Capstone Project

Machine Learning Nanodegree

Jorge Pinzón

November 2017

PROBLEM LOAN PREDICTION

Overview

Banks, and other financial institutions, make their profits by lending money (www.investopedia.com). Every time a loan is issued the institution takes a risk. Risks include the fact that some borrowers, or customers, will not return the money as agreed during the lending process. Lack, or delay, payments results in "Problem loans," or loans where the banks lose money as they cannot recover the original amount, or have to invest additional money during the repaying process (www.iedunote.com). According to the US Department of Treasury, banks deal with problem loans by: renewing or extending the loan terms, extending /adding credit, restructuring the loan, or foreclosure. Regardless of the direction taken, during this process, known as loan workout, the bank is forced to choose the alternative where the recovery is maximized and the risk and expenses are minimized (www.occ.treas.gov). In these circumstances, machine learning techniques and historical data can provide additional information during the initiation of the loan process in order to reduce the risk by identifying customers likely to generate a problem loan.

In this project, an Artificial Neural Network classifier is built to predict problem loans on data from the Lending Club dataset (https://www.lendingclub.com/info/download-data.action) corresponding to quarters 1 and 2 of 2017. The model is built in Python 3 using Keras with Tensorflow.

Objective

Generate a neural network model capable of efficiently predict if a loan given to a bank customer will become a problem loan or not.

Procedure

This project includes seven steps that will result in an efficient model for the prediction of problem loans (Figure 1).

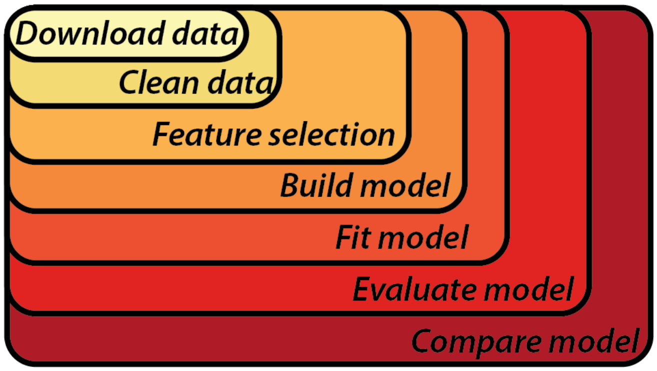


Figure 1 Schematics organization of the procedures taken in the creation of a predictive model for problem loans

Step 1. Data:

The data for this project comes from the Lending Club statistics for quarters 1 and 2 of 2017 (https://www.lendingclub.com/info/download-data.action). The data is divided in two files, one for each quarter. The files are loaded into Python separately and merge into a single data frame. The data contains a total of 139,919 loans with 145 columns of information. The columns include information on the loan type, amount, dates, and personal data on the credit history of the customer among others. More specific descriptions on all 145 columns can be found in the dictionary file: LCDataDictionary.xlsx.

Step 2. Cleaning:

An initial inspection of the data frame discovered two problems with the data that need to be addressed.

a. There are descriptions of the data in the last rows of the files. These were removed by sorting the data and deleting the last 8 rows, resulting in 139,311 loans

b. There are many cells with NaNs, with some columns having no information but NaNs. These columns were identified and removed by counting NaNs. Initially reducing the number of columns to 142 and then to 36, maintaining the number of NaN per column to a maximum of 10.

Step 3. Feature selection:

Target feature
In the data set, the target feature is loan_status. This variable contains nine different classifiers that can be grouped into three main categories: Current, Fully paid, and Problem loan.

Current loans are current, they do not have any problem and therefore were remove from the dataset. Fully paid loans are those that have already been paid in full. The caveat here is the assumption that they did not showed any problems or delays before payment. Problem loans group all loans that have a problem, including Default, Grace Period, Do not meet credit policy, and/or Charged off.

After filtering status_loan, the total of loans in the dataset has 58,006 loans.

