American University Of Armenia

Capstone Project

Customer Segmentation from XAI perspective

Author :Iren Aghajanyan

“Bachelor of Science In Data Science College of Science and Engineering”

Supervisor: Arnak Poghosyan

Contents

1. Abstract

The main goal of any business is proper strategic planning for revenue/profit maximization. The crucial factors for profitability are building user trust, a clear understanding of customer demands, their shopping patterns, and product use-case scenarios. Those factors can be revealed via proper customer segmentation by grouping customers into relevant clusters that share some general (male/female, age, geographics, etc.) or specific market similarities. In general, the customer segmentation problem can be supervised or unsupervised. However, we will focus our attention towards supervised issues for solutions and measurable assessments.

Modern AI/ML solutions suffer from the main drawback of the lack of explainability/interpretability. Flexible algorithms and especially deep learning approaches act as black boxes without revealing the paths of decisions. At the same time, the latest is essential for a user's trust and confidence to follow those instructions. Thus, for business impact, AI/ML solutions should provide a sufficient level of explainability (XAI), becoming very demanded recently.

Our goal is to revisit the classical customer segmentation supervised learning problems and supplement with required explainability by directly showing the rules or paths that drive the final predictions or data understanding. More specifically, showing why a customer is assigned to a specific cluster and based on what feature(s) that decision is made. The classical and interesting approach should be a combination of flexible approaches with the rule-induction systems that reveal the participation of features with the corresponding threshold values.

2. Introduction

Recent developments in Artificial Intelligence are introducing new Machine Learning techniques to solve increasingly complicated problems with higher predictive capacity. However, this predictive power comes with increasing complexity which can result in difficulties in interpreting these models. Despite the fact that these models produce very accurate results, there needs to be an explanation in order to understand and trust the model’s decisions. This is where eXplainable Artificial Intelligence (XAI) takes the stage.

2.1 What is XAI?

XAI is an emerging field that focuses on different techniques to break the black-box nature of Machine Learning models and produce human-level explanations. This black box represents the models that are too complex to interpret, and that's why they can't explain the results thoroughly. The explanation plays a massive role because many ML applications, besides the daily tasks that concern only the company, sometimes even concern human safety. Due to explainability, the model enables trust. Of course, for some Machine Learning models, the explanation isn't needed as the model itself is not complex, but for this project, the XAI will be used to give some explanation.

3. Data Source

The dataset comes from the Kaggle website, which is a crowd-sourced platform for data scientists and machine learning practitioners.

The original dataset contains 10,695 observations, each with the following attributes:

- ID - Unique ID for each customer
- Gender - The gender of the customer
- Ever_Married - The marital status of the customer
- Age - The age of the customer
- Graduated - Whether the costumer graduated or not
- Profession - The profession of the customer
- Work_Experience - Work Experience of the customer in years
- Spending_Score - Spending score of the customer
- Family_Size - Number of family members for the customer (including the customer)
- Var_1 - Anonymised Category for the customer
- Segmentation - Customer Segment of the customer

My target variable will be Segmentation, which comes in 4 segments from A to D, and in order to analyze it and use it in the models, it will be encoded, later will discuss it more deeply.

4. Literature Review

4.1 Gradient Boosting

Gradient Boosting is an iterative functional gradient algorithm that minimizes a loss function by iteratively choosing a function that points toward the negative gradient, a weak hypothesis.

Gradient Boosting refers to a methodology in machine learning where an ensemble of weak learners is used to improve the model performance in terms of efficiency, accuracy, and interpretability. These learners are defined as having better performance than random chance. Such models are typically decision trees, and their outputs are combined for better overall results. Gradient boosting can optimize Regression, Classification, and Ranking.

For this project, we will focus on Classification models.

Gradient Boosting has three main components:

- Loss Function - The role of the loss function is to estimate how good the model is at making predictions with the given data. This could vary depending on the problem at hand.
- Weak Learner - A weak learner is one that classifies our data but does so poorly, perhaps no better than random guessing. In other words, it has a high error rate. These are typically decision trees.
- Additive Model - This is the iterative and sequential approach of adding the trees (weak learners) one step at a time. After each iteration, we need to be closer to our final model. In other words, each iteration should reduce the value of our loss function.

