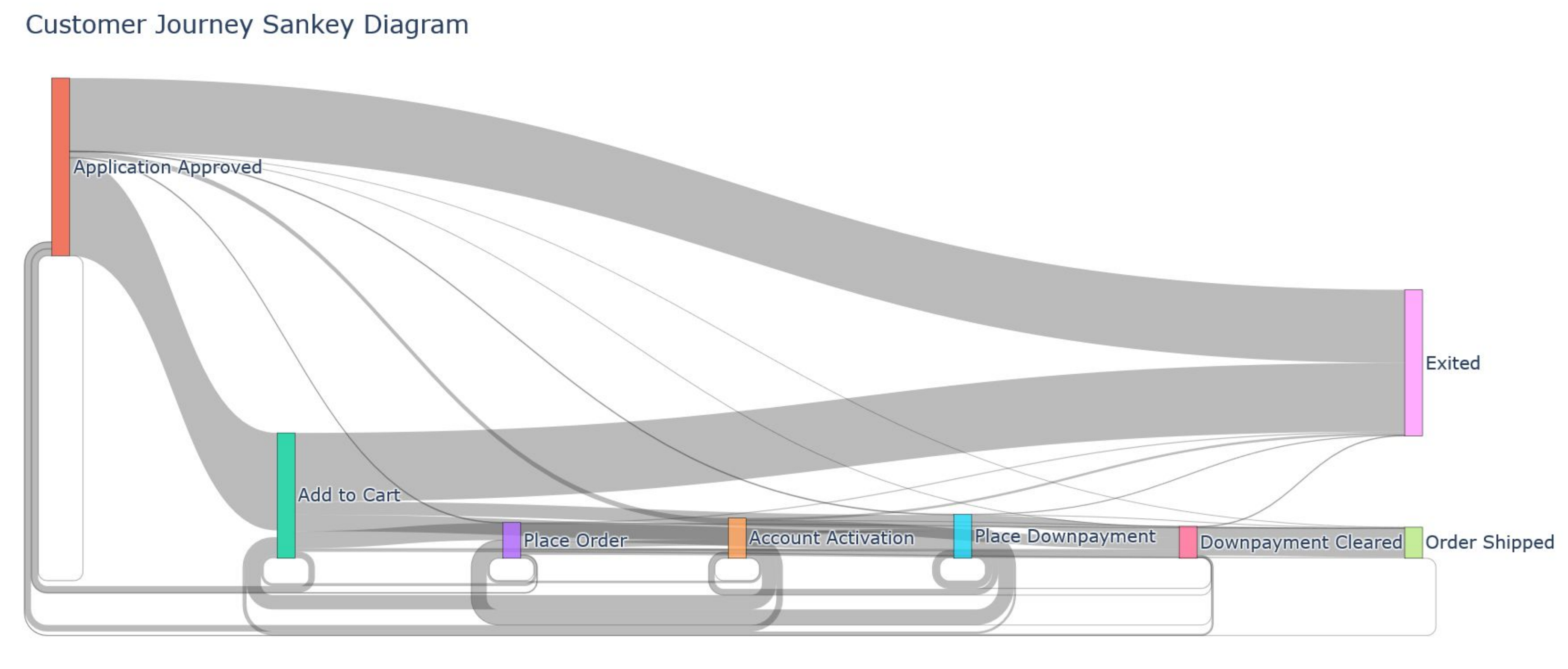
The Data Framers

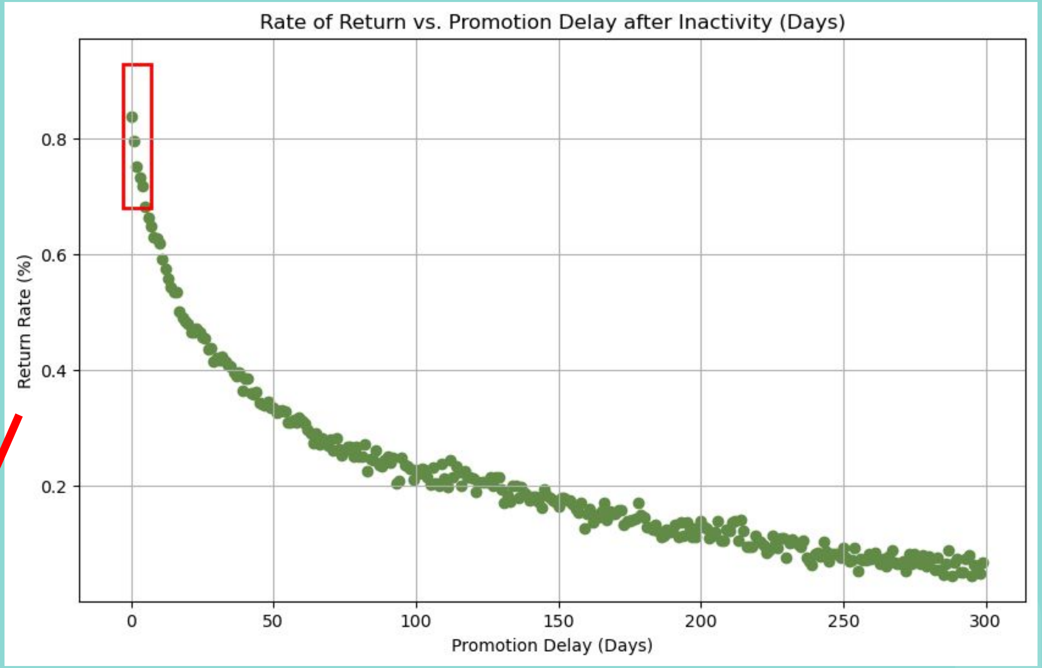
Samira Ahmed, Giselle Kurniawan,
Daniel Neufeldt, Roger Wilson



