Customer Churn Predictions

Abstract

Churn prediction is becoming a major focus of **banks** who wish to retain customers by satisfying their needs under resource constraints. In churn prediction, an important yet challenging problem is the imbalance in the data distribution. In this paper, we propose a novel learning method, called improved balanced random forests (IBRF), and demonstrate its application to churn prediction. We investigate the effectiveness of the standard random forests approach in predicting customer churn, while also integrating sampling techniques and cost-sensitive learning into the approach to achieve a better performance than most existing algorithms. We apply the method to a real bank customer churn data set. It is found to improve prediction accuracy significantly compared with other algorithms, such as Support vector machine, decision trees, and Logistic Regression, GaussianNB. Moreover, RF will produce better prediction results than other algorithms.

Introduction

With the growing competition in banking industry, banks are required to follow customer retention strategies while they are try to increase their market share by acquiring new customers

Customer churn is a fundamental problem for companies and it is defined as the loss of customers because they move out to competitors. Being able to predict customer churning behaviour in advance, gives an institution a high valuable insight in order to retain and increase their customer base. In this paper, we work on predictive analysis of churning behaviour in financial institution base on the dataset extracted from the bank database.

Data Preparation

To achieve better performance, the categorical data was transformed to numerical format using the Label Encoder function in Python. Feature scaling was applied to normalize the data and improved the computational time. And done the EDA process to analyse the missing values, outlier detection.

Tools and Libraries

Matplotlib: is an amazing visualization library in Python programming language for two-dimensional plots of arrays. One of the greatest benefits of visualization is that it allows high dimensionality data to visualize and easily understandable and it consists of several plots like line, bar, scatter, histogram and so on.

Pandas: is the most popular Python programming language package that offers powerful, expensive and flexible data structures that make data manipulation and analysis easy.

NumPy: is the fundamental package for scientific computing in Python programming language that contains a powerful N-dimensional array object and also useful in linear algebra. Label Encoder: is a Python programming language package that is used to transform non-numerical labels

Algorithms used:

AdaBoostClassifier: AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances.

Random Forest: "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."

Support Vector Machine: Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

GridSearchCV: GridSearchCV is a library function that is a member of sklearn's model_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters

Conclusion:

After introducing the datasets performing the EDA to analyse the data, splitting it and using feature engineering to build a model using algorithms and cross validation after than hyper tunning the data getting 50% accuracy.

Submitted By

Nidhi