Cold Started Analysis

Cesar L. Jaitman Labaton

Table of Contents

1.0 Introduction

Advertising and promotion are pivotal to the marketing of the American food supply. The U.S. food marketing system is the second largest advertiser in the American economy, and a leading supporter of network, spot, and cable television, newspapers, magazines, billboards, and commercial radio. Groceries account for about 70 percent of all manufacturers' coupons. Food manufacturers also spend massive amounts promoting the product to the retailer through discounts and allowances, incentives, and actual slotting allowances in order to secure scarce space on the Nation's grocery shelves.

Why so much advertising? There are several reasons for it. First, the food market is huge, capturing about 12.5 percent of consumer income, and there is vigorous competition among food firms to compete for this market. Second, food is a repeat-purchase item, lending itself to swift changes in consumer opinions. Third, food is one of the most highly branded items in the American economy, thus lending itself to major advertising. [1]

This analysis tries to understand human behavior and their purchasing options when they go to a new store (we will call them cold start shoppers). We define cold start shoppers, customers that perform a purchase for the first time in any of the stores that the dataset covers. I will also split the data, and compare the first purchase of all customers and analyze whether or not there is a correlation between other customer's first purchase and if the first purchase can give us a better understanding of purchasing patterns after the customer comes back for the 2nd purchase (whether or not is the same store or the same products).

I will also use clustering to see whether or not we can classify these customers into subcategories based on different metrics (age range, income, location, products purchased). If we can learn what is popular in certain areas for specific age ranges, we can target the appropriate coupons to the right audience without wasting any resources from advertisers, but also not frustrating and encumbering potential new clients with useless advertisements.

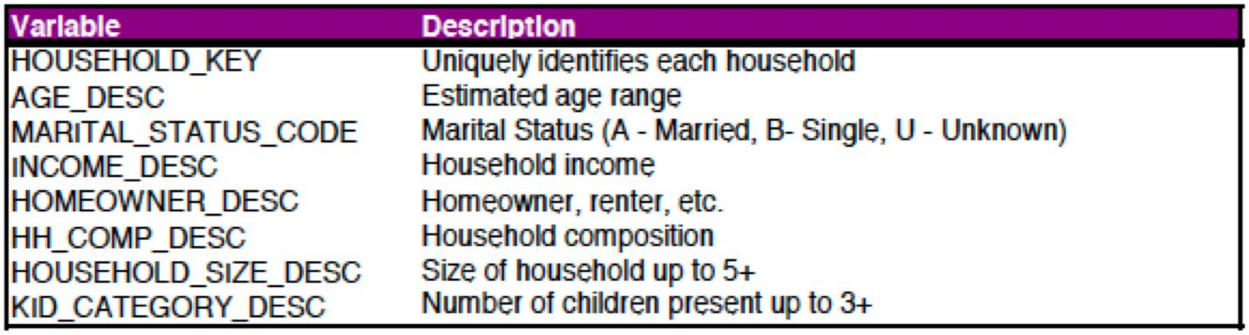
[1] https://www.ers.usda.gov/webdocs/publications/42215/5838_aib750i_1_.pdf?v=41055

2.0 Data Overview

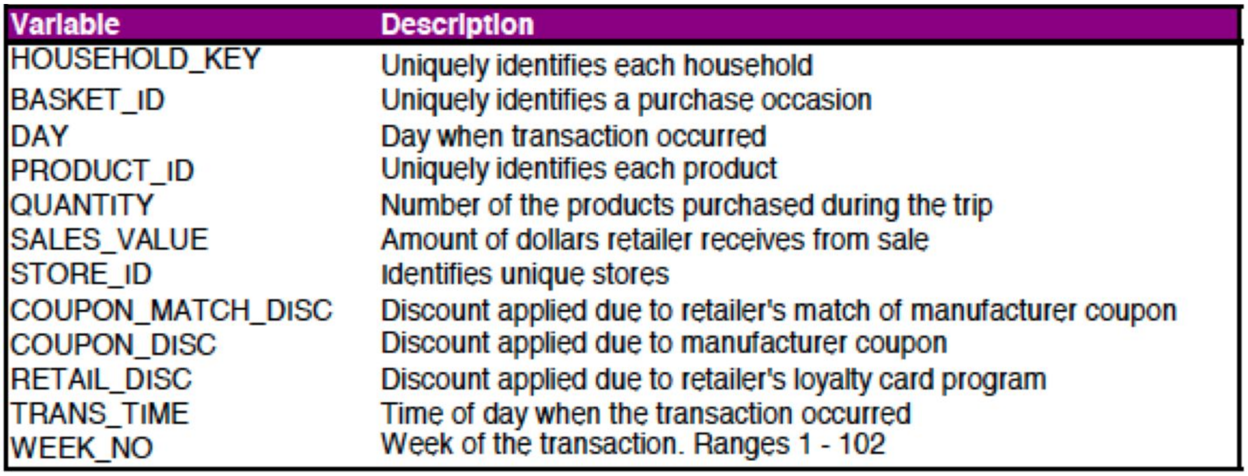
2.1 Glossary of each column

The dataset contains 39 columns with 2993708 rows. Our target columns will the client information and splitting the data with clients that only have one purchase and the first purchase made.

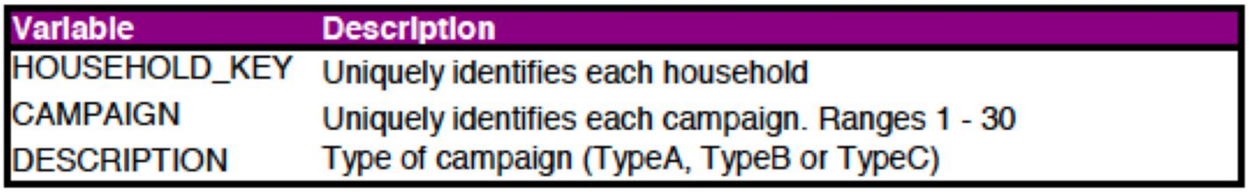
Overview of the different columns and their descriptions:



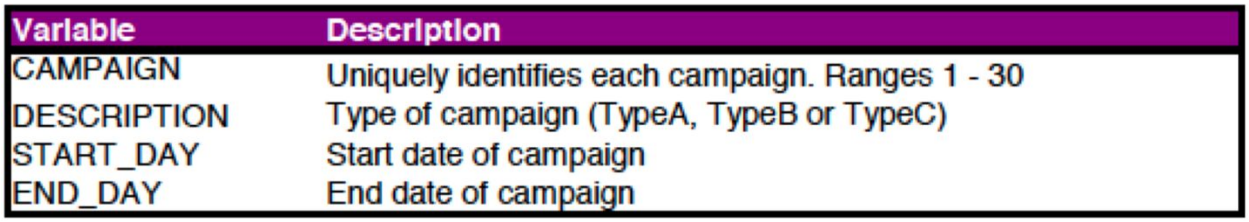
Variable	Description
HOUSEHOLD_KEY	Uniquely identifies each household
AGE_DESC	Estimated age range
MARITAL_STATUS_CODE	Marital Status (A - Married, B- Single, U - Unknown)
INCOME_DESC	Household income
HOMEOWNER_DESC	Homeowner, renter, etc.
HH_COMP_DESC	Household composition
HOUSEHOLD_SIZE_DESC	Size of household up to 5+
KID_CATEGORY_DESC	Number of children present up to 3+



