H&M Data Scientist CRM Case Study Report

Bryan Clark

8/18/2018

Business Understanding

Goals

The goals of this Marketing CRM case study for H&M are listed below along with a prospposed analytical solution.

1. Postcard Send-Outs

We have postcards in several markets and would like to send them to the right customer.

- [**Response Prediction**] A model could be built to predict the propensity of a customer to make a purchase in response to receiving a postcard. This model can be run prior to each batch of postcards sent to customers. H&M can then only send postcards to customers with higher probabilities of responding to the postcard to maximize the return on investment of sending the postcards.

2. Strategic Segmentation

We want to learn more about our customer base to support customer insight in the organization. We also want to use it for how we will work with customers, e.g. which customer groups we will target different activities.

- [**Customer Segmentation**] A model could be built to segment customers based on various attributes such as the cateogry of prodcuts they buy, affinity for particular brands/collections, and/or their purchase behavior. This information could be used to decide which product offers, email content, messaging, and/or promotions to send to each segment of customers. Based on the number of clusters and DNA of each cluster, marketing strategies can be formed separately for each customer.

3. New Customers

A project that works with Online customers needs our analytical help. Many customers shop once, but never come back. How do we get new customers to return?

- [**Lifecycle Segmentation**] Customers can be segmented based on their last interaction with H&M. The exact timeframes used for the development of customer journey lifecycles are dependent on purchase cadences of current customers. Marketing strategies can then be developed for each customer journey segment. For example, new customers can be provided "Thank You" messaging along with an introduction to new categories of products relevant to their first purchase and/or additional information about how to have a successful relationship with H&M.
- [**Value Segmentation**] Customers can be segmented based on their value to H&M. The top 10% of customers are considered VIP or high-value customers, the next 60% of customers are considered medium-value customers, and the bottom 30% of customers labeled as low-value customers. Marketing offers can be adjusted to cater to each value segment. For example, potential high-value customers (based on their first/last purchase) can receive deeper offers enticing them to make another purchase while lower-value customers can receive a smaller offer.

Data Sensemaking

Each of these potential solutions assumes the data is available to move forward. Normally, we would move forward with identifying the data that could be useful for exploring each of the potential analytics solutions. However, in this case we have been provided a dataset to move forward to develop our marketing strategies.

```r
# load in provided customer dataset
customers <- read_csv("Case_data_2018.csv")
print(glimpse(customers))
```

```
## Observations: 10,000
## Variables: 18
## $ dayssincefirst   <int> 2036, 2073, 1518, 1935, 1888, 2073, 2093, 161...
## $ dayssincelast    <int> 799, 397, 379, 587, 947, 2, 279, 1618, 183, 2...
## $ PurchaseMen      <chr> "No", "No", "No", "No", "No", "Yes", "No", "N...
## $ Purchasekids     <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ Purchasesports   <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Ye...
## $ Purchaseswim     <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Ye...
## $ Purchaseeco      <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ Purchasejackets  <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Ye...
## $ zipcode          <int> 130, 730, 68300, 74700, 1700, 88900, 28450, 3...
## $ sharehighfashion <chr> ".", ".", ".", ".", ".", "0.04", ".", ".", "0...
## $ purchaseonline   <chr> ".", "1", "1", ".", ".", ".", ".", ".", "1", ...
## $ purchlast1year   <chr> "0", "92.94", "263.91", "0", "0", "284.76", "...
## $ purchlast2years  <chr> "129.93", "506.69", "263.91", "192.93", "89.9...
## $ age              <int> 46, 46, 48, 63, 48, 50, 55, 41, 38, 48, 60, 4...
## $ clubmember       <int> 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ emailsubscriber  <chr> ".", ".", "1", ".", ".", ".", ".", ".", "1", ...
## $ salesdriven      <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## $ purchasepostcard <chr> "No", "No", "No", "No", "No", "No", "No", "No...
## # A tibble: 10,000 x 18
##    dayssincefirst dayssincelast PurchaseMen Purchasekids Purchasesports
##             <int>         <int> <chr>       <chr>        <chr>
##  1           2036           799 No          No           Yes
##  2           2073           397 No          No           Yes
##  3           1518           379 No          No           Yes
##  4           1935           587 No          No           Yes
##  5           1888           947 No          No           Yes
##  6           2073             2 Yes         No           Yes
##  7           2093           279 No          No           Yes
##  8           1618          1618 No          No           Yes
##  9           2059           183 No          No           Yes
## 10             28            28 Yes         Yes          No
## # ... with 9,990 more rows, and 13 more variables: Purchaseswim <chr>,
## #   Purchaseeco <chr>, Purchasejackets <chr>, zipcode <int>,
## #   sharehighfashion <chr>, purchaseonline <chr>, purchlast1year <chr>,
## #   purchlast2years <chr>, age <int>, clubmember <int>,
## #   emailsubscriber <chr>, salesdriven <chr>, purchasepostcard <chr>
```

General Notes

We have a dataset that consists of 10,000 records and 18 variables. Upon first review, it seems that we have variables that provide us information around the customers first and most recent purchase timeframes, categories of purchases, purchase amounts for the last 1 and 2 years, flags for if the customer has made purchases online, is a member of the H&M club, is subscribed to emails, is sales driven (responsive to sales/promotions?), and has responded to a postcard by making a purchase.

Feasibility of Proposed Solutions

Perhaps more importantly, it seems like we do not have data available on the number of visits/purchases for each customer, which would allow us to determine time-between-purchases to guide our customer lifecycle journey segmentation. In the absence of this data, we can use outside research to derive our customer journey segments. According to statista (2018), 45% of people purchase clothing at least once every 90 days.

We should be able to move forward with each of the proposed solutions based on the dataset available.

Data Validity Concerns

Additionally, we will have some data cleansing to do. In addiition to "." present in the dataset, the second record seems to indicate a customer with a purchase amount in the last 1 year that also has had 397 days since their last purchase. We'll need to identify potentially faulty records to remove for our modeling. It is also possible these could indicate returns or exchanges (returns if purchase value is negative and exchanges if it is 0), but we want to identify and consider removing them to be safe.

NOTE: We will assume that the validity of the data will not prevent us from removing forward with mining insights for this case study. Normally, we would want to address the source(s) of this data to determine why this issue exists. Is the SQL query faulty? Did someone merge multiple sources of data from different timeframes? Are there any other columns that have been corrupted? Are these exchanges?

Data Exploration

In this section, we are going to address the data quality concerns noticed in the data sensemaking phase, explore our data further, and derive any new variables of interest for our business goals.

Data Cleansing Before we begin exploring our data, we will need to address some of our data validation concerns.

For each column that contains a "." placeholder, we are assuming the actual value should be 0. Additionally, for uniformity, we'll convert the Yes/No columns to binary numbers (1 for yes, 0 for no).

```r
# columns to convert to numeric
num_cols <- c(10:13, 16)
customers[, num_cols] <- lapply(customers[, num_cols], as.numeric)
customers[is.na(customers)] <- 0

# Yes/No columns to 1 for Yes and 0 for No
yes_no_cols <- c(3:8, 17:18)
customers[yes_no_cols] <- ifelse(customers[ , yes_no_cols] == "Yes", 1, 0)
# columns to convert to binary factors
print(glimpse(customers))
```

```
## Observations: 10,000
## Variables: 18
## $ dayssincefirst   <int> 2036, 2073, 1518, 1935, 1888, 2073, 2093, 161...
## $ dayssincelast    <int> 799, 397, 379, 587, 947, 2, 279, 1618, 183, 2...
## $ PurchaseMen      <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, ...
## $ Purchasekids     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, ...
## $ Purchasesports   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, ...
## $ Purchaseswim     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ Purchaseeco      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...
## $ Purchasejackets  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, ...
## $ zipcode          <int> 130, 730, 68300, 74700, 1700, 88900, 28450, 3...
## $ sharehighfashion <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.04, 0.00, 0.0...
## $ purchaseonline   <dbl> 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, ...
## $ purchlast1year   <dbl> 0.00, 92.94, 263.91, 0.00, 0.00, 284.76, 129....
## $ purchlast2years  <dbl> 129.93, 506.69, 263.91, 192.93, 89.96, 1488.9...
## $ age              <dbl> 46, 46, 48, 63, 48, 50, 55, 41, 38, 48, 60, 4...
## $ clubmember       <int> 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ emailsubscriber  <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, ...
## $ salesdriven      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, ...
## $ purchasepostcard <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, ...
## # A tibble: 10,000 x 18
##    dayssincefirst dayssincelast PurchaseMen Purchasekids Purchasesports
##             <int>         <int>       <dbl>        <dbl>          <dbl>
##  1           2036           799           0            0              1
##  2           2073           397           0            0              1
##  3           1518           379           0            0              1
##  4           1935           587           0            0              1
##  5           1888           947           0            0              1
##  6           2073             2           1            0              1
##  7           2093           279           0            0              1
##  8           1618          1618           0            0              1
##  9           2059           183           0            0              1
## 10             28            28           1            1              0
## # ... with 9,990 more rows, and 13 more variables: Purchaseswim <dbl>,
## #   Purchaseeco <dbl>, Purchasejackets <dbl>, zipcode <int>,
## #   sharehighfashion <dbl>, purchaseonline <dbl>, purchlast1year <dbl>,
## #   purchlast2years <dbl>, age <dbl>, clubmember <int>,
## #   emailsubscriber <dbl>, salesdriven <dbl>, purchasepostcard <dbl>
```