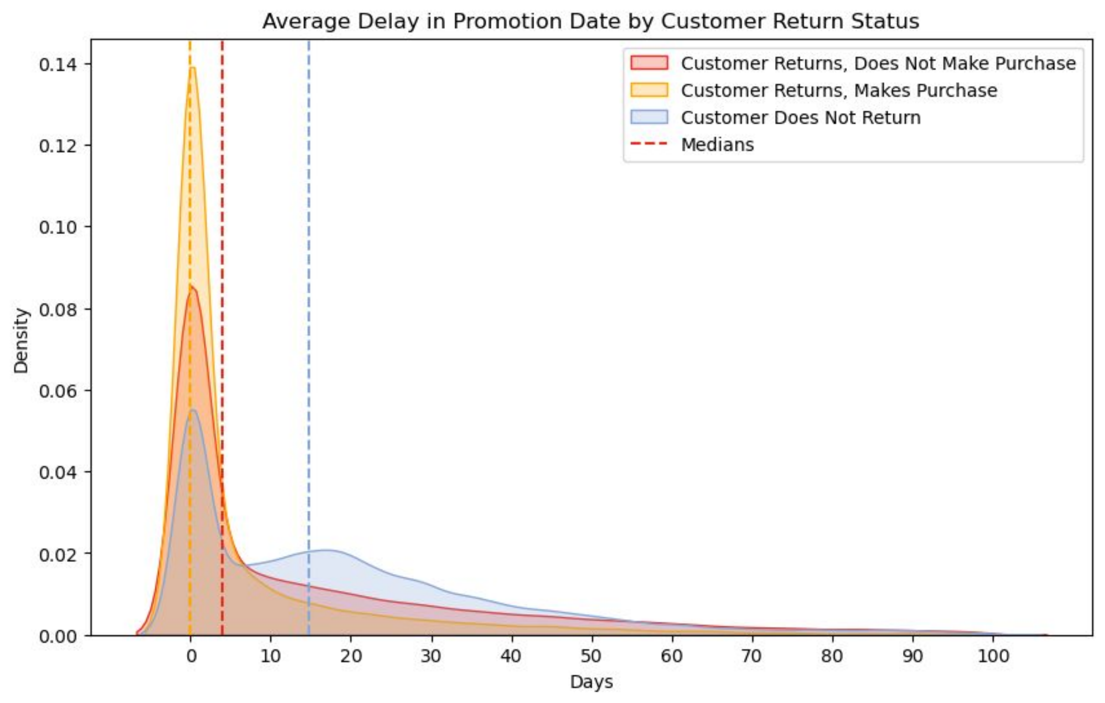
- Of 10,000 customers, 41% would never add an item to cart. Around 18% would have an order shipped.
- We added 'add_to_cart' as an important milestone

