DSBA 6156 - Applied Machine Learning

Customer Segmentation and Customers' Buying Behavior Predictive Analysis

Submitted to: Dr. Minwoo Lee

Date: 05. 04. 2022

Github: Link

Authors: Connor Brown, Matt Campbell, Sean Oberer, Lenka Raslova,

Syed Muhammad Suffwan

Introduction

Problem Statement

The disruption of the COVID19 pandemic has caused drastic changes to consumers' buying patterns in the B2C retail marketplace. With an exponential increase in e-commerce, the significance of accurate customer profiling and effective targeted marketing has become highly important for businesses. For this purpose, we intended to use K-Means clustering in order to identify the correct segments for the business.

Secondly, targeted marketing is the key to successful business growth with the emerging e-commerce trend. Successful targeting can cause higher responsiveness from customers. For this reason, we intended to use classification models for predicting whether the customer will respond to the campaign or not.

Motivation

With the retail industry being so saturated, the retail industry is extremely competitive and as a result - it's very difficult for any particular company to stand out enough to dominate their particular market. Marketing effectively and efficiently is one of the biggest challenges many businesses face and especially one that companies in the retail industry face. For decades, the retail industry has predominantly relied on descriptive analytics in order to derive insights to try and market in the most efficient way possible. Additionally, many companies have terabytes of data that they don't know what to do with despite there being so much valuable information in that data. By using

machine learning, we can utilize not just descriptive analytics but also predictive analytics. Predictive analytics can help us forecast what our future earnings will be in the retail industry based on the demographics of the customers we are targeting. We can also use predictive analytics to forecast how much revenue we can generate based on what retailers are supplying and stocking up on in their stores. Through machine learning, we can uncover trends and insights that otherwise would not be able to be uncovered through descriptive analytics alone. With predictive analytics, we can figure out who specifically we need to target, why we need to target those individuals, and what products each particular customer is likely to buy based on their demographics, location, and past behavior.

Summary

For classification, our goal was to predict which customers would respond to a marketing campaign. In order to make these predictions we used six models with various implementations of each model. Those models are logistic regression, ridge regression, gaussian Naive Bayes, support vector machine, decision tree, and random forest. We found that the random forest model performed the best.

For clustering, we wanted to come up with multiple groups of customers that have similar demographic characteristics among each other. By doing this, we can find out what particular customers are likely to buy specific products. Additionally, we can also find which particular customers tend to spend more money than others so that we can target them and generate as much revenue as possible. Furthermore, we can also use clustering to find which particular customers are more likely to respond to offers than

others based on a variety of factors like their age, income, how many kids they have, how much they spend on particular products, etc..

Related Work

The methods we have chosen are similar to many of the methods used in the field of data science to find meaning in similar kinds of data to ours.

“Clustering technique is a critically important step in the data mining process. It is a multivariate procedure quite suitable for segmentation applications in the market forecasting and planning research” (Kashwan, Velu 2013). This is a paper detailing a process of customer segmentation using clustering, and “Results were quite encouraging and showed high accuracy.” (Kashwan, Velu 2013). The next paper we reviewed detailed different methods to pick the best number of clusters. “We focus on six different approaches : i) By rule of thumb; ii) Elbow method; iii) Information Criterion Approach; iv) An Information Theoretic Approach; v) Choosing k Using the Silhouette and vi) Cross-validation.” (Kodonariya, Makwana 2013)

One of the other papers we looked at also detailed customer segmentation using various data mining and machine learning techniques, this time, not just using k means. Specifically, they used: K-means, Naive Bayes and support vector machine classification. Their result showed “Naïve Bayesian Classification achieves the highest accuracy and specificity. However, the worst classification was performed by Support Vector Machine technique” (Das 2015). The last paper we reviewed used machine learning techniques to predict a customer’s spending score. They concluded “adaptive

spline regression technique outperforms the linear regression and linear spline regression methods in terms of the principle factor of interest, which is the RMSE." (Chakraborty, Sanyal, Sharma 2019)

Open questions in the domain

1. Which customer should be targeted for the marketing campaign?
2. What should be the content and structure of the offer to ensure that the customer opens it.
3. How many distinctive groups/clusters are present in the customer base of a retail business
    - What is the extent of similarity in the defining features of cases in the same cluster?
    - What are the distinguishing features between cases of different clusters?

Backgrounds

The first paper we reviewed: *Customer Segmentation Using Clustering and Data Mining Techniques* by Kishana R. Kashwan detailed a process of segmenting customers into groups using k-means clustering. The data "consisted of usages of brands under different conditions, demographic variables and varying attitudes of the customers" (Kashwan, Velu 2013). We used this paper to gain a better understanding of the ways that k means clustering is used in the field to segment customers. The paper goes on to explain the details of how k-means clustering works, and how each cluster moved

during the running of the k-means algorithm. The paper concludes with a breakdown of each of the 4 clusters, detailing the kind of people that made up each cluster. (Kashwan, Velu 2013)

The second paper we reviewed: *Review on determining number of Cluster in K-Means Clustering* by Trupti M. Kodinariya and Dr. Prashat R. Makwana furthered our understanding of the k-means algorithm. This paper details different methods for choosing the ideal number of clusters for k-means clustering. There were five discussed in detail: the elbow method, the information criterion approach, an information theoretic approach, choosing K using the silhouette, and cross validation (Kodinariya, Makwana, 2013). A detailed breakdown of each of these methods is outside the scope of this paper, but this paper is what reinforced our decision to use the elbow method in our own k-means clustering. The elbow method “is a visual method. The idea is that Start with K=2, and keep increasing it in each step by 1, calculating your clusters and the cost that comes with the training. At some value for K the cost drops dramatically, and after that it reaches a plateau when you increase it further. This is the K value you want.” (Kodinariya, Makwana, 2013)

The next paper we reviewed: *A customer classification prediction model based on machine learning techniques* by T. K. Das detailed the process and success of customer classification using k-means, Naive Bayes, and support vector machine classification. It explained a five step process: data collection, data pre-processing, data mining, data analysis and visualization, and lastly, report generation. The data pre-processing step and the data mining were the two of great use to us. In the data ming step “ the data has been cleaned by removing blank space, replacing missing

value, filling the data gaps and removing unused values" (Das 2015). Next the paper went over attribute selection. This step was automated by the tool being used for the pre-processing and, as such, was of little use to us (Das 2015). Next, the paper detailed the three different classification methods and different evaluation metrics. This chapter was of great use to us and helped us in running our own classification. The paper concluded that "For selecting the right customers they can use the Naïve Bayesian Classification algorithm which correctly fits the data set when compared with any other classification algorithms. This technique would help the marketing department to identify the respondents so that they would be basically targeted for specific campaigning activity." (Das 2015)