Other features
The other 35 features (excluding status_loan) were classified into: non-numeric/categorical and numeric features.

There are 18 non-numerical variables. From these, zip_code, sub_grade, and initial_list_status, were removed from the data as zipcode included only 877 of the 43,000 in the USA and the information they may provide to the model might be contained in the state column. Subgrade is a subcategory of grade, and initial_list_status was not complete and not clearly described.

Interest rate (int_rate) and employment length (emp_lengh) had string characters in most of the cells, these were replaced/removed and the column converted into a numeric variable.

Finally, there are two columns, issued date (issue_d) and last credit check (last_credit_pull_d) that have dates. Both are formatted as Month-year. Data in these columns were converted into delta time in days using the latest date as the maximum date. After this transformation, both variables became numeric. Leaving a total of 11 categorical variables that have more or less good representation across the categories of the loan status variable (Figure 2)

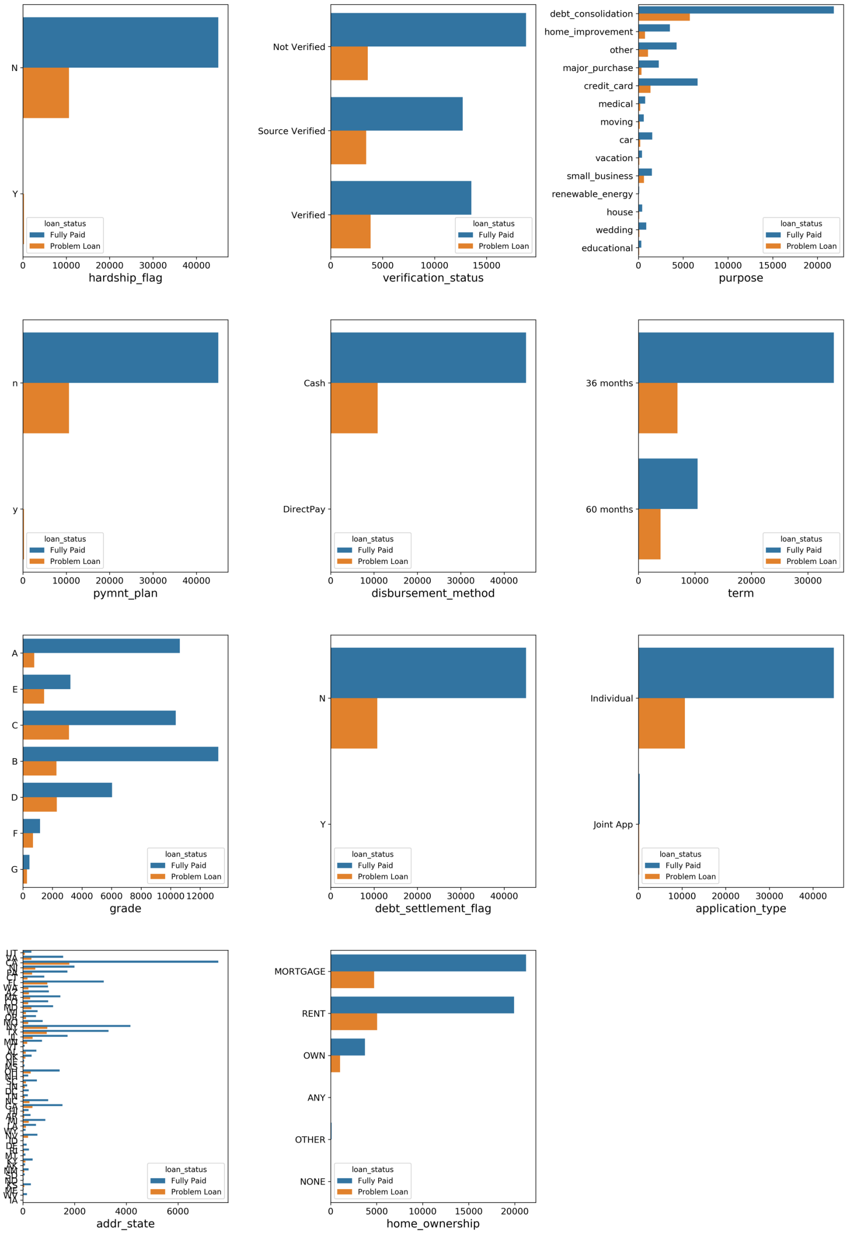
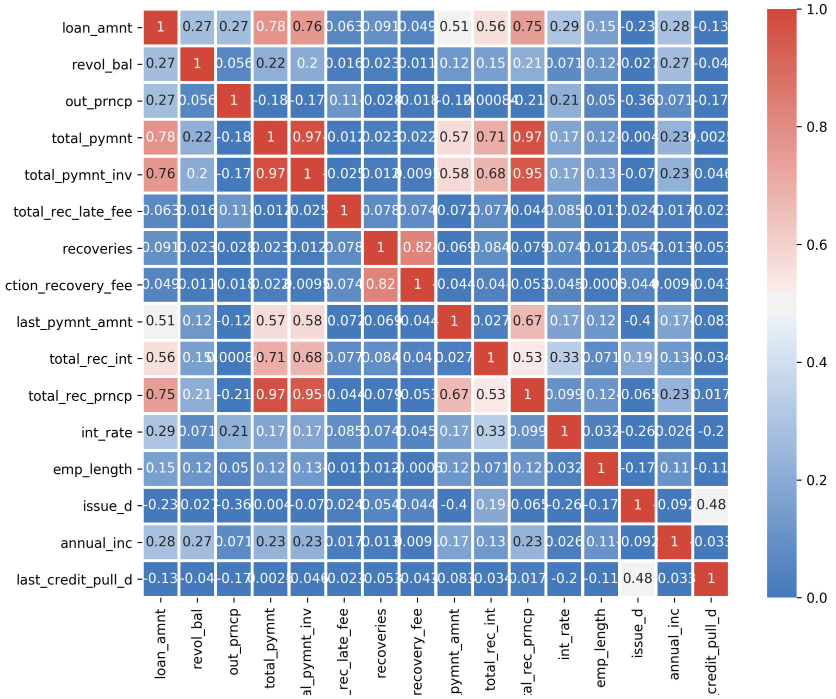


