InstaCart & Machine Learning

A Not-So "Insta" Analysis

Madison Dimaculangan
June 2020

01 Introduction

What is InstaCart?
Concepts and goals.

02 Model Preparation

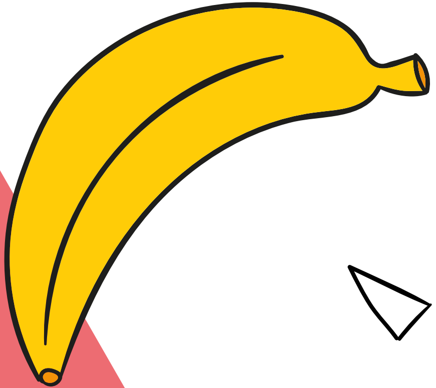
Dataset transformation.
Data exploration.

03 Unsupervised Learning

Using clustering to find patterns in the data.

04 Supervised Learning

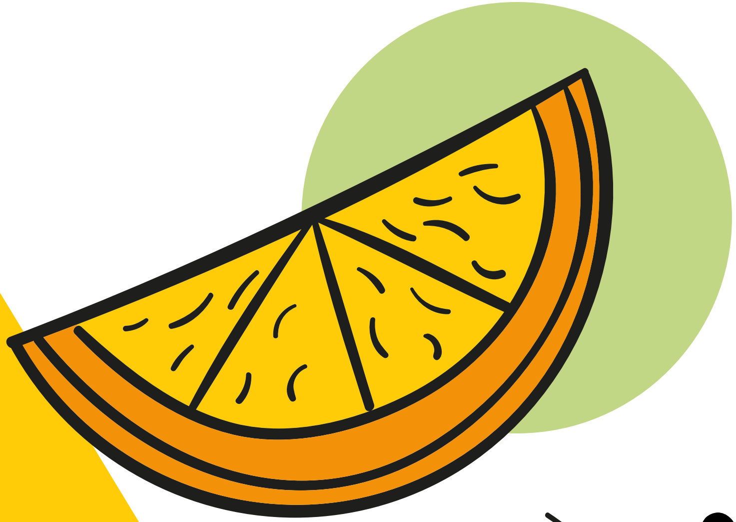
Using algorithms to make predictions.



01

Introduction

Background and goals.



What is Instacart?

- alternative to traditional grocery experience
- on-demand grocery delivery service
- presence in 5,500 cities in US & Canada
- employs personal shoppers to fulfill & deliver
- partnerships with 350 retailers (over 25,000 locations)

Project Goals

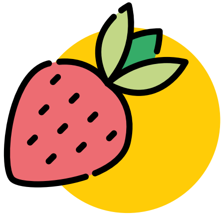
The main goal of this project is to determine whether machine learning techniques can be applied to the Instacart dataset to suggest new features to improve user experience.

This can be accomplished by learning more about the users themselves:

- Can the user population be divided into groups of users with similar characteristics?
- Can we make any predictions on future purchases based on the ordering history?
- Are there particular products that users are ordering more often than others?

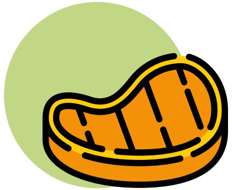
These and other questions, once answered, can lead to actions that allow for increased customer retention and product usage.

What Is Machine Learning?



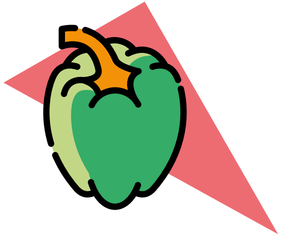
Machine Learning

Computer algorithms that learn and improve from experience without being explicitly programmed.



Unsupervised

Type of machine learning that searches for patterns in a data set with no pre-existing labels.



Supervised

Type of machine learning that predicts an output given a set of inputs based on example input-output pairs

Data Set

The data used for this project is from the 2017 competition that Instacart hosted on Kaggle.com.

The Instacart Market Basket Analysis competition data set consists of five csv files:

1. aisles.csv
2. departments.csv
3. orders.csv
4. products.csv
5. order_products_*_.csv (where * = [prior, train])

The Model Preparation section of this slide deck will detail how the various files will be transformed and merged to create our model.

02

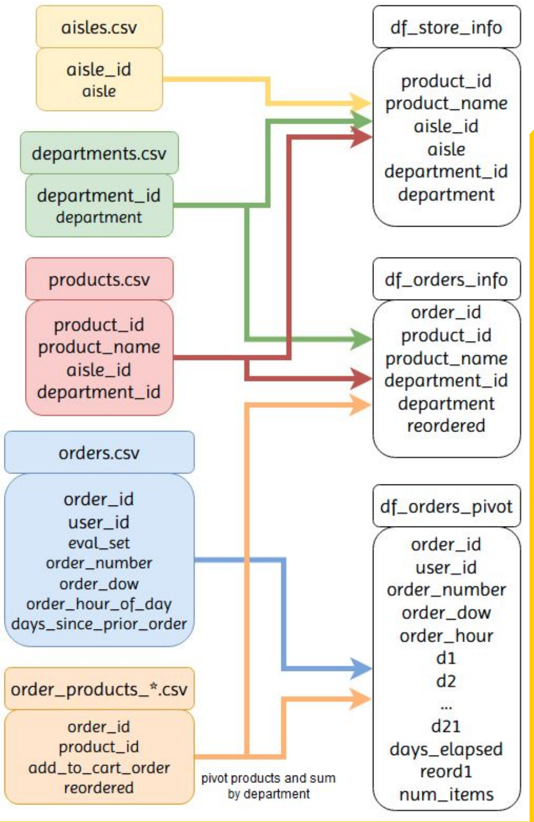
Model Preparation

Data merging and visualization.

