Finance and Risk Engineering

Tandon School of Engineering

New York University

Machine Learning in Financial Engineering

FRE-GY 7773

COURSE SYLLABUS

Professor Amine Mohamed Aboussalah

NYU Tandon Finance & Risk Engineering

1 Metrotech Center, Suite 1001, Office 1005

Brooklyn, NY 11201

Email: ama10288@nyu.edu

Office hours: By appointment (happy to meet, discuss, support, zoom). Email preferred.

COURSE OVERVIEW

In this course we will give an overview of several applications of machine learning to capital market forecasting and credit modeling, beginning with regressions, "shallow" layered machine learning models (e.g. Support Vector Machines, Random Forests), and ending with "deep" layered machine learning models (e.g. Long Short Term Memory Networks). Each model is discussed in detail as to what input variables and what architecture is used (rationale), how the model's learning progress is evaluated, and how machine learning scientists and capital market traders evaluate the model's final performance. By the end of the course, students should be able to identify the main features of a machine learning model for capital market forecasting and to evaluate if it is likely to be useful and if it is structured efficiently in terms of inputs and outputs.

The course covers (but it is not limited to) the following subjects:

- Training and testing workflow: scaling, cross-validation pipelines.
- Gradient descent: mini-batch, stochastic.
- Financial metrics: profitability and risk.
- Financial feature engineering.
- Models: multivariate regression, logistic regression, support vector machines, principal component analysis, decision trees, random forests, k-means, and hierarchical clustering, Gaussian mixtures, MLPs, LSTMs, and auto-encoder neural networks.
- Applications: credit modeling, financial time-series forecasting, investment portfolio design, and spread trading, credit cycle regime identification.

Students will be required to program in Python to complete assignments and a research project. Basic knowledge in linear algebra, probability and statistics is required. The project will require reading and understanding research articles. The course helps prepare students for quant or research related positions.

Picture from the financialtribune.com

COURSE OUTLINE

Week 1:
Introduction

- Why machine learning and deep learning are relevant to financial engineering
- Modelling for inference and modelling for prediction
- Traditional statistical modelling vs machine and deep learning modelling
- Types of models (by data): cross-sectional vs time-series
- Types of machine learning models: supervised, unsupervised, semi-supervised, reinforcement learning

Week 2:

Feature engineering for time series modeling

- Autocorrelation features: ACF and PACF
- Lags, date and time features
- Visualizing feature importance
- Simple autoregressive models: AR(1), MA(1), ARMA(1,1)
- Time-series decomposition example using pmdarima, lags and differencing

Week 3:

Basic machine learning workflow

- Train-Cross-Validate-Test workflow
- Scaling, cross-validation and pipelines: validation and test metrics
- Parameter optimization: grid search, random search, Gaussian process
- Metrics: MAE, MedAE, MSE, MSLE, MAD/MEAN-Ratio, MAPE, RSquared

Regularization

- Underfitting, overfitting, good fit, unknown fit
- Use of regularization to correct overfitting: L1 and L2 regularization
- L2 regularization to correct "jumping coefficients" caused by multicollinearity
- Optimizing regularization parameters: scaling, pipelines and grid search
- Scaling: Standardization, Min-Max, Mean normalization, Unit length scaling
- Scaling in finance: linear detrending (deterministic trends), differencing (stochastic trends)

Week 4:

Features selection

- White's Reality Check
- Time series split utility
- Principal component analysis in depth
- Feature reduction by selection or extraction: RFE, PCA, LDA
- Use of pipelines for feature selection or extraction
- Examples: PCA and the yield curve, factor modeling of stock portfolios (regression, with and without PCA)

Week 5:

Support vector machines (SVMs)

- Support vector machine algorithm
- Using features or kernels to model non-linearity with SVMs
- SVMs parameter optimization, with scaling and pipelines
- SVM and feature engineering
- Example: Support vector classifier for prediction of price movement

Week 6:

Trees:

- Relation between trees and binned features
- Trees and extrapolation
- Gini score and entropy score
- Tree regularization and tree parameter optimization
- Tree visualization and feature importance
- Two ways of trading a tree: extreme leaf trading and whole leaf trading
- Example: factor modeling of stock portfolio (tree regressor)

Week 7:

Tree Ensembles

- Ensembles in general, the law of large numbers and the binomial distribution
- Bagging and random forests
- Gradient boosting
- Modeling correlated multiple outputs with trees or daisy chaining
- The beta of a stock, length of beta lookback window, idiosyncratic volatility
- Empirical asset pricing via machine learning: best predictors and models
- Example: factor modeling of stock portfolio (random forest regressor, with PCA), Piotroski factor model (random forest classifier)

Week 8:

Machine learning building blocks

- Popular frameworks
- The unreasonable effectiveness of data
- Keras building blocks
- Use of KerasClassifier and KerasRegressor wrappers for cross-validation and parameter optimization
- Neural networks and scaling issues
- Rules of thumb for neural network architectures

Machine learning for financial time series

- Multivariate processing for time-series
- Walk-forward validation with grid search

- Benchmarking a grid search
- Supervised: Logistic Regression, Random Forest, SGBoost, Keras MLP
- Example: MLP classifier for price prediction with class weights, callback
- Example: MLP regressor and classifier ensembles to predict bitcoin price

Week 9:

Autoencoders

- Autoencoder algorithm
- Autoencoder and PCA comparison
- Outlier identification
- Scaling, oversampling
- Unsupervised: PCA, autoencoder
- Example: Credit card fraud identification

Week 10:

Recurrent neural network (RNN) & long short-term memory (LSTM)

- RNN and LSTM algorithms and relation to ARMA models
- The exploding/vanishing gradient problem
- Example: LSTM applied to stock price prediction (with and without window normalization)
- Example: RNN, LSTM and ARIMA applied to massive data (web page views)

