Churn Prediction using Supervised Machine Learning Algorithms - Impact of Oversampling

[1]Mallika Naresh Panchal, [2]Dr. Anala A Pandit

[1]*Student, Master of Computer Application Veermata Jijabai Technological Institute, Mumbai, India.*
[2]*HOD, Masters of Computer Applications, Veermata Jijabai Technological Institute, Mumbai, India & IEEE, Senior Member, Mumbai, India.*

Abstract - *Telecommunication industries are growing rapidly, providing new and competitive services to the customers. This has led to a rapid increase in the churn rate in the telecommunication sector and has become a major problem for the telecommunication industries. Hence, customer churn prediction is crucial to predict the customers that are going to churn out and help plan retention campaigns to hold their existing customers. Customer data can be used to predict the customer behavior using machine learning algorithms and the telecommunication industries receive a lot of customer data every day. We perform supervised machine learning algorithms to predict customer churn along with taking into consideration the challenges that are faced during the development of the prediction model.*

Key Words: churn prediction, customer retention, telecommunication, machine learning, supervised algorithms, sampling, boosting

1. INTRODUCTION

The figure of mobile phone users has reached 6.8 billion and the telecommunication industry has advanced rapidly since last years and which is getting close to the world population [1]. That is, the telecommunication industry is expected to be saturated, i.e. there will not be many customers who are not taking service from anyone. If the customer is not with you, it is most likely to assume that the customer is with some other competitive company. Hence, recruiting customers who are not using the service at all, or taking away the customers from the competitors is very difficult and also very costly. Therefore, retaining the existing customers is very important in the telecommunication industry to sustain in this competitive market as well as increase their earnings to maximum.

There are two types of customers in telecommunication sector, post-paid and prepaid. Customers using the prepaid type of accounts have a larger probability of switching to other competitors as compared to the post-paid customers at any time when they are unsatisfied with their current company services. Such customers affect the overall reputation of the company and may result in disesteem [2]. As a result, several telecommunication industries are predicting customer churn using a wide range of machine learning techniques. In this paper, we put forward various supervised machine learning algorithms to predict churn. These models are evaluated using various performance metrics viz; the accuracy calculated from the rate of true positives and false positives, recall, precision and f-measure. To overcome the issue of imbalanced class, several techniques are adopted to enhance the accuracy of the model that will be used for prediction. The objective of this particular research is to develop a suitable model that will be used in predicting the customer data which will give insights for making strategic decisions for customer retention.

The model is validated on the data that is taken from IBM Sample data sets of CDR (Call Detail Record). Feature selection techniques such as correlogram matrix and chi-square test are implemented in order to acknowledge the most significant features from the total number of features available in the CDR. Before feature selection the SMOTE sampling technique is used to balance the data set. Ensemble technique such as the Random Forest, has given favorable prediction performance and hence used to predict churn customers at a very large scale [6][2]. C5.0 is another tree-based model which performs well for classification problem and is fast to implement. Another classification model is KNN classifier which is good due to its feature of finding distance between each point of train and test value, which can sometimes give good results. Other than that, we use Logistic Regression with RFE (Recursive Feature Elimination), XGBoost and LightGBM for better accuracy.

The rest of the paper is framed as: Part II provide the literature review. Part III states the proposed strategy for the implementation of predicting the churn customers. Part IV depicts the comparisons of the model performances along with visuals to get a quick insight on the comparisons. Finally, Part V includes the conclusion arrived from the implementations and the future work that can be adopted further.

2. LITERATURE REVIEW

2.1 Basic idea of churn prediction

The papers reviewed reflected the basic idea of churn prediction using machine learning. It consists of pre-processing the data, splitting the data into testing and training sets and then develop the models to get ideas and facts on the problem, i.e. churn prediction. The most common way to define the best suitable model is by calculating the false positives, true positives, false negatives, and true negatives from the test data, which will then be used to calculate the accuracy and other measures. Majorly used and considered to be the best models for churn prediction are decision tree-based models.

