Machine Learning and Large Language Models for Business and Management

From classical prediction models to text analysis and generative AI | Upper-level Undergraduate | 13 Weeks

Note: This is a sample syllabus designed for a teaching portfolio by Yuxuan Chen. Specific datasets, readings, software platforms, and assessment weights can be adapted to program level and departmental needs.

Level	Upper-level undergraduate	**Format**	Lecture + Python lab
Prerequisites	Introductory statistics/econometrics	**Software**	Python, scikit-learn, Keras, Hugging Face / LLM APIs

Course Overview

This course provides business and management students with a coherent pathway from classical machine learning to large language models (LLMs). The first half builds reusable foundations in statistical inference, including prediction, classification, regularization, tree-based models, ensemble learning, and neural networks. The second half shows how these ideas lead naturally to text representations, attention, Transformer models, prompting, retrieval-augmented generation (RAG), and LLM evaluation. The emphasis is on mathematical intuition, reproducible Python workflows, and business-relevant applications rather than proof-heavy derivations or large-scale model training.

Learning Objectives

- Explain the core logic of supervised machine learning, including loss functions, generalization, cross-validation, and model complexity.
- Apply and compare regression, classification, regularization, tree-based, ensemble, and neural-network models using Python.
- Understand how text is represented numerically and how attention and Transformer architectures support modern LLMs.
- Use prompting and RAG to extract, classify, summarize, and retrieve information from business documents.
- Evaluate machine-learning and LLM outputs using appropriate performance metrics, robustness checks, and reproducible workflows.

Teaching Approach

Each week combines a compact conceptual lecture with a guided coding exercise/tutorial. Mathematics is introduced when it clarifies how a model works—for example, the loss function behind OLS, the probability mapping in Logit, the penalty term in Lasso, or the similarity calculation behind attention. Students work with structured firm data and unstructured sources such as annual reports, news, CSR/ESG disclosures, patents, and consumer text.

Indicative Assessment

Weekly labs	20%	**Applied assignments**	25%
Midterm	20%	**Final project**	35%

Course Architecture

Weeks 1–3	Weeks 4–7	Weeks 8–10	Weeks 11–13
Foundations	**Predictive ML**	**From Text to LLMs**	**LLM Workflows**
OLS, Logit, generalization, regularization	Classifiers, trees, Random Forest, XGBoost	Neural networks, embeddings, attention, Transformer	Prompting, RAG, evaluation, research applications

Course Schedule

Weeks 1–7: Classical machine learning as the foundation for modern AI

Week	Topic	Core Concepts & Applied Lab
1	**Machine Learning and Python: A First Workflow**	What machine learning is; supervised vs. unsupervised learning; features and labels; training/test data; NumPy, Pandas, and the scikit-learn workflow. *Lab: load, clean, and explore a business dataset; train a first predictive model.*
2	**From OLS to Machine Learning**	OLS as prediction; mean squared error; loss functions; gradient descent intuition; overfitting; bias–variance trade-off; cross-validation; polynomial regression. *Lab: visualize overfitting and compare in-sample and out-of-sample performance.*
3	**Logit and Classification**	Binary and multinomial classification; logistic function; probabilities and odds; decision thresholds; confusion matrix; sensitivity/specificity; ROC and AUC. *Lab: predict customer churn, firm misconduct, or another binary business outcome.*
4	**High-Dimensional Data: Ridge, Lasso, and Elastic Net**	Why high-dimensional models overfit; L1 and L2 penalties; shrinkage; variable selection; tuning regularization with cross-validation. *Lab: select predictive firm characteristics or high-dimensional text features.*
5	**Classical Classifiers: Naive Bayes, KNN, and SVM**	Three views of classification: probability, distance, and margins; Laplace correction; feature scaling; maximum-margin intuition; kernel idea. *Lab: spam or short-text classification; compare classifiers on the same data.*
6	**Decision Trees and Random Forests**	Classification and regression trees; Gini index and entropy; pruning; bagging; bootstrap sampling; random forests; variable importance; partial dependence. *Lab: predict firm performance, credit risk, or customer outcomes using tree-based models.*
7	**Boosting and XGBoost**	Ensemble learning; AdaBoost; gradient boosting as sequential error correction; learning rate and tree depth; XGBoost intuition and model comparison. *Lab: compare Logit, Random Forest, and XGBoost on a structured business dataset.*

Objective of this part:

They give students a compact but usable machine-learning toolkit while introducing the ideas that reappear in LLMs: loss minimization, probabilistic classification, regularization, nonlinear representation, and iterative optimization. Students who wish to pursue classical machine learning further will also have a complete foundation for independent study and research applications.

Weeks 8–13: From neural networks and text to Transformers, LLMs, and research applications

Week	Topic	Core Concepts & Applied Lab
8	**Neural Networks and Text Representation**	Perceptrons and feedforward neural networks; activation functions; forward propagation and back-propagation intuition; stochastic gradient descent; Bag-of-Words, TF-IDF, Word2Vec, and cosine similarity. *Lab: represent annual-report or news text numerically and build a simple text classifier.*
9	**Sequences and Attention**	Why word order matters; RNN/LSTM intuition; limitations on long sequences; encoder–decoder idea; attention; Query, Key, Value; similarity scores and attention weights. *Lab: use a small example to visualize which words or sentences receive greater attention.*

Week	Topic	Core Concepts & Applied Lab
10	**Transformer, BERT, and GPT**	Self-attention; multi-head attention intuition; positional information; encoder vs. decoder; tokenization; BERT for representation/classification; GPT for generation; pretraining and transfer learning. *Lab: classify CSR/ESG or business-news text with a pretrained model and generate concise document summaries.*
11	**Prompting and LLM-Based Text Analysis**	How LLMs generate responses; zero-shot and few-shot prompting; task instructions; structured outputs; temperature intuition; classification, extraction, summarization, and batch-processing workflows. *Lab: turn patents, announcements, executive biographies, or annual-report passages into structured variables.*
12	**RAG and LLM Evaluation**	Why LLMs hallucinate; embeddings and vector similarity; document chunking; retrieval-augmented generation; grounding; human-coded benchmarks; accuracy, precision, recall, F1; prompt robustness and reproducibility. *Lab: build a small document-Q&A workflow and evaluate extracted answers against human coding.*
13	**Machine Learning and LLMs in Business Research**	Choosing between classical ML and LLMs; prediction vs. measurement vs. explanation; text-based constructs; examples from management and business research; limitations, ethics, and reproducible workflows. *Capstone: team project presentation using either a classical ML model or an LLM-based text workflow.*

Final Project

Teams select a substantive business or management question and complete one end-to-end analysis. Projects may predict a business outcome with Lasso, Random Forest, or XGBoost; construct a text measure using TF-IDF, BERT, or an LLM; or develop a small RAG workflow for annual reports, disclosures, or other business documents. Deliverables include a concise research question, documented data, model choice, evaluation metrics, reproducible Python code, interpretation of results, and a short presentation.

Recommended Resources

Chollet, F., & Watson, M. (2025). *Deep learning with Python* (3rd ed.). Manning Publications.

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An introduction to statistical learning: With applications in Python*. Springer. https://doi.org/10.1007/978-3-031-38747-0

Jurafsky, D., & Martin, J. H. (2026). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models* (3rd ed.) [Online manuscript]. https://web.stanford.edu/~jurafsky/slp3/