Machine Learning in Finance: Theory & Applications

A Mathematical Foundation for Financial ML

Advanced Course in Quantitative Finance

4-Hour Comprehensive Workshop

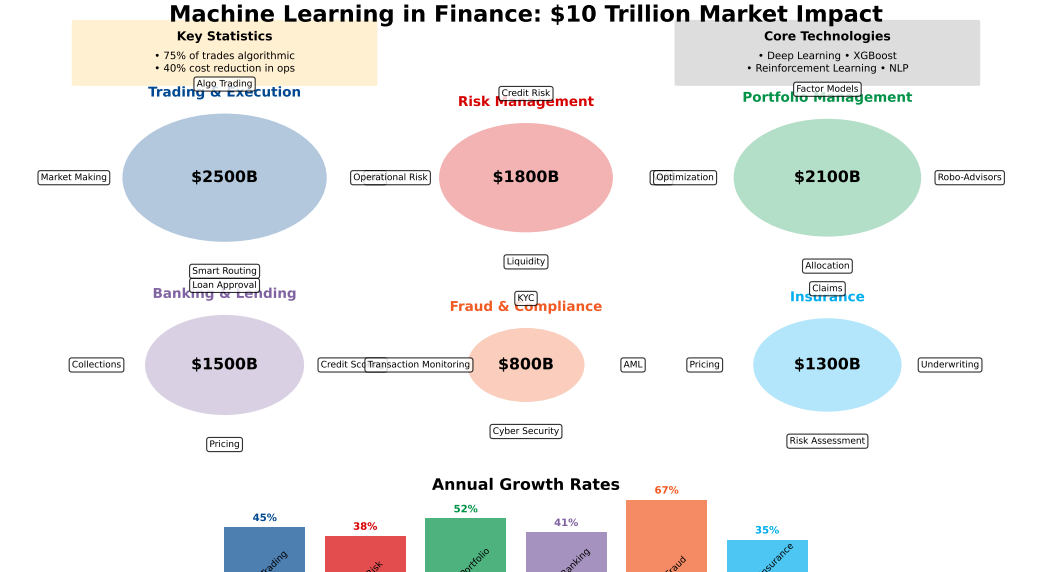
Program Overview: 4-Hour Journey Through ML Finance

1. Machine Learning Foundations
2. Statistical Learning Theory
3. Supervised Learning Methods
4. Unsupervised Learning Methods
5. Risk & Portfolio Management
6. Algorithmic Trading & Pricing
7. Credit Risk & Fraud Detection
8. Ethics, Regulation & Future

From Theory to Trading Floor

Part 1: Machine Learning Foundations

Theory, Mathematics, and Finance Applications


Machine Learning in Finance: $10 Trillion Market Impact
Key Statistics
• 75% of trades algorithmic
• 40% cost reduction in ops
Core Technologies
• Deep Learning • XGBoost
• Reinforcement Learning • NLP
Trading & Execution
$2500B
Algo Trading
Market Making
Smart Routing
Risk Management
$1800B
Credit Risk
Operational Risk
Liquidity
Portfolio Management
$2100B
Factor Models
Optimization
Robo-Advisors
Allocation
Banking & Lending
$1500B
Loan Approval
Collections
Pricing
Fraud & Compliance
$800B
KYC
Transaction Monitoring
AML
Cyber Security
Insurance
$1300B
Claims
Pricing
Underwriting
Risk Assessment
Annual Growth Rates
45%
Trading
38%
Risk
52%
Portfolio
41%
Banking
67%
Fraud
35%
Insurance

Tom Mitchell's Definition (1997)

A computer program learns from:

- **Experience** E with respect to
- **Task** T and
- **Performance measure** P

if its performance at task T, as measured by P, improves with experience E.

Finance Example:
E: Historical stock prices
T: Predict tomorrow's return
P: Sharpe ratio of predictions

Mathematical View

Learn function $f : \mathcal{X} \to \mathcal{Y}$
Given training set:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$$

Find:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda R(f)$$

where:

- L: Loss function
- R: Regularization term
- λ: Regularization strength

Machine Learning vs Traditional Finance Models

Traditional Finance

Black-Scholes-Merton (1973)

$$C = S_0\Phi(d_1) - Ke^{-rT}\Phi(d_2)$$

- Strong assumptions
- Closed-form solutions
- Model-driven
- Limited parameters
- Interpretable

Limitations:

- Assumes log-normal returns
- Constant volatility
- No transaction costs
- Perfect markets

Machine Learning

Neural Network Pricing

$$\hat{C} = NN(S_0, K, T, \sigma_{impl}, \text{Greeks}, ...)$$

- Data-driven discovery
- Non-parametric
- Flexible architecture
- High-dimensional
- Accurate but opaque

Advantages:

- Captures market microstructure
- Adapts to regime changes
- Includes all observables
- Learns from anomalies

Three Paradigms of Machine Learning

Supervised

$$\{(x_i, y_i)\}_{i=1}^{n} \to \hat{f}$$

Finance Applications:

- Credit scoring
- Stock prediction
- Fraud detection
- Option pricing

Key Algorithms:

- Random Forest
- XGBoost
- Neural Networks

Unsupervised

$$\{x_i\}_{i=1}^{n} \to \text{Structure}$$

Finance Applications:

- Portfolio clustering
- Anomaly detection
- Risk factors
- Market regimes

Key Algorithms:

- K-means
- PCA/ICA
- Autoencoders

Reinforcement

$$(s_t, a_t, r_t, s_{t+1}) \to \pi^*$$

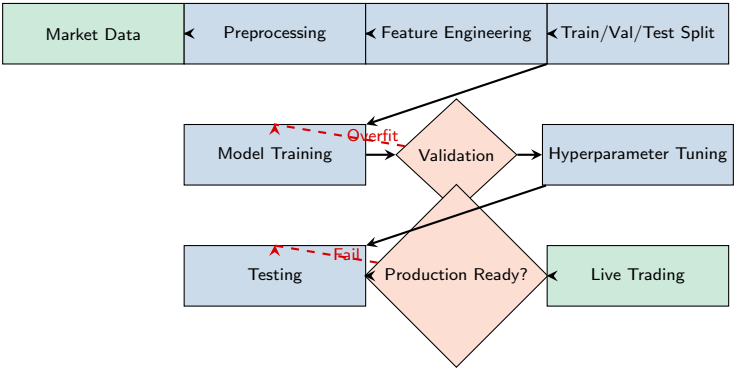
Finance Applications:

- Portfolio optimization
- Execution strategies
- Market making
- Hedging

Key Algorithms:

- Q-Learning
- PPO
- A3C

The Machine Learning Pipeline in Finance



Critical: 70/15/15 Split for Financial Time Series

Regression Losses

Mean Squared Error (MSE):

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Mean Absolute Error (MAE):

$$L_{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Huber Loss (Robust):

$$L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & \text{if } |y-\hat{y}| \le \delta \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

Quantile Loss (VaR):

$$L_\tau = \sum_i \rho_\tau(y_i - \hat{y}_i)$$

where $\rho_\tau(u) = u(\tau - \mathbb{1}_{u<0})$

Finance-Specific Losses

Sharpe Ratio Loss:

$$L_{Sharpe} = -\frac{\mathbb{E}[R_p]}{\sqrt{\text{Var}[R_p]}}$$

Maximum Drawdown:

$$L_{MDD} = \max_{t\in[0,T]}\left(1 - \frac{P_t}{\max_{s\in[0,t]} P_s}\right)$$

Directional Accuracy:

$$L_{DA} = -\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}[\text{sign}(y_i) = \text{sign}(\hat{y}_i)]$$

Profit & Loss:

$$L_{PnL} = -\sum_{i=1}^{n}\hat{y}_i \cdot r_i$$

where r_i is actual return

Fundamental ML Theorem

For squared loss, the expected error decomposes as:

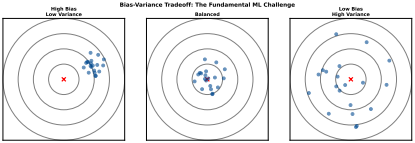
$$\mathbb{E}[(y - \hat{f}(x))^2] = \underbrace{\text{Bias}^2[\hat{f}(x)]}_{\text{Underfitting}} + \underbrace{\text{Var}[\hat{f}(x)]}_{\text{Overfitting}} + \underbrace{\sigma^2}_{\text{Irreducible}}$$