Next we will explore how many rows may possibly be corrupt. If days since last purchase are 365 or less and the customer has a 0 value for purchase amount in the last year, OR if the days since last purchase are 730 or less and the customer has a 0 value for the two-year purchase total, we have a corrupt record.

```
# create logical checks for each scenario
check_1 <- (customers$dayssincelast <= 365 & customers$purchlast1year == 0)
check_2 <- (customers$dayssincelast <= 730 & customers$purchlast2years == 0)

# add columns to indicate if column is flagged as potential concern
customers$invalid <- (check_1 | check_2)
```

** Data Quality Report **

As a sanity check, we will review the breakdown of our numeric and binary columns. This will help us identify any other potential issues before exploring the distributions of our variables visually.

Below, we see several interesting things of note about what our dataset contains:

- A very high percentage (99%+) of customers have purchased jackets and swim items, and a very low percentage (< 1%) of customers have purchased eco items. Not only will those variables provide little information for our models, but any insights gleaned from this sample of customers may not be useful for customers that have not made purchases in either of jackets/swim or have made purchases in eco.
- The minimum age in the dataset is 0 and the maximum age is 551, which means we have another quality issue. If we want to use this variable for analysis, we will need to address those records.
- The mean is much larger than the median for `dayssincelast`, `purchlast1year`, and `purchlast2years` indicating these are skewed to the right (long tail to the right). There appear to be some outliers with our purchase columns with max values of 5x the 3rd quartile for purchases in the last year and 4x the 3rd quartile for purchases in the last two years. We'll have to address this before building our predictive and segmentation models.
- In terms of ouor first goal to identify customers to target with postcards, 30% of our dataset has made a purhcase from a postcard.

```
# create function to run summary on numeric features
df_num_summary <- function(df, cols = NULL) {

  if (is.null(cols)) {
    num.cols <- colnames(select_if(df, is.numeric))
  } else {
    num.cols <- cols
  }

  df <- subset(df, select = num.cols)

    df.num.summmary <- data.frame(
      Count = round(sapply(df, length), 2),
      Miss = round((sapply(df, function(x) sum(length(which(is.na(x)))) / length(x)) * 100), 1),
      Card. = round(sapply(df, function(x) length(unique(x))), 2),
      Min. = round(sapply(df, min, na.rm = TRUE), 2),
      `25 perc.` = round(sapply(df, function(x) quantile(x, 0.25, na.rm = TRUE)), 2),
      Median = round(sapply(df, median, na.rm = TRUE), 2),
      Mean = round(sapply(df, mean, na.rm = TRUE), 2),
      `75 perc.` = round(sapply(df, function(x) quantile(x, 0.75, na.rm = TRUE)), 2),
      Max = round(sapply(df, max, na.rm = TRUE), 2),
      `Std Dev.` = round(sapply(df, sd, na.rm = TRUE), 2)
    ) %>%
      rename(`1st Qrt.` = X25.perc.,
             `3rd Qrt.` = X75.perc.,
             `Miss Pct.` = Miss)
    return(df.num.summmary)
}

customers_num_summary <- df_num_summary(df = customers)

# display in table
kable(customers_num_summary#, type = "html"
      ) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "left",
                latex_options = "scale_down")
```

	Count	Miss Pct.	Card.	Min.	1st Qrt.	Median	Mean	3rd Qrt.	Max	Std.Dev.
dayssincefirst	10000	0	1520	8	1666.00	1981.00	1774.76	2056.00	2095.00	425.25
dayssincelast	10000	0	1751	2	73.00	269.50	478.50	772.00	2094.00	514.61
PurchaseMen	10000	0	2	0	0.00	0.00	0.19	0.00	1.00	0.39
Purchasekids	10000	0	2	0	0.00	0.00	0.38	1.00	1.00	0.48
Purchasesports	10000	0	2	0	1.00	1.00	0.79	1.00	1.00	0.40
Purchaseswim	10000	0	2	0	1.00	1.00	1.00	1.00	1.00	0.05
Purchaseeco	10000	0	2	0	0.00	0.00	0.01	0.00	1.00	0.07
Purchasejackets	10000	0	2	0	1.00	1.00	1.00	1.00	1.00	0.06
zipcode	10000	0	1720	100	6100.00	33400.00	38552.19	66805.00	99980.00	31991.03
sharehighfashion	10000	0	101	0	0.00	0.00	0.24	0.47	1.00	0.32
purchaseonline	10000	0	2	0	0.00	0.00	0.37	1.00	1.00	0.48
purchlast1year	10000	0	4788	0	0.00	90.45	286.82	359.63	15467.57	551.72
purchlast2years	10000	0	7041	0	70.38	333.83	719.31	902.79	36237.25	1212.71
age	10000	0	65	0	31.00	37.00	38.16	43.00	511.00	10.34
clubmember	10000	0	2	0	0.00	0.00	0.27	1.00	1.00	0.44
emailsubscriber	10000	0	2	0	0.00	1.00	0.57	1.00	1.00	0.50
salesdriven	10000	0	2	0	0.00	0.00	0.21	0.00	1.00	0.40
purchasepostcard	10000	0	2	0	0.00	0.00	0.30	1.00	1.00	0.46

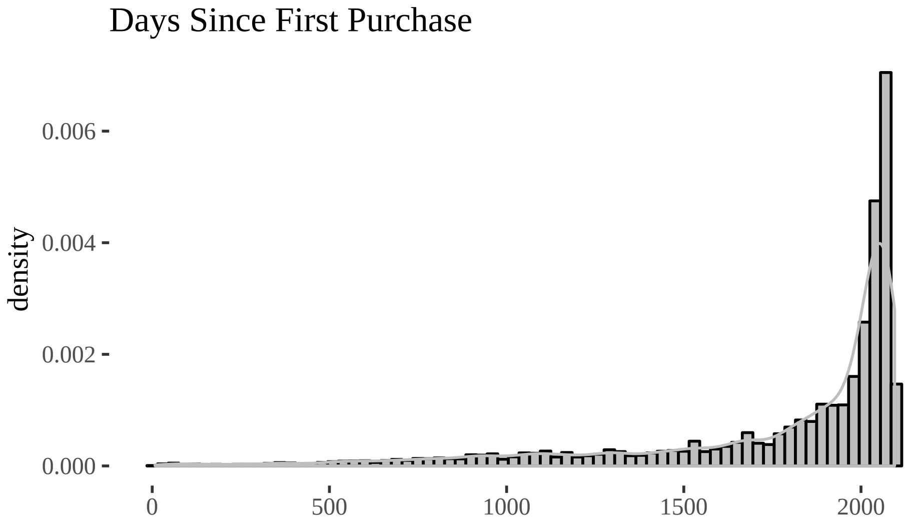
Data Exploration Plots

To get a better idea of the distribution of days since first/last purchase, purchases in last 1/2 years, and age, we'll create histograms of each.

Days Since First Purchase

We see our dataset primarily contains customers that made their first purchase over 4 years ago.

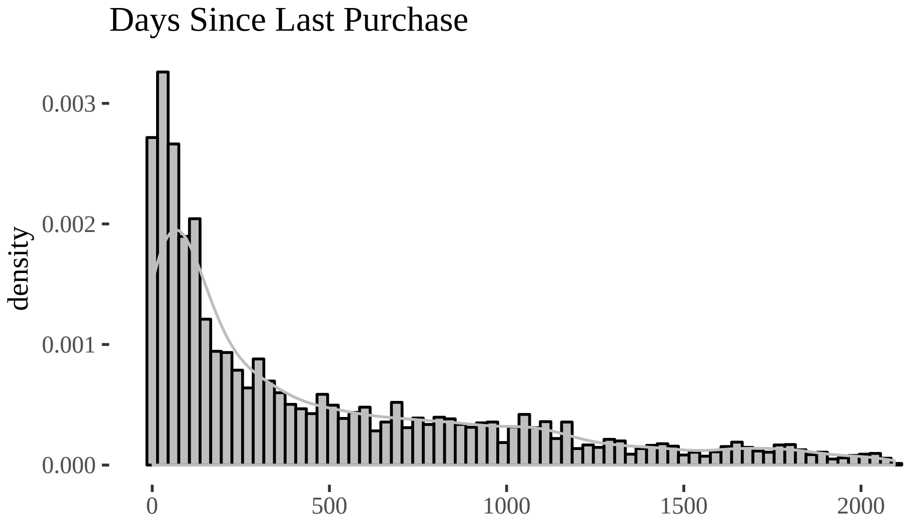
```
ggplot(customers, aes(x = dayssincefirst)) +
  geom_histogram(aes(y =..density..), color = "black", fill = "grey", binwidth = 30) +
  geom_density(color = "grey", alpha = 0.4) +
  labs(title = "Days Since First Purchase",
       x = "")
```



Days Since Last Purchase

When plotting the distribution of `dayssincelast`, we see the majority of customers have made a purchase in the last year, and there are customers that have been inactive (> 2-3 years since last purchase) that we could potentially target with reactivation campaigns.

