Clustering Customers & Predicting Purchasing Behavior with Instacart Data

Team 24: Marion Cassim, Nicole Flowerhill, Eugene Huang, Dongwon Jang, Pruek Vanna-iampikul

INTRODUCTION

The emergence of COVID-19 and its subsequent impact on consumer behavior created a shockwave through the global retail industry, creating winners of those who prioritized digital and omnichannel funnels, and losers of those that prioritized a physical retail footprint.

To succeed in this new environment, the use of data analytics is necessary to better understand customers while ensuring that they have the right product mix, sufficient inventory, and appropriate levels of staffing to guarantee an experience that will help generate customer loyalty.

PROBLEM DEFINITION

This project seeks to predict whether a customer will repurchase an item using logistic regression, random forest, and XGBoost supervised learning methods. We also seek to determine the possibility of segmenting grocery customers using K-Means clustering unsupervised learning method and the principles of RFM analysis.

DATA COLLECTION

Data Collection, Data Cleaning, Preprocessing, and Feature Engineering

We used Instacart's Online Grocery Shopping Dataset from 2017. It contains anonymized data of over 3 million orders from over 200,000 users, covering between 4 and 100 orders per user. The dataset has order information, product information, department and aisle information per product, the order in which products were added to cart and whether a user ordered the product previously.

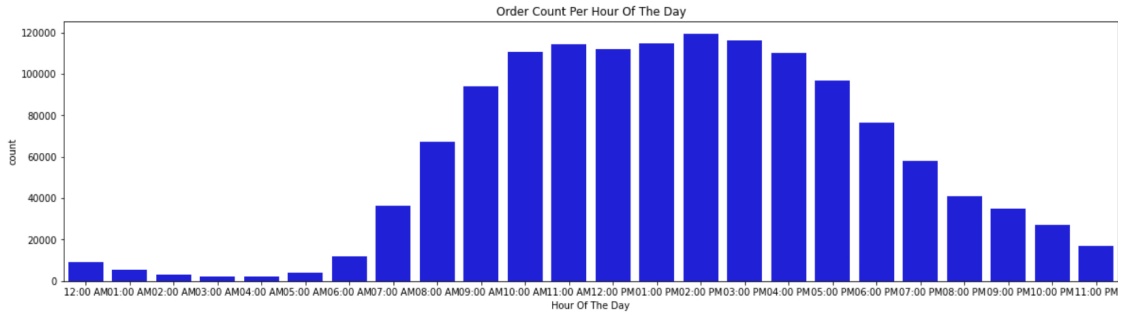
A thorough review of the dataset revealed that the data provided by Instacart was quite clean and there was not any missing information or items that did not make sense. We joined several csv files containing order information, product information, department, and aisle information per product to create a holistic dataset to use in our model. Given the dataset size and limitations of our personal computers, we used a subset of the data that contained the items ordered in the last order of a sequence of orders by Instacart customers. This dataset contained 1,384,617 data points and 16 variables.

To prepare the dataset for our prediction models, we conducted feature engineering by creating and modifying several variables. We changed the order day of week, which was a range of integers from 0 to 6, to Saturday through Friday, respectively. We also changed the order hour of day, which was a range of integers from 0 to 23, to a 12HR datetime format that included whether the order was placed in the AM or PM. Based on further review, we decided to create a new variable, order_part_of_day, to bin the hours the order was placed into higher level categories. This reduced the number of levels for time of day from 24 to 7.

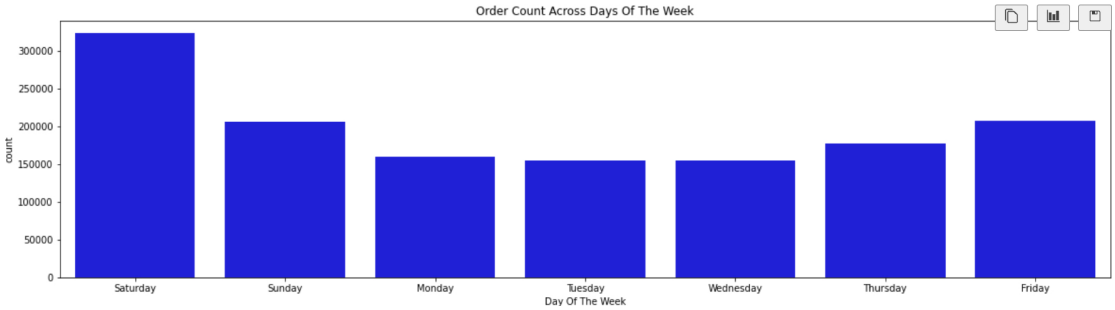
Exploratory Data Analysis

Through EDA using python's matplot library, we were able to observe several key points in terms of user's orders and date & time of order.

Analyzing the hour of day allowed us to conclude that most orders were placed between 10am to 3pm (Figure 1), with highest counts of orders on Saturday and Sunday (Figure 2)

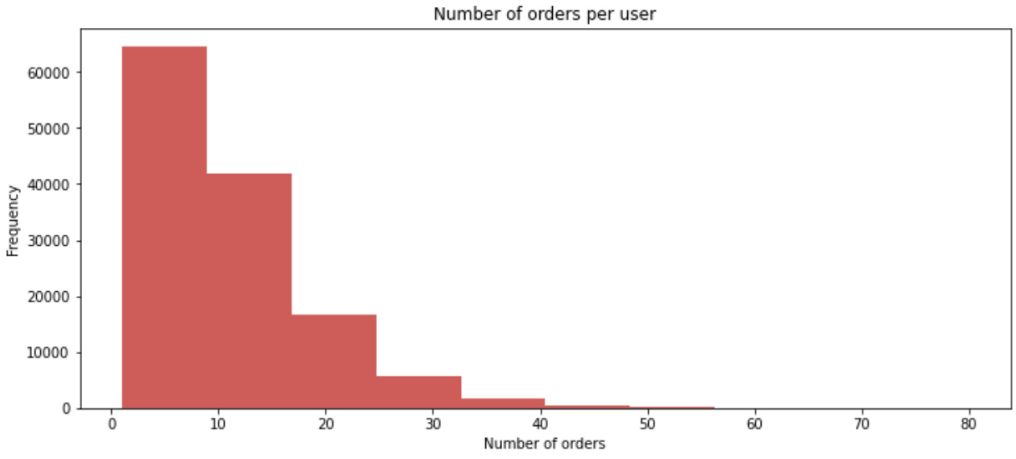


(Figure 1: Count of Orders per Hour of the Day)



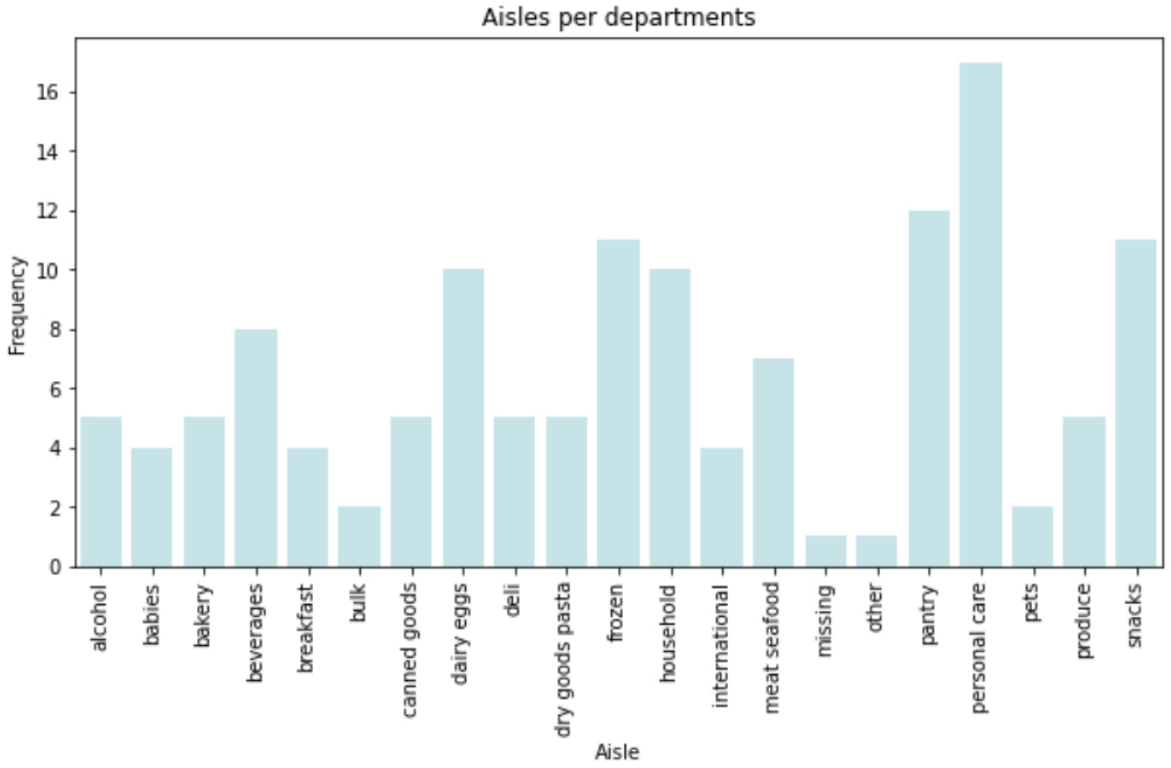
(Figure 2: Count of Orders Across Day of the Week)

In comparing the number of orders for each user, we can see that our sample dataset has users ordering between 3-10 orders (Figure 3).

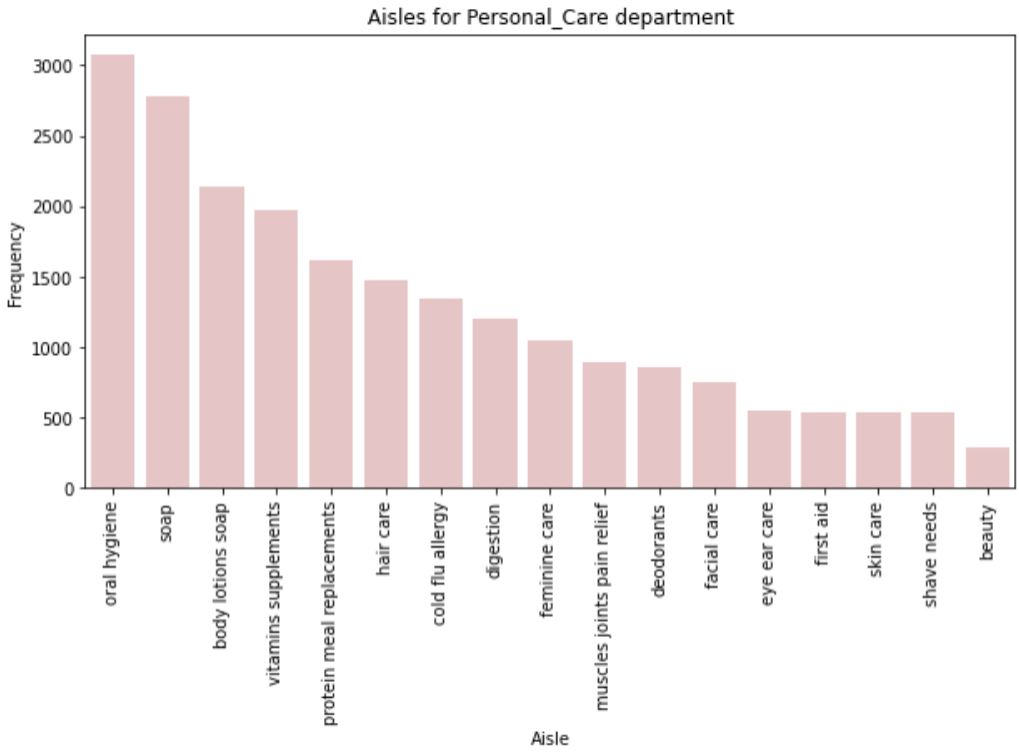


(Figure 3: Number of Orders per User)

Looking into the possible types of orders that a user can make, we see that products are categorized into different departments, each with their own number of aisles (Figure 4). The department with the biggest number of aisles is the “personal care” department, which we further analyzed in Figure 5 to see the different possible products.