Dataset Merging

Using PySpark and SQL commands:

- from 6 csv files, created 3 dataframes to better associate order, user, and product information
- grouped products by their departments and recorded the sums as separate columns
- grouping and aggregating reduced observation count ten-fold, from ~32 million to ~3.2 million
- added a new column summing the number of items ordered per order_id



Dataset Inspection

	Details
aisles	134 unique aisles, including 1 for “missing”
departments	21 unique departments
products	49,687 unique products
orders	3,421,083 unique orders
users	206,209 unique users

Aisles

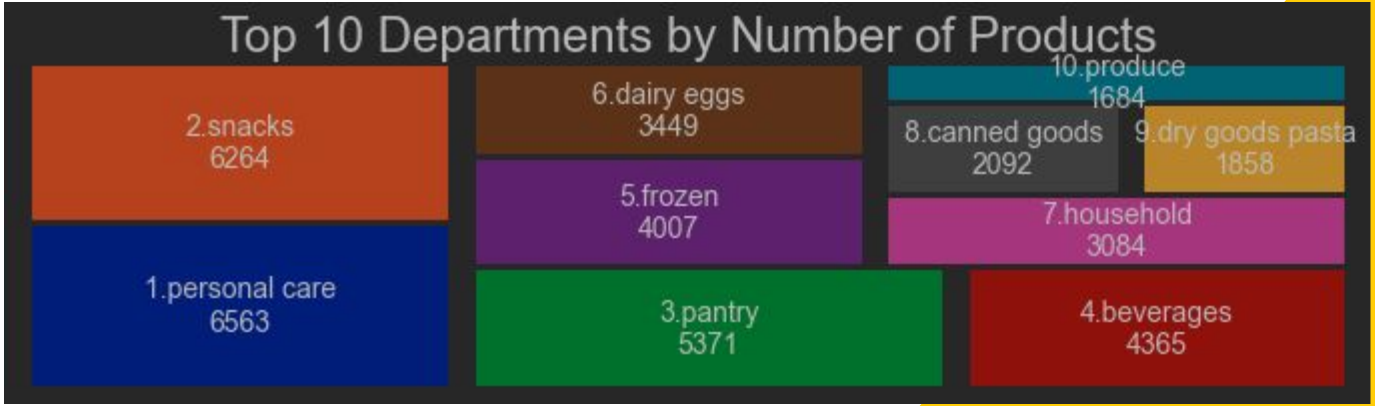
In total, there are 133 unique aisles (plus 1 additional “aisle” for uncategorized products).

Below is a a graphic showing the 10 aisles with the greatest number of unique products.

Departments

In total, there are 21 unique departments.

Below is a a graphic showing the 10 departments with the most number of products.

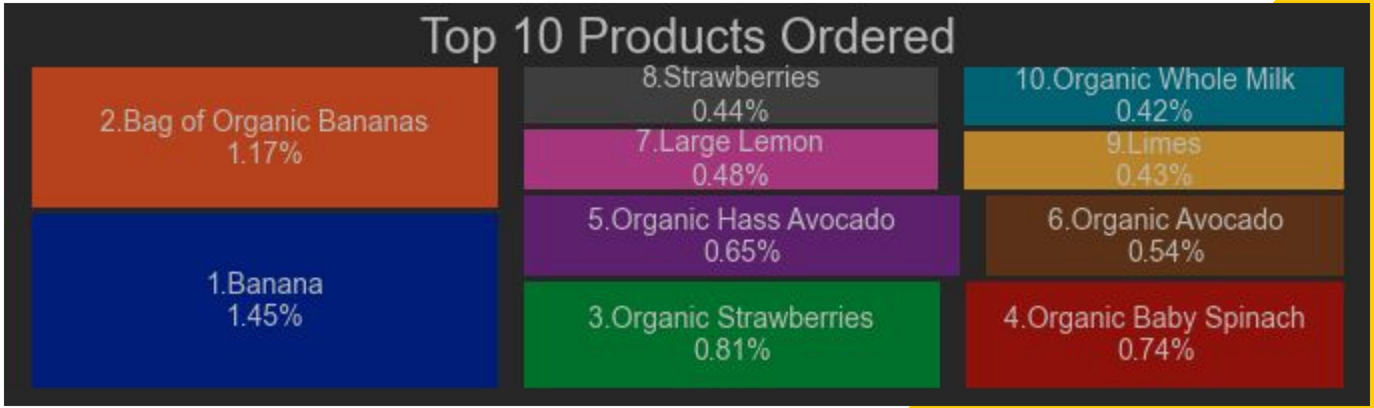


Orders- by products

In total, there are 3,346,083 orders.

Below is a a graphic showing the 10 products ordered by users.