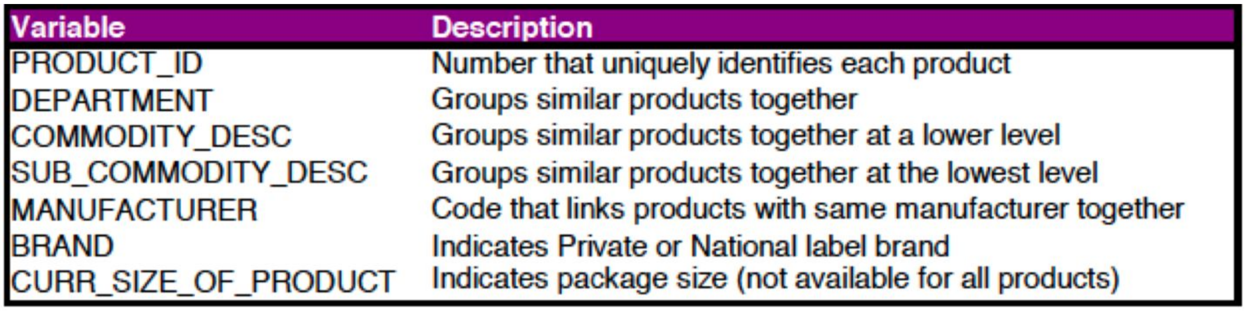
Variable	Description
HOUSEHOLD_KEY	Uniquely identifies each household
BASKET_ID	Uniquely identifies a purchase occasion
DAY	Day when transaction occurred
PRODUCT_ID	Uniquely identifies each product
QUANTITY	Number of the products purchased during the trip
SALES_VALUE	Amount of dollars retailer receives from sale
STORE_ID	Identifies unique stores
COUPON_MATCH_DISC	Discount applied due to retailer's match of manufacturer coupon
COUPON_DISC	Discount applied due to manufacturer coupon
RETAIL_DISC	Discount applied due to retailer's loyalty card program
TRANS_TIME	Time of day when the transaction occurred
WEEK_NO	Week of the transaction. Ranges 1 - 102



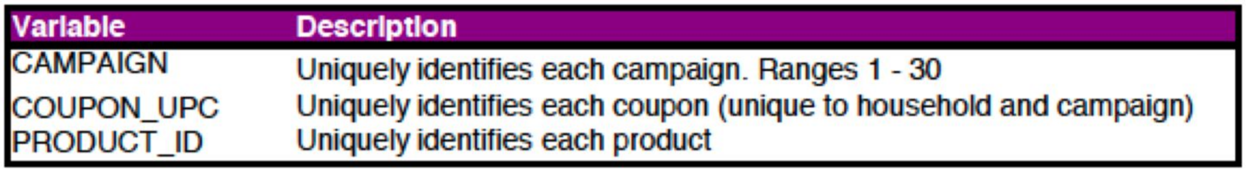
Variable	Description
HOUSEHOLD_KEY	Uniquely identifies each household
CAMPAIGN	Uniquely identifies each campaign. Ranges 1 - 30
DESCRIPTION	Type of campaign (TypeA, TypeB or TypeC)



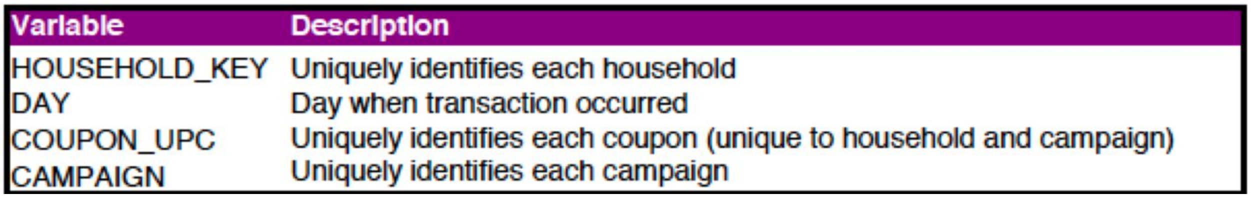
Variable	Description
CAMPAIGN	Uniquely identifies each campaign. Ranges 1 - 30
DESCRIPTION	Type of campaign (TypeA, TypeB or TypeC)
START_DAY	Start date of campaign
END_DAY	End date of campaign



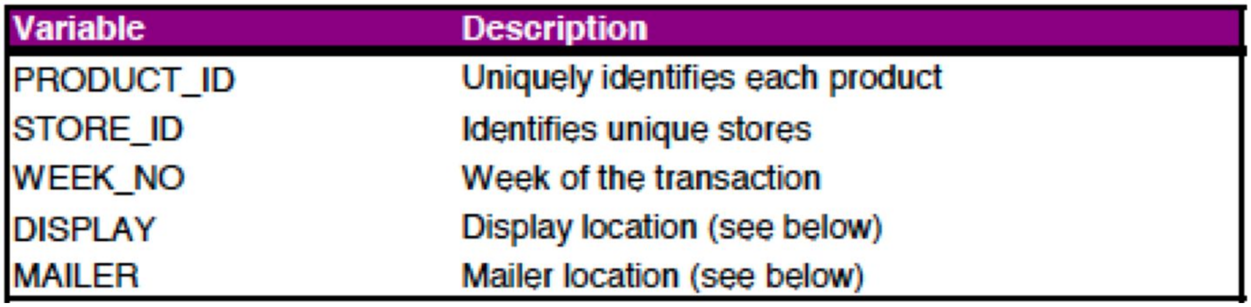
Variable	Description
PRODUCT_ID	Number that uniquely identifies each product
DEPARTMENT	Groups similar products together
COMMODITY_DESC	Groups similar products together at a lower level
SUB_COMMODITY_DESC	Groups similar products together at the lowest level
MANUFACTURER	Code that links products with same manufacturer together
BRAND	Indicates Private or National label brand
CURR_SIZE_OF_PRODUCT	Indicates package size (not available for all products)



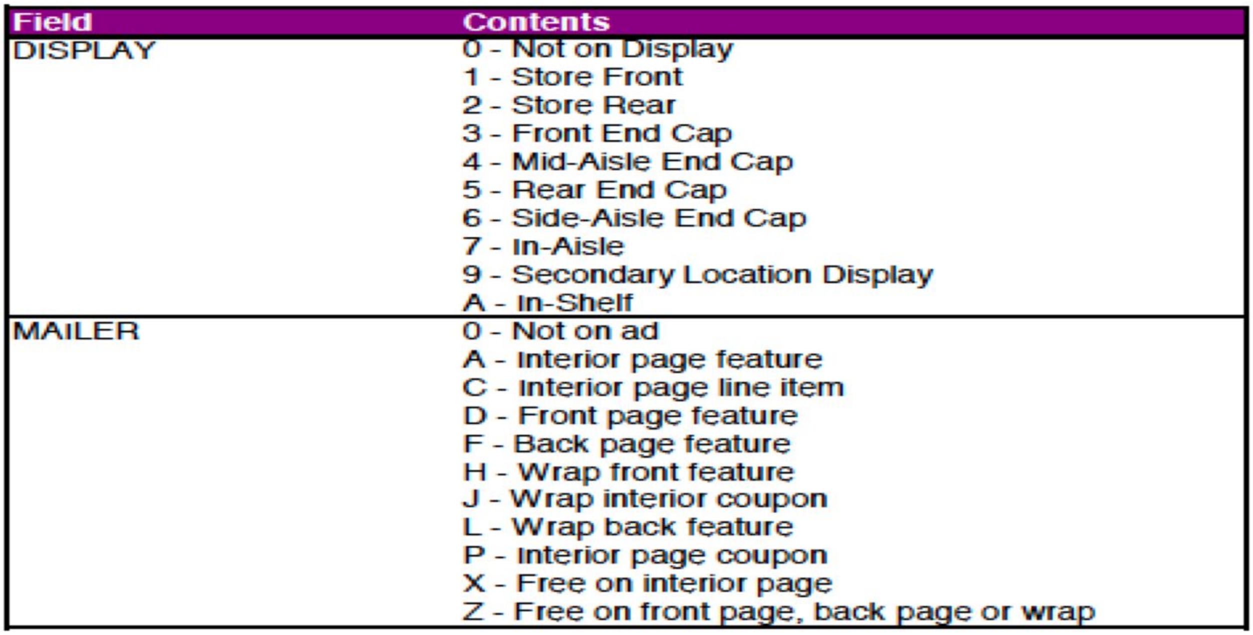
Variable	Description
CAMPAIGN	Uniquely identifies each campaign. Ranges 1 - 30
COUPON_UPC	Uniquely identifies each coupon (unique to household and campaign)
PRODUCT_ID	Uniquely identifies each product