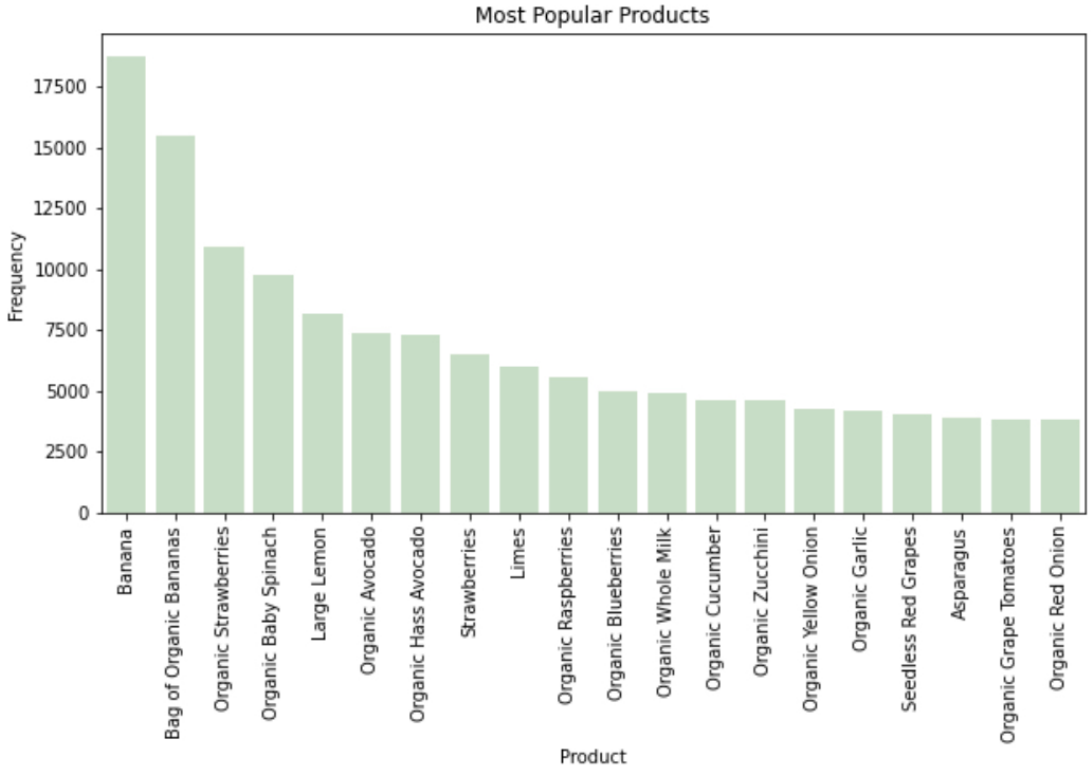


(Figure 4: Number of Aisles per Department)



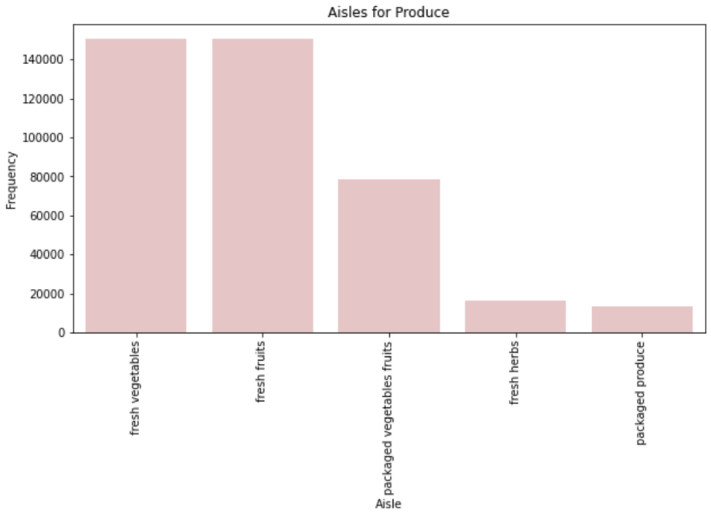
(Figure 5: Number of Aisles for “Personal Care” Department)

Although the “personal care” department had the highest number of aisles, we realized that its quantity of aisles may or may not correlate with the purchase quantity from this department. To analyze further, we created an analysis of the most popular products purchased (Figure 6) comparing the amount of times a product was purchased. We concluded no significant relationship between department aisle quantity and amount of products purchased after observing that the most popular products purchased were not of the “personal care” department. Since there is a large quantity of unique products, we have limited our Figure 6 graph to the top 20 most purchased products.



(Figure 6: Top 20 Most Popular Products)

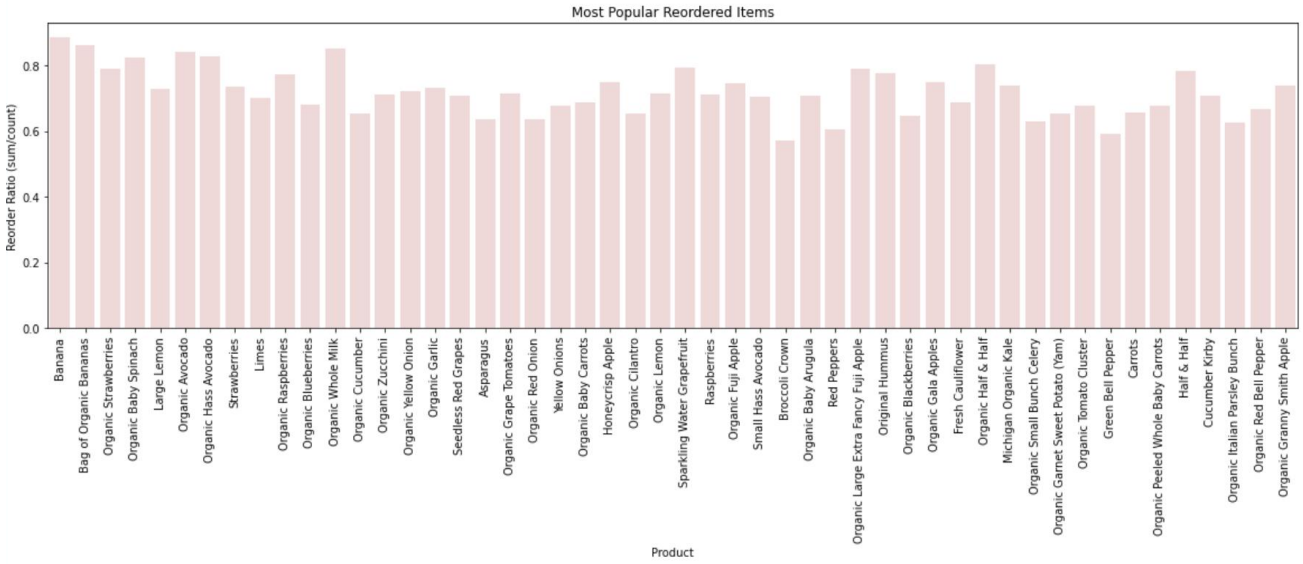
We notice that the highest frequency of products ordered are fresh produce, belonging to the "produce department," and observe in Figure 7 that only 5 aisles belong to this department (compared to the "personal care" department's 17 aisles).



(Figure 7: Aisles for "Produce" Department)

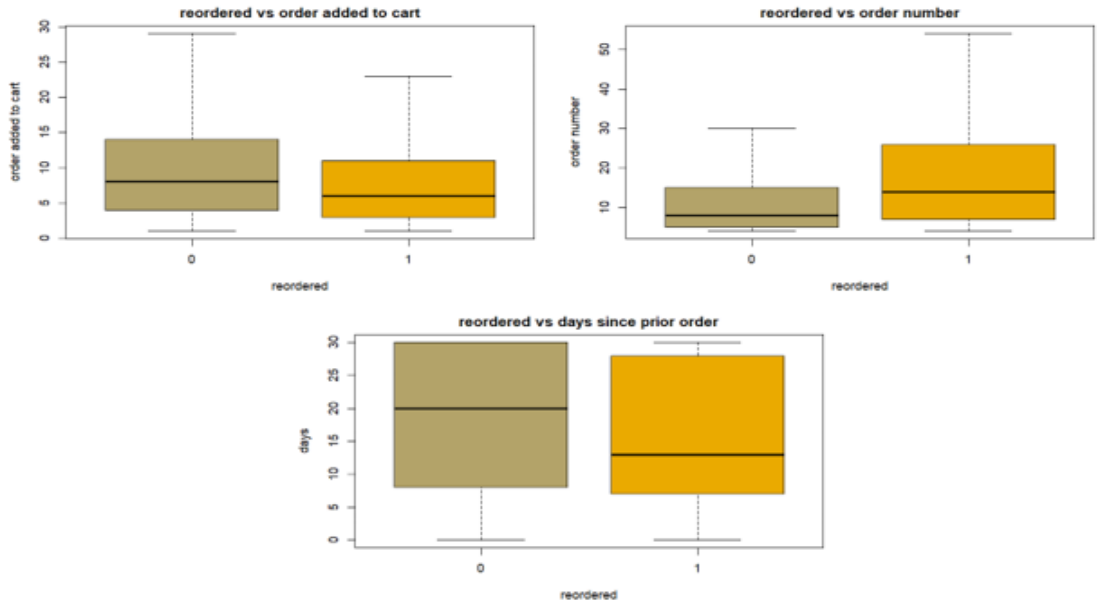
One of the most important pieces of our EDA includes the analysis of reordered items in our sampled dataset to find how many times a customer has re-ordered the same product. A product reorder ratio metric was created using the sum of each product divided by how many times that product was re-ordered, and a graph was created to show the top 50 products with the highest reorder ratio

(excluding products that have not been reordered at all). The results are parallel with previous findings in acknowledging that "produce" products are the ones most frequently reordered by customers.



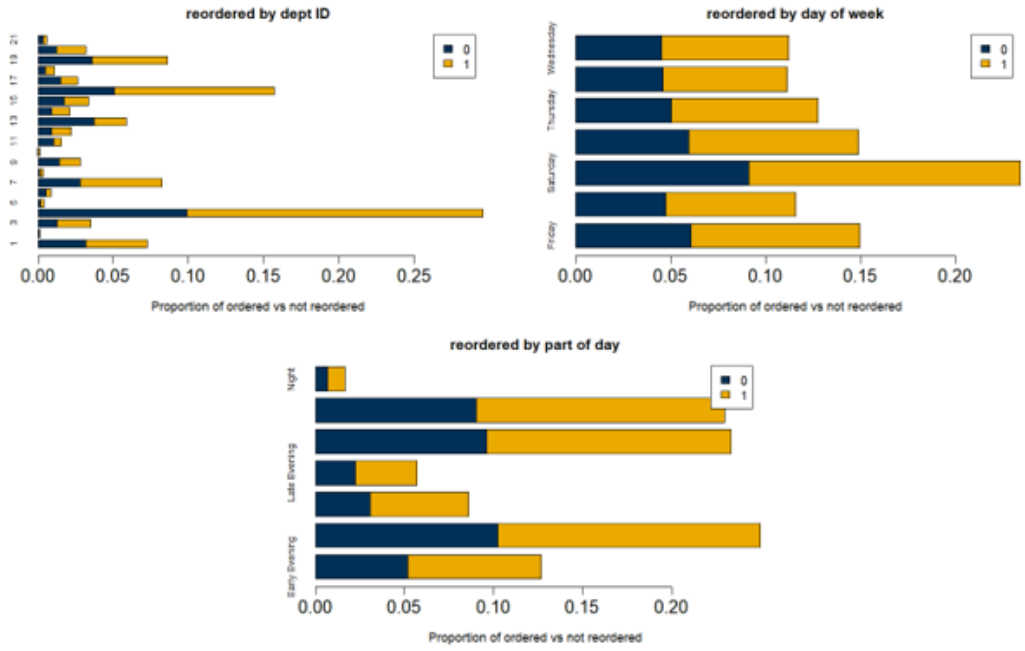
(Figure 8: Top 50 Most Popular Reordered Items)

We also conducted further EDA in R comparing our predictors with the response variable of whether a product was reordered or not. Based on the boxplots in Figure 9, we can see that there appears to be a significant difference between the number of days since the previous order made by a customer and whether a product was reordered or not, where items associated with a lower number of days since the prior order were reordered while those associated with a higher of number of days since the prior order were not reordered.



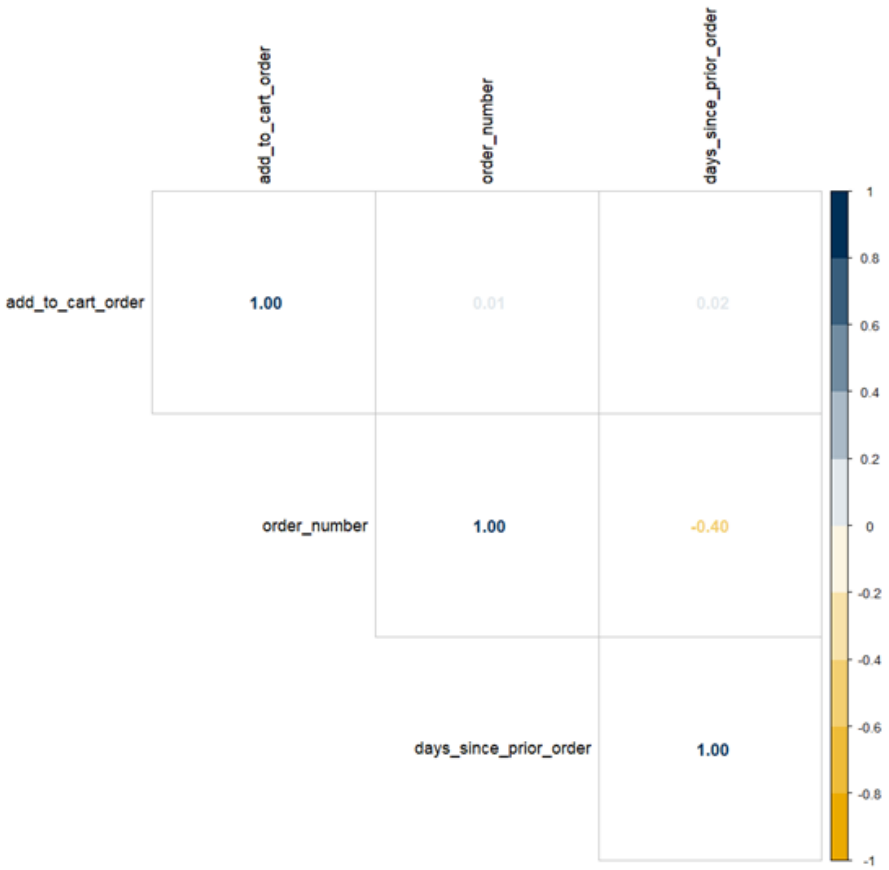
(Figure 9: Comparison of Numerical Features with Reordered Response Variable)

We can also see that items associated with customers who placed a higher number of orders were more likely to be reordered. Items that were ordered to a customer's cart earlier in the add to cart sequence were more likely to be reordered compared to those that were added later in the sequence.



