Machine Learning in Finance

Course Instructor: Dr Ferdinantos Kottas

Email: ferdinantosk@econ.uoa.gr

Website: https://fkottas.github.io/FerdinantosKottas.github.io/

Course Structure

Duration: 13 weeks

Format: 2-hour lectures + 1-hour Lab

Language: English

Assessment: 20% Weekly exercises, 40% Capstone/Project, 40% Exam, 10% Bonus Exercise

Course requirements

- **Audience:** statistics, probability theory and mathematics (linear algebra and calculus); Financial markets knowledge; no prior knowledge of Python is required, but general basic coding knowledge is needed, especially knowledge of R is helpful.
- **Primary dataset(s):** Kaggle credit-risk style tables (e.g., Home Credit, Give Me Some Credit, Default of Credit Card Clients) and yahoo finance for portfolio management.
- **Software and Libraries:** Python 3.11+, Jupyter, pandas, numpy, matplotlib, scikit-learn, imbalanced-learn, shap, lightgbm/xgboost, category_encoders.
- **Main Interface:** Colab, a free platform that requires only a Gmail account and functions similarly to Jupyter Notebook.

Course Overview

The primary objective of this course is to equip students with the theoretical knowledge and practical skills required to apply modern machine learning techniques to financial decision-making and data-driven problem solving. Students will learn how to develop, evaluate, and interpret predictive models using Python while understanding the mathematical principles that underpin contemporary machine learning algorithms.

This course provides a comprehensive introduction to machine learning with a strong emphasis on financial applications. It combines statistical learning concepts, data preprocessing, predictive modelling, model validation, and explainable artificial intelligence within a practical programming environment. Throughout the semester, students will work with real-world financial datasets to develop reproducible and industry-relevant machine learning solutions.

The course explores a broad range of financial applications, including:

- **Credit Risk Analytics:** Default prediction, credit scoring, probability of default estimation, and risk assessment.
- **Investment and Financial Analytics:** Return prediction, financial forecasting, portfolio analytics, and data-driven investment decision support.

Throughout the course, students will gain experience in building complete machine learning workflows, from data acquisition and preprocessing to model evaluation, interpretation, and communication of results.

Key Features

- Balanced integration of theory and practical implementation
- Hands-on programming using Python and Google Colab
- Real-world financial datasets and industry case studies
- Mathematical intuition behind machine learning algorithms
- Explainable Artificial Intelligence (XAI) and model interpretability
- Best practices for reproducible research and machine learning workflows
- Responsible use of Artificial Intelligence tools to support learning and programming

Lecture Structure

Each weekly session consists of a **3-hour lecture** designed to integrate theoretical concepts with practical financial applications.

The lecture is divided into two complementary components:

- **Part I (approximately 2 hours):** Fundamental concepts, mathematical foundations, machine learning algorithms, statistical principles, and methodological discussions.
- **Part II (approximately 1 hours):** Practical financial applications, Python implementation, case studies, model development, interpretation of results, and discussion of best practices.

This integrated teaching approach enables students to connect theoretical knowledge with practical implementation while developing analytical and programming skills applicable to real-world financial problems.

Each lecture in part II is accompanied by a **1-hour practical laboratory session** where students apply the concepts introduced during the lecture through guided programming exercises.

Laboratory activities include:

- Python programming using Google Colab
- Data preprocessing and exploratory data analysis
- Development of machine learning models
- Model evaluation and comparison
- Feature engineering
- Explainable AI techniques
- Writing clean, reproducible, and well-documented code
- Good programming practices and workflow organization

Every week, students will complete a practical exercise designed to reinforce the lecture material and progressively build the skills required for the capstone project.

AI-Assisted Learning

Students are encouraged to responsibly use modern Artificial Intelligence tools (e.g., ChatGPT, Gemini, Claude, GitHub Copilot) to assist with programming, debugging, documentation, and understanding machine learning concepts. These tools are intended to enhance learning rather than replace it. Students remain fully responsible for understanding, explaining, and justifying all submitted code, modelling decisions, and analytical results.