Variable	Description
HOUSEHOLD_KEY	Uniquely identifies each household
DAY	Day when transaction occurred
COUPON_UPC	Uniquely identifies each coupon (unique to household and campaign)
CAMPAIGN	Uniquely identifies each campaign



Variable	Description
PRODUCT_ID	Uniquely identifies each product
STORE_ID	Identifies unique stores
WEEK_NO	Week of the transaction
DISPLAY	Display location (see below)
MAILER	Mailer location (see below)



Field	Contents
DISPLAY	0 - Not on Display 1 - Store Front 2 - Store Rear 3 - Front End Cap 4 - Mid-Aisle End Cap 5 - Rear End Cap 6 - Side-Aisle End Cap 7 - In-Aisle 9 - Secondary Location Display A - In-Shelf
MAILER	0 - Not on ad A - Interior page feature C - Interior page line item D - Front page feature F - Back page feature H - Wrap front feature J - Wrap interior coupon L - Wrap back feature P - Interior page coupon X - Free on interior page Z - Free on front page, back page or wrap

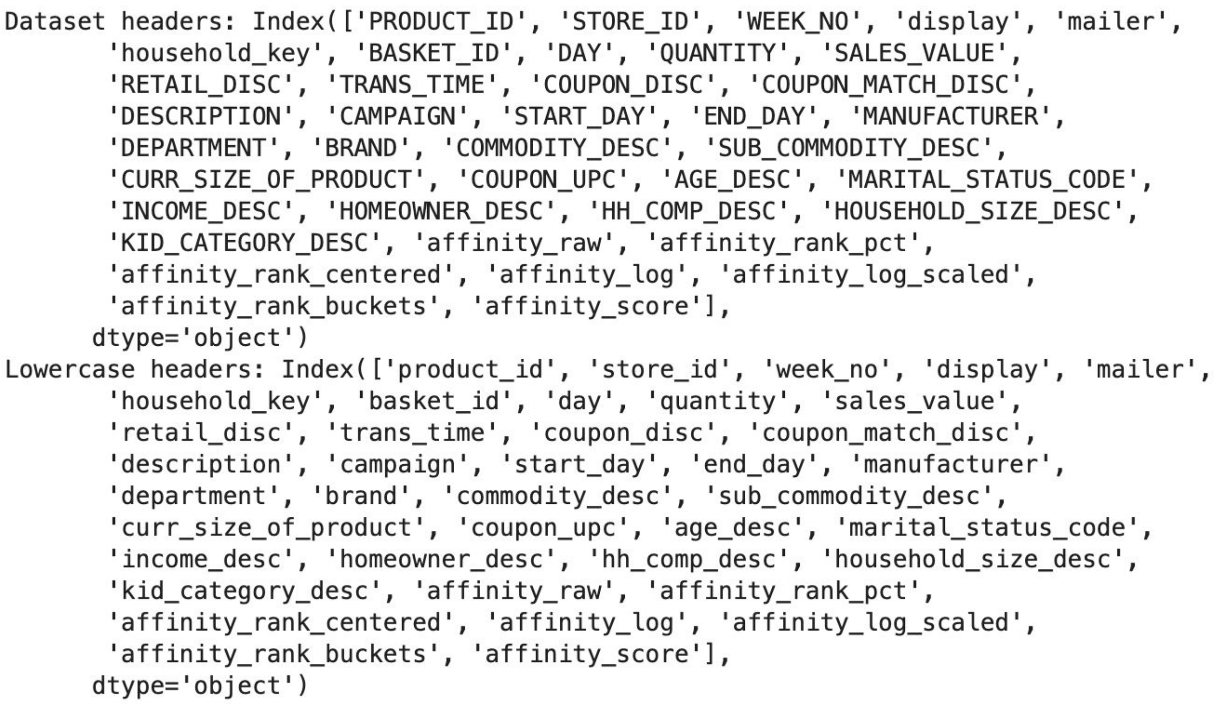
[2]

3.0 Data Pre-Processing

3.1 Headers

Changing all headers to lower case for a uniform and easy usage:

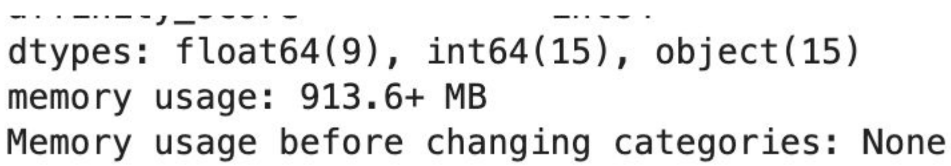
[2] dunnhumby - The Complete Journey User Guide.pdf



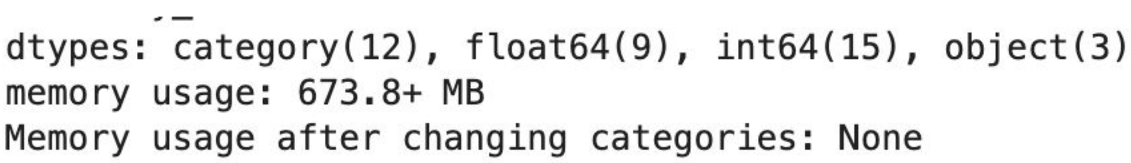
```
Dataset headers: Index(['PRODUCT_ID', 'STORE_ID', 'WEEK_NO', 'display', 'mailer',
       'household_key', 'BASKET_ID', 'DAY', 'QUANTITY', 'SALES_VALUE',
       'RETAIL_DISC', 'TRANS_TIME', 'COUPON_DISC', 'COUPON_MATCH_DISC',
       'DESCRIPTION', 'CAMPAIGN', 'START_DAY', 'END_DAY', 'MANUFACTURER',
       'DEPARTMENT', 'BRAND', 'COMMODITY_DESC', 'SUB_COMMODITY_DESC',
       'CURR_SIZE_OF_PRODUCT', 'COUPON_UPC', 'AGE_DESC', 'MARITAL_STATUS_CODE',
       'INCOME_DESC', 'HOMEOWNER_DESC', 'HH_COMP_DESC', 'HOUSEHOLD_SIZE_DESC',
       'KID_CATEGORY_DESC', 'affinity_raw', 'affinity_rank_pct',
       'affinity_rank_centered', 'affinity_log', 'affinity_log_scaled',
       'affinity_rank_buckets', 'affinity_score'],
      dtype='object')
Lowercase headers: Index(['product_id', 'store_id', 'week_no', 'display', 'mailer',
       'household_key', 'basket_id', 'day', 'quantity', 'sales_value',
       'retail_disc', 'trans_time', 'coupon_disc', 'coupon_match_disc',
       'description', 'campaign', 'start_day', 'end_day', 'manufacturer',
       'department', 'brand', 'commodity_desc', 'sub_commodity_desc',
       'curr_size_of_product', 'coupon_upc', 'age_desc', 'marital_status_code',
       'income_desc', 'homeowner_desc', 'hh_comp_desc', 'household_size_desc',
       'kid_category_desc', 'affinity_raw', 'affinity_rank_pct',
       'affinity_rank_centered', 'affinity_log', 'affinity_log_scaled',
       'affinity_rank_buckets', 'affinity_score'],
      dtype='object')
```

3.2 Defining column types

The dataset has almost 3 million rows with 39 rows. I change some columns types to better categorize them, but also to allow better performance when processing the entire dataset in my local machine:



```
dtypes: float64(9), int64(15), object(15)
memory usage: 913.6+ MB
Memory usage before changing categories: None
```