The final paper we reviewed: *Machine Learning based Prediction of Customer Spending Score* by Aratrika Chakraborty, Judhajit Sanyal and Prashant Sharma, explained "the use of linear regression to estimate customer spending score at a mall using metrics such as age and annual income". The paper begins with a brief literature review, followed by an explanation of their models. They used the age and annual income of 200 customers to predict each customer's spending score on a scale of 1-100. They broke their basic linear models into spline regression models by breaking their 200 customers into 10 clusters of 20 customers each, allowing them to create piecewise linear spline regression models. (Chakraborty, Sanyal, Sharma 2019). The paper concluded that "The adaptive technique registers the lowest RMSE value among all the techniques studied here since it bases its estimate in a manner by which marginal outliers and points at a greater than average distance from the regression lines for each data cluster add to the estimate computed by the model" and that "clustering

increases the accuracy of estimate, as evident from the decrease in the RMSE values for linear spline regression compared to the RMSE value of estimates calculated using the simple linear regression model” (Chakraborty, Sanyal, Sharma 2019)

Data Description

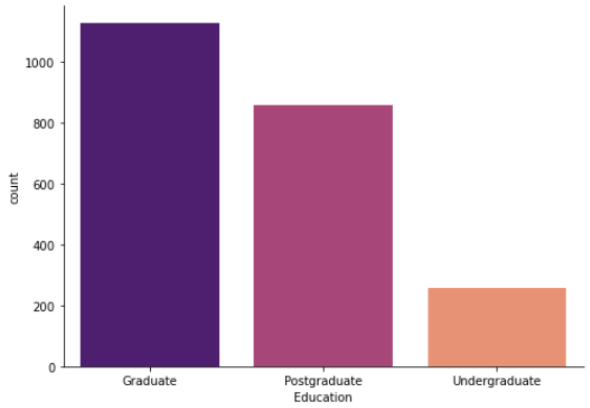
Our dataset was derived from a retail company that contained customer data including customers demographics, what products the customers purchased, and if they responded to discounts and other offers.

In total, there were 29 columns that were contained in this data set which can be seen in the image below.

ID	int64
Year_Birth	int64
Education	object
Marital_Status	object
Income	float64
Kidhome	int64
Teenhome	int64
Dt_Customer	datetime64[ns]
Recency	int64
MntWines	int64
MntFruits	int64
MntMeatProducts	int64
MntFishProducts	int64
MntSweetProducts	int64
MntGoldProds	int64
NumDealsPurchases	int64
NumWebPurchases	int64
NumCatalogPurchases	int64
NumStorePurchases	int64
NumWebVisitsMonth	int64
AcceptedCmp3	int64
AcceptedCmp4	int64
AcceptedCmp5	int64
AcceptedCmp1	int64
AcceptedCmp2	int64
Complain	int64
Z_CostContact	int64
Z_Revenue	int64
Response	int64

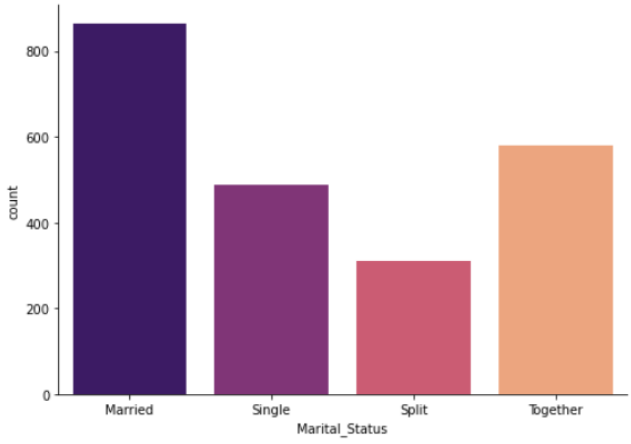
Exploratory Data Analysis

Education



Most of the customers achieved graduate and postgraduate education. Only a few hundred customers' highest achieved education level is undergraduate.

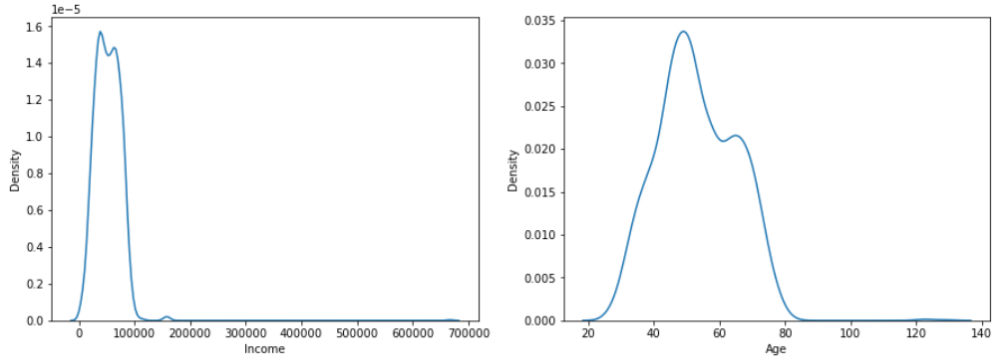
Marital Status



We can split customers into four categories based on their marital status. The most numerous category is Married status. The second most numerous category is Together status. Based on that, we can conclude that more than half of the customers are in a

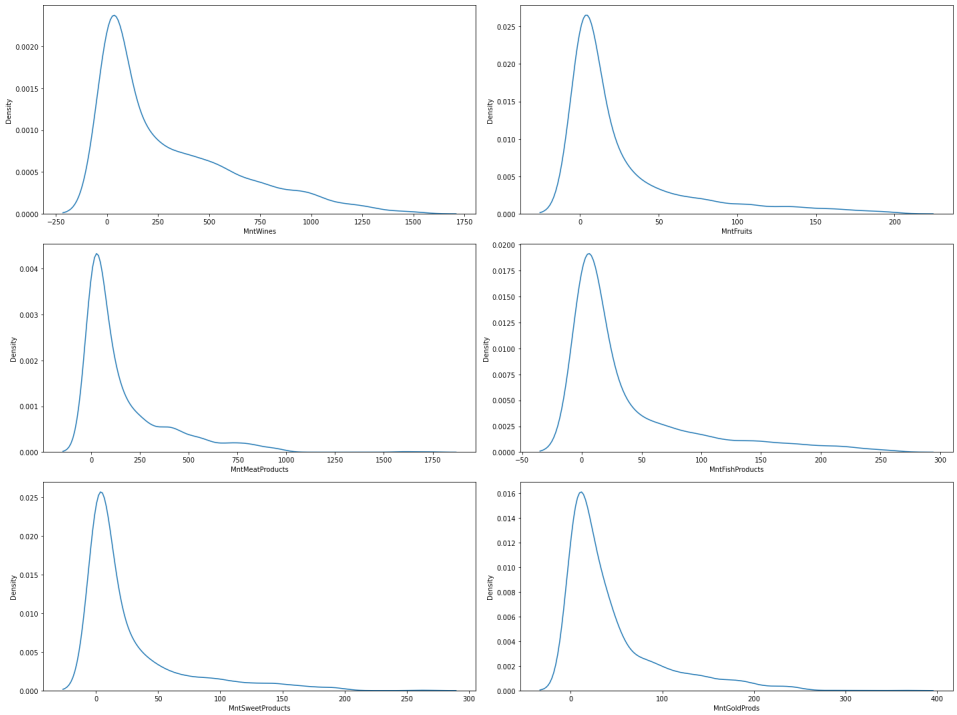
relationship. The third most numerous category is Single status and the last category is Split status.

