

Machine Learning in Finance: Theory & Applications

A Mathematical Foundation for Financial ML

Advanced Course in Quantitative Finance

4-Hour Comprehensive Workshop

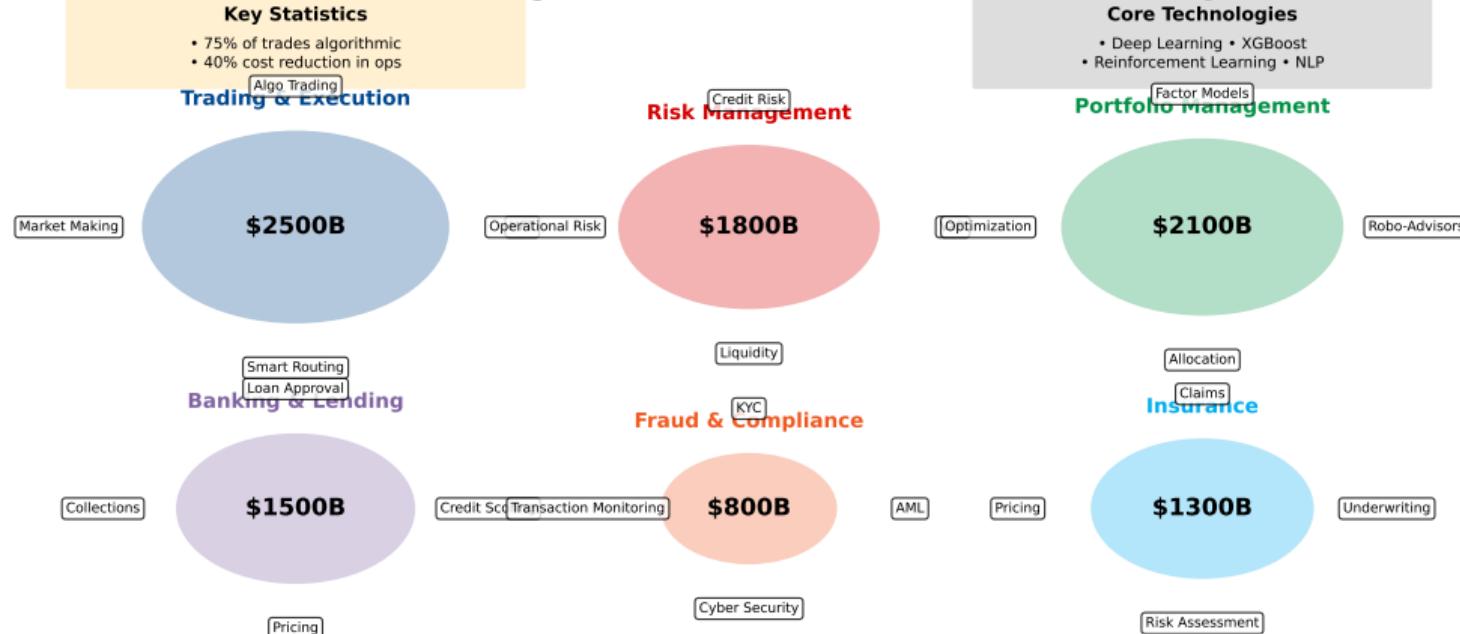
From Theory to Trading Floor

Part 1: Machine Learning Foundations

Theory, Mathematics, and Finance Applications

The \$10 Trillion ML Revolution in Finance

Machine Learning in Finance: \$10 Trillion Market Impact



Annual Growth Rates



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 $f : \mathcal{X} \rightarrow \mathcal{Y}$

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

Traditional Finance

Black-Scholes-Merton (1973)

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

- 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

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

- 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



Unsupervised



Reinforcement



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

Finance Applications:

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

Key Algorithms:

- Random Forest
- XGBoost
- Neural Networks

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

Finance Applications:

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

Key Algorithms:

- K-means
- PCA/ICA
- Autoencoders

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

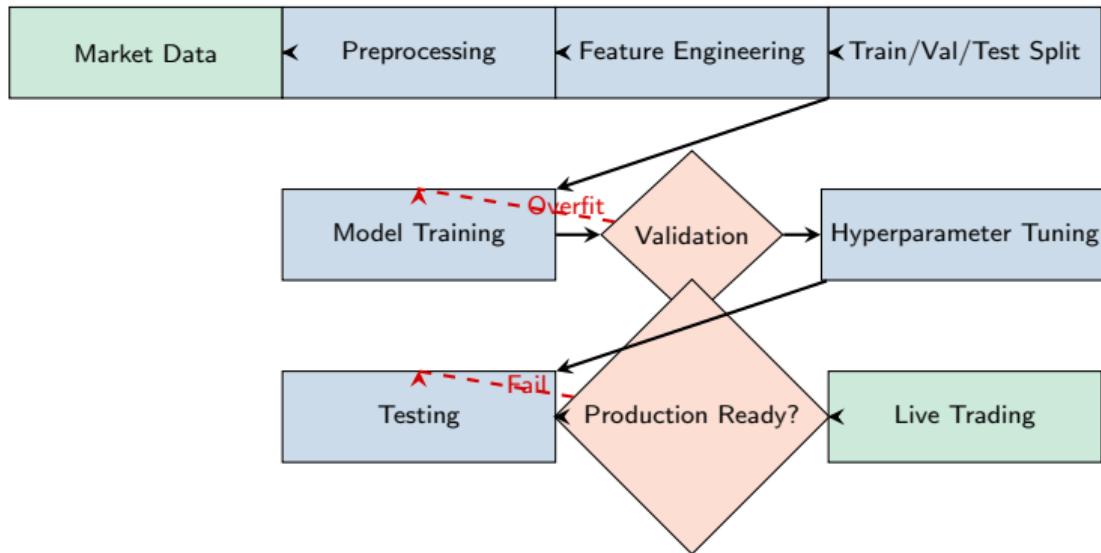
Finance Applications:

- 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):

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

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

The Bias-Variance Decomposition

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:

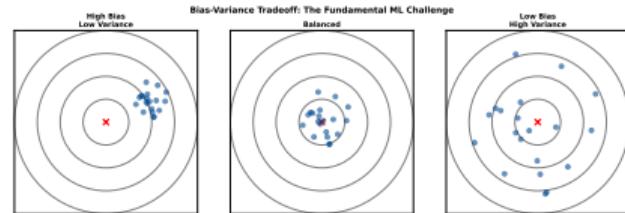
$$\text{Bias}[\hat{f}(x)] = \mathbb{E}[\hat{f}(x)] - f(x)$$

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

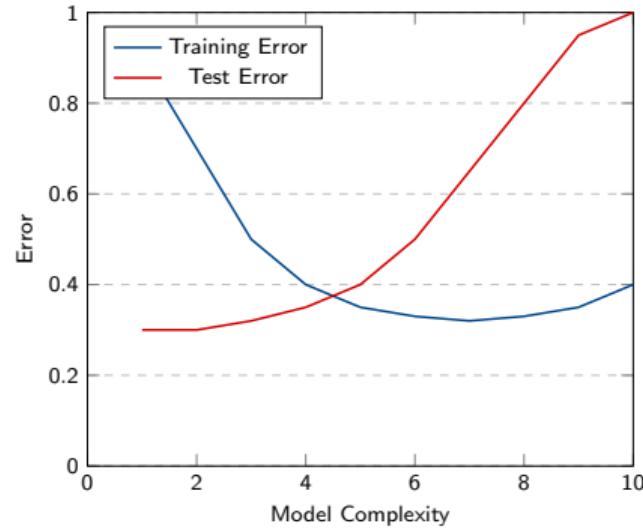
$$\sigma^2 = \text{Var}[\epsilon]$$

Finance Interpretation:

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



Model Complexity & Error



Part 2: Statistical Learning Theory

Mathematical Foundations for Financial ML

Core Concepts

Random Variable:

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

Expectation:

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

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:

- $P(\text{Fraud}) = 0.001$ (prior)
- $P(\text{Alert}|\text{Fraud}) = 0.95$
- $P(\text{Alert}|\text{Normal}) = 0.02$

Find: $P(\text{Fraud}|\text{Alert})$

Solution:

$$\begin{aligned} P(F|A) &= \frac{0.95 \times 0.001}{0.95 \times 0.001 + 0.02 \times 0.999} \\ &= \frac{0.00095}{0.02093} = 0.045 = 4.5\% \end{aligned}$$