4.2 Extreme Gradient Boosting (XGBoost)

XGBoost stands for Extreme Gradient Boosting. It is a scalable and highly accurate implementation of gradient boosting that pushes the limits of computing power for boosted tree algorithms. It is built mainly for energizing machine learning model performance and computational speed. With XGBoost, trees are built in parallel instead of sequentially. It follows a level-wise strategy, scanning across gradient values and using these partial sums to evaluate the quality of splits at every possible split in the training set. So the advantages of XGBoost is the following:

- Highly Flexible.
- Uses the power of parallel processing.
- Faster than Gradient Boosting.
- Supports regularization.
- Designed to handle missing data with its in-build features.
- The user can run a cross-validation after each iteration

5. Research Methodology

5.1 Data Cleaning and Preprocessing

As already mentioned, the original dataset contains 10,695 observations and has 11 columns. For a better understanding of our dataset, the data was imported. As we have a Segmentation column that shows the segment of the customer, it is intuitive to make our target variable Segmentation itself. After deciding the target, we continue to get acquainted with the dataset. First of all, it is checked whether the data has duplicates or not, so when it is found out that there are duplicated values, they are dropped. The next and one of the most important steps was to check the existence of the null values. After checking, it turns out that it has some nulls in different columns. There are many options to fill them, such as replacing them with the mean, median, and so on, but the most convenient and optimal solution was to replace them with their mean for numeric columns. And for categorical columns, it was intuitive to drop all the nulls as there was no optimal way to fill them.

When the null was replaced or dropped, the next step was to make the dataset binary. For doing so, One Hot Encoding was used. It is a common way of preprocessing categorical features for machine learning models. This type of encoding creates a new binary feature for each possible category and assigns a value of 1 to the feature of each sample that corresponds to its original category.

The next step was Label Encoding which refers to converting the labels into a numeric form to convert them into a machine-readable form. For our data, we need to encode only target variables. After encoding, we can apply the feature importance process.

5.2 Feature Engineering

Feature Engineering is the process of extracting and organizing the important features from raw data in such a way that it fits the purpose of the machine learning model. It can be thought of as the art of selecting the important features and transforming them into refined and meaningful features that suit the model's needs. Feature Engineering encapsulates various data engineering techniques such as selecting relevant features, handling missing data, encoding the data, and normalizing it.

There are 3 ways to compute the feature importance for the Xgboost:

1. built-in feature importance.
2. permutation based importance.
3. importance computed with SHAP values.

For the built in feature importance i get the following result

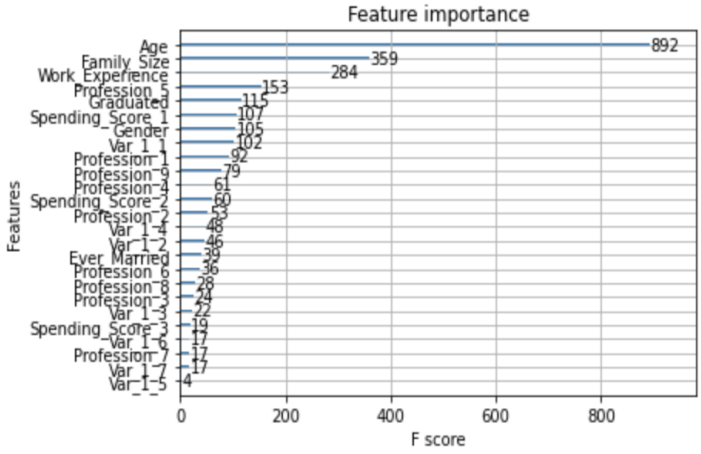

Figure 1: Built in feature importance

From Figure 1, we clearly see that the top 5 features are Age, Family_Size, Work_Experience, Profession_5, and Graduated. But for future work, not only five important ones were chosen, but around 15 features were chosen.

