DSC540 Advance Machine Learning

A comparison of machine learning techniques for bank customer churn prediction

Lavinia Wang
1473704

Abstract

Customer churn is a universal problem and the customer churn dataset is seriously imbalanced. In order to improve the prediction accuracy of churn customers as well as strengthen to identify non-churn customers, this paper presents a comparative study on the most popular machine learning methods applied to the challenging problem of customer churning prediction on bank customer data. In the first phase of analysis, five different under-sampling methods were applied to the original dataset for rebalance. In the second phase, all models were applied and evaluated using test data (35% of under-sampling data). In order to determine the generalization of the model, 10-fold cross validation was performed on the original dataset and feature selection was also performed for most efficient parameter combinations. The results demonstrate clear superiority of the boosted versions of the models against the plain (non-boosted) versions. The best overall classifier was the Random Forest using entropuy with accuracy of almost 85% and AUC 84%.

1. Introduction

Customer churn refers to when a customer (player, subscriber, user, etc.) ceases his or her relationship with a company. Online businesses typically treat a customer as churned once a particular amount of time has elapsed since the customer's last interaction with the site or service. The full cost of customer churn includes both lost revenue and the marketing costs involved with replacing those customers with new ones. Reducing customer churn is a key business goal of every online business.

The ability to predict that a particular customer is at a high risk of churning, while there is still time to do something about it, represents a huge additional potential revenue source for every online business. Besides the direct loss of revenue that results from a customer abandoning the business, the costs of initially acquiring that customer may not have already been covered by the customer's spending to date. (In other words, acquiring that customer may have actually been a losing investment.) Furthermore, it is always more difficult and expensive to acquire a new customer than it is to retain a current paying customer.

The dataset used in this project is from Kaggle.com(https://www.kaggle.com/shrutimechlearn/churn-modelling), which contains details of 10,000 bank's customers and the target variable is a binary variable reflecting the fact whether the customer left the bank (closed his account) or he continues to be a customer.

2. Literature Review

Such churn prediction has gained much attention from both academic research and industry practice. In e-commerce area, researchers (Yu, Guo & Huang, 2011) proposed extended support vector machine (ESVM) by introducing parameters to tell the impact of churner, non-churner and nonlinear. Artificial neural network (ANN), decision tree, SVM and ESVM were utilized as alternative predication algorithms to forecast customer churn with the innovative framework. Result shows that ESVM performs best among them in the aspect of accuracy, hit rate, coverage rate, lift coefficient and treatment time. This novel ESVM can process large scale and imbalanced data effectively based on the framework.

E-commerce customer churn rate is high, and the customer churn dataset is seriously imbalanced. In order to improve the prediction accuracy of churn customers as well as strengthen to identify non-churn customers, another paper presents churn prediction model based on improved SMOTE and AdaBoost (Wu & Meng, 2016).

In telecommunications industry, similar customer churn prediction has been studied and similar algorithms were presented by scholars. One paper mainly focused on bagging and stochastic gradient boosting approach (Lemmens & Croux, 2006) which significantly improve accuracy in predicting churn. Another paper presented using SVM- Poly with AdaBoost (Vafeiadis, Diamantaras, Sarigiannidis, Chatzisavvas, 2015) achieved accuracy of almost 97% and F-measure over 84%.

3. Methodology

In the first phase of analysis, five rebalancing approaches are introduced and compared. In the second phase after preprocessing, including resampling, train and test split and normalization, seven classification algorithms are utilized to build training and testing models. The first comparison will be conducted on all models. Next, the optimal models will be applied back on the original dataset in terms of generalization test using 10- fold cross validation. Measurement of performance are accuracy and ROC-AUC scores.

3.1 Preprocessing Rebalance Methods

Oftentimes in practical machine learning problems there will be significant differences in the rarity of different classes of data being predicted. For example, there are less customer churned than those who remain with the company. The overall performance of any model trained on such data will be constrained by its ability to predict rare points. In problems where these rare points are only equally important or perhaps less important than non-rare points, this constraint may only become significant in the later "tuning" stages of building the model. But in problems where the rare points are important, or even the point of the classifier, dealing with their scarcity is a first-order concern for the model builder.