Maximum Likelihood

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

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

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

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):

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

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:

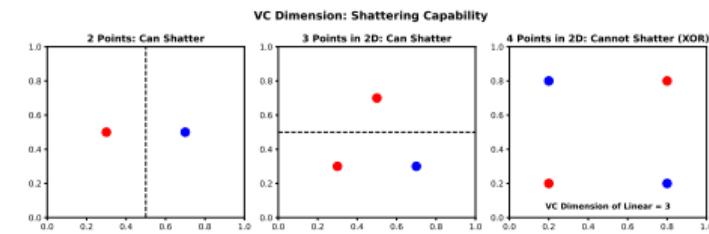
- Linear classifiers in \mathbb{R}^d : $VC = d + 1$
- Decision trees depth k : $VC \approx 2^k$
- Neural networks: $VC \propto \# \text{parameters}$

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: $VC \approx 3$
- Multi-factor model: $VC \approx 20$
- 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:

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

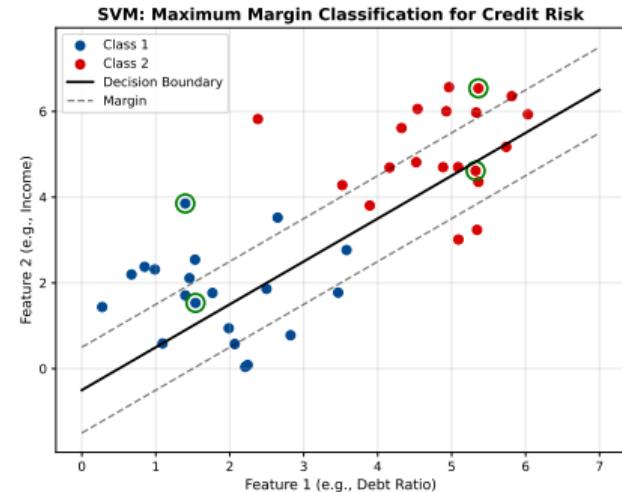
$$\text{s.t. } \alpha_i \geq 0, \sum_i \alpha_i y_i = 0$$

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:

$$IG = H(\text{parent}) - \sum_j \frac{n_j}{n} H(\text{child}_j)$$

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

Eigenportfolios and Risk Factors

Covariance Matrix Decomposition:

$$\Sigma = V \Lambda V^T$$

where V contains eigenvectors (factor loadings), Λ contains eigenvalues (factor variances).

Principal Components as Factors:

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

Variance Explained:

$$\text{Explained} = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i}$$

Typical Results:

- First 3 PCs explain 60-80% of equity return variance
- PC1: Market factor (beta)
- PC2: Size factor

Finance Applications: Factor Models:

- Fama-French 3-factor model
- Statistical arbitrage
- Risk attribution
- Portfolio hedging

Advantages:

- Dimensionality reduction (1000 stocks -> 10 factors)
- Interpretable risk sources
- Computational efficiency
- Noise filtering

Limitations:

- Linear assumptions
- Unstable in crisis periods
- Ignores tail dependencies

Clustering-Based Portfolio Construction

Algorithm Steps:

- ① Compute correlation matrix from returns
- ② Hierarchical clustering of assets
- ③ Recursive bisection of dendrogram
- ④ Inverse-variance weighting within clusters

Weight Formula:

$$w_i = \frac{1/\sigma_i^2}{\sum_{j \in \text{cluster}} 1/\sigma_j^2}$$

Cluster Allocation:

$$w_{\text{cluster}} \propto \frac{1}{\sigma_{\text{cluster}}^2}$$

Key Properties:

- Stable to estimation error
- No matrix inversion required
- Accounts for hierarchical structure
- Robust to outliers

vs Traditional Markowitz:

| Metric | MVO | HRP |
|-------------------|------|------|
| Stability | Low | High |
| Turnover | High | Low |
| Out-sample Sharpe | 0.4 | 0.6 |
| Max Drawdown | -35% | -22% |

Why HRP Works:

- Clustering identifies natural asset groups
- Hierarchical structure captures market organization
- Diversification across and within clusters
- Reduces estimation error impact

Applications:

- Multi-asset allocation
- Pension fund management
- Risk parity strategies

Identifying Market States

Feature Engineering:

- Volatility: Rolling 20-day std
- Momentum: 60-day return
- Volume: Relative to 90-day average
- Correlation: Cross-asset correlation

Clustering Process:

$$\min_{\mu_1, \dots, \mu_K} \sum_{k=1}^K \sum_{t \in C_k} \|x_t - \mu_k\|^2$$

Typical Regimes Discovered:

- ① **Bull Market:** Low vol, positive momentum, low correlation
- ② **Bear Market:** High vol, negative momentum, high correlation
- ③ **Consolidation:** Medium vol, low momentum, medium correlation
- ④ **Crisis:** Extreme vol, negative momentum, correlation -1

Trading Applications: Regime-Dependent Strategies:

- Bull: Long-only momentum
- Bear: Defensive hedging
- Consolidation: Mean reversion
- Crisis: Risk-off positioning

Performance Metrics:

| Approach | Sharpe |
|-----------------|--------|
| Static strategy | 0.5 |
| Regime-aware | 0.9 |

Implementation:

- `sklearn.cluster.KMeans`
- Weekly regime updates
- 3-5 regimes typical
- Combine with HMMs for transitions

Identifying Unusual Market Behavior

One-Class SVM Approach:

$$\min_{w, \rho} \frac{1}{2} \|w\|^2 - \rho + \frac{1}{\nu n} \sum_i \xi_i$$

$$\text{s.t. } w^T \phi(x_i) \geq \rho - \xi_i$$

Isolation Forest:

- Random feature selection
- Random split values
- Anomaly score: Average path length
- Faster than distance-based methods

Applications:

- ① **Flash crashes:** Detect abnormal price movements
- ② **Fat finger errors:** Identify erroneous orders
- ③ **Market manipulation:** Spot pump-and-dump schemes
- ④ **System failures:** Data feed anomalies

Real-World Example: 2010 Flash Crash Detection:

- Normal: SPY volatility 15-25%
- May 6, 2010: Volatility spike to 150%
- Anomaly score: 12 std deviations
- Duration: 5 minutes

Detection Metrics:

- Precision: 95% (few false alarms)
- Recall: 88% (catches most anomalies)
- Latency: \downarrow 100ms detection time

ML Techniques:

- Autoencoders for reconstruction error
- LSTM for temporal anomalies
- Isolation Forest for multivariate
- Ensemble methods for robustness

Part 5: Risk & Portfolio Management

ML-Enhanced Risk Analytics

Risk Measurement Fundamentals

Definition:

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

α -quantile of loss distribution (typically $\alpha = 0.05$ or 0.01)

1. Historical VaR:

- Use empirical quantile from historical data
- $\text{VaR}_{0.05} = 5\text{th percentile of returns}$
- Non-parametric, distribution-free
- Assumes past predicts future

2. Parametric VaR (Gaussian):

$$\text{VaR}_\alpha = \mu + \sigma \Phi^{-1}(\alpha)$$

where Φ^{-1} is inverse normal CDF

3. Monte Carlo VaR:

- Simulate 10,000+ scenarios
- Estimate empirical quantile
- Flexible for complex portfolios

ML-Enhanced VaR: GARCH-based VaR:

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

Time-varying volatility improves accuracy

Quantile Regression:

$$\min_{\beta} \sum_i \rho_\alpha(r_i - x_i^T \beta)$$

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

Directly estimates quantiles without distributional assumptions

Comparative Performance:

| Method | Coverage | Violations |
|-------------|----------|------------|
| Historical | 95.2% | 48/1000 |
| Gaussian | 92.8% | 72/1000 |
| GARCH | 94.8% | 52/1000 |
| Quantile RF | 95.1% | 49/1000 |

CVaR: Coherent Risk Measure

Definition:

$$\text{CVaR}_\alpha = \mathbb{E}[L | L > \text{VaR}_\alpha]$$

Average loss beyond VaR threshold

Mathematical Properties:

- **Sub-additivity:** $\text{CVaR}(X + Y) \leq \text{CVaR}(X) + \text{CVaR}(Y)$
- **Monotonicity:** $X \leq Y \Rightarrow \text{CVaR}(X) \leq \text{CVaR