Figure 2. Bar plots showing the counts of fully paid and problem loans for each category in the categorical variables from the 2017 first two quarters of data.

Originally the number of numeric variables was 17, but with the conversion of categorical variables, the total increased to 21. Of these 21 columns, policy_code had only one value, and installment, funded_amnt, funded_amnt_inv, and out_prncp_inv corresponded to similar types of data (loan_amnt and out_prncp), all of them removed from the dataset.



	loan_amnt	revol_bal	out_prncp	total_pymnt	al_pymnt_inv	_rec_late_fee	recoveries	recovery_fee	pymnt_amnt	total_rec_int	al_rec_prncp	int_rate	emp_length	issue_d	annual_inc	credit_pull_d
loan_amnt	1	0.27	0.27	0.78	0.76	0.063	0.091	0.049	0.51	0.56	0.75	0.29	0.15	-0.23	0.28	-0.13
revol_bal	0.27	1	0.056	0.22	0.2	0.016	0.023	0.011	0.12	0.15	0.21	0.071	0.12	-0.027	0.27	-0.04
out_prncp	0.27	0.056	1	-0.18	-0.17	0.11	-0.028	0.018	-0.12	0.00084	-0.21	0.21	0.05	-0.36	0.071	-0.17
total_pymnt	0.78	0.22	-0.18	1	0.97	-0.012	0.023	0.022	0.57	0.71	0.97	0.17	0.12	-0.004	0.23	0.0025
total_pymnt_inv	0.76	0.2	-0.17	0.97	1	-0.025	0.012	0.0095	0.58	0.68	0.95	0.17	0.13	-0.07	0.23	-0.046
total_rec_late_fee	0.063	0.016	0.11	-0.012	0.025	1	0.078	0.074	0.072	0.077	0.044	0.085	0.011	0.024	0.017	0.023
recoveries	0.091	0.023	0.028	0.023	0.012	0.078	1	0.82	-0.069	0.084	0.079	0.074	0.012	0.054	0.013	0.053
ction_recovery_fee	0.049	0.011	0.018	0.022	0.0095	0.074	0.82	1	-0.044	0.04	-0.053	0.045	0.0006	0.044	0.009	0.043
last_pymnt_amnt	0.51	0.12	-0.12	0.57	0.58	0.072	0.069	0.044	1	0.027	0.67	0.17	0.12	-0.4	0.17	-0.083
total_rec_int	0.56	0.15	0.00084	0.71	0.68	0.077	0.084	0.04	0.027	1	0.53	0.33	0.071	0.19	0.13	-0.034
total_rec_prncp	0.75	0.21	-0.21	0.97	0.95	-0.044	0.079	0.053	0.67	0.53	1	0.099	0.12	-0.065	0.23	0.017
int_rate	0.29	0.071	0.21	0.17	0.17	0.085	0.074	0.045	0.17	0.33	0.099	1	0.032	-0.26	0.026	-0.2
emp_length	0.15	0.12	0.05	0.12	0.13	-0.011	0.012	0.0006	0.12	0.071	0.12	0.032	1	-0.17	0.11	-0.11
issue_d	-0.23	0.027	-0.36	0.004	-0.07	0.024	0.054	0.044	-0.4	0.19	-0.065	-0.26	-0.17	1	-0.092	0.48
annual_inc	0.28	0.27	0.071	0.23	0.23	0.017	0.013	0.0091	0.17	0.13	0.23	0.026	0.11	-0.092	1	-0.033
last_credit_pull_d	-0.13	-0.04	-0.17	0.0025	0.046	0.023	0.053	0.043	0.083	0.034	0.017	-0.2	-0.11	0.48	0.033	1

Figure 3 Correlations between numerical variables in the 2017 Loan Lending dataset

A correlation analysis between numerical values did not indicate any concerning correlation between any pair of variables (Figure 3). There are however variables that appear to have several outliers (Figure 4 - left).

