Predicting Default Payment of Credit Card Clients

PSTAT 135 Group Project

By Leah Li, Wanqing Liu, Wenhao Song, Angel Chen

March 19, 2021

I. Abstract

In this project, our goal is to predict whether the client will default on credit card payments based on their gender, education, marital status, age, amount of the given credit, history of past payment, amount of bill statements and amount of previous payments. We built the pipeline to preprocess the data and construct models. We applied several feature transformers such as VectorAssembler and StandardScaler from the PySpark ML package. The four models we developed were Logistic Regression, Decision Tree, Random Forest and Support Vector Machine. We evaluated our models' performance using the metrics derived from the confusion matrix, and we finally chose Support Vector Machine as our champion model because it has the best accuracy of 0.8065.

II. Data

Background

This dataset contains information on clients in Taiwan and their credit card payment history, presented as 30,000 rows and 25 columns. The dataset originated from the UCI Machine Learning Repository, and it can be found here: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients.

The response variable is "default payment next month," coded as 1 = yes and 0 = no.
The 24 features in the dataset are listed below. All amounts are in New Taiwan dollars.

- ID: client ID
- LIMIT_BAL: amount of the given credit in New Taiwan dollars
- SEX: sex of the client, 1 = male and 2 = female
- EDUCATION: education level of the client, 1 = graduate school, 2 = university, 3 = high school, 4 = other
- MARRIAGE: marital status of the client, 1 = married, 2 = single, 3 = other
- AGE: age of the client in years
- PAY_0: repayment status in September 2005, coded as -1 = paid on time, 1 = payment delay for one month, 2 = payment delay for two months, ..., 8 = payment delay for eight months, 9 = payment delay for nine months or more
- PAY_2: repayment status in August 2005, with the same code
- PAY_3: repayment status in July 2005, with the same code
- PAY_4: repayment status in June 2005, with the same code
- PAY_5: repayment status in May 2005, with the same code
- PAY_6: repayment status in April 2005, with the same code
- BILL_AMT1: amount billed for the statement in September 2005
- BILL_AMT2: amount billed for the statement in August 2005
- BILL_AMT3: amount billed for the statement in July 2005
- BILL_AMT4: amount billed for the statement in June 2005

- BILL_AMT5: amount billed for the statement in May 2005
- BILL_AMT6: amount billed for the statement in April 2005
- PAY_AMT1: amount paid in September 2005
- PAY_AMT2: amount paid in August 2005
- PAY_AMT3: amount paid in July 2005
- PAY_AMT4: amount paid in June 2005
- PAY_AMT5: amount paid in May 2005
- PAY_AMT6: amount paid in April 2005

Data Preprocessing

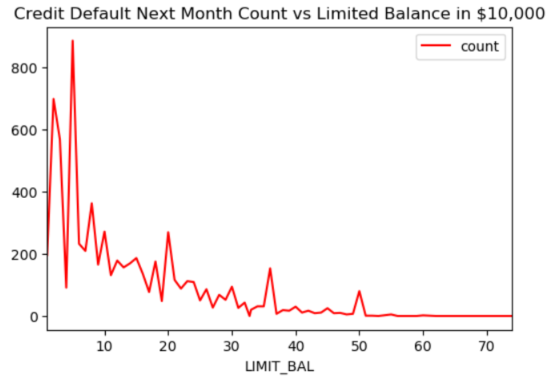
For our data preprocessing step, we checked for duplicates, missing values, and outliers. Since the number of rows in our dataset and the number of distinct rows are the same, we concluded that there were no duplicate rows. Then, by using the aggregate function, we calculated the percentage of missing data for each column, but thankfully there were none. Finally, we checked for outliers in the non-categorical features by creating lower and upper bounds based on the interquartile range. If a value falls outside of the bounds, then it would be considered an outlier. In the end, there were 18,052 rows that contained outliers so we decided that perhaps it would be for the best to not remove any outliers.