2.2 Challenges in churn prediction

Data for churn prediction is majorly skewed, since the number of churners are very less compared to the non-churners. The problem of over-fitting of the non-churners and under-fitting of our churners affects our model in a large way. For example, if we have a training data set where 90 percent of the customers are non-churners and 10 percent of the customers are churners and our model is not detecting any churners, i.e. a completely incorrect and is of no practical value. Such a model will also be 90 percent accurate. This is a major challenge that is faced during predicting churn. The solution to the class skew problem is oversampling and modern boosting methods.

2.3 Oversampling

Oversampling refers to the method of multiplying the minority class samples to an acceptable ratio to prevent the model from over fitting. Then, we can believe our model to be reliable to future predictions and observations without the model to correspond too closely to the data.

The basic method: This is a manual approach which is the most simple way to handle under sampling or oversampling in data sets. Here, to balance the imbalanced data in data sets, an amount of the samples of the minority class are replicated i.e. the minority class is over-sampled. Other way is to eliminate some amount of the samples of the majority class i.e. the majority class is under-sampled. Yet, if we adopt this method to under-sample our data, it has a downside since it lays by the possibly significant samples in the majority class and hence can cast down the classifier's performance. Besides, this method for oversampling do not deprave the classifier's performance, though it most often take additional time to train the classifier[4].

The advanced method: This method uses advanced approaches for sampling possibly enclosing statistical approach or data mining to either prune the samples or merge the under sampling and oversampling strategy. Various examples of exceptional techniques for managing class imbalance problems are SMOTE- "Synthetic Minority Over Sampling Technique", MTD-(F), ADASYN-"Adaptive Synthetic Sampling Approach", etc[4].

Random method: This method merges the majority class by randomly eliminating its samples, with the samples of the minority class or the minority class by randomly duplicating its samples, with the majority class [4].

2.4. Boosting

Boosting is a prevailing and an effective approach that seeks to "boost" the accuracy of every anonymous algorithm[6]. However, boosting is an algorithm that is uncontrollable. Maximum boosting algorithms engage iterative studying of the classifier, every time involving weak classifiers so that it can close in alongside to a determinate strong classifier. Every involved weak classifier is generally given a weight in conformity with the accuracy further training it using the re-weighted training data. Refer Fig. 1 to have a figurative illustration of boosting.

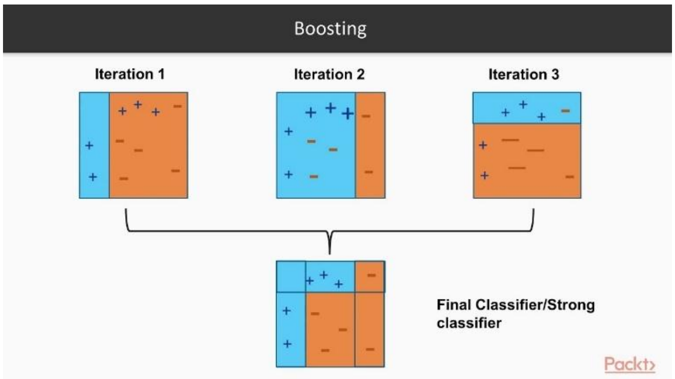
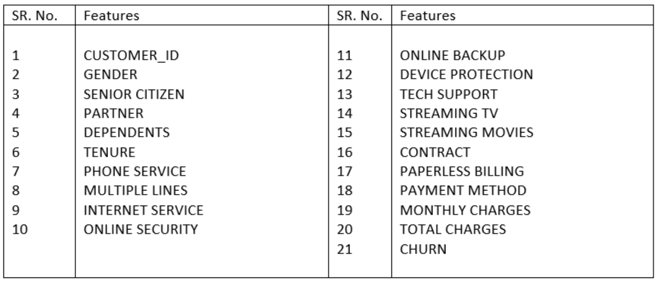


Fig -1: Boosting

3. PROPOSED MODEL FOR CHURN PREDICTION

3.1. Selecting data for churn prediction

The selection of a data set from a choice of separate or distinct implicitly applicable ones seem to be nothing but a numerical comparison between the essentials of the business with the practicable data quality enumerated in the metadata description. The data set that we chose for predicting was a simple data set from kaggle. Table 1 shows the features of the data set.