Tangentially, note that the relative importance of performance on rare observations should inform your choice of error metric for the problem to work on; the more important they are, the more your metric should penalize underperformance on them. Several different techniques exist in the practice for dealing with imbalanced dataset. The naivest class of techniques is sampling:

changing the data presented to the model by undersampling common classes, oversampling (duplicating) rare classes, or both. In extreme cases where the number of observations in the rare class(es) is really small, oversampling is better, as you will not lose important information on the distribution of the other classes in the dataset. But it is also biased as the modeler believe the distribution of current data would correctly reflect the rare class(es). Undersampling doesn't introduce new information in the dataset, it (hopefully) merely shifts it around so as to increase the "numerical stability" of the resulting models. The ratio of retained customer vs churned customer is roughly 4:1(*figure 1*). So, it is more appropriate to apply undersampling methods on the bank dataset.

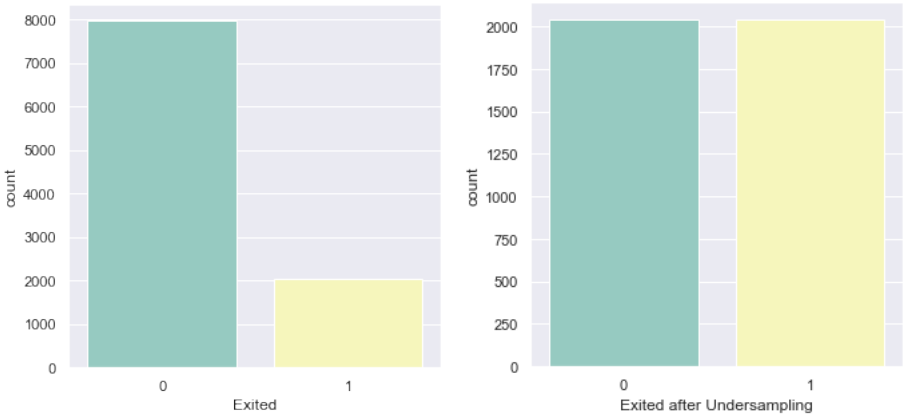


Figure 1 Comparison of unbalanced vs undersampling

3.1.1 Random under-sampling for the majority class

A simple under-sampling technique is to under-sample the majority class randomly and uniformly. This can potentially lead to loss of information. But if the examples of the majority class are near to others, this method might yield good results.

3.1.2 ClusterCentroids

This method undersamples the majority class by replacing a cluster of majority samples This method finds the clusters of majority class with K-mean algorithms. Then it keeps the cluster centroids of the N clusters as the new majority samples.

3.1.3 TomekLinks

In the same manner, Tomek (1976) proposed an effective method that considers samples near the borderline. Given two instances a and b belonging to different classes and are separated by a distance d(a,b), the pair (a, b) is called a Tomek link if there is no instance c such that $d(a,c) < d(a,b)$ or $d(b,c) < d(a,b)$. Instances participating in Tomek links are either borderline or noise so both are removed.

3.1.4 NeighbourhoodCleaningRule

Neighborhood Cleaning Rule (NCL) deals with the majority and minority samples separately when sampling the data sets. NCL uses ENN to remove majority examples. for each instance in the training set, it finds three nearest neighbors. If the instance belongs to the majority class and the classification given by its three nearest neighbors is the opposite of the class of the chosen instance, then the chosen instance is removed. If the chosen instance belongs to the minority class and is misclassified by its three nearest neighbors, then the nearest neighbors that belong to the majority class are removed.

3.1.5 NearMiss

In order to attack the issue of potential information loss, "near neighbor" method and its variations have been proposed. The basic algorithms of the near neighbor family are this: first, the method calculates the distances between all instances of the majority class and the instances of the minority class. Then k instances of the majority class that have the smallest distances to those in the minority class are selected. If there are n instances in the minority class, the "nearest" method will result in k*n instances of the majority class.

