CSE/ECE 343: Machine Learning Final Project Report

Predicting and Analysing Customer Attrition

Ishit Bajpai 2020380 | **Keshav Rajput** 2020308 | **Prachi** 2020098 | **Satyam Arora** 2020330

December 4, 2022

1 Abstract

Customer churn is a major problem and one of the most important concerns for large companies. Due to the direct effect on the companies' revenues, especially in the telecom field, companies seek to develop means to predict potential customer churn. We first define the problem of customer churn rigorously for replicability. We then perform various data pre-processing techniques like data cleaning, imputing null values, encoding categorical data, selecting appropriate features using random forest, standardizing and normalizing the dataset, performing dimensionality reduction, and splitting the dataset into testing and training sets. We then implement various machine learning algorithms like logistic regression, naive Bayes, SVM, Random Forest, AdaBoost, XGBoost, and ANN. We end with a discussion of which algorithms performed best and the reason for being the best performing algorithm and also comment on the future work that can be conducted for the same problem.
Git link

2 Introduction

Customer Attrition is defined as the loss of customers by a business.
Predicting the churn/attrition rate is very useful because a low attrition rate implies customers are happy with the service provided. In contrast, a high attrition rate implies customer dissatisfaction with the service, causing them to leave. High attrition is a clear sign of losing a foothold and rivals rising as the customers who leave would mostly turn up to some other service provider.
According to Forbes, it costs 5 times more to acquire a new customer than it does to retain one, which results in Companies losing about $1.6 trillion per year due to customer attrition. Moreover, the Harvard Business School report claims that on average, a 5% increase in customer retention rates results in a 25% – 95% increase in profits.
We aim to analyze the factors on which attrition depends, along with predicting customer attrition in order to sustain the company.

3 Literature Survey

1. [1]Model developed in this work uses machine learning techniques on big data platforms and builds a new way of feature engineering and selection. In order to measure the performance of the model, the Area Under Curve (AUC) standard measure is adopted, and the AUC value obtained is 93.3%. Another main contribution is using customer social networks in the prediction model by extracting Social Network Analysis (SNA) features. The use of SNA enhanced the model's performance from 84 to 93.3% against the AUC standard. The model experimented with four algorithms: Decision Tree, Random Forest, Gradient Boosted Machine Tree "GBM," and Extreme Gradient Boosting "XGBoost." The best results were obtained by applying the XGBoost algorithm.

2. [4]Current machine learning models are of 2 types, Classification(good for small scale, uncertain at large scale) AI based(more complex, low generalization ability). This paper uses SVM on structural risk minimization to improve the prediction. On comparing, it proved to be the best in accuracy rate, hit rate,

covering rate, and lift coefficient. 2 datasets are used (the UCI database of the University of California and home telecommunication carry. Results using the SVM model are superior to the other methods except that ANN in the hit rate and decision tree C4.5 in the coverage rate. Final Outcome Analysis: Customers with big churn rates have the following traits: strong charge willingness, long mobile service, and considerable customer care. With large scale, high dimension, non-linearity, non-normality, time series, and rare class, SVM can also be developed using similar customer churn data in the banking industries.

3. [5]In this paper, a novel learning method called improved balanced random forests (IBRF) is used. Using 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. In it, the best features are iteratively learned by altering the class distribution and by putting higher penalties for the misclassification of the minority class. It improves prediction accuracy significantly compared with other algorithms, such as artificial neural networks, decision trees, and class-weighted core support vector machines (CWC-SVM). Moreover, IBRF also produces better prediction results than other random forest algorithms, such as balanced random forests and weighted random forests.

4 Dataset

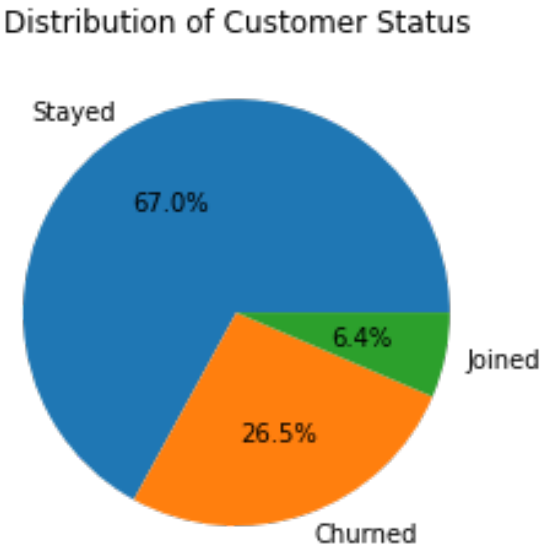
4.1 Data Source

The public dataset is completely available on the Maven Analytics website platform, where it stores and consolidates all available datasets for analysis in the Data Playground. The specific telecom customer churn dataset at hand can be obtained in this link below: [2] After downloading the dataset, we can access the telecom_customer_churn.csv file.