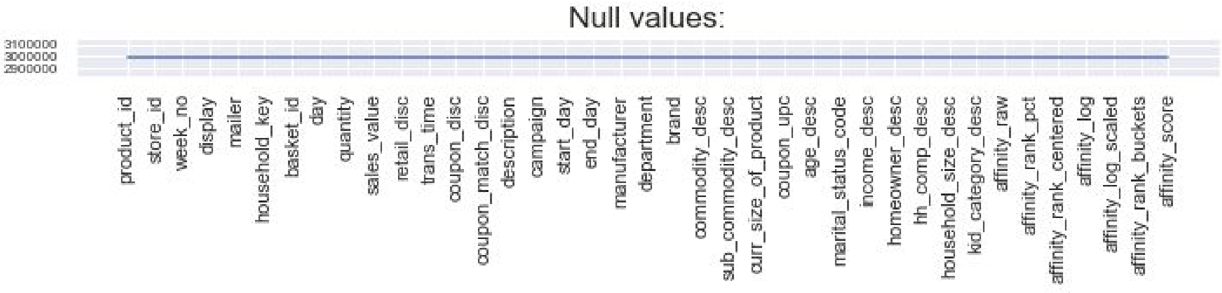


```
dtypes: category(12), float64(9), int64(15), object(3)
memory usage: 673.8+ MB
Memory usage after changing categories: None
```

By changing some object columns to categories, I was able to save 250GBs that allows for a faster render of the dataset and plotting some categories for future comparison.

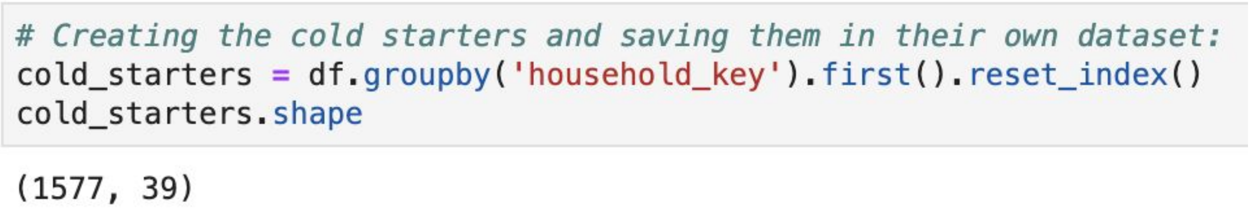
3.3 Null Values

The dataset does not have any null values present:



3.4 Creating Cold Starters Dataset

By focusing on the first purchase of a customer, I am extracting the 1st purchases recorded in the dataset. As a result, we gather 1577 unique customer that can be compared to the different relationships that will be extracted next.



```
# Creating the cold starters and saving them in their own dataset:
cold_starters = df.groupby('household_key').first().reset_index()
cold_starters.shape
```

```
(1577, 39)
```

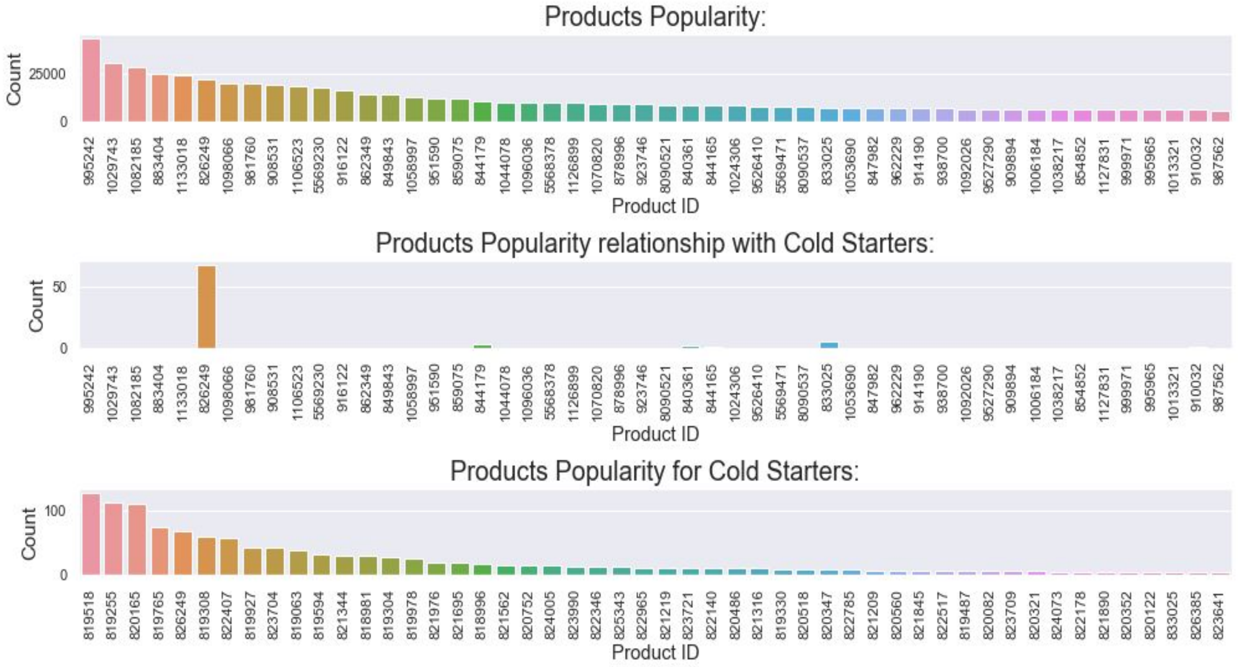
4.0 Data Analysis

In order to better understand our dataset, we want to understand the data and what relations some columns have with each other’s. Especially if cold starter users tend to shop in specific shops, for specific products, and if some decisions are swayed by the income, age and location of the product within the store. We will compare the analysis with the cold starter dataset that was separated from the main dataset.

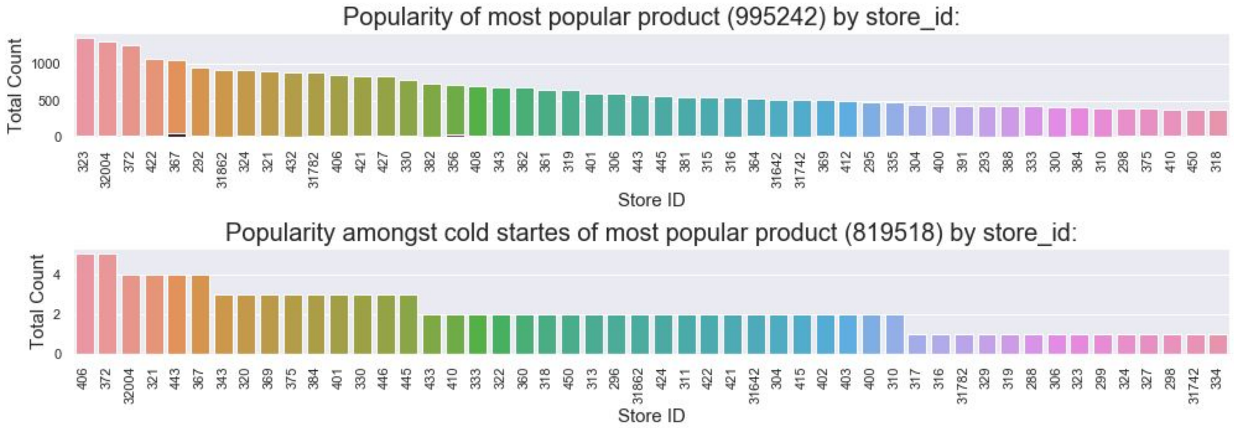
4.1 Product Popularity

Product 995242 tends to sell at least 25% overall better than the 2nd most popular product. However, this product is usually not purchased by cold starters. The most popular is product 819518 with over 130+ counts. Focusing on this product can help us better understand if there is a correlation with necessity, locations of products and whether or not coupons were the cause to help sway the decision to buy this product in the first place.

There is also not a big correlation with the products most often purchased (as seen in the 2nd graph). Only one of the product's show some similar popularity, but overall there is no correlation.