A simulation of original data, the make_classification function generates the repeated (useless) features from the informative and the redundant features. The redundant features are simply the linear combinations of the informative features. Each class has consisted of 2 gaussian clusters. For each cluster, informative features are drawn independently from N (0, 1) and then linearly combined together within each cluster. It is important to know if the parameter weights are left blank, then classes are balanced. RandomUnderSampler shows the best resampling result (*figure 2*) where TomekLinks and NeighborhoofCleaningRule failed to rebalance the data. After resampling, there are 4074 total samples, which are later split in to training and testing after normalization at 65% and 35%.

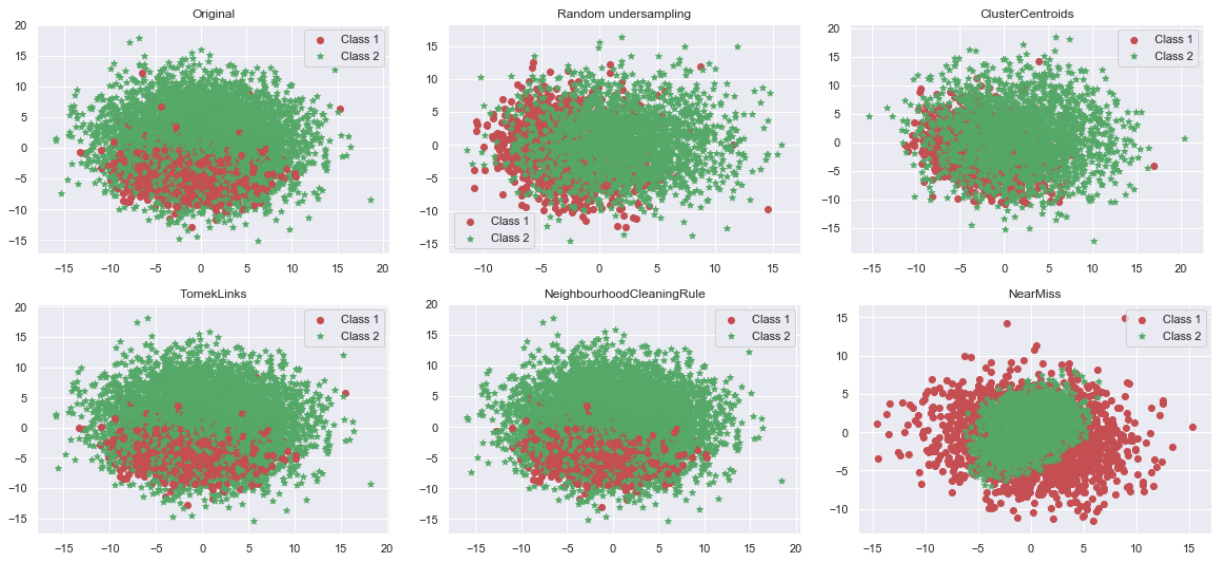


Figure 2 Comparison of undersampling methods

3.2 Classification Methods

To predict customer churn, many scholars have mainly focused on the following two approaches: the first is the traditional classification methods, including decision tree (Wei & Chiu, 2002), logistic regression (Kim & Yoon, 2004), etc. The methods can be used to analyze qualitative data, continuous data. However, it cannot guarantee the accuracy and generalization ability of the constructed models for large scale, nonlinearity and high dimensionality; the second is the artificial intelligence methods, including ANN (Yan, Miller, Mozer, et al., 2001), evolutionary leaning (Au, Chen, & Yao, 2003), etc. The methods can overcome the above defects and have nonlinear mapping ability, strong robustness and good prediction precision (Yan et al., 2001). However, the methods based on empirical risk minimization always lead to low generalization ability and fuzzy construction of the models (Vapnik, 2004). So, above methods have been some limitations in real application.

To solve a problem in a different but similar industry, a comparison of traditional statistical approach and popular machine learning algorithms were proposed in this paper. Seven well established and popular techniques used for churn prediction, taking into consideration reliability, efficiency and popularity in the research community. Logistic regression model would be built as a baseline model for bank churn prediction. The rest algorithms are SVM, Naïve Bays, Decision Tree, Random Forest, Gradient Boost and Ada Boost.

