STUDENT NAME: Nilay Anand

ASSIGNMENT: Final Assignment – Segmentation

Customer Segmentation using K-means

Summary

The data provided for analysis contains information about customers. It has 5000 rows and 59 columns. The columns have information about spending habits, items owned, pets owned, demographic etc. (see Appendix Table 1)

After visualizing and analysing all the variables, continuous variable was encoded as categorical, and each category was encoded to numerical values.

The goal was to create customer segments from the data to sell them plans of a Telecommunications company. The initial idea was to select variables according to their spending ability and needs. The following variables were considered -

- Region – Which area the customer belongs
- Card Spend Month – Money spent by customer in a month
- Voice Last Month – Voice used over a month
- Homeowner – Customer owns a house or not
- Owns Mobile Device – Customer owns a mobile device or not
- Phone Co Tenure – Amount of time customer has been with the company
- Age
- Debt To Income Ratio
- Loan Default – Customer defaults on a loan or not
- Gender
- Marital Status
- Wireless Data – If they use wireless data or not
- Data Over Tenure – Amount of data consumed by the customer

The variables were used to use create 5 segmentation using K-means. I ran K-means on all combinations of variables, selecting them in range of 2 to 13 at a time (see Appendix Code Segment 1). The best result was attained with 7 variables. The result was based on K-means parameter “inertia” and silhouette score. The final variable and the rational foe each of these is as following:

- Card Spend Month – Amount spend using card in a month, this was a clear indication of spending habits. It was encoded into 5 equal segments between the highest and lowest amount.
- Voice Last Month – Amount spend on voice calls; this was an indication of usage which helps in understanding their need. It was encoded into 5 segments.
- Owns Mobile Device – Whether a customer owns a device, helps understand his needs and in turn provide information about what plan a user might use,
- Phone Co Tenure – Tenure with phone company helps to know if the customer is willing to continue and if some better plan can be sold to them.
- Age – Different age groups have different need, so they should be targeted accordingly.

- Loan Default – Whether a customer defaults on a loan help in deciding if there will timely bill payments by the customer. This is useful for analysing the risk.
- Wireless Data – Whether a user has uses wireless data or not, if yes then sell them plans that offer more data.

Checking how the cluster were created. Visualizing the feature importance (see Appendix Figure 1), the most important features were the Loan Default (Risk factor) and Wireless Data (Needed or not). The segments are further divided by how long they have been a customer.

Plotting the cluster centres, we can see five individual segment (see Appendix Figure 2)

Also, looking the number of customers belonging to each segment, we get the following: -

Segment	Number of Customers
2	1471
1	1435
5	1047
3	918
4	129

Table 1

Looking into each segment from Table 1, we can define the segment as following:

Segment 1 – User with highest amount of voice data, customers in these segments generally own a device and it also has customers that have been with the company the longest.

Segment 2 (Highest Value) – All users in this segment do not use wireless data and have not defaulted on their loans yet, have been customers for a short time and the lowest voice used the last month. This segment has the maximum room for expansion.

Segment 3 – This segment has the highest number of wireless data users, looking into their data usage can help developing plans for this segment. Rest of the variables are average in this segment, although the debt-to-income ratio is high for this segment.

Segment 4 (Lowest Value) – This segment has the least number of customer so there is no real analysis required for this segment. It is the least useful segment of all.

Segment 5 (High Risk) – This segment contains users with low debt-to-income ratio, lowest tenure with the company, low voice used in last month and equal number of wireless data user and non-user. It contains the highest number of loan defaulters, so proceeding to create plans for this segment should be done with caution.

