Predicting loan repayment

Name: Lúcia Moreira

Date: April, 13th, 2020

Summary

A binary classification problem is addressed for predicting loan repayment. Logistic regression, Random Forest (RF), Naïve Bayes and Neural Networks (NN) were considered. Hyperparameters for the RF and NN algorithms were selected using 'hyperopt' library. Results show the best classifiers were RF, logistic Regression and NN providing an area under the curve (AUC) from the ROC curve of 0.68 for all the three of them. Similar AUC were obtained in datasets similar to this one when comparing to previous studies.

Dataset

The chosen dataset concerns Predicting loan repayment. A publicly available dataset was retrieved from LendingClub.com, a website that connects borrowers and investors over the Internet. The present dataset represents 9 578 3-year loans that were funded through the LendingClub.com platform between May 2007 and February 2010.

In the lending market, investors provide loans to borrowers in exchange for the promise of repayment with interest. If the borrower repays the loan, then the lender profits from the interest. However, if the borrower is unable to repay the loan, then the lender loses money. Therefore, lenders face the problem of predicting the risk of a borrower being unable to repay a loan.

The binary dependent variable notFullyPaid indicates that the loan was not paid back in full (the borrower either defaulted or the loan was "charged off," meaning the borrower was deemed unlikely to ever pay it back). To predict the dependent variable, the following 13 independent variables available to the investor when deciding whether to fund a loan will be used:

1) creditPolicy: 1 if the customer meets the credit underwriting criteria of LendingClub.com, and 0 otherwise.
2) purpose: The purpose of the loan (takes values "credit_card", "debt_consolidation", "educational", "major_purchase", "small_business", and "all_other").
3) intRate: The interest rate of the loan, as a proportion (a rate of 11 % would be stored as 0.11). As judged by LendingClub.com, riskier borrowers are assigned higher interest rates.
4) instalment: The monthly instalments ($) owed by the borrower if the loan is funded.
5) logAnnualInc: The natural log of the self-reported annual income of the borrower.
6) dti: The debt-to-income ratio of the borrower (amount of debt divided by annual income).
7) fico: The FICO credit score of the borrower.
8) daysWithCrLine: The number of days the borrower has had a credit line.
9) revoBal: The borrower's revolving balance (amount unpaid at the end of the credit card billing cycle).
10) revolUtil: The borrower's revolving line utilization rate (the amount of the credit line used relative to total credit available).
11) inqLast6mths: The borrower's number of inquiries by creditors in the last 6 months.

12) delinq2yrs: The number of times the borrower had been 30+ days past due on a payment in the past 2 years.

13) pubRec: The borrower's number of derogatory public records (bankruptcy filings, tax liens, or judgments).

Despite missing values were small (0.15 %), i.e. dataset is not sparse, rows with two or more missing values were removed from the data and for the others an imputation of the missing values was performed when there was a missing value. One-hot encoding was performed to the categorical variable 'purpose' by creating a 1-0 array with one column per categorical value, and 1's indicating rows belonging to that category. After one-hot encoding we have 19 final features.

Analysis of the data set indicated that a slightly unbalanced dataset is observed with 16 % of the instances classified as 1, i.e, loan was not fully paid. Dataset was divided into training (80 %) and test (20 %). The option 'stratify' in 'train_test_split' was used for keeping the same proportion of elements of each target class in the training and test sets. Numeric features were divided into continuous and integer; in order to allow better scaling and further imputing process.

Classification problem modelling

Logistic regression

Logistic regression provided a test accuracy of 83 % i.e. provided the same as random guess. So accuracy is not a good performance measure for this problem. So it was used instead the ander under the ROC curve. Fig. 1 shows the ROC curve for the logistic regression fitting. An area of 0.68 can be observed.

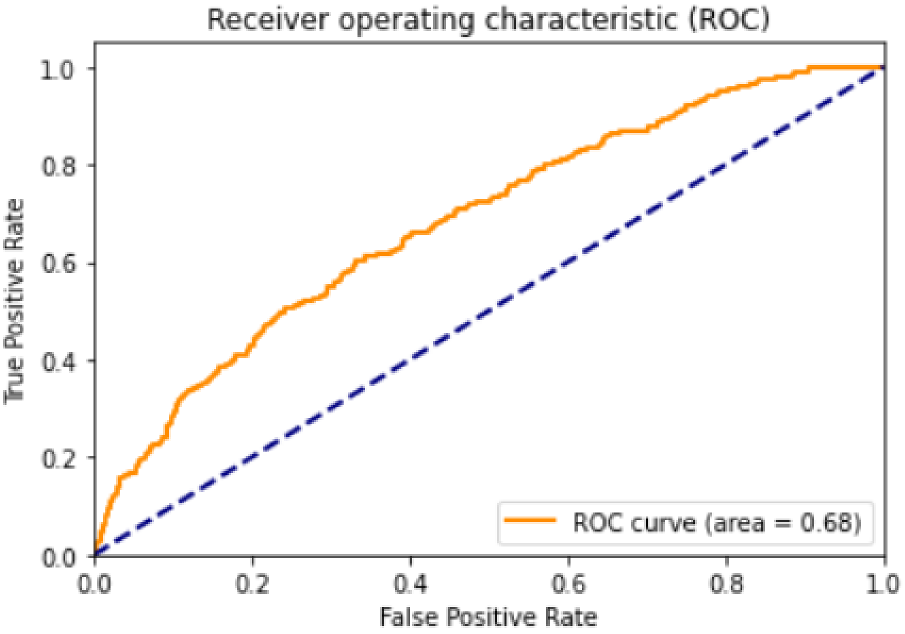

Fig. 1- ROC plot for the Logistic Regression algorithm.