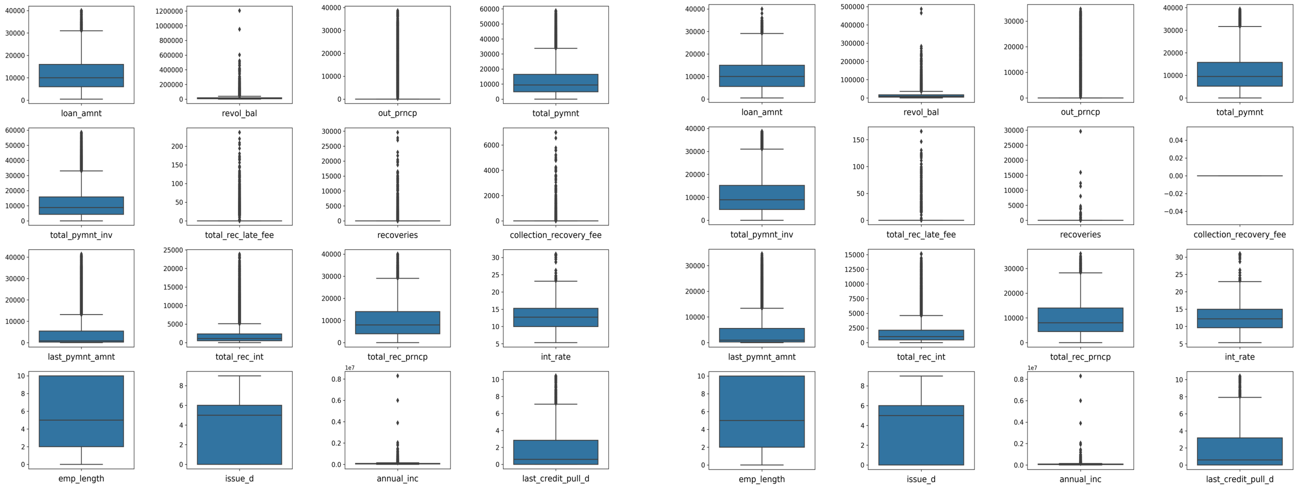


Figure 4 Distribution of the data in the numeric variables before (left) and after (right) outlier removal.

A total of 8383 rows contained data identified as outliers across all columns. These data points were eliminated. The whole dataset after this step contained 47,505 loans and 28 features.

As a final step in feature selection, all 11 categorical variables were encoded into numbers using LabelEncoder from sci-kit learn. Similarly loan_status was also encoded.

Step 4. Model construction:

Before building the model, the data was split into target variable and features, and then each of these portions divided into training (75%), validation (20%), and test (5%) subsets.

The raw classifier model, an artificial neural network, consisted of: one input layer, one hidden layer, and one output layer. All layers being regular densely-connected layers uniformly initiated. The input and hidden layers use the relu activation function and 14 units. The output layer is activated by a sigmoid function and uses a single unit. The network compiler uses the adam function as optimizer, binary_crossentropy for the loss, and set the accuracy as the metrics for the model.

Step 5. Fitting:

After compiling, the model was fitted with the training set and run for 10 epochs, with a batch size of 10. This initial run had an accuracy of 99.7%.

Step 6. Evaluation and optimization:

Evaluation of the model was performed using both accuracy and F1-score. Accuracy is defined as the total number of correct predictions divided the total number of predictions. The F-score provides a balance between precision and recall complementing both measures, and reducing the accuracy paradox.

The model was validated with the validation test, resulting in a 99.74% accuracy. This result combined with the accuracy during fitting may be the results of overfitting. Before continuing with testing on the test set, the model was tested on other datasets. First, the classifier was used on data from the 4th quarter of 2016 (2016_pred.ipynb). This test resulted in an accuracy of 98.49% (20,005 loans). The model was also tested in a dataset that included loans from 2007-2015 and is compiled in kaggle (https://www.kaggle.com/wendykan/lending-club-loan-data). This time the model showed an accuracy of 99.824% (2007-2015_pred.ipynb). It is very unlikely that the model is over fitted given that these three datasets are different.

An optimization of the model using grid earch to iterate through different parameters suggested to change the model by adding 10 extra epochs during fitting. This resulted in an accuracy of 99.6389% which is not higher than the original, but not that different either. This classifier had an F1-score of 0.9837, only misclassifying 43 loans from the test set (11,877 loans - Figure 5).