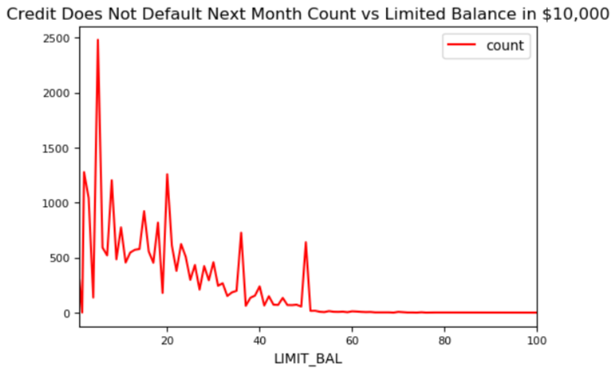
Exploratory Analysis

Among the 23 variables, we chose X1:Limit_Bal, X2: Sex, X3: Education to explore and visualize their relationship with type of default.

Limit Balance

Since each card holder is granted different levels of balance limit in New Taiwanese Dollars for their credit card spendings, we are interested in the existence of a correlation between limit balance and credit card default. First we separated the clients into two groups: Default Payment and Not Default Payment next month. To analyze the distribution of default/ not default among various limit balance levels, we grouped the data by the limit balances for both groups. Then, we plotted the count of default/ not default payment count vs limit balance in $10,000.

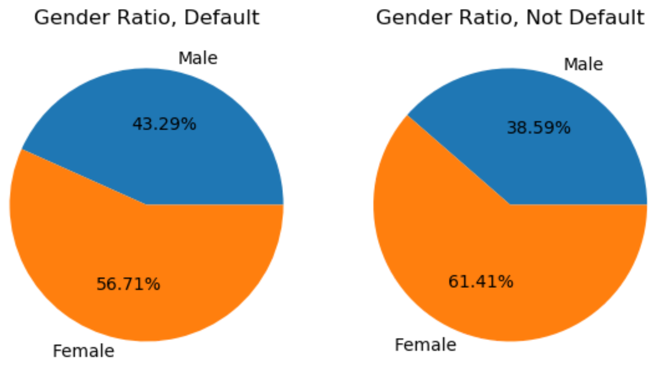




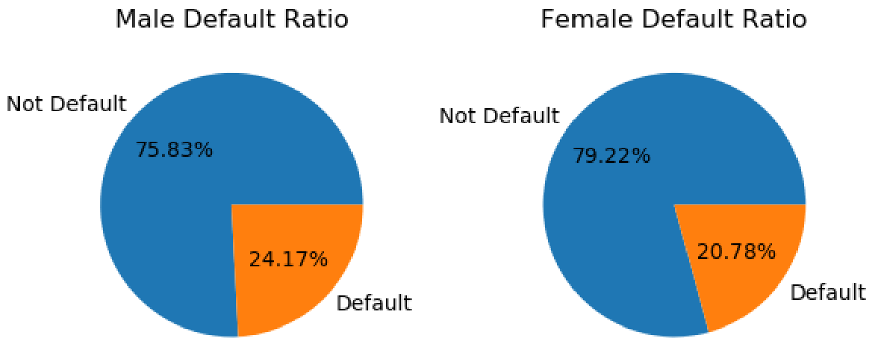
From the plots on the previous page, we did not see a significant difference between the distributions of the two groups. Although there seems to be a trend of decreasing counts as limit balance increases in both groups, this can be explained by the fact that as limit balance increases, the qualifications are harder to meet, thus there are less card holders with high limit balance. Thus, we could conclude that there is a correlation between default/not default payment counts and limit balance.

Sex

Interested to see whether there is a relationship between gender and the type of default, we separated the clients into two groups: Default and Not Default. For each group, we calculated the proportion of males and females. In the pie charts shown below, we can see that in both groups, the proportion of females is higher, indicating that there are more female clients recorded in this dataset.