Promotions

- Random sample of 240,000 customers
- For each customer:
    - Record the delay of a promotion based off of last user activity
    - A promotion is ‘successful’ if the user returns to the site
- Calculate the return rate for each day that a promotion is delayed



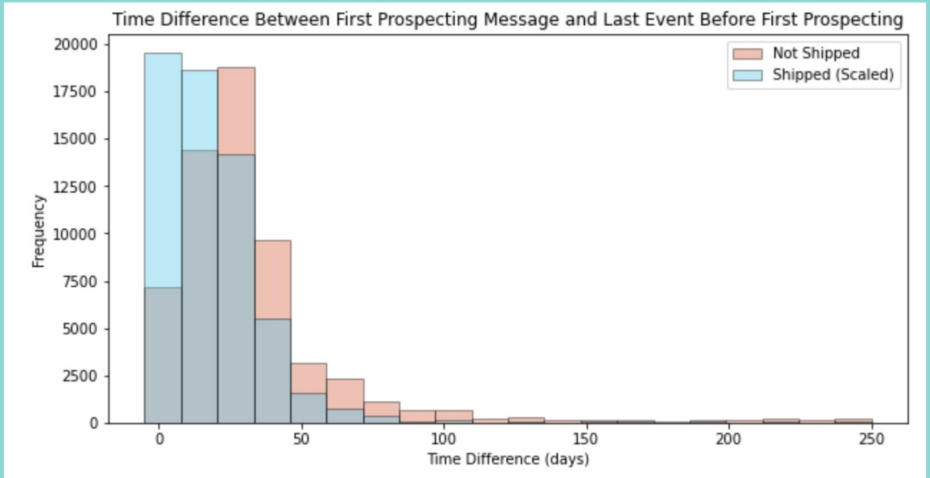
- Promotions sent within the first six hours after inactivity had the greatest success in customer return rate
- From 0 to 25 days, the likelihood of a customer returning drops by half (80% -> 40%)

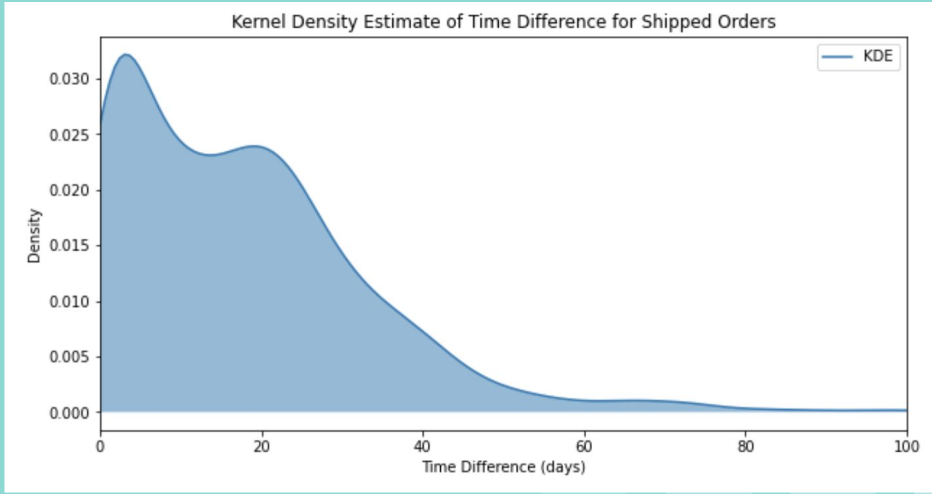


Promotions that lead to:	Return, Made Purchase	Return, No Purchase	No Return
Mean (days)	8.6	19.7	23.7
Median (days)	0.08 (1.9 hours)	5.5	15.8
Number of Customers	19.3%	40.0%	40.7%

Prospecting

- Used sample of ~165,000 random customers
- Both groups have received prospecting mail, but split by orders completed or not
- Median Values:
    - Shipped: 15 days
    - Not Shipped: 26 days
- KDE Proposed Range: 1 - 6 days