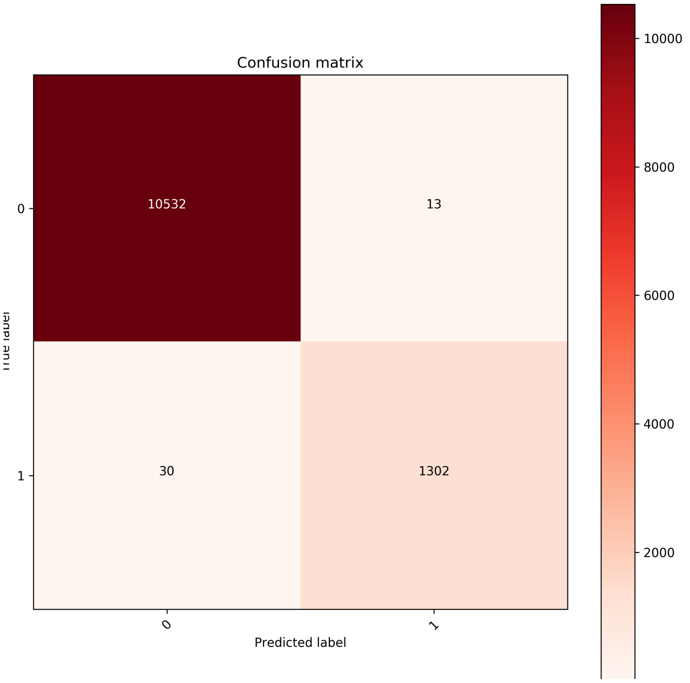


Figure 5. Confusion matrix of the 2017 dataset predictions.

Step 7: Comparison:

In order to compare the performance of the proposed ANN classifier, two additional models were selected from a kaggle dataset This models are labelled **Kaggle-1** (https://www.kaggle.com/mina20/building-a-neural-net-to-predict-default/notebook) and **Kaggle-2** (https://www.kaggle.com/gagrawal/neural-net-with-keras) for the purpose of this project.

Table 1 Summary table with the comparisons between the proposed model and the models from two models uploaded in kaggle.

	Kaggle-1	***Kaggle-2***	***ANN Proposed here***
Model	DNN Regressor	Classifier 3 layers	Classifier 3 layers
Library	Tensorflow	Keras-tensorflow	Keras-tensorflow
No. loans tested	37,379	1,000	232,650
No. features	78	68	23
Loss during fitting	0.025	1.12	0.0072
Accuracy on test dataset	94.47	93.0	99.8
F1 Score	Did not predict Problem loans 1 in Figure 6A	Did not predict Problem loans 0 in Figure 6B	0.9943

Kaggle-1 and kaggle-2 predict default loans on Lending Club data from 2007 to 2015 and both also use neural networks as their preferred method.

Kaggle-1 uses tensorflow with a DNNRegressor on 78 features and 37,379 loans. The original script only reports the loss during fitting with a value of 0.0253. The accuracy measured here as number of correct predictions over total number of predictions, was 94.47%. This model was not able to identify problem loans, therefore it was not possible to determine the F1-score (Table1, Figure 6A).