Random Forest

Fig. 2 shows the ROC plot for the RF approach. The hyper parameter optimization using the 'hyperopt' library did not show any further improvement (data not shown) on ROC area. The optimized parameters were loss function, the maximum depth and the number of trees over 50 hyperopt evaluations. Moreover, one can see that RF performs similarly to the Logistic regression. Fig. 3 shows the features importance, indicating that the most relevant features for the classification problem are the monthly instalments owed, the number of days the borrower has had a credit line, the amount unpaid at the end of the credit card billing cycle, the amount of the credit line used relative to total credit available, the annual income of the borrower, the debt-to-income ratio and interest rate.

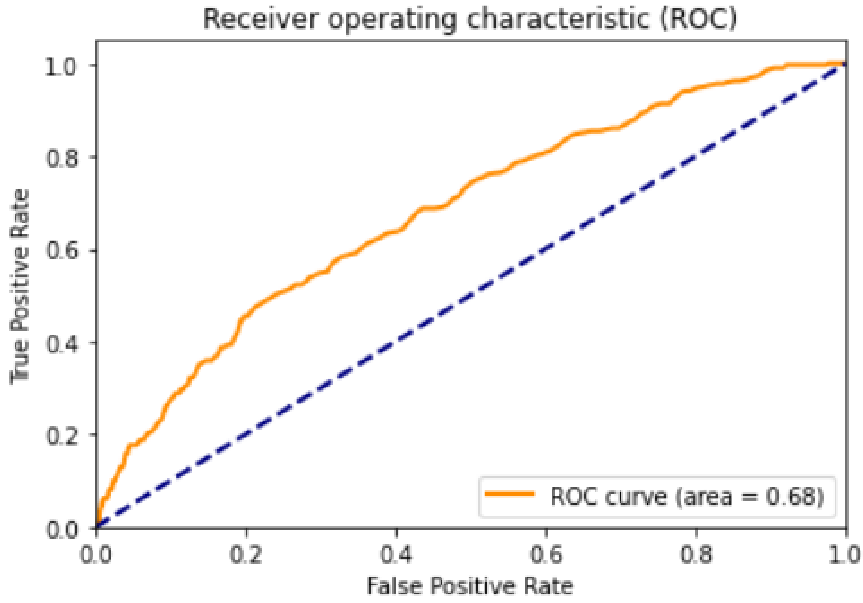

Fig. 2 – ROC curve for the RF algorithm.

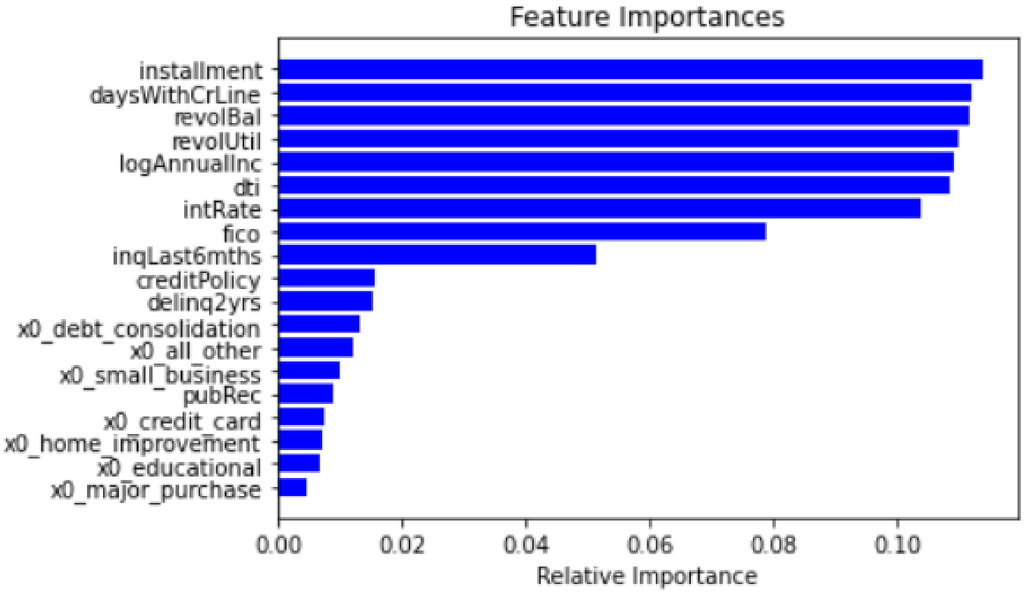

Fig. 3 - Most important features from the RF algorithm.

