95-828: Machine Learning for Problem Solving **Spring 2019**

Case Study Phase 3: Predictive Modeling

Instructor: Leman Akoglu

- In this Phase of the Case Study (CS), you will first employ predictive analytics to predict how likely a loan is to default. Then, you will employ predictive models for developing various investment strategies.

- You are expected to submit two files via *Canvas*:

  (1) An assignment *write-up*, including answers to the questions in this sheet, should be submitted as **.pdf**, and
  (2) An iPython *notebook*, including all your code, visualizations, and outputs, should be submitted as **.ipynb**.

  Your write-up can be hand-written, but should be clearly legible—if the TAs cannot read your solutions, you will not be given credit. Submissions can also be written in Word or LaTeX.

- The assignment is due at **11:59 PM on Friday May 3, 2019**.

- If you have any questions, please use *Piazza* or visit course staff during office hours and recitations.

- Do not copy from other sources, share your work with others, or search for solutions on the web. Plagiarism will be penalized according to the university rules.

3.1 Predictive Models of Default

1. Using the data you prepared in Phase II, implement machine learning models to predict the probability that each loan defaults. You may want to try the following models: Decision Tree (DT), Random Forest (RF), (L1 and L2 regularized) Logistic Regression (LogR), Naïve Bayes (NB), and (multi-layer) Neural Network (NN). (Feel free to use others as well.) As baseline, include the Random classifier.

   Carefully explain:

   (*i*) How did you set up your model training and evaluation?
   (*ii*) Which model hyper-parameters did you tune (for each model)?
   (*iii*) Which performance measure(s) did you use? Report your evaluation results.

2. In splitting the data into training and test sets, there are (at least) two possible ways you could go about:

- **Random:** Randomly assign each loan to a training set or a testing set. Within the training set, randomly assign loans to one of the folds for k-fold cross-validation.
- **Temporal:** Set a "cut-off" date. Assign all loans that were issued *before* that date to the training set, and all loans *after* that date to the evaluation/test set. Within the training set, create folds using a sliding time window in a similar manner for k-fold cross-validation.

(You possibly used Random in part 1., which is alright.) What are some advantages and disadvantages of using these data splitting procedures?

3. After training and evaluating your models, you realized that the features you used in your models were not all underlying facts about the loan applicants, but possibly *statistics calculated by LendingClub* using its own models. Now you want to assess whether the predictive power of your models came simply from LendingClub's own predictors. Carry out this investigation:

   (*i*) Provide a list of aforementioned features that are derived by LendingClub and any other features that correlate/reflect those.

   (*ii*) Fit a (L1 or L2 regularized) Logistic Regression model using *only one* of the features you identified in (*i*). What is the predictive power as compared to that for the models you trained in part 1.?

   (*iii*) Remove the features you identified in (*i*), refit your models onto remaining features and report new performance measure(s). What are your conclusions?

   Note: To ensure robustness of the results and avoid a single unfortunate split, it is advisable to attempt the modeling over many different train/test splits. The code provided to you allows you to use different seeds to generate these random train/test partitions. Generate 100 independent train/test splits with different seeds and report *average* performance values along with *standard deviation* for each model. Use the format (avg_perf, ±std).

4. Moving forward, pick the best-performing model in part 3. We will refer to it as YourModel. After modifying YourModel to ensure you did not include any features calculated by LendingClub, you want to assess the extent to which YourModel's scores agree with the grades assigned by LendingClub. How can you go about doing that? What is your observation?

5. Next you will assess the stability of YourModel over time. To this end, analyze whether YourModel trained (using the **Random** data splitting procedure in part 2. for cross validation) in 2009 performs worse in 2017 than YourModel trained on more recent data in 2016. What conclusion can you draw? Is your model stable?