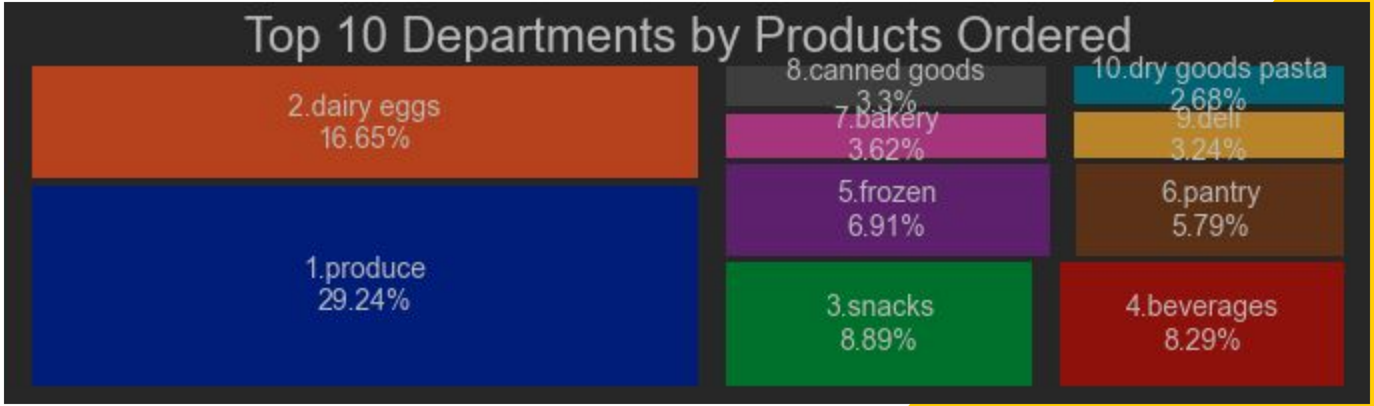
Instacart users love bananas!



Orders - by departments

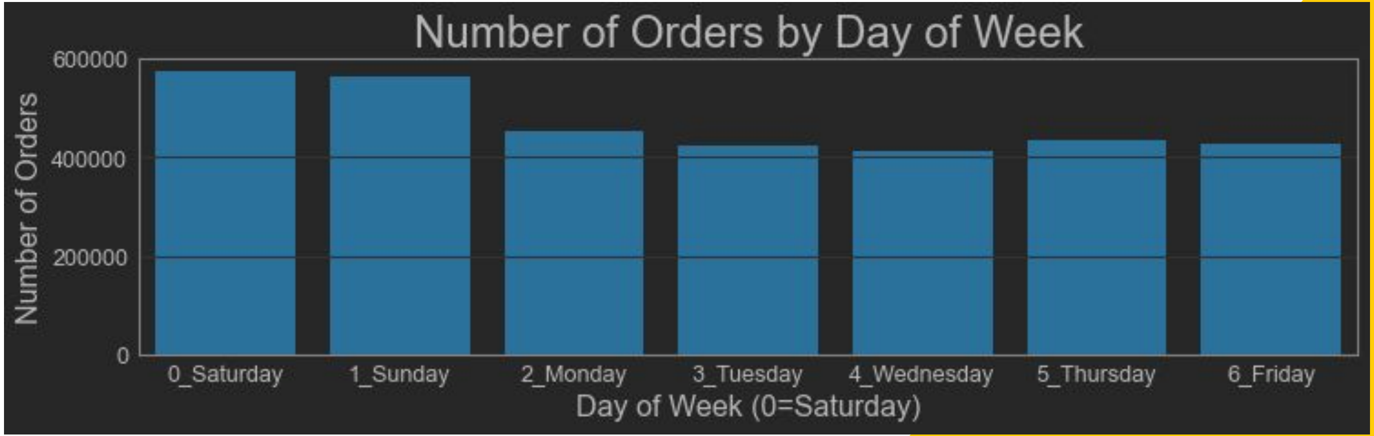
Below is a graphic showing the 10 departments with the most products ordered.

Perishable items are the most ordered with produce and dairy items and eggs accounting for over 45% of all products ordered.



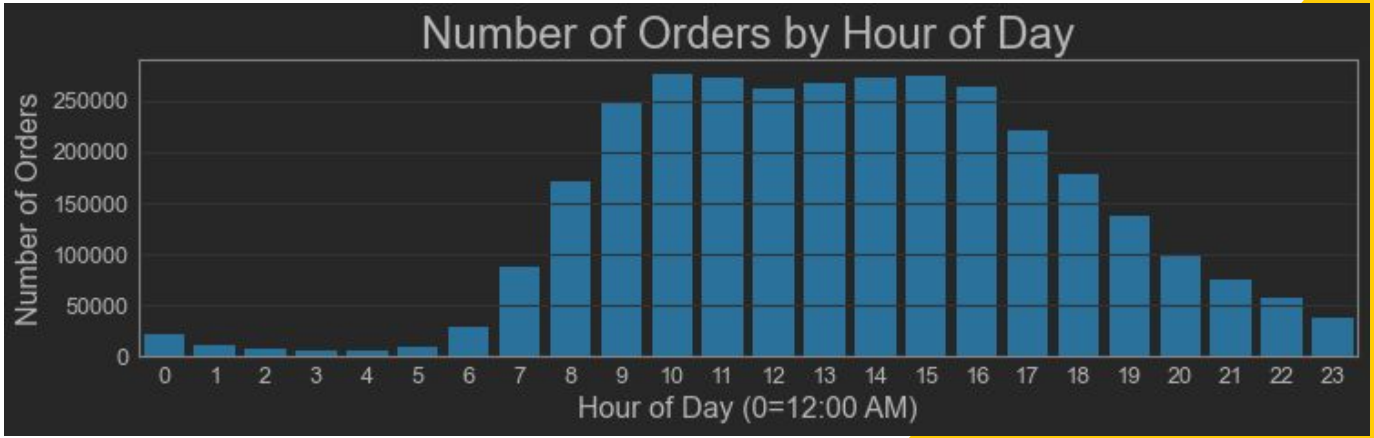
Orders – By day of week

- Orders are most often placed during the weekends
- The number of orders decrease towards the middle of the workweek.