SR. No.	Features	SR. No.	Features
1	CUSTOMER_ID	11	ONLINE BACKUP
2	GENDER	12	DEVICE PROTECTION
3	SENIOR CITIZEN	13	TECH SUPPORT
4	PARTNER	14	STREAMING TV
5	DEPENDENTS	15	STREAMING MOVIES
6	TENURE	16	CONTRACT
7	PHONE SERVICE	17	PAPERLESS BILLING
8	MULTIPLE LINES	18	PAYMENT METHOD
9	INTERNET SERVICE	19	MONTHLY CHARGES
10	ONLINE SECURITY	20	TOTAL CHARGES
		21	CHURN

Table -1: Features

3.2. Oversampling - SMOTE

Considering our emphasis on making our data balanced in order to reduce false predictions we have performed oversampling using SMOTE (Synthetic Minority Oversampling Technique).

A fresh and new method "SMOTE - Synthetic Minority Oversampling Technique" was introduced by Chawla. The pseudo code of this algorithm and the details can be referred from [7]. The over-sampling method is a standard methodology to classify the imbalanced data [8]. This approach brings about artificial instances with leveraging the feature area instead of the data space. It is used to prevail the problem of over-fitting by amplifying the area of the resolution of the minority instances [4]. It can also be said that it creates artificial data rather than filled-in random over-sampling methods. Over-sampling was the first approach that established non-existent or replicated instances in the training data set so that the data space is accentuated along with battling the deficiency in the sample distribution [7]. Oversampling has acquired implausible effort in the examination of machine learning field over the last decade.

3.3. Model Development

Implementation of various models to predict churn on the data before and after oversampling the data, after splitting the data into test and train set. The train set includes the values in the churn column which is used to train the model and same model will be applied on test set, to test the quality of the model. Following are the different models used to predict churn with its implementation considering ensemble modeling and boosting methods for best results:

Random Forest: Random forest, Fig. 2, like the name indicates, conforms a substantial number of unique decision trees that emits a class prediction and administer as an ensemble. Finally, the model's prediction is decided to be the class with the maximum votes.

A great number of relevantly uncorrelated models is the reason that the random forest model strives so well i.e. the individual trees administering as a committee performs better than any of the individual component models.

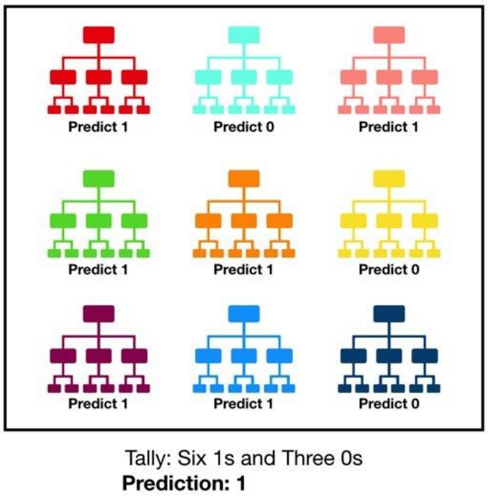


Fig -2: Random Forest

C5.0: As much as there are multiple implementations of decision trees, one of the most renowned decision tree algorithm is the "C5.0 algorithm". The C5.0 algorithm has grown to be the industry metric for generating decision trees, as it does well for majority of the kinds of challenges straight out of the box. In comparison to more advanced and complicated machine learning models, the C5.0 algorithm usually perform almost as well but are much lighter to understand and deploy.

KNN Classifier: "The k-nearest neighbors (KNN) algorithm" is an absolute and easy to implement machine learning algorithm. It makes an assumption that alike entities exists in closeness to each other.