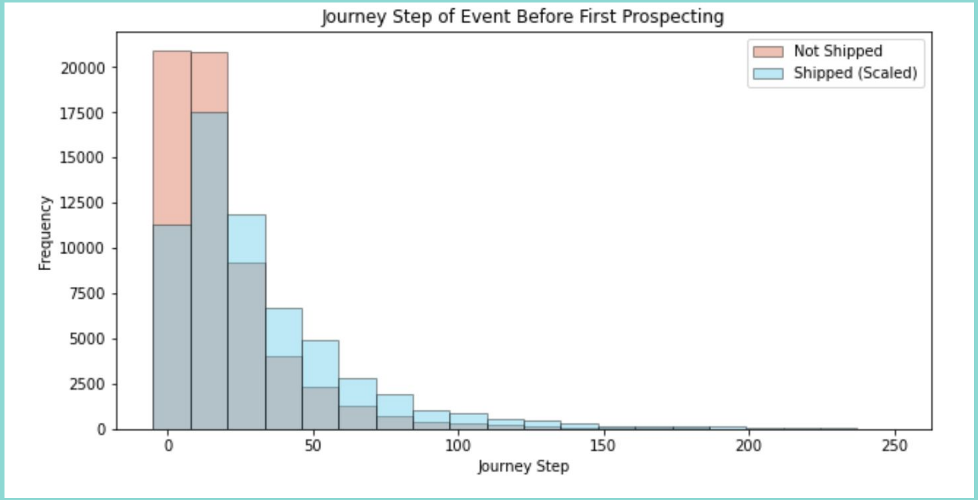


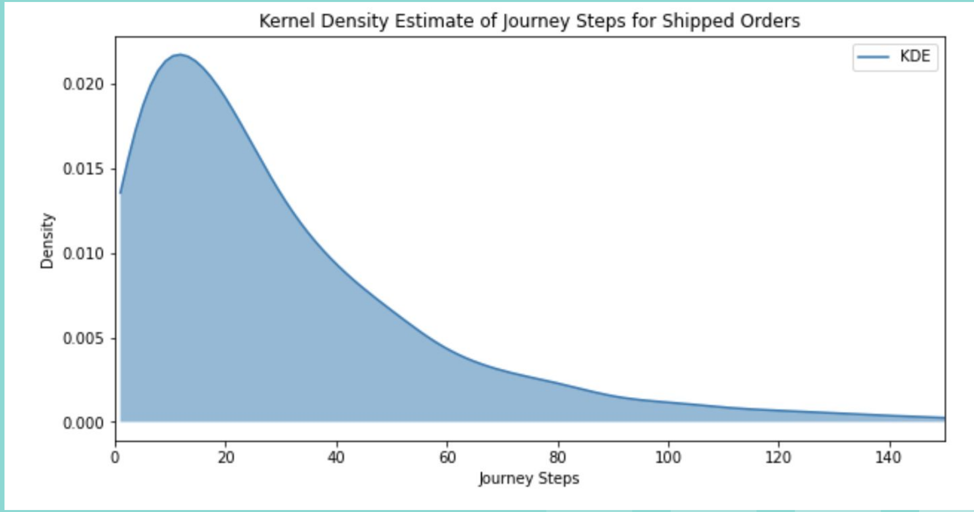


- Using same data, looked into the journey step of the customer before receiving first prospecting mail

- Median Values:
    - Shipped: 22 Steps
    - Not Shipped: 12 Steps

- KDE Proposed Range: Steps 6 - 19





Can we estimate the likelihood of a customer reaching order completion?
What helps a customer reach order completion?

XGBoost Algorithm

- **Goal:** Develop an XGBoost model for predicting customer order placement.

- **Result:** Leveraging n-gram sequences and numerical features like session duration, a model achieving an approximate accuracy score of 94% was developed for predicting customer order placement.

Data Preprocessing and Feature Engineering

Data Preprocessing

- Data is grouped according to customer_id and is sessionized, allowing for multiple sessions per customer_id with a session timeout set at 2 days.

Feature Engineering

- **1-gram extraction**: We extracted 1-grams from the customer interaction data to capture the individual actions taken by each customer.
    - Combined the event names for each customer's session into a list of strings.
    - Used CountVectorizer to transform the column into a sparse numeric matrix, allowing us to represent each unique event as a feature and quantify its occurrence within each session
- **Numerical Feature**: Calculated time duration (in minutes) for each session.