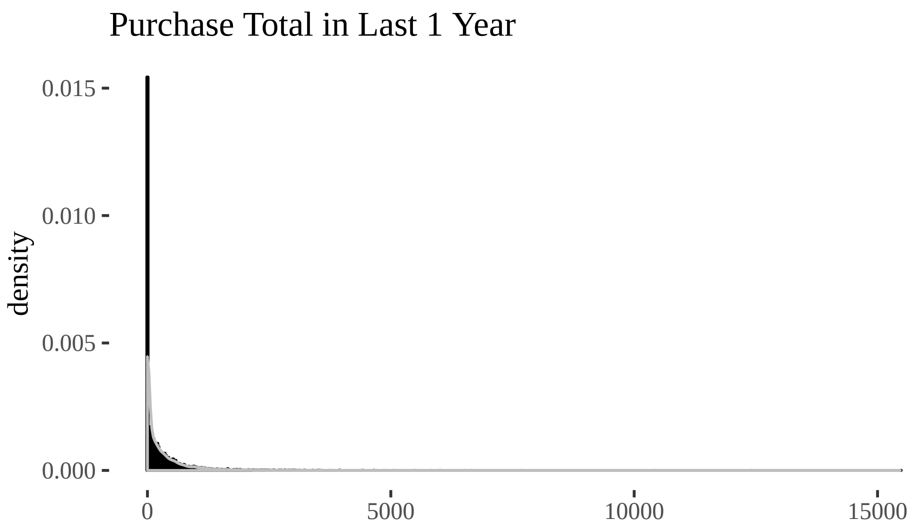
```r
ggplot(customers, aes(x = dayssincelast)) +
  geom_histogram(aes(y =..density..), color = "black", fill = "grey", binwidth = 30) +
  geom_density(color = "grey", alpha = 0.4) +
  labs(title = "Days Since Last Purchase",
       x = "")
```



Purchases in Last 1 Year

When using bins of $25, our plot helps illustrate the outliers in our sample as the tail trails off to $15k.

```r
ggplot(customers, aes(x = purchlast1year)) +
  geom_histogram(aes(y =..density..), color = "black", fill = "grey", binwidth = 25) +
  geom_density(color = "grey", alpha = 0.4) +
  labs(title = "Purchase Total in Last 1 Year",
       x = "")
```



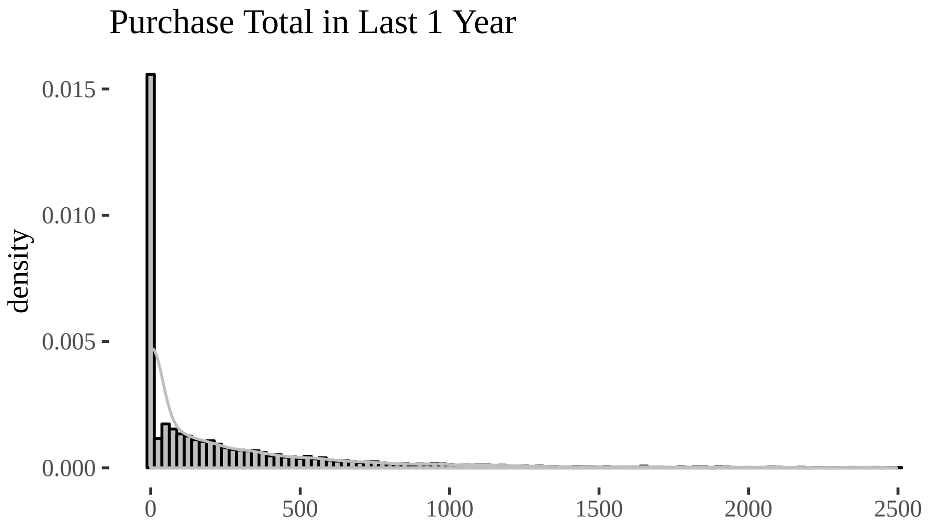
How many customers have a purchase amount greater than $2.5k? Less than 1% of our sample spent more than $2.5k in the last year.

```
print(prop.table(table(customers$purchlast1year > 2500)))
```

```
##
##  FALSE   TRUE
## 0.9911 0.0089
```

When we remove the extreme outliers from our graph, we see most customers have spent $25 or less in the last year.

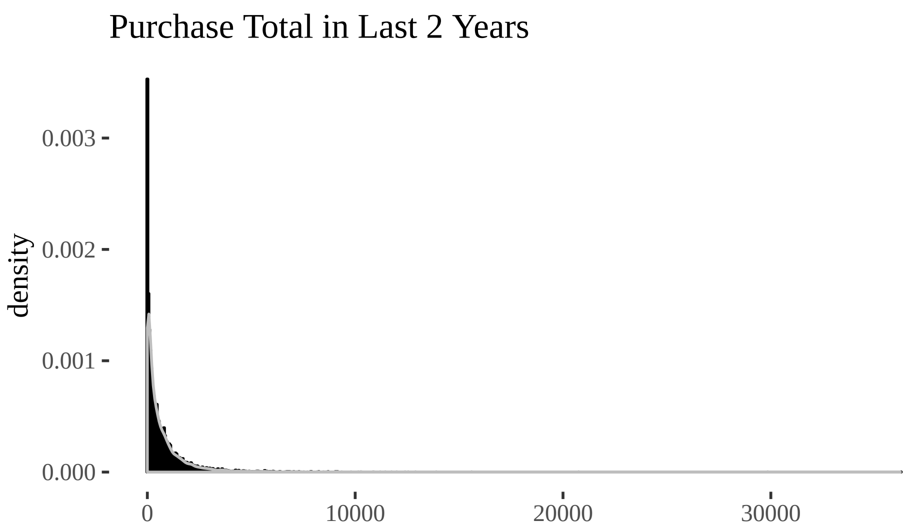
```
ggplot(customers[customers$purchlast1year <= 2500, ], aes(x = purchlast1year)) +
  geom_histogram(aes(y =..density..), color = "black", fill = "grey", binwidth = 25) +
  geom_density(color = "grey", alpha = 0.4) +
  labs(title = "Purchase Total in Last 1 Year",
       x = "")
```



Purchases in Last 2 Year

When using bins of $50, our plot helps illustrate the outliers in our sample as the tail trails off past $30k.

```r
ggplot(customers, aes(x = purchlast2years)) +
  geom_histogram(aes(y =..density..), color = "black", fill = "grey", binwidth = 50) +
  geom_density(color = "grey", alpha = 0.4) +
  labs(title = "Purchase Total in Last 2 Years",
       x = "")
```

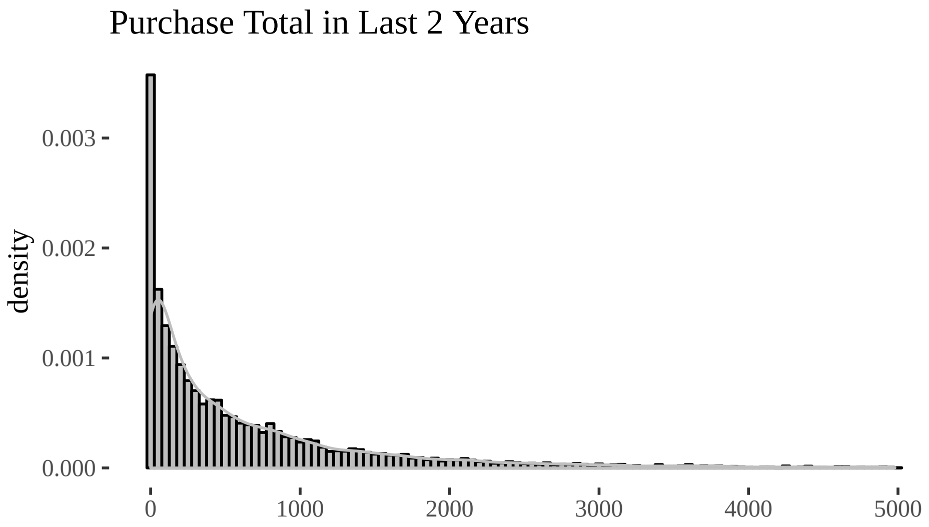


How many customers have a purchase amount greater than $5k for the last 2 years? About 1% of our sample spent more than $2.5k in the last year.

```r
print(prop.table(table(customers$purchlast2years > 5000)))
```

```
##
##  FALSE   TRUE
## 0.9876 0.0124
```