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Explain the fundamental principles of machine learning and their applications in finance.
2. Prepare, clean, and preprocess financial datasets for predictive modelling.
3. Develop and evaluate supervised machine learning models using Python.
4. Apply classification techniques to solve credit risk and other financial prediction problems.
5. Compare machine learning algorithms using appropriate validation and performance metrics.
6. Perform feature engineering and model optimisation to improve predictive performance.
7. Interpret machine learning models using Explainable AI techniques and effectively communicate analytical findings.
8. Develop reproducible and well-structured machine learning workflows suitable for real-world financial applications.
9. Critically assess the strengths, limitations, and ethical considerations associated with machine learning in finance.
10. Design and implement complete data-driven solutions to support financial decision-making.

Module Structure

Week 1: Python Foundations I

Aim: Develop the fundamental programming skills required for data analysis and machine learning using Python and Google Colab.

Topics: Introduction to Google Colab and Jupyter Notebooks; Python syntax and programming fundamentals; variables, data types, control structures, functions, file handling, virtual environments, and package management. Loading financial datasets, computing descriptive statistics, and producing introductory data visualisations.

Application: Introduction to financial datasets and exploratory analysis using Python.

Weekly Exercise 1: Develop a Python notebook implementing basic programming tasks, grouped descriptive statistics, and visualisations for a financial dataset.

Recommended Readings:

- Matthes (2023), *Python Crash Course* – Selected Chapters.
- McKinney (2022), *Python for Data Analysis* – Chapters 1–3.

Laboratory Session 1: Google Colab setup, Python programming fundamentals, importing datasets, descriptive statistics, and introductory visualisation.

Week 2: Python Foundations II – Data Manipulation with NumPy and Pandas

Aim: Develop proficiency in manipulating and exploring structured financial data.

Topics: NumPy arrays; Pandas DataFrames; indexing, filtering, joins, group operations, data transformation, missing values, correlations, and exploratory data analysis.

Application: Exploratory analysis of a real-world credit risk dataset.

Weekly Exercise 2: Perform a complete exploratory data analysis and identify key data quality issues and statistical characteristics.

Recommended Readings:

- McKinney (2022), *Python for Data Analysis* – Chapters 4–8.
- James et al. (2023), *An Introduction to Statistical Learning with Applications in Python* – Chapter 2.

Laboratory Session 2: Exploratory Data Analysis (EDA) using Pandas and visualization libraries.

Week 3: Data Cleaning and Preprocessing

Aim: Construct reliable and reproducible datasets suitable for predictive modelling.

Topics: Missing value treatment, duplicate detection, outlier identification, feature scaling, categorical encoding, preprocessing pipelines, and ColumnTransformer.

Application: Data preparation for credit risk modelling.

Weekly Exercise 3: Build a complete preprocessing pipeline and document the most significant data quality improvements.

Recommended Readings:

- James et al. (2023) – Chapters 2 and 3.
- Géron (2023), *Hands-On Machine Learning* – Data Preprocessing.

Laboratory Session 3: Building preprocessing pipelines using Scikit-learn.

Week 4: Feature Engineering for Financial Data

Aim: Develop meaningful predictive variables while avoiding information leakage.

Topics: Feature construction, financial ratios, interaction terms, nonlinear transformations, feature selection, leakage prevention, and domain-driven feature engineering. Special topic is the binning and woe.

Application: Credit risk and fraud detection.

Weekly Exercise 4: Design and evaluate 6–10 new predictive variables.

Recommended Readings:

- Géron (2023) – Feature Engineering.
- Kuhn & Johnson (2019), *Feature Engineering and Selection*.

Laboratory Session 4: Feature engineering techniques for financial datasets.

Week 5: Logistic Regression for Credit Risk Modelling

Aim: Develop an interpretable baseline classification model.