(Figure 10: Comparison of Categorical Features with Reordered Response Variable)

Based on the bar plots in Figure 10, there does not appear to be much of a difference in the proportion of products that were reordered versus those that were not when viewed by the part of day the order was made and the day of the week the order was made. There appears to be a significant difference in the proportion of products that were reordered versus those that were not when viewing them by the department the products belonged to.



(Figure 11: Correlation Plot of Numerical Predictors)

Based on the correlation plot in Figure 11, we can see that there is a low negative correlation between the number of orders placed by a customer and the days since the customer's prior order (-0.40). There is a negligible correlation between the other numerical predictors.

METHODS, RESULTS, AND DISCUSSION: SUPERVISED LEARNING

Initial Dataset

The dataset that we used for our model contained 1,384,617 data points, 7 predictors and 1 response variable. The features are listed below in Figure 12:

FEATURE	DESCRIPTION	TYPE
reordered	1 if this product has been ordered by this user in the past, 0 otherwise	RESPONSE
add_to_cart_order	order in which each product was added to cart	NUMERICAL
aisle_id	aisle identifier	CATEGORICAL
order_number	number of orders made by customer	NUMERICAL
order_dow	the day of the week the order was placed on	CATEGORICAL
days_since_prior_o	days since the last order, capped at 30	NUMERICAL
part_of_day	the part of day an order was made	CATEGORICAL

(Figure 12: Features of Initial Dataset)

We randomly split the dataset into training and test sets using a 70% / 30% split, respectively. The training set contains 580,177 data points that are classified as reordered, accounting for approximately 59.9% of the training set. The test set contains 248,647 data points that are classified as reordered, accounting for approximately 59.9% of the test set.

Feature Selection

We built an initial logistic regression model in R with all the features as a foundation from which to conduct feature selection. VIF analysis of the initial model results produced an error, indicating that there was perfect correlation between two or more of the predictors in the model. Subsequent analysis of the relationships between the categorical predictors using Cramer V produced the following results:

Feature 1	Feature 2	Cramer V
REORDERED	ASILE	0.2346
REORDERED	DEPT	0.1972
REORDERED	DOW	0.01686
REORDERED	POD	0.03348
AISLE	DEPT	1
AISLE	DOW	0.03246
AISLE	POD	0.02552
DEPT	DOW	0.02462
DEPT	POD	0.0165
ORDER	POD	0.02475

(Figure 13: Cramer V Analysis of Categorical Variables)

The range of values that Cramer V can produce is between 0 and 1, where 0 indicates that there is no association between two variables and 1 indicates that there is perfect association between two variables. Based on our analysis results in Figure 13, we can see that there is perfect collinearity between the aisle_id and department_id predictors.

We conducted forward-backward stepwise regression for feature selection, and it confirmed the multicollinearity issue by removing all levels associated with the department_id predictor, thus effectively eliminating the predictor from the model. A subsequent VIF analysis of the reduced logistic regression model generated from stepwise regression indicated that multicollinearity was no longer an issue:

```
                          GVIF  Df GVIF^(1/(2*Df))
order_number          1.144415   1        1.069774
aisle_id              1.041450 133        1.000153
add_to_cart_order     1.033005   1        1.016369
days_since_prior_order 1.131455   1        1.063699
part_of_day           1.011866   6        1.000984
order_dow             1.017353   6        1.001435

VIF Threshold: 10
```

(Figure 14: VIF results from Logistic Regression Model after Stepwise Regression)

Logistic Regression Model

We used the supervised learning method of logistic regression to predict whether a customer would reorder a product (0 means will not reorder, 1 means will reorder), which could potentially be used to make future product recommendations based on their previous orders [Peng], [Huang].

The coefficients of the logistic regression model selected by stepwise regression are as follows:

```
Call:
glm(formula = reordered ~ order_number + aisle_id + add_to_cart_order +
    days_since_prior_order + part_of_day + order_dow, family = "binomial",
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0874  -1.1447   0.6475   0.9758   2.6966

Coefficients:
                       Estimate Std. Error  z value Pr(>|z|)
(Intercept)            0.325082   0.046886    6.933 4.11e-12 ***
order_number           0.567756   0.003217  176.472  < 2e-16 ***
aisle_id2             -0.318752   0.061403   -5.191 2.09e-07 ***
aisle_id3              0.070729   0.050126    1.411 0.158238
aisle_id4             -0.190791   0.052532   -3.632 0.000281 ***
aisle_id5             -1.170807   0.069144  -16.933  < 2e-16 ***
aisle_id6             -0.836972   0.076702  -10.912  < 2e-16 ***
aisle_id7             -0.174872   0.079839   -2.190 0.028502 *
aisle_id8             -0.375426   0.079120   -4.745 2.09e-06 ***
aisle_id9             -0.183567   0.052706   -3.483 0.000496 ***
aisle_id10            -1.763326   0.156486  -11.268  < 2e-16 ***
aisle_id11            -1.779032   0.096843  -18.370  < 2e-16 ***
aisle_id12            -0.073094   0.077854   -0.939 0.347801
aisle_id13             0.210591   0.061262    3.438 0.000587 ***
aisle_id14             0.375164   0.058868    6.373 1.85e-10 ***
aisle_id15            -0.108585   0.094973   -1.143 0.252902
aisle_id16            -0.171635   0.050275   -3.414 0.000640 ***
aisle_id17            -1.159949   0.051954  -22.326  < 2e-16 ***
aisle_id18             0.078039   0.106349    0.734 0.463070
aisle_id19            -0.883532   0.052681  -16.771  < 2e-16 ***
aisle_id20            -0.887852   0.065898  -13.473  < 2e-16 ***
```

aisle_id21	0.114418	0.047878	2.390	0.016857	*
aisle_id22	-1.287405	0.086034	-14.964	< 2e-16	***
aisle_id23	0.073822	0.055337	1.334	0.182187	
aisle_id24	0.654368	0.046794	13.984	< 2e-16	***
aisle_id25	-0.951140	0.068153	-13.956	< 2e-16	***
aisle_id26	0.099553	0.054009	1.843	0.065291	.
aisle_id27	0.109992	0.075908	1.449	0.147336	
aisle_id28	-0.075424	0.084667	-0.891	0.373018	
aisle_id29	-0.886079	0.067284	-13.169	< 2e-16	***
aisle_id30	-0.536682	0.062738	-8.554	< 2e-16	***
aisle_id31	0.285873	0.049310	5.797	6.73e-09	***
aisle_id32	0.547969	0.051907	10.557	< 2e-16	***
aisle_id33	-0.947927	0.113867	-8.325	< 2e-16	***
aisle_id34	-0.056845	0.063111	-0.901	0.367739	
aisle_id35	0.271308	0.058477	4.640	3.49e-06	***
aisle_id36	0.105716	0.052363	2.019	0.043496	*
aisle_id37	-0.378636	0.049108	-7.710	1.25e-14	***
aisle_id38	0.201735	0.049854	4.047	5.20e-05	***
aisle_id39	-0.219464	0.090274	-2.431	0.015054	*
aisle_id40	-0.011632	0.078335	-0.148	0.881959	
aisle_id41	0.328003	0.067048	4.892	9.98e-07	***
aisle_id42	0.139849	0.059016	2.370	0.017803	*
aisle_id43	-0.073260	0.058090	-1.261	0.207259	
aisle_id44	-1.270671	0.125584	-10.118	< 2e-16	***
aisle_id45	-0.081248	0.051931	-1.565	0.117691	
aisle_id46	0.100385	0.094806	1.059	0.289672	
aisle_id47	-1.312295	0.076744	-17.100	< 2e-16	***
aisle_id48	0.173688	0.064517	2.692	0.007100	**
aisle_id49	0.274002	0.057457	4.769	1.85e-06	***
aisle_id50	0.127285	0.055724	2.284	0.022361	*
aisle_id51	-0.219887	0.057839	-3.802	0.000144	***
aisle_id52	0.325971	0.053177	6.130	8.79e-10	***
aisle_id53	0.410638	0.052131	7.877	3.35e-15	***
aisle_id54	-0.099685	0.051241	-1.945	0.051726	.
aisle_id55	-1.202671	0.125627	-9.573	< 2e-16	***
aisle_id56	-0.493218	0.088696	-5.561	2.69e-08	***
aisle_id57	0.047416	0.061937	0.766	0.443944	
aisle_id58	-0.122809	0.079130	-1.552	0.120663	
aisle_id59	-0.052414	0.051663	-1.015	0.310323	
aisle_id60	-0.875274	0.079424	-11.020	< 2e-16	***
aisle_id61	0.011872	0.052672	0.225	0.821676	
aisle_id62	0.393894	0.093252	4.224	2.40e-05	***
aisle_id63	-0.609277	0.056572	-10.770	< 2e-16	***
aisle_id64	0.238150	0.059384	4.010	6.06e-05	***
aisle_id65	-0.380813	0.077279	-4.928	8.32e-07	***
aisle_id66	-0.913542	0.055918	-16.337	< 2e-16	***
aisle_id67	0.206312	0.050672	4.072	4.67e-05	***
aisle_id68	-0.236691	0.110758	-2.137	0.032597	*
aisle_id69	-0.427876	0.050476	-8.477	< 2e-16	***
aisle_id70	-0.584737	0.087176	-6.708	1.98e-11	***
aisle_id71	-0.018748	0.076196	-0.246	0.805642	
aisle_id72	-0.995029	0.053507	-18.596	< 2e-16	***
aisle_id73	-0.815301	0.107924	-7.554	4.21e-14	***
aisle_id74	-0.730289	0.061932	-11.792	< 2e-16	***
aisle_id75	-0.429492	0.058894	-7.293	3.04e-13	***

aisle_id75	-0.429492	0.058894	-7.293	3.04e-13	***
aisle_id76	-0.305621	0.103563	-2.951	0.003167	**
aisle_id77	0.284110	0.050483	5.628	1.82e-08	***
aisle_id78	0.063067	0.049628	1.271	0.203797	
aisle_id79	0.172254	0.054505	3.160	0.001576	**
aisle_id80	-1.301176	0.106820	-12.181	< 2e-16	***
aisle_id81	-0.419476	0.051334	-8.172	3.05e-16	***
aisle_id82	-0.260444	0.155532	-1.675	0.094026	.
aisle_id83	0.137659	0.046697	2.948	0.003199	**
aisle_id84	0.913489	0.049164	18.580	< 2e-16	***
aisle_id85	-1.319876	0.069246	-19.061	< 2e-16	***
aisle_id86	0.628557	0.050187	12.524	< 2e-16	***
aisle_id87	-1.437724	0.105702	-13.602	< 2e-16	***
aisle_id88	-0.361909	0.051564	-7.019	2.24e-12	***
aisle_id89	-0.934466	0.060369	-15.479	< 2e-16	***
aisle_id90	-0.518846	0.088660	-5.852	4.85e-09	***
aisle_id91	0.435828	0.049084	8.879	< 2e-16	***
aisle_id92	-0.135465	0.051234	-2.644	0.008191	**
aisle_id93	0.307043	0.053225	5.769	7.99e-09	***
aisle_id94	-0.298602	0.053031	-5.631	1.80e-08	***
aisle_id95	-0.229043	0.063656	-3.598	0.000321	***
aisle_id96	0.148864	0.050221	2.964	0.003035	**
aisle_id97	-1.993271	0.110760	-17.996	< 2e-16	***
aisle_id98	0.067879	0.050827	1.336	0.181711	
aisle_id99	-0.005581	0.060741	-0.092	0.926787	
aisle_id100	-0.955861	0.054403	-17.570	< 2e-16	***
aisle_id101	-0.979518	0.092840	-10.551	< 2e-16	***
aisle_id102	-1.367189	0.164143	-8.329	< 2e-16	***
aisle_id103	-1.155524	0.130898	-8.828	< 2e-16	***
aisle_id104	-2.041161	0.058157	-35.098	< 2e-16	***
aisle_id105	-0.549944	0.058992	-9.322	< 2e-16	***
aisle_id106	0.043862	0.051297	0.855	0.392521	
aisle_id107	0.134516	0.048398	2.779	0.005447	**
aisle_id108	0.074840	0.051354	1.457	0.145023	
aisle_id109	-1.523855	0.131970	-11.547	< 2e-16	***
aisle_id110	-0.407754	0.058504	-6.970	3.18e-12	***
aisle_id111	-0.689812	0.072953	-9.456	< 2e-16	***
aisle_id112	0.428970	0.049356	8.691	< 2e-16	***
aisle_id113	-0.372915	0.155717	-2.395	0.016628	*
aisle_id114	-1.105892	0.058073	-19.043	< 2e-16	***
aisle_id115	0.662884	0.048503	13.667	< 2e-16	***
aisle_id116	-0.032910	0.049177	-0.669	0.503350	
aisle_id117	-0.255519	0.051430	-4.968	6.76e-07	***
aisle_id118	-1.754733	0.140724	-12.469	< 2e-16	***
aisle_id119	-0.426370	0.093886	-4.541	5.59e-06	***
aisle_id120	0.447355	0.047614	9.395	< 2e-16	***
aisle_id121	0.075613	0.050306	1.503	0.132821	
aisle_id122	-0.143841	0.064238	-2.239	0.025144	*
aisle_id123	0.321342	0.047165	6.813	9.55e-12	***
aisle_id124	-0.137105	0.091789	-1.494	0.135253	
aisle_id125	0.191090	0.082161	2.326	0.020028	*
aisle_id126	-1.059739	0.094425	-11.223	< 2e-16	***
aisle_id127	-0.841177	0.073301	-11.476	< 2e-16	***
aisle_id128	-0.034810	0.053805	-0.647	0.517649	
aisle_id129	-0.033245	0.053408	-0.622	0.533631	
aisle_id130	-0.287333	0.055878	-5.142	2.72e-07	***
aisle_id131	-0.324768	0.051901	-6.257	3.91e-10	***