	customer_id	session_id	events	session_duration
0	-2147452610	3945806	[application_web_approved, promotion_created, ...	7453.400000
1	-2147425125	235845	[campaign_click, application_web_view, applica...	1729.300000
2	-2147425125	235846	[browse_products, account_activitation, campai...	6550.700000
3	-2147417277	1727803	[application_web_approved, promotion_created, ...	407.916667
4	-2147395611	2295445	[account_activitation, campaign_click, applica...	5760.000000
5	-2147395574	4275402	[application_web_approved, promotion_created]	0.398667
6	-2147395574	4275403	[browse_products, browse_products, add_to_cart...	18274.801333
7	-2147379618	4356472	[campaign_click, application_web_view, applica...	5400.000000
8	-2147357371	6922986	[campaign_click, application_web_approved, bro...	1664.266667
9	-2147357371	6922987	[browse_products, browse_products, add_to_cart...	22426.863050

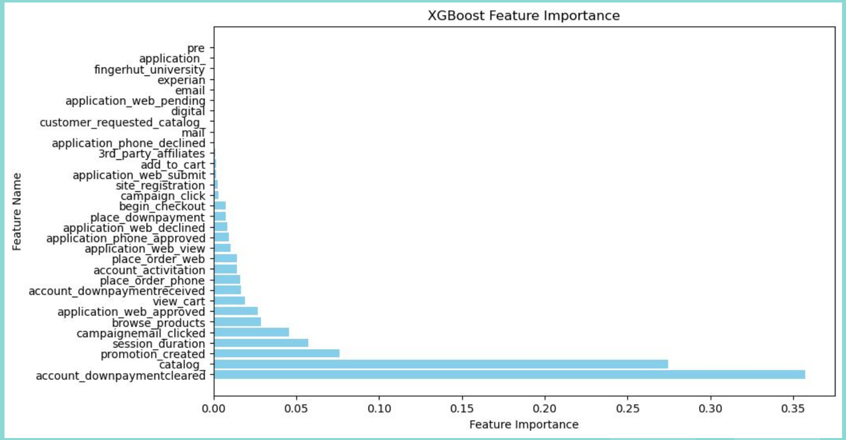
XGBoost Model Evaluation

Training and Testing

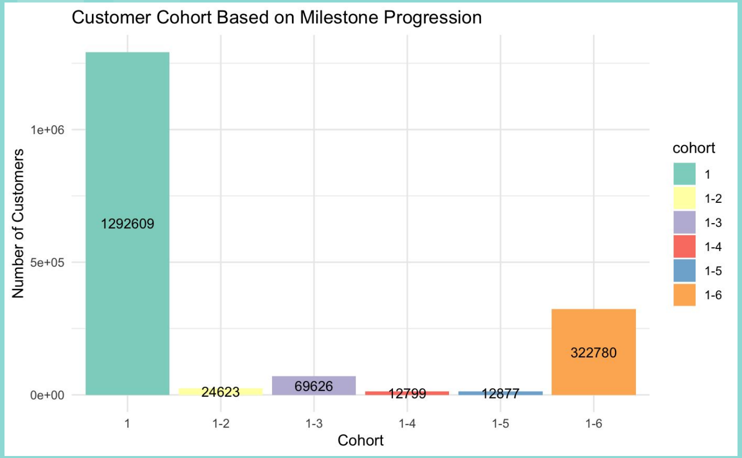
- Using scikit-learn's train_test_split, we partitioned the combined feature matrix into training and testing sets.
  - Training: 70%
  - Testing: 30%
- Employed the XGBClassifier to fit our model, and generated predictions.

Interpretability

- Upon evaluating our model using the f1 score metric, we attained an accuracy score of approximately 93.9%
- Uses XGBoost's Feature Importance Ranking to determine which events significantly impact the model's outcome
  - Event "account_downpaymentcleared" is the most influential factor behind model's predictions, suggesting that customers who have their account down payment cleared are more inclined to place an order.