Distribution of Age & Income



Income and Age distribution is right-skewed. Income has few outliers. Based on the age distribution, we can say that most of the customers turn their 40s (our customers are middle age and older people).

Distributions of Products



All six distributions of products are right-skewed and consist of outliers.

Relationships with Response Variable

The plot above is showing relationships between variable Response (overall response of customers to the campaign).

Descriptive Analytics

The goal of our descriptive analytics and exploratory analysis was to find relationships between our customer demographic variables and their product browsing habits, vice versa, and between customer demographics and their preferred avenues of purchase. We used a collection of regression techniques (linear regression, ridge regression, lasso regression and elastic net) to create models to predict each of the aforementioned variables. The initial approach saw minimal data preprocessing, and many more regression models. Of the 154 different regressions run, less than ten percent of the models broke 50% accuracy. After some data preprocessing, the “teen” and “kids” fields were combined into one “children” field, and the relationship status field was changed from 8 different possible relationship status to a categorical “marriage status” column. This approach saw substantially more success and we were able to find some useful information.

Wine Product Interest:

We found that Income, Age, Marriage Status, and Education level were all positive indicators of customer interest in wine products, while number of children was a negative indicator. We found these findings to be relatively intuitive. More kids means more responsibility, which means less interest in Wine. The positive indicators fit the image of someone that would spend a lot of time browsing wine products: People with money & time, sophisticated by their education level, and someone to enjoy with.

Meat Product Interest:

Our findings with this product stumped us. We found that the only positive predictor of meat product interest was Income. Age, marriage status, education level and number of children were all negatively correlated with meat product interest. This could point to young single people with money being more interested in more decadent and exotic meats, but that is the only explanation we could find.

Personal Income Indicators:

We found that increased interest in any product indicated higher income. This is an important realization because it implies that people only browse the products they have the purchasing power to actually buy. This implies a lack of virtual window shopping. This means that in a situation where there is not much personal data on people, just marketing sales and deals based on time they spend browsing their products is a semi-reliable strategy.

Deals Interest Indicators:

We found that the demographic variables that positively affect a customer's interest in deals were personal income, time spent online browsing, age, and marriage. Variables that had a negative impact were education level, number of children, and individual website visits. This speaks to an older, more frugal population with time on their hands to spend looking for deals, retirees and empty nesters, for example.

In-Store Purchase Frequency Indicators

We discovered that the positive indicators of In-Store purchasing were personal income, time spent online browsing, age, and marriage. The negative indicators were education level, number of children, and individual website visits. This is a notably similar pattern to deals interest. We think these indicators highlight a demographic that enjoys shopping, and has the free time and schedule flexibility to be able to do so, and often.

Methods

Clustering

We used k-means clustering in order to find distinct customer segments using the available data. We tried our unsupervised model (KMeans) with combinations of several feature transformations and extraction techniques. We also implemented hyper parameter optimization techniques based on literature review; Elbow Method to find optimal value of ‘k’ for clustering.

Data Preprocessing

Missing values: There were only 24 missing ‘Income’ values in the dataset.

```
display(df.isna().sum())
```

```
ID                      0
Year_Birth              0
Education               0
Marital_Status          0
Income                 24
Kidhome                 0
Teenhome                0
Dt_Customer             0
Recency                 0
MntWines                0
MntFruits               0
MntMeatProducts         0
MntFishProducts         0
MntSweetProducts        0
MntGoldProds            0
NumDealsPurchases       0
NumWebPurchases         0
NumCatalogPurchases     0
NumStorePurchases       0
NumWebVisitsMonth       0
AcceptedCmp3            0
AcceptedCmp4            0
AcceptedCmp5            0
AcceptedCmp1            0
AcceptedCmp2            0
Complain                0
Z_CostContact           0
Z_Revenue               0
Response                0
dtype: int64
```

We used KNNImputer from sklearn for imputation the 24 missing values for income.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	1.0	58138.0	635.0	88.0	546.0	172.0	88.0	88.0	3.0	8.0	10.0	4.0	7.0	65.0	0.0	0.0	1617.0	25.0	0.0
1	1.0	46344.0	11.0	1.0	6.0	2.0	1.0	6.0	2.0	1.0	1.0	2.0	5.0	68.0	0.0	2.0	27.0	6.0	0.0
2	1.0	71613.0	426.0	49.0	127.0	111.0	21.0	42.0	1.0	8.0	2.0	10.0	4.0	57.0	1.0	0.0	776.0	21.0	0.0
3	1.0	26646.0	11.0	4.0	20.0	10.0	3.0	5.0	2.0	2.0	0.0	4.0	6.0	38.0	1.0	1.0	53.0	8.0	0.0
4	2.0	58293.0	173.0	43.0	118.0	46.0	27.0	15.0	5.0	5.0	3.0	6.0	5.0	41.0	1.0	1.0	422.0	19.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2235	1.0	61223.0	709.0	43.0	182.0	42.0	118.0	247.0	2.0	9.0	3.0	4.0	5.0	55.0	1.0	1.0	1341.0	18.0	0.0
2236	2.0	64014.0	406.0	0.0	30.0	0.0	0.0	8.0	7.0	8.0	2.0	5.0	7.0	76.0	1.0	3.0	444.0	22.0	1.0
2237	1.0	56981.0	908.0	48.0	217.0	32.0	12.0	24.0	1.0	2.0	3.0	13.0	6.0	41.0	0.0	0.0	1241.0	19.0	1.0
2238	2.0	69245.0	428.0	30.0	214.0	80.0	30.0	61.0	2.0	6.0	5.0	10.0	3.0	66.0	1.0	1.0	843.0	23.0	0.0
2239	2.0	52869.0	84.0	3.0	61.0	2.0	1.0	21.0	3.0	3.0	1.0	4.0	7.0	68.0	1.0	2.0	172.0	11.0	0.0

