Course Outline
Data-driven Credit Modeling & FinTech Analytics
15.S12
(Draft, subject to change)

Instructor: Roger M. Stein, Ph. D.
Adjunct Professor, Stern School of Business, NYU
Research Affiliate, MIT Laboratory for Financial Engineering
E-mail: rstein@stern.nyu.edu
Lectures: Weekly sessions of 2 - 2.5 hours each; see syllabus
Office Hours: By appointment

Course Description: This course focuses on the practical challenges that arise in implementing a variety of credit models (e.g., bankruptcy and default models retail and commercial entities). With a focus on large data sets, we explore a number of data-driven approaches to modeling the likelihood that credit-risky borrowers will default on their obligations. This course tends heavily towards discussions of practical model implementations and the "frictions" that make these implementations difficult in real-world settings. We discuss a number of modeling frameworks for estimating default probabilities (PDs) and loss given default LGD. We pay special attention to validating discrete-choice models in real-world settings. We do not focus as heavily on the structure of credit markets or the details of pricing a broad variety of credit-risky instruments.

We will take the view that an effective, practical credit modeling framework will be rough around the edges with the odd inconsistency (usually to deal with available data or the lack thereof). This implies that seemingly incompatible models can each have value in specific contexts, resulting in retention of several models despite their theoretical inconsistency. Because the focus is applied, we will discuss model validation and calibration in detail and highlight data issues in estimation and validation. Since credit models for corporate debt are most well developed, we deal extensively with these models, though we will discuss certain retail asset classes as well. Lectures will to focus on conceptual themes and practical issues, with much of the technical detail underlying these to be found in the instructor's text.

We will use a number of large data sets over the course of the semester. Students will be required to implement a number of modules in the R language. Though this is not a programming course, some level of comfort with high-level programming languages will be beneficial. The instructor will devote a lecture to programming in R and a detailed, end-to-end sample modeling session for the project will be distributed, but students without programming experience in a structured programming language (e.g., writing short subroutines) may find the course challenging.

Students are required to do one main project (see below) which is drawn from industry and which will provide real-world exposure to realistic problems in credit risk. There will also be mini-projects, usually dealing with bite-sized components of the main project. (Mini-projects can usually be coded using less than a page of R code.) Subject to scheduling constraints, we will also be joined by industry experts who will present on their areas of expertise.

Course Objective: To expose students to the practical challenges associated with *building and testing single-borrower credit risk models*, such as those used by banks, as well as to the types of modeling techniques that can be used to build them; and to give students *credible and realistic experience in building models in real-world settings with large data sets*. When students complete this course, they should achieve solid foundation in some of the challenges (and potential solutions) in developing data-driven default models. They will also leave with a *toolkit of a number of useful modules* that can be applied in practical commercial settings.

Major project: There is one major assignments required for successful completion of this course.

- The *Default Modeling Project*. The objective is to develop, individually or in small teams, a default prediction model using a realistic data set. This is not a programming project, though students will find it useful to use the programming language R to estimate the model. Much of the basic command-line code that is required for a basic model is given in the attached project description, along with examples. Successful completion of the project will include presenting and documenting the model in a realistic setting.

Grading:

Modeling project:		40%
Formulation:	*10%*	
Out of sample performance:	*10%*	
Documentation	*20%*	
Mini-projects		10%
Mid-term		20%
Final exam:		30%
Extra credit for best model	$10\%/n,\ n \le 3 \equiv$	the number of team members

Homework: Aside from completing the readings, the only formal homework for this course is the work associated with completing the main modeling project and mini-projects. *It is strongly advised that you get started on the main modeling project early as it has a number of components. However, the mini-projects are designed to be used on the final project.* We will go over the mini-projects in class.

Attendance: No points are given for attendance. Some of the material on the mid-term and final exams will be drawn from the lectures and discussion. *However, students missing more than three class sessions without permission of the professor will lose 10 points on their final course grade.*

Primary Text: Bohn, J. R. and R. M. Stein, (2009) *Active Credit Portfolio Management in Practice*, NY, Wiley. (ACPMIP). It is important that you read the assigned material as portions will be covered on the final exam.

Additional reading: 1-2 papers per course session may be assigned (see syllabus). *Complete copies of lecture slides will not be distributed, although key slides will be distributed to reduce note-taking load.*

Draft Syllabus
Data-driven Credit Models

Week 1 **9/25**	**Introduction to credit risk modeling concepts and how the challenge of data analytics** • Data problems and resolutions • Key components of credit risk – PD, LGD, (EAD), correlation, size • Differing modeling paradigms • Diversification • How can we add value in developing data-driven analytics **ACPMIP**: Chapter 1, pp. 2-16; 19-23; 32-34; 38; 42-43. Chapter 2, pp. 60-62; 72-74. **Supplemental readings**: • Dhar, V. and R. Stein (1997), *Seven Methods for Transforming Corporate Data into Business Intelligence*, Prentice Hall, NJ. Chapter 3.