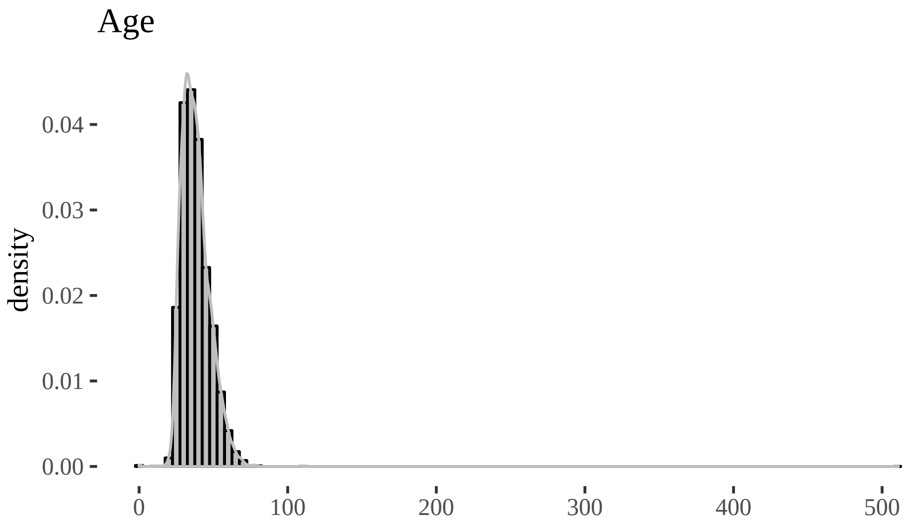
```r
ggplot(customers[customers$purchlast2years <= 5000, ], aes(x = purchlast2years)) +
  geom_histogram(aes(y =..density..), color = "black", fill = "grey", binwidth = 50) +
  geom_density(color = "grey", alpha = 0.4) +
  labs(title = "Purchase Total in Last 2 Years",
       x = "")
```



Age

When using bins of 5 year intervals, we see our issues with the min and max values of age.

```
ggplot(customers, aes(x = age)) +
  geom_histogram(aes(y =..density..), color = "black", fill = "grey", binwidth = 5) +
  geom_density(color = "grey", alpha = 0.4) +
  labs(title = "Age",
       x = "")
```

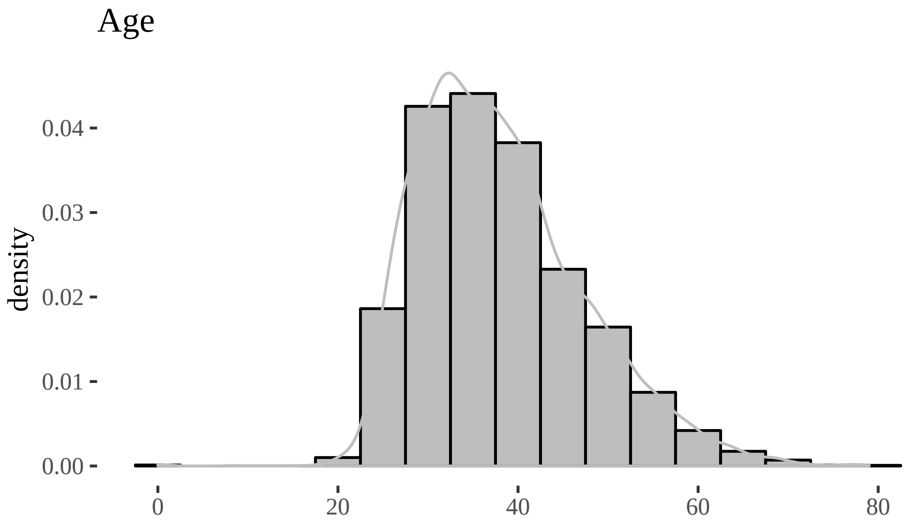


What percentage of our sample has an age greater than 85?

```
print(prop.table(table(customers$age > 85)))
```

```
##
##  FALSE   TRUE
## 0.9998 0.0002
```

```
ggplot(customers[customers$age <= 85, ], aes(x = age)) +
  geom_histogram(aes(y =..density..), color = "black", fill = "grey", binwidth = 5) +
  geom_density(color = "grey", alpha = 0.4) +
  labs(title = "Age",
       x = "")
```



Data Cleansing Conclusion

To further validate our dataset for modeling, we will remove the outliers found for purchase amounts and age. 1 year purchases greater than $2,500, 2 year purchases greater than $5,000, and ages less than 18 and greater than 85 will be removed. Unsure of legality of marketing directly to non-adults, we will play it safe and remove those under 18 from the sample.

```
# create logical checks for each purchase scenario
check_3 <- (customers$purchlast1year > 2500)
check_4 <- (customers$purchlast2years > 5000)
check_5 <- (customers$age < 18 | customers$age > 85)

# add columns to indicate if column is flagged as potential concern
customers$invalid <- (check_1 | check_2 | check_3 | check_4 | check_5)
```

How much of our sample do we have remaining? We are able to retain 98% of the original 10,000 rows.

```
print(prop.table(table(customers$invalid)))
```

```
##
##  FALSE   TRUE
## 0.9838 0.0162
```

We will remove our questionable records, the columns for categories with little separation, and zip code as we will not be using it for this analysis.

```
customers_clean <- customers %>%
  filter(invalid == FALSE) %>%
  select(-zipcode, -Purchaseswim, -Purchaseeco, -Purchasejackets, -invalid)

print(glimpse(customers_clean))
```

```
## Observations: 9,838
## Variables: 14
## $ dayssincefirst   <int> 2036, 2073, 1518, 1935, 1888, 2073, 2093, 161...
## $ dayssincelast    <int> 799, 397, 379, 587, 947, 2, 279, 1618, 183, 2...
## $ PurchaseMen      <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, ...
## $ Purchasekids     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, ...
## $ Purchasesports   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, ...
## $ sharehighfashion <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.04, 0.00, 0.0...
## $ purchaseonline   <dbl> 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, ...
## $ purchlast1year   <dbl> 0.00, 92.94, 263.91, 0.00, 0.00, 284.76, 129....
## $ purchlast2years  <dbl> 129.93, 506.69, 263.91, 192.93, 89.96, 1488.9...
## $ age              <dbl> 46, 46, 48, 63, 48, 50, 55, 41, 38, 48, 60, 4...
## $ clubmember       <int> 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ emailsubscriber  <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, ...
## $ salesdriven      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, ...
## $ purchasepostcard <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, ...
## # A tibble: 9,838 x 14
##    dayssincefirst dayssincelast PurchaseMen Purchasekids Purchasesports
##             <int>         <int>       <dbl>        <dbl>          <dbl>
##  1           2036           799           0            0              1
##  2           2073           397           0            0              1
##  3           1518           379           0            0              1
##  4           1935           587           0            0              1
##  5           1888           947           0            0              1
##  6           2073             2           1            0              1
##  7           2093           279           0            0              1
##  8           1618          1618           0            0              1
##  9           2059           183           0            0              1
## 10             28            28            1             1             0
## # ... with 9,828 more rows, and 9 more variables: sharehighfashion <dbl>,
## #   purchaseonline <dbl>, purchlast1year <dbl>, purchlast2years <dbl>,
## #   age <dbl>, clubmember <int>, emailsubscriber <dbl>, salesdriven <dbl>,
## #   purchasepostcard <dbl>
```

```
customers_clean_num_summary <- df_num_summary(df = customers_clean)

# display in table
kable(customers_clean_num_summary
      #,type = "html"
      ) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "left",
                latex_options = "scale_down")
```