4.2 Data Description

This dataset contains two tables in CSV format: The Customer Churn table contains information on all 7,043 customers from a Telecommunications company in California in Q2 2022 Each record represents one customer and contains details about their demographics, location, tenure, subscription services, payment information, and status for the quarter (joined, stayed, or churned). Data has 7043 rows and 38 columns(23 categorical and 15 numerical) The target Variable is Customer Status 14 columns have null values with missing value percentages ranging from 10-70% 61% of Categorical variables have 2 categories and 87% of Categorical variables have less than 5 categories(excluding nan) 60% of Numeric variables have more than 1000 unique values. The number of outliers is very less(¡5)% when we compare class-wise, so we have opted not to remove them also since churn is a rare class, excluding outliers may not help with training the model.12

Class Distribution



4.3 Data Preprocessing

1. Data Cleaning, this involved the removal of the Customer ID column(index column).
2. Majority of the columns had 20% NULL values. Missing Values are filled using KNN imputer(categorical) and Mean(numerical).
3. Encoding Categorical Data: Label encoding was performed for all categorical features as well as the class we are going to try to predict(Customer_Status)
4. Feature Selection here, we employed three methods. First, we checked if the absolute

pairwise correlation of any two features was greater than 0.85; if yes, one of the features was removed. Numerical Features with less Variance threshold of less than 0.05 were removed because they contribute very little to the overall explainability of the model. Finally, we used hypothesis testing using the Chi-Square method to get all the relevant features in the dataset.

5. Standardization and Normalization, the dataset was normalized using sklearn's implementation called Standard Scaler. It normalizes and standardizes the data, which helps improve model accuracy for certain models like logistic regression, Naive Bayes, SVM, and ANN.

6. Dimensionality Reduction, we performed Principal Component Analysis to reduce some of the dimensions down to 8(from 31) since it helps reduce the time it takes for models like logistic regression to converge. We determined 8 to be the best using the trial and error method.

7. Modifying the target variable customer status by combining "not churned" and "joined" to "stayed".

7. Splitting the dataset into training and testing datasets of size 75:25, respectively.

8. Separability of data, we performed TSNE to check whether the classes are separable or not. The data is not completely separable but might be separable in sufficiently larger dimensions.

TSNE

5 Methodology

Predicting customer churn is a binary classification problem. We are going to be implementing six algorithms.

1. Logistic Regression using the default values as given by sklearn's implementation.

2. Naive Bayes using Gaussian and Bernoulli apriori distributions with default values vai sklearn's implementation.

3. Support Vector Machines using linear, radial basis, polynomial, and sigmoid kernels.

4. Random Forest using sklearn's implementation with the following parameters, estimators = 20, depth = 10, and criteria = gini. We obtained these parameters using a grid search. The grid includes estimators from 2 to 20, depth from 5 to 20, and criteria such as gini and entropy.

5. Boosting Techniques on random forests such as AdaBoost and XGBoost. We again used the sklearn's implementation and used the same grid for searching for optimal parameters as we did for the random forest. The parameters for AdaBoost were estimator = 20, depth = 10, criteria = gini, and for XGBoost, we had 20 estimators each with a depth of 8 and the objective(criterion) as log-loss(gini).

6 . Artificial Neural Networks, the parameters were found using grid search, the activation function 'relu,' learning rate as adaptive, learning rate initialized using 0.001 and having solver as 'adam.' The hidden layer sizes were (256,32). We determined this to be the best layer size because we started with two hidden layers, increased the layer size in each of them one by one, and trained them for a small number of epochs to determine which one was better. The number of epochs was 100.

The K-fold cross-validation method was used for calculating all the metrics with five folds. The valuation metrics are the commonly used ones in classification problems like accuracy score,f1, and recall. The accuracy score is a good indicator of the true positive rate.

$$Accuracy = \frac{TP}{TP + FP}$$

Recall tells us the proportion of true positives identified correctly, mathematically it is given

by

$$Recall = \frac{TP}{TP + FN}$$

The F1 score is a good accuracy score when the given data is unbalanced and can help us detect whether the rare class is being recognized. Mathematically it is given by

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Finally, we plot ROC-AUC curves to determine how our classifiers are performing. ROC-AUC plots true positive rate vs. false positive rate; if the graph is skewed to the top right corner, it is said to be a good classifier. The best case is when it attains a horizontal line at y = 1. We have also computed the ROC-AUC scores for completeness and accuracy in assessing our models.

