Customer Churn Prediction in Telecommunication Industry

Niruthikka Sritharan
Department of Computer Science and Engineering
University of Moratuwa
Colombo, Sri Lanka
niruthikka.19@cse.mrt.ac.lk

***Abstract*—This paper presents a comparative study on some popular machine learning techniques such as eXtreme Gradient Boosting, Support Vector Classifier, Light Gradient Boosting Machine and K Nearest Neighbors that were applied to customer churn prediction, which is a challenging problem in the telecommunication industry. The methodology proposed in this study consists of the phases: data pre-processing, feature engineering, handling class imbalance and modeling. The highest accuracy of 97.6% and F1 score of 97.1% were achieved by the hypertuned Light Gradient Boosting Machine.**

***Index Terms*—Churn Prediction, SMOTE, Machine Learning Models, Feature Engineering**

I. Introduction

Customer churn is the termination of their contract with a subscription-based business. Amongst several industries that suffer from customer churn issues, the telecommunications industry holds a significant position with a yearly churn rate that ranges from 23.4% to 43% [1]. Prevention of customer churn facilitates customer retention. New customer acquisition requires the devotion of many resources, which makes customer retention relatively less expensive. However, tracking just real churn only serves as a lagging indicator of poor customer experience. In contrast, a predictive churn model developed using machine learning allows proactive engagement with potential churning customers, paving way to take apt customer retention actions. This study aims to develop a prediction model as a solution for the above-mentioned problem using machine learning algorithms.

This paper is organized as follows. Section II describes the provided dataset, data preprocessing and feature engineering techniques, and model selection. Section III presents the results and analysis. Finally, section IV concludes the paper.

II. Methodology

A. Dataset

The provided Churn dataset consists of 19 predictor variables and 1 target variable. The training and test datasets had 2321 and 1500 records respectively. Both the datasets had missing values, outliers and certain erroneous values (e.g.: - negative values in total number of daytime calls). Furthermore, the provided dataset was characterized by class imbalance since it had 1741 (75.2%) data records of non-churn customers and only 575 (24.8%) data records of churn customers. However, the problem of this study demands focus on the churn customers who belong to the minority group.

B. Data Preprocessing

Duplicate rows were identified when examining training dataset without customer_id attribute. The 4 redundant rows were removed.

Histogram plots of attributes showed the skewed distribution of data in some of these attributes. Thus, the measure of central tendency, median, was used to impute the missing values and replace the ridiculously high values observed in metric variable columns (such as 'account_length', 'customer_service_calls', 'total_day_calls', 'total_eve_calls', 'total_night_calls', 'total_intl_calls').

Missing values in the categorical feature columns such as 'location_code' and 'intertiol_plan' were imputed using the mode.

Owing to the association between 'voice_mail_plan' and 'number_vm_messages', missing values in 'voice_mail_plan' was imputed with 'no' if corresponding 'number_vm_messages' was 0 and with 'yes' otherwise. Similarly, missing values in 'number_vm_messages' was imputed with 0 if corresponding 'voice_mail_plan' was 'no' and with median of 'number_vm_messages' otherwise.

The high correlation between the charge and respective minutes column were used to handle the missing value and abnormally high values in those columns. The charge/minute column was synthesized for each of day, evening, night and international calls. These turned out to be columns of low variation.

TABLE I
Measures of Central Tendency of Synthesized Charge/Minute Columns

Measures of Central Tendency	Charge/Minute			
	Day	***Evening***	***Night***	***International***
Mean	0.170	0.085	0.047	0.269
Median	0.170	0.085	0.045	0.270
Mode	0.170	0.085	0.045	0.270

For instance, the missing values and outliers in 'total_day_charge' were replaced using 0.170*'total_day_min', whereas the NaNs and outliers in 'total_day_min' were handled using 'total_day_charge'/0.170. Thus, in this manner,

the rates found in the above table were used for imputation of missing values and outliers in all the charge and minute columns.

The negative values were simply made positive, since the corresponding positive values seemed to be a reasonable substitution.

Finally, the 5 rows with missing values for the target variable were removed.

C. Feature Engineering

a) Feature Encoding: Most of the Machine learning algorithms can not handle categorical variables unless converted to numerical values. Therefore, the dichotomous attributes 'intertiol_plan', 'voice_mail_plan' and 'Churn' were label encoded.

b) Feature Selection: Exploratory analysis done prior to the feature engineering helped in the derivation of some valuable insights. Accordingly, it was discovered that there is not much variation among the Churn and Non-churn percentages in the 3 locations as indicated in "Fig. 1". Thus, 'location_code' is not included in Churn analysis. This feature elimination improved the performance of classifiers as well. Furthermore, the correlation matrix displayed a high correlation, as expected, between the charge and corresponding minutes columns for day, evening, night and international calls. However, eliminating either the charge columns or minute columns resulted in the degradation of the performance of the classifiers experimented in this study.

