Credit Card Default:

Imbalanced Target and Prediction Scorecard

Mary Donovan Martello

Project 2: Milestone 4

Executive Summary

Defaults of credit card debt hurt both the lender company as well as the economy as a whole. Credit card companies must endeavor to predict if the status of customers' accounts indicate that the customers are on the verge of default so that they may mitigate future defaults. This project expanded on a previous project that tested five classification models' ability to predict if credit card account holders would default in the next month. The goal of this current project was to (1) employ methodologies other than Principal Component Analysis to address multicollinearity and feature selection in an effort to be able to analyze individual factors that are significant for defaults, (2) test multiple approaches to addressing an imbalanced target variable in a binary classification model, and (3) implement a deployment strategy of making actual predictions and converting the logistic regression coefficients to a risk score that can be used in making lending decisions.

Feature engineering was used to reduce the number of collineated features and employ two new predictive features. Five different techniques were tested to balance the target feature, including: (1) adjusting the probability classification threshold (without resampling the data), (2) split the dataset with the simple holdout method to oversample the training data only, (3) split the data with the k-fold cross validation method (sklearn's GroupKFold method) to oversample only the training portion of each fold, (4) split the data with the k-fold cross validation method (imblearn's Pipeline method) to oversample only the training portion of each fold, and (5) split the data using k-fold cross validation method to both oversample and under sample the training portion of each fold.. In addition, different methods of oversampling were tested.

In addition, each of these techniques were tested on different combinations of input features. The results show that the best results came from using the predicted probabilities to find best classification threshold technique and the new engineered features of months_late and repayment_ratio as input features. A technique to make default predictions on new account data was developed. In addition, a scoring system for the account holder's continuing credit risk was developed.

The conclusions from this project are that the account holders' features of the length of time the due payments are late, the ratio of payments made to outstanding debt, and the credit limit in relationship to the amount of debt incurred are predictive of pending default. Adjusting the probability classification threshold (without resampling the data) showed the best results for predicting pending defaults. The models built to make default predictions and to assign credit risk to account holders will be valuable to credit card companies for detecting pending defaults.

Introduction/Background

Defaults of credit card debt are not only harmful to the lender company but also to the economy as a whole. This is demonstrated by the fact that the United States Federal Reserve tracks the delinquency rate and the charge-off rates of credit cards of commercial banks (Charge-Off and Delinquency Rates on Loans and Leases at Commercial Banks, 2021). There is $893 billion of outstanding commercial credit card debt in the United States and a 2.62% credit card charge-offs rate due to credit card defaults (15 Shocking Credit Card Debt Statistics, 2020) (Charge-Off and Delinquency Rates on Loans and Leases at Commercial Banks, 2021). Credit card companies must endeavor to predict if the status of customers' accounts indicate that the customers are on the verge of default so that they may mitigate future defaults.

The dataset used for this project includes actual credit card records at points in time in the past. Each record in the dataset also includes a label of whether the account holder defaulted the next month (default = 1) or did not default the next month (default = 0). The dataset is imbalanced with respect to the percent of records with default labels (22%) versus records with non-default labels (78%). This imbalance in the target variable must be addressed before or within the models in order to properly evaluate accuracy metrics from the models as the imbalance makes it much more likely to predict a non-default and be accurate regardless of the skill of the model (Alam, 2020) .

This project expanded on a previous project that tested five classification models' ability to predict if credit card account holders would default in the next month. The goal of this project was to (1) improve the predictive classification models by identifying individual credit account factors that are significant for defaults, (2) test multiple approaches to addressing an imbalanced target variable in a binary classification model, and (3) implement a deployment strategy of making actual predictions and converting the logistic regression coefficients to a risk score that can be used in making lending decisions.

Methods

The analysis followed the CRISP-DM stages for data science projects including Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment. The

Business Understanding analysis is summarized in the Introduction / Background section above. The Data Understanding, Data Preparation, and Modeling, stages are discussed in this Methods section. Evaluation and Deployment is subsequently discussed below.

Data Understanding and Data Preparation

As this project is an extension of a previous project analyzing this data, in this paper we will only address the new data understanding and preparation techniques and results. Our initial paper is available for review of the previous analysis.

The dataset used for this project includes actual credit card records at points in time in the past. The credit card account features include the account holder's credit limit, six monthly values indicating if the account holder was late in making payments, six monthly values of the then-amount of debt outstanding, and six monthly values of the amounts of payments made in those six previous months. Appendix A includes full details on the dataset.

Goals of feature selection are to find a reduced subset of the input features in which maximum redundant and irrelevant information is eliminated and to identify key features in a dataset that result in accurate predictions (Alam, 2020). In our first project using the credit account holder dataset, the number of highly correlated features resulting from six months of data for three different features were addressed by performing a Principal Component Analysis (PCA) on the quantitative features. This second project sought to employ methodologies other than PCA to address multicollinearity and feature selection for the purpose of being able to analyze individual factors that are significant for defaults.