	Count	Miss Pct.	Card.	Min.	1st Qrt.	Median	Mean	3rd Qrt.	Max	Std.Dev.
dayssincefirst	9838	0	1520	8	1659.00	1976.00	1771.22	2055.00	2095.00	426.81
dayssincelast	9838	0	1749	2	75.00	278.50	484.91	784.00	2094.00	515.84
PurchaseMen	9838	0	2	0	0.00	0.00	0.18	0.00	1.00	0.39
Purchasekids	9838	0	2	0	0.00	0.00	0.37	1.00	1.00	0.48
Purchasesports	9838	0	2	0	1.00	1.00	0.80	1.00	1.00	0.40
sharehighfashion	9838	0	101	0	0.00	0.00	0.24	0.47	1.00	0.33
purchaseonline	9838	0	2	0	0.00	0.00	0.36	1.00	1.00	0.48
purchlast1year	9838	0	4647	0	0.00	85.94	246.56	343.71	2495.34	374.05
purchlast2years	9838	0	6893	0	69.94	321.86	627.50	867.41	4979.31	806.69
age	9838	0	60	18	31.00	37.00	38.18	44.00	79.00	9.13
clubmember	9838	0	2	0	0.00	0.00	0.27	1.00	1.00	0.44
emailsubscriber	9838	0	2	0	0.00	1.00	0.56	1.00	1.00	0.50
salesdriven	9838	0	2	0	0.00	0.00	0.21	0.00	1.00	0.41
purchasepostcard	9838	0	2	0	0.00	0.00	0.29	1.00	1.00	0.45

Variable Creation

To assist with our strategic marketing goals, we will create four new variables:

- Customer Lifecyle Segment
- Customer Value Segment (purchases in last year)
- Customer Value Segment (purchases in last two years)
- Value Migration (change from last two years to last year)

We will define customer lifecycle segments based on days since first/last purchase:

- 0 - 90 days for first purchase (new customer)
- 91 - 180 days (active customer)
- 180 - 365 days (at-risk)
- Greater than 365 days (inactive customer).

```
customers_clean$lifecycle <- ifelse(customers_clean$dayssincefirst <= 90
                                & customers_clean$dayssincelast <= 90, "New",
                                ifelse(customers_clean$dayssincelast <= 180, "Active",
                                       ifelse(customers_clean$dayssincelast <= 365, "At-Risk",
                                              "Inactive")))

customers_clean$lifecycle <- factor(customers_clean$lifecycle)
```

We will define value segments based on percentiles of value:

- Top 10% (high-value)

- Next 50% (medium-value)
- Bottom 40% (low-value)

```r
# note: originally had the bottom 30%, but purchases of $0 went past 30th percentile
apply_value_segment <- function(x) {
  cut(x, breaks = c(quantile(x, probs = c(0, 0.4, 0.9, 1))),
      labels = c("Low", "Medium", "High"), include.lowest=TRUE)
}
# apply value segmentations
customers_clean$value_seg_1yr <- factor(apply_value_segment(customers_clean$purchlast1year))
customers_clean$value_seg_2yr <- factor(apply_value_segment(customers_clean$purchlast2years))

# determine value migration from 2-year window to 1-year window
no_change <- (customers_clean$value_seg_2yr == customers_clean$value_seg_1yr)
increase <- ((customers_clean$value_seg_2yr == "Low"
              & customers_clean$value_seg_1yr %in% c("Medium", "High")) |
             (customers_clean$value_seg_2yr == "Medium"
              & customers_clean$value_seg_1yr == "High"))
decrease <- ((customers_clean$value_seg_2yr == "High"
              & customers_clean$value_seg_1yr %in% c("Medium", "Low")) |
             (customers_clean$value_seg_2yr == "Medium"
              & customers_clean$value_seg_1yr == "Low"))

# apply value migrations
customers_clean$value_change <- factor(ifelse(decrease, "Decrease",
                                              ifelse(no_change, "Neutral",
                                                     "Increase")))
```

Lastly, we will get an understanding of our new categorial variables to confirm our classifications and understand the value migration.

We see that 44% of our dataset are inactive customers and another 42% are active customers. 77% of the customers have maintained their value segment with 11% moving upward to a higher value segment.

```r
# create function to run summary on categorical features
df_cat_summary <- function(df, cols = NULL) {

  if (is.null(cols)) {
    cat.cols <- colnames(select_if(df, is.factor))
  } else {
    cat.cols <- cols
  }

  df <- subset(df, select = cat.cols)

  df.cat.summary <- data.frame(
     Count = round(sapply(df, length), 2),
     Miss = round(sapply(df, function(x) sum(length(which(is.na(x)))) / length(x)), 2),
     Card. = round(sapply(df, function(x) length(unique(x))), 2),
     Mode = names(sapply(df, function(x) sort(table(x), decreasing = TRUE)[1])),
     Mode_Freq = sapply(df, function(x) sort(table(x), decreasing = TRUE)[1]),
     Mode_pct = round((sapply(df, function(x) sort(table(x),
                                                   decreasing = TRUE)[1] / length(x)) * 100), 1),
     Mode_2 = names(sapply(df, function(x) sort(table(x), decreasing = TRUE)[2])),
     Mode_Freq_2 = sapply(df, function(x) sort(table(x), decreasing = TRUE)[2]),
     Mode_pct_2 = round((sapply(df, function(x) sort(table(x),
                                                     decreasing = TRUE)[2] / length(x)) * 100), 1)
       )

  df.cat.summary$Mode <- gsub("^.*\\.","", df.cat.summary$Mode)
  df.cat.summary$Mode_2 <- gsub("^.*\\.","", df.cat.summary$Mode_2)

  df.cat.summary <- df.cat.summary %>%
    rename(`Miss Pct.` = Miss,
           `Mode Freq.` = Mode_Freq,
           `Mode Pct.` = Mode_pct,
           `2nd Mode` = Mode_2,
           `2nd Mode Freq.` = Mode_Freq_2,
           `2nd Mode Pct.` = Mode_pct_2
           )

    return(df.cat.summary)
}

# create categorical summary
customers_clean_cat_summary <- df_cat_summary(df = customers_clean)

# display in table
kable(customers_clean_cat_summary
      #, type = "html"
      ) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "left",
                latex_options = "scale_down")
```

	Count	Miss Pct.	Card.	Mode	Mode Freq.	Mode Pct.	2nd Mode	2nd Mode Freq.	2nd Mode Pct.
lifecycle	9838	0	4	Inactive	4275	43.5	Active	4120	41.9
value_seg_1yr	9838	0	3	Medium	4916	50.0	Low	3938	40.0
value_seg_2yr	9838	0	3	Medium	4919	50.0	Low	3935	40.0
value_change	9838	0	3	Neutral	7619	77.4	Increase	1111	11.3

Model Prototyping

Postcard Response Prediction

The first part of our analytics model development will focus on a predictive model to obtain probabilities for customers to respond with a purchase to postcards.

For our predictive model, we are going to focus on our non-value segmentation variables and use 10-fold cross-validation on a 75/25 train/test split to choose the best logistic regression model. Need be, we can explore more complex models to understand a customer's propensity to respond to the postcard.

```
# set random seed for reproducibility
set.seed(123)

# create 75/25 split of train and test data indices
trainIndex <- createDataPartition(customers_clean$purchasepostcard, p = .75,
                                  list = FALSE,
                                  times = 1)

# create partitions
customers_clean$split <- ifelse(row.names(customers_clean) %in% trainIndex,
                                "Train",
                                "Test")

# convert reponse variable to categorical
customers_clean$purchasepostcard <- factor(ifelse(customers_clean$purchasepostcard == 1,
                                                  "Yes", "No"))
```

** Logistic Regression **

Model 1

Our first model is a logistic regression model with all of our variables. Our evaluation metric of interest will be the ROC score, which evaluates the trade-off between the true-positive rate (sensitivity) and the true-negative rate (specificity). Using all the variables, we see a very high ROC score of 98%.