4. Results

4.1 Exploratory Data Analysis

There are 14 variables in the original dataset, 4 of which are considered redundant and got removed. 9 feature variables kept and 1 target variable with binary class (0,1). Continuous variables showed a wide range of varies so normalization was applied to make sure all variables have the same distribution and is easier to compare (*table 1*).

I also compared categorical variables with the target. There is no obvious correlation between tenure and exited. Number of products and exited seems to be inversely correlated, which suggests the more products customer have with the bank, the more loyal the customer is and is less likely to churn. So is active member. Having a credit card has a positive relationship with exit, which could be interpreted as having a credit card would increase the probability of customer churn (see *figure 3*).

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive Member	EstimatedSalary	Exited
Count	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
Mean	650.52	0.54	38.92	5.01	76485.88	1.53	0.70	0.51	100090.23	0.20
Std	96.65	0.49	10.48	2.89	62397.40	0.58	0.45	0.49	57510.49	0.40
Min	350.00	0.00	18.00	0.00	0.00	1.00	0.00	0.00	11.58	0.00
Max	850.00	1.00	92.00	10.00	250898.09	4.00	1.00	1.00	199992.48	1.00

Table 1. 5 Summary of kept variables

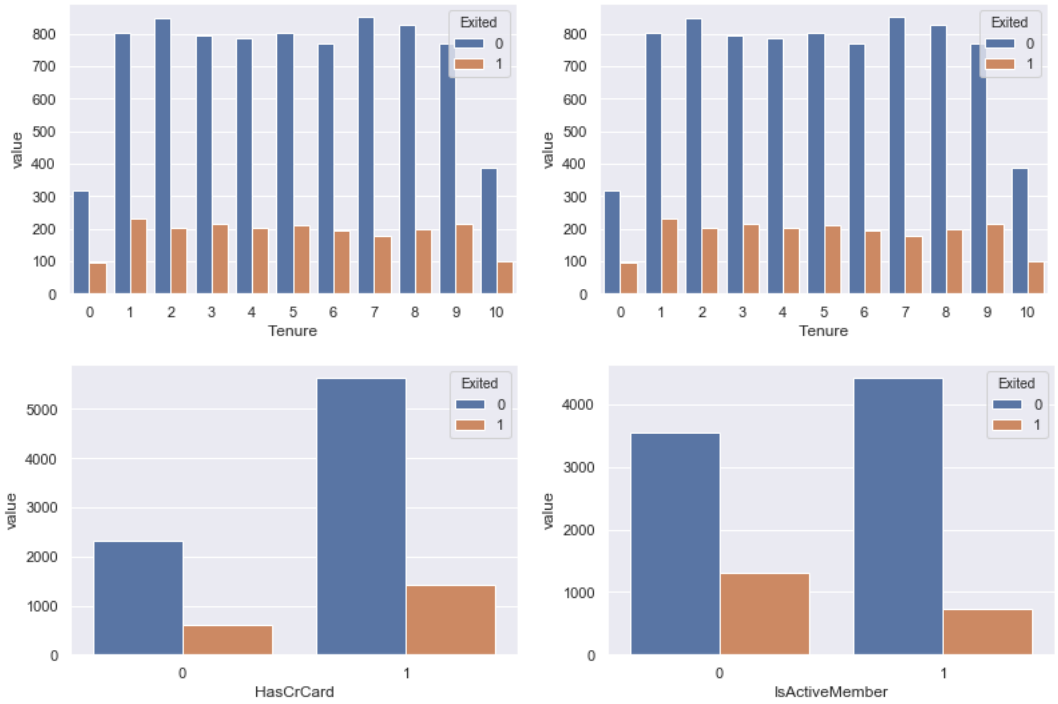


Figure 3 Crosstab of categorical features with target variable

4.2 Model outputs

A total of 11 models were built in the first round applying seven algorithms. The baseline model using Logistic Regression (LR) achieved accuracy of 0.69 and AUC 0.77 on the test data (0.35 of undersampled data)(*table 2*). Support Vector Machine models were applying linear, poly, RBF and sigmoid kernels respectively. SVM-Linear kernel achieved a similar performance as LR, which is accuracy of 0.71 and AUC 0.77. SVM-poly kernel improved both accuracy and AUC to 0.74 and 0.82 respectively. SVM-RBF kernel also increased the performance and was similar as SVM-poly with accuracy 0.73 and AUC 0.81. SVM-sigmoid, however, didn't