The six pay status variables were used to create a single feature of the total number of months the account holder had late payments, and the six pay status variables were then dropped. The twelve payment amount and outstanding bill amount features were used to create a single feature of the ratio of payments made over six months to the outstanding debt for those six months. In some models, the payment ratio feature was used instead of the twelve features making up the payment. In other models, the twelve original features were used to test the difference in results.

For these two engineered features, we examined differences in the features when the target feature was designated as default vs non-default. We looked at differences in the data distributions and statistics for a subset of the default records versus a subset of the non-default records. The histograms and descriptive statistics showed distinctions between the default records and the non-default records for both these months late and repayment ratio features. The cumulative distribution plot for the months late feature also showed distinction. The feature importance statistics from logistic regression, decision tree classifier, random forest classifier, and XGBoost classifier models confirmed that the repayment ratio feature was more important to the target variable than its individual component features. Also, both the months late and repayment ratio were more important to the target feature than the other input features.

Modeling

The Modeling phase for this project tested different techniques for balancing the target feature. The dataset is imbalanced with respect to the percent of records with default labels (22%) versus records with non-default labels (78%). This imbalance in the target variable must be addressed before or within the models in order to properly evaluate accuracy metrics from the models as the imbalance makes it much more likely to predict a non-default and be accurate regardless of the skill of the model (Alam, 2020) .

To test the different imbalance techniques, we used a Logistic Regression model and a grid search for tuning the model's hyperparameters. These models were then tested with different combinations of input features. The various balancing techniques tested were: (1) adjusting the probability classification threshold (without resampling the data), (2) split the dataset with the simple holdout method and use SMOTE to oversample the training data only, (3) split the data with the k-fold cross validation method (sklearn's GroupKFold method) and use SMOTE to oversample only the training portion of each fold, (4) split the data with the k-fold cross validation method (imblearn's Pipeline method) and use four different SMOTE methods to oversample only the training portion of each fold, and (5) split the data using k-fold cross validation methods and use SMOTE to both oversample and under sample the training portion of each fold.

1. Adjusting the Probability Classification Threshold / No Resampling

Without resampling an imbalanced dataset, the imbalance can be addressed by selecting a threshold (other than the standard of 0.5) for classifying the default target variable. Classification predictive modeling involves predicting a class label for records. Models with imbalanced target variables can address the imbalance with the prediction of a probability of class membership (Brownlee, 2020) (Evgeniou & Zoumpoulis, n.d.) (Rosen, 2020).

We used the best parameters from a grid search to make predicted probabilities for positive outcome predictions of the target classification and used a Receiver Operating Characteristic (ROC) test to select the best classification threshold (Rosen, 2020). The probability predictions were then allocated between 1 and 0 (the binary classification values) based on the best threshold. Thus, we now have classification predictions with tuned parameters using an optimized classification threshold for an imbalanced target variable.

Resampling the Training Data (Only)

The remaining techniques we tested for addressing the imbalanced dataset involve resampling the dataset, either by oversampling the minority class or both under sampling the majority class and oversampling the minority class. In either oversampling or under sampling, it is important to only resample the training data because resampling before splitting into training and testing datasets "bleeds" information from the test set into the training of the model (Becker, 2016) (Brownlee, SMOTE for Imbalanced Classification with Python, 2021).

2. Holdout Method / SMOTE Oversampling of Training Data

The next technique for addressing the imbalanced dataset was to split the dataset with the simple holdout method and use SMOTE to oversample the training data. This results in a new overbalanced training dataset that was used in training the model.

3. **K-Fold Cross Validation Method / SMOTE Oversampling of Training Data**

Using the holdout method to create a training set to be resampled is a simple approach to addressing the imbalanced dataset. However, using k-fold cross-validation to split the data is often a

better approach to split the data in order to address bias and variation issues with the holdout method. sklearn's GroupKFold method and imblearn's Pipeline function provide methods for resampling only the training portion of the data in each fold during cross-validation (Rosen, dabruro.medium.com, 2021) (Brownlee, SMOTE for Imbalanced Classification with Python, 2021). We tested both sklearn's GroupKFold method and imblearn's Pipeline function to oversample the minority class. The imblearn library has several SMOTE versions for oversampling data. We tested of the following SMOTE versions: SMOTE, BorderlineSMOTE, SVMSMOTE, and ADASYN.

4. **K-Fold Cross Validation Method / SMOTE Oversampling and Under Sampling of Training Data**

Another resampling approach is to both oversample the minority class and under sample the majority class. We used k-fold cross validation and imblearn's SMOTETomek class to test this method.