6. Now go back to the original data (before cleaning and feature selection) and fit YourModel to predict the Default likelihood using *all of the features*. (For the sake of simplicity, it will be sufficient to limit yourself to the following features: `id, loan_amnt, funded_amnt, funded_amnt_inv, term, int_rate,`

`installment, grade, sub_grade, emp_title, emp_length, home_ownership, annual_inc, verification_status, issue_d, loan_status, purpose, title, zip_code, addr_state, dti, total_pymnt, delinq_2yrs, earliest_cr_line, open_acc, pub_rec, last_pymnt_d, last_pymnt_amnt, fico_range_high, fico_range_low, last_fico_range_high, last_fico_range_low, application_type, revol_bal, revol_util, recoveries.)`

Does anything surprise you about the performance of this model (averaged on out-of-sample test datasets) compared with the other models you have fit earlier?

3.2 Investment Strategies

You are finally ready to start building investment strategies. It is worth noting that there are many potential strategies you could try. Here we ask you to consider four strategies, but you are encouraged to try others. The four strategies here are as follows:

a. **Random** (Rand): randomly picking loans to invest in

b. **Default-based** (Def): using YourModel from previous section; particularly, sorting loans by their estimated Default likelihood, and selecting the ones with the lowest likelihood to invest in.

c. **Return-based** (Ret): training a simple regression model (e.g., linear, random forest, NN regressor) to predict the (calculated) return on historical loans directly. Then, sorting out-of-sample loans by their predicted returns and selecting the loans with the highest predicted returns to invest in.

d. **Default-& Return-based** (DefRet): training two additional models: one to predict the return on loans that did *not* default, and another to predict the return on loans that *did* default. Then, using the likelihood of Default predicted by YourModel from previous section to find the *expected value* of the return from each future loan, and investing in the loans with the highest expected returns.

Note: Even though LendingClub allows investors to invest in fractions of loans (by purchasing any number of *notes*), for simplicity we assume you choose loans to invest in *fully*.

7. First, build the three regression models described above: (1) regressing against all returns, (2) regressing against returns for defaulted loans, and (3) regressing against returns for nondefaulted loans.

   In each case, use each one of the four return variables you calculated in Phase II as your target variable (recall $M1$, $M2$, $M3(2.4\%)$, and $M3(6\%)$) and try (L1 and L2 regularized) linear regression, random forest regression, and multi-layer NN regression.

   Report the performance results in corresponding entries in Table 3.1. Do they perform well? Can you tell?

	Performance for each return calculation			
Model	$M1$	$M2$	$M3$ (2.4%)	$M3$ (6%)
L1 regressor				
L2 regressor				
Neural Network regressor				
Random Forest regressor				

Table 3.1: Table to be filled based on Q7.

8. Next, implement each of the investment strategies described above using the best performing regressor from part 7.

   (*i*) Suppose you were to invest in 1000 loans using each of the four strategies, what would your returns be? Average your results over 100 independent train/test splits.

   (*ii*) Include the best possible solution (denoted Best) that corresponds to the top 1000 performing loans *in hindsight*, that is, the best 1000 loans you could have picked.

   Fill in the corresponding entries in the table below.

	Return calculation			
Strategy	$M1$	$M2$	$M3$ (2.4%)	$M3$ (6%)
Rand				
Def				
Ret				
DefRet				
Best				

   (*iii*) Based on the above table, which data-driven investment strategy performs best? What can you tell about using the **Random** strategy? Does it cause you any loss? Why do think that is the case? How do the data-driven strategies compare to **Random** as well as BEST?

9. The strategies above were devised by investing in top 1000 loans. You are worried, however, that if you wanted to increase the number of loans you wished to invest in, you would eventually “run out” of good loans to invest in. Test this hypothesis using the best-performing data-driven strategy from part 8.

   Specifically, plot the return (using the $M1$ return calculation, averaged over 100 runs) versus your portfolio size (i.e., number of loans invested in). What trend do you observe? Why do you think that is the case?

Acknowledgements

This Case Study is inspired by and based on the following:

- Provost F, Fawcett T. (2013) *Data Science for Business: What you need to know about data mining and data-analytic thinking.* Sebastopol, CA: O'Reilly Media, Inc.
- Cohen MC, Guetta CD, Jiao K, Provost F. (2018) *Data-driven investment strategies for peer-to-peer lending: a case study for teaching data science.* Big Data 6:3, 191–213, DOI: 10.1089/ big.2018.0092.