show any improvement but rather decreased to 0.54 and 0.55. So it's seems inappropriate to apply sigmoid kernel on this data. Naïve Bayes had a similar performance with SVM-poly, which is 0.74 accuracy and 0.81 AUC. Decision Tree models applied both gini index and entropy as split measurements but didn't show much difference. The performances are worse than baseline LR with accuracy and AUC of 0.69. Random Forest models using increased accuracy to 0.74 and AUC to 0.83 and using entropy achieved the highest accuracy of 0.75 among all tested and AUC of 0.83. Gradient Boosting stand at par with Naïve Bayes, SVM-poly and SVM-RBF, with accuracy of 0.73 and AUC 0.81. Ada Boost also achieved the highest accuracy of 0.75 and AUC 0.82. When plot all model performance accuracy and AUC in a scatterplot, it's easy to observe a cluster in the upper right corner are those who win the first round (*figure 4*).

A good trained model should achieve a similar performance with unseen data, which could be considered generalizing well. Next, the seven candidate models are challenged with the original data, which would have imbalanced class. 10-fold cross validation is performed for model validation. CV runtime is another measurement for choosing the best model.

When applied back on the original dataset, both RF models showed an impressive performance, with 0.85 accuracy and 0.84 AUC. Ada Boost model achieved the same scores with RF models. Gradient Boost model also achieved the same accuracy but slightly lower AUC. Both SVM models have 0.80 accuracy but much lower AUC ay 0.59. Naïve Bayes model achieved a similar performance as the train-test version, with accuracy of 0.78 and AUC of 0.74. Considering CV runtime, NB model cost less than 0.1s for computation, which no other algorithm could beat. Ada Boost comes second with less than 5s. Both RF models cost less than 10s to compute. SVM models, however, take much longer time for cross validation, especially SVM-poly kernel. In contrast to fast CV computation of Ada Boost, Gradient Boosting takes about 35 seconds (*figure 5*).