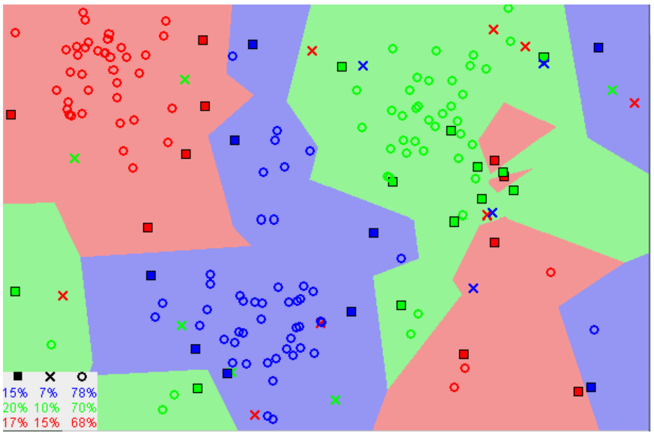


Fig -3: KNN Classifier

In the Fig. 3, majority of the time, the data points that are similar to each other are near each other. For this algorithm to be useful, it assumes this to be true. The KNN algorithm determines the proximity by calculating the distance between the points on the graph.

Logistic Regression + RFE: Logistic regression predicts the probability of a classification by calculating the relationship between one or more independent features with a dependent variable. Definitively, logistic regression predicts the chances of a data point that belongs to the default bracket.

Recursive Feature Elimination (RFE) iteratively builds a model by choosing the worst or best performing feature, digressing the feature further redo the process with the remaining features repeatedly until all the features in the data set are burned out. The goal of RFE is to select features by repeatedly considering diminutive batches of features.

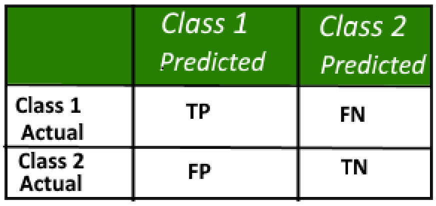
XGBoost Classifier: XGBoost is the implementation of the gradient boosted tree algorithms that is commonly used for classification and regression problems. Gradient boosting is an algorithm consisting of a group of weaker trees that add up their calculations to predict a target variable with better accuracy.

Tuning XGBoost Parameters: For bigger data sets, training time is greater and expensive too. Hence, it is significant to appreciate the behaviour of the parameters and focus on the phases that we anticipate to impact our results the most. We had to tune about 4 of the hyper parameters that are normally having a large impact on the performance.

LightGBM Classifier: Another ”gradient boosting framework” that uses decision tree based learning algorithms that is used for classification and ranking is LightGBM. Some advantages of LightGBM are speedy training and greater efficiency and accuracy. This algorithm can handle huge volume of data and uses lesser memory. LightGBM also abides parallel and GPU learning.

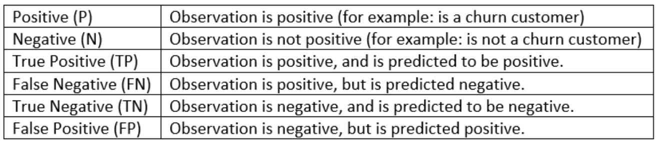
3.4. Model Evaluation

Confusion Matrix: Table 2 and 3, is a summary of prediction results of a classification model which defined as the confusion matrix. This matrix exhibits how after making predictions our classification model is confused. It gives perception into the errors and the types of errors of the classifier.



	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Table -2: Confusion Matrix



Positive (P)	Observation is positive (for example: is a churn customer)
Negative (N)	Observation is not positive (for example: is not a churn customer)
True Positive (TP)	Observation is positive, and is predicted to be positive.
False Negative (FN)	Observation is positive, but is predicted negative.
True Negative (TN)	Observation is negative, and is predicted to be negative.
False Positive (FP)	Observation is negative, but is predicted positive.

Table -3: Definition of terms

Measures of evaluation: In this research, the aimed model is evaluated with measures known and defined below. Equation 1 defines the accuracy measure. It is the number of samples that were classified correctly[2].

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

Recall is the ratio of the number of instances that were correctly classified and the total number of positive instances. If this value is high, we can know that the classification is done correctly since the number of instances that are incorrectly classified as negative is low. It is calculated by using Equation 2

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \tag{2}$$

Precision is derived by dividing the sum of positive instances that are classified correctly by the sum of all the positive instances that are predicted. If this value is high, we can assure that if the model is predicting an instance to be positive, that instance is believed to be positive since the number of instances that are incorrectly classified as positive is low. It is calculated by using Equation 3.

$$\text{Precision} = \frac{(TP)}{(TP + FP)} \tag{3}$$

High recall with low precision means that majority of the positive instances are correctly recognized (low FN) but there are a lot of false positives. Low recall with high precision means that we missed a lot of positive instances (high FN) but the ones that we predicted as positive are indeed positive (low FP).