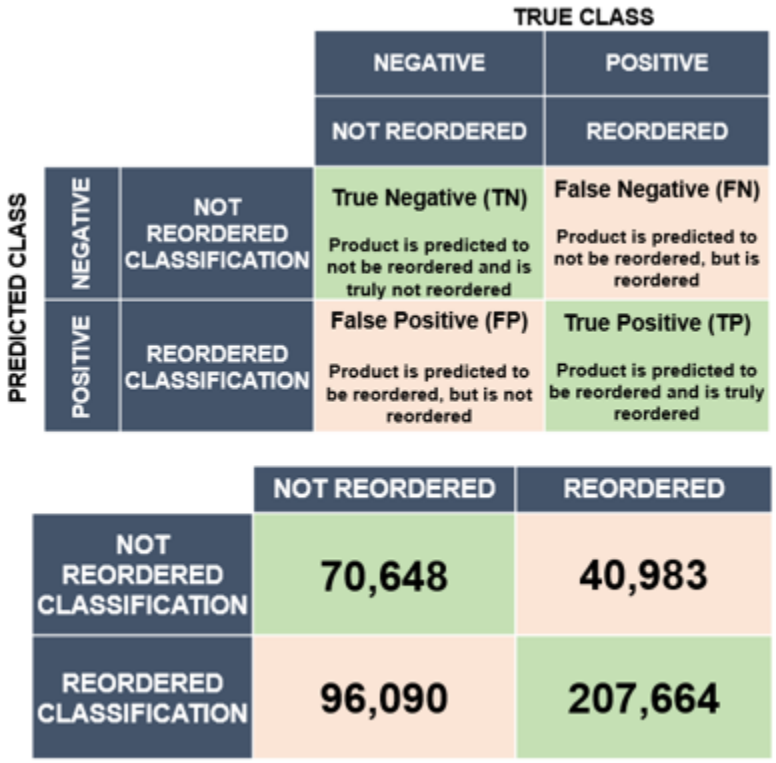
```
aisle_id132                -1.745145   0.191059   -9.134  < 2e-16 ***
aisle_id133                -0.951173   0.097726   -9.733  < 2e-16 ***
aisle_id134                -0.247564   0.123985   -1.997 0.045855 *
add_to_cart_order          -0.290281   0.002285 -127.011  < 2e-16 ***
days_since_prior_order     -0.158617   0.002387  -66.460  < 2e-16 ***
part_of_dayEarly_Afternoon  0.015740   0.007623    2.065 0.038948 *
part_of_dayEarly_Morning    0.152960   0.009939   15.390  < 2e-16 ***
part_of_dayLate Evening     0.031618   0.011212    2.820 0.004802 **
part_of_dayLate_Afternoon   0.001455   0.007699    0.189 0.850090
part_of_dayLate_Morning     0.060377   0.007754    7.786 6.90e-15 ***
part_of_dayNight           -0.009286   0.018220   -0.510 0.610300
order_dowMonday             0.016228   0.008650    1.876 0.060637 .
order_dowSaturday           0.057876   0.007357    7.866 3.65e-15 ***
order_dowSunday             0.072383   0.008115    8.920  < 2e-16 ***
order_dowThursday          -0.002392   0.008481   -0.282 0.777943
order_dowTuesday           -0.014690   0.008754   -1.678 0.093333 .
order_dowWednesday         -0.022869   0.008762   -2.610 0.009053 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1305706  on 969231  degrees of freedom
Residual deviance: 1173431  on 969083  degrees of freedom
AIC: 1173729

Number of Fisher Scoring iterations: 4

               aisle_id3               aisle_id12               aisle_id15               aisle_id18
                       4                       13                       16                       19
              aisle_id23               aisle_id26               aisle_id27               aisle_id28
                      24                       27                       28                       29
              aisle_id34               aisle_id40               aisle_id43               aisle_id45
                      35                       41                       44                       46
              aisle_id46               aisle_id54               aisle_id57               aisle_id58
                      47                       55                       58                       59
              aisle_id59               aisle_id61               aisle_id71               aisle_id78
                      60                       62                       72                       79
              aisle_id82               aisle_id98               aisle_id99              aisle_id106
                      83                       99                      100                      107
             aisle_id108              aisle_id116              aisle_id121              aisle_id124
                     109                      117                      122                      125
             aisle_id128              aisle_id129 part_of_dayLate_Afternoon         part_of_dayNight
                     129                      130                      141                      143
         order_dowMonday         order_dowThursday          order_dowTuesday
                     144                      147                      148

 Variables not selected by forward-backward stepwise: department_id2 department_id3 department_id4 department_id5
department_id6 department_id7 department_id8 department_id9 department_id10 department_id11 department_id12
department_id13 department_id14 department_id15 department_id16 department_id17 department_id18 department_id19
department_id20 department_id21
```

(Figure 15: Logistic Regression Model Results Following Stepwise Regression)

As we can see from the results above, all the numerical predictors, and many levels of the categorical predictors, have a statistically significant relationship with the reordered response variable at the 99.999% significance level.

When checking the p-value for the overall model, we have a value of zero, indicating that the overall model is statistically significant and has explanatory power.

Logistic Regression Prediction Results



(Figure 16: Confusion Matrix of Logistic Regression Prediction Results and Interpretation Guide)

When evaluating prediction model results, we can use accuracy, precision, sensitivity, and specificity ratios:

Accuracy is the ratio of correctly labeled data points compared to the entire set of data points and is represented by the following equation:

$$\frac{TP+TN}{TP+FP+FN+TN}$$

Precision is the ratio of data points that were correctly labeled as positive compared to the entire set of data points that were labeled as positive and is represented by the following equation:

$$\frac{TP}{TP+FP}$$

Sensitivity is the ratio of data points that were correctly labeled as positive compared to the entire set of data points that are truly positive and is represented by the following equation:

$$\frac{TP}{TP+FN}$$

Specificity is the ratio of data points that were correctly labeled as negative compared to the entire set of data points that are truly negative. Specificity is represented by the following equation:

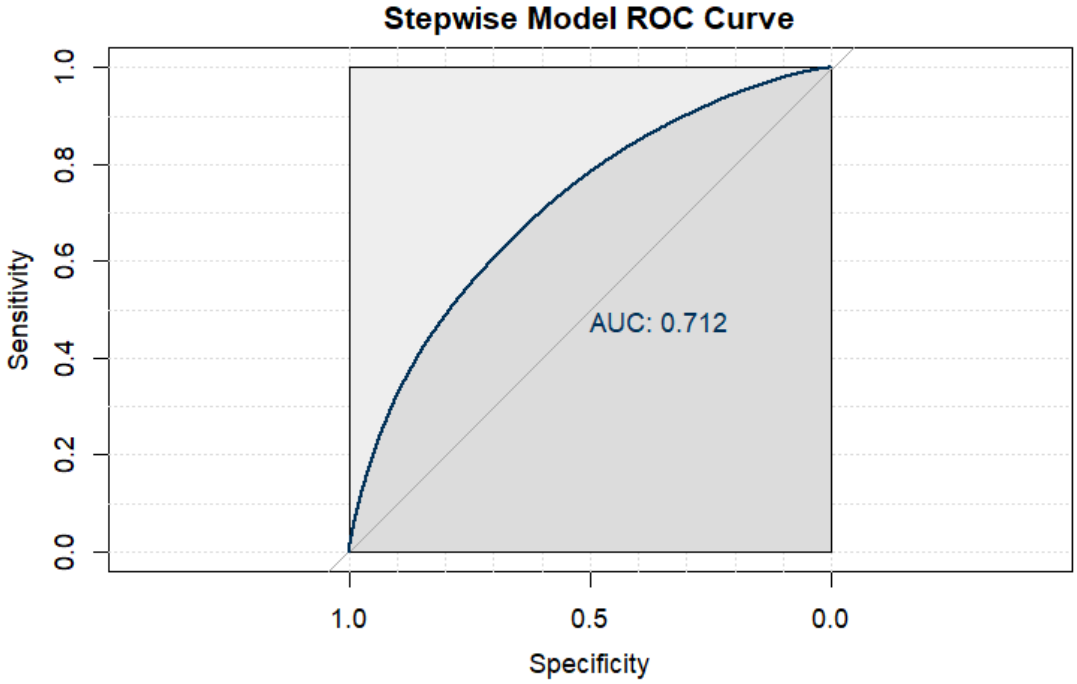
$$\frac{TN}{TN + FP}$$

Our model's accuracy, precision, sensitivity, and specificity scores are as follows:

MODEL TYPE	ACCURACY	PRECISION	SENSITIVITY	SPECIFICITY
STEPWISE	0.6700097	0.6836585	0.8351760	0.4237067

(Figure 17: Logistic Regression Prediction Metrics)

Precision and sensitivity are the two most important metrics for our model, as we want to know if a customer will reorder an item or not. Our logistic regression model predicts 83 out of 100 reorders and 68 out of 100 reorder predictions were correct.

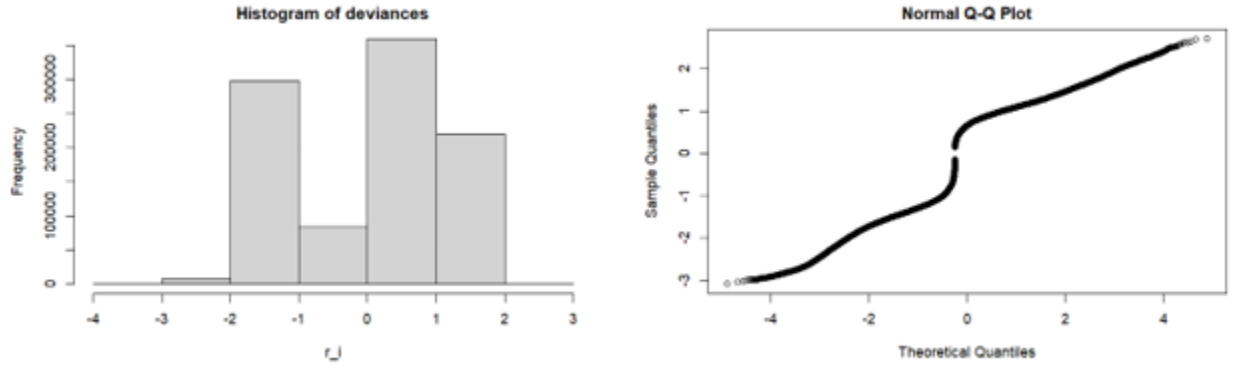


(Figure 18: Logistic Regression ROC-AUC Curve)

An ROC curve is a graphical interpretation of how capable our model is at distinguishing between products that were reordered versus products that were not reordered. The AUC score is a summary of the curve with a score between 1 and 0, with 1 representing perfect separation between two classes. Our model accomplished an AUC score of 0.712, which is significantly better than the 0.5 score of a random classifier.

Logistic Regression Goodness-of-Fit

As deviance residuals are the equivalent to the residuals in standard linear regression, one of the things we want to check is normality to see if the model is a good fit for the data. We used a histogram and QQ plot of the deviance residuals to check for normality:



(Figure 19: Histogram and QQ Plot of Residuals)

Basedon the histogram and QQ plot of the deviance residuals, there appears to be a significant departure of normality. This is confirmed by a deviance test for goodness-of-fit using deviance residuals, where we achieve a p-value of 0, indicating that the model is not a good fit for the data.

We can try to improve the model fit by adding interaction terms, transforming predictors, and identifying and removing outliers. Sometimes it may also be the case where the logit function may not fit the data and we can try other s-shape functions, such as probit or complementary log-log.

Random Forest

We subsequently created a Random Forest model in R using the default parameters (300 trees) to see if it could perform better than the stepwise Logistic Regression model at prediction. However, the model refused to run with the *aisle_id* feature included as it had 134 levels. We decided to substitute *aisle_id* with *department*, as it aggregated detailed *aisle* information into higher level groupings (i.e. aisle:"Bulgarian Yogurt" → department:"dairy eggs") resulting in a more manageable 21 levels.

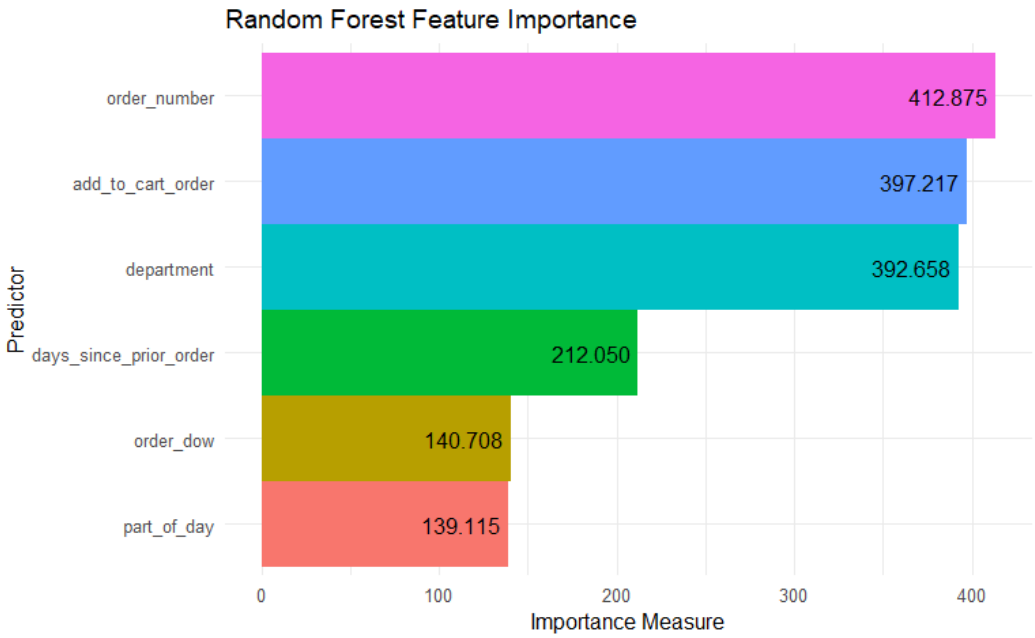


Figure 20: Random Forest Feature Importance Plot

According to the Random Forest model, *order_number* (the total number of orders placed by a customer), was the most important feature for determining whether an item was reordered or not, followed *add_to_cart_order* (the order in which an item was added to a cart), and *department* (category of the product).