Naïve Bayes

Fig. 4 shows that Naïve Bayes algorithm performed slightly worse than Logistic Regression and RF, as observed by the ROC curve area= 0.67.

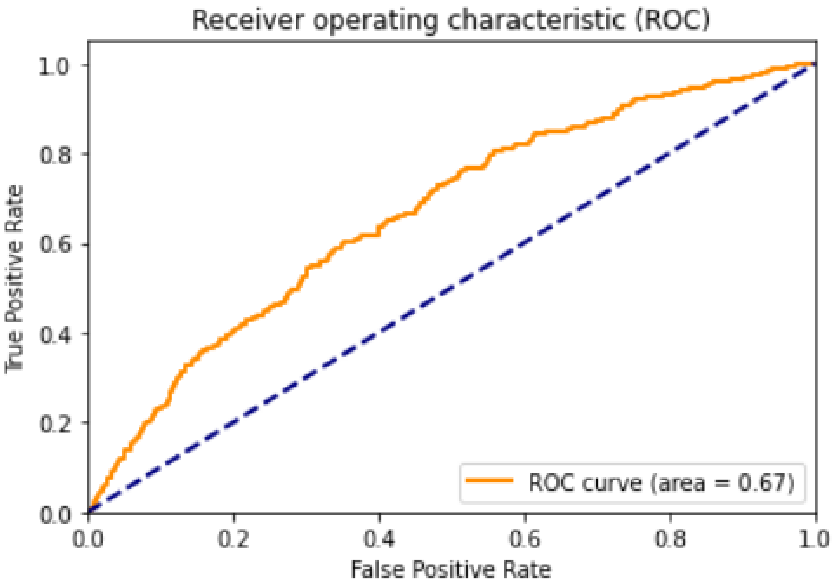

Fig. 4– ROC curve for the Naïve Bayes algorithm.

Neural networks

Full model considered 3 hidden layers and dropout as a regularization approach. Hyperopt output helped on the choice of the relevant parameters. The optimized parameters included the number of units in each hidden layer, batch size, number of epochs and the dropout regularization in each layer by using 100 hyperot evaluations. Indeed, the best result was obtained for 64 units in the first layer with a 0.9 rate regularization, 128 units in the second layer and also a 0.9 rate in the dropout regularization, a batch size of 16 and 100 epochs. A single hidden layer or 3 hidden layers provided slightly worse results. A binary_crossentropy loss function was considered, using the 'adam' optimizer and a sigmoid activation output function. PReLU was used as activation layer in each hidden layer and initialization was 'glorot_normal'. Fig. 5 shows that the NN analysis provided similar results as logistic regression and RF.

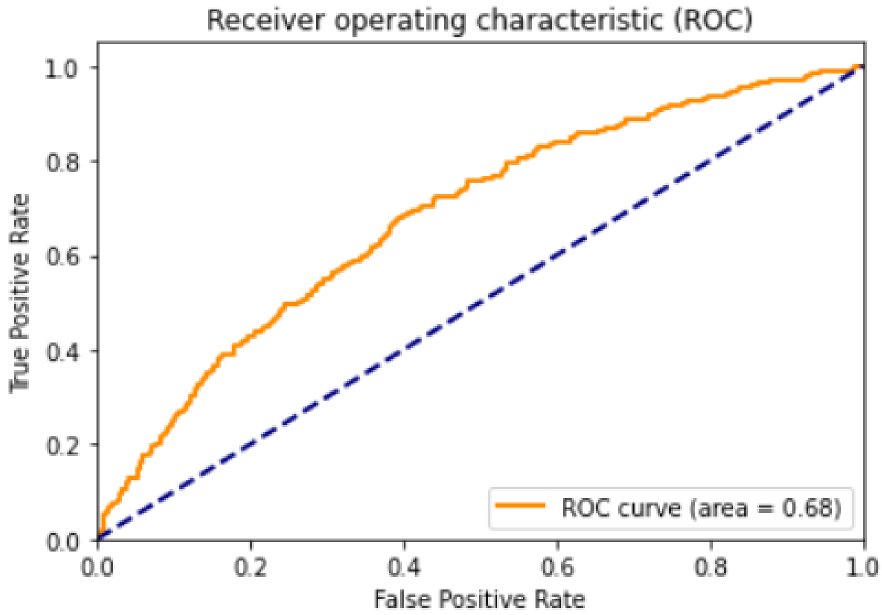

Fig. 5– ROC curve for the Neural network algorithm.

The maximum area under the RUC curve compare with previous studies on machine learning using a dataset obtained from the LendingClub's website as well [1].

Reference

[1]-https://rstudio-pubs-static.s3.amazonaws.com/203258_d20c1a34bc094151a0a1e4f4180c5f6f.html