Machine Learning Project Report: Loan Default Predictor

Alex Suarez and Lisbeth Santana

Introduction:

The mysteries of the loan market are run deep and wide. As banks and other financial institutions seek to minimize the risks they are taking when lending out money, loan applications get more and more complex. This is especially true when the amount of money being lent out increases significantly, e.g. mortgages. In this case, banks need to know that the risk they are taking will pay off. This is traditionally handled by not ceding complete ownership to the property until the loan is paid in full. However, the market crash of 2008 is proof that this is not a perfect system. In 2008, such a high number of people defaulted on their home loan payments that foreclosing their homes was not enough to support the losses these companies faced. It is believed this crash occurred because the requirements to give out mortgages were relaxed, which meant that loans were being given out to people that were not necessarily equipped to pay them back, which leads to the purpose of this project.

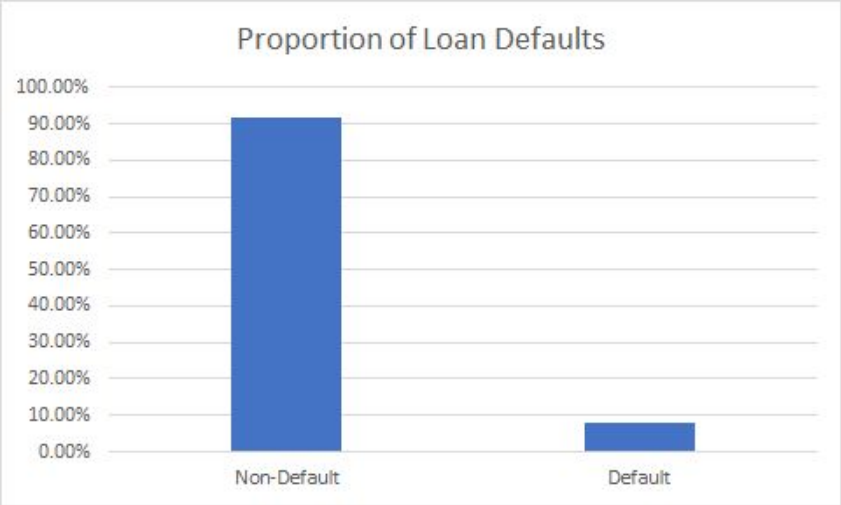
Using a data set issued by the Home Credit Group CITE, we want to determine if we can accurately predict whether a future customer will default on their home loan, by using features that are traditionally found in home loan applications. This question will be answered by using a binary classifier that simply determines if the customer is likely to default on their loan or not.

Hypothesis:
The Neural Network will outperform the SVM, Logistic Regression and Random Forests in the testing dataset in both accuracy and true positive rate (recall).

Data Exploration:

The data set was issued by the Home Credit Group. The original data set consisted of 308 thousand entries with 121 feature columns and one target variable. The target variable is 0 if the customer had no payment issues in the lifetime of the loan, and 1 if they did. This implies that a customer that had issues paying his or her loan is likely to eventually default on their loan. The feature columns consists of both numerical variables, e.g. income and age, and categorical variables, e.g. gender and employment type.

Default	Count	Percentage
Non-Default	282,686	91.93%
Default	24,825	8.07%
Grand Total	307,511	100.00%



Approximately 8% of the data was labelled 1. Clearly there is a class imbalance here. Thus in order to train a model on this data, we must pick training/testing data that are of relatively equal proportions.

Remove Null Values:
Approximately 50% of the initial 121 features contained empty row values, 51 had over 50% of their row values missing, so we removed those. These included information about the home of the customer such as the average square footage of their home, or the median building age of their apartment etc…

Remove imbalanced features:
Of the 71 features remaining, we investigated further to note any extreme feature imbalances. For example, the feature 'AMT_REQ_CREDIT_BUREAU_HOUR' which is the number of inquiries the bank made within an hour of the customer signing off on the loan was virtually all 0, or left blank, thus we removed it. Another set of features such as the 'FLAG_DOCUMENT_X', where X ranges from (2, …, 21) had mostly zeros too(over 90% across customers that defaulted or did not default), thus removed them from analysis. FLAG_CONT_MOBILE was removed because virtually all customers owned a cell phone.

Normalization of Numerical Data:
The final dataset consisted of 44 features, with 12 of them being numeric, eligible for normalization. These included: Days from birth, income, annuity value, etc.. Features that we did not normalize included: Hour of day the loan was initiated, customer rating (1-5), client city region rating (1-5).

One Hot Encoding:
The final step in our preprocessing was the One Hot Encoding of categorical data. This included: Day of the week, employment type (pension, employed, unemployed).

The final dataset contained 67 features and 245 thousand entries.