Random Forest Prediction Results

Our model's prediction results are as follows:

	NOT REORDERED	REORDERED
NOT REORDERED CLASSIFICATION	82,252	81,486
REORDERED CLASSIFICATION	50,989	197,658

Figure 21: Random Forest Prediction Results

Our model's accuracy, precision, sensitivity, and specificity scores are as follows:

MODEL TYPE	ACCURACY	PRECISION	SENSITIVITY	SPECIFICITY
RANDOM FOREST	0.6810790	0.7949342	0.7080861	0.6257441

Figure 22: Random Forest Prediction Metrics

Precision and sensitivity are the two most important metrics for our model, as we want to know if a customer will reorder an item or not. Our Random Forest model predicts 79 out of 100 reorders and 70 out of 100 reorder predictions were correct.

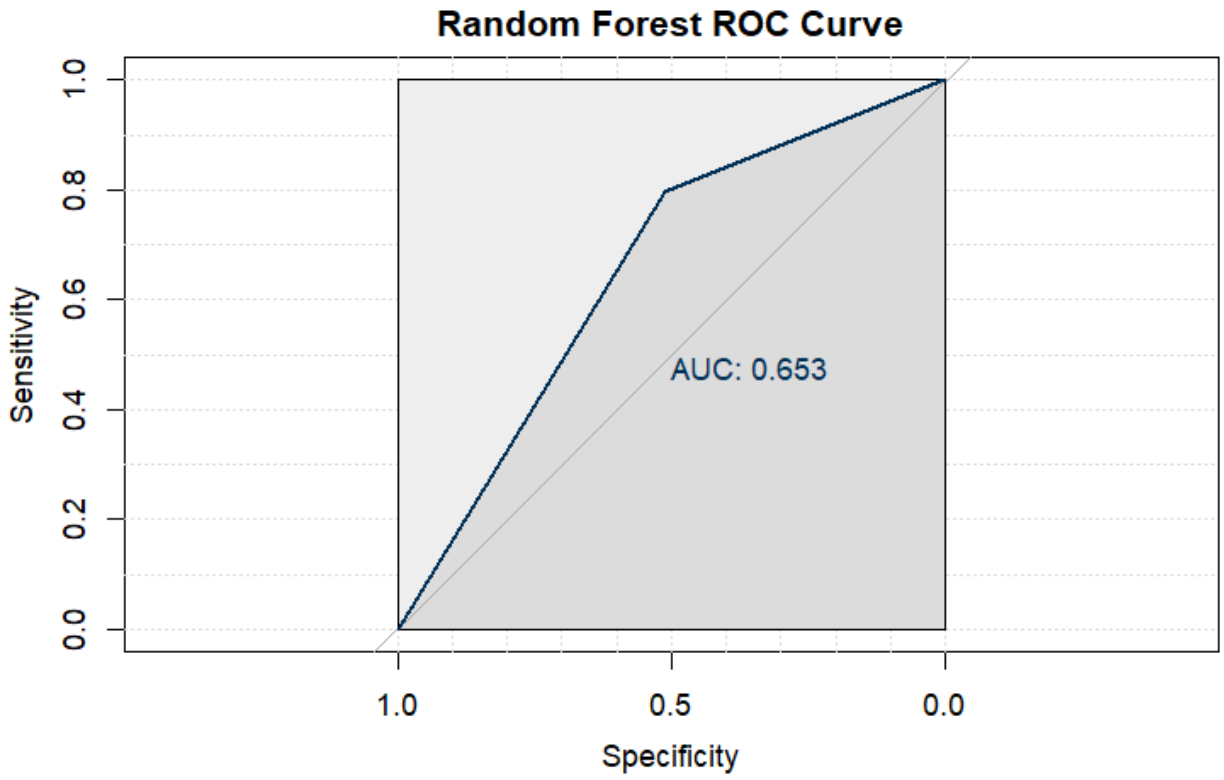


Figure 23: Random Forest ROC-AUC Curve

Our Random Forest model accomplished an AUC score of 0.653, which is significantly better than the 0.5 score of a random classifier, but worse than the 0.712 score of the Logistic Regression Model.

XGBoost

We then created an XGBoost model in R using the *department* feature and model default parameters (number of iterations = 100, learning rate = 0.3, regularization = 0, max depth of tree = 6, minimum number of instances required in a child node = 1, samples supplied to a tree = 1, number of features supplied to a tree = 1) to compare its performance to the Logistic Regression and Random Forest models.

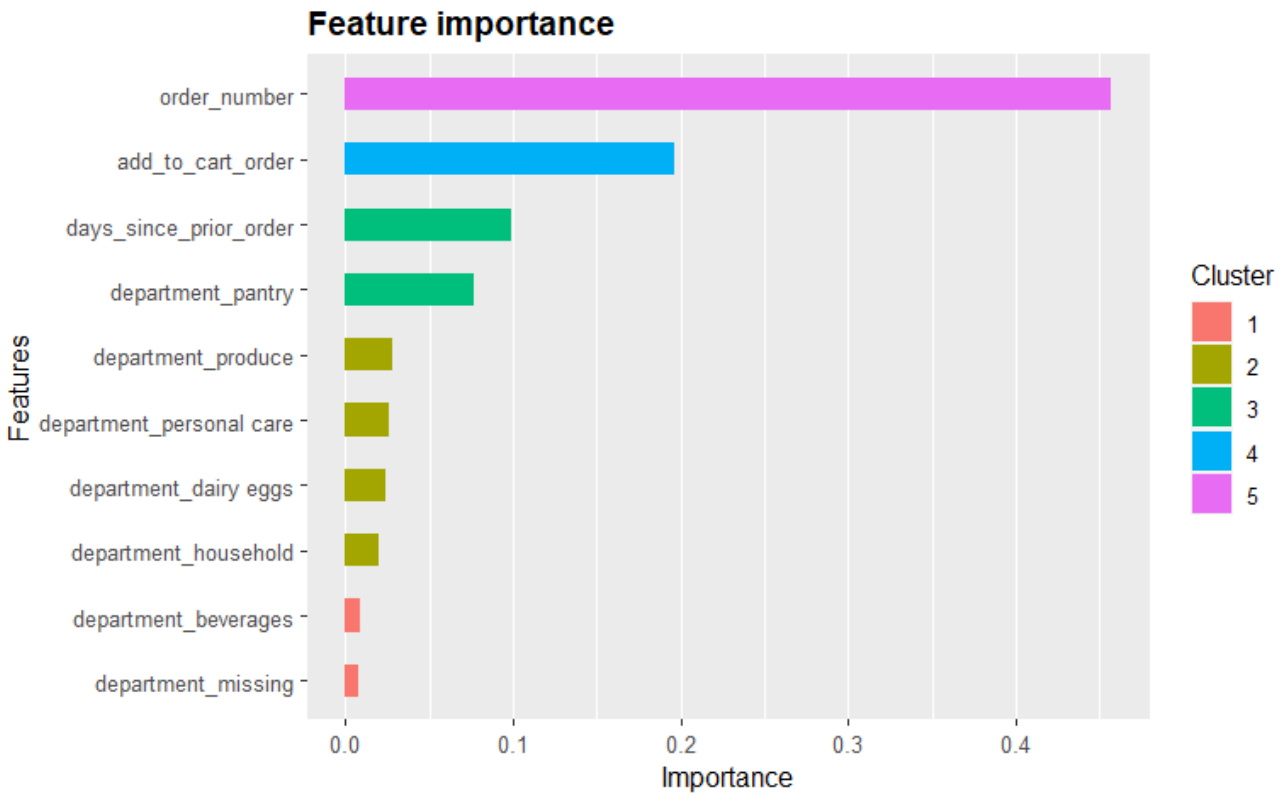


Figure 24: XGBoost Feature Importance Plot

The total number of orders by a customer was clearly the most important feature as determined by the XGBoost model regarding if an item would be repurchased or not, followed by steep dropoff to the next important feature, the order in which the item was added to the cart. The third most important feature was the number of days that had elapsed since the previous order.

XGBoost Prediction Results

Our model's prediction results are as follows:

	NOT REORDERED	REORDERED
NOT REORDERED CLASSIFICATION	81,331	48,352
REORDERED CLASSIFICATION	85,407	200,295

Figure 25: XGBoost Prediction Results

Our model's accuracy, precision, sensitivity, and specificity scores are as follows:

MODEL TYPE	ACCURACY	PRECISION	SENSITIVITY	SPECIFICITY
XGBOOST	0.6779879	0.7010626	0.8055396	0.4877772

Figure 26: XGBoost Prediction Metrics

Precision and sensitivity are the two most important metrics for our model, as we want to know if a customer will reorder an item or not. Our XGBoost model predicts 70 out of 100 reorders and 80 out of 100 reorder predictions were correct.

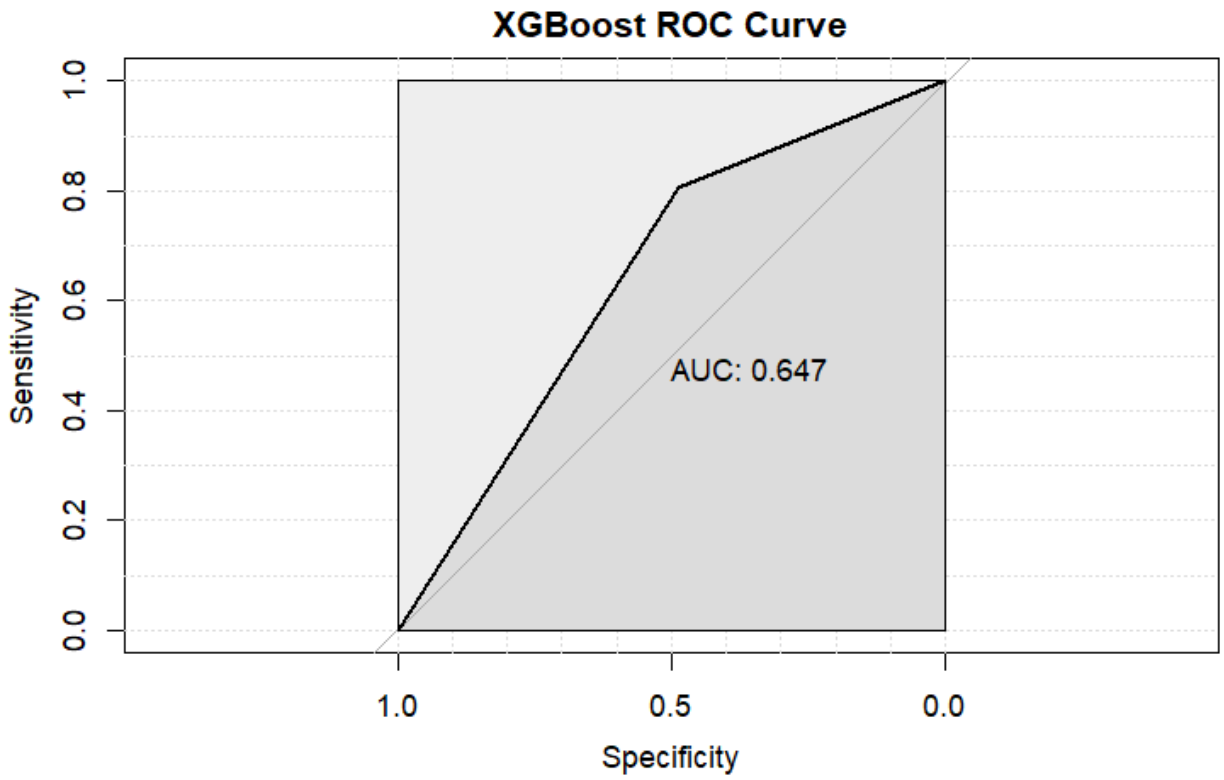


Figure 27: XGBoost ROC-AUC Curve

Our XGBoost model accomplished an AUC score of 0.647, which is significantly better than the 0.5 score of a random classifier, but worse than the 0.712 score of the Logistic Regression Model and 0.653 score of the Random Forest model.

Comparison of Supervised Learning Models

MODEL TYPE	ACCURACY	PRECISION	SENSITIVITY	SPECIFICITY	AUC
LOG REG W/AISLE	0.6700097	0.6836585	0.8351760	0.4237067	0.712
LOG REG W/DEPT	0.6627129	0.6742728	0.8444904	0.3916384	0.704
RANDOM FOREST	0.6810790	0.7949342	0.7080861	0.6257441	0.653
XGBOOST	0.6779879	0.7010626	0.8055396	0.4877772	0.647

Figure 28: Prediction Metrics Across All Classification Models

To compare the Random Forest and XGBoost models directly with the logistic regression model, we reran the logistic regression model with the *department* feature instead of the *aisle* feature. Not surprisingly, there was a slight dropoff in most metrics as the *aisle* feature contains more detailed information than the aggregate *department* feature.

