Predicting Churned Bank Customers

Carrington Gregori
Fordham University
cgregori@fordham.edu
Henry Edgington
Fordham University
hedgington@fordham.edu

I. Abstract

Our project predicts which customers will be attrited in order to preemptively send them incentives to stay with their current bank. This project uses a dataset of over 10,000 customers with 23 features, with the Attrited flag as the class variable. Key features include total transaction count, total revolving balance, credit limit, and average card utilization ratio. Since there is a large amount of class imbalance in this dataset (84% vs 16%), we performed various oversampling and undersampling techniques to construct the best predictive model. To classify these customers, we used Naive Bayes, Random Forest, JRip, and KStar classifiers. The oversampled RandomForest and JRip were the most precise predictors.

II. Introduction

Our project sets us in the role of a third party contractor, working for an imaginary bank where we aim to help them determine which customers will churn, or stop using the bank. With this data, the bank can proactively provide better services at the customers who are most likely to leave. Our imaginary bank has generously allowed us to mine some of the massive amounts of data the bank has about its customers. Since there's a large class imbalance between existing customers (84%) and attrited customers (16%), we apply various oversampling and undersampling techniques to create balance. This class imbalance also compelled us to use Cohen's kappa statistic to compare models. With this data we have built a predictive model to predict the customers who will leave.

III. Experiment Methodology

A. Dataset Description

Our dataset consists of over 10,000 customers data with 23 data points per customer. The information about each customer includes:

Age	Gender	Dependent count
Education level	Marital status	Income level
Months on book	Relationship count	Card category
Months Inactive	Contacts Count	Credit Limit
Revolving Balance	Avg Open to Buy	
Total transaction amount	Change in transaction count (Q4 over Q1)	Change in transaction amount (Q4 over Q1)
Total transaction count		Avg Utilization Ratio

Relationship count is the total number of products held by the customer. Average open to buy is the customer's credit limit minus the present balance averaged over the last four months.

Because of the hard work of our client, the bank, the vast majority of the customers in the dataset are consistently recurring customers. This leads to there being a much larger majority class of customers that are not atritted, making the data imbalanced. To correct for the imbalance, we separately both under sampled the majority class and oversampled the minority class.

B. Setup of experiments:

Because of the expansive size of our data set, in order to create a more accurate model we had to prune some features that were not correlating to increased accuracy. To find the ideal features we did some research into other projects with the same or similar data sets, then confirmed our findings by testing them in WEKA. We ended up partitioning the data into two different sets. The first set was

constructed with Weka's CorrelationAttributeEval using Ranker as the search method. Everything under an arbitrary value of .05 was pruned, leaving 9 features remaining. This '*correlation*' dataset features are:

Total_Trans_Ct	Contacts_Count_12_mon
Total_Ct_Change_Q4_Q1	Total_Trans_Amt
Total_Revolving_Bal	Months_Inactive_12_mon
Avg_Utilization_Ratio	Total_Relationship_Count
Total_Amt_Chng_Q4_Q1	

The second set was constructed with Weka's WrapperSubsetEval with the BestFirst search method. The merit of the best subset found was 0.949 and 8 features remained. This '*learner*' dataset features are:

Customer_Age	Total_Trans_Amt
Total_Revolving_Bal	Total_Relationship_Count
Avg_Utilization_Ratio	Credit_Limit
Total_Amt_Chng_Q4_Q1	Total_Trans_Ct

After determining the ideal features, we undersampled and oversampled our data set. Undersampling is the process of using equal numbers of the majority and minority class to build the predictive model. We did this using Weka's SpreadSubSample filter. Oversampling, on the other hand, is creating synthetic data from the minority class using very similar feature values to existing data .We oversampled using Weka's SMOTE filter with 5 nearest neighbors at 100% (Doubling the minority class). We also performed 10-fold cross validation on each run of the algorithms.

C. Evaluation Metrics

To evaluate our findings, we chose the metrics of accuracy, f-measure, and kappa. While accuracy may be self explanatory, f-measure and kappa are not so common. F-measure is the harmonic mean of the precision and recall of a test. Kappa on the other hand is a measure of how much better a classifier is than random guessing according to frequency. By both measures the ideal output is one, with anything less than one being less ideal and greater than one impossible.

D. Algorithms

The four algorithms we used in this experiment were Weka's Naive Bayes, Weka's JRip rules, Weka's Lazy KStar, and Weka's RandomForrest. We planned from the beginning to use as many different classification algorithms as possible in order to have a wider dataset in order to observe which model works best. Even with our wide breadth of models, we were able to determine the ideal features tailored to each classification task. Knowing which features were most relevant also allowed us to try and understand what was the main cause of the misclassifications in our output.

IV. Results

The oversampled dataset runs performed better than the undersampled and standard runs. Both the correlation and learner datasets had higher accuracies, kappa, and f-measure for oversampling as opposed to undersampled and standard runs. This is possibly because the oversampled dataset had access to the most data, as a result of it creating synthetic data similar to that of the mino. The correlation and learner datasets each found different strengths in the different models. Overall, the correlation dataset performed better with the naive bayes and KStar algorithms. The learner dataset performed better on the JRip and RandomForest classifier.