In this case, *not making a purchase* is the positive class, so sensitivty refers to predciting a non-purchaser while specificity refers to a purchaser. We will flip these when evaluating the test set.

```
mod1_formula <- {purchasepostcard ~ dayssincefirst + dayssincelast + PurchaseMen +
    Purchasekids + Purchasesports + sharehighfashion + purchaseonline + purchlast1year +
    purchlast2years + age + clubmember + emailsubscriber + salesdriven + lifecycle}

# add controls for training model
ctrl <- trainControl(method = "repeatedcv",
                     number = 10,
                     repeats = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE
                     )

# set seed to compare models
set.seed(123)
# build model using training set
glm_all <- train(mod1_formula,
                 data = customers_clean[customers_clean$split == "Train", ],
                 method = "glm",
                 metric = "ROC",
                 trControl = ctrl)

# view results
print(glm_all)
```

```
## Generalized Linear Model
##
## 7379 samples
##   14 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 6641, 6641, 6641, 6641, 6642, 6641, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.9844435  0.9594697  0.8814605
```

```
# view summary
print(summary(glm_all$finalModel))
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5146  -0.1199  -0.0246   0.0015   3.8463
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -6.267e+00  4.981e-01 -12.581  < 2e-16 ***
## dayssincefirst     -4.852e-05  1.452e-04  -0.334 0.738330
## dayssincelast       2.426e-03  7.129e-04   3.403 0.000667 ***
## PurchaseMen         3.309e-01  1.489e-01   2.222 0.026259 *
## Purchasekids        2.129e-01  2.171e-01   0.981 0.326686
## Purchasesports     -1.167e-01  1.438e-01  -0.812 0.417078
## sharehighfashion    9.846e-01  1.754e-01   5.615 1.97e-08 ***
## purchaseonline     -1.818e-01  1.176e-01  -1.546 0.122018
## purchlast1year      1.497e-02  6.210e-04  24.104  < 2e-16 ***
## purchlast2years     5.159e-04  1.582e-04   3.261 0.001111 **
## age                -1.415e-02  6.895e-03  -2.053 0.040115 *
## clubmember          4.133e-02  1.263e-01   0.327 0.743421
## emailsubscriber     9.536e-01  1.381e-01   6.903 5.07e-12 ***
## salesdriven         6.936e-01  1.230e-01   5.639 1.71e-08 ***
## `lifecycleAt-Risk`  2.166e-01  2.350e-01   0.921 0.356843
## lifecycleInactive  -3.839e+00  7.594e-01  -5.056 4.28e-07 ***
## lifecycleNew       -4.549e-01  1.114e+00  -0.408 0.682946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8812.2  on 7378  degrees of freedom
## Residual deviance: 2080.5  on 7362  degrees of freedom
## AIC: 2114.5
##
## Number of Fisher Scoring iterations: 9
```

```
# assess model
print(anova(glm_all$finalModel, test="Chisq"))
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: .outcome
##
## Terms added sequentially (first to last)
##
##
##                      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                  7378     8812.2
## dayssincefirst      1    264.9      7377     8547.2 < 2.2e-16 ***
## dayssincelast       1   3364.1      7376     5183.2 < 2.2e-16 ***
## PurchaseMen         1      2.4      7375     5180.8 0.1221510
## Purchasekids        1     12.0      7374     5168.8 0.0005399 ***
## Purchasesports      1      1.6      7373     5167.2 0.2057169
## sharehighfashion    1     21.3      7372     5145.9 3.987e-06 ***
## purchaseonline      1    131.8      7371     5014.1 < 2.2e-16 ***
## purchlast1year      1   2756.0      7370     2258.1 < 2.2e-16 ***
## purchlast2years     1     12.0      7369     2246.1 0.0005204 ***
## age                 1      6.8      7368     2239.3 0.0089483 **
## clubmember          1      0.3      7367     2239.0 0.6002795
## emailsubscriber     1     50.3      7366     2188.6 1.287e-12 ***
## salesdriven         1     44.1      7365     2144.5 3.138e-11 ***
## `lifecycleAt-Risk`  1     27.7      7364     2116.8 1.417e-07 ***
## lifecycleInactive   1     36.2      7363     2080.7 1.803e-09 ***
## lifecycleNew        1      0.2      7362     2080.5 0.6680938
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 2

Based on the variables that lack statistical significance and low deviance reduction, we will reduce the number of variables in our logistic model and re-test. First, we will create a flag to indicate if the customer is an inactive customer and then discard the general lifecycle variable.

Additionally, we will remove variables indicating Club Member, Mens category, and sports category. We will also remove the variable for days since frist purchase. The remaining vairables showed as statistically signficant in our summary output and produced double-digit residual deviance reduction.

Our second logistic regression model shows that customers that are email subscribers, sales driven, and have higher share of purchases being high fashion are most likely to respond to the postcard. On the flip side, inactive customers (> 365 days since last purchase) show much lower odds of responding to a postcard. Simply, removing inactive customers from the list of potential postcard customers could greatly reduce the cost and increase the conversion rate of postcards.

We are able to improve our ROC and sensitivity scores slightly and maintain a very good ROC score above 98%, so we will move forward with testing our model on the test set to see how it performs on new data.

```r
customers_clean$inactive <- ifelse(customers_clean$lifecycle == "Inactive", 1, 0)

mod2_formula <- {purchasepostcard ~  dayssincelast +
    sharehighfashion + purchaseonline + purchlast1year +
    purchlast2years + age + emailsubscriber + salesdriven + inactive}

# add controls for training model
ctrl <- trainControl(method = "repeatedcv",
                     number = 10,
                     repeats = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE
                     )

# set seed to compare models
set.seed(123)
# build model using training set
glm_mod_2 <- train(mod2_formula,
                 data = customers_clean[customers_clean$split == "Train", ],
                 method = "glm",
                 metric = "ROC",
                 trControl = ctrl)

# view results
print(glm_mod_2)
```

```
## Generalized Linear Model
##
## 7379 samples
##    9 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 6641, 6641, 6641, 6641, 6642, 6641, ...
## Resampling results:
##
##   ROC        Sens       Spec
##   0.9850119  0.9606818  0.8802219
```

```r
# view summary
print(summary(glm_mod_2$finalModel))
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6063  -0.1216  -0.0242   0.0016   3.8014
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -5.9875723  0.3735679 -16.028  < 2e-16 ***
## dayssincelast     0.0015773  0.0005172   3.050  0.00229 **
## sharehighfashion  0.9745239  0.1743012   5.591 2.26e-08 ***
## purchaseonline   -0.1837724  0.1170544  -1.570  0.11642
## purchlast1year    0.0149681  0.0006175  24.241  < 2e-16 ***
## purchlast2years   0.0004885  0.0001542   3.169  0.00153 **
## age              -0.0142058  0.0068695  -2.068  0.03865 *
## emailsubscriber   0.9515286  0.1374065   6.925 4.36e-12 ***
## salesdriven       0.6770043  0.1218023   5.558 2.73e-08 ***
## inactive         -3.5718038  0.5839669  -6.116 9.57e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8812.2  on 7378  degrees of freedom
## Residual deviance: 2088.2  on 7369  degrees of freedom
## AIC: 2108.2
##
## Number of Fisher Scoring iterations: 9
```

```
# assess model
print(anova(glm_mod_2$finalModel, test="Chisq"))
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: .outcome
##
## Terms added sequentially (first to last)
##
##
##                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                              7378     8812.2
## dayssincelast     1   3466.4      7377     5345.8 < 2.2e-16 ***
## sharehighfashion  1     17.6      7376     5328.2 2.675e-05 ***
## purchaseonline    1    162.3      7375     5165.9 < 2.2e-16 ***
## purchlast1year    1   2905.2      7374     2260.7 < 2.2e-16 ***
## purchlast2years   1     13.5      7373     2247.2 0.0002352 ***
## age               1      6.8      7372     2240.4 0.0091611 **
## emailsubscriber   1     50.3      7371     2190.1 1.308e-12 ***
## salesdriven       1     44.1      7370     2146.0 3.113e-11 ***
## inactive          1     57.8      7369     2088.2 2.843e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Test Evaluation

To assess the performance of our model, we will make predictions on the test set and review the confusion matrix and ROC scores.

```
# partition test set for evaluation
customers_test <- customers_clean[customers_clean$split == "Test", ]

# predicted probabilities for test set
glmTestPred <- predict(glm_mod_2,
                       customers_test,
                       type = "prob")

# extract probability of purchase
customers_test$glm_prob <- glmTestPred[ , "Yes"]
# extract class assignment
customers_test$glm_class <- predict(glm_mod_2, customers_test)

# create confusion matrix
confusionMatrix(data = customers_test$glm_class,
                reference = customers_test$purchasepostcard,
                positive = "Yes")
```

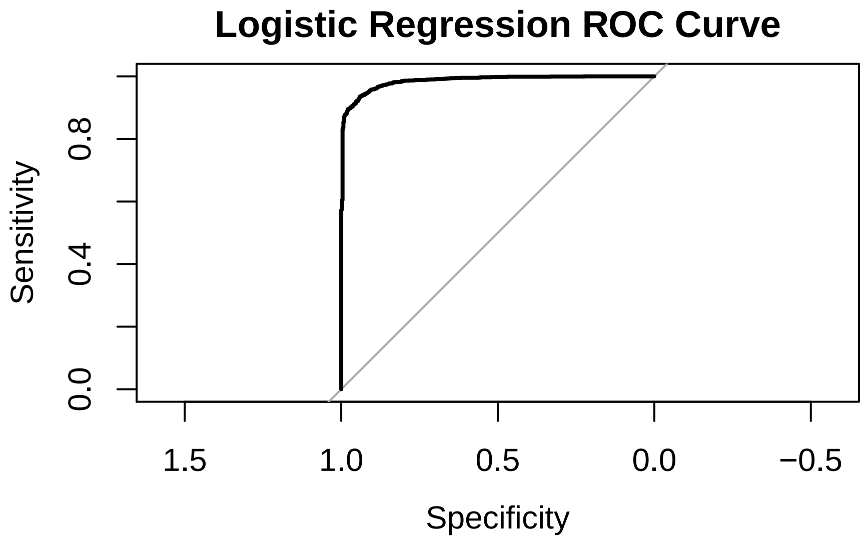
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1659   70
##        Yes   73  657
##
##                Accuracy : 0.9418
##                  95% CI : (0.9319, 0.9508)
##     No Information Rate : 0.7044
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8605
##  Mcnemar's Test P-Value : 0.8672
##
##             Sensitivity : 0.9037
##             Specificity : 0.9579
##          Pos Pred Value : 0.9000
##          Neg Pred Value : 0.9595
##              Prevalence : 0.2956
##          Detection Rate : 0.2672
##    Detection Prevalence : 0.2969
##       Balanced Accuracy : 0.9308
##
##        'Positive' Class : Yes
##
```

We can assess the ROC curve and generate the area under the ROC curve for the test set. We see we are still able to maintain our ROC score to the test sit, which is a good indication of how well our model will generalize to a similar set of customers.