As we can see, Random Forest performed the best in general across all the models with the highest accuracy, precision, and specificity scores. Since we are concerned with predicting whether a customer will reorder an item or not, precision and sensitivity are the two metrics we focus on the most. Our Random Forest model scored the highest precision value, indicating that it correctly predicts 79 out of 100 reorders, while Logistic Regression using the *department* feature scored the highest sensitivity value, indicating that 84 out of 100 reorder predictions were correct.

Based on the metrics, we think the Random Forest model is the best choice out of the models we tested on this dataset, especially with its significantly higher precision score. There is the possibility that tuning of the Random Forest hyperparameters could produce even better results. Despite the XGBoost model not performing as well as the Random forest model, it has the reputation of outperforming Random Forest in many situations. As we used the base parameters for the XGBoost model in this project, it is possible that with some proper tuning, it can outperform the Random Forest model.

Logistic Regression with Additional Feature Engineering

We later decided to build another logistic regression model in Python after engineering new features that we felt would better capture the relationship between reorders and closely related features. The engineered features are generated from existing features and can largely be categorized into four groups: user-product, user, product, and aisle groups. The following is the list of all the additional features generated through feature engineering.

FEATURE	DESCRIPTION	TYPE
up_nb_reordered	the total reorder of each product for each user	NUMERICAL
up_tot_purchase	the total purchase of each product for each user	NUMERICAL
up_tot_reordered	the total number user have reordered the same product	NUMERICAL
u_mean_dow	the average day of the week which user have made orders	NUMERICAL
u_order_no	the number of order which user have made in the past	NUMERICAL
u_mean_cart	the mean cart size for each user	NUMERICAL
u_max_cart	the maximum cart size for each user	NUMERICAL
p_reorder_ratio	the reorder ratio of particular products	NUMERICAL
p_user	number of unique users who have purchased the product	NUMERICAL
up_reorder_ratio	user-product reorder ratio	NUMERICAL
a_reorder_ratio	the reorder ratio of particular ailes	NUMERICAL

Figure 29: New Engineered Features

The subsequent logistic regression model that we built included all the features that we used previously, along with our newly engineered features. We used the *department* feature instead of *aisle_id* in this example so it would be comparable with our previous model runs. Below is the complete list of features used in this model:

FEATURE	DESCRIPTION	TYPE
reordered	1 if this product has been ordered by this user in the past	RESPONSE
add_to_cart_order	order in which each product was added to cart	NUMERICAL
days_since_prior_order	days since the last order	NUMERICAL
up_nb_reordered	the total reorder of each product for each user	NUMERICAL
up_tot_purchase	the total purchase of each product for each user	NUMERICAL
up_tot_reordered	the total number user have reordered the same product	NUMERICAL
u_mean_dow	the average day of the week which user have made orders	NUMERICAL
u_order_no	the number of order which user have made in the past	NUMERICAL
u_mean_cart	the mean cart size for each user	NUMERICAL
u_max_cart	the maximum cart size for each user	NUMERICAL
p_reorder_ratio	the reorder ratio of particular products	NUMERICAL
p_user	number of unique users who have purchased the product	NUMERICAL
up_reorder_ratio	user-product reorder ratio	NUMERICAL
a_reorder_ratio	the reorder ratio of particular ailes	NUMERICAL
order_dow	the dat of the week the order was placed on	CATEGORICAL
department	the department of product	CATEGORICAL
part_of_day	the part of day an order was made	CATEGORICAL

Figure 30: List of All Features used in Logistic Regression with Additional Feature Engineering

The newly built logistic regression model generated the following coefficient values:

Generalized Linear Model Regression Results

Dep. Variable:	reordered	No. Observations:	969232
Model:	GLM	Df Residuals:	969187
Model Family:	Binomial	Df Model:	44
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-4.3693e+05
Date:	Tue, 06 Dec 2022	Deviance:	8.7386e+05
Time:	18:40:13	Pearson chi2:	2.86e+06
No. Iterations:	6	Pseudo R-squ. (CS):	0.3595
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.1625	0.044	-49.532	0.000	-2.248	-2.077
add_to_cart_order	-0.1354	0.001	-225.449	0.000	-0.137	-0.134
days_since_prior_order	-0.0039	0.000	-14.249	0.000	-0.004	-0.003
up_nb_reordered	-0.0006	4.65e-05	-12.550	0.000	-0.001	-0.000
up_tot_purchase	-0.2656	0.002	-148.028	0.000	-0.269	-0.262
up_tot_reordered	0.4517	0.001	362.366	0.000	0.449	0.454
u_mean_dow	0.0016	0.001	1.053	0.292	-0.001	0.004
u_order_no	0.0127	0.000	51.743	0.000	0.012	0.013
u_mean_cart	0.1175	0.003	39.356	0.000	0.112	0.123
u_max_cart	0.0658	0.001	51.339	0.000	0.063	0.068
p_reorder_ratio	4.4601	0.022	202.261	0.000	4.417	4.503
p_user	0.6854	0.264	2.593	0.010	0.167	1.203
up_reorder_ratio	0.0056	0.000	17.094	0.000	0.005	0.006
a_reorder_ratio	-0.1677	0.029	-5.717	0.000	-0.225	-0.110
order_dow_1	-0.0024	0.008	-0.300	0.764	-0.018	0.013
order_dow_2	-0.0225	0.008	-2.774	0.006	-0.038	-0.007
order_dow_3	-0.0012	0.008	-0.156	0.876	-0.016	0.014
order_dow_4	-0.0031	0.007	-0.439	0.661	-0.017	0.011
order_dow_5	0.0003	0.006	0.040	0.968	-0.012	0.013
order_dow_6	0.0106	0.006	1.781	0.075	-0.001	0.022
department_babies	-0.2987	0.048	-6.245	0.000	-0.392	-0.205
department_bakery	-0.1953	0.042	-4.597	0.000	-0.279	-0.112
department_beverages	-0.2115	0.041	-5.138	0.000	-0.292	-0.131
department_breakfast	-0.2396	0.044	-5.475	0.000	-0.325	-0.154
department_bulk	-0.3974	0.093	-4.258	0.000	-0.580	-0.214
department_canned goods	-0.0537	0.043	-1.262	0.207	-0.137	0.030
department_dairy eggs	-0.2099	0.041	-5.144	0.000	-0.290	-0.130
department_deli	-0.2198	0.043	-5.151	0.000	-0.303	-0.136
department_dry goods pasta	-0.1530	0.043	-3.556	0.000	-0.237	-0.069
department_frozen	-0.1483	0.041	-3.599	0.000	-0.229	-0.068
department_household	-0.1740	0.043	-4.028	0.000	-0.259	-0.089
department_international	-0.1655	0.050	-3.314	0.001	-0.263	-0.068
department_meat seafood	-0.1900	0.044	-4.354	0.000	-0.276	-0.104
department_missing	-0.4033	0.053	-7.561	0.000	-0.508	-0.299
department_other	-0.2313	0.088	-2.625	0.009	-0.404	-0.059
department_pantry	-0.1432	0.042	-3.422	0.001	-0.225	-0.061
department_personal care	-0.2476	0.046	-5.364	0.000	-0.338	-0.157
department_pets	-0.1169	0.063	-1.855	0.064	-0.240	0.007
department_produce	-0.1329	0.043	-3.121	0.002	-0.216	-0.049
department_snacks	-0.2187	0.041	-5.325	0.000	-0.299	-0.138
part_of_day_Early_Afternoon	-0.0117	0.009	-1.296	0.195	-0.029	0.006
part_of_day_Early_Morning	-0.0021	0.012	-0.175	0.861	-0.025	0.021
part_of_day_Late Evening	-0.0107	0.013	-0.798	0.425	-0.037	0.016
part_of_day_Late_Afternoon	-0.0082	0.009	-0.895	0.371	-0.026	0.010
part_of_day_Late_Morning	-0.0071	0.009	-0.769	0.442	-0.025	0.011
part_of_day_Night	0.0585	0.022	2.666	0.008	0.015	0.101

Figure 31: List of Coefficients for Logistic Regression with Additional Feature Engineering

And here are the features in order of importance as determined by our enhanced model:

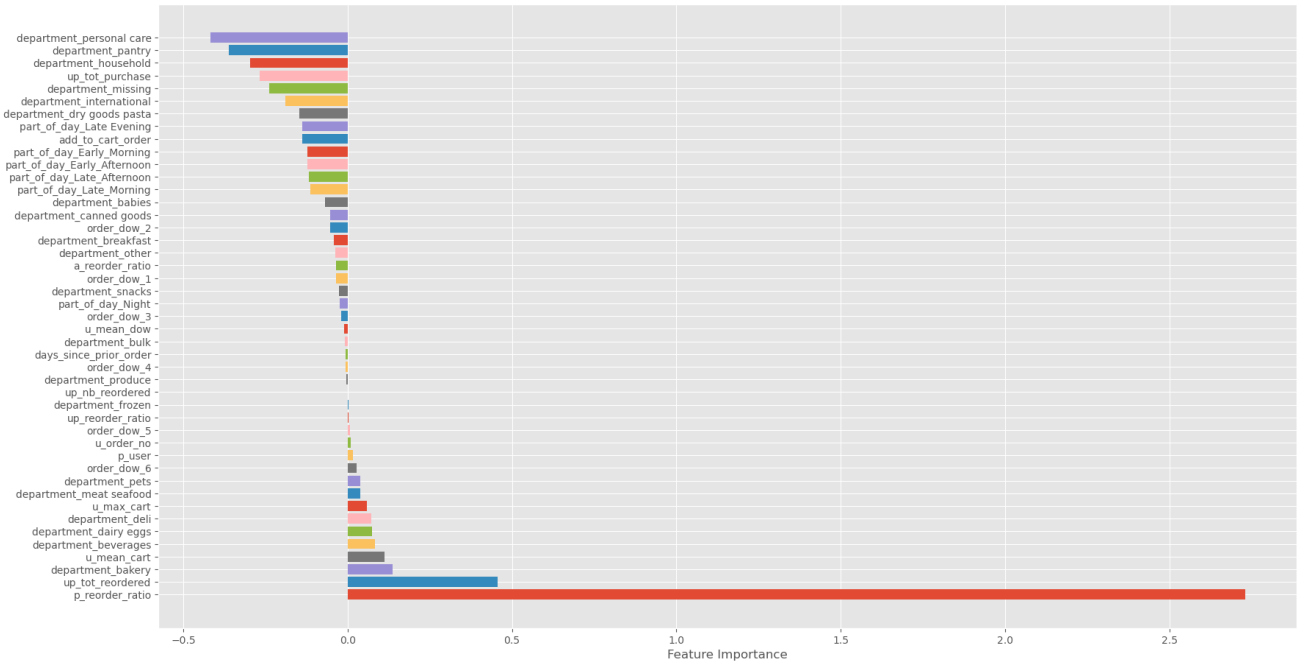


Figure 32: Logistic Regression with Feature Engineering - Feature Importance Plot

Out of the 45 total features resulted from feature engineering, a product's reorder ratio from all users, a user's total number of reorders of a product, and the bakery department features were the 3 most important features in the enhanced logistic regression model. It is observed that the product's reorder ratio was immensely influential and is the largest deciding feature of the model. We predict that the total number of reorders of a product by a user may have been more impactful if the dataset included more order data of users with multiple orders, but since the dataset mostly consisted of data from users who only made one order, that was not the case.

Logistic Regression with Additional Feature Engineering Prediction Results

Our model's prediction results are as follows:

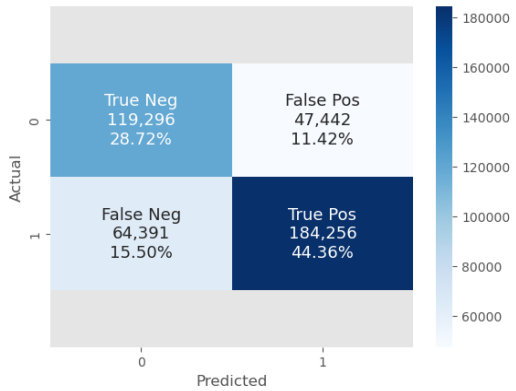


Figure 33: Improved Logistic Regression Confusion Matrix

Our model's accuracy, precision, sensitivity, and specificity scores are as follows:

MODEL TYPE	ACCURACY	PRECISION	SENSITIVITY	SPECIFICITY
LOG REG W/FEAT E	0.7307727	0.7952421	0.7410345	0.7154698

Figure 34: Prediction Metrics of Logistic Regression with Additional Engineered Features

Precision and sensitivity are the two most important metrics for our model, as we want to know if a customer will reorder an item or not. Our enhanced logistic regression model predicts 79 out of 100 reorders and 74 out of 100 reorder predictions were correct. As a result, the new model outperforms the other models due to additional features that we engineered. These features added a meaningful relationship to the response variable.