Input Features

At this point we have tested models for each of the above-described balancing approaches using one set of input features. We will be testing each of the approaches with various sets of input features.

Results

We used the Recall metric to evaluate the different Logistic Regression models created for each of the above-described approaches for addressing the imbalanced dataset. The following are the results:

[Continued on the next page.]

			Subset 1	Subset 2	Subset 3	Subset 4
Approach to Imbalance	**Split Method**	**Resampling Class**	**Recall**	**Recall**	**Recall**	**Recall**
Predicted probabilities to find best classification threshold	N/A	N/A	0.7073	0.7175	0.7016	0.7196
Oversample training data	Holdout Method	SMOTE	0.7008	0.7158	0.6990	0.7090
Oversample training data	GroupKFold CV	SMOTE	0.5000	0.5000	0.6814	0.7172
Oversample training data	imblearn Pipeline	SMOTE	0.6993	0.7160	0.6997	0.7090
Oversample training data	imblearn Pipeline	BorderlineSMOTE	0.6988	0.7151	0.6976	0.7092
Oversample training data	imblearn Pipeline	SVMSMOTE	0.7012	0.7160	0.6961	0.7101
Oversample training data	imblearn Pipeline	ADASYN	0.6973	0.7159	0.6983	0.7092
Oversample and under sample training data	imblearn Pipeline	SMOTETomek	0.6998	0.7135	0.6995	0.7092

	Subset 1	Subset 2	Subset 3	Subset 4
Approach to Imbalance	**Recall**	**Recall**	**Recall**	**Recall**
Predicted probabilities to find best classification threshold	1	1	1	1
Oversample training data	6	6	4	3
Oversample training data	2	4	8	6
Oversample training data	8	7	2	8
Oversample training data	4	2	7	7
Oversample training data	5	5	5	5
Oversample training data	7	8	6	4
Oversample and under sample training data	3	3	3	2

Deployment

We will introduce two deployment methods for the credit card company to use for making decisions of whether to issue new credit or suspend available credit. First, we set up a program to input the account or new application data and have our best model predict whether that account holder will default in the next month. Second, we followed a program for assigning a risk score to an account holder or account applicant for the credit card company to evaluate risk of extending or continuing credit.

1. **Predictions**.

We have set up a program to input the new account or new application data and have our best model predict whether that account holder will default in the next month. It is a simple program to employ, and the results are simple to interpret: either a prediction of pending default or a prediction of no pending default. The credit card company can choose to act on or reevaluate their account holder.

2. **Credit Score**

Instead of making a prediction of whether an account holder will default in the next month, a credit card company may want to evaluate credit risk based on an applicant or account holder's credit score. Asad Mumtaz developed a credit scorecard based on a logistic regression model's coefficients (Mumtaz, 2020). Since we have trained logistic regression models to make default predictions, we incorporated Mumtaz's system to create a scorecard for our credit card company to use.

The scorecard program takes as input the account holder or applicants account information and returns a risk score between 100 and 500, with 500 being the best credit risk. The risk score is based on a whether the observation falls within certain categories. These categories are calculated based on Weight of Evidence and Information Value measures as they used extensively used in the credit scoring domain (Mumtaz, 2020). Weight of Evidence is a measure of the predictive power of an independent variable in relation to the target variable and Information Value assists with ranking the features based on their relative importance. Thus, we have a scorecard with categories and then the points allocated to the categories are based on the intercept and coefficients obtained from fitting the Logistic Regression model.

Based on our Data Understanding and Modeling, we have designed our scorecard to return a score based on the three key inputs of months late, credit limit, and repayment ratio. The following is our scorecard:

	index	Feature name	Coefficients
0	0	Intercept	1.088768
1	2	MONTHS_LATE:<0.693147	0.004906
2	3	MONTHS_LATE:0.693147-1.098612	0.392497
3	4	MONTHS_LATE:1.098612-1.609438	0.112620
4	5	MONTHS_LATE:1.609438-1.791759	0.000000
5	6	PAYMENT_RATIO:<0.955	0.120415
6	7	LIMIT_BAL:<10.131	-0.887972
7	8	LIMIT_BAL:10.131-11.052	-0.700523
8	9	LIMIT_BAL:11.052-11.973	-0.458005
9	10	LIMIT_BAL:11.973-12.894	-0.276142
10	0	MONTHS_LATE:>1.791759	0.000000
11	1	PAYMENT_RATIO:>0.954	0.000000
12	2	LIMIT_BAL:>12.894	0.000000

The following are outputted scores:

Scores

```
: # matrix dot multiplication of test set with scorecard scores
  y_scores = X_test_woe_transformed.dot(scorecard_scores)
  y_scores.head()
```

:

	0
15717	258.0
11597	189.0
21942	310.0
11331	258.0
13503	389.0

Discussion/Conclusion

In this project, we were better able to identify which of the credit card holder account features were significant for predicting a pending default. Months_late and repayment_ratio are significant features. We tested multiple ways to address the imbalance in the target feature and tested different subsets of features in our Logistic Regression model.