Some features for each dataset were similar. These were:

Total_Revolving_Bal	Total_Trans_Amt
Avg_Utilization_Ratio	Total_Relationship_Count
Total_Amt_Chng_Q4_Q1	Total_Trans_Ct

The different features between each dataset were:

Correlation	Learner
Months_Inactive_12_mon	Credit_Limit
Total_Ct_Chng_Q4_Q1	Customer_Age
Contacts_Count	

We found that credit limit and customer age are more helpful for rules-based and trees-based classifiers, while months inactive, contacts count, and total count change are more helpful for bayes-based and lazy classifiers.

TABLE I. Accuracy of Classifications(%)

	Naive Bayes	JRip	KStar	Random Forest
Correlation	88.2394	94.3122	94.0851	93.463
Learner	87.9727	94.5097	92.3669	96.3267
Undersampled Correlation	78.3651	91.0572	90.8728	94.2225
Undersampled Learner	73.4481	90.9957	87.0621	93.7615
Oversampled Correlation	83.4439	94.3424	94.7252	96.4097
Oversampled Learner	81.2404	94.47	94.3679	96.4353

The highest accuracy (96%) was in Random Forest's oversampled correlation, oversampled learner, and standard learner datasets. Although, this high accuracy may be because of overfitting. The lowest accuracies were consistently made by the Naive Bayes algorithms. The oversampled datasets were consistently the most accurate, while the undersampled consistently underperformed. This underperformance is quite possibly due to the dataset being small. Because the minority class only has ~1600 examples, the total dataset was ~3200 examples (½ existing, ½ attrited).

TABLE II. Kappa Statistic of Classifiers

	Naive Bayes	JRip	KStar	RandomForest
Correlation	0.5696	0.7868	0.7742	0.7583
Learner	0.5406	0.7937	0.7073	0.8603
Undersampled Correlation	0.5673	0.8211	0.8175	0.8844
Undersampled Learner	0.469	0.8199	0.7412	0.8752
Oversampled Correlation	0.5954	0.8586	0.87	0.9101
Oversampled Learner	0.5376	0.8613	0.8625	0.9105

Naive Bayes consistently performs the worst once again. Meanwhile, Random Forest performs the best, with its kappa stats being the closest to 1. Oversampled KStar follows close behind, while oversampled JRip takes third. When looking at the undersampled and standard datasets, JRip takes second while KStar takes third. Overall, the correlation datasets perform better when undersampled or standard sampled. The learner datasets perform best when its data is oversampled.

TABLE III. F-Measure of Classifications
Correlation-pruned Dataset

	Naive Bayes	JRip	KStar	Random Forest
Existing	0.930	0.966	0.965	0.961
Attrited	0.640	0.821	0.809	0.797
Total	0.883	0.943	0.940	0.935

TABLE IV. F-Measure of Classifications
Undersampled Correlation-pruned Dataset

	Naive Bayes	JRip	KStar	Random Forest
Existing	0.778	0.910	0.908	0.942
Attrited	0.789	0.911	0.910	0.943
Total	0.784	0.911	0.909	0.942

TABLE V. F-Measure of Classifications
Oversampled Correlation-pruned Dataset

	Naive Bayes	JRip	KStar	Random Forest
Existing	0.884	0.961	0.963	0.975
Attrited	0.711	0.896	0.907	0.935
Total	0.836	0.943	0.948	0.964

RandomForest performs the best in each of the three different types of datasets (standard, over/undersampled). Overall, the correlation dataset performs the best for attrited customers, which is the most important class value. This leads us to believe that the bank would get the most success for their models using the oversampled learner dataset's features. This, combined with RandomForest or JRip will provide the most precise predictive model.

TABLE VI. F-Measure of Classifications
Learner-pruned Dataset

	Naive Bayes	JRip	KStar	Random Forest
Existing	0.929	0.967	0.955	0.978
Attrited	0.612	0.826	0.752	0.882
Total	0.878	0.945	0.922	0.963

TABLE VII. F-Measure of Classifications
Undersampled Learner-pruned Dataset

	Naive Bayes	JRip	KStar	Random Forest
Existing	0.723	0.909	0.869	0.937
Attrited	0.745	0.911	0.872	0.938
Total	0.734	0.910	0.871	0.938

TABLE VIII. F-Measure of Classifications
Oversampled Learner-pruned Dataset

	Naive Bayes	JRip	KStar	Random Forest
Existing	0.869	0.962	0.960	0.975
Attrited	0.668	0.899	0.902	0.935
Total	0.814	0.945	0.944	0.964

V. Conclusion

Somewhat unsurprisingly, the oversampled dataset performed better than both the undersampled and standard datasets. This is most likely a result of the algorithms having access to more data in the oversampled data sets due to the synthetic data. The correlation and learner datasets both had higher accuracies, kappa, and f-measure for oversampling than the undersampled and standard runs. The correlation and learner datasets each found different

strengths in the different models. Overall however, the correlation dataset performed better with the Naive Bayes and KStar algorithm and the learner dataset turned out to have performed better on the JRip and RandomForest classifiers. However, the oversampled data sets bring the ratings of each classifier to a relatively similar level (when compared against itself). It is therefore our recommendation that the bank take the RandomForest or JRip algorithm, and oversampled learner dataset. This will provide the most accurate, and precise model when it comes to predicting the soon-to-churn customers.

VI. References

1. Goyal, Sakshi. “Credit Card Customers.” *Kaggle*, 19 Nov. 2020, www.kaggle.com/sakshigoyal7/credit-card-customers.
2. dwight1225. “Predicting Churned Customers(over/Undersample).” *Kaggle*, Kaggle, 15 Dec. 2020, www.kaggle.com/dwight1225/predicting-churned-customers-over-undersample.