Week 2
10/2
	R tutorial and data sets
	• The R language
	• R studio
	• Basic operations
	• Objects
	• Vectorization
	• Packages
	• Data sets
	• Sample modeling session
	
	Mini-project: *Function to calculate a simple moving average.* Write an R function to calculate the simple moving average of a time series. The function definition should be
	• *Definition:* `ma<-function(x,k)` where
	  • `x` is a vector for which to calculate moving averages and
	  • `k` is the number of periods over which to calculate the moving average (including the current period).
	• *Return*: The function should return a vector of length `length(x)` with the current period's `k`-period moving average in place of each original data point. The beginning of the moving average vector will be padded with `NA`s .
	• *Tasks*:
	1. Implement this in two different ways:
	  a. Using a loop
	  b. Using vectorization.
	2. Use the `sys.time()` function to time the two approaches.

10/9	**NO CLASS- ENJOY THE BREAK!**

Week 3 10/16	**PD model validation – Part I**
	• Validating model power using ROC curves
	• Validating model calibration using probability-based measures
	
	ACPMIP: Chapter 7, pp. 361-397.
	
	Supplemental readings:
	• Stein, R. M., A. E. Kocagil, J. Bohn and J. Akhavain (2003). "Systematic and Idiosyncratic Risk in Middle-Market Default Prediction: A Study of the Performance of the RiskCalc and PFM Models." Moody's KMV.
	
	Mini-project: *Function to calculate the AUC ROC for two different subsets of a single data set.*
	• *Definition:* `subROC<-function(x, split.val, split.on, score, outcome,...)` where
	  • `x` is a dataframe
	  • `split.val` is a scalar, factor value, date or string used to divide the data frame
	  • `split.on` is a vector of length `nrow(x)` of the same type as `split.val; split.on <= split.val` goes to one data subset while the remainder goes to the other
	  • `score` is a numerical vector of length `nrow(x)` for calculating the ROC AUC
	  • `outcome` is a binary numerical vector of length `nrow(x)` for calculating the ROC AUC
	  • `...` additional parameters
	• *Return*: The function should return a list with three elements:
	  • A vector of length `2` with the ROC AUC for each data subset.
	  • A vector of length `nrow(x.subset1)` giving the indices of `x` that for the records included in `x.subset1`
	  • A vector of length `nrow(x.subset2)` giving the indices of `x` that for the records included in `x.subset2`
	• *Tasks*:
	  • Implement `subROC`
	  • Describe how you would make `subROC` more general so that it could take in an arbitrary one or two variable statistic (function) as an input and return the appropriate data

Week 4 10/23	**Regression-based models of default and data preprocessing** • Discrete choice models • Survival models • Case Study - PDs for firms with public information: the RiskCalc models of private firm default **ACPMIP**: Chapter 4, pp. 183-215, 238-252.

Week 5 10/30

PD model calibration

- Calibrating to empirical data using calibration curves
- Adjusting for differing baseline default rates
- Mapping between ratings and PDs and back again

ACPMIP: Chapter 4, pp. 215-233.

Supplemental readings:

- Stein, R. M., A. E. Kocagil, J. Bohn and J. Akhavain (2003). "Systematic and Idiosyncratic Risk in Middle-Market Default Prediction: A Study of the Performance of the RiskCalc and PFM Models." Moody's KMV.

Mini-project:

- *Function to create a calibration curve mapping a variable to a default rate.*
- *Function to use the calibration curve to map a variable to a default rate (including interpolation).*

- *Definition:* `estimateCalibCurve<-function(x, outcome, k, ...)` where
  - `x` is a vector of model scores
  - `outcome` is a is a binary numerical vector of length `length(x)`
    `k` is a scalar, denoting the number of "buckets" to use in the mapping

- *Return*: The function should return a list with three elements:
  - A list containing
    - `map` a dataframe of length `k` with two columns
      - the cutoff (on the same scale as `x`)
      - the mapped PD corresponding to the cutoff
    - `baseline` a scalar containing the baseline PD for `outcome`

- *Definition:* `applyCalibCurve <-function(x, map, baseline=NULL, ...)` where
  - `x` is a vector of model scores
  - `map` is a is a dataframe returned in `map` by `buildCalibCurve`
    `baseline` is a scalar, to be used if baseline adjustment is to be applied after mapping

- *Return*: The function should return a vector of length `length(x)` containing the mapped PD for each element of `x`.
- *Tasks*:
  - Implement `estimateCalibCurve`
  - Implement `applyCalibCurve`
  - Test on `estimateCalibCurve` and `applyCalibCurve` on Week 5 data set

Week 6 11/6

Mid-term

Week 7 11/6

Tree-based models

- CART
- RandomForests

Readings:

- Dhar, V. and Stein, R. (1997), *Seven Methods for Transforming Corporate Data into Business Intelligence*, Chapter 10.
- Friedman, Hastie and Tibshirani (2013), *Elements of Statistical Learning*, Section 9.2, pp. 305-311.