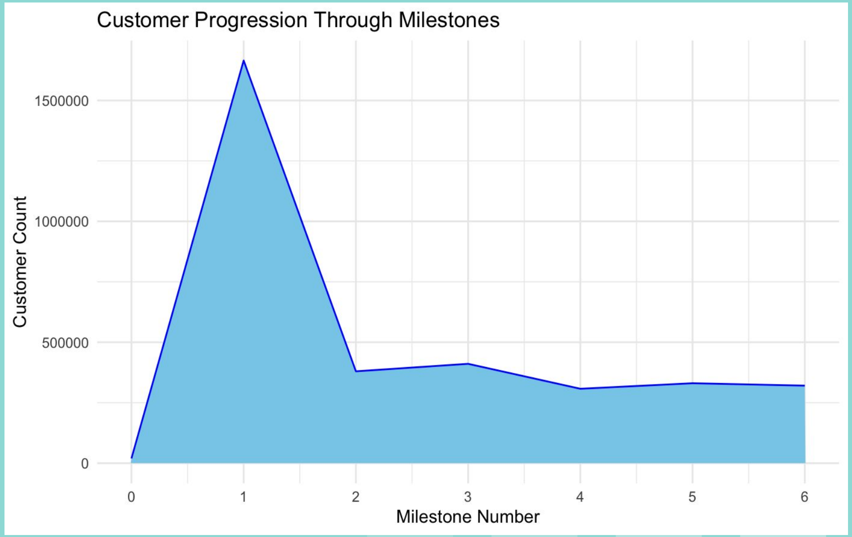


Customer Cohort and Milestone Progressions



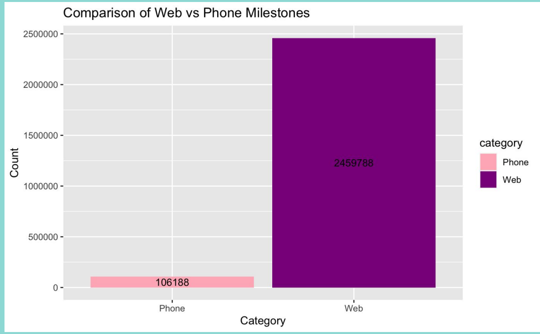
- Group each customer by their minimum and maximum milestones
- Find the count of each cohort
- Move on to further Analysis of why customers might drop off from milestones 1 to 2
- About 18.60% conversion of customers making it to milestone 6 from combined customer and account identification (1,735,767)

- The peak at milestone 1 (application approved) suggests that the majority of customers have reached this point in their journey. This also indicates that customers that are successfully starting their journey are not proceeding to later milestones.
- There's a significant decrease in the number of customers who reach milestone 2 (place order) and beyond.
- After the initial drop-off, the number of customers at each subsequent milestone appears relatively stable. This could indicate that customers who proceed past milestone 2 are likely to continue on their journey and get an order shipped, i.e., milestone 6.



Preprocessing, Feature Engineering, Benefits

- Concentrated on customer accounts up to milestone 2 because of stabilization after this point
- Combine phone and web user events since our primary focus was on customer’s 'place_order', regardless of platform they were using phone users were only about 4% of the overall data
- Remove Identifier Columns and other columns we found that did not contribute to our previous analysis about customer progression (customer_id, milestone_number, journey_steps_until_end, ed_id, event_timestamp, stage) only kept event_defintion and account_id
- Pivoted wider where each row detailed the frequency of these events per customer account (20 total columns)
- Removed Features (columns) with variances less than 0.1 (left with 11 columns) these features typically meant that there is a substantial spread in the data points which in turn leads to meaningful differences in customer behavior that our clustering algorithm can leverage
- Remove Outliers
- Overall these all helped reduce the dimensionality of our data and helped with computational efficiency




```
> variances
                    add_to_cart                 application_approved              begin_checkout
                   2.575006e+01                         2.380425e-03                9.310889e+00
                browse_products                       campaign_click                 place_order
                   6.020500e+02                         2.614305e-01                2.188581e-01
              promotion_created                            view_cart      application_web_submit
                   1.395145e+01                         5.698789e+01                3.033258e+00
           application_web_view pre-application_(3rd_party_affiliates)     campaignemail_clicked
                   4.094749e+01                         8.347408e-02                8.018667e+00
                 catalog_(mail)                    site_registration        application_declined
                   3.799848e+00                         1.281351e-02                3.228419e-02
      catalog_(email)_(experian)                  application_pending       fingerhut_university
                   3.497977e-04                         3.076789e-05                4.229115e-04
customer_requested_catalog_digital
                   1.206620e-06
```