Methodology/Cross Validation:

Model Selection:
Multiple models were selected that have been known to work well with binary classification. The first one, our baseline model, is logistic regression. The simplicity of this model makes it preferable, unless the other more complex models outperform it significantly. The other models chosen are also known to perform well for classification problems. A Support Vector Machine that will seek to find a hypersurface that accurately separates the data into the two desired classes. A Random Forest Classifier that will find partitions in the features to determine the characteristics of each class. Finally, a neural network to determine if a deep learning algorithm can outperform the previous methods.

Balancing Training Data:
As we mentioned earlier, there was a pretty substantial class imbalance in our data (25,000 vs. 283,000) for 0 and 1 labelled data. So our strategy was the following:

- Create a training dataset for 3-Fold Cross Validation. This training dataset will have a 50/50 split between Non-Defaulters/Defaulters.
- Separate out a testing dataset with the same class distribution as the original dataset (appx. 90/10 split Non-Default/Default)

This resulted in our final data set having a training size of about thirty thousand entries and a testing size of sixty thousand.

Cross Validation:
The training dataset was further broken into 3 cross validation sets, of around ten thousand entries each. The best performing model was later retrained on the full training dataset. Accuracy and Recall were recorded for each training and testing dataset and the models were rated based on this criteria.

Performance Metrics:

The metrics used to compare the performance of the different models are going to be accuracy and recall. The accuracy will give us an idea of the overall performance of the model, however it may be misleading due to the class imbalance in our data set. Predicting all 0's would give us a 92% accuracy, but that is useless. Therefore, we are going to use the recall to supplement the accuracy because the purpose of our model is to correctly identify the "positives" or the customers who are likely to default. The ideal model would aim to maximize both metrics. Confusion matrices were generated on each of the left out sets of the cross validation set and the overall test set.

Results:

Logistic Regression:
The Logistic Regression was fit with L2 Regularization. The cross validation scores can be seen in the table below, as well as the metrics when the model was used on the entire data set. The model was optimized using the 'SAGA' solver which is a variation of stochastic gradient descent, and it ran for 5000 iterations.

Dataset	Train Accuracy	Train Recall	Test Accuracy	Test Recall
Cross-Validation	0.6830	0.6768	0.6808	0.6749
Full Dataset	0.6843	0.6774	0.6917	0.6631

Random Forest Classifier:

The Random Forest Classifier was trained until the number of samples in a leaf went below twenty. This prevented the model from memorizing the training set. The cross validation scores can be seen in the table below, as well as the metrics when the model was used on the entire data set.

Dataset	Train Accuracy	Train Recall	Test Accuracy	Test Recall
Cross-Validation	0.7464	0.7362	0.6742	0.6621
Full Dataset	0.7503	0.7359	0.6869	0.6530

Support Vector Machine:
Parameters:
Radial Basis Kernel, C=1.0 and gamma=0.1

Dataset	Train Accuracy	Train Recall	Test Accuracy	Test Recall
CV 1,2	0.9104	0.9065	0.6706	0.6570
CV 1,3	0.9085	0.9042	0.6716	0.6718
CV 2,3	0.9118	0.9073	0.6644	0.6464
Best Model (1,3)	0.8984	0.8945	0.71045	0.8206

Feed Forward Neural Network:
Parameters:
Hidden Layers: 3
Alpha: 0.001
Epochs: 1200

Dataset	Train Accuracy	Train Recall	Test Accuracy	Test Recall
CV 1,2	0.8766	0.8655	0.6583	0.6478
CV 1,3	0.8346	0.8232	0.7885	0.7917
CV 2,3	0.8647	0.8594	0.6578	0.7917
Best Model (1,3)	0.8517	0.8424	0.6900	0.7840

The Neural network underperformed the SVM, but outperformed the logistic regression and the random forest classifier. Let's recall that our baseline model was the logistic regression with an accuracy of 69.17% and a recall of 66.31%. Even though the accuracy of all the models seem to be around 68-71%, the recall increases significantly with the supper vector machine and the neural network, to 82.06% and 78.40% respectively. We believe this to be a significantly improvement when it comes to accurately identifying the customers who would be considered "risky" to a home loan lender.

Conclusion:

Our hypothesis proved to be incorrect in this experiment. A Support Vector Machine yielded the model with the highest testing recall over the test dataset. The Feed Forward Neural Network yielded the next best test Recall.

We learned a very practical way of performing a cross validation for selecting the best model in the classification problem. The logistic

Even though the random forest was at the same level of performance as the logistic regression, it allows us to take a look at what it considered to be the important features. If you take a quick scan you can see that education, employment and family status were flagged as important. This is a feature that could potentially be used to continue to reduce the dimensionality of the data. The best performing models could be trained on the features the random forest deemed as important and those results can be analyzed to see if there were any improvements.