```
rocCurve <- roc(response = customers_test$purchasepostcard,
                predictor = customers_test$glm_prob,
                levels = rev(levels(customers_test$purchasepostcard)))

# ROC score of test set
auc(rocCurve)
```

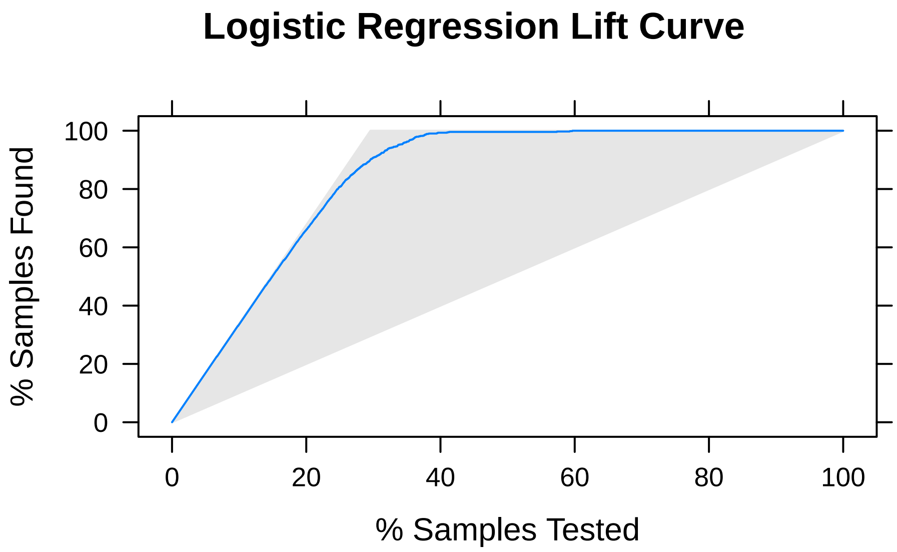
```
## Area under the curve: 0.9854
```

```
# plot of ROC curve
plot(rocCurve, main = "Logistic Regression ROC Curve")
```



Viewing the lift curve of our model, we see the logistic regression does an excellent job at detecting customers that made a purchase from the postcard.

```
liftCurve <- lift(purchasepostcard ~ glm_prob, data = customers_test,
                  class = "Yes")

plot(liftCurve, main = "Logistic Regression Lift Curve")
```



Based on the results of the prediction model, we are comfortable moving forward with testing the logistic regression model on the next batch of customers that are similar to those in this sample. Additionally, we have identified potential drivers of responding to the postcard with a purchase and can eliminate inactive customers from the distribution list.

Customer Segmentation Clustering

So far we have learned that this dataset appears to be a sample of customers that primarily shop the Swim, Eco, and Jackets categories. We have already created potential customer segmentations via customer value over the last one year, customer value over the last 2 years, customer value migration, and the stage of the customer lifecycle journey.

We will explore finding additional clusters in the dataset using 9 variables of customer purchase behavior and attributes. Since we will be using variables mixed variables in our clustering experiement, we will need to use a distance metric that can handle mixed data types (Gower's distance – each data type receives a distance calculation that works well with it).

We will explore clusters ranging from 2 to 10 in order to ensure the results are managable for marketing strategies to be developed. Too many clusters may result in the inability to create campaigns for each cluster.

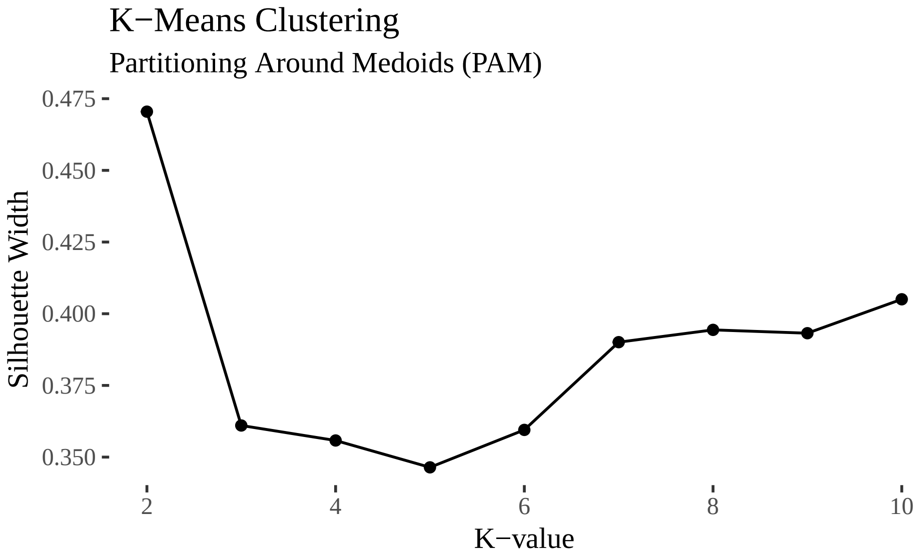
```
cluster_vars <- c("PurchaseMen", "Purchasekids", "Purchasesports", "sharehighfashion",
                  "purchaseonline", "clubmember", "emailsubscriber", "salesdriven",
                  "purchasepostcard")

gower_dist <- daisy(customers_clean[ , cluster_vars],
                    metric = "gower")

summary(gower_dist)
```

```
## 48388203 dissimilarities, summarized :
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2222  0.3867  0.3896  0.5556  1.0000
## Metric :  mixed ;  Types = I, I, I, I, I, I, I, I, N
## Number of objects : 9838
```

To assess our clustering outcomes, we will use the silhouette width to select the optimal number of clusters. This metric is a measure of how similar the points within the cluster are to one another with values ranging from -1 to 1 and larger values being better.

Our results show a k of 2 producing the largest silhouette width.

```
# Calculate silhouette width for many k values up to 10 clusters
sil_width <- c(NA)

for(i in 2:10){

  pam_fit <- pam(gower_dist,
                 diss = TRUE,
                 k = i)

  sil_width[i] <- pam_fit$silinfo$avg.width

}

# plot silhouette width to determine k with highest value (intra-cluster similarity metric)
ggplot() +
  geom_line(aes(x = 2:10, y = sil_width[2:10]))  +
  geom_point(aes(x = 2:10, y = sil_width[2:10])) +
  labs(y = "Silhouette Width", x = "K-value", title = "K-Means Clustering",
       subtitle = "Partitioning Around Medoids (PAM)")
```



We will move forward with 2 clusters and see what the results of our segmentation for this customer sample look like. One benefit of the PAM method is that the medoids represent best examples (center) of each cluster. So in additional to summary statistics of each cluster, we can use an actual record to develop a persona for each customer segementation group.

In reviewing the cluster summary, we can see that most of our customers fall into cluster #1. Cluster #1 contains customers that have a low propensity to Men and Kids categories and an affinity for the Sports category. They consist mainly of inactive, low to medium value customers. This cluster also skews towards in-store purchases with a lower response rate to postcards.

On the other hand, cluster #2 has customers with a larger affinity towards the Mens and Kids categories, skews towards online purchases, is more prone to be an email subscriber, is more sales driven, has a higher propensity to purchase via postcards, and consists of higher value, active customers. These customers also have a much more recent purchase timeframe with a median of 53 days and average of 66 days since their last purchase.

```
# fit a k=2 PAM model
pam_fit_2 <- pam(gower_dist, diss = TRUE, k = 2)

# assign clusters to the original dataset
customers_clean$cluster <- pam_fit_2$clustering

# summarize based on cluster number
cluster_summary <- customers_clean %>%
  group_by(cluster) %>%
  do(the_summary = summary(.))