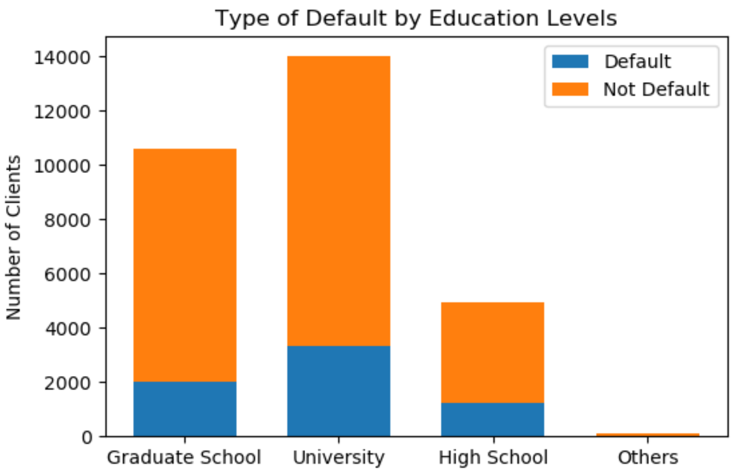
We also calculated the default ratio for each gender:

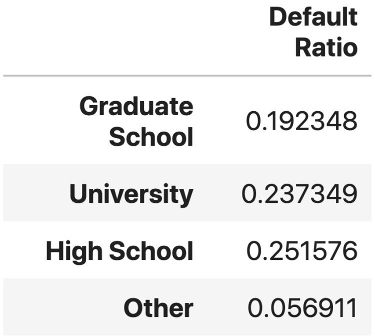


We can see that male clients have a slightly higher default ratio of 24.17%, while female clients have a default ratio of 20.78%.

Education Levels

Different education levels may have an effect on defaulting credit card payments. Curious to find out whether there is a relationship, we calculated the number of clients in each education level. For each level, we also separated them into two groups 'Default' and 'Not Default'.





	Default Ratio
Graduate School	0.192348
University	0.237349
High School	0.251576
Other	0.056911

From the histogram above, we see that customers with a University or Graduate School education take up the biggest proportion of clients. Clients with a High School education have the highest percentage of default, and the default ratio drops as the education level gets higher. "Other" Education level clients have the lowest default ratio, but this ratio may be biased since there are not many records in this category.

III. Methods and Results

Data Splitting and Balancing

To prepare for our model training, we split the dataset, using 70% as the training set and the remaining 30% as the test set. As a result of our exploratory analysis, we found out that there were way more clients with no default payments than clients with default payments. About 77.88% of clients had no default payments, compared to the 22.12% that do. Since the data is imbalanced, we needed to balance the training set using the oversampling method. Before oversampling, there were 4,640 rows labelled as 1 (default payment next month) and 16,285 rows labelled as 0 (no default payment next month) for the training set. After oversampling, there were 13,920 rows labelled as 1 and the number of rows labelled as 0 remained the same.

Models

We decided to try out various models using Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine to predict whether a client will default on their credit card payment next month. We fitted all the models on the training set, made predictions on the test set, and finally used different evaluation metrics to compare the performance of models.

- **Benchmark**
  Our benchmark model is the simplest model and serves as a sanity check against our other models. For this pipeline, we applied VectorAssembler() and assembled only three of the features, LIMIT_BAL, SEX, and AGE into a feature vector. Then we trained a logistic regression model, which gave us an accuracy of 0.6365 on our test set.

- **Logistic Regression**
  In addition to using just three features with logistic regression, we also tried using all features aside from ID in another logistic regression model. For this pipeline, we assembled all features aside from ID into an unscaled feature vector. Then we applied StandardScaler() to make a column for the scaled feature vector and created the logistic regression model. We wanted to explore the different settings of hyperparameters so we set up a grid of parameters to do a 10-fold cross-validation over. After fitting the train set, the best model chosen from the cross validation gave an accuracy of 0.7718 for the test set.

- **Decision Tree**
  Since our response variable is binary, we are interested to see whether a decision tree model would perform well on this task. As a decision tree model is scale-invariant, the pipeline only contains two steps: vector assembling and model building. As always, we used 10-fold cross-validation and did the tuning on the pipeline. When making predictions on the test set, the best model gives an accuracy of 0.7662.

- **Random Forest**
  Random Forest is an ensemble learning method which creates many decision trees and takes the aggregate of these trees to generate a final model and improve overall classification. It attempts to decorrelate each of the bootstrap trees by allowing each tree to use only a random subset of variables for splitting. For this pipeline, we only created a new variable called features by assembling all important features into one vector. We also used a 10-fold cross-validation to search the optimal values for hyperparameters numTrees, maxDepth and maxBins. The prediction accuracy for the model based on those optimal values is 0.7948.