2240 rows × 19 columns

KNN Imputed rows

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
10	1.0	24411.4	5.0	5.0	6.0	0.0	2.0	1.0	1.0	1.0	0.0	2.0	7.0	39.0	1.0	1.0	19.0	4.0	0.0
27	1.0	34768.0	5.0	1.0	3.0	3.0	263.0	362.0	0.0	27.0	0.0	0.0	1.0	36.0	0.0	1.0	637.0	27.0	0.0
43	2.0	44179.0	81.0	11.0	50.0	3.0	2.0	39.0	1.0	1.0	3.0	4.0	2.0	63.0	0.0	0.0	186.0	9.0	0.0
48	1.0	45001.2	48.0	5.0	48.0	6.0	10.0	7.0	3.0	2.0	1.0	4.0	6.0	71.0	0.0	3.0	124.0	10.0	0.0
58	1.0	27469.0	11.0	3.0	22.0	2.0	2.0	6.0	2.0	2.0	0.0	3.0	6.0	40.0	0.0	1.0	46.0	7.0	0.0
71	0.0	33296.6	25.0	3.0	43.0	17.0	4.0	17.0	3.0	3.0	0.0	3.0	8.0	49.0	1.0	1.0	109.0	9.0	0.0
90	2.0	60831.2	230.0	42.0	192.0	49.0	37.0	53.0	12.0	7.0	2.0	8.0	9.0	65.0	1.0	3.0	603.0	29.0	0.0

Feature Extraction

Created new features like; ‘Age’ feature using ‘Year_Birth’, ‘time_spent’ from all the minute spent information, ‘total_purchase’ from all the purchase information, total_campaigns from all the responses for 6 campaigns, ‘Children’ from adding the kid and teen at home values, ‘Response’ feature which would 1 if customer has accepted even one campaign.

Feature Engineering

Reducing the number of classes to reduce the skewness and then encoded categorical features like; ‘Education’ and ‘Marital Status’.

```
## Reducing Education to 3 values only

df["Education"]=df["Education"].replace({"Basic":"Undergraduate","2n Cycle":"Undergraduate",
                                         "Graduation":"Graduate", "Master":"Postgraduate", "PhD":"Postgraduate"})

## Creating new column Married by Reducing Marital Status to married or unmarried creating a new column.
df["Married"] = df["Marital_Status"].apply(lambda x: 1 if (x == "Married" or x == "Together") else 0)
```

```
#encoding the education

df['Education'] = df["Education"].replace({"Undergraduate":0, "Graduate":1, "Postgraduate":2})
```

```
df['Education'].value_counts()
```

```
1    1127
2     856
0     257
Name: Education, dtype: int64
```

Final features for clustering:

	0	1	2	3	4	5
Education	1.0	1.0	1.0	1.0	2.0	2.0
Income	58138.0	46344.0	71613.0	26646.0	58293.0	62513.0
MntWines	635.0	11.0	426.0	11.0	173.0	520.0
MntFruits	88.0	1.0	49.0	4.0	43.0	42.0
MntMeatProducts	546.0	6.0	127.0	20.0	118.0	98.0
MntFishProducts	172.0	2.0	111.0	10.0	46.0	0.0
MntSweetProducts	88.0	1.0	21.0	3.0	27.0	42.0
MntGoldProds	88.0	6.0	42.0	5.0	15.0	14.0
NumDealsPurchases	3.0	2.0	1.0	2.0	5.0	2.0
NumWebPurchases	8.0	1.0	8.0	2.0	5.0	6.0
NumCatalogPurchases	10.0	1.0	2.0	0.0	3.0	4.0
NumStorePurchases	4.0	2.0	10.0	4.0	6.0	10.0
NumWebVisitsMonth	7.0	5.0	4.0	6.0	5.0	6.0
Age	65.0	68.0	57.0	38.0	41.0	55.0
Married	0.0	0.0	1.0	1.0	1.0	1.0
Children	0.0	2.0	0.0	1.0	1.0	1.0
time_spent	1617.0	27.0	776.0	53.0	422.0	716.0
total_purchase	25.0	6.0	21.0	8.0	19.0	22.0
total_campaigns	0.0	0.0	0.0	0.0	0.0	0.0

Clustering Models & Techniques:

Baseline Model (k_5_features_18): Used KMean from sklearn with k = 5 and tried to fit the whole dataset (18 columns).

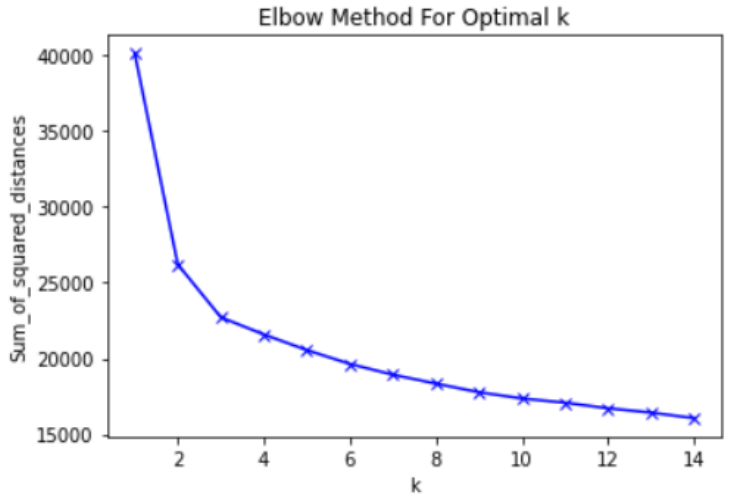
```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=5, random_state=1).fit(X)
kmeans.labels_
```

```
array([3, 3, 1, ..., 3, 1, 3])
```

KMeans with Standard Scalar (k5_features18_StdScaled): Similar to baseline but this time we just transformed the features using standard scaling technique.

We knew that we must opt for statistical or systematic methods to determine two key factors; optimal K and feature selections based on their power for explaining the data.

Optimal K: We used elbow method in order to find the optimal K and concluded k = 4 which was giving us a significantly low sum of squared distances in comparison to the number of clusters.