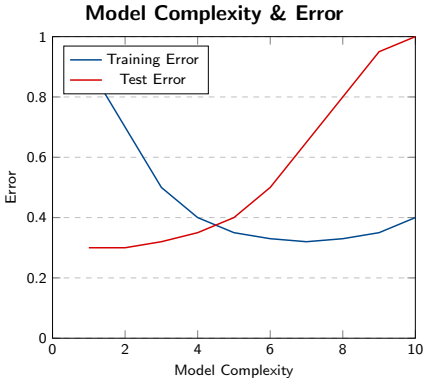
where:

$$\text{Bias}[\hat{f}(x)] = \mathbb{E}[\hat{f}(x)] - f(x)$$
$$\text{Var}[\hat{f}(x)] = \mathbb{E}[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2]$$
$$\sigma^2 = \text{Var}[\epsilon]$$

Finance Interpretation:

- **Bias**: Systematic pricing errors
- **Variance**: Model instability
- **Noise**: Market microstructure





Part 2: Statistical Learning Theory

Mathematical Foundations for Financial ML

Core Concepts

Random Variable:

$$X : \Omega \to \mathbb{R}$$

Expectation:

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x f_X(x) dx$$

Variance:

$$\text{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

Covariance:

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$$

Correlation:

$$\rho_{X,Y} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

Financial Applications

Return Distribution:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} \sim \mathcal{N}(\mu, \sigma^2)$$

Portfolio Variance:

$$\sigma_p^2 = w^T \Sigma w$$

where Σ is covariance matrix

Value at Risk (95%):

$$\mathbb{P}(L > \text{VaR}_{0.95}) = 0.05$$

Kelly Criterion:

$$f^* = \frac{p(b+1) - 1}{b}$$

where f^* = optimal fraction to bet

Fundamental Formula

$$P(H|D) = \frac{P(D|H) \cdot P(H)}{P(D)}$$

Components:

- $P(H)$: Prior probability
- $P(D|H)$: Likelihood
- $P(H|D)$: Posterior probability
- $P(D)$: Evidence (normalizing constant)

Expanded Form:

$$P(H_i|D) = \frac{P(D|H_i)P(H_i)}{\sum_j P(D|H_j)P(H_j)}$$

Finance Example: Fraud Detection

Given:

- P(Fraud) = 0.001 (prior)
- P(Alert—Fraud) = 0.95
- P(Alert—Normal) = 0.02

Find: P(Fraud—Alert)

Solution:

$$P(F|A) = \frac{0.95 \times 0.001}{0.95 \times 0.001 + 0.02 \times 0.999}$$

$$= \frac{0.00095}{0.02093} = 0.045 = 4.5\%$$

Maximum Likelihood

Given data $\mathcal{D} = \{x_1, ..., x_n\}$

Likelihood Function:

$$\mathcal{L}(\theta|\mathcal{D}) = \prod_{i=1}^{n} p(x_i|\theta)$$

Log-Likelihood:

$$\ell(\theta) = \sum_{i=1}^{n} \log p(x_i|\theta)$$

MLE Estimate:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \ell(\theta)$$

Example: Normal Returns

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} r_i$$

$$\frac{1}{n} \sum_{i=1}^{n}$$

Bayesian Inference

Posterior Distribution:

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta) \cdot p(\theta)$$

Conjugate Priors:

- Beta-Binomial
- Normal-Normal
- Gamma-Poisson

Black-Litterman Model: Combine market equilibrium (prior) with views (likelihood):

$$\mathbb{E}[R|\text{views}] = \left[(\tau\Sigma)^{-1} + P^T\Omega^{-1}P\right]^{-1} \times \left[(\tau\Sigma)^{-1}\Pi + P^T\Omega^{-1}Q\right]$$

Learning Guarantees

A learning algorithm is PAC if:
Given:

- Error parameter $\epsilon > 0$
- Confidence parameter $\delta > 0$
- Training sample size m

It outputs hypothesis h such that:

$$\mathbb{P}[\text{error}(h) \leq \epsilon] \geq 1 - \delta$$

Sample Complexity Bound:

$$m \geq \frac{1}{\epsilon}\left(\ln|\mathcal{H}| + \ln\frac{1}{\delta}\right)$$

where $|\mathcal{H}|$ is hypothesis space size

Finance Interpretation:

- ϵ: Maximum tolerable error (e.g., 5% mispricing)
- δ: Risk of failure (e.g., 1% chance)
- m: Required historical data

Example: Trading Strategy

- Want: 95% confidence
- Max error: 2%
- Hypothesis space: 1000 strategies
- Need: $m \geq \frac{1}{0.02}(\ln 1000 + \ln 100) \approx 689$ samples

Capacity of Learning Algorithms

Definition: The VC dimension of hypothesis class $\mathcal{H}$ is the maximum number of points that can be shattered (classified in all possible ways) by $\mathcal{H}$.