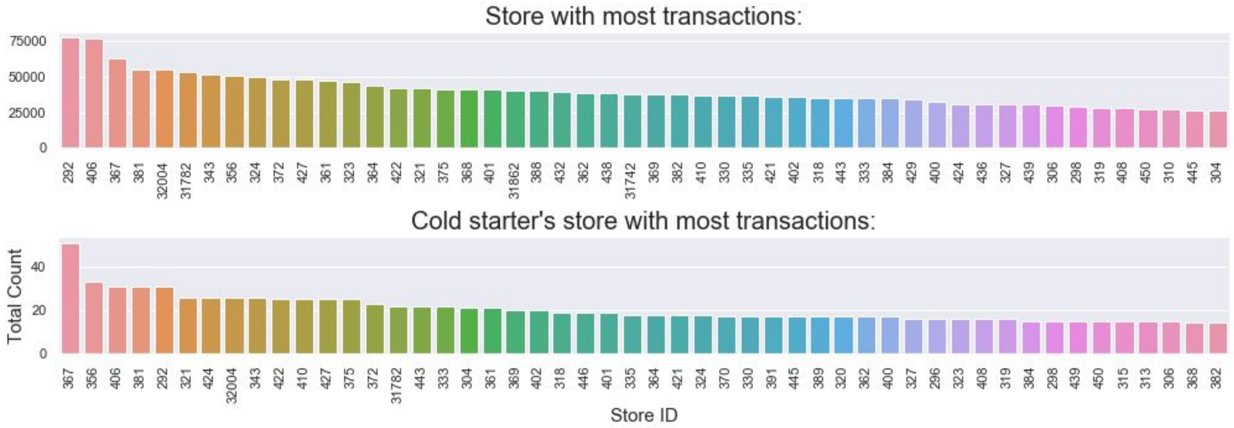


Measuring the performance of the most popular product against the most popular cold starter products, we can see a slight correlation between the store ids. Even tough, the ones that have the most transactions do not match with the ones that recorded first time purchases, we can see that customers might be going to the same stores. Also, the measurement can be skewed since the same customer could have gone multiple times to the same store (either for big or small purchases) and increase the total number of transactions made.



4.2 Store popularity

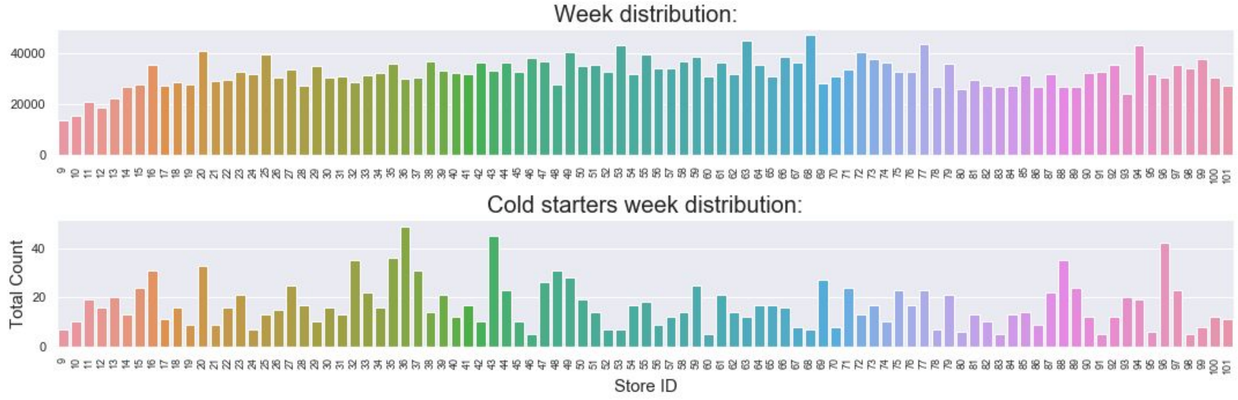
Measuring store popularity can give us an idea of the location of different customers. From location we can correlate the income and the age of the customer. Some neighborhoods tend to have a higher income median, whereas others have an older population. Understanding the different locations, will help me understand if these areas have a direct relationship to what cold starter and returning customers are doing during their trips to the grocery store.



Store 292 made close to 78K transactions in the dataset (either selling 1 product or multiple), followed closely by store 406. The rest of the stores fall behind by 12K and slowly descend in a linear decline. Later one we'll check whether or not there's a relation between the stores that make the most

4.3 Week distribution

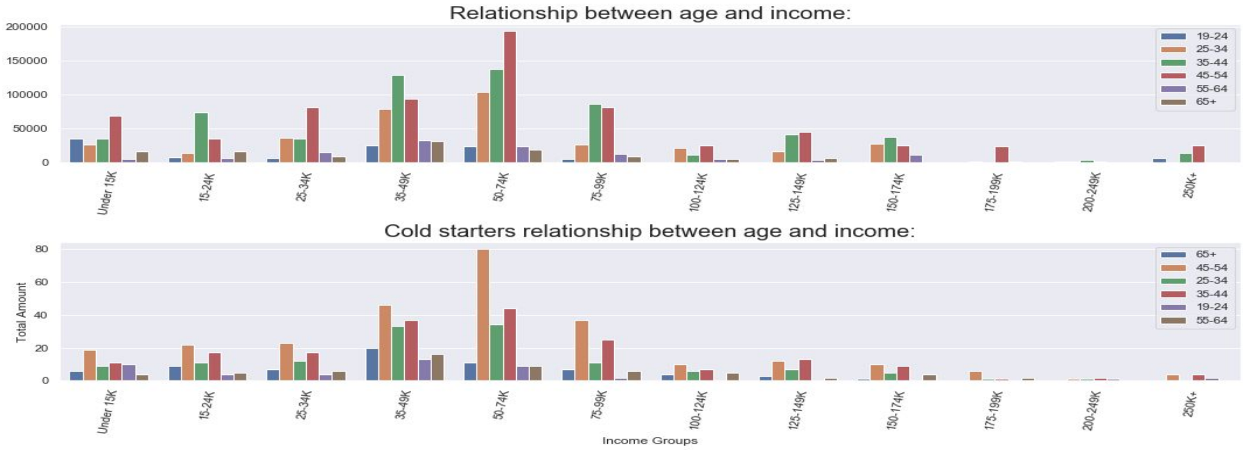
We also want to comprehend whether or not some customers shop for the first time more often in certain times of the year, in comparison to others (for example Christmas or Thanksgiving).



At the beginning of the dataset, when data starts to get collected; there is a slow increase in activity amongst the customers that are being tracked. After week 16 we can see that it plateaus and then evens out into a uniform distribution. In the case of the cold starters, their distribution is more uneven with some weeks showing more than 40+ new customers making their first purchase, to sometimes all the way down to 2 or 3 new customers.

4.4 Income relationship with age

One of the big factors that we want to test is whether or not cold starter users will use coupons based on their income and age. As the income is higher, the customer will spend less time shopping for coupons, since that customer has more disposable income than the lower income customer. Also, we want to test whether or not if older customers are more susceptible to use coupons.

