Camille Breuil

Technical tesh

07.07.2023

Introduction

This report presents the findings of the data science technical test focused on customer churn analysis. The objective of this project was to address the challenge of customer churn and its implications for the organization. Customer churn refers to the phenomenon where customers stop using the products or services, resulting in a loss of revenue and potential financial impact.

The dataset contains features such as gross user turnover percentages, net user turnover, average cart value, and customer tenure, among others. Through exploratory data analysis (EDA) and the development of a machine-learning model, we aimed to understand the dataset, predict customer churn, and make informed recommendations. This report provides an overview of the dataset, key findings from the EDA, the machine-learning model approach, and actionable recommendations to mitigate customer churn. The analysis and recommendations are intended to support effective customer retention strategies and improve business outcomes. Further validation and refinement may be required before implementation in a production environment.

Data overview

The dataset provided for this analysis consists of a comprehensive collection of features related to customer behavior and turnover. It encompasses various aspects that contribute to understanding and predicting customer churn. Here is an overview of the key features included in the dataset:

1. **Gross User Turnover Percentages**:

- _3_pcts_gross_turnover_6: Gross user turnover percentage (before subtracting returns) in rayon 6 (percentage).
- _3_pcts_gross_turnover_7: Gross user turnover percentage (before subtracting returns) in rayon 7 (percentage).
- _3_pcts_gross_turnover_3: Gross user turnover percentage (before subtracting returns) in rayon 3 (percentage).
- _3_pcts_gross_turnover_4: Gross user turnover percentage (before subtracting returns) in rayon 4 (percentage).
- _3_pcts_gross_turnover_9: Gross user turnover percentage (before subtracting returns) in rayon 9 (percentage).
- _3_pcts_gross_turnover_1: Gross user turnover percentage (before subtracting returns) in rayon 1 (percentage).

2. **Net User Turnover**:
   - _3_turnover_24months: Net user turnover over the past 24 months (returns subtracted).
3. **Average Cart Value**:
   - _3_average_cart_24months_percentile: Percentile of average cart value over the last 24 months.
4. **Order Frequency**:
   - _2_pct_frequency_6months: Percentage of orders made by the user in the last 6 months.
   - _2_avg_days_between_orders: Average number of days between user transactions.
   - _2_pct_frequency_12months: Percentage of orders made by the user in the last 12 months.
   - _2_std_days_between_orders: Standard deviation of the number of days between user transactions.
   - _2_frequency_24months: Number of orders made by the user in the past 24 months.
5. **Transactional Recency**:

    - _2_transactional_recency: Number of days since the last purchase-related activity (e.g., return, exchange).
6. **Customer Tenure**:
    - _1_customer_tenure_months: Number of months the user has been a client at Leroy Merlin, up until the max_cutoff_date.
7. **Target Variable**:
    - _10_target_is_churn: Indicates whether the customer churned in the last year (1 for churned, 0 for not churned).

Each feature contributes to our understanding of customer churn dynamics. It is important to analyze the relationship between these features and the target variable in order to build an effective predictive model.

In the subsequent sections, we will explore the dataset further through exploratory data analysis (EDA) techniques, uncovering insights and patterns that will inform our understanding of customer churn dynamics.

Exploratory Data Analysis (EDA)

EDA was performed in the notebook eda.py, please refer to this document for details.

Key observations :

1. **Dataset Overview**: The dataset consists of 15 numerical features and 1 label column (_10_target_is_churn). It contains a total of 100,903 entries, and there are no missing data points.
2. **Feature Statistics**: The features' mean values generally range from 0 to 100, except for the feature _3_turnover_24months, which has a mean value around 1000. This indicates that most of the features are on a similar scale, except for the turnover feature that might have a higher magnitude.
3. **Class Imbalance**: The dataset exhibits a strong class imbalance, with approximately 10% of positive values in the target variable indicating churn. This imbalance may need to be

addressed during model training to ensure accurate predictions and avoid biased results.

4. **Correlations**: There are no strong linear correlations observed between features or between individual features and the target variable. This implies that the relationship between the features and churn may not be linear and may require more advanced modeling techniques to capture the underlying patterns.
5. **Outliers**: Outliers are present in each feature, except for _3_average_cart_24months_percentile. These outliers may represent extreme values or anomalies in the data and could potentially have an impact on the modeling process. Handling outliers appropriately may be necessary to prevent them from unduly influencing the model's performance.
6. **Skewed Distributions**: The distributions of almost all features are skewed. This suggests that the data is not symmetrical and may have a longer tail on one side. Addressing the skewness may be important to ensure the assumptions of the machine-learning algorithms are met and to improve the model's performance.