Topics: Logistic regression, class imbalance, class weighting, L1/L2/Elastic-Net regularisation, coefficient interpretation, probability calibration, and model diagnostics.

Mathematical Foundations: Odds and log-odds, maximum likelihood estimation, gradient optimisation, and regularisation penalties.

Application: Probability of Default (PD) modelling.

Weekly Exercise 5: Develop and compare regularised logistic regression models using cross-validation.

Recommended Readings:

- James et al. (2023) – Chapter 4.
- Hosmer, Lemeshow & Sturdivant (2013), *Applied Logistic Regression*.

Laboratory Session 5: Logistic Regression implementation for credit scoring.

Week 6: Model Evaluation and Performance Metrics

Aim: Evaluate predictive models using appropriate statistical and business performance measures.

Topics: Confusion matrix, Precision, Recall, F1-score, ROC-AUC, PR-AUC, KS Statistic, Gini coefficient, Log-Loss, Brier Score, Lift Charts, Gain Charts, and probability calibration.

Mathematical Foundations: ROC and Precision–Recall analysis, Negative Log-Likelihood, and the relationship between KS and Gini.

Application: Performance evaluation of credit risk models.

Weekly Exercise 6: Develop a reusable model evaluation toolkit for future assignments.

Recommended Readings:

- James et al. (2023) – Chapter 4.
- Hand, D. (2009), *Measuring Classifier Performance*.

Laboratory Session 6: Comprehensive evaluation of classification models.

Week 7: Decision Trees and Random Forests

Aim: Develop nonlinear predictive models and understand ensemble learning.

Topics: Decision Trees (CART), Gini impurity, entropy, pruning strategies, Random Forests, feature importance, partial dependence, and hyperparameter tuning.

Application: Credit risk prediction using ensemble methods.

Weekly Exercise 7: Optimise Random Forest models and compare predictive performance against Logistic Regression.

Recommended Readings:

- Breiman (2001), *Random Forests*.
- James et al. (2023) – Chapters 8 and 9.

Laboratory Session 7: Decision Trees and Random Forest implementation.

Week 8: Model Validation and Generalisation

Aim: Develop reliable models capable of generalising to unseen data.

Topics: Overfitting, underfitting, training-validation-test methodology, Stratified K-Fold Cross-Validation, nested cross-validation, learning curves, validation curves, and early stopping.

Application: Robust validation strategies for financial machine learning.

Weekly Exercise 8: Evaluate learning behaviour and diagnose model generalisation.

Recommended Readings:

- James et al. (2023) – Chapter 5.
- Géron (2023) – Model Validation.

Laboratory Session 8: Cross-validation and model diagnostics.

Week 9: Gradient Boosting Methods

Aim: Develop high-performance machine learning models for structured financial data.

Topics: Gradient Boosting, XGBoost, LightGBM, CatBoost, regularisation, hyperparameter optimisation, early stopping, feature importance, and model comparison.

Mathematical Foundations: Functional gradient descent and additive modelling.

Application: Advanced credit scoring and fraud detection.

Weekly Exercise 9: Optimise Gradient Boosting models using hyperparameter search.

Recommended Readings:

- Chen & Guestrin (2016), *XGBoost*.
- James et al. (2023) – Chapter 8.

Laboratory Session 9: Building high-performance Gradient Boosting models.

Week 10: Alternative Machine Learning Algorithms

Aim: Compare complementary machine learning techniques for financial classification.

Topics: Support Vector Machines, k-Nearest Neighbours, Naïve Bayes, kernel methods, distance metrics, probabilistic classifiers, stacking method and optional anomaly detection using Isolation Forest.

Application: Benchmarking multiple classification techniques.

Weekly Exercise 10: Evaluate an alternative classifier and justify its suitability.

Recommended Readings:

- James et al. (2023) – Chapters 4 and 12.

Laboratory Session 10: Comparative evaluation of machine learning algorithms.

Week 11: Model Calibration and Explainable Artificial Intelligence

Aim: Improve model reliability and interpretability.