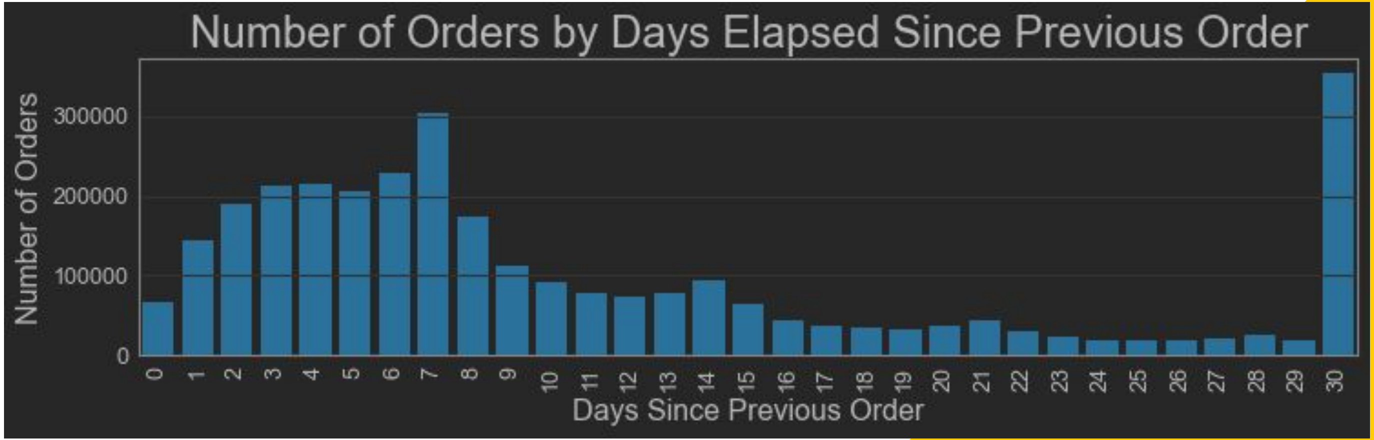
Orders - By hour of day

- The peak hours are between 10 AM and 4 PM.
- The number of orders decrease gradually until midnight.
- There is a dead period of low number of orders between midnight and 6 AM.



Orders - By days since previous order

- The lag between orders ranges from 0 to 30 days.
- Excluding first orders, there are several distributions centered around:
    - every 3-4 days
    - every week
    - every other week
    - longer than 3 weeks

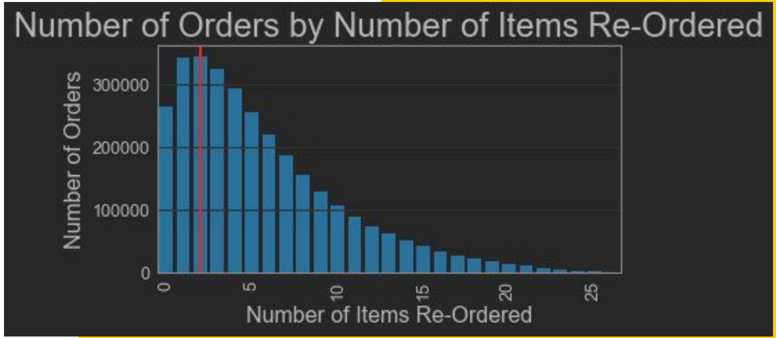


Orders - By Order Number

- order_number indicates which (1st, 2nd, 3rd, etc.) order it is for the customer
- 25% of all orders places occur within the first 4 orders
- after 4 orders there is a dropoff, an opportunity for retention improvement

Orders – By number of items ordered/re-ordered

- The number of items ordered is distributed around 5 items with tail of up to 26 items
- The number of items re-ordered is distributed around 2 items with a tail up to 26 items



Correlations

- d4 (produce) and d16 (dairy/eggs) correlate the best with total number of items ordered, which is not surprising given that we saw these two departments as having the most number of items ordered

- no variable correlates well with the order lag time, days_elapsed

	index	corr
22	num_items	1.000000
21	reord1	0.731229
3	d4	0.653757
15	d16	0.593018
18	d19	0.401165
0	d1	0.390361
12	d13	0.381021
2	d3	0.353039
14	d15	0.346875
19	d20	0.335703

	index	corr
23	days_elapsed	1.000000
0	d1	0.033226
16	d17	0.032168
8	d9	0.026241
14	d15	0.022098
11	d12	0.016751
22	num_items	0.016646
10	d11	0.015481
19	d20	0.015340
13	d14	0.011909

03

Unsupervised Learning

Clustering analysis.

Model Definition – feature selection

1. grouped data by user_id
2. aggregated each feature as shown in the table below

	Features	Details
sum	d1, d2, d3,...,d21	indicates the number of products ordered from each department
mean	num_items	indicates the number of items ordered
	reord1	indicates the number of items re-ordered
	days_elapsed	indicates the number days since the previous order
last	order_number	Indicates the number of orders

Model Definition - data transformation

Several features were transformed as shown in the table below:

	Features	Transformation
Sparse data	d1, d2, d3,...,d21	PCA dimensionality reduction First 5 components used (~53% of variance)
Skewed data	num_items	log transformation
	reord1	
	days_elapsed	
	order_number	

Selecting the Algorithm

Algorithm

The clustering algorithms below were tested to determine the optimal clusters.
The k-means algorithm resulted in the greatest similarity scores.

- hierarchical clustering
- gaussian-mixture model
- DBSCAN
- **k-means** → best silhouette analysis results!