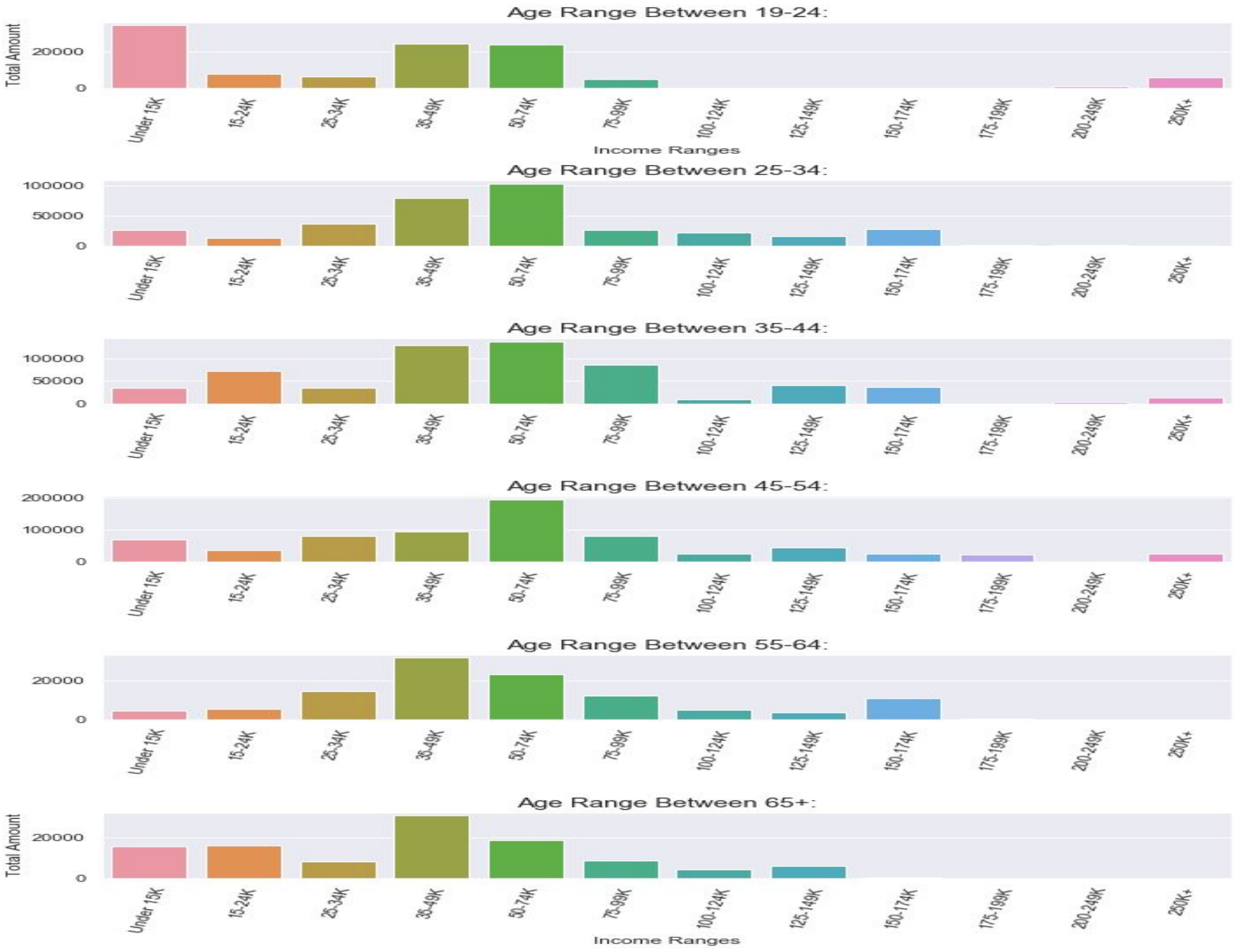


As displayed, lower income customers tend to use coupons more often than customer with much higher wages (all incomes over 100k are displayed in the later half part of the graph). Also, customers between the age group of 35-54 tend to be the ones that use the most coupons. Some irregularities are:

- Why aren't the lowest income customer using the most? Why customer that make 75-99K use more coupons that customers that make less than 15K?
- Overall, people that are between the ages 35-44 tend to use coupons more often than any other age group. Why is this?
- Why is there a big spike of usage for customers between the age of 45-54 when making 50-74K?

4.5 Income Distributions in Dataset

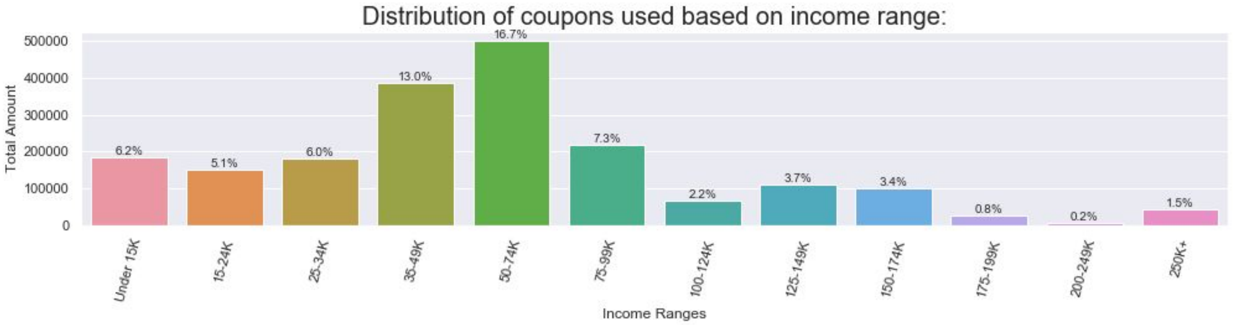
We want to also make sure that all income groups are represented or distributed in an equal manner since having too many of the same income group could skew the overall conclusion of the dataset. Income is also one of the main factors' customer might use coupons, go to a store, or even purchase certain products.

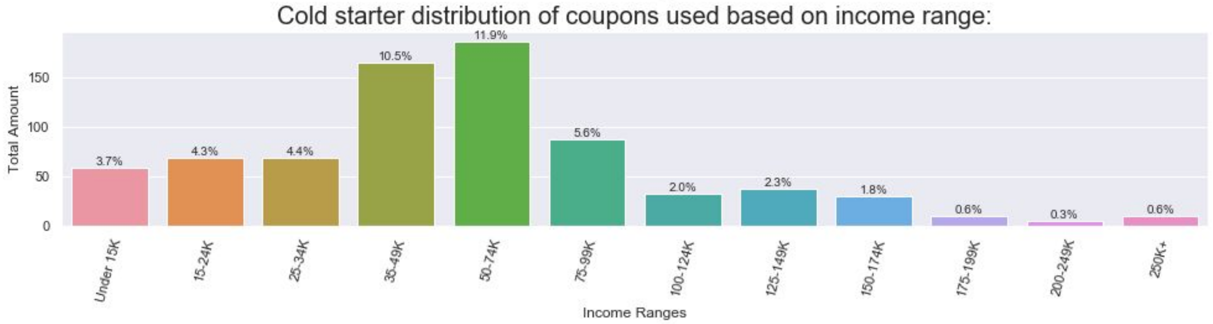


As we can see, customer that make between 35-74K are the ones that show the most transactions in the dataset. The other ages are very similar to each other despite the age group. One interesting finding is that the youngest age group with the lowest income is one that also shows some of the most transactions overall. Besides the group that makes between 15-24K, all of them stay have a uniform distribution.

4.6 Distribution of data set

We want to make sure if most of the dataset is evenly distributed or a certain age group is better represented in the overall dataset.