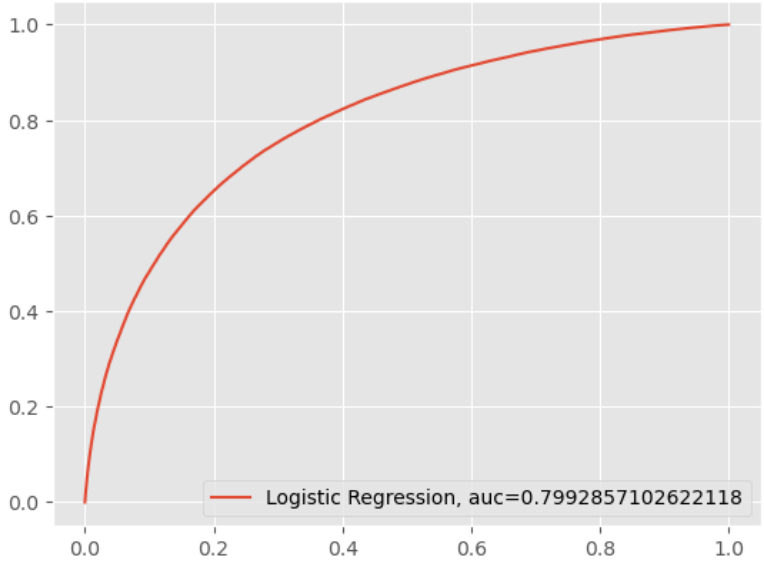


Figure 35: Improved Logistic Regression AUC Curve

Our enhanced logistic regression model yielded an AUC score of 0.799, showing the most improvement over all prior prediction models, including random classifier (0.5), XGBoost (0.647), random forest (0.653), and logistic regression before feature engineering (0.712).

Comparing Logistic Regression Model with Additional Feature Engineering to Earlier Models

MODEL TYPE	ACCURACY	PRECISION	SENSITIVITY	SPECIFICITY	AUC
LOG REG W/AISLE	0.670	0.684	0.835	0.424	0.712
LOG REG W/DEPT	0.663	0.674	0.844	0.392	0.704
RANDOM FOREST	0.681	0.795	0.708	0.626	0.653
XGBOOST	0.678	0.701	0.806	0.488	0.647
LOG REG W/FEAT E	0.731	0.795	0.741	0.715	0.799

Figure 36: Prediction Metrics Across All Classification Models

The overall prediction metrics of the feature-engineered logistic regression model showed great improvement over the other prediction models, which is a result consistent with the increase in AUC score. In 4 out of 5 total metrics, accuracy, precision, specificity, and AUC score, the logistic regression with feature engineering boasts a vastly improved quality in prediction. However, the highest sensitivity score was still achieved by logistic regression using the *department* feature.

Based on the results from the previous section, we imagine that the Random Forest and XGBoost models would have also benefited from the inclusion of these engineered features and believe that it would be beneficial to trial these engineered features in those models in any subsequent steps.

METHODS, RESULTS, AND DISCUSSION: UNSUPERVISED LEARNING

Introduction

One of the ways that retail companies can segment their customers is through RFM analysis of their purchasing behavior. The R stands for *Recency*, and measures the amount of time that has elapsed since a customer last made a purchase, while the F stands for *Frequency*, which measures the total number of purchases that the customer has made, and the M stands for *Monetary Value*, which seeks to capture the total or average amount that a customer has spent with the company [Makhija].

The simplest implementation of RFM analysis involves averaging the three numbers together and then sorting customers from highest to lowest and setting arbitrary thresholds to segment their customers. Our project seeks to use K-Means clustering to perform this segmentation instead.

Dataset

We started off with the original dataset that we used for prediction and grouped by unique users, taking care to only consider the last order placed in the order sequence for each unique user. After conducting the feature engineering described in the subsection below, we ended up with a dataset containing 131,209 data points, one for each unique user, and 3 features, *days_since_prior_order, order_ct,* and *order_avg_size*.

Feature Engineering and Preprocessing

Because the data required for clustering was unrelated to the data, preprocessing, or feature engineering that we conducted for prediction, we needed to perform additional feature engineering in order to conduct our RFM analysis using K-Means clustering.

The Instacart dataset provided the feature *days_since_prior_order*, which we used for *Recency*, however there were no features in the dataset that we use for *Frequency* and *Monetary Value*. For *Frequency*, we calculated the total number of orders made by each customer and created the feature *order_ct*. For *Monetary Value*, we decided that the number of items purchased could serve as a proxy for total spend since the dataset did not include any pricing information. To accomplish this, we created the feature *order_avg_size* by creating several intermediate features and then calculating the average number of items a customer purchased across their orders.

Prior to building the K-means model, we wanted to see what our newly created features looked like and created density plots for each of the features:

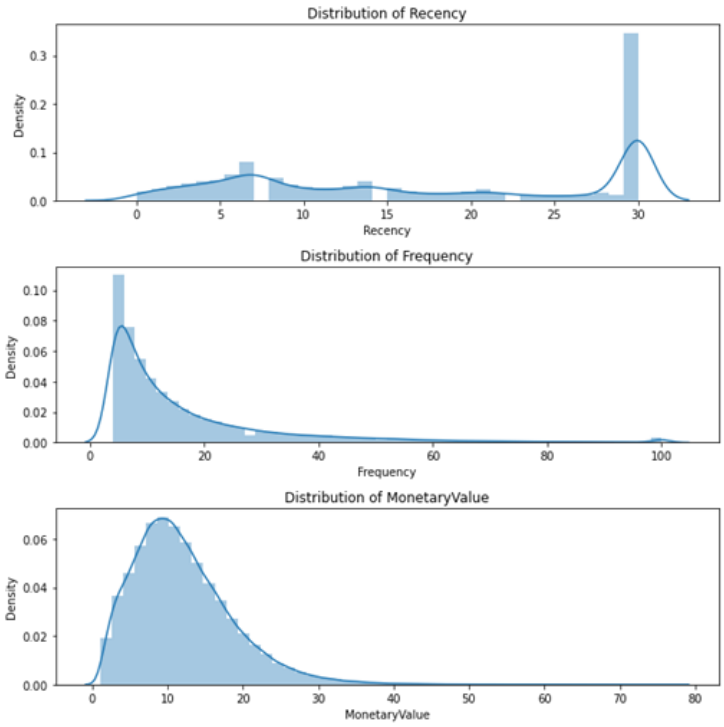
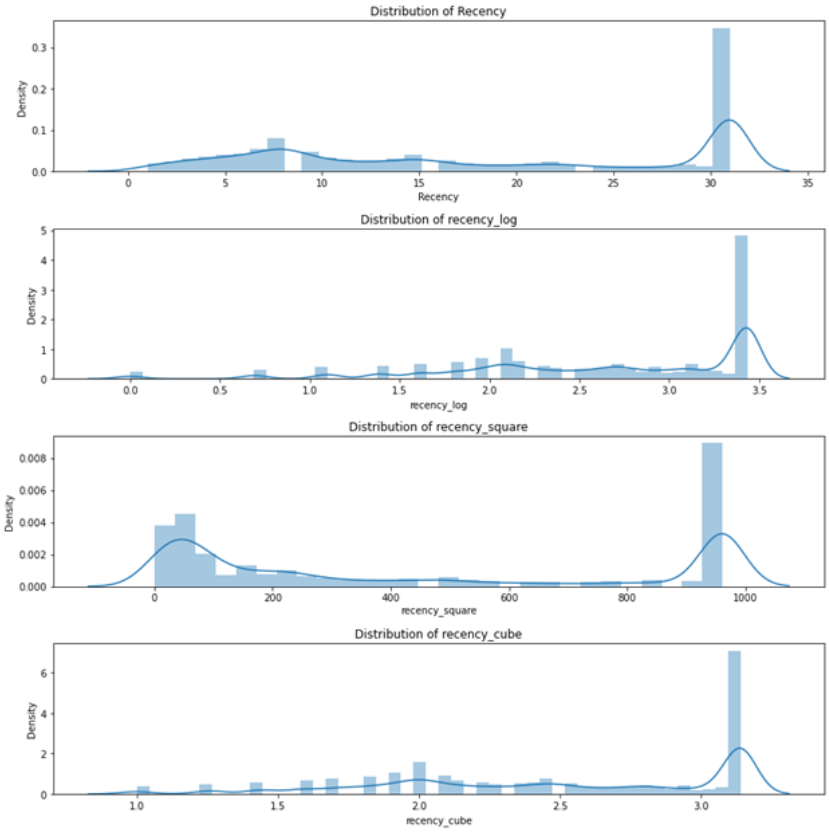


Figure 37. Distribution Plots of Recency, Frequency, Monetary Value

We can see from the plots that *Recency* (number of days since the last order) is heavily skewed to the left, where the majority of customers have not placed an order in 30 or more days, while *Frequency* (total number of orders) and *MonetaryValue* (average order size) are heavily skewed to the right, which indicate that most customers placed a total of 5 or less orders, and purchased around 10 items per order.

To reduce the heavy right skew of *Recency*, we evaluated square, cube root, and log-transforms versus the untransformed data to determine if any of the transformations would be an improvement by visualizing the distributions as well as the evaluating the skewness using the sciPy *skew* function:

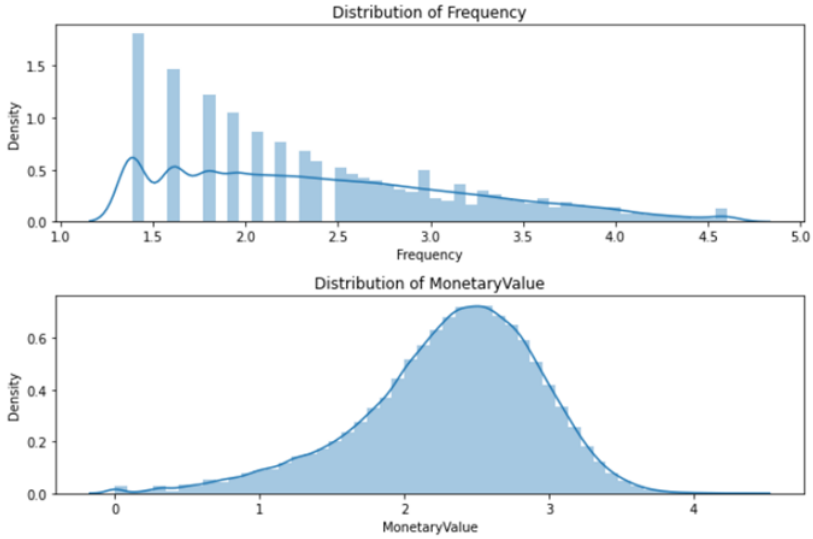


RECENCY TYPE	SKEW
Untransformed	0.038538
Log	-0.938108
Square	0.341162
Cube	-0.449349

Figure 38 – Transformation Distribution Plots and sciPy skew Values for Recency

Based on visual plotting of the distributions, the transformations do not appear to offer much of an improvement over the untransformed data. This is confirmed by the values generated by sciPy's skew function, where a value of zero indicates normal distribution. The untransformed *Recency* data exhibits a skew closest to normal, while the transformed data exhibits more skewness to varying degrees. Because of this, we will utilize the untransformed *Recency* data for the K-Means clustering model.

For the *Frequency* and *Monetary Value*, we performed the log-transformation and compared the results to the non-transformed data using plots of the distributions along with comparing the values generated by the sciPy *skew* function:



DATA TYPE	SKEW
Untransformed Frequency	2.393478
Log-Transformed Frequency	0.554102
Untransformed MonetaryValue	1.118745
Log-Transformed MonetaryValue	-0.697962

Figure 39: Distribution Plots of Log-Transformed Frequency and Monetary Value and table of sciPy *skew* values for Untransformed and Log-Transformed Frequency and MonetaryValue.

Comparing the distribution plots and skew values between the untransformed data and the transformed data, we can see that the transformed data for both *Frequency* and *Monetary Value* exhibit less skewing than the untransformed data.

Based on the results above, we selected the untransformed *Recency* data, along with the log-transformed *Frequency* and *MonetaryValue* data, to standardize and run through the K-means model.

Determining the Optimal Number of Clusters

After standardizing the data, we created elbow plots, generated silhouette scores, and silhouette plots to determine the optimal number of clusters.

