Assignment 6.2: Building a Credit Risk Model for Consumer Lending

Submission Details

- Submit through Canvas (Deadline: Nov.12)
- Use R Markdown for Assignment 6.2
- You have to submit ONLY
  - RMD file
  - Output in PDF format (Hide Code)
  - You don't need to submit any datasets
  - Short and concise explanation (1-3 lines) for the specification used in the regressions (economic rationale for each variable and expected sign). For example, I chose *credit utilization rate* as one of the explanatory variables because high utilization rate implies that a consumer tend to take big risks and higher utilization rate is expected to result in a higher (+) likelihood of default.
  - Do not run the regressions first, see the sign and then input this sign in the writeup. Provide an economic explanation for why the variable should matter for default prediction and how (positive or negative) it would affect the default likelihood.

Assignment Tasks

1. Come up with a list of possible covariates that should matter for default prediction. Some possible sources are

    - Lecture on credit scoring
    - Data Dictionary
    - Lending Club
    - Prosper
    - Lend Academy
    - Nickel Steamroller
    - Guide to Credit Scoring in R
    - End to end Logistic Regression in R
    - How to Perform a Logistic Regression in R
    - `https://cran.r-project.org/web/packages/survminer/vignettes/Informative_Survival_Plots.html`
    - `https://rviews.rstudio.com/2017/09/25/survival-analysis-with-r/`
    - Cox regression in R
    - `https://www.r-bloggers.com/cox-proportional-hazards-model`

2. Compute these explanatory variables and calculate their descriptive statistics (n, mean, standard deviation, min, p25, p50, p75, max) over all sample period

3. Conduct a Survival Analysis with R packages "survival" and "survminer". Follow these steps

- Use variable Default to identify charged-off loans from the entire sample of loans issued in 2007-2014 and calculate their survival time (month) based on variable issue_d and last_pymnt_d
- Plot the Kaplan-Meier survival curve and produce the risk table (use ggsurvplot() function)
- Select a variable (either categorical or binned variable) to divide the sample into multiple groups and generate the survival curve and risk table for each group (put them in one chart)

4. Fit a Cox Proportional Hazards Model with function coxph() in survival package. Follow these steps

   - Use variable Default to identify charged-off loans from the entire sample of loans issued in 2007-2014 and calculate their survival time (month) based on variable issue_d and last_pymnt_d
   - Fit a hazard model with the selected covariates and plot the estimated distribution of survival times
   - Select a variable (either categorical or binned variable) to divide the sample into multiple groups and generate the estimated distribution of survival times for each group based on the fitted hazard model (put them in one chart)

5. Do a *in-sample* estimation and prediction using Logistic regression model. Follow these steps

   - Use the entire sample of loans issued in 2007-2014 for estimation

- Run a Logistic regression model with Default as the LHS variable and the variables in step 1 as explanatory variables
- Present the output and fit statistics for the model (output of function summary())

6. Do a *out-of-sample* prediction using Logistic regression model. Follow these steps
    - Divide the sample into in-sample training data (loans issued in 2007-2013) and out-of-sample testing data (loans issued in 2014)
    - Estimate the model with 2007-2013 data
    - Forecast default for out-of-sample testing data using the model estimates from in-sample training data and explanatory variables from testing data
    - Rank the default probabilities into deciles (10 groups). Use function gains()
    - Compute the number (and percentage) of defaults in each of the 10 groups for testing data (loans issued in 2014)
    - A good model is one that has the majority of defaults in decile 1 or 2 and very few in other deciles
    - Plot the ROC curve and calculate AUC and KS statistics
7. iterate steps 1, 2 and 6 till you get a model with good out-of-sample performance
8. Based on the practice from previous steps, do a *out-of-sample* prediction using Post LASSO Logistic regression. Follow these steps

- Run a Logistic LASSO regression on the in-sample training data to automatically select a good combination of covariates using glmnet() function (You may throw all relevant variables into the model and let it pick the good covariates for you)
- Calibrate the hyperparameter λ in Logistic LASSO regression using in-sample data to find the optimal value which gives you the best model performance
- After building the best Logistic LASSO regression (with the optimal λ), use the good covariates selected to fit a standard Logistic regression on the in-sample training data (this is so called "Post LASSO" Logistic regression)
- Forecast default for out-of-sample testing data using the Post LASSO Logistic regression model estimates from in-sample training data (same logic as step 6)
- Rank the default probabilities into deciles (10 groups)
- Compute the number (and percentage) of defaults in each of the 10 groups for testing data (loans issued in 2014)
- Plot the ROC curve and calculate AUC and KS statistics
- Compare model performance of the Post LASSO Logistic regression model with the Logistic regression model built in step 6. Does it perform better?

9. Based on the practice from previous steps, do a *out-of-sample* prediction using K-Nearest Neighbor algorithm. Follow these steps

- For each observation in the out-of-sample data, identify the top K closest observations from the in-sample data by calculating the Euclidean distance based on the good covariates from step 8 and classify each observation in the out-of-sample data by taking a majority vote (use KNN() fuction)
- Calibrate the hyperparameter K in K-Nearest Neighbor algorithm to find the optimal value which minimizes the misclassification rate on the out-of-sample data
- Compare misclassification rate of K-Nearest Neighbor algorithm with the Post LASSO Logistic regression model and the Logistic regression model built in step 6 (use the cutoff that minimizes the misclassification rate for logistic regression models). Which one has the lowest misclassification rate?

Data Extraction

- In order to reduce the time demands for the assignment, I have uploaded the required dataset to QCF STATS server
- I have also greatly simplified the assignment by creating the response variable (Default) based on loan status and excluding some irrelevant variables and loan status variables which may cause look-ahead bias
- Get your current working directory using getwd() command
- Set the working directory to `Q:\Data-ReadOnly\LendingClub` using the setwd() command
- The name of the csv file is `lc_loans_2007_2014.csv`

R functions that may be useful for the Data Analysis

Some of the following functions may be useful:

- glm()
- summary()
- predict()
- prediction()
- performance()
- plot()
- gains()
- read_csv()
- as_tibble()
- select()
- filter()
- arrange()
- mutate()
- group_by()
- summarise()
- ggplot()

- cor()
- stri_sub()
- ifelse()

Data Analysis

Steps in the assignment

1. First, make a list of variables you expect to matter for default prediction
2. Pre-process the loan origination data to create variables that you require. There are many important variables in the string format. Think about how to transform them into meaningful numeric variables. Some of them can be simply transformed into dummy variables, like home ownership, while some of them can not, like employment length
3. Compute the required variables
4. Compute the required statistics
5. Use function glm() to model default occurrence
6. Save the results in PDF format. I don't want to see hundreds of pages of output
7. Check the R Console to see if there are any errors in your code
8. Upload the RMD file and results (in PDF form) to Canvas