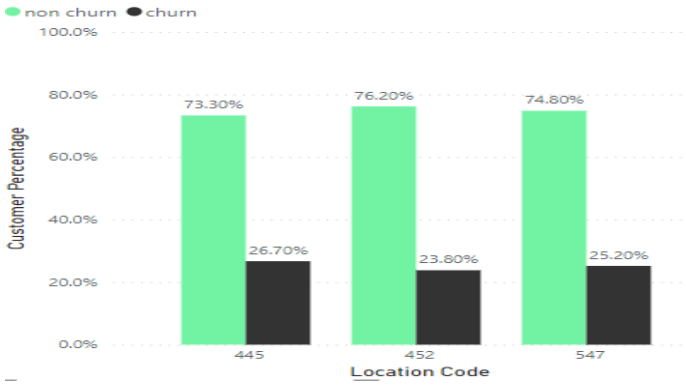

Fig. 1. Churn and Non-churn percentage by location code

c) Feature Synthesis: The feature creation that worked best, with the models experimented in this study, out of the different features that were synthesized is elaborated in this section. A new feature, 'total_charge', was engineered as the summation of 'total_day_charge', 'total_eve_charge' and 'total_night_charge'. Similarly, 'total_minutes' and 'total_calls' features were also synthesized. With the inclusion of these 3 new features, the charge and minute columns corresponding to day, evening and night time calls were removed. This approach also assisted in dimensionality reduction, since 9 features were reduced to 3. Columns pertaining to International calls were excluded from this aggregation, since doing so reduced the performance of the classifying models.

d) Feature Importance: The feature importance graphs were plotted for each of the experimented models. Accordingly, in each case, the exclusion or inclusion of features of low importance were decided upon the performance of the models as gauged by the evaluation metrics.

e) Feature Transformation: Normalization of features was employed for certain models that compute distances among the data points (e.g.: - KNN) and for models that assume data to be in a standard range (e.g.: - SVC). Normalization attempts to give equal weight to all the attributes, regardless of the range of their values.

D. Handling Class Imbalance

The 2 main problems that have to be handled when mining imbalanced classes are selection of appropriate evaluation metrics and dealing with lack of data in minority classes in comparison to the large amount of data in majority class [2]. This subsection represents the approaches used in this study to address the class imbalance.

a) Appropriate Evaluation Metrics: The metrics chosen to evaluate the model performance are precision and recall (or sensitivity), since both focus on the minority class and thus are useful for imbalanced classification. F1 score, which is the harmonic mean of precision and recall, and thus a popular metric for imbalanced classification was also used. Further, Precision-Recall curve, an effective diagnostic for imbalanced binary classification models, was also plotted as and when required.

b) Oversampling: In a classic oversampling technique, the minority data is duplicated from the minority data population. While it increases the number of data, it does not give any new information or variation to the machine learning model. However, another oversampling approach, SMOTE (Synthetic Minority Oversampling Technique) creates new synthetic data points via randomized interpolation between nearest neighbors of instances belonging to the minority class [2]. Thus, SMOTE was used to handle the data imbalance in this study. Sampling strategy ratios and k neighbors were varied to experiment and find the parameter values that could best supplement different models' performance.

E. Modeling Approach

The classification models that were experimented in this study are K Nearest Neighbors (KNN), Support Vector Machines (SVM), eXtreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine (LightGBM). Additionally, some of the ensemble algorithms implemented by Sklearn.ensemble library were also experimented. Finally, an ensemble was formed with VotingClassifier using three models of the ensemble algorithms that scored the most: LightGBM, XGBoost and Histogram-Based Gradient Boosting Ensemble.

The dataset was split into train and test sets having split percentages of 80% and 20% respectively. Different subsets of features (including the newly synthesized ones) were given as input to each of the classifiers used in the study. Each of the models were learnt with and without SMOTE. Further,

models were also learnt with and without the application of Min-max normalization. Using the evaluation metrics the best combination was chosen for each classifier, prior to the hypertuning of model parameters.

Cross-validation was used to assess the performance of models. This study used 10-fold cross-validation. Accordingly, in cross-validation, the training set was divided into 10 segments and 9 of them were used for training and remaining 1 segment for testing. Average scores of all tests would be displayed. This approach helps in keeping the model generalized without overfitting to take in the peculiarities of the given dataset [3]. The hyperparameters chosen via cross-validation were further validated using the test dataset obtained from the train-test split.