- **Support Vector Machine**
  Support Vector Machine is a popular supervised machine learning technique. It is mainly used for binary classification based on the idea of a separating hyperplane. Regarding this pipeline, we first created a variable named unscaled_features to assemble all significant

predictors. Since the support vector machine algorithm is sensitive to features scaling, we standardized unscaled_features and created a new variable called scaledFeatures. Similar to other approaches we tried, we used 10-fold cross-validation to select the appropriate values for hyperparameters maxIter and regParam. The prediction on the test set based on the best model from cross-validation gave an accuracy of 0.8065.

Model Comparison

We evaluated our models using five different metrics. Accuracy is the ratio of the number of correct predictions to the total number of predictions. Precision is the ratio of correct positive predictions to the total predicted positives. Recall is the ratio of correctly predicted positive observations to the total number of observations that are actually in the positive class. F1 score is the weighted average of precision and recall. AUC is the area under the Receiver Operating Characteristic (ROC) curve which measures the models' ability to classify each observation into the correct groups. Accuracy, precision, recall and F1 score were calculated based on the confusion matrix. AUC is generated from BinaryClassificationEvaluator().

Based on the table below, we observed that the four advanced models we built all outperform the benchmark model. The Support Vector Machine has the best accuracy of 0.8065 and best precision of 0.5709. Decision Tree has the highest recall of 0.5992. Random Forest has the highest F1 score of 0.5456 and highest AUC of 0.7749. Since we expected to correctly predict whether a client will default on their credit card payment, we decided to use accuracy as the primary metric, and we concluded that our champion model is the Support Vector Machine model because it has the highest accuracy.

	Benchmark	**Logistic Regression**	**Decision Tree**	**Random Forest**	**Support Vector Machine**
Accuracy	0.6365	0.7718	0.7662	0.7948	0.8065
Precision	0.3071	0.4832	0.4750	0.5319	0.5709
Recall	0.5195	0.5391	0.5992	0.5601	0.4840
F1 Score	0.3860	0.5096	0.5299	0.5456	0.5239
AUC	0.6229	0.7254	0.6801	0.7749	0.7309

Sensitivity Analysis

We used F1 score to measure the sensitivity of our models to hyperparameter changes. For the Logistic Regression, the F1 score did not change too much for different values of regParam. For the tree-based model (i.e., Decision Tree and Random Forest), the F1 score for Decision Tree tended to decrease when the value for maxDepth became larger while the changes of maxDepth did not have a great impact on F1 score for Random Forest. Similar to Logistic Regression, the F1 score for Support Vector Machine was not sensitive to the changes of hyperparameter regParam as well. Hence, we summarized that Logistic Regression, Random Forest and Support Vector Machine have low sensitivity.

IV. Conclusions

The champion model with an accuracy of 0.8065 indicates that the model did a reasonably good job in predicting whether the client will default on their credit card payment. However, a relatively low precision, recall and F1 score did not come up to expectations. We thought this might be due to the imbalance characteristic of our data. Even though we balanced the training set by oversampling the minority class, the model still incorrectly predicted approximately half of the observations in the positive class as negative. In order to improve our model performance, we will consider attempting other data balancing methods such as synthetic minority over-sampling technique in our future research. In addition, we can also check the correlation between predictor variables and use principal component analysis to remove multicollinearity.

References

Joshi, R. (2016). Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog. Exsilio Blog. Retrieved March 15, 2021, from https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

ML Tuning: Model selection and hyperparameter tuning. (n.d.). Retrieved March 15, 2021, from https://spark.apache.org/docs/latest/ml-tuning.html

Tashman, A. (2021, January 25). Data Preprocessing. Lecture.

Wan, J. (2020, February 09). Oversampling and Undersampling With pyspark. Retrieved March 16, 2021, from https://medium.com/@junwan01/oversampling-and-undersampling-with-pyspark-5dbc25cdf253

Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.