Based on these observations, it is evident that the dataset presents some specific challenges such as class imbalance, outliers, and skewed distributions. These factors need to be taken into consideration during the subsequent stages of the analysis and model development. By addressing these challenges appropriately, we can improve the performance of the machine-learning model for predicting customer churn.

Machine learning models

Data preprocessing

Handling Missing Data:

- Since the dataset has no missing data, no specific step is required.

Addressing Outliers:

-
- Not implemented yet but we could consider methods such as winsorization, logarithmic transformation, or removing extreme values to handle the presence of outliers in the dataset.

Feature Scaling:

- Feature scaling is important to ensure that all features contribute equally during model training. We have tested standardization (z-score normalization) and min-max scaling.

Handling Skewed Distributions:

- It may be important to address the skewness to meet the assumptions of some machine-learning models (normal distributions for linear regression for example). Logarithmic or power transformations could be used but have not be implemented at this stage.

Feature Engineering:

- We could include feature engineering steps , such as creating interaction terms, polynomial features, or aggregating variables to capture higher-order relationships or domain-specific insights. Not included at this stage.

The file /src/preprocessing.py contains all the useful functions for data preprocessing.

Model training

Model Selection

Considering the dataset and objectives, I selected 5 different models :

- **Random Forest Classifier**: Random Forests are known for their ability to handle imbalanced datasets well. They are fast to train and can handle a large number of features and observations effectively. Random Forests also provide built-in feature importance rankings, allowing you to understand which features are most influential in predicting churn.
- **Gradient Boosting Classifier**: Gradient Boosting models have shown great performance on imbalanced datasets. They work by combining weak learners to create a strong model and can effectively handle class imbalance through techniques like class weights or sample weights.
- **Support Vector Machines (SVM)**: SVMs can perform well on imbalanced datasets, especially when combined with appropriate class weighting techniques. They can handle high-dimensional feature spaces efficiently and provide robust decision boundaries.
- **AdaBoost Classifier**: AdaBoost is an ensemble learning technique that combines multiple weak classifiers to create a strong model. It can handle imbalanced datasets by assigning higher weights to the minority class samples during training, making it more sensitive to minority class instances.
- **Logistic Regression with class weights**: Logistic Regression is a popular and interpretable model that can be effective on imbalanced datasets. By assigning appropriate class weights, you can address class imbalance and ensure that the model focuses on correctly predicting both the majority and minority classes.

Class Imbalance Handling and Evaluation Metrics

Because the classes are imbalanced and we want to be sure to detect a customer with a risk of churn we will use metrics such as **recall** or **F1-score** (balance between recall and precision).

We could have considered oversampling the minority class with techniques such as SMOTE or ADASYN (not included at this stage). We also **adjusted class weights** during training of the models (except for AdaBoost, no simple implementation).

Train-test split

We used a split ratio of **80:20** for train-test split of data. We performed a **stratified** split, where the class proportions are preserved in both the training and testing sets.

Cross-Validation

We also used cross validation with **stratified k-fold** cross-validation technique. Cross-validation can provide more robust performance estimates by repeatedly training and testing the model on different subsets of the data. It helps to mitigate the potential impact of a single train-test split on the evaluation metrics. **8 folds** were used.

Model Evaluation and Results

Results are stored as .csv file in /results repository.

We can observe that RandomForest performs poorly and would require hyperparameters optimization to get satisfying results.

As SVM is the model for which we have the best results we went further with hyperparameters optimization and improved recall score from 0.8273 to 0.8449 with the following hyperparameter values: {'svc__C': 0.20584494295802447, 'svc__gamma': 'auto', 'svc__kernel': 'rbf'}.

All the models with recall score > 0.8 have been saved in /models repository.

Next Steps and Improvements

A lot of things could still be improved on this project with more time. Some of them have already been mentioned in this report, here are some others:

- Test other models
- Add other parameters such as decision thresholds
- Add lint and tests
- Improve code quality (comments, factorisation,...)
- Model deployment : the model could be deployed through an API (Flask, fastAPI). We should ensure new test data will go through all the preprocessing steps and use a script for prediction, loading the model, making the preprocessing and predicting the target value.

Conclusion

These initial results show that the data presented can be used to predict and prevent customer churn and that efficient machine learning models could be trained for this task.