F-measure helps to have a measurement that represents Recall as well Precision. F-measure is calculated using the Harmonic Mean instead of the Arithmetic Mean. F-measure can be calculated using Equation 4

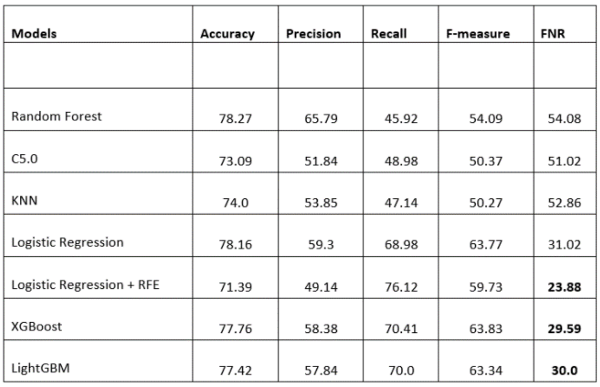
$$F - \text{measure} = \frac{(2 * \text{Precision} * \text{Recall})}{(P\ \text{recision} + \text{Recall})} \tag{4}$$

The model with the highest accuracy or least false negative ratio can be considered for predicting future churn customers.

$$\mathrm{FNR} = \frac{(F\,N)}{(F\,N\ +\ T\,P)} \qquad (5)$$

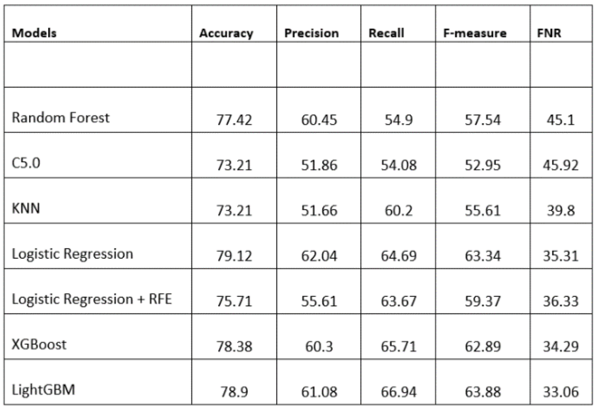
4. COMPARING MODEL PERFORMANCES

We developed model using Random Forest, C5.0, KNN Classifier, Logistic Regression, Logistic Regression with Recursive Forward Elimination, XGBoost and LightGBM on the data set before and after oversampling. Table 4 shows the measures of each models before oversampling. Table 5 shows the measures of each models after oversampling. The minority class was over sampled to the ratio of 70-30 percent with the majority class.



Models	Accuracy	Precision	Recall	F-measure	FNR
Random Forest	78.27	65.79	45.92	54.09	54.08
C5.0	73.09	51.84	48.98	50.37	51.02
KNN	74.0	53.85	47.14	50.27	52.86
Logistic Regression	78.16	59.3	68.98	63.77	31.02
Logistic Regression + RFE	71.39	49.14	76.12	59.73	**23.88**
XGBoost	77.76	58.38	70.41	63.83	**29.59**
LightGBM	77.42	57.84	70.0	63.34	**30.0**

Table -4: Measures of all the implemented models before oversampling



Models	Accuracy	Precision	Recall	F-measure	FNR
Random Forest	77.42	60.45	54.9	57.54	45.1
C5.0	73.21	51.86	54.08	52.95	45.92
KNN	73.21	51.66	60.2	55.61	39.8
Logistic Regression	79.12	62.04	64.69	63.34	35.31
Logistic Regression + RFE	75.71	55.61	63.67	59.37	36.33
XGBoost	78.38	60.3	65.71	62.89	34.29
LightGBM	78.9	61.08	66.94	63.88	33.06