For the Permutation based feature importance i got the following result

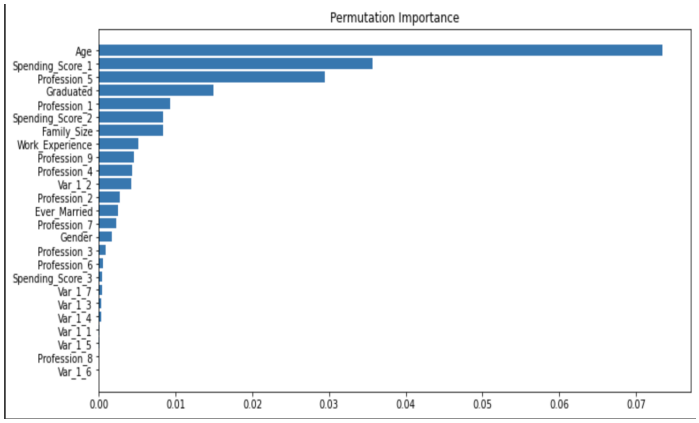

Figure 2: Permutation feature importance

From Figure 2, we see that the 5 most important features are Age,

Spending_Score_1, Profession_5, Graduated and Profession_1.

For the importance feature for SHAP values i got the following result

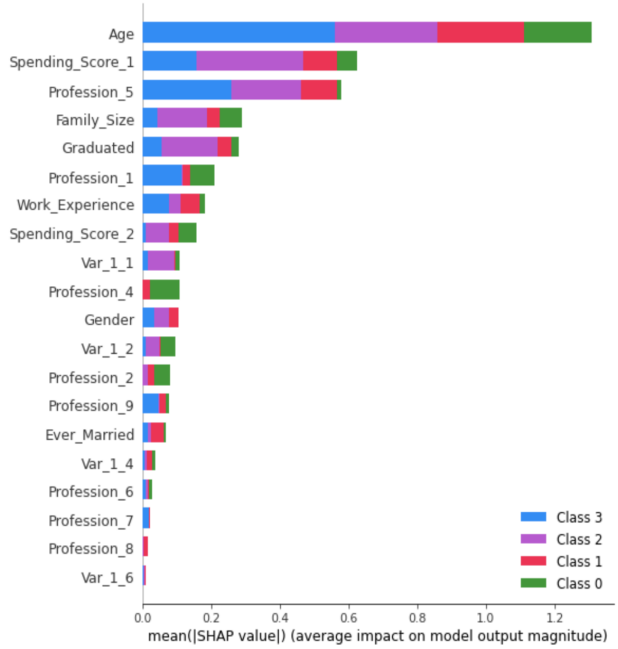

Figure 3:SHAP values for importance features

From Figure 3 we see that the top 5 important features are Age, Spending_Score_1, Profession_5, Family_Size, Graduated.

There are some overlaps with three different ways of calculating feature importance, so the output does not vary very much. So I take the 16 most important features and save them to the new dataset. For later work, the new dataset will be fed to the model, but before that, parameter tuning will be applied.

5.3 Parameter Tuning and Model

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process.

The same machine learning model can require different constraints, weights, or learning rates to generalize different data patterns. These measures are called hyperparameters and have to be tuned so that the model can optimally solve the machine learning problem. Hyperparameter optimization finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined loss function on given independent data. The objective function takes a tuple of hyperparameters and returns the associated loss. Cross-validation is often used to estimate this generalization performance.

For my dataset, I used XGBClassifier, which has many parameters that can be tuned on, mainly n_estimators, max-depth, and learning_rates.
The learning rate is a configurable hyperparameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0. The learning rate controls how quickly the model is adapted to the problem. So for this project, I tried different learning_rates, and the best output I got was with learning-rate = 0.
n_estimators represent the number of trees in the forest. Usually, the higher the number of trees, the better to learn the data, so for

this project, the best output with the n_estimates = 1000.
max_depth represents the depth of each tree in the forest. The deeper the tree, the more splits it has, and it captures more information about the data, so for this project, the best output is given with the max_depth = 5
Also, I specify other parameters in order to get higher valued output, such as min_child_weight, gamma, and subsample. So the output of the XGBClassifier gave a 0.8976 accuracy score(Figure 4).