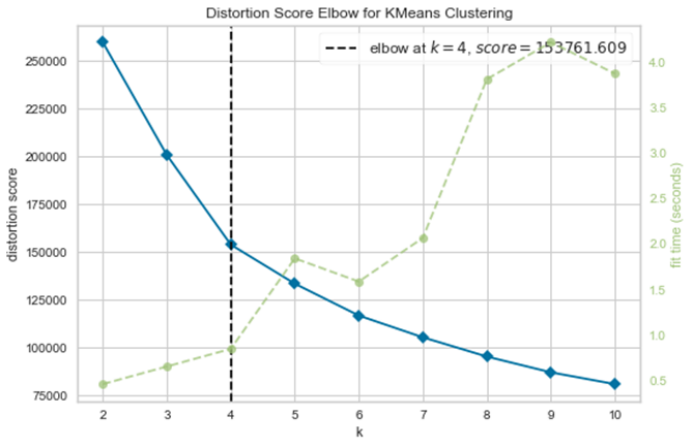


Figure 40: Elbow Plot to Determine Optimal K-Means Clusters

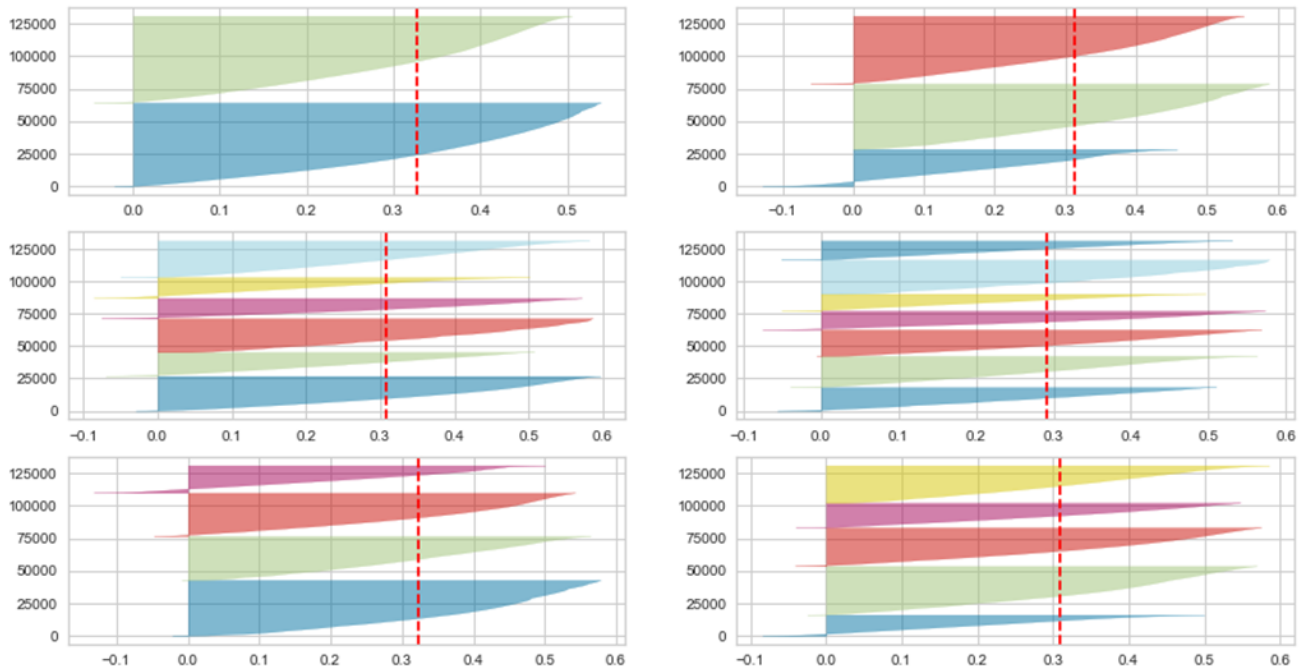
Based on the elbow plot that we generated using YellowBricks, the optimal number of clusters for this dataset is 4.

NUMBER OF CLUSTERS	SILHOUETTE SCORE
2	0.326
3	0.312
4	0.324
5	0.309
6	0.307
7	0.291
8	0.281
9	0.276
10	0.273

Figure 41: Silhouette Score for 2-10 Clusters

Silhouette scores range between -1 and 1, with -1 indicating that means clusters are assigned incorrectly, 0 indicating that the distance between clusters is not significant, and 1 indicating that the means clusters are well separated from one another [Bhardwaj]. Based on the silhouette scores that we generated, it appears that the optimal number of clusters is 2, followed closely by 4, as these had the highest score of .326 and .324 respectively.

Figure 42: Silhouette Plots for 2-7 Clusters



The Silhouette plots for 2-7 clusters indicate that the optimal number of clusters is 2. One of the reasons for this is because all clusters within this plot exceed the dotted red line, which indicates the average Silhouette score. Furthermore, the clusters in the 2-cluster plot are uniform in thickness, whereas the other plots exhibit non-uniform thickness. Finally, the 2 clusters in the 2-cluster plot are relatively equal in length, whereas the clusters in the other plots are less equal in length.

K-Means Clustering Results

Because of the conflicting results regarding the optimal number of clusters between the elbow plot and the Silhouette Scores/plots, we decided to run K-Means for 2-5 clusters and generate t-SNE plots to evaluate the separation between the clusters generated by K-Means:

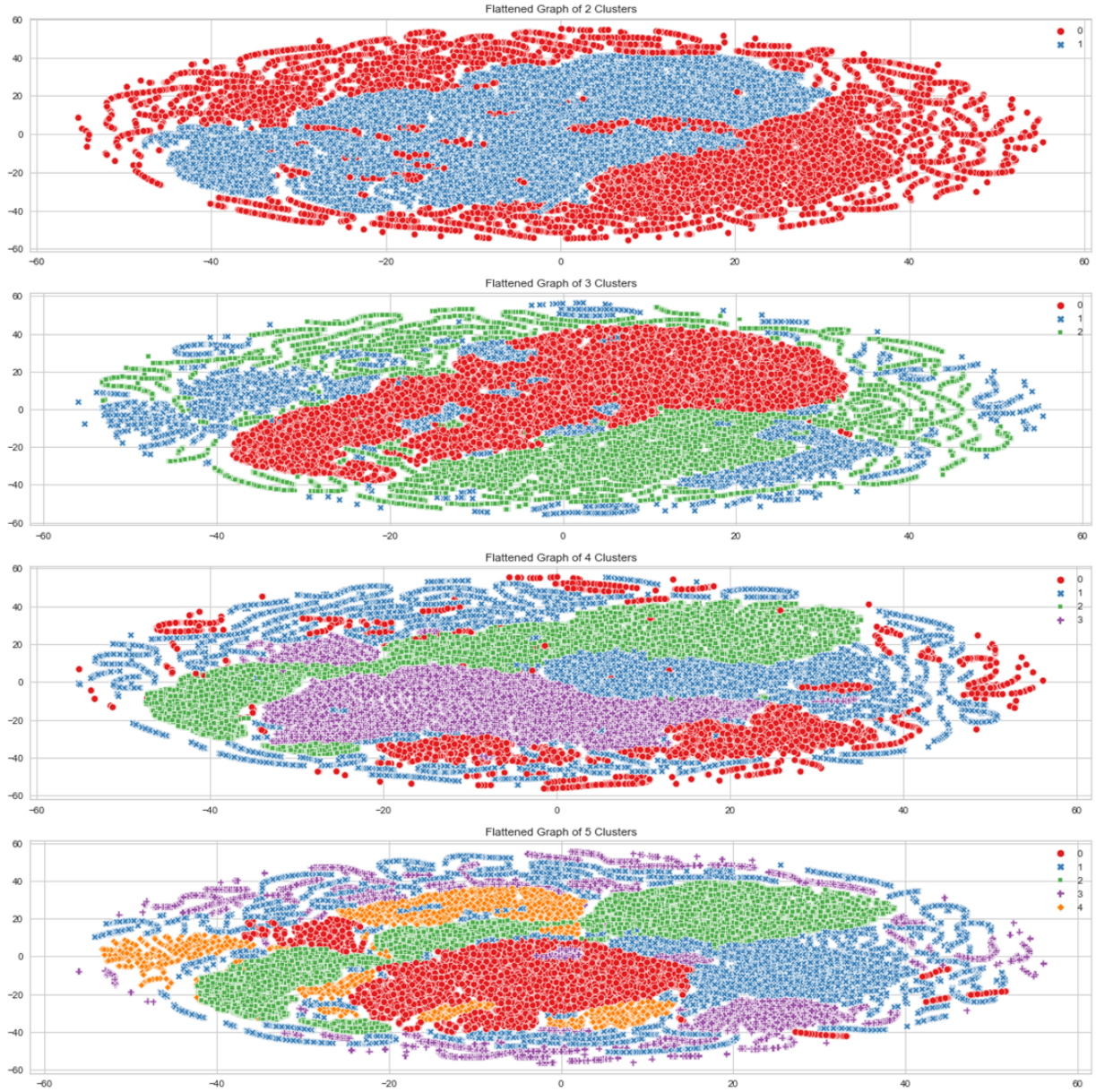


Figure 43: t-SNE plots of K-Means Results for 2-5 Clusters

t-SNE plots compress high-dimensional data into 2-D plots, which we can use to visualize the formation and separation of the clusters. While not perfect, K-Means with 2 clusters, which was the optimal number of clusters according to Silhouette score and plot, appears to do the best job at forming clusters and separation.

Segmentation Snake Plots

To better understand the characteristics of the customers that form these clusters, we created a snake plot for K-Means with 2 clusters, a tool that is commonly used by the marketing industry to understand their segments:

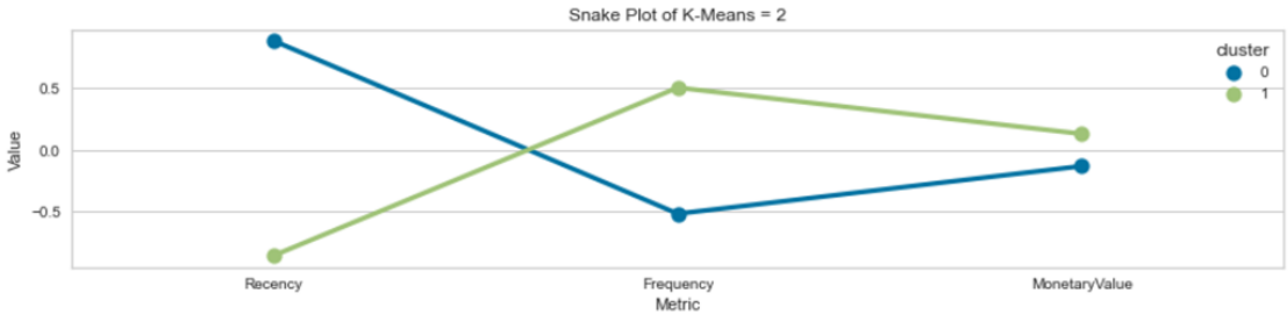


Figure 44: Snake Plot for K-Means = 2

Based on the snake plot, we can see that the customers in cluster 0 did not purchase anything in a long time, have a low overall count of orders, and purchased less items per order on average, whereas the customers in cluster 1 made an order recently, have made a large number of orders, and tend to purchase more items per order on average.

Using the results from the snake plot, Instacart could come up with tailored marketing strategies for each cluster in order to drive sales and achieve other internal KPIs.

CONCLUSION

We collected the data from Instacart's Online Grocery Shopping Dataset from 2017 and performed data cleaning by checking for any missing information. Next, we combined all the data into a single table and performed data analysis to understand the statistics of the dataset in order to perform feature selection for our prediction model. For our prediction model, we first built a supervised model using logistic regression to predict the reordering of individual products. We selected certain features by using Cramer V to find the best correlation between each feature and the response, and the correlation among pairs of features. The most impactful features were chosen and those with high correlation with the already selected features were omitted. Our logistic regression model with feature engineering proved to be especially successful with an AUC score of 0.799, which is a significant improvement over the XGBoost (0.647), the random forest (0.653), and the regular logistic regression (0.712) models. Since we did not use the later engineered features for the Random Forest and XGBoost models, that would be our next step in improving the results. Improved accuracy in

predicting what customers reorder can allow Instacart to have better insight for product recommendations, inventory, and what products to offer.

As our final task, we further explored the dataset using an unsupervised method of customer segmentation. Our k-means model allowed for distinguishing customer segments based on timeline of purchase orders, count of orders, and average orders, leading to a successful implementation of our unsupervised prediction model. Future steps to further explore the segmentation of customers could be using other unsupervised learning models such as Gaussian Mixture Model or Hierarchical Clustering, or using different features in the k-means model. Instacart can use these improving customer segments and tailor their marketing strategies accordingly, as mentioned previously.

Gantt Chart:

https://github.com/instakartproject/cs7641/blob/main/CS7641_Fall2022_Group24_GanttChart.xlsx

Updated Contribution table:

TASK TITLE	**TASK OWNER**
Final Proposal Paper	All
Introduction & Background	EH, NF
Problem Definition	NF, EH
EDA Methods	MC, EH
Supervised Method Results & Discussion	NF, MC, DJ, PV, EH
Supervised (Bonus) Logistic Regression with Feature Engineering	PV, DJ
Unsupervised Results & Discussion	EH
Conclusion	DJ, PV, NF, EH
Final Presentation	MC, NF
GitHub Page	EH, MC

Citations:

Bhardwaj, A. (2020, May 27). *Silhouette coefficient : Validating clustering techniques*. Medium. Retrieved December 2, 2022, from https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c

Briedis, H., Kronschnabl, A., Rodriguez, A., & Ungerman, K. (2020). *Adapting to the next normal in retail: The Customer Experience Imperative*. McKinsey & Company. Retrieved October 4, 2022, from https://www.mckinsey.com/industries/retail/our-insights/adapting-to-the-next-normal-in-retail-the-customer-experience-imperative

Cikovic, K. F. (2020). Customer Profiles in the Antiques and Collectibles Industry in Croatia Using Gaussian Mixture Model Clustering: An Empirical Study. *Economic and Social Development: Book of Proceedings*, 148-156.

Covid-driven recession impact on retail industry. Deloitte United States. (2020). Retrieved October 4, 2022, from https://www2.deloitte.com/us/en/pages/consumer-business/articles/retail-recession.html

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD)* (pp. 21-34).

Jiang, P., Zhu, Y., Zhang, Y., & Yuan, Q. (2015). Life-stage prediction for product recommendation in e-commerce. In Proceedings of the 21th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1879-1888).

Kashwan, K. R., & Velu, C. M. (2013). Customer segmentation using clustering and data mining techniques. *International Journal of Computer Theory and Engineering*, 5(6), 856.

Khalili-Damghani, K., Abdi, F., & Abolmakarem, S. (2018). Hybrid soft computing approach based on clustering, rule mining, and decision tree analysis for customer segmentation problem: Real case of customer-centric industries. *Applied Soft Computing*, *73*, 816-828

Lawson, C., & Montgomery, D. C. (2006). Logistic regression analysis of customer satisfaction data. Quality and reliability engineering international, 22(8), 971-984.

Makhija, P. (2021, June 3). *RFM analysis for Customer SegmentationPushpa Makhija*. CleverTap. Retrieved December 2, 2022, from https://clevertap.com/blog/rfm-analysis/

Dataset:

"The Instacart Online Grocery Shopping Dataset 2017", Accessed from https://www.instacart.com/datasets/grocery-shopping-2017 on <date>