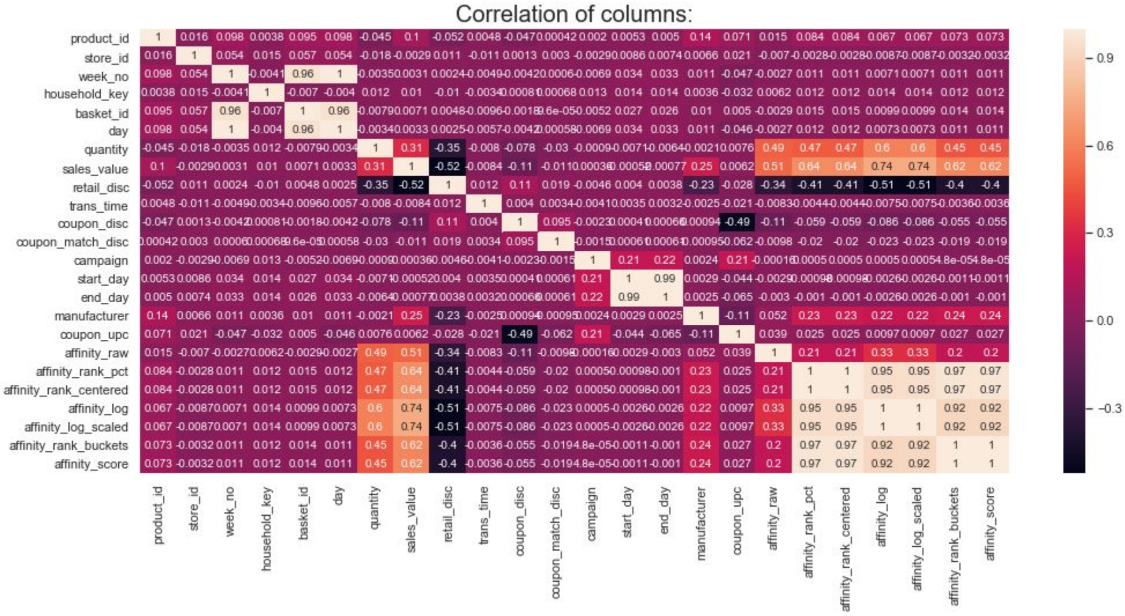


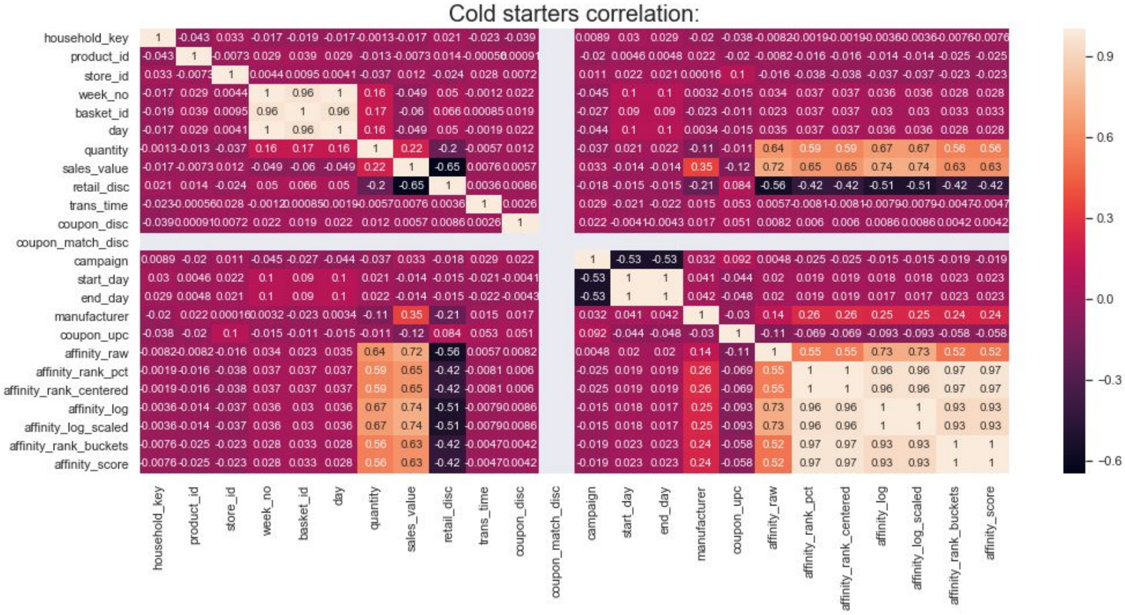


Most of the dataset is distributed between the lower income ranges; especially between 35-49K and 50-74K, compromising 29.7% of the total dataset. We can focus more on these group since we have more data to test on. Also, it might explain why the income distribution is focused in these range more than other ranges.

5.0 Correlations of Dataset

Finding correlations between the different columns, will help us understand if some of the columns can be analyzed together and give us a better insight on different statistics.

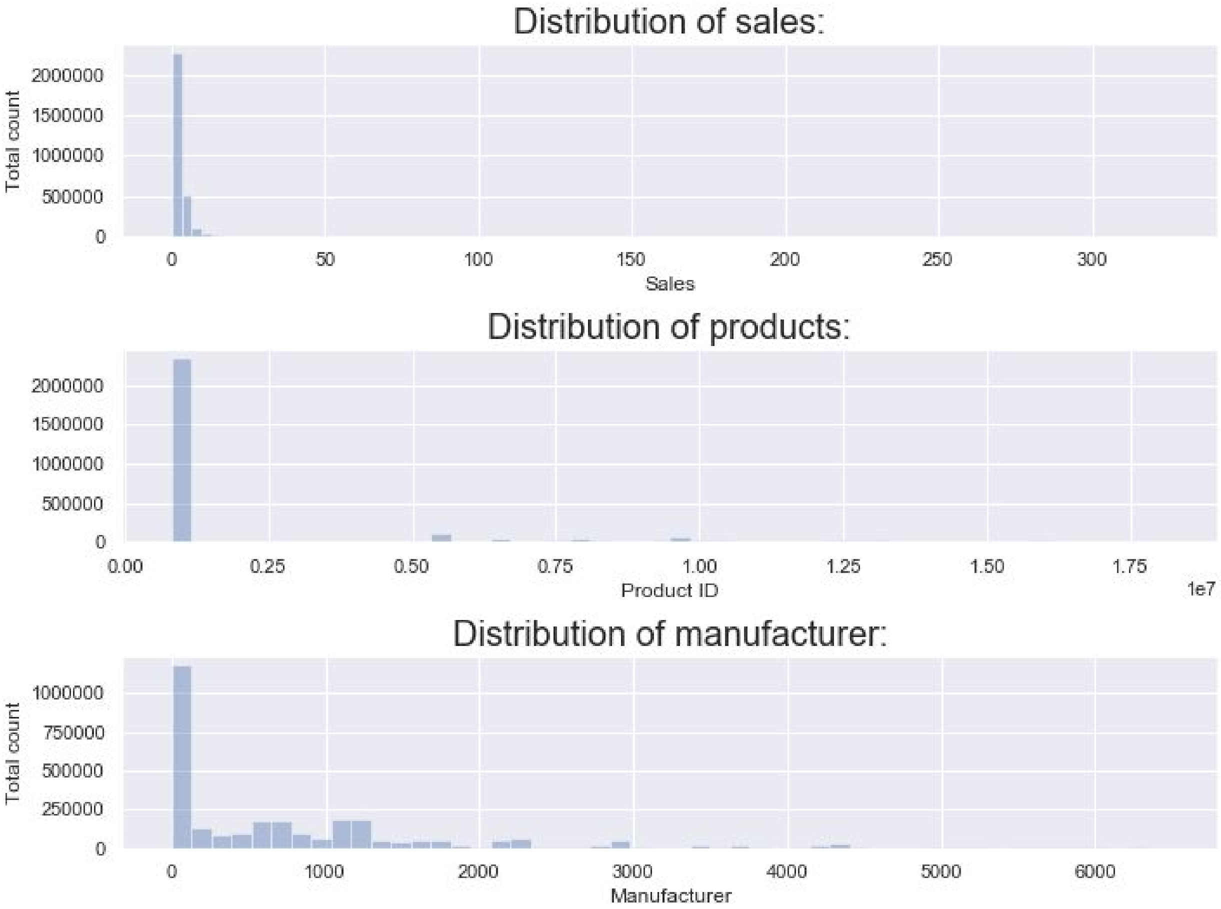




Comparing both datasets, we can see that correlations stays similar to each other and some correlations can be noticed:

- Affinity information have a correlation between each other.
- Quantity and sales value have a close relationship wo the affinity information.
- There’s a slight correlation with the manufacturer and the sales value.
- Most of the dataset does not show a relationship between the other columns.