Topics: Probability calibration, Platt Scaling, Isotonic Regression, reliability diagrams, SHAP values, global and local explanations, and model documentation.

Application: Explainable credit scoring models.

Weekly Exercise 11: Produce an interpretable and calibrated Probability of Default model accompanied by a SHAP analysis and concise model documentation.

Recommended Readings:

- Lundberg & Lee (2017), *A Unified Approach to Interpreting Model Predictions*.
- Molnar (2024), *Interpretable Machine Learning*.

Laboratory Session 11: Model explainability using SHAP.

Week 12: Introduction to Neural Networks

Aim: Introduce deep learning methods for structured financial datasets.

Topics: Artificial Neural Networks, multilayer perceptrons, activation functions, backpropagation, dropout, regularisation, early stopping, and model calibration.

Application: Credit risk and fraud detection using deep learning.

Optional Exercise: Compare the predictive performance of Neural Networks with the best traditional machine learning model developed during the course.

Recommended Readings:

- Goodfellow, Bengio & Courville (2016), *Deep Learning* – Chapters 6–7.
- Géron (2023) – Neural Networks.

Laboratory Session 12: Developing neural network models using TensorFlow/Keras.

Week 13: Capstone Project

Students will integrate the knowledge acquired throughout the semester by developing a complete machine learning solution using a real-world financial dataset.

The project should include:

- Data acquisition and preprocessing
- Exploratory Data Analysis
- Feature engineering
- Development of baseline and advanced machine learning models
- Model validation and hyperparameter optimisation
- Calibration and performance evaluation

- Explainable AI using SHAP
- Critical discussion of limitations and practical implications

Deliverables:

- Complete Python notebook
- Technical report
- 10-minute presentation
- Supporting code and documentation

Core Textbooks

The following textbooks provide the theoretical and practical foundations of the course and are strongly recommended throughout the semester.

1. **James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023).** *An Introduction to Statistical Learning with Applications in Python (2nd ed.).* Springer.
    - Primary textbook covering statistical learning, supervised learning, model evaluation, and practical Python implementations.
    - Available free online: https://www.statlearning.com/
2. **Hastie, T., Tibshirani, R., & Friedman, J. (2009).** *The Elements of Statistical Learning (2nd ed.).* Springer.
    - Advanced reference covering the mathematical foundations of statistical learning and machine learning algorithms.
    - Available free online: https://hastie.su.domains/ElemStatLearn/
3. **Géron, A. (2023).** *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (3rd ed.).*
    - Practical implementation of machine learning algorithms using modern Python libraries.
4. **Goodfellow, I., Bengio, Y., & Courville, A. (2016).** *Deep Learning.*
    - Comprehensive reference for neural networks and deep learning methods.
    - Available free online: https://www.deeplearningbook.org/

Additional Recommended References

Students interested in advanced financial machine learning are encouraged to consult the following references.

- López de Prado, M. (2018). *Advances in Financial Machine Learning.* Wiley & Sons
- Molnar, C. (2022). *Interpretable Machine Learning.* A Guide for Making Black Box Models Explainable. Leanpub

- McKinney, W. (2022). *Python for Data Analysis.* Data Wrangling with pandas, NumPy & Jupyter, O Reilly
- Kuhn, M., & Johnson, K. (2026). *Feature Engineering and Selection.* A practical Approach for Predictive Models, Taylor & Francis Group

---

Software and Learning Resources

Required Software

Students will complete all programming activities using Python and modern open-source data science libraries.

Required software includes:

- Python 3.11 or later
- Google Colab
- Jupyter Notebook
- NumPy
- Pandas
- Matplotlib
- Scikit-learn
- XGBoost
- LightGBM
- SHAP
- Imbalanced-learn

- **Additional Software**

Students may also use

- Visual Studio Code
- Spyder
- PyCharm Community Edition
- Git and GitHub
- Anaconda or Miniconda
- Kaggle Notebooks

- **Datasets**

The course uses publicly available financial datasets throughout the semester. Typical datasets include

- LendingClub Consumer Loans
- Home Credit Default Risk
- Give Me Some Credit
- Default of Credit Card Clients
- IEEE-CIS Fraud Detection Dataset (selected examples)

Additional datasets may be introduced during laboratory sessions and the capstone project.