Week 8 11/13

PD model validation – Part II

- Calculating confidence bounds
- Noisy data
- Walk-forward analysis
- What is a more powerful model worth?

ACPMIP: Chapter 7 pp. 396-437

Mini-project: *Walk-forward engine* Write an R function to implement the walk-forward approach to estimating a linear regression model.

- *Definition:* The function definition should `be wf<-function(x,k,time.idx,f)` where
  - `x` is a dataframe
  - `k` is the minimum value of the time to include in the analysis (start walking forward after `k`)
  - `time.idx` is the index of the column in which the timestamp for the record is located
  - `formula` is the formula for the regression (all variables in `formula` must be included in `x`)
- *Return*: The function should return a list with three elements:
  - A list containing:
    - `models` list of `lm` objects, one for each out-of-sample (walk-forward) period (you may wish to turn-off storage of original data frames in `lm` return)
    - `pred` A data frame of length `nrow(x[x$time.idx>k,])` and width 2 containing one prediction for each record in each out of sample period (all periods concatenated)
- *Tasks*:
  - Implement `wf`
  - Test `wf` on Week8 data set

Week 9 11/20

Loss Given Default: theory, data acquisition and modeling

- The default and resolution process
- Definitions and measures of LGD
- What makes estimating LGD hard (when everyone used to think it was easy)

ACPMIP: Chapter 5.

Supplemental reading:

- Van de Castle, K., D. Keisman and R. Yang "Suddenly Structure Mattered: Insights into Recoveries from Defaulted Debt."

Week 10 11/27

Mortgages

- Mortgage structures
- Mortgage dynamics
- Hazard rate models
- One model or many?

Supplemental reading:

- Stein, R. M., A. Das, Y. Ding and S. Chinchalkar. 2011. "Mortgage Portfolio Analyzer: A Quasi-Structural Model of Mortgage Portfolio Losses" Working Paper. Moody's Research Labs. New York. pp. 13-43.
- Khandani, Amir, Adlar J. Kim and Andrew W. Lo. "Consumer Credit Risk Models via Machine-Learning Algorithms." Journal of Banking & Finance Vol. 34, No. 11 (2010): 2767-2787.

Week 11 12/4	**Presentation of Student Projects**
Week 12 12/11	**Wrap-up and review**
Week 13 12/18	**Final Exam**

Credit Default Model Project

Objective: You are being asked to estimate and test a simple model of corporate default. The model should take as input financial statement data on each of the firms being evaluated, and produce as output a one-year probability of default (and any ancillary measures you wish to include) for each firm. You will be provided a development data set and you may use either R or Matlab to estimate and test the model. *It is strongly recommended that you use R.* The model will be tested on a holdout sample that the instructor maintains. You may work individually or in small groups.

Deliverables: In completion of this assignment you will be expected to turn in

1. A PowerPoint or Keynote deck (10-15 slides) describing:
   a. your data
   b. the definitions of the variables you included
   c. the relative importance of the variables in the model
   d. the functional form of the models you considered
   e. any data preprocessing you performed
   f. the details of your final model
   g. your testing results, and
   h. a technical appendix (if needed);

2. The source code you used to estimate and test your model;

3. Source code that takes as input a data file of the same format as the development sample and produces outputs in the form of probabilities of default for each firm in the holdout sample
4. A file containing PDs for a the validation data (holdout sample) you will be given

Software: You will be using the R language to estimate and test your models. Much of the statistical support for estimation and testing of R models is freely available in Open Source form. In addition, the examples in class and the extended example of the main project are implemented in R. Some useful links are given below:

R software downloads: http://cran.us.r-project.org

(You will also find a large repository of statistical routines including the `caTools` package for ROC analysis.)

RStudio R IDE: http://www.rstudio.com/ide/download/

This is an integrated development environment, currently also free, which makes loading data, installing packages and overall development generally easier than in native R. I recommend that you try this out as I have found it streamlines the model estimation process quite a bit.

Data: You will be provided a data set (the *development* data set) containing abbreviated financial statement data and financial ratios for public companies. A similar validation data set (the *holdout* data set) will be used to test your models, though you will not have access to this data. The holdout sample may contain future financial statements for the firms in the sample as well as financial statements for additional firms not in your data set. This development data set is relatively large. You may wish to sample the data set down for initial experiments before using larger portions for estimation of your final models. It is also strongly recommended that you split your sample into both an estimation and a testing sample to allow you to evaluate the robustness of your model before you submit it.

Testing: After you finish your model, you will be given a new data set, in the same format as the first one, but with no default flags. You will use your model to produce PDs for each record in the data set and to then submit this for grading.

Important dates:

Week 3: Working groups due

Week 9: Final PowerPoint deck and source code due

Week 11: Presentation of selected student models