Feature Extraction:

KMeans with PCA (k4_pca18): We tried a range of components and evaluated the performance of clustering using Silhouette scores and the highest score was lesser than

our baseline model using 18 components which means that PCA is not effective here.

```
scaler = StandardScaler()
pca = PCA(n_components = 18)
pipe = Pipeline(steps=[("scaler", scaler), ("pca", pca)])
X_pca = pipe.fit_transform(X)
kmeans = KMeans(n_clusters=4, random_state=0)
kmeans.fit(X_pca)
```

KMeans with TSNE (k4_tsne3): We tried TSNE with 3 components (maximum possible) with our KMeans algorithm and we got slightly close to our base model but even then it was lower.

```
mms = StandardScaler()
mms.fit(X_)
X_transformed = mms.transform(X_)
X_tsne = TSNE(n_components=3).fit_transform(X_transformed)
kmeans = KMeans(n_clusters=4, random_state=0)
```

Final Model (k4_features7)

Based on the literature review and reading a few articles, we figured out that while using real world data which is skewed in its own ways, the book methods are not always supposed to work.

We ran K Means with k = 4 and just 7 features. Features include: Education, Income, Age, Married, Children, time_spend and total_purchase.

Classification

We used various classification models in order to determine the best model for predicting whether customers will respond to the campaign or not. For this aim, we used

the following models: Ridge Classifier, Gaussian Naive Bayes, Support Vector Machines, Logistic Regression, Decision Tree, and Random Forest.
Our original dataset was highly imbalanced. Therefore, we first ran all models with this imbalance data. Then we ran all models again with a balanced dataset.

Data Preprocessing

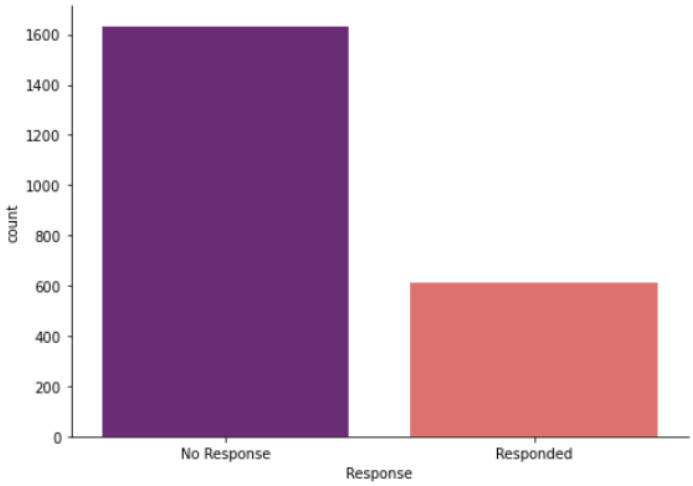
We used the same techniques for Missing Values, and Feature Extraction as in the clustering part. For Feature Reducing, we also reduced the number of classes in categorical features such as ‘Education’ and ‘Marital Status’. The technique for ‘Education’ is the same as in the clustering part. For ‘Marital Status’ we used the code below.

```
df["Marital_Status"] = df["Marital_Status"].replace({"Absurd":"Single", "Alone":"Single", "Divorced":"Split",
                        "Married":"Married", "Single":"Single", "Together'":"Together",
                                             "Widow":"Split", "YOLO":"Single" })

labels = df['Marital_Status'].astype('category').cat.categories.tolist()
print(labels)
```

```
['Married', 'Single', 'Split', 'Together']
```

The original dataset consisted of separate responses of customers to individual campaigns. Thus, we created a new target variable “Response”, this variable indicates whether customers responded to any campaign or not (“overall response”).

Unfortunately, the “Response” variable is highly imbalanced.



In addition to the preprocessing steps outlined above, we wanted to try some additional techniques to see how performance was impacted. The preprocessing to be described below relates to the results you see labeled “Preprocessed Chi2 - 18 var.” in our results table. To start, a column named “Days_as_Customer” was created to determine how long an individual has been a customer at the store. This was done using the column “Dt_Customer” which was a date column that recorded the date a customer joined the store. To calculate how long a customer has been with the store the “Dt_Customer” column was subtracted from the current date to give us the number of days each individual has been a customer.

```
# define the current date used to calculate how long user has been a customer
from datetime import date
currDate = np.datetime64("2022-04-26")
```

```
# print current date
currDate
```

```
numpy.datetime64('2022-04-26')
```

```
# subtract Dt_Customer from currDate to get time delta
delta = currDate - df["Dt_Customer"]
```

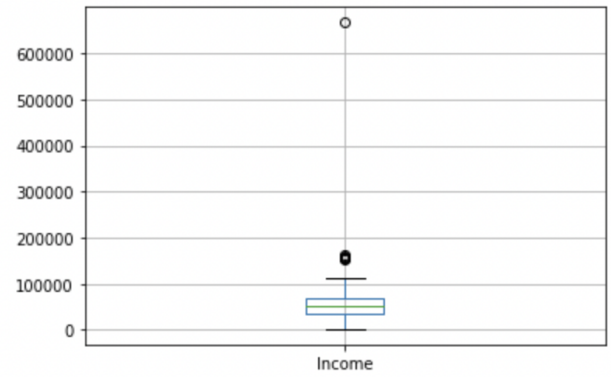
```
# check data
delta
```

```
0       3521 days
1       2971 days
2       3170 days
3       2997 days
4       3019 days
           ...
2235    3239 days
2236    2877 days
2237    3013 days
2238    3014 days
2239    3480 days
Name: Dt_Customer, Length: 2240, dtype: timedelta64[ns]
```

This data was then converted to an integer and stored in a new column named “Days_as_Customer”. The next area we wanted to take a look at was the distribution of data in the “Income” column. Below is a boxplot used to visualize outliers in the data.

```
# plot boxplot to visualize potential outliers
df.boxplot(column="Income")
```

```
<AxesSubplot:>
```



As you can see there are outliers in our data with one being a significant outlier. In order to handle these outliers, the IQR of income was calculated and the datawas filtered based upon the IQR.

```
q1=df["Income"].quantile(0.25)
q3=df["Income"].quantile(0.75)
iqr=q3-q1
```

```
# get locations of outliers in Income
np.where(df["Income"] > (q3 + 1.5*iqr))
```

