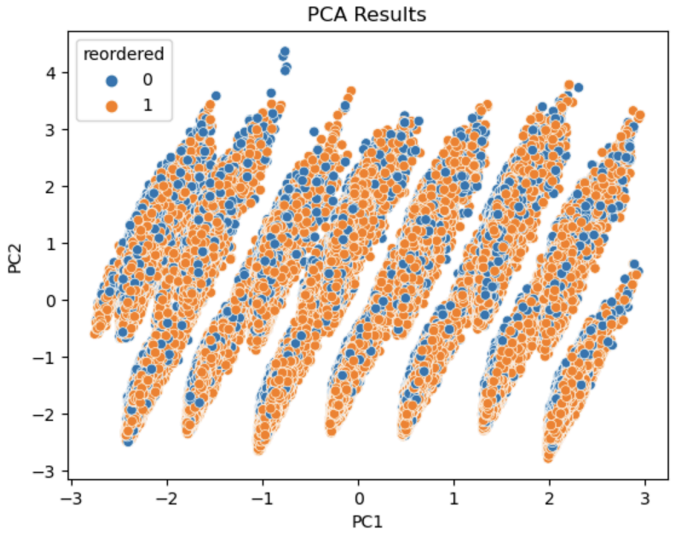
encoded), order day (one hot encoded), days_since_prior order, and add to cart order, the r^2 value was 0.095. Since the r^2 value increased each time we added a new variable in comparison to the isolated single linear regressions, we were able to determine multicollinearity is present in the data. Multicollinearity is a common phenomenon in data sets that occurs when more than one variable is correlated with another. When these variables are regressed on each other, they cancel out and can not accurately predict the correlation between them. PCA (principal component analysis) and LASSO (Least Absolute Shrinkage and Selection Operator) are two methods for reducing multicollinearity. The former transforms features into a separate set of uncorrelated features, while the latter reduces the coefficients of insignificant features.

PCA

Based on our results from the multi-linear regression, we chose to perform PCA on the reordered variable. The first step is to run the multiple linear regression on select features to provide a baseline r^2. While PCA accounts for all possible features when determining which independent variables may influence the dependent variable, we chose the most relevant features–excluding variables like product_id or departmet_id, which would not affect our results–due to runtime errors. We also chose to split the data into train and test sections to evaluate how future data might perform. We calculated an r^2 of -3.33 for the train set, indicating severe overfitting of the model and multicollinearity between variables. Before we could apply the PCA model, we had to pick the number of components to perform the decomposition on. The number of components is related to the number of independent features and visualizing a scree plot shows how much variation each component explains. The elbow point indicates where a

larger number of components stops explaining variation in the data. Our elbow was sharply at 2, so we created a PCA model using 2 principal components.

After determining the number of components, we ran the PCA transformation on the train and test sets. To our surprise, the r^2 value decreased to -3.37. While we expected this result to increase, we ultimately concluded that there is more to the story. The graph pictured to the right illustrates the first two components of reorders in a scatter plot, with 1 (orange) indicating a customer reordered a product and 0 (blue) indicating a customer did not reorder a product. The x-axis is the first component and the y-axis is the second component. Unfortunately, this graph illustrates no relationship between the components and consumer reordering behavior, as the goal is to separate the two components visually and determine the magnitude of the two components. The orange and blue dots are quite interspersed, suggesting neither component has strong predictive power in determining whether a customer reorders. While it is interesting that the data follows a continuous zigzag pattern, more successful PCA results would show clear clusters of instances. Clearly, neither component captures the variability or predictive patterns in reordered behavior.



LASSO

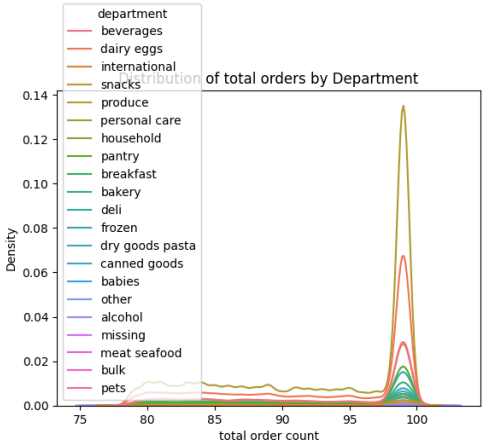
Since we did not identify meaningful conclusions from PCA, we ultimately decided to try LASSO for feature selection. We hoped that LASSO would pinpoint the features with the most

influence on key numerical variables. LASSO performs better on continuous variables so we pivoted from reorder to focus on other key numeric variables. In our analysis, we focused on add_to_cart_order, which represents the order in which a customer added a product to their order, and total_order_count, which is the total number of orders a customer placed in the 30 days observed.