Choosing k

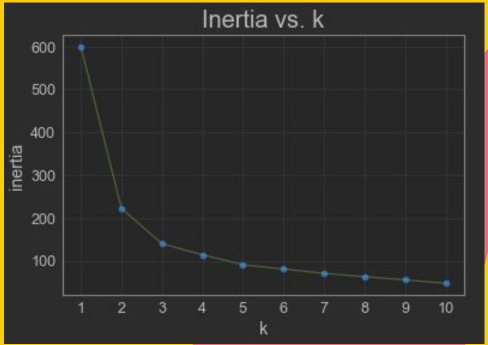
The number of clusters, **k**, was chosen based on the following methods:

- **Elbow method** - indicated values of 4, 5, or 6 might be suitable
- **Silhouette analysis** - indicated that 4 clusters would result in the greatest similarity score amongst the options

Elbow Method

The elbow method plots the **inertia**, the sum of squared distances of the samples from the cluster centers, vs. the number of clusters, **k**.

The optimal **k**, is the value at which the rate of decrease in inertia becomes more linear. Visually, this appears as the bend in the plot, much like the elbow of a bent arm, hence the name.



Silhouette Analysis

The **silhouette coefficient** measures the similarity of data points within a cluster with one another. It is calculated as follows:

$$\frac{b_i - a_i}{max(b_i, a_i)}$$

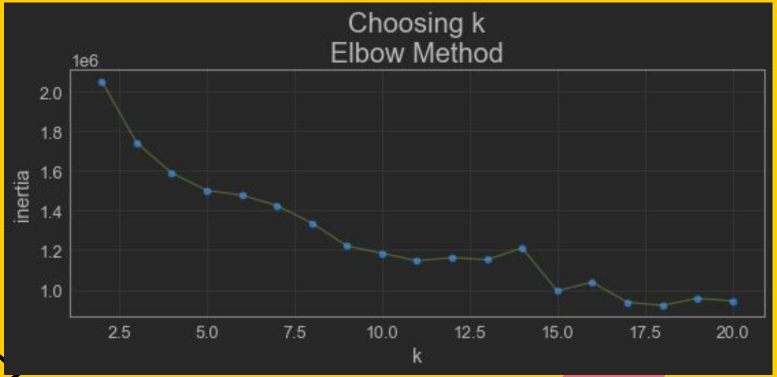
where for each data point **i**,

- a_i = mean distance between **i** and all data points in its cluster
- b_i = mean distance between **i** and all data points in neighboring clusters

The **silhouette average** is the average of all silhouette coefficients of all of the data points.

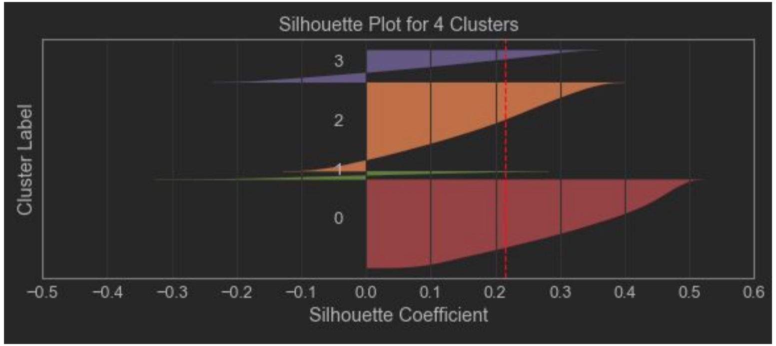
Elbow Method

(results)



Silhouette Analysis

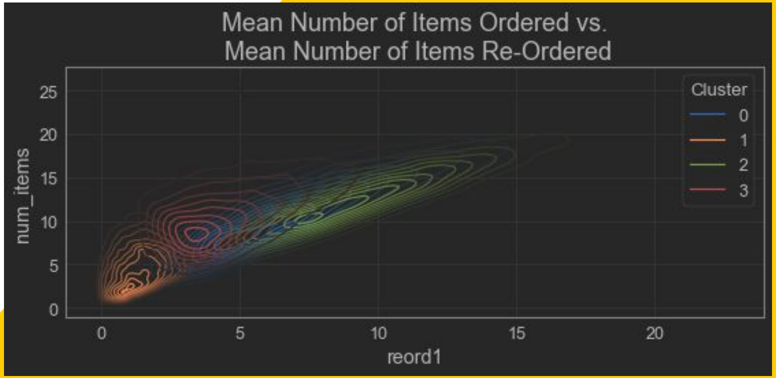
(results)



Clustering Evaluation - number of items

Observations:

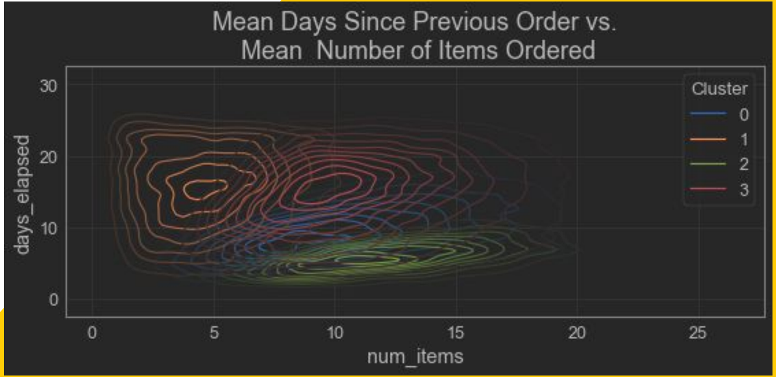
- **Cluster 1** users on average ordered and re-ordered the fewest number of items.
- **Cluster 2** users on average ordered and re-ordered the greatest number of items.