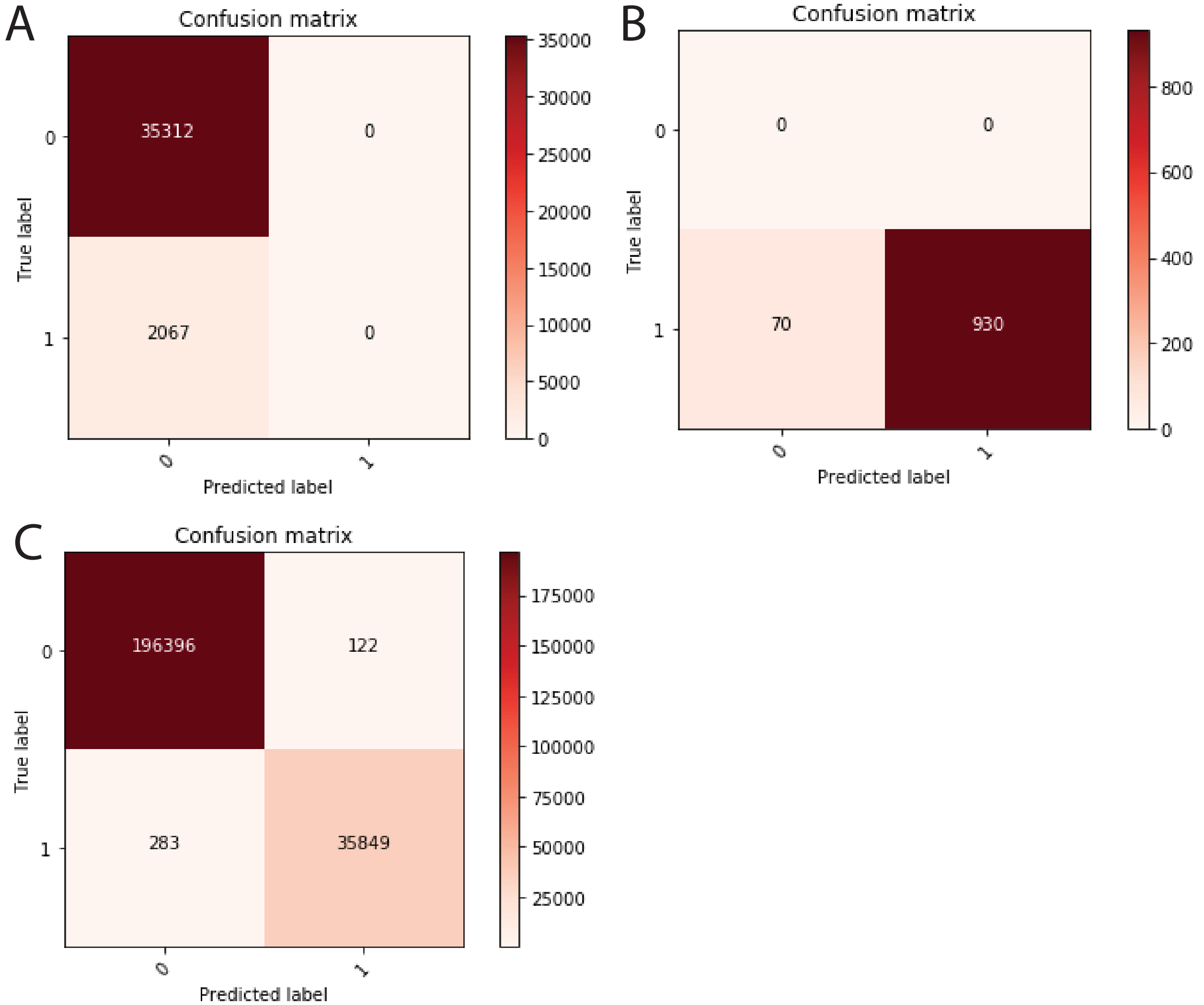


Figure 6 Confusion matrices for kaggle-1 (A), kaggle-2 (B), and the ANN proposed in this project (C) using the 2007-2015 dataset.

Kaggle-2 uses keras on tensorflow with a 3-layer classifier network using 68 features on 1,000 loans. The loss during fitting is 1.4029 and the testing accuracy was 91.2%. This model did not predict problem loans, and the F1-score was not determined (Table 1, Figure 6B).

The ANN model proposed here is also based on keras and tensorflow with 3-layers, but it uses 23 features and all the loans in the dataset (232,6560). This model has an accuracy of 99.8%, loss of 0.0072, and predicted both problem and paid loans with an F1-score of 0.9943 (Table 1, Figure 6 C).

The proposed model outperforms the other two by 5-7 points in terms of accuracy and predicts both categories. The outperformance could be the result of:

a. Cleaning data
b. Labeling of the categorical features.
c. The number of selected features.
d. The standardization of the features before the model.

Kaggle-1 uses a different structure to that in the current model, making the comparison slightly difficult. On the other hand, kaggle-2 and the ANN proposed here have the same 3-layer structure but the labelling is different, and in kaggle-2 the number of selected features is higher, and it lack feature standardization. Combined these differences may account for the differences in accuracy seen in the models. Models using traditional machine learning algorithms result in lower accuracies, as shown by at least one project that used R and several models to predict problem loans in a dataset from the Lending Club (https://rstudio-pubs-static.s3.amazonaws.com/203258_d20c1a34bc094151a0a1e4f4180c5f6f.html)

Conclusion

The model proposed here is highly effective, with accuracies higher than 99.5% in different datasets, at predicting both paid and problem loans from the Landing Club dataset. It outperformed other models, likely as a results of improved data cleaning, encoding, and standardization, as well as the use of a smaller set of features.