# display summary
print(cluster_summary$the_summary)
```

```
## [[1]]
##  dayssincefirst dayssincelast     PurchaseMen       Purchasekids
##  Min.   :   8   Min.   :   2.0   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:1570   1st Qu.: 268.0   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :1931   Median : 570.0   Median :0.00000   Median :0.0000
##  Mean   :1734   Mean   : 690.9   Mean   :0.04292   Mean   :0.1048
##  3rd Qu.:2046   3rd Qu.:1029.0   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :2095   Max.   :2094.0   Max.   :1.00000   Max.   :1.0000
##  Purchasesports   sharehighfashion purchaseonline   purchlast1year
##  Min.   :0.0000   Min.   :0.000    Min.   :0.0000   Min.   :   0.00
##  1st Qu.:1.0000   1st Qu.:0.000    1st Qu.:0.0000   1st Qu.:   0.00
##  Median :1.0000   Median :0.000    Median :0.0000   Median :   0.00
##  Mean   :0.9751   Mean   :0.191    Mean   :0.1982   Mean   :  78.58
##  3rd Qu.:1.0000   3rd Qu.:0.310    3rd Qu.:0.0000   3rd Qu.: 102.95
##  Max.   :1.0000   Max.   :1.000    Max.   :1.0000   Max.   :2246.39
##  purchlast2years        age          clubmember     emailsubscriber
##  Min.   :   0.00   Min.   :18.00   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:  14.99   1st Qu.:32.00   1st Qu.:0.0000   1st Qu.:0.0000  
##  Median : 142.94   Median :38.00   Median :0.0000   Median :0.0000  
##  Mean   : 291.98   Mean   :39.26   Mean   :0.2606   Mean   :0.4006  
##  3rd Qu.: 407.74   3rd Qu.:45.00   3rd Qu.:1.0000   3rd Qu.:1.0000  
##  Max.   :4159.28   Max.   :79.00   Max.   :1.0000   Max.   :1.0000  
##   salesdriven     purchasepostcard    lifecycle    value_seg_1yr
##  Min.   :0.0000   No :6207          Active  :1031   Low   :3927  
##  1st Qu.:0.0000   Yes: 386          At-Risk :1273   Medium:2601  
##  Median :0.0000                     Inactive:4275   High  :  65  
##  Mean   :0.1247                     New     :  14                
##  3rd Qu.:0.0000                                                  
##  Max.   :1.0000                                                  
##  value_seg_2yr   value_change     split              inactive     
##  Low   :3783   Decrease: 908   Length:6593        Min.   :0.0000  
##  Medium:2709   Increase: 734   Class :character   1st Qu.:0.0000  
##  High  : 101   Neutral :4951   Mode  :character   Median :1.0000  
##                                                   Mean   :0.6484  
##                                                   3rd Qu.:1.0000  
##                                                   Max.   :1.0000  
##     cluster 
##  Min.   :1  
##  1st Qu.:1  
##  Median :1  
##  Mean   :1  
##  3rd Qu.:1  
##  Max.   :1  
## 
## [[2]]
##  dayssincefirst dayssincelast     PurchaseMen      Purchasekids   
##  Min.   :  26   Min.   :  2.00   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:1828   1st Qu.: 24.00   1st Qu.:0.0000   1st Qu.:1.0000  
##  Median :2025   Median : 53.00   Median :0.0000   Median :1.0000  
##  Mean   :1847   Mean   : 66.33   Mean   :0.4684   Mean   :0.9088  
##  3rd Qu.:2065   3rd Qu.: 90.00   3rd Qu.:1.0000   3rd Qu.:1.0000  
##  Max.   :2095   Max.   :365.00   Max.   :1.0000   Max.   :1.0000  
##  Purchasesports   sharehighfashion purchaseonline   purchlast1year  
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :   7.99  
##  1st Qu.:0.0000   1st Qu.:0.0300   1st Qu.:0.0000   1st Qu.: 266.83  
##  Median :0.0000   Median :0.3300   Median :1.0000   Median : 470.73  
##  Mean   :0.4348   Mean   :0.3541   Mean   :0.6915   Mean   : 587.85  
##  3rd Qu.:1.0000   3rd Qu.:0.6000   3rd Qu.:1.0000   3rd Qu.: 778.79  
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :2495.34  
##  purchlast2years        age          clubmember     emailsubscriber
##  Min.   :  18.98   Min.   :19.00   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.: 607.73   1st Qu.:30.00   1st Qu.:0.0000   1st Qu.:1.0000  
##  Median :1054.66   Median :35.00   Median :0.0000   Median :1.0000  
##  Mean   :1309.19   Mean   :35.99   Mean   :0.2823   Mean   :0.8955  
##  3rd Qu.:1764.80   3rd Qu.:40.00   3rd Qu.:1.0000   3rd Qu.:1.0000  
##  Max.   :4979.31   Max.   :71.00   Max.   :1.0000   Max.   :1.0000  
##   salesdriven     purchasepostcard    lifecycle    value_seg_1yr
##  Min.   :0.0000   No : 805          Active  :3089   Low   :  11  
##  1st Qu.:0.0000   Yes:2440          At-Risk : 137   Medium:2315  
##  Median :0.0000                     Inactive:   0   High  : 919  
##  Mean   :0.3781                     New     :  19                
##  3rd Qu.:1.0000                                                  
##  Max.   :1.0000                                                  
##  value_seg_2yr   value_change     split              inactive    cluster 
##  Low   : 152   Decrease: 200   Length:3245        Min.   :0   Min.   :2  
##  Medium:2210   Increase: 377   Class :character   1st Qu.:0   1st Qu.:2
##  High  : 883   Neutral :2668   Mode  :character   Median :0   Median :2
##                                                   Mean   :0   Mean   :2
##                                                   3rd Qu.:0   3rd Qu.:2
##                                                   Max.   :0   Max.   :2
```

Information from each of these clusters can be used to create campaigns to target customers in each cluster. For example, customer in cluster 2 can be provided messaging around new items with sales promotions sprinkled in based on the value of the customer. Customers in cluster 1 can be targeted with campaigns for sports with the intent of re-activating lapsed customers. Offers can be considered based on the value change of the customer with customers showing increases in value getting deeper offers as an incentive to re-engage.

New Customers

There are a handful of steps we can take to increase the liklihood of a new customer returning. The simplest strategy is to send new customers a "Thank You" follow-up, whether by e-mail or traditional mail. This can start from the moment the customer received their order in the mail, or from the store associate ringing up the transaction. If a customer is flagged in the system as being a new customer, processes can trigger to engage these customers differently over the course of the next 90 days.

In addition to a genuine "Thank you," these customers should be provided with information that helps them succeed in their relationship with H&M. What resources can these customers be given that provides them value WITHOUT having to transact again? For example, is there a free newsletter, style guide, etc. that can be sent to the customer to help educate them on other styles that might go well with their purchase?

As this process gets going, additional analytical steps can be taken to identify predictors of a high-value customer and/or what items they might want to buy next. For example, are there certain items that over-index in terms of high-value customer acquisition? Is there market basket analysis data available on these particular items to develop a drip campaign that consists of H&M resources along with offers for additional items? What channels or offers do customers typically engage with for their 2nd and 3rd purchases? Do they respond to email campaigns, or is their repeat purchase agnostic of any offer? Do they purchase a similar or complimentary item?

To add additional fuel to the cycle, what customer feedback data is available? How do new customers rate their satisfaction and how do eventual repeat customers compare to those that churn? Are their common topics in the comments or areas of improvement that are available in order to drive the satisfaction of future new customers?

Results Activation

To conclude, we have several key takeaways as it relates to our initial business goals to activate our results.

1. Postcard Send-Outs

We have postcards in several markets and would like to send them to the right customer.

- Use the logistic regression model to predict the probability of a customer responding to a postcard and monitor results over time
- Remove inactive customers from the potential pool of recipients and develop a separate re-activation campaign
- Create regression model to predict value of response to postcard and then in conjunction with the logistic model, determine the expected value of mailing a postcard to each customer
  - Only mail postcards to customers with an expected value greater than the cost of the postcard

2. Strategic Segmentation

We want to learn more about our customer base to support customer insight in the organization. We also want to use it for how we will work with customers, e.g. which customer groups we will target different activities.

- Use results of the clustering to better understand the attributes of micro-customer groups and tailor marketing campaigns accordingly
- Depending on the availability of resources, perform the clustering segmentation to include a larger number of clusters for more specific targeting
- Obtain additional relevant variables to include in the results

- Create humanized personas that summarize the behavior of these customers to distribute to relevant parties within the organization (e.g. “This is Jane. She subscribes to emails and likes to make online purchases.”)

3. New Customers

A project that works with Online customers needs our analytical help. Many customers shop once, but never come back. How do we get new customers to return?

- Map out the customer journey lifecycle and develop marketing strategies that trigger based on different phases
- Start simple and thank new customers for starting a relationship with H&M and identify ways to provide customer value outside of transactions
- Apply data-mining techniques to understand predictors of high-value customers and use results to drip incentives in egagement for the customer to repeat purchase (e.g. relevant product promotional offers)
- Leverage customer feedback to understand drivers of satisfaction and how to relate to increased purchase frequency