Appendix

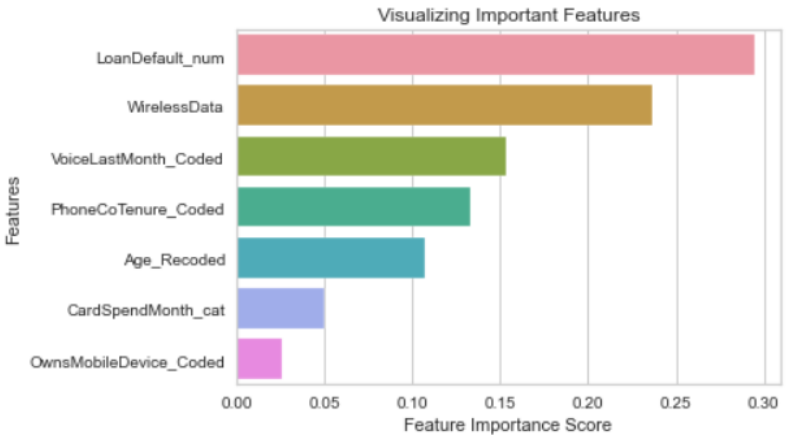

Figure 1

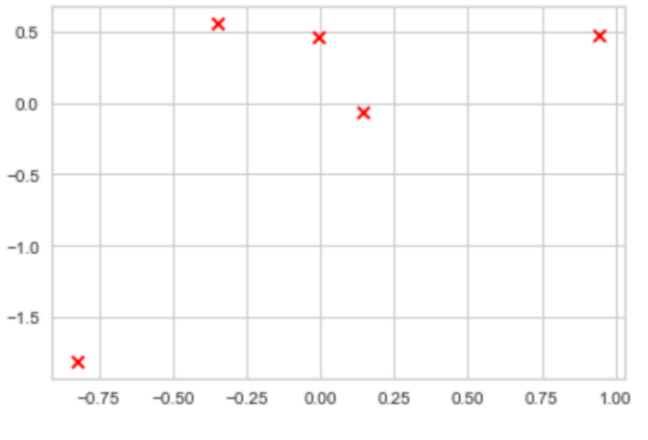

Figure 2

CustomerID	object
Region	int64
TownSize	object
Gender	object
Age	int64
EducationYears	int64
JobCategory	object
UnionMember	object
EmploymentLength	int64
Retired	object
HHIncome	object
DebtToIncomeRatio	float64

CreditDebt	float64
OtherDebt	float64
LoanDefault	object
MaritalStatus	object
HouseholdSize	float64
NumberPets	float64
NumberCats	float64
NumberDogs	float64
NumberBirds	float64
HomeOwner	float64
CarsOwned	int64
CarOwnership	object
CarBrand	object
CarValue	object
CommuteTime	object
PoliticalPartyMem	object
Votes	object
CreditCard	object
CardTenure	int64
CardItemsMonthly	int64
CardSpendMonth	object
ActiveLifestyle	object
PhoneCoTenure	int64
VoiceLastMonth	object
VoiceOverTenure	object
EquipmentRental	object
EquipmentLastMonth	object
EquipmentOverTenure	object
CallingCard	object
WirelessData	object
DataLastMonth	object
DataOverTenure	object
Multiline	object
VM	object
Pager	object
Internet	object
CallerID	object
CallWait	object
CallForward	object
ThreeWayCalling	object
EBilling	object
TVWatchingHours	int64
OwnsPC	object
OwnsMobileDevice	object
OwnsGameSystem	object
OwnsFax	object
NewsSubscriber	object

Table 1

```
col_comb = list(combinations(col_list,12))
sol = {}
for i in col_comb:
    cust_df_kmeans = cust_df[list(i)]
    cust_df_kmeans = cust_df_kmeans.dropna(how='any',axis=0)
    scaler = StandardScaler()
    cust_kmeans_scaled = scaler.fit_transform(cust_df_kmeans)
    cust_kmeans_scaled_df = pd.DataFrame(cust_kmeans_scaled, index = cust_df_kmeans.index)
    cust_kmeans_scaled_df.head()
    kmeans_5 = KMeans(n_clusters = 5, init='k-means++')
    kmeans_5.fit(cust_kmeans_scaled_df)
    sol[kmeans_5.inertia_] = i
min_inertia = []
for i in sol.keys():
    min_inertia.append(i)
print(sol[min(min_inertia)])
```

Code Segment 1