Using predicted probabilities to find the best threshold had the highest recall scores for every subset of input features. Using SVMSMOTE to overbalance the training data also did well in three of the four data sets. The GroupKfold method overall did not do well and was very inconsistent.

Regarding which input features were best performing, the highest overall recall score came from the subset that only used the two engineered features discussed above: months late and repayment ratio. In fact, each of the imbalance techniques tested had its best score using a set of input features that included these two engineered features.

Overall, it looks like the best technique for addressing the target imbalance likely depends on the input features used and multiple techniques should be tested.

The results of the Logistic Regression model were used to create a scorecard of credit risk for new observations. We can also input account status information into our model and predict whether the account will have a default within the next month. The credit card company can use these two deployment methods for making decisions of whether to issue new credit or suspend available credit.

Our next steps should be to continue to fine-tune the models, explore using prediction probability thresholds to set credit approval cut-offs, and test and implement the deployment models.

References

15 Shocking Credit Card Debt Statistics. (2020, August 10). Retrieved from CardRates.com: https://www.cardrates.com/advice/shocking-credit-card-debt-statistics/

Alam, T. M. (2020). An Investigation of Credit Card Default Prediction in the Imbalanced Datasets. *IEEE Access*, Vol. 8 201173-201198.

Becker, N. (2016, December 16). *The Right Way to Oversample in Predictive Modeling*. Retrieved from https://beckernick.github.io/oversampling-modeling/

Brownlee, J. (2020, January 14). *A Gentle Introduction to Probability Metrics for Imbalanced Classification*. Retrieved from machinelearningmastery.com: https://machinelearningmastery.com/probability-metrics-for-imbalanced-classification/

Brownlee, J. (2021, March 17). *SMOTE for Imbalanced Classification with Python*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Charge-Off and Delinquency Rates on Loans and Leases at Commercial Banks. (2021, February 21). Retrieved from federalreserve.gov: https://www.federalreserve.gov/releases/chargeoff/

Comparative Analysis of Machine Learning Techniques for Detecting Insurance Claim Fraud. (n.d.). Retrieved from wipro: https://www.wipro.com/en-US/analytics/comparative-analysis-of-machine-learning-techniques-for-detectin/

Evgeniou, T., & Zoumpoulis, S. (n.d.). *Classification for Credit Card Default*. Retrieved from http://inseaddataanalytics.github.io/: http://inseaddataanalytics.github.io/INSEADAnalytics/CourseSessions/ClassificationProcessCreditCardDefault.html

Mumtaz, A. (2020, August 13). *How to Develop a Credit Risk Model and Scorecard*. Retrieved from Towards Data Science: https://towardsdatascience.com/how-to-develop-a-credit-risk-model-and-scorecard-91335fc01f03

Rosen, D. B. (2020, August 1). *How To Deal With Imbalanced Classification, Without Re-balancing the Data*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/how-to-deal-with-imbalanced-classification-without-re-balancing-the-data-8a3c02353fe3

Rosen, D. B. (2021, March 3). Retrieved from dabruro.medium.com: https://dabruro.medium.com/as-an-alternative-to-implementing-the-oversampling-inside-the-cross-validation-loop-you-could-use-64a04e4f6425

Appendix A

This is the source of the data:

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Here is a data dictionary of the dataset:

- ID: ID of each client
- LIMIT_BAL: Amount of the given credit in NT (New Taiwan) dollars (includes individual and family/supplementary credit)
    - For reference, 1 US dollar = 28.57 NT dollars on 3/19/2021.
- SEX: Gender (1=male, 2=female)
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- MARRIAGE: Marital status (1=married, 2=single, 3=others)
- AGE: Age in years
- PAY_0: Repayment status in September, 2005
- PAY_2: Repayment status in August, 2005 (scale same as above)
- PAY_3: Repayment status in July, 2005 (scale same as above)
- PAY_4: Repayment status in June, 2005 (scale same as above)
- PAY_5: Repayment status in May, 2005 (scale same as above)
- PAY_6: Repayment status in April, 2005 (scale same as above)
    - The above Pay variables have the following values: -2 means no account usage, -1 means balance paid in full, 0 means at least the minimum payment was made, 1 means payment delay for one month, 2 means payment delay for two months, etc. up to 9, which means payment delay for nine months and above.
- BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
- BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
- BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
- BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
- BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
- BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
- PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
- PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
- PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
- PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
- PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
- PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
- default.payment.next.month: Label for default payment (1=yes, 0=no)