Week 11:

Clustering & Gaussian mixture models

- Clustering
- Gaussian Mixtures
- The credit cycle
- Hierarchical-Risk-Parity
- Example: Gaussian mixtures for price regime identification, credit cycle phase identification
- Example: PCA and clustering for co-integrated pairs identification, PCA for eigen-portfolios
- Example: Hierarchical clustering for portfolio construction

Week 12:

Financial indicators

- Technical and Fundamental Financial Indicators

Week 13:

Project presentations and peer review

Week 14:

Project presentations and peer review

LEARNING OUTCOMES

Global: The goal of this course is to expose the participant through lectures, readings, and hands-on homework to the following topics:

- Students understand the machine learning workflow.
- Students are familiar with different types of panel data encountered in finance: cross-sectional and sequential (time or location indexed). Brownian processes (random walks) and mean-reverting processes.
- Students understand the differences between: supervised vs. unsupervised, linear vs. non-linear, regression vs. classification, cross-sectional vs. sequential.
- Students can use multivariate regression, logistic regression, principal component analysis, support vector machines, decision trees, random forests, k-means, hierarchical clustering, Gaussian mixtures, multi-layer perceptron, recurrent neural networks, LSTMs, and auto-encoder neural networks.
- Students can understand the mathematical and algorithmic structure of the models, their assumptions and their purpose, their strengths and their weaknesses.
- Students can apply the machine learning models to credit modeling, time-series and financial time-series forecasting, investment portfolio design, spread trading, credit cycle regime identification.
- Students can utilize the financial metrics of model adequacy: profit or risk evaluation metrics associated with financial predictive models: information coefficient, Sharpe ratio, CAGR, annualized volatility, White Reality Check (a version of Superior Predictive Ability).

Instructional: After completing this course, participants will be able to use Python and out-of-the-box statistical learning libraries (e.g. Scikit-Learn, Keras/TensorFlow) to program a basic machine learning workflow applied to panel data and involving the following steps:

- Obtain panel data from Wharton Research Data Service using web-queries or obtain free data from data providers such as Yahoo, Quandl etc. using Pandas-Datareader.
- Prepare the data into indexed dataframes using Pandas functions for date indexing, for hierarchical indexing, and for table management: pivot, join and merge.
- Engineer alpha-factors and risk-factors with specialized libraries including Ta-lib, FINTA, the Fama-Macbeth linear factor model, Pandas date processing functions.
- Engineer time-series decomposition features such as trend, seasonality, lookback window etc. using Pmdarima, HoltWinter. ExponentialSmoothing, simple and partial autocorrelation functions.
- Engineer categorical features with one-hot-encoding.
- Extract features using principal component analysis and autoencoders.
- Select the best features using Scikit-learn feature selection functions and Quantopian's Alphalens module.
- Scale the features using Scikit-learn functions.
- Construct machine learning workflows using Scikitlearn pipelines and Keras out-of-the-box functions including: splitting of data with train_test_split, model evaluation with cross-validation and model parameter tuning with grid-or-randomized-search-cross-validation.
- Apply these workflows to cross-sectional and time-series panel data using various types of models.
- Evaluate a model using simple statistical criteria (e.g. mean squared error, precision-recall), more sophisticated statistical criteria (e.g. bootstrap based), and financial criteria (Information coefficient, Sharpe ratio, CAGR, annualized volatility etc.)
- Display a model's feature importance and predictive adequacy using Scikit-Learn and Keras out-of-the-box functions and matplotlib.

STUDENT LEARNING ACTIVITIES

- Weekly lectures and discussions
- Weekly readings
- Programming homework
- Group research project and presentation
- Peer-review of research projects

ASSESSMENT

- Participation 10% (lecture scribing, asking good questions in class, and participating in discussions)
- ML topic presentation 10%
    - Exhibition and demonstration by hand of a concrete ML algorithm
    - Algorithm and problem selection 2%
    - Clarity & Organization 2%
    - Accuracy & Rigor 2%
    - Analysis & Persuasion (evaluation of the algorithm) 2%
    - Written quality of notes (no handwritten notes please) 2%
- Programming homework assignments 30% (3 assignments 10% each)
- Final project 50%
    - Research proposal 10% [2 pages]
    - Progress report 10% [4-5 pages]
    - Peer review of another team's project 5%
    - Final report and presentation 25% (Functional code has to be submitted. The contributions of each member have to be reported.) [8 pages plus references]

GRADING POLICY

I do not "give" grades – I report the grades that students earn. In general, this class' requirements can be compared to the job requirements of a professional. If the quality of the work you hand to me would, if I were your supervisor in a workplace, place you on the fast-track to a promotion then you have earned an "A". If I were your supervisor and you were to give me work that would lead me to believe that you were a steady employee but not a potential "star" then that work would correspond to a grade of "B". Work that would cause me to think about whether or not you had an extended future as my employee is "C" work, and work submitted that would cause me to call the HR department is "F" work. NYU-Tandon uses a grading system for graduate courses where possible grades are: {A, A-, B+, B, B-, C+, C, F}.

SUGGESTED TEXTBOOKS

Jake VanderPlas (2017) Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media, Inc. (Publicly available online: https://jakevdp.github.io/PythonDataScienceHandbook/)

Aurélien Géron (2019) Hands-on Machine Learning with Scikit-Learn, Kearas, and TensorFlow, 2nd Edition. O'Reilly Media, Inc.

Machine Learning in Finance: From Theory to Practice – Matthew F. Dixon, Igor Halperin, Paul Bilokon, 2020.

Machine Learning for Asset Managers (Elements in Quantitative Finance) Part of: Elements in Quantitative Finance (3 Books) by Marcos M. López de Prado, 2020.

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron, 2019.