Course Policies

- **Academic Integrity**

Academic honesty is an essential component of professional and scientific practice.

Students are expected to:

- submit original individual work unless collaboration is explicitly permitted;
- acknowledge all external sources, datasets, software libraries, and research publications appropriately;
- avoid plagiarism, fabrication of results, or unauthorised collaboration;
- ensure that all submitted code and written work accurately reflect their own understanding.

Any breach of the University's Academic Integrity Policy may result in disciplinary action in accordance with institutional regulations.

- **Responsible Use of Artificial Intelligence**

Artificial Intelligence tools are becoming an integral part of modern data science and software development. Students are encouraged to use AI systems responsibly as learning assistants.

Acceptable uses include:

- debugging Python code;
- understanding programming errors;
- exploring alternative implementations;

- improving code documentation;
- clarifying theoretical concepts.

Students remain fully responsible for every line of submitted code, all modelling decisions, and the interpretation of results. During assessments, students may be required to explain or reproduce their work independently.

- ## Collaboration Policy

Collaborative learning is encouraged throughout the course, particularly during laboratory sessions and class discussions.

The following guidelines apply:

- laboratory exercises may be discussed with classmates, but submitted work must remain individual unless otherwise specified;
- collaboration within approved project teams is encouraged for the capstone project;
- all examinations and individual assessments must be completed independently.

- ## Assessment Submission Policy

Students are responsible for submitting all coursework before the published deadlines.

Late submissions will normally incur a penalty in accordance with the University's assessment regulations unless an approved extension has been granted.

Requests for deadline extensions should be supported by appropriate documentation and submitted as early as possible.

- ## Attendance and Participation

Regular attendance and active participation are strongly recommended.

Students are expected to

- attend lectures and laboratory sessions;
- complete weekly programming exercises;
- contribute to classroom discussions;
- seek clarification whenever concepts are unclear.

Consistent engagement throughout the semester will significantly improve learning outcomes and preparation for the capstone project.

- **Programming Standards**

Students are expected to write clean, well-structured, and reproducible Python code.

Submitted notebooks should include

- meaningful variable names;
- appropriate comments and documentation;
- modular functions where appropriate;
- reproducible results using fixed random seeds;
- clear presentation of outputs and visualisations.

Why This Course Is Unique

This course has been designed to combine rigorous academic foundations with practical machine learning skills that reflect current industry practice in finance and data analytics.

Its distinguishing characteristics include:

1. **Practical Python Development**
   - Every theoretical concept is accompanied by hands-on implementation using real financial datasets and modern Python libraries.
2. **Complete Machine Learning Workflow**
   - Students experience the entire modelling process, from data collection and preprocessing to feature engineering, model optimisation, evaluation, interpretation, and reporting.
3. **Financial Industry Applications**
   - Machine learning methods are presented through realistic problems in credit risk, fraud detection, financial forecasting, and decision support, highlighting the unique challenges of financial data.
4. **Explainable and Responsible AI**
   - Particular emphasis is placed on model transparency, interpretability, calibration, fairness, and responsible use of artificial intelligence in financial decision-making.
5. **Industry-Relevant Programming Practices**
   - Students learn reproducible workflows, version control principles, documentation standards, and coding practices commonly adopted in professional data science teams.

6. **Capstone Project**
    - The course culminates in a comprehensive end-to-end project requiring students to integrate theoretical knowledge and practical programming skills to solve a realistic financial machine learning problem.
7. **Research-Informed Teaching**
    - Lectures incorporate recent developments from the machine learning and finance literature, exposing students to both established methodologies and emerging research directions.