Clustering Evaluation - days elapsed

Observations

- **Cluster 1** users ordered the fewest items and had a wide range of order lag
- **Cluster 2** users place orders with the least lag and had a wide range of number of items ordered.
- **Cluster 0** and **Cluster 3** have similar ranges of number of items ordered; however, Cluster 3 users have less order lag than Cluster 0 users.

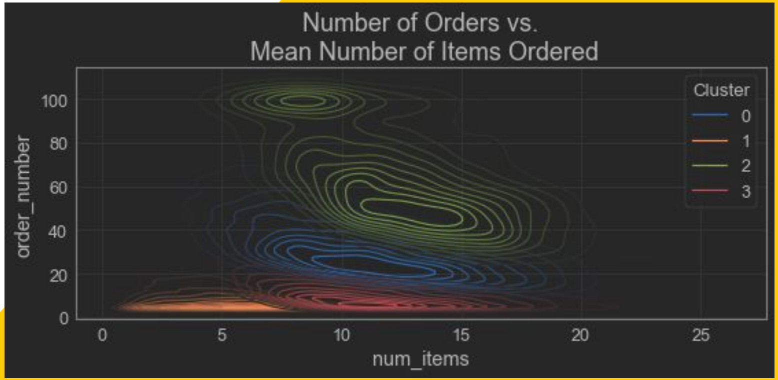


Clustering Evaluation – number of orders

Observations

- **Cluster 1** and **Cluster 3** users placed the fewest number of orders, though Cluster 3 users ordered more items on average

- **Cluster 2** users placed the greatest number of orders, though there is two densities of users within this cluster
    - Users that placed ~100 orders
    - Users that placed ~30-~80 orders

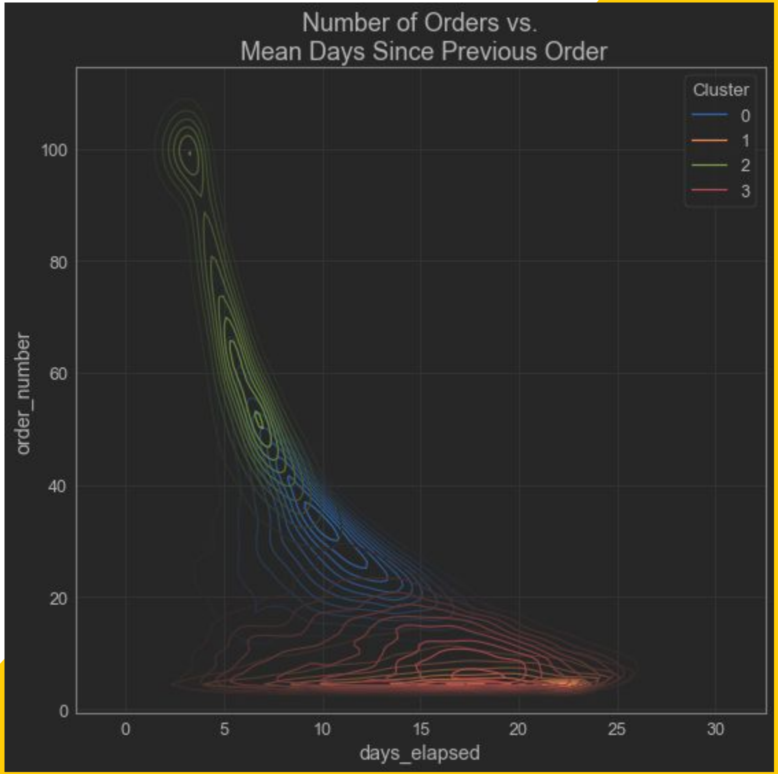


Clustering Evaluation – number of orders

Observations

- For **Cluster 0** and **Cluster 2** users, there is a negative correlation between the number of orders placed and the lag time between orders

- No such correlation is present for **Cluster 1** and **Cluster 3**



Cluster Evaluation - summary

	Summary
Cluster 0	Semi-frequent shoppers who order a moderate number to many items
Cluster 1	Occasional shoppers who order very few items
Cluster 2	Frequent shoppers who order a moderate number to many items
Cluster 3	Occasional shoppers who order a moderate number of items

04

Supervised Learning

Predicting order frequency.

Model Definition – feature selection

1. grouped data by user_id
2. aggregated each feature as shown in the table below
3. days_elapsed, a continuous variable, selected as the output variable, requiring **regression**

	Features	Details
sum	d1, d2, d3,...,d21	indicates the number of products ordered from each department
mean	num_items	indicates the number of items ordered
	reord1	indicates the number of items re-ordered
	days_elapsed	indicates the number days since the previous order
last	order_number	indicates the number of orders

Model Definition - data transformation

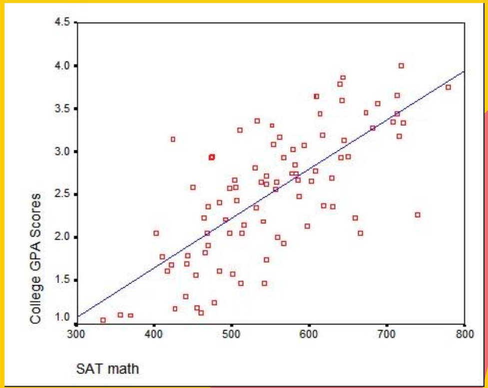
Due to issues faced, several features were transformed as shown in the table below:

	Features	Transformation
sparsity	d1, d2, d3,...,d21	PCA did not appear to impact regression so for final results, PCA was skipped.
	num_items	
	reord1	
skew	days_elapsed	log transformation
	order_number	

Regression

Type of supervised learning in which the task is to predict the values of a **continuous** outcome variable.