III. Results and Analysis

The evaluation metrics obtained for each of the approach experimented in the study are presented in this section. First, the accuracy and F1 scores obtained with and without the application of normalization are illustrated below.

TABLE II
RESULTS WITH AND WITHOUT MIN-MAX NORMALIZATION

Classifying Model	Without Normalization		With Normalization	
	Accuracy	*F1 Score*	*Accuracy*	*F1 Score*
XGBoost	0.961	0.917	0.961	0.917
SVC	0.752	0.645	**0.881**	**0.709**
LightGBM	0.961	0.917	**0.965**	**0.926**
KNN	0.801	0.558	**0.907**	**0.790**
Ensemble	0.957	0.907	**0.963**	**0.921**

Accordingly, it is evident that the performance of KNN and Support Vector Classifier (SVC) have increased significantly, as expected, with Min-max normalization. At the same time, LightGBM and Ensemble models perform slightly better with Min-max normalization, while performance of XGBoost remains indifferent to this normalization. Though not illustrated in Table II, precision and recall scores have also improved with the application of Min-max normalization. Henceforth, normalized datasets were used for all 5 classification models.

Next, SMOTE was applied to handle the imbalance in the dataset. To determine the ratio by which the minority class should be oversampled, repeated 10-fold cross validation was employed using average F1 score as the performance indicator. The ratios were to be chosen from the values: 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0. The ratio that worked best for each of the models is indicated in the following table.

TABLE III
SAMPLING STRATEGY FOR SMOTE

Model	*XGBoost*	*SVC*	*LightGBM*	*KNN*	*Ensemble*
Ratio	0.6	0.5	0.7	0.8	0.6

The accuracy and F1 scores obtained with and without SMOTE are displayed in "Fig. 2" and "Fig. 3" respectively. When applying SMOTE, the minority class was oversampled by the ratio chosen for each model as in Table III.

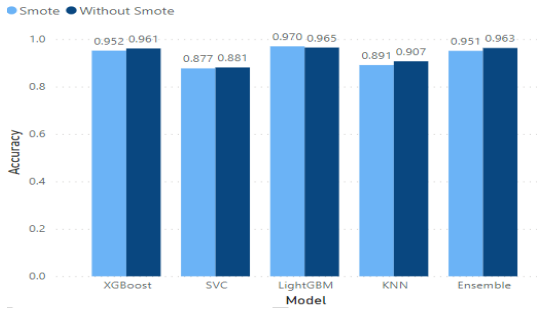

Fig. 2. Accuracy score of Models with and without SMOTE

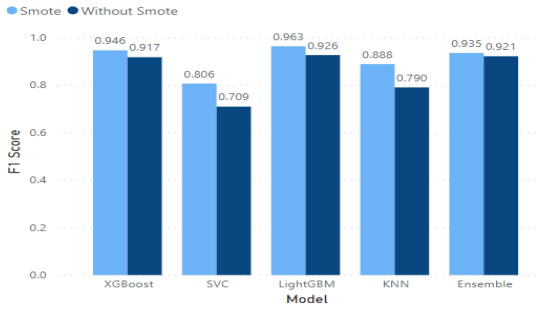

Fig. 3. F1 score of Models with and without SMOTE

From "Fig. 3" it is evident that F1 score has improved with the application of SMOTE for all 5 models. In contrast, accuracy seems to have reduced, only insignificantly, in 4 of the models with the application of SMOTE. Nevertheless, since F1 is a better measure for imbalanced classification, SMOTE was decided to be applied for all models henceforth.

Manual Search and Grid Search were used for finding the optimal hyperparameters to increase the model performance. The best hyperparameters thus obtained and evaluation metrics pertaining to those hypertuned models are illustrated below.

A. XGBoost

Optimal hyperparameters: learning_rate = 0.3, max_depth = 12, n_estimators = 500

Accuracy	*F1 Score*	*Precision*	*Recall*
0.962	0.957	0.985	0.931

B. SVC

Optimal hyperparameters: C = 100, gamma = 1, kernel = rbf

Accuracy	*F1 Score*	*Precision*	*Recall*
0.917	0.869	0.916	0.826

C. LightGBM

The validation curves gauged with F1 scores for different values of n_estimators and learning_rates used in the hypertuning of LightGBM are indicated in "Fig. 4" and "Fig. 5".