Here is the implementation of the model

```
xgb_cl = XGBClassifier(
    learning_rate = 0.1,
    max_depth = 5,
    n_estimators= 1000,
    min_child_weight = 1,
    gamma = 0,
    subsample = 0.8)
#Fit
xgb_cl.fit(X_cleaned, y)

#Predict
preds = xgb_cl.predict(X_test)

#Score
accuracy_score(y_test, preds)
```

```
0.8779832935560 86
```

Figure 4: Model prediction

As shown in Figure 4, the model prediction gave around a 0.88 accuracy score. This is the best result that can be obtained with this dataset because I tried to predict the model with different estimates classification models but got low scores for accuracy. Also, I tried to model it with Keras, the high-level API of TensorFlow 2: a highly-productive interface for solving machine learning problems with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity. So when I try to implement it for my dataset, it does not give the desired results.

6. Conclusion

The project aims to find an alternative and more reasonable solution to a customer segmentation problem, such as finding the segment of a person and giving the explainability of why this person belongs to that specific segment. As many ML applications can't provide the explainability part, the XAI algorithms are used. The best model suited for this project was XGBoost which was implemented and predicted with an accuracy score higher than 0,8.
For future work, I will focus more on Neural Networks and find an appropriate model to implement on this dataset.

7. References

1 Arrieta, A. B., íaz-Rodríguez, N., et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Information Fusion. Elsevier, 58, pp. 82-115.

2 Kaggle. Customer Segmentation Classification, Retrieved February 2, 2022, from https://www.kaggle.com/kaushiksuresh147/customer-segmentation

3 Fürnkranz, J., Gamberger, D., and Lavrac, N. (2012). Foundations of Rule Learning. Springer-Verlag.

4 Fürnkranz, J., & Kliegr, T. (2015). A Brief Overview of Rule Learning. Springer Link, 9202, pp. 54-69.

5 Singh, H. (2020). Use of Machine Learning in Customer Segmentation. http://103.47.12.35/bitstream/handle/1/1988/1613114022_HARBHAJAN_SINGH_Final_Project_Report%20-%20Harbhajan%20Singh.pdf?sequence=1&isAllowed=y

6 S. Ozan (2018), "A Case Study on Customer Segmentation by using Machine Learning Methods," International Conference on Artificial Intelligence and Data Processing (IDAP), pp. 1-6.

7 Kurama, V. (2020). *Gradient Boosting In Classification: Not a Black Box Anymore!* https://blog.paperspace.com/. Retrieved March, 2022, from https://blog.paperspace.com/gradient-boosting-for-classification/

8 Morde, V. (2019, April 7). *XGBoost Algorithm: Long May She Reign! | by Vishal Morde*. Towards Data Science. Retrieved March, 2022, from https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

9 Saha, S. (2022, February 10). *XGBoost vs LightGBM: How Are They Different - neptune.ai*. Neptune.ai. Retrieved March, 2022, from https://neptune.ai/blog/xgboost-vs-lightgbm

10 Natassha. (2021, June 5). *Customer segmentation with Python*. Natassha Selvaraj. Retrieved February, 2022, from

https://www.natasshaselvaraj.com/customer-segmentation-with-python/

11 TOTH, D. J. (2021, August 28). *Binary Classification: XGBoost Hyperparameter Tuning Scenarios by Non-exhaustive Grid Search and*…. Towards Data Science. Retrieved 2022, from https://towardsdatascience.com/binary-classification-xgboost-hyperparameter-tuning-scenarios-by-non-exhaustive-grid-search-and-c261f4ce098

12 Formation. (2022, April 25). *Customer Segmentation Models: A Better Approach for 2022*. Formation.ai. Retrieved 2022, from https://formation.ai/blog/customer-segmentation-models-theres-a-better-approach-for-2022/

13 Brownlee, J. (2021, March 29). *Tune XGBoost Performance With Learning Curves*. Machine Learning Mastery. Retrieved 2022, from https://machinelearningmastery.com/tune-xgboost-performance-with-learning-curves/

14 nvidia. (n.d.). *What is XGBoost? | Data Science | NVIDIA Glossary*. Nvidia. Retrieved 2022, from https://www.nvidia.com/en-us/glossary/data-science/xgboost/

15 Sreemany, T. (2021, October 3). *What is Feature Engineering? Definition and FAQs*. HEAVY.AI. Retrieved 2022, from https://www.omnisci.com/technical-glossary/feature-engineering