Table -5: Measures of all the implemented models after oversampling

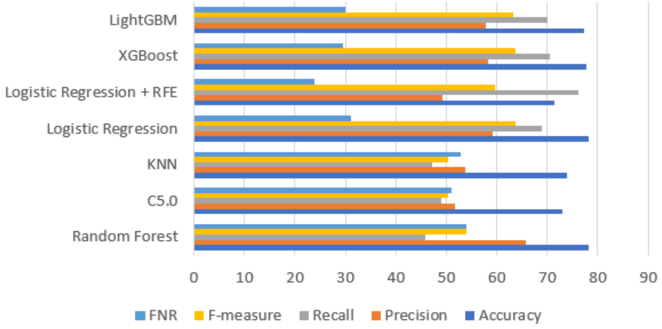


Chart -1: Measures of all the implemented models before oversampling

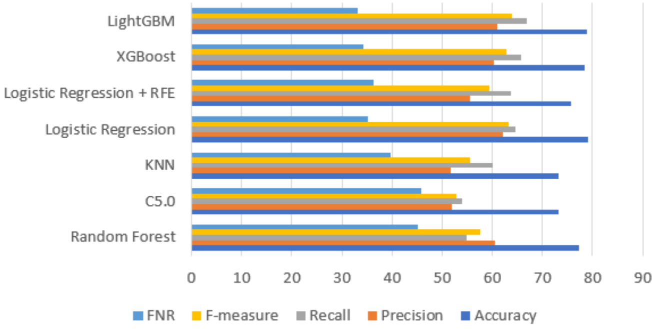


Chart -2: Measures of all the implemented models after oversampling

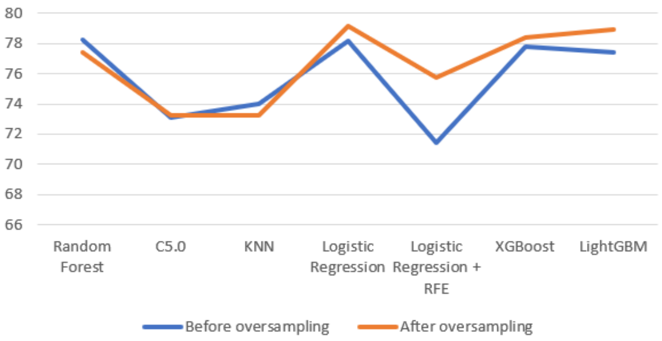


Chart -3: Accuracy before and after oversampling

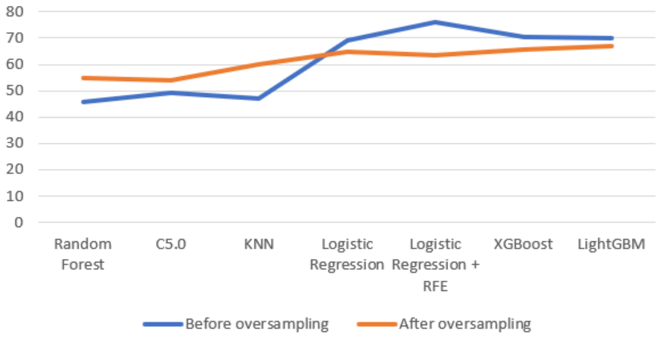


Chart -4: Recall before and after oversampling

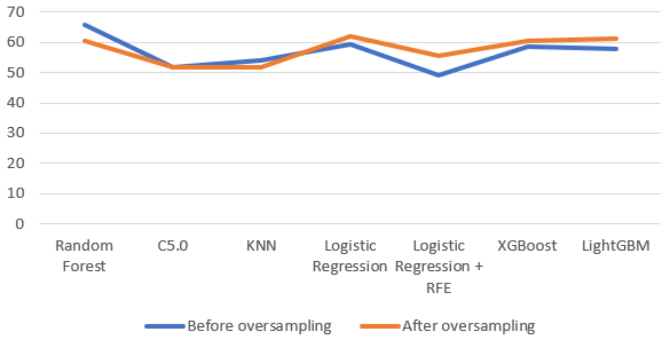


Chart -5: Precision before and after oversampling

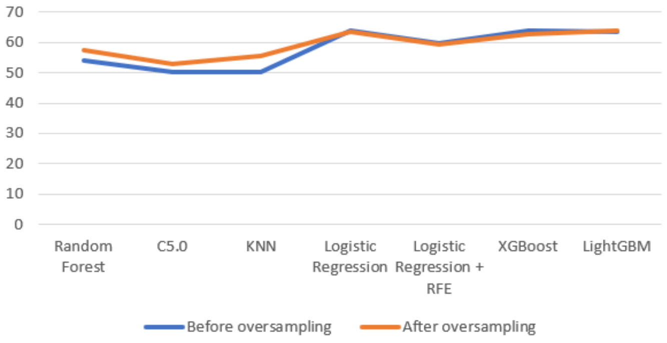


Chart -6: F-measure before and after oversampling

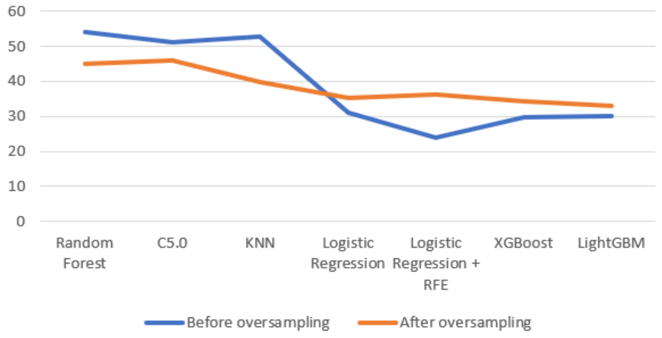


Chart -7: FNR before and after oversampling

Boosting methods and RFE has the lowest FNR value. Random Forest and Logistic Regression have good accuracy although in an imbalance data set it is recommended to rely on a lower FNR rather than a higher accuracy.

5. FUTURE SCOPE AND CONCLUSION

In this research, we are handling an investigation by observing the prediction of customer churn based on real data set where we will know the customers with the highest probability of churning out and respective retention actions can be planned for them. This means that we can only prioritize of customers using the most accurate churn prediction model. However we would also want to identify the customer's need based on the behaviour of his/her churn which are lacked by our organisation and the reason of his/her churn.