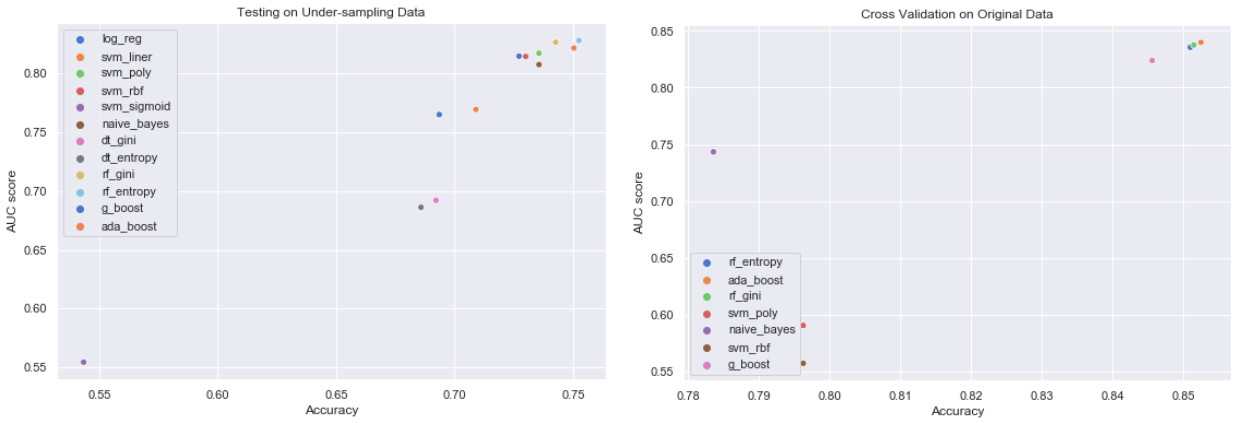


Figure 4 Scatterplots of accuracy and AUC on various models

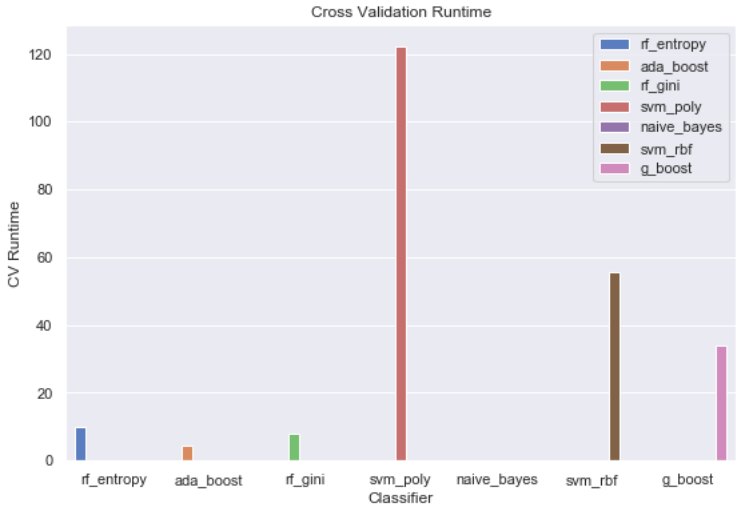


Figure 5 CV runtime

5. Discussion

When churn prediction is completed, it is also important to know what factors would cause customers to leave the company. So a further feature selection step was conducted using Random Forest classifier. Three different approaches were applied for top 5 feature selection, which are 1) Stepwise Recursive Backwards Feature removal, 2) Wrapper Select via model, 3) Univariate Feature Selection - Chi-squared. The common features shared by all methods are: Credit Score, Age, Balance and Estimated Salary.

The average credit score in the dataset is 650(*figure 6*), where 70% of U.S. consumers' FICO Scores are higher than 650. In the original data, geography variable indicates that the customers reside in France, Spain or Germany. So I was wondering if the credit scoring system is different from what exists in the US. Average Age of the customers near 40 and average estimated salary of $100,090, suggested that the studied customers are high earners and may have more choices in terms of managing personal finance. It's also interesting to observe a large number of 0 balance in the data that shows a bimodal distribution. It should give more insights if 0 balance indicates those who don't have checking account but only credit card and have already paid off or just have 0 balance in the account.

6. Conclusions

Random Forest with entropy splitting measurement is the best model for the predicting bank customer churn. It has the highest accuracy on test dataset and generalize well back on the original imbalanced dataset. It has the largest correct classification number of class 1 (churned) and the second largest correct classification number of class 0(retained customer) compared with Ada Boost.

Future work includes more parameter tuning as the models used in this analysis are all using default configuration. Since it's observed that has credit card would have an impact on customer churn, more credit card related features should be collected and included in this dataset, for example, minimum payment, APR rate, whether customer pay card on time. Since age was chosen as a predictor, I was wondering if binning age into age group would make interpretation more meaningful. Other rebalancing methods could be implemented like oversampling on churned classes to compare and find the optimal model.

Appendix

	Accuracy	AUC		Accuracy	AUC		Accuracy	AUC
Logistic Regression	0.69	0.77	SVM-sigmoid	0.54	0.55	Random Forest - Gini	0.74	0.83
SVM-Linear	0.71	0.77	Naïve Bayes	0.74	0.81	Random Forest - entropy	0.75	0.83
SVM-poly	0.74	0.82	Decision Tree - Gini	0.69	0.69	Gradient Boost	0.73	0.81