First, we created a model displaying add_to_cart_order regressed on select features. While regressing on all variables in the data set would provide the best results, we similarly chose the same relevant features as our PCA model due to runtime errors. In the case of Instacart, we hypothesized that customers may place their most prioritized or top-of-mind products into their cart first. However, since product_name has 31,357 unique entries, our model did not have the capacity to process such a large number of products. So, we excluded product_name from our list of features and instead, since departments essentially capture the same information, we used department. By shrinking non-relevant features to zero, this model identified the most important features. On the right, the feature with the highest slope is babies, with a slope/coefficient of 0.918, showing products related to babies have the strongest influence on our add_to_cart_order prediction. Since add_to_cart_order represents the order in which customers put items in their cart, the LASSO results illustrate how customers frequently put baby products in their cart first.

	variable	slope
0	order_number	0.004207
1	days_since_prior_order	0.235986
2	order_hour_of_day	-0.021183
3	order_dow	-0.054817
4	reordered	-1.211459
5	total_order_count	-0.001457
6	friday	0.000000
7	monday	-0.402037
8	saturday	0.000000
9	sunday	0.060852
10	thursday	0.068638
11	tuesday	-0.385121
12	wednesday	-0.188231
13	alcohol	-0.463989
14	babies	0.917915
15	bakery	-0.000000
16	beverages	-0.804052
17	breakfast	0.393952
18	bulk	-0.000000
19	canned goods	0.261164
20	dairy eggs	-0.744989
21	deli	0.000000
22	dry goods pasta	0.332854
23	frozen	0.000000
24	household	-0.052272
25	international	0.000000
26	meat seafood	-0.000000
27	missing	0.000000
28	other	-0.000000
29	pantry	0.030761
30	personal care	-0.000000
31	pets	-0.000000
32	produce	-0.409442
33	snacks	0.652613

Secondly, we created a model that displays total_order_count regressed on key features. As mentioned above, we had to remove non-relevant features, such as product_id, due to runtime errors. After one hot encoding and running the LASSO model, our results illustrated that produce and dairy eggs–two departments–had two of the highest slopes. Produce has a slope of 0.124 and dairy eggs has a slope of 0.0275. Although neither coefficient is particularly large, their values being above zero indicate that both features contribute to predicting the total_order_count. So, in this model, as customers order more items from the produce or dairy eggs departments, their total order counts tend to increase as well. This may reflect how, since produce and dairy/eggs spoil faster, customers are ordering from these departments more often, and therefore, have a higher total order count. When performing EDA in earlier stages of our project, we noticed that produce (in yellow) and dairy eggs (in orange) have the highest density of total orders, as you can see in the kernel density plot to the right. This works to support our LASSO results that customers who have a larger total order count tend to purchase from the product and dairy eggs department the most.

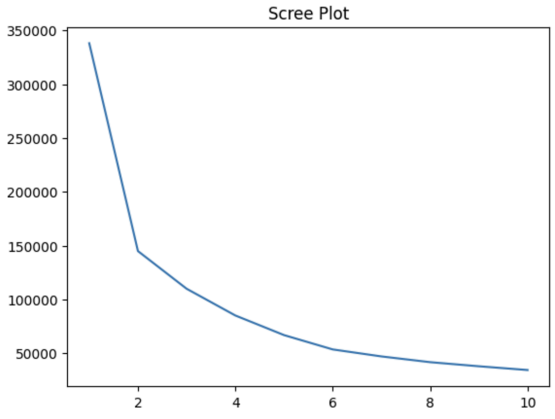


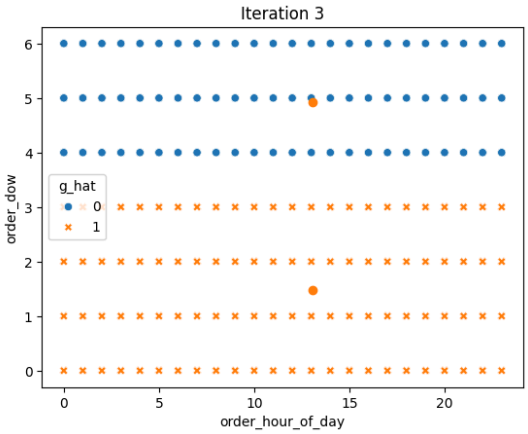
K-Means Clustering

Through *k*-Means Clustering (*k*-MC), an unsupervised algorithm method that focuses on patterns rather than a singular predictive answer, we sought to identify relevant patterns indicative of relationships between key variables in our dataset. The initial attempts at the standard *k*-MC algorithm with order_hour_of_day and order_dow were unsuccessful due to long runtimes. Because of this runtime error, the condensed version of the *k*-MC algorithm, with

pre-fixed iteration numbers, was used instead, which was successful. We originally hoped to find a pattern between order_dow, order_hour_of_day, and departments. While the initial scatterplot did not illustrate any clear relationships between the variables of interest, the *k*-MC plot had some clustering. Thus, we decided to make a scree plot that would minimize the sum of squared error for the model.