6 Results and analysis

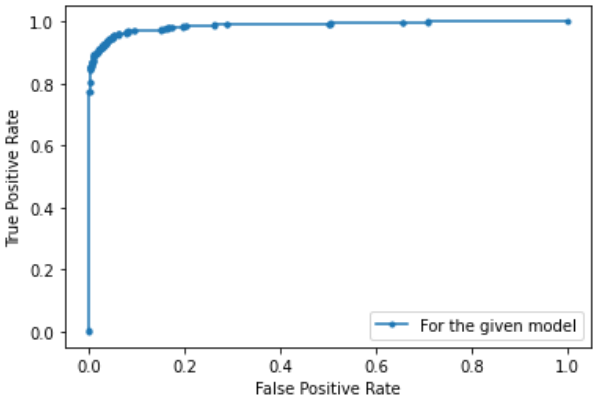
1. Logistic Regression is a good algorithm for binary classification, which is exactly what we are trying to do. As a result, we obtain a very nice accuracy of 95.7% for it. The model can recognize the rare class that we are trying to predict. As it also boasts a good f1 score.

2. Naive Bayes, we implemented it using the Gaussian and Bernoulli as our apriori distributions. Since we normalized and scaled the dataset, therefore, we expect the dataset to "inherit" some Gaussian traits. As a result, we get a higher accuracy score of 93.16% on Gaussian vs. 87.03% on Bernoulli. Another thing to note is that we get a very low recall score for the Bernoulli apriori. This means that the model cannot recognize the rare class correctly.

3. Support Vector Machine is a good classifier, especially if one chooses the right kernel. Since we went with an infidimensional kernel(radial basis function), the dataset would become separable even if it is not linearly separable in the current dimensions. As a result, we get the highest accuracy score and highest f1 score of 96.72% and 93.515%, respectively. The result also seems to be in line with our literature review. In particular, papers one and three report great results using SVM with the RBF kernel. The model is easily able to recognize the rare class.

Kernel	Accuracy Score	Precision	Recall	F1 score
Linear	96.38	96.8	89.30	92.89
rbf	96.72	98.42	89.08	93.51
poly	85.83	91.18	51.64	65.91
sigmoid	91.84	85.52	83.81	84.50

Summary of SVM kernels

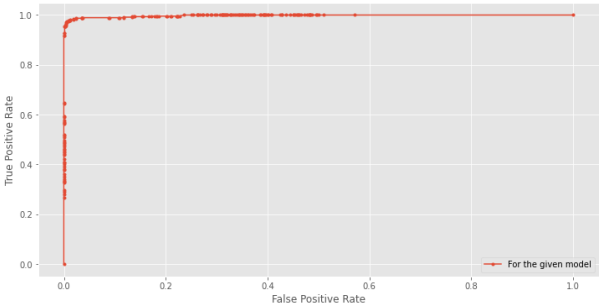


ROC-AUC curve for SVM(rbf)

4. Random Forest is one of the best algorithms in machine learning, with the only downside being that it easily overfits. We can also see this in our results, where the model boasts a high accuracy score of 94.51% but has a comparatively lower recall score of just 83.95%. The model over-fitted the dominant class and didn't properly account for the rare class.

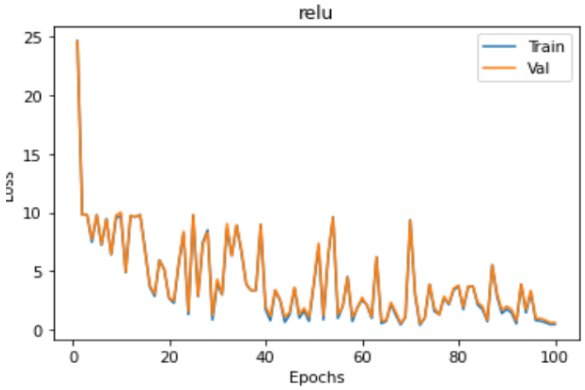
5. Boosting Techniques, here we employed AdaBoost and XGBoost techniques to increase the accuracy That we obtained from our random forest classifier. Random Forests work by bagging different decision trees together(ensemble them). AdaBoost simply assigns a higher weight to weak learners while bagging. XGBoost is a relatively new algorithm that assigns a loss function to each tree and then uses gradient descent to minimize that loss function. XGBoost is slightly more effective than AdaBoost. This theoretical result is consistent with the results obtained. AdaBoost achieved an accuracy

score of 95.23% and XGBoost achieved an accuracy score of 95.47%. The F1 score on XGBoost is also higher because it introduces a regularization parameter that reduces the variance, thus disallowing the model from over-fitting, making it more sensitive to the rare class. The model also boasts the highest ROC-AUC score out of all the models. It has a ROC-AUC score of 99.8% which means it is a very good classifier.