In this study, we are calculating various performance metrics and concluding the best customer churn model on our data set. The evaluation matrix performances can help determine which model is the most accurate for our given data set. Keeping in mind the problem of the class skew, we used advanced sampling technique such as SMOTE as well as other modern techniques such as boosting,. Analysing the performance measures of each of our models, we can conclude LightGBM to be the best performing model for the given data set with low false negative ratio and high accuracy on over sampled data.

REFERENCES

[1] ITU-ICT-2014(2014) The World in 2014 ICT Facts and Figures. Union, I. T., Geneva, Switzerland.

[2] Irfan Ullah, Basit Raza, Ahmad Kamran Malik, Muhammad Imran, Saif Ul Islam, and Sung Won Kim (2019), "A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector", Digital Object Identifier, 10.1109/ACCESS.2019.2914999.

[3] Ning Lu, Hua Lin, Jie Lu, Member, IEEE, and Guangquan Zhang (2014), "A Customer Churn Prediction Model in Telecom Industry Using Boosting", IEEE Transactions on industrial informatics, Vol.10, No.2.

[4] Adnan Amin, Sajid Anwar, Awais Adnan, Muhammad Nawah, Newton Howard, Junaid Qadir, (Senior Member, IEEE), Ahmad Hawalah, and Amir Hussain, (Senior Member, IEEE) (2016), "Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study", Digital Object Identifier 10.1109/ACCESS.2016.2619719.

[5] Adnan Idris and Asifullah Khan (2016), "Churn Prediction System for Telecom using Filter–Wrapper and Ensemble Classication", The Computer Journal.

[6] Y. Freund and R. E. Schapire, "A short introduction to boosting," J. Jpn. Soc. Artif. Intell., vol. 14, no. 5, pp. 771–780, Sep. 1999.

[7] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in Data Mining and Knowledge Discovery Handbook. Springer, 2005, pp. 853–867.

[8] J. Wang, M. Xu, H. Wang, and J. Zhang, "Classification of imbalanced data by using the smote algorithm and locally linear embedding," in Proc. 8th Int. Conf. Signal Process., vol. 3. 2006, pp. 1–4.