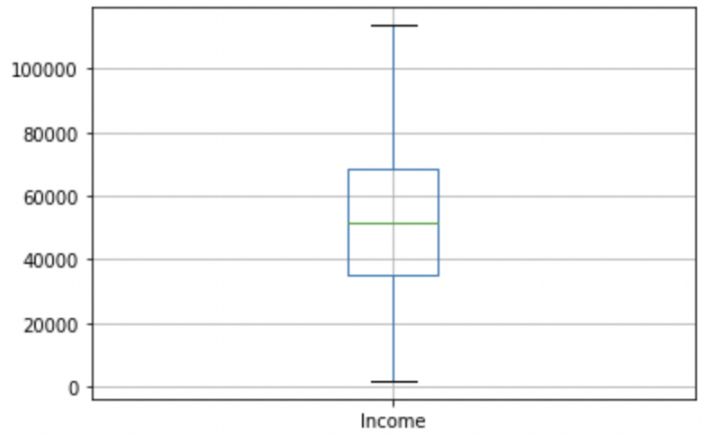
```
(array([ 164,  617,  655,  687, 1300, 1653, 2132, 2233]),)
```

```
# drop income outlier records
test_df = df.drop([164,  617,  655,  687, 1300, 1653, 2132, 2233])
```

In this instance, we chose to remove the records for simplicity. The boxplot below shows the new data.

```
# view income boxplot of new test_df
test_df.boxplot(column="Income")
```

```
<AxesSubplot:>
```

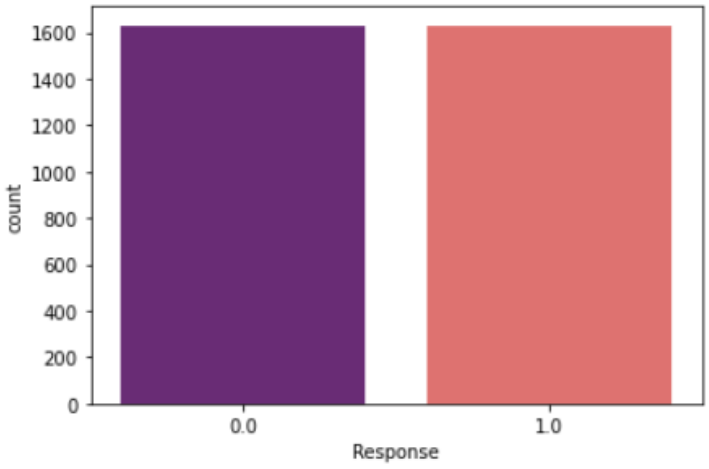


Finally, a Chi2 test was performed on this data which concluded that “Marital_Status” was not a significant variable. It had a P-value of 0.929 which is greater than 0.05, so this column was dropped. This is now the data that was used for the results found labeled “Preprocessed Chi2 - 18 var”.

Balancing Data

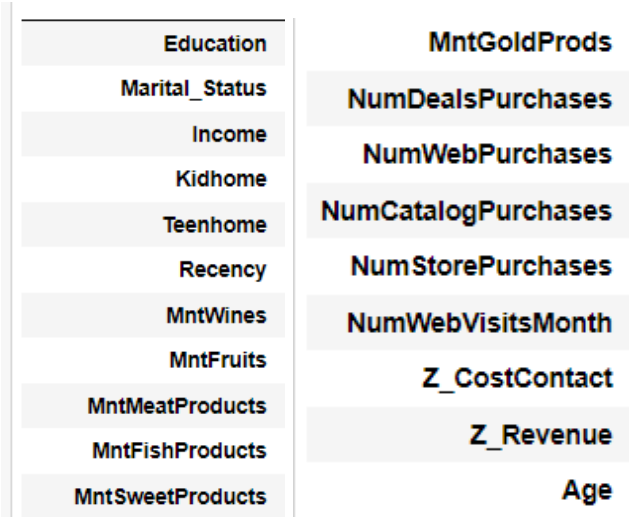
For balancing the data, we used SMOTE method (Synthetic Minority Oversampling Technique). The SMOTE method is suitable for balancing classification datasets. The SMOTE algorithm chooses samples that are close in the feature space, models a line between the samples in the feature space, and creates a new sample based on the line (Machine Learning Mastery, 2021).

We used the library “imbalanced-learn” for balancing data. This library provides SMOTE method. This method successfully balanced our data as can be seen below.



Feature Selection

We selected the final 20 features as input variables for classification models.



Education
Marital_Status
Income
Kidhome
Teenhome
Recency
MntWines
MntFruits
MntMeatProducts
MntFishProducts
MntSweetProducts
MntGoldProds
NumDealsPurchases
NumWebPurchases
NumCatalogPurchases
NumStorePurchases
NumWebVisitsMonth
Z_CostContact
Z_Revenue
Age

We had a high number of input variables. Therefore, we decided to reduce them via feature selection. We used three following approaches for the feature selection:

1. Variance Threshold (Cut of value = 1.00),
2. Sequential Feature Selection (10 variables),
3. Chi2 Test.

Experiments

Clustering:

Method and Evaluations

Clustering is an unsupervised learning method and it highly depends upon the quality of data and domain knowledge to finalize the features. As discussed above; we tried several models which consisted of different combinations of machine learning techniques. In order to evaluate our models; we used silhouette scores which is a metric used to calculate the goodness of a clustering technique.

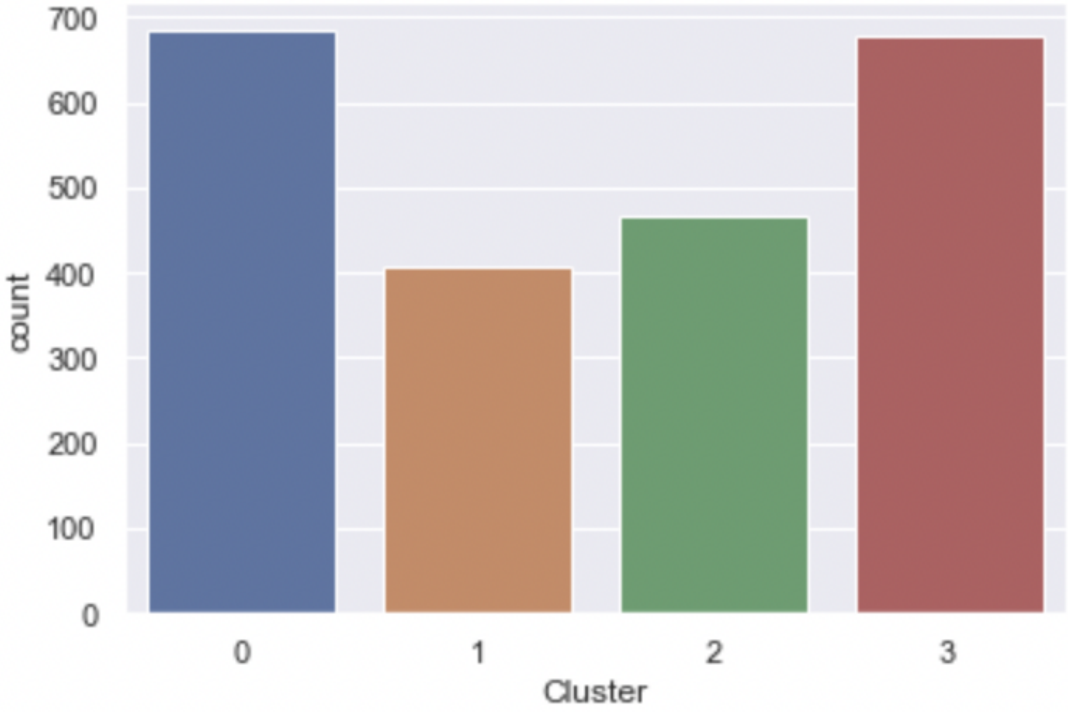
The reason for not using other metrics like homogeneity score was that we did not have any target variable for this problem.