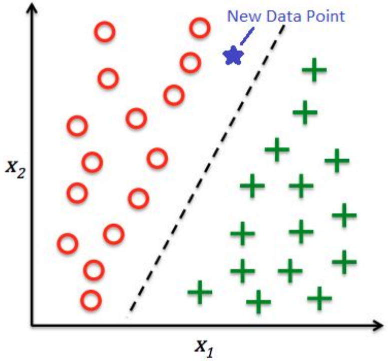
A continuous variable, such height or revenue, can take on an infinite number of values. In regression, we are trying to **quantify**.



Classification

Type of supervised learning in which the task is to predict the values of a **categorical** outcome variable.

A categorical variable, such as hair color or car model, can take on only a limited number of values. In classification, we are trying to **select**.



Selecting the Algorithm

Algorithm

The supervised learning algorithms below were tested for sample sizes of up to 50%. The *k*-nearest-neighbor regressor was chosen as it resulted in the lowest RMSE.

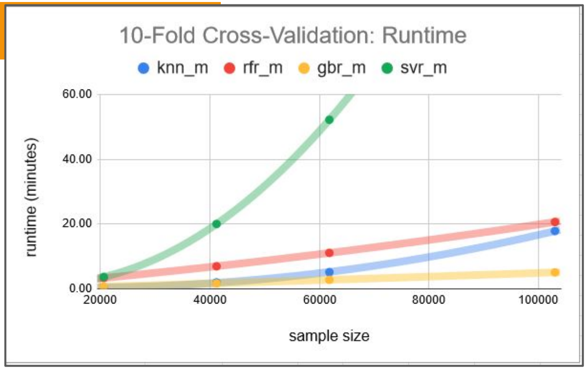
- random forest
- gradient boosting
- support vector machine
- **knn** → lowest root mean squared error!

Parameters

Searches were performed to determine the optimal parameter values.

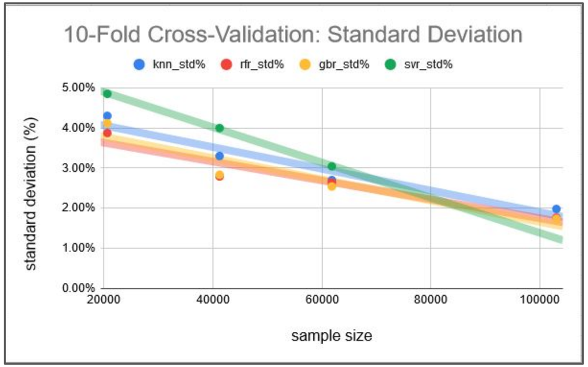
- `n_neighbors` = 50
- `weights` = 'distance'
- `algorithm` = 'ball tree'

Comparing Algorithms



Runtime

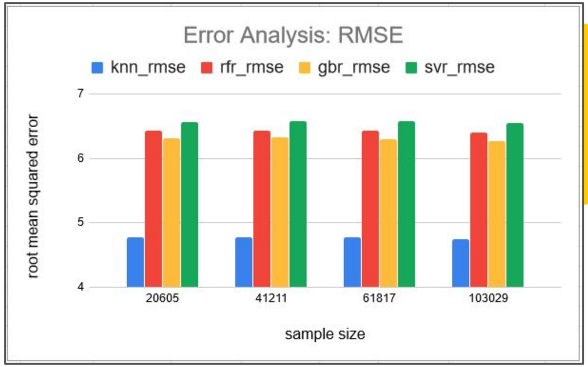
Support vector machine scales the worst with sample size while gradient boosting scales the best.



Standard Deviation

Standard deviation across folds decreases with sample size.

Similar values across algorithms (1.25% - 1.98%) at 50% sampling rate



RMSE

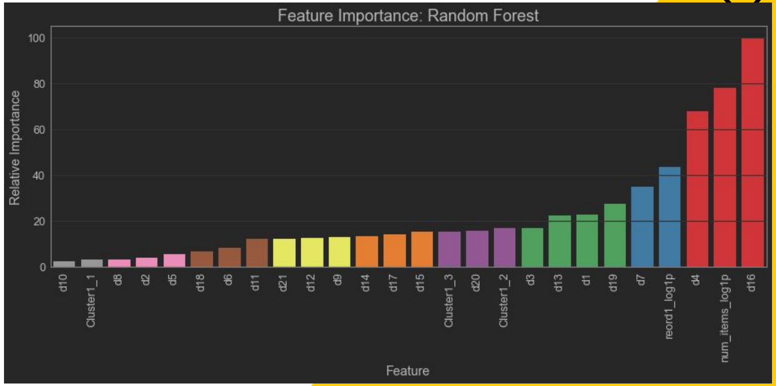
RMSE consistent across sample size with KNN algorithm having the lowest error.

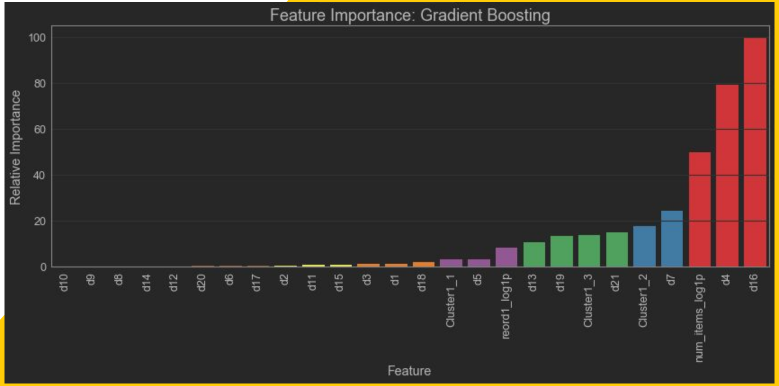
Error Evaluation – feature importance

Observations:

- Notably, the most important features includes the departments with the most purchases (d16, d4)

- While the order is slightly different, both the random forest and gradient boosting algorithms consider d16, d4, and num_items (log transformed) as the 3 most important features.



