Automobile Customer Segmentation

Overview

Customer segmentation is the practice of dividing a customer base into groups of individuals that are similar in specific ways relevant to marketing, such as age, gender, interests and spending habits.
Companies employing customer segmentation operate under the fact that every customer is different and that their marketing efforts would be better served if they target specific, smaller groups with messages that those consumers would find relevant and lead them to buy something. Companies also hope to gain a deeper understanding of their customers' preferences and needs with the idea of discovering what each segment finds most valuable to more accurately tailor marketing materials toward that segment.

Problem Description

An automobile company has plans to enter new markets with their existing products (P1, P2, P3, P4 and P5). After intensive market research, they've deduced that the behavior of the new market is similar to their existing market. In their existing market, the sales team has classified all customers into 4 segments (A, B, C, D). Then, they performed segmented outreach and communication for different segments of customers. This strategy has worked exceptionally well for them. They plan to use the same strategy on new markets and have identified 2627 new potential customers.
The goal is to predict the right group of the new customers.

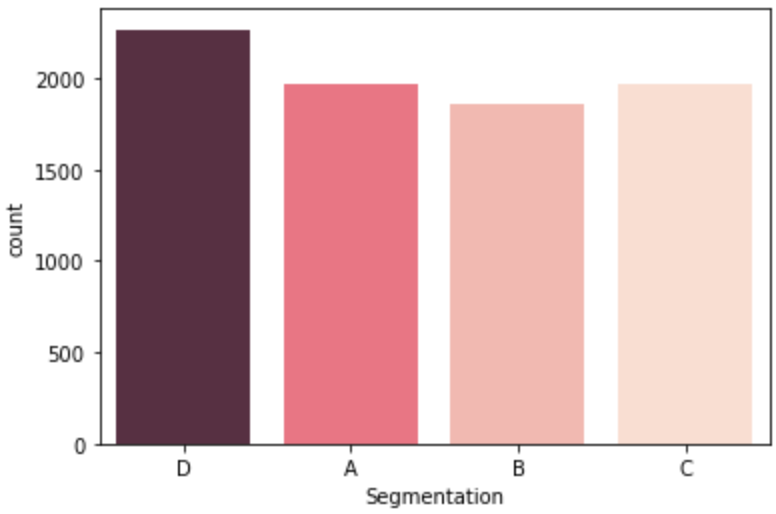
Data Description

col	Description	Type
ID	Customer ID	string
Gender	Customer gender (male/female)	string
Ever_Married	Marital status	string
Age	customer age	int
Graduated	yes/no, gradtuation from college	string
Profession	customer's profession	string
Work_Experience	customer work experience in years	float
Spending_Score	a score of customer spending habbits (Low, Average, High)	string
Family_Size	number of family members	float
Var_1	anonymization feauter	string
Segmentation (target)	segment to which the customer belongs	string