	account_id	add_to_cart	begin_checkout	browse_products	campaign_click	place_order	promotion_created	view_cart	application_web_submit	application_web_view	campaignemail_clicked	catalog_(mail)
1	-2147477843	3	2	9	1	1	4	5	0	0	0	0
2	-2147476504	15	7	18	1	1	3	8	1	6	0	0
3	-2147476077	0	0	8	1	0	0	0	1	9	0	0
4	-2147475397	0	0	0	1	0	0	0	0	0	0	0
5	-2147473858	2	1	6	0	0	2	2	1	3	0	0
6	-2147468021	2	3	4	0	1	1	3	0	1	0	0
7	-2147467127	4	6	35	1	1	3	11	2	6	3	0
8	-2147465451	2	0	29	1	0	2	1	2	10	3	0
9	-2147464163	46	21	107	1	0	6	52	0	0	0	0
10	-2147463978	1	1	12	0	1	1	1	0	0	0	0
11	-2147463068	0	0	4	1	0	9	0	1	11	3	2
12	-2147462231	11	1	50	1	0	6	12	0	0	0	2
13	-2147461160	2	1	20	1	0	2	4	1	10	0	1
14	-2147459581	1	4	7	1	0	4	4	0	0	0	0
15	-2147457073	1	0	1	0	0	2	0	0	0	0	0
16	-2147450374	1	2	2	1	1	1	2	0	1	0	0
17	-2147449445	1	0	2	0	0	1	1	1	1	0	0
18	-2147448595	1	1	10	1	1	0	1	0	0	0	0

Results and Evaluation of Clusters

- ‘High Value Customer’ - Cluster 1 : highest mean over all in place_order, begin_check_out, browse_product, add_to_cart
- ‘Engaged Customer’ - Cluster 2 : highest mean for ‘promotion_created’ and second highest mean for ‘browse_products’, ‘view_cart’
- ‘Window Shopper’ - Cluster 3 and Cluster 4: high engagement overall (not as high as cluster 1 and cluster 2) but lowest place_order means
- When we integrated back into the dataset to evaluate order_shipped we get 80% of the ‘High Value Customers’ we clustered have an order_shipped
- For Promotion Created we see that 99.99% of ‘Engaged Customers’ have Promotion created for them and they 1.59% have an order_shipped

	cluster	add_to_cart_mean	add_to_cart_median	add_to_cart_sd	begin_checkout_mean	begin_checkout_median	begin_checkout_sd	browse_products_mean	browse_products_median	browse_products_sd
1	1	3.482993	2	3.283318	2.6904217	2	2.245784	15.130836	10	15.642691
2	2	1.723960	1	2.582028	1.0414224	0	1.762740	10.749130	5	14.105291
3	3	1.018395	0	1.780022	0.6540606	0	1.275799	4.930002	2	8.584649
4	4	1.180133	0	2.006212	0.7693334	0	1.438952	5.623521	2	9.491248

place_order_mean	place_order_median	place_order_sd	promotion_created_mean	promotion_created_median	promotion_created_sd	view_cart_mean	view_cart_median	view_cart_sd
0.968283523	1	0.17524451	2.225144	2	2.368107	5.836125	4	5.538159
0.018846137	0	0.13598165	8.201403	8	2.358748	2.662821	1	4.117029
0.001019651	0	0.03191573	2.288650	2	2.316833	1.448178	0	2.700051
0.000000000	0	0.00000000	2.574032	2	2.491831	1.719206	0	3.069899

campaign_click_mean	campaign_click_median	campaign_click_sd	place_order_mean	place_order_median	place_order_sd	promotion_created_mean	promotion_created_median	promotion_created_sd
0.5670183	1	0.5062016	0.968283523	1	0.17524451	2.225144	2	2.368107
0.6568517	1	0.4838864	0.018846137	0	0.13598165	8.201403	8	2.358748
1.0145810	1	0.1198684	0.001019651	0	0.03191573	2.288650	2	2.316833
0.0000000	0	0.0000000	0.000000000	0	0.00000000	2.574032	2	2.491831

cluster_label	Total_Customers	Promotions_Created	Proportion
<chr>	<int>	<int>	<dbl>
Engaged Customers	383707	383684	0.9999401
High Value Customers	317460	216870	0.6831412
Window Shoppers	805674	542583	0.6734523
NA	151021	139286	0.9222956

cluster_label	Total_Customers	Shipped_Orders	Proportion
<chr>	<int>	<int>	<dbl>
Engaged Customers	383707	6118	0.015944458
High Value Customers	317460	254882	0.802879103
Window Shoppers	805674	6222	0.007722727
NA	151021	51222	0.339171374

Thank you!