	0	1	2	3	4
Model	k_5_features_18	k5_features18_StdScaled	k4_pca18	k4_tsne3	k4_features7
Silhoutte_Score	0.45	0.03	0.1	0.4	0.6

The model k4_features7 gives the highest score with the lowest computational cost as it only used 7 features. The reason we could not get a higher score was due to data quality constraints and skewness because it is real retail store data.

Clustering Results

Below you can find the distribution of our clusters created. As you can see from the visual below, clusters 0 and 3 were the largest whereas clusters 1 and 2 were slightly smaller. Overall, the distribution of the clusters was relatively balanced.



Below you can find our main takeaways from each cluster:

Cluster 0:

- Second youngest group of customers (53 years old)

- Second lowest annual income among all clusters

- Second lowest total amount spent

- Largest number of customers among all clusters (684)

- Has the most kids on average

- Second highest response rate to offers

Cluster 1:

- Second oldest group of customers (roughly 54 years old)

- Highest annual income among all clusters

- Highest total amount spent

- Smallest number of customers among all clusters (407)

- Second lowest number of kids on average

- Highest response rate to offers

Cluster 2:

- Youngest group of customers (average age of 47 years old)

- Lowest annual income among clusters

- Lowest total amount spent

- Second smallest number of customers among all clusters (466)

- Lowest number of kids on average

- Lowest response rate to offers

Cluster 3:

- Oldest group of customers (average age of 56 years old)

- Second highest average annual income

- Second highest total amount spent

- Second largest number of customers among all clusters (679)

- Second highest number of kids on average

- Second lowest response rate to offers

Classification

Classification models belong under supervised machine learning. They create models that classify outputs into categories. The goal of our classification models is to determine whether a specific customer will respond to the campaign or not.
We have created several models which consist of different input variables. Moreover, we trained these models on imbalanced data at first and then we trained them on balanced data.
We split data to train and test set (test set consists of 20 % of data). Our target variable is “Response”.

For the evaluation of classification models, we used Accuracy (balanced one for imbalance data and ordinary one for balanced data), Precision, and Recall. All these measurements were calculated from test data. We ran each model several times based upon different selections of input variables.

Model Results - Imbalanced Data:

Model	Variable Selection	Balanced Accuracy	Precision	Recall
Ridge Classifier	All - 20 var.	0.654	0.638	0.383
	Variance T. - 15 var.	0.625	0.585	0.33
	Seg. Feat. Selection - 10 var.	0.642	0.609	0.365
	Preprocessed Chi2 - 18 var.	0.647	0.574	0.379
	Without Preprocessing - 18 var.	0.665	0.719	0.387
Gaussian Naive Bayes	All - 20 var.	0.660	0.453	0.548
	Variance T. - 15 var.	0.661	0.457	0.548
	Seg. Feat. Selection - 10 var.	0.681	0.676	0.435
	Preprocessed Chi2 - 18 var.	0.631	0.378	0.524
	Without Preprocessing - 18 var.	0.674	0.493	0.555
Support Vector Machines	All - 20 var.	0.657	0.683	0.374
	Variance T. - 15 var.	0.642	0.656	0.348
	Seg. Feat. Selection - 10 var.	0.632	0.692	0.313
	Preprocessed Chi2 - 18 var.	0.644	0.607	0.359
	Without Preprocessing - 18 var.	0.696	0.776	0.437
Logistic Regression	All - 20 var.	0.690	0.662	0.461
	Variance T. - 15 var.	0.651	0.620	0.383
	Seg. Feat. Selection - 10 var.	0.641	0.621	0.357
	Preprocessed Chi2 - 18 var.	0.655	0.577	0.398
	Without Preprocessing - 18 var.	0.669	0.712	0.395
Decision Tree (Pruned)	All - 20 var.	0.652	0.573	0.409
	Seg. Feat. Selection - 10 var.	0.641	0.672	0.339
Random Forest	All - 20 var.	0.673	0.667	0.417
	Seg. Feat. Selection - 10 var	0.674	0.676	0.417

Model Results - Balanced Data:

Model	Variable Selection	Accuracy	Precision	Recall
Ridge Classifier	All - 20 var.	0.744	0.754	0.689
	Variance T. - 15 var.	0.729	0.747	0.654
	Seg. Feat. Selection - 10 var.	0.718	0.742	0.628
	Preprocessed Chi2 - 18 var.	0.766	0.845	0.673
	Without Preprocessing - 18 var.	0.749	0.850	0.654
Gaussian Naive Bayes	All - 20 var.	0.662	0.653	0.622
	Variance T. - 15 var.	0.658	0.653	0.609
	Seg. Feat. Selection - 10 var.	0.698	0.712	0.619
	Preprocessed Chi2 - 18 var.	0.662	0.715	0.580
	Without Preprocessing - 18 var.	0.698	0.758	0.654
Support Vector Machines	All - 20 var.	0.813	0.793	0.824
	Variance T. - 15 var.	0.799	0.782	0.804
	Seg. Feat. Selection - 10 var.	0.776	0.752	0.795
	Preprocessed Chi2 - 18 var.	0.821	0.856	0.787
	Without Preprocessing - 18 var.	0.822	0.855	0.811
Logistic Regression	All - 20 var.	0.746	0.748	0.705
	Variance T. - 15 var.	0.737	0.748	0.676
	Seg. Feat. Selection - 10 var.	0.720	0.747	0.625
	Preprocessed Chi2 - 18 var.	0.765	0.827	0.691
	Without Preprocessing - 18 var.	0.757	0.845	0.676
Decision Tree (Pruned)	All - 20 var.	0.746	0.772	0.663
	Seg. Feat. Selection - 10 var.	0.772	0.794	0.705
Random Forest	All - 20 var.	0.896	0.891	0.891
	Seg. Feat. Selection - 10 var	0.888	0.877	0.891

Best Model

Our best model is the Random Forest model with all 20 variables after using SMOTE to balance the data. This model has the highest accuracy, precision, and recall. The second best model is the Random Forest using sequential feature selection with ten variables. The Support Vector Machines model without preprocessing containing 18 variables is the third best model we created.