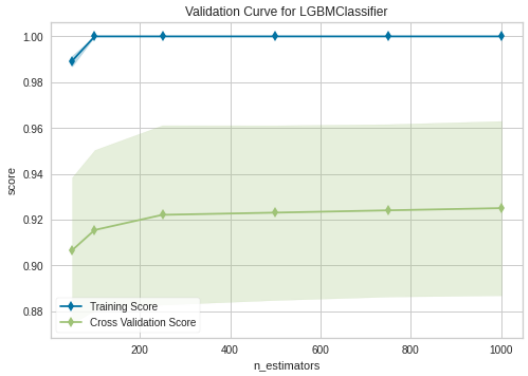

Fig. 4. Validation curves scored by F1 score for n_estimators

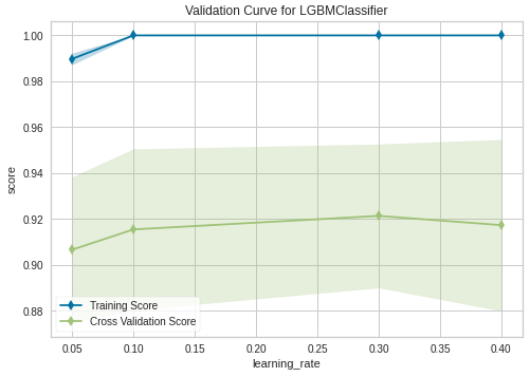

Fig. 5. Validation curves scored by F1 score for learning_rate

Optimal hyperparameters: learning_rate = 0.3, n_estimators = 750

Accuracy	*F1 Score*	*Precision*	*Recall*
0.976	0.971	0.983	0.959

D. KNN

Optimal hyperparameter: n_neighbors = 5

Accuracy	*F1 Score*	*Precision*	*Recall*
0.898	0.896	0.847	0.952

Finally, precision-recall curves for all the 4 models described above is illustrated in "Fig. 6".

From the evaluation metrics indicated for each of the hypertuned models and Precision-Recall Curves, it is evident that LightGBM, with the chosen optimal hyperparameters, gives the best performance in Churn prediction.

Likewise, in Kaggle contest also the best public and private scores, 0.92666 and 0.93833 respectively, were obtained for the LightGBM classifier with the hyperparameters learning_rate = 0.3 and n_estimators = 750, for which normalized (Min-max scaling) and oversampled (using SMOTE) data with reduced features (account_length, intertiol_plan, voice_mail_plan, number_vm_messages, total_charge, total_calls, total_minutes, total_intl_minutes, total_intl_calls,total_intl_charge, customer_service_calls) was used.

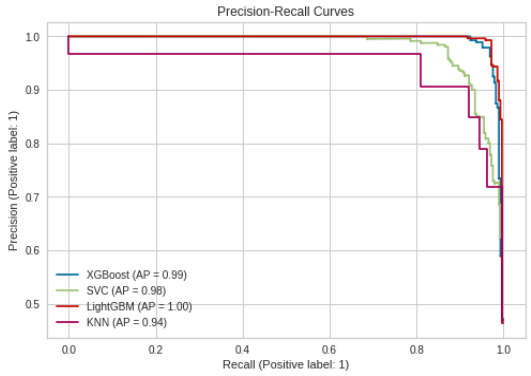

Fig. 6. Precision-Recall Curves

IV. Conclusion

The aim of this study was to develop a Customer Churn prediction model for the telecommunication industry. Several data preprocessing, feature encoding, feature selection, feature synthesis, feature transformation techniques were experimented. Furthermore, oversampling was employed and appropriate evaluation metrics were chosen to address the class imbalance by which the provided dataset was characterized. Many machine learning models were trained with different datasets (e.g.: - with and without SMOTE oversampling) and hypertuned using cross validation. Finally the model that gave the best performance proved to be hypertuned LightGBM which was trained on Min-max normalized and oversampled data with reduced features.

References

[1] M. Odusami, O. Abayomi-Alli, S. Misra, A. Abayomi-Alli, M. M. Sharma. (2021). A Hybrid Machine Learning Model for Predicting Customer Churn in the Telecommunication Industry. *Advances in Intelligent Systems and Computing*, vol. 1372. [Online] Available: https://doi.org/10.1007/978-3-030-73603-3_43

[2] N. N. Nguyen, A. T. Duong. (2021, Feb.). Comparison of two main approaches for handling imbalanced data in churn prediction problem. *Journal of Advances in Information Technology*, vol. 12, no. 1, pp. 29-35. [Online]. Available: https://doi.org/10.12720/JAIT.12.1.29-35

[3] S. Wu, W. Yau, T. Ong, S. Chong. (2021, Apr.). Integrated Churn Prediction and Customer Segmentation Framework for Telco Business. *IEEE Access*, vol. 9 [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3073776