Methodology

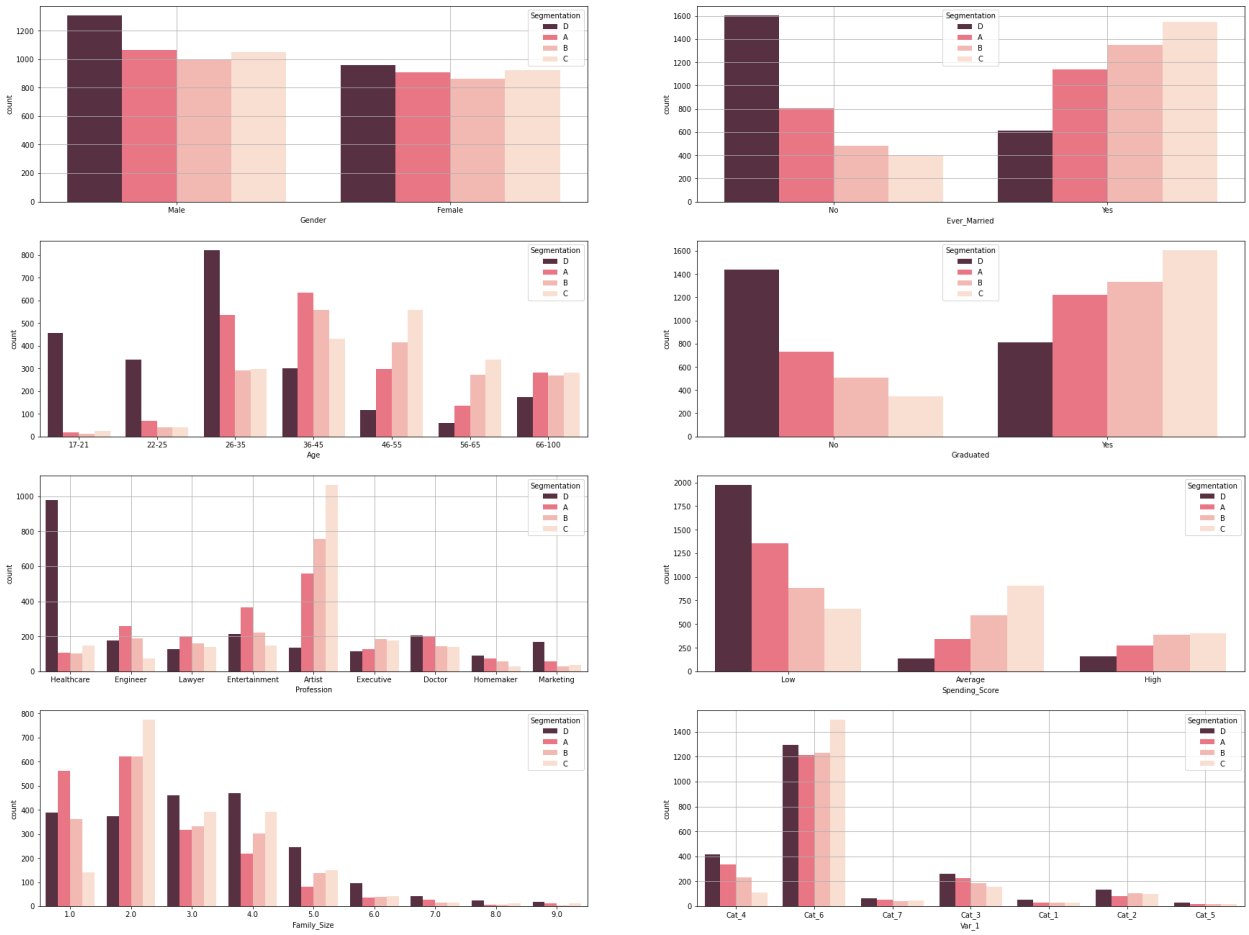
EDA

Checking data balance



By looking at the previous plot we see that the data is almost balanced, therefore imbalance techniques weren't used.

Analyzing Segments



A) 0.461 females, 0.539 males, 0.587 have married before, most are between the age of 26-45, 0.626 graduated from college, most are artists, work in the entertainment industry , or engineers, they have low spending score, most have small families, and in anonymization category 4 and 6.

B) 0.463 females, 0.537 males, 0.738 have married before, most are between the age of 36-55, 0.724 graduated from college, most are artists, work in the entertainment industry , or engineers, they have low spending score, most have small families, and in anonymization category 4 and 6.

C) 0.468 females, 0.532 males, 0.796 have married before, most are between the age of 36-55, 0.822 graduated from college, most are artists, they have average spending score, most have average sized families, and in anonymization category 4 and 6.

D) 0.423 females, 0.577 males, 0.725 have not been married before, most are younger than 35, 0.640 have not graduated from college, most are healthcare providers, they have low spending score, most have average sized families, and in anonymization category 4 and 6.

We notice that sgements B and C are very similar, the most unique segment is segment D, the usual customers of this company are artists, but customers in segment D are mostly healthcare providers, they are young, and never been married unlike the rest of the segments

.

Dealing with nulls

- Fill na with mode if str
- Fill na with median if float

Encoding

- Label encoding and Ordinal Encoding
- Encoded Columns: Ever Married, spending score, graduated, gender)

Feature engineering

- Added new column called ‘AgeGroup’, that
- Family Cat

Dropping Unnecessary Columns

- Dropped ID column because it has no relevant information.
- Dropped the Age column because it’s highly correlated with AgeGroup, and less correlation with segmentation than AgeGroup.

Experiments:

Data split :

- Train: 0.8
- Validation: 0.2

Models

- Basemodel - logistic regression - numerical - not scaled
- Basemodel - logistic regression - ALl features - not scaled
- Basemodel - logistic regression - gridSearch - not scaled
- Basemodel - logistic regression - gridSearch - scaled
- model- knn - gridSearch - not scaled
- model - knn - gridSearch - scaled
- model- decision tree- gridSearch - not scaled
- model- SVM - not scaled
- Model - RandomForestClassifier - not scaled
- Model - PCA - scaled
- Model- xgboost - not scaled
- Model - xgboost - CV - not scaled
- Ensemble Techniques - voting - not scaled
- Ensemble Techniques - Stacking classifier - not scaled

Advance feature engineering

After viewing the accuracy results we noticed that the class B and class C had very similar properties after which we concluded it was most likely causing the low accuracy. To test that theory we plotted the confusion matrix and it showed us that our conclusion is in fact true. It proved that the models confuse class B and class C the most and with highest error rates. So we combined class B and class C as one class named ‘class 1’.

Models:

- Basemodel - logistic regression - numerical - not scaled
- Basemodel - logistic regression - All features - not scaled
- model- knn - gridSearch - not scaled
- model- decision tree- gridSearch - not scaled
- model- SVM - not scaled
- Model - RandomForestClassifier - gridSearch - not scaled
- Model - PCA - scaled
- Model- xgboost - not scaled
- Model - xgboost - CV - not scaled
- Ensemble Techniques - voting - not scaled
- Ensemble Techniques - Stacking classifier - not scaled
- To test whether the imbalance caused by combining two classes to one is causing the model to perform lesser than expected we balanced the data and ran the following models:
- Basemodel - logistic regression - All features - not scaled
- Model - RandomForestClassifier - gridSearch - not scaled

Conclusion

After running the mentioned models it showed that the model performed the best on validation is the RandomForestClassifier: 0.685 on train | 0.671 on val