Conclusions

Clustering

We also learned that for customer segmentation using unsupervised learning; domain knowledge and base variables are the most important factors. We only discussed this in class about the situations where professionals have to decide between the evaluation metrics scores and their domain experience. We experienced this during this project where the features that worked for us came out of domain experience and literature reviews instead of statistical techniques. We also had to do quite a bit of data preprocessing before we ran our clustering. First off, there were multiple variables for the total amount spent split up between the type of product spent so we decided to combine all of these columns to create a “total spent” column showing the total amount a person has spent. There were also two variables for people that either had a child or teen at home. Rather than keeping these two split up, we combined them so that it was

just how many kids were at home in total. Additionally, we also had a few outliers where people stated they were 100 years or older. Furthermore, there were also some outliers with income where there were a few individuals that had over $300,000 in income per year whereas the vast majority of our data was well below that. We also had to change our categorical variables to be numerical like the Marriage status. Doing all of these things improved the accuracy of our clustering. We also tried a variety of scalers, dimensionality reduction methods, and numerous combinations of variables to use to see which clustering algorithm would be the most accurate. Ultimately - we found that creating a K means algorithm with k = 4 and 7 specific features would give us the best accuracy for our K means clustering. As a result of this, we found that reducing the overall number of variables and combining them can greatly improve our K means model.

In conclusion, we learned that trying different combinations of variables and different dimensionality reduction methods will yield the best possible results we can get when it comes to K-means clustering. Even after doing these things, we still believe that we can improve our model by reducing the data to be as simple as possible while still containing all of the valuable information that we need from it.

Classification

Throughout our analysis we learned a great deal about model performance and things we can do to improve performance. Our initial dataset was quite imbalanced with regard to our target variable. This imbalance negatively impacted the performance of our models. We saw a significant uptick in performance (increased accuracy, precision, and

recall) after implementing SMOTE to better balance our dataset. We do believe there are areas where we can improve upon in order to maximize the performance of our models even more. One area we think could use improving is the handling of outliers in the data. We did an outlier removal on the Income column for one analysis, but we could potentially benefit from replacing those values with a mean or median value from the data instead of outright removing the records. Also, there are other columns in the dataset that contain outliers that should also be handled. This is something we plan to explore in the future to determine how it impacts model performance. Also, another area that we can improve upon is creating a more accurate representation of our data. Specifically, the data types need to be improved. In our data, we have two categorical data columns, “Marital_Status” and “Education”. However, the data type for these two columns is float64, meaning that they are interpreted as numbers instead of representing a category. For example, a value of 1 in “Marital_Status” represents being married and a value of 2 represents being single. Since this column does not have a categorical type, our models will interpret being single as better because it has a higher number value. This needs to be corrected to get a better representation of our data and is something we will explore in the future.

In conclusion, we learned many techniques throughout this project and steadily improved our models with each step. Even with this improvement, we believe our models can get even better after cleaning up the areas discussed above.

Response to Feedback

Based on the feedback during our presentation, we changed the evaluation of our imbalanced models to balanced accuracy. Moreover, we balanced the data via SMOTE method and evaluated models again. Because of that, we achieved better results for our models.

Our Contributions

We adapted several techniques from the class lectures and literature review in this project and performed several experiments to decide which combination of techniques would generate optimal results.

For centroid-based clustering; the common practice in the literature was to use PCA for feature transformation but that did not work for us. Instead; based on our EDA, analysis of components and domain knowledge; we choose 7 best features which generated the best results. Moreover, we did not stop at transforming features by scaling or encoding; instead we created new features by joining similar variables to reduce the sparsity. Another different approach was that we performed clustering without any labels; even though clustering is unsupervised but at the end we do use evaluations metrics which takes some target into account. We did not use that as there was no well-balanced target in this dataset unlike the literature discussed above.

The literature review of classification showed us that we should use various models for classification problems because each problem is different and needs different

approaches. Based on the *A Customer Classification Prediction Model Based on Machine Learning Techniques* paper, we implemented Naive Bayes and Support Vector Machines models. Moreover, we extended this list of classification models by Logistic Regression, Ridge Classifier, Decision Tree, and Random Forest. Then we fitted these models via imbalanced and balanced data. We also used a few different approaches for selecting input variables. And finally, we compared these models and picked the best one. The literature review helped us with balancing data, we used the SMOTE method which was described in the article *SMOTE for Imbalanced Classification with Python.* Due to the SMOTE method, we were able to rapidly improve Accuracy, Precision and Recall scores.

Intent to Share

We are comfortable sharing this project with anyone.

References

SMOTE for Imbalanced Classification with Python (2021). *Machine Learning Mastery* [online]. Copyright © 2022 Machine Learning Mastery. All Rights Reserved. [cit. 05.01.2022]. Available from: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Patel, A. (2021, August 22). Customer Personality Analysis. Kaggle. Retrieved February 25, 2022, from https://www.kaggle.com/imakash3011/customer-personality-analysis

Kharwal, A. (2021, June 26). Customer personality analysis with python. Thecleverprogrammer. Retrieved February 25, 2022, from https://thecleverprogrammer.com/2021/02/08/customer-personality-analysis-with-python/

Kashwan, K. R., Velu, C. M., (2013, December 6). Customer Segmentation Using Clustering and Data Mining Techniques. ResearchGate. Retrieved February 25, 2022, from https://www.researchgate.net/profile/Kr-Kashwan/publication/271302240_Customer_Segmentation_Using_Clustering_and_Data_Mining_Techniques/links/57093e7908ae2eb9421e2d86/Customer-Segmentation-Using-Clustering-and-Data-Mining-Techniques.pdf

Syakur, M. A., Khotimah, B. K., Rochman, E. M. S., & Satoto, B. D. (2018, April 9). Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster. IOPScience. Retrieved February 25, 2022, from https://iopscience.iop.org/article/10.1088/1757-899X/336/1/012017/meta

Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. International Journal, 1(6), 90-95.

Das, T. K. (2016, April 21). A customer classification prediction model based on machine learning techniques. IEEE Xplore. Retrieved February 25, 2022, from

https://ieeexplore.ieee.org/abstract/document/7456903/authors#authors

Sharma, P., Chakraborty, A., & Sanyal, J. (2020, February 3). Machine learning based prediction of customer spending score. IEEE Xplore. Retrieved February 25, 2022, from https://ieeexplore.ieee.org/abstract/document/8978374/authors#authors