ROC-AUC curve for XGBoost

6. Artificial Neural Network provides us with a universal algorithm that can, in theory, solve any problem. In our case, we use the multi-layer perceptron implementation as churn prediction is a classification problem. After performing a grid search, an accuracy score of 94.38% was obtained on the testing set. The model also boasted the highest recall score out of the other models. This means it is very good at identifying true positives. The loss curve is plotted below for the relu activation function. Relu is considered to be the best activation function because it provides a good mix of linearity and non-linearity, which makes it easy for the model to make the decision boundary for the dataset.



Loss curve for Relu ANN

Model Name	Accuracy Score	Precision	Recall	F1	ROC-AUC Score
Logistic	95.79	94.84	89.01	91.82	98.1
Naive Bayes(Gaussian)	93.16	97.18	76.4	85.56	97.3
Naive Bayes(Bernoulli)	87.03	85.91	61.20	71.44	93.4
SVM(kernel=rbf)	96.72	98.42	89.08	93.515	98.8
Random forest (estimator=20,depth=10,criteria=gini)	94.51	95.63	83.95	89.00	98.8
AdaBoost(estimator=20,depth=10,criteria=gini)	95.23	94.20	87.45	90.31	99.7
XGboost(estimator=20,dept=8,objective='binary:logistic)	95.47	94.34	88.23	91.11	99.8
ANN	94.38	80.48	99.66	89.05	97.8

Summary of results

Overall Many of the models got high accuracy, but the most useful will be the ones with higher f1 scores as we are interested in correctly identifying the rare class. Overall, SVM performed the best, so it can be termed state of the art for this report. SVM performed the best because the data is linearly separable, as evidenced by the TSNE algorithm.

7 Conclusion

The results of the final report are much more sound than the midterm report. This is because of faulty feature selection that was done before the midterm. This time around, this was not the case. Performing secondary tests with Chi-Square and variance threshold increased the model's explanatory power and removed the redundant features. As a result, the accuracy of the various results aligns pretty well with the theory behind those algorithms. For example, we did expect SVM to perform the best due to the theory discussed in the literature and the classroom. The lecture slides, for

example, share that one of the applications of SVM is churn prediction. We also get sound results from Multilayer Perceptron(ANN), Logistic Regression and Naive Bayes, and Random Forest. The precision and recall results for SVM indicate that it is a very strong classifier that can predict churn rate at a consistent level. Although, since churn is a rare class, even a tiny 1% difference can significantly impact the company's overall profitability. Therefore, there is a need for algorithms that are correct 99% so that the companies can correctly target customers that are about to unsubscribe from their service.

What we learned: We all had to learn many new techniques before they were taught in class formally. As a result, sometimes we would implement the idea/concept/technique incorrectly, but by the time support vector machines were taught in the classroom, we all had learned the majority of the work that we needed to learn. Most of the time was spent pre-processing and performing exploratory data analysis on feature selection and reiteratively calculating the results of each model again.

Comments on the timeline. Most of the work was completed before the midterm interim report, and we also learned what went wrong with feature selection. So it was easier to pick things back up and finish the work fast. The final code-related work and parameter tuning were finished a week before the deadline.

8 Contribution

All the co-authors contributed equally.

References

[1] Abdelrahim Kasem Ahmad, Assef Jafar, and Kadan Aljoumaa. Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*, 6(1):1–24, 2019. 1

[2] Enrique Ruiz. Maven churn challenge. *Maven Analytics*, 1(1):1, 2022. 2

[3] A Saran Kumar and D Chandrakala. A survey on customer churn prediction using machine learning techniques. *International Journal of Computer Applications*, 975:8887, 2016. 6

[4] JIN Wei-dong XIA Guo-en. Model of customer churn prediction on support vector machine. *ScienceDirect*, 1(28):71–77, 2008. 1

[5] E.W.T. Ngai Weiyun Ying Yaya Xie, Xiu Li. Customer churn prediction using improved balanced random forests. *Computing*, 1(36):5445–5449, 2009. 2

[3]