Examples:

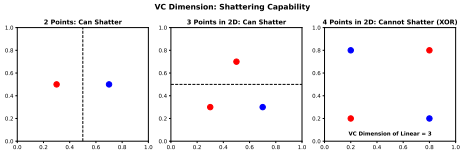
- Linear classifiers in $\mathbb{R}^d$: VC $= d + 1$
- Decision trees depth k: VC $\approx 2^k$
- Neural networks: VC $\propto$ #parameters

Generalization Bound: With probability $1 - \delta$:

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{d(\ln(2m/d) + 1) + \ln(4/\delta)}{m}}$$

where:

- $R(h)$: True risk
- $\hat{R}(h)$: Empirical risk
- d: VC dimension



Trading Strategy Complexity:

- Simple MA crossover: VC ≈ 3
- Multi-factor model: VC ≈ 20
- Deep neural network: VC $\approx 10{,}000$

Warning: Higher VC = More overfitting risk!

Core Measures

Entropy (Uncertainty):

$$H(X) = -\sum_x p(x) \log_2 p(x)$$

Cross-Entropy (Loss):

$$H(p, q) = -\sum_x p(x) \log q(x)$$

KL Divergence:

$$D_{KL}(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

Mutual Information:

$$I(X; Y) = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

Finance Applications

Portfolio Diversification:

$$H(\text{portfolio}) = -\sum_i w_i \log w_i$$

Maximum entropy = equal weights
Market Efficiency:

$$I(\text{Signal}; \text{Returns}) \approx 0$$

Efficient markets have low MI
Model Selection (AIC):

$$AIC = 2k - 2\ln(\mathcal{L})$$

where k = number of parameters
Active Information Ratio:

$$IR = \frac{\alpha}{\omega} = \sqrt{\text{Breadth} \times IC^2}$$

Part 3: Supervised Learning Methods

Prediction and Classification in Finance

Linear Regression Family

Ordinary Least Squares:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

$$\min_{\beta} ||y - X\beta||_2^2$$

Ridge Regression (L2):

$$\hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T y$$

$$\min_{\beta} ||y - X\beta||_2^2 + \lambda ||\beta||_2^2$$

LASSO (L1):

$$\min_{\beta} ||y - X\beta||_2^2 + \lambda ||\beta||_1$$

No closed form - use coordinate descent

Finance Applications:
Factor Models:

$$R_{i,t} = \alpha_i + \sum_{j=1}^{k} \beta_{i,j} F_{j,t} + \epsilon_{i,t}$$

Risk Premia Estimation: Fama-MacBeth regression
Elastic Net (Best of Both):

$$\min_{\beta} ||y - X\beta||_2^2 + \lambda_1 ||\beta||_1 + \lambda_2 ||\beta||_2^2$$

Useful for correlated predictors

Optimization Problem

Primal Form:

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$\text{s.t. } y_i(w^T x_i + b) \geq 1, \forall i$$

Dual Form:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

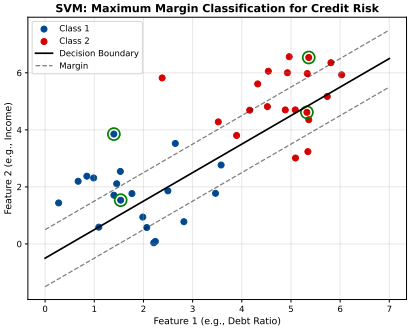
$$\text{s.t. } \alpha_i \geq 0, \sum_i \alpha_i y_i = 0$$

Kernel Trick:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

Common kernels:

- RBF: $K(x, z) = e^{-\gamma||x-z||^2}$
- Polynomial: $K(x, z) = (x^T z + c)^d$



Credit Default Application:

- Features: Financial ratios
- Target: Default/No default
- Kernel: RBF for non-linearity
- Result: 92% accuracy

Decision Trees

Splitting Criterion:

Gini Impurity:

$$G = \sum_{k=1}^{K} p_k(1 - p_k)$$

Information Gain:

$$IG = H(parent) - \sum_{j} \frac{n_j}{n} H(child_j)$$

CART Algorithm:

1. Find best split
2. Partition data
3. Recurse on children
4. Prune tree

Ensemble Methods

Random Forest:

- Bootstrap samples
- Random feature subsets
- Average predictions

Gradient Boosting:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

where h_m fits residuals

XGBoost Objective:

$$\mathcal{L} = \sum_{i} l(y_i, \hat{y}_i) + \sum_{k} \Omega(f_k)$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2$$

Architecture and Learning

Forward Propagation:

$$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$$

$$a^{[l]} = g^{[l]}(z^{[l]})$$

Backpropagation:

$$\frac{\partial \mathcal{L}}{\partial W^{[l]}} = \frac{\partial \mathcal{L}}{\partial z^{[l]}} \cdot a^{[l-1]T}$$

Update Rule (SGD):

$$W^{[l]} := W^{[l]} - \alpha \frac{\partial \mathcal{L}}{\partial W^{[l]}}$$

Activation Functions:

- ReLU: $\max(0, x)$
- Sigmoid: $\frac{1}{1+e^{-x}}$
- Tanh: $\frac{e^x - e^{-x}}{e^x + e^{-x}}$

Finance Applications:

Option Pricing NN:

- Input: S, K, T, r, σ
- Hidden: 3 layers, 100 units
- Output: Option price
- Loss: MSE vs Black-Scholes

Universal Approximation: Any continuous function on compact set can be approximated to arbitrary accuracy

Regularization:

- Dropout: $p = 0.5$
- L2 weight decay
- Early stopping

Part 4: Unsupervised Learning

Discovering Structure in Financial Data

Hierarchical Risk Parity

$$w_i = \frac{1/\sigma_i}{\sum_{j=1}^{n} 1/\sigma_j}$$

Clustering reveals natural asset groupings for better diversification.

Part 5: Risk & Portfolio Management

ML-Enhanced Risk Analytics

Markowitz Optimization Enhanced

Classic Formulation:

$$\min_{w} \quad w^T \Sigma w$$

$$\text{s.t.} \quad w^T \mu \geq r_{target}$$

$$w^T \mathbf{1} = 1$$

ML Enhancements:

- Shrinkage estimators for Σ
- Factor models for dimensionality
- Regime-switching covariance

Part 6: Algorithmic Trading

ML on the Trading Floor

Microstructure Prediction

Order Book Dynamics:

$$P_{t+\Delta t} = f(LOB_t, Trades_t, \text{Features}_t)$$

ML captures complex non-linear microstructure patterns.

Part 7: Credit & Fraud

Protecting Financial Systems

From FICO to Deep Learning

Probability of Default:

$$PD = \frac{1}{1 + e^{-(\beta_0 + \beta^T X)}}$$

ML models capture non-linear relationships and interactions.

Part 8: Ethics & Regulation

Responsible AI in Finance

Preventing Discrimination

Fairness Metrics:

- Demographic Parity: $P(\hat{Y} = 1|A = 0) = P(\hat{Y} = 1|A = 1)$
- Equal Opportunity: $P(\hat{Y} = 1|Y = 1, A = 0) = P(\hat{Y} = 1|Y = 1, A = 1)$

Regulators require explainable and fair ML models.

Appendix: Mathematical Foundations

Proofs, Derivations, and Advanced Theory

From Stochastic Calculus to Option Pricing

Stock Price Dynamics:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

Ito's Lemma Application:

$$df = \left(\frac{\partial f}{\partial t} + \mu S \frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} \right) dt + \sigma S \frac{\partial f}{\partial S} dW_t$$

Risk-Neutral Valuation:

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = rV$$

Solution (European Call):

$$C = S_0 \Phi(d_1) - Ke^{-rT}\Phi(d_2)$$

where:

$$d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T}$$

Karush-Kuhn-Tucker Conditions

For optimization problem:

$$\min_x f(x) \quad \text{s.t.} \quad g_i(x) \leq 0, \quad h_j(x) = 0$$

KKT conditions (necessary for optimality):

1. Stationarity: $\nabla f(x^*) + \sum_i \lambda_i \nabla g_i(x^*) + \sum_j \mu_j \nabla h_j(x^*) = 0$
2. Primal feasibility: $g_i(x^*) \leq 0, h_j(x^*) = 0$
3. Dual feasibility: $\lambda_i \geq 0$
4. Complementary slackness: $\lambda_i g_i(x^*) = 0$

Application: Portfolio Optimization with Constraints

Thank You

Questions & Discussion

Contact: finance.ml@university.edu