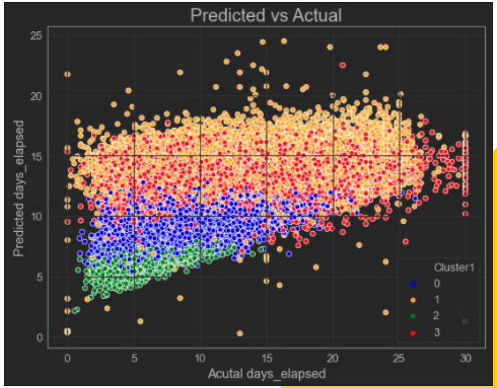


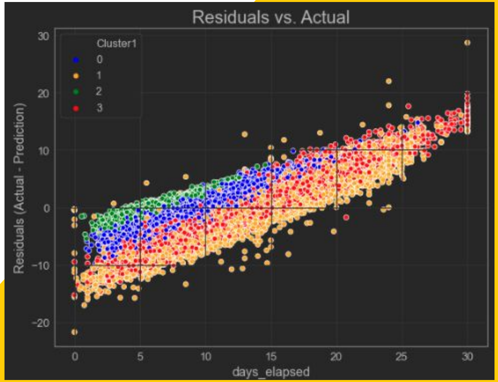
Error Evaluation - residuals

Observations:

- **Cluster 2** and **Cluster 0** appear to have a slight linear relationship between predicted and actual values of order lag time (days_elapsed)

- **Cluster 2** and **Cluster 0** also appear to have a narrower band of residual values relative to Cluster 1 and Cluster 3

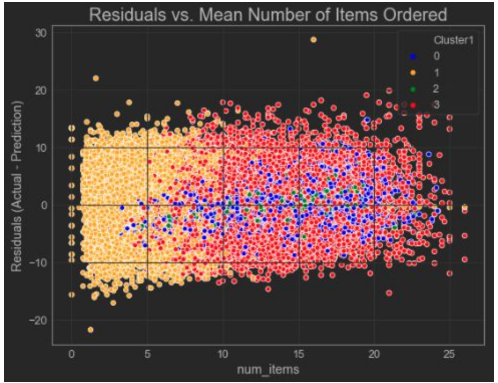


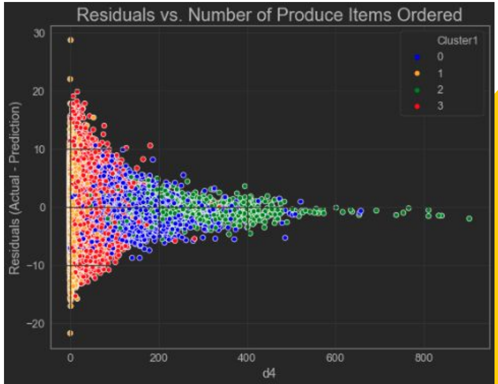


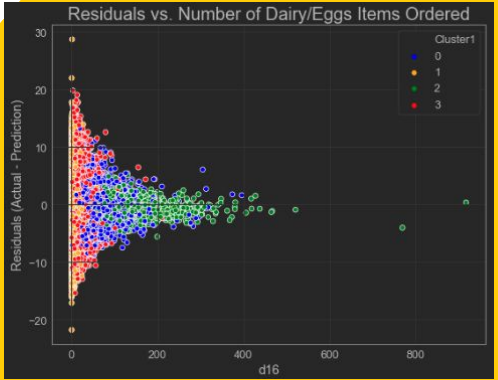
Error Evaluation – residuals

Observations:

- The magnitude of the residuals is independent from the mean number of items ordered
- The magnitude of the residuals are smaller for greater total number of items ordered
- Data indicates users who order more items more frequently (Cluster 2) are easier to predict







05

Conclusions

and future work

Summary

	Summary	Ideas
Cluster 0 (17.3%)	• Semi-frequent shoppers • Shoppers order moderate number to many items • Moderate difficulty in predicting lag time	• Provide rewards for every nth order to increase loyalty
Cluster 1 (37.1%)	• Occasional shoppers • Shoppers order very few items • Most difficult to predict lag time	• Place second in priority • Offer discounts or provide rewards for first n orders • Target with weekly reminders and local specials
Cluster 2 (4.7)%	• Frequent shoppers • Shoppers order a moderate number to many items • Least difficult to predict lag time	• Continue monitoring cluster for sudden changes in behavior
Cluster 3 (40.8%)	• Occasional shoppers • Shoppers order a moderate number of items • Difficult to predict lag time	• Prioritize targeting this cluster • Target with weekly reminders and local specials • Repeat analysis on this cluster using products or aisles as features rather than departments for increased granularity on interests

Future Work

Grocery List

- Address sparsity in features more effectively
- Repeat clustering and regression analysis on subset of data (Cluster 3)
- Repeat clustering and regression analysis using products or aisles as features instead of departments

Questions?

...

Thanks!

Do you have any questions?
youremail@freepik.com
+91 620 421 838
yourcompany.com