SVM-RBF	0.73	0.81	Decision Tree - entropy	0.69	0.69	Ada Boost	0.75	0.82

Table 2 Accuracy and AUC table of twelve models on test dataset

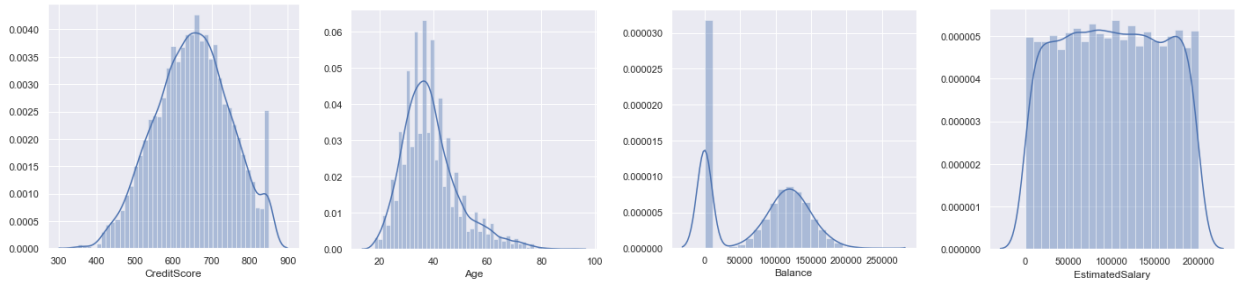
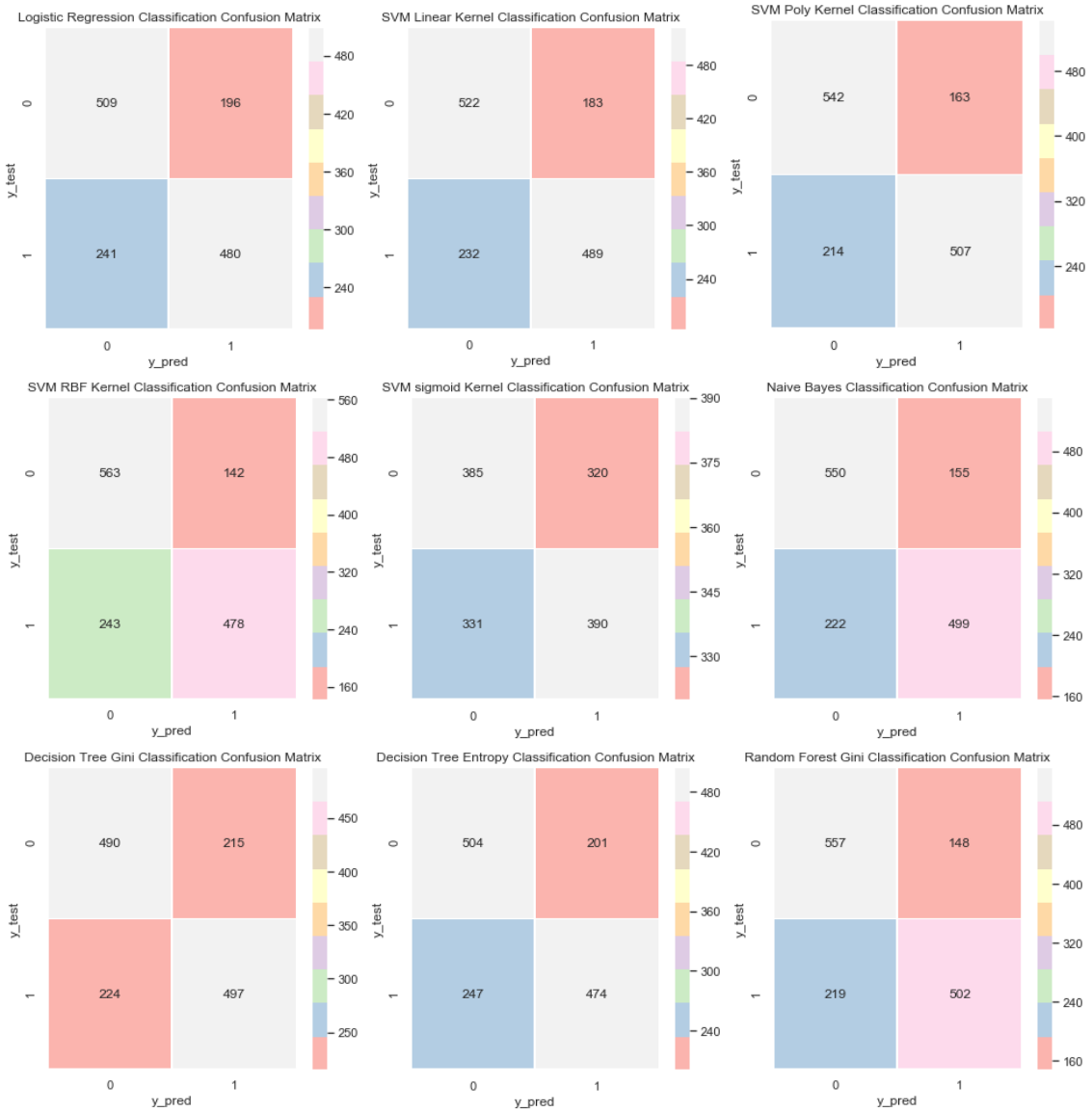