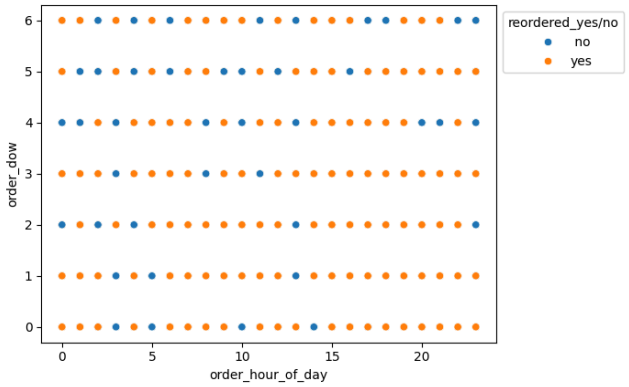
Again, a scree plot helps identify the most efficient *k* value for the number of cluster choices that minimize the sum of squared error of the *k*-means clustering model, which can be found at the 'elbow point.' This was observed to occur around 2, as seen in the left plot below. Since the initial *k*-MC was produced with a cluster value of 4, and we did not find a meaningful result, and we wanted to ensure that the sum of squared error was being minimized, we conducted another round of *k*-MC, this time with a cluster value of two.





When the k-MC algorithm was applied with a cluster value of 2, the plot to the right above was produced. In an attempt to conduct hierarchical clustering to further determine patterns, the runtime unfortunately crashed. Because of the difficulty in determining an observable pattern between the original scatterplot and the 4-cluster *k*-MC for the relationship between department, order_hour_of_day, and order_dow, we created another standard scatterplot

to see if there was potentially a relationship between the reordered_yes/no variable, order_hour_of_day, order_dow. When this standard scatterplot was created, as seen below, it closely resembled the clustering seen in the *k*-MC plot when the number of clusters was reduced to 2.

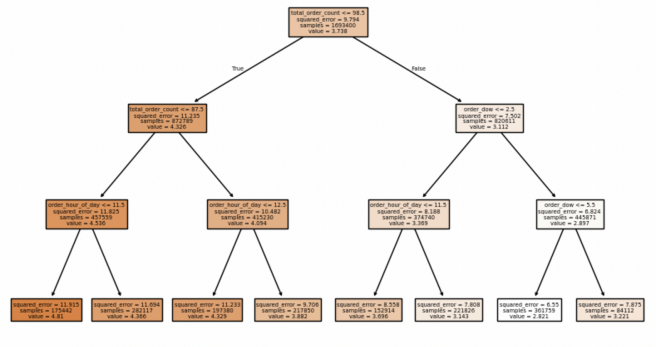


Based on the comparison between this scatterplot and the *k*-MC plot, we can determine that there is a relevant pattern between order_dow, order_hour_of_day, and reordered_yes/no. It seemed that reordering of products most often occurred on the weekend and earlier days of the week, those being Saturday, Sunday, Monday, and Tuesday, at relatively all hours of the day, while customers were more likely to not be reordering products later in the week, Wednesday through Friday. This could be because customers who reordered products early in the week did not need them again later on in the week. Later in the week, customers might be more likely to order different, or even less routinely-purchased, products that they did not previously reorder earlier in the week. This could be indicative of broader patterns in customer ordering, as customers might be more likely to place orders on the weekend and early days of the week, reordering repeat items from previous orders in a routine manner, as they are likely by that time running out of the repeat products they ordered the previous weekend.

Decision Trees

For our decision tree analysis, we aimed to identify the features that most strongly predict days_since_prior_order to understand patterns in time between orders placed by customers. The

five variables used as features for the decision tree were order_hour_of_day, order_dow, total_order_count, aisle_id, and department_id. Our mean squared error, which explains the difference between predicted and action values, was 9.3. This means that, on average, the model was 9 days off from the actual number of days_since_prior_order. Our mean absolute error was 2.02, showing that our model's average error in predicting reorder intervals is relatively low. Since our mean absolute error was significantly lower than our mean square error, we can infer that there are potential outliers in the data that inflate the squared error. The total_order_count was the dominant feature for prediction with 84%, which means that customers who have placed more total orders tend to reorder more frequently. Order_dow and order_hour_of_day also had some impact on the output, explaining 7% and 8% respectively, but they had significantly less influence than total_order_count.



This analysis highlights that the frequency of customer orders, as measured by days_since_prior_order, is more closely tied to the cumulative number of orders over time rather than specific days or times of the week. Unfortunately, the analysis found that departments and aisle-level variables had little meaningful impact on whether an item was reordered compared to the other features.