5.1 Analyzing Correlations

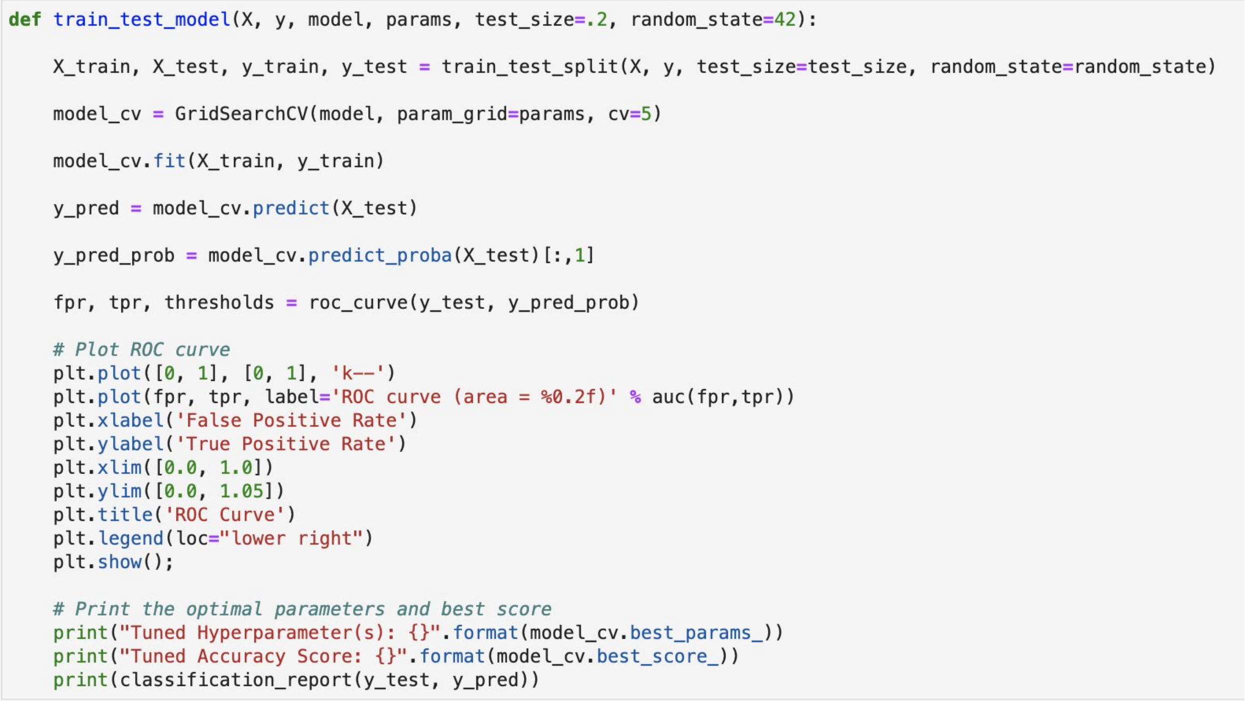


6.0 Modelling

In order to implement different models to our dataset, I used the “df_clean” dataset that has the appropriate column types. I’m using “sales_value” as the target and chose: 'sales_value', 'product_id', 'store_id', 'week_no', 'retail_disc', 'age_desc', 'marital_status_code', 'income_desc' as the features.

6.0.1 Model Function

Introduced a function that is able to split the data an and chose different models for faster changes of models in future references.



```
def train_test_model(X, y, model, params, test_size=.2, random_state=42):

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=random_state)

    model_cv = GridSearchCV(model, param_grid=params, cv=5)

    model_cv.fit(X_train, y_train)

    y_pred = model_cv.predict(X_test)

    y_pred_prob = model_cv.predict_proba(X_test)[:,1]

    fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)

    # Plot ROC curve
    plt.plot([0, 1], [0, 1], 'k--')
    plt.plot(fpr, tpr, label='ROC curve (area = %0.2f)' % auc(fpr,tpr))
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.title('ROC Curve')
    plt.legend(loc="lower right")
    plt.show();

    # Print the optimal parameters and best score
    print("Tuned Hyperparameter(s): {}".format(model_cv.best_params_))
    print("Tuned Accuracy Score: {}".format(model_cv.best_score_))
    print(classification_report(y_test, y_pred))
```

6.0.3 Creating/Splitting the data

Split the data outside the function for simpler models that not require model definition and would be able to use the split data outside the function.

6.1 Linear Regression

A linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called **multiple linear regression**. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable[3]

Using linear regression, I was able to adjust the different features that I use from the dataset in order to guess the different sales value of the different products. By dividing the dataset into 75% training and 25% testing, I was able to get a ~30% estimation rate.

```
Model: 1.7572611340484525, Actual: 6.58, Percentile: 26.71%
Model: 2.8658017610586075, Actual: 2.5, Percentile: 114.63%
Model: 4.743590848601831, Actual: 7.93, Percentile: 59.82%
Model: 2.5357901502897864, Actual: 2.0, Percentile: 126.79%
Model: 2.9998699519736935, Actual: 3.0, Percentile: 100.0%
```

6.2 Classification

Since the dataset presents different categories of data, I decided to first try to use various methods of classification (supervised learning), and clustering (unsupervised learning) to learn the different results that I could get from either one.

6.2.1 Random Forest Classifier

Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.[4]

With the classifier, we get a better overall percentile of 35%

6.2.2 Random Forest Regressor

Whereas, Random Forest Regressors predict some value, which could be almost anything.

Different metrics are used for classification and regression.

Usually it not advised to use both for the same topic since they input differ so much from each other. As I can see from my results, I got a 10% accuracy compared to 35%.

6.2.3 Naïve Bayes

Method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features.[5]

With Naïve Bayes, I also get a higher percentile 34%, similar to Random Forest Classifier.

[3] https://en.wikipedia.org/wiki/Linear_regression

[4] https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1

[5] https://en.wikipedia.org/wiki/Naive_Bayes_classifier

6.3 Clustering

It is basically a type of unsupervised learning method . An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.[6]

6.3.1 KMeans

The K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids

It halts creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.[7]

K-Means is my dataset is not best model since I'm getting a negative percentile.

[6] https://www.geeksforgeeks.org/clustering-in-machine-learning/

[7] https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

7.0 Conclusion

This analysis started on the shopping patterns of new customers or Cold Starters, and indirectly diverged into the analysis of the pricing of different products through Machine Learning and testing different models and how they perform depending on the data. The first notebook presents the analysis comparing how these shoppers purchase their products, and the Machine Learning notebook predicts the price of the products using different models (with low accuracy percentile). The dataset after it was separated into 2 distinct groups, the normal dataset and the dataset that showed all the customer in the dataset, but only their 1st purchase. Through this, I was able to have a better idea on what stores, products and patterns were given by these customers.

This analysis is still in progress and will evolve as I gather more data, but for now the conclusions that can be taken from this are:

- Depending on income, customers are more susceptible to use coupons.
- Age is also a factor on how often customer use coupons.
- Location of customers has an effect on how this customer use coupons (this was based on store id and which customers shopped there).
- Relationship of how price overall on the coupon usage.
- Some stores are busier than others.
- Some products tend to be more popular for first time customers, but then they are not anymore.
- Some products are bought more often after the 1st visit.
- Customers do not shop in the same store, which can change the definition of a cold starter (a cold starter in this analysis is the first purchase recorded in the record; whereas for some store it could the 1st purchase made in their store).

After analyzing the dataset through the different graphs and correlations, I noticed that most customer do not tend to behave the same way after their first purchase. In fact, this was the case with the most popular product that was purchased by cold starters, but when comparing this specific product in the overall dataset, it lagged behind; which was one of the main challenges on creating an effective model that would be able to predict those behaviors.

The dataset presents lots of variables and many different interesting points that can be analyzed for days. These are a couple of things to still analyze in the future for both types of customers:

- Does location of the products in the store have an effect on the customer?
- Does shopping in another store change the behavior of the customer? If we treat the same customer as a cold starter when they shop in another store, will it be biased to apply that information to a model?
- Does specific coupons apply for certain customers based on their age/income/marital status, etc?
- Do less well known products benefit from the popularity of other products if they are placed close to them and use that indirect advertisement?
- Do coupons in general create a cold starter shopper or do they motivate them to come back to that specific store?