Figure 6 Histogram of continuous variables



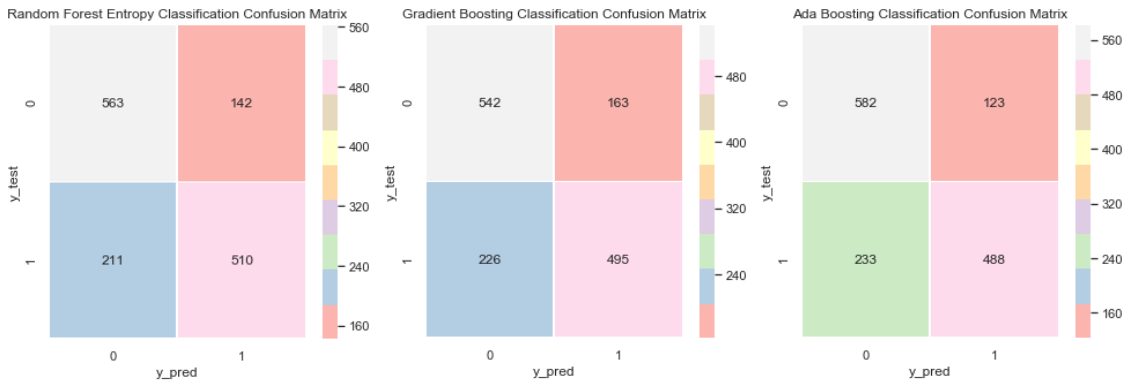


Figure 7 Confusion Matrices of different classification models

References

Wei, C. P., & Chiu, I. T. (2002). *Turning telecommunications call details to churn prediction: A data mining approach.* Expert Systems with Applications, 23(2), 103–112.

Kim, H. S., & Yoon, C. H. (2004). *Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market.* Telecommunications Policy, 28(9–10), 751–765.

Yan, L., Miller, D. J., & Mozer, M. C., et al. (2001). Improving prediction of customer behavior in non stationary environments. In *Proceedings of the international joint conference on neural net-works.*

Au, W., Chen, K. C. C., & Yao, X. (2003). *A novel evolutionary data mining algorithm with applications to churn prediction.* Evolutionary Computation, IEEE Transactions, 7(6), 532–545.

Vapnik, V. N. (2004). *The nature of statistical learning theory*. Beijing: Publishing House of Electronics Industry.

Xiaobing Yu, Shunsheng Guo, Jun Guo, Xiaorong Huang. (2011). *An extended support vector machine forecasting framework for customer churn in e-commerce.* Expert Systems with Applications 38 (2011) 1425–1430.

XiaojunWu, Sufang Meng. (2016). *E-commerce Customer Churn Prediction Based on Improved SMOTE and AdaBoost*. 2016 13th International Conference on Service Systems and Service Management (ICSSSM), 2161-1904.

Aurélie Lemmens, Christophe Croux. (2006). *Bagging and Boosting Classification Trees to Predict Churn*. Journal of Marketing Research, Vol 43, Issue 2, 2006.

T. Vafeiadis, K.I. Diamantaras, G. Sarigiannidis, K.Ch. Chatzisavvas. (2015). *A comparison of machine learning techniques for customer churn prediction*. Simulation Modelling Practice and Theory. Volume 55, June 2015, Pages 1-9