AWS Machine Learning Engineer Nanodegree
Capstone Proposal

Churn classification

Matheus Ribeiro Cerqueira

2023

Domain Background

Customer churn, also referred to as customer attrition, poses a significant challenge for businesses. It arises when customers discontinue the use of a company's products or services, and high churn rates can have adverse effects on a company's revenue and profitability [1] .

To tackle this issue, machine learning algorithms can be leveraged to identify the factors that contribute to churn. Churn models are designed to identify early warning signs and recognize customers who are more likely to voluntarily leave [2] and [3]. As part of this project, I will be delving into three different algorithms: logistic regression, decision tree, and random forest. Through the application of these three powerful tools, I aim to develop a highly accurate classifier that can predict which customers are likely to churn and which are not.

Problem Statement

In the fiercely competitive industry, customer churn poses a significant challenge and is critical to the success of many businesses. Churn transpires when a customer decides to shift to a rival that provides a superior solution or product. It is imperative for any business to identify customers who are likely to churn and implement effective retention strategies for long-term success, given that acquiring new customers is more expensive than retaining current ones. However, manually identifying such customers or using conventional techniques can be arduous and time-consuming. Hence, developing machine learning techniques that can accurately identify customers at risk of churning based on their past behavior and interactions can enhance the ability to spot and prevent future churning customers. The objective of this project is to create a model capable of accurately predicting churn and enhancing customer retention.

Datasets and Inputs

For this project I will use a churn dataset that can be founded at Kaggle Bank Customer Churn Dataset. This dataset comprises 10.000 observations, 12 features and are divided into two classes 0 (not churn) and 1 (churn). The features are described below:

Column Name	Description
customer_id	Unique identifier for each customer. This column is not used as an input for the model.
credit_score	A numerical value that represents the creditworthiness of a customer. It is used as an input to the model to predict churn. Higher scores indicate lower risk and vice versa.
country	The country where the customer resides. It is used as an input to the model to predict churn.
gender	The gender of the customer. It is used as an input to the model to predict churn.
age	The age of the customer. It is used as an input to the model to predict churn.
tenure	The number of years that the customer has been with the bank. It is used as an input to the model to predict churn.
balance	The amount of money that the customer has in his/her account. It is used as an input to the model to predict churn.
products_number	The number of banking products that the customer has with the bank. It is used as an input to the model to predict churn.
credit_card	Whether the customer has a credit card with the bank or not. It is used as an input to the model to predict churn.
active_member	Whether the customer is an active member of the bank or not. It is used as an input to the model to predict churn.
estimated_salary	The estimated salary of the customer. It is used as an input to the model to predict churn.
churn	The target variable. A binary variable that indicates whether the customer has left the bank or not during a certain period. It takes a value of 1 if the customer has left and 0 if not.

In this project, I will divide the dataset into three subsets: Training (70%), Validation (20%), and Test (10%).

Solution Statement

I intend to train three models for this project: a Logistic Regression model, a Decision Tree model, and a Random Forest model. I will evaluate the performance of each model using metrics such as accuracy, recall, and precision. To select the optimal model, I will use the AUC-ROC metric.

Benchmark Model

To establish a benchmark, I will utilize the previously mentioned models alongside the final model. I will compare the final model's performance with the others using the

AUC-ROC metric. To develop these models, I will leverage the Scikit-learn library and its features such as Pipelines for creating pipelines, OneHotEncoder for creating dummies, RobustScaler for data normalization, and SimpleInputer for missing data imputation. For data manipulation, I will employ Polars, a DataFrame library that is fully written in Rust and provides an API for Python. To visualize data, I will use the Matplotlib, Folium, and Searborn libraries. Given that the positive class has a low number of examples, I will apply the Synthetic Minority Oversampling Technique (SMOTE) from the imblearn library to address the issue of imbalance [5], [6] and [7].

Evaluation Metrics

For evaluation, I will use several metrics such as: accuracy, confusion matrix and AUC-ROC.

Project Design

Data Download

For downloading the dataset, I will use the Kaggle API provided by Kaggle to download the zip file and then extract it using the bash unzip command.

Data Preprocessing

Subsequently, I will utilize Polars to load the complete dataset and exclude unwanted features such as customer_id. Then, I will calculate statistics like class distribution, mean, median, minimum, and maximum values, as well as employ box plots to detect outliers and compute correlations between some features.

Subset Data Splitting

Then, I will split the selected dataset into 70% training, 20% validation, 10% test datasets using scikit-learn help functions.

Model Training and Evaluation

Afterwards, I will use the scikit-learn classes LogisticRegression, DecisionTree and RandomForest with default hyperparameters to measure the improvement by using the SMOTE technique. Then, I will do the hyperparameter optimization of the models and evaluate them by applying a k-fold cross-validation for measure accuracy [8] and plot the confusion matrix of each model. Finally, I will compare each other using the AUC-ROC metric to select the best one.

References

[1]. N. Forhad, M. S. Hussain, and R. M. Rahman, "Churn analysis: Predicting churners," in Proceedings of the Ninth International Conference on Digital Information Management (ICDIM 2014), Phitsanulok, Thailand, 2014, pp. 237-241, doi: 10.1109/ICDIM.2014.6991433.

[2]. Qureshi, Saad, Ammar Rehman, Ali Qamar, Aatif Kamal, and Ahsan Rehman. "Telecommunication Subscribers' Churn Prediction Model Using Machine Learning." In Proceedings of the 8th International Conference on Digital Information Management (ICDIM 2013), 2013, pp. 133-137, doi: 10.1109/ICDIM.2013.6693977.

[3]. Ullah, Irfan, Basit Raza, Ahmad Malik, Muhammad Imran, Saif Islam, and Sung Won Kim. "A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector." IEEE Access, vol. 7, pp. 104634-104647, 2019, doi: 10.1109/ACCESS.2019.2914999.

[4]. Khan, Muhammad, Johua Manoj, Anikate Singh, and Joshua Blumenstock. "Behavioral Modeling for Churn Prediction: Early Indicators and Accurate Predictors of Custom Defection and Loyalty." In Proceedings of the IEEE International Congress on Big Data (BigData Congress), 2015, pp. 7-14, doi: 10.1109/BigDataCongress.2015.107.

[5]. G. Menardi and N. Torelli, "Training and assessing classification rules with imbalanced data," Data Mining and Knowledge Discovery, vol. 28, no. 1, pp. 92-122, 2014, https://doi.org/10.1007/s10618-012-0295-5.

[6]. V. S. Spelmen and R. Porkodi, "A Review on Handling Imbalanced Data," in Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, India, 2018, pp. 1-11, doi: 10.1109/ICCTCT.2018.8551020.

[7]. N. V. Chawla, K. W. Bowyer, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[8]. R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95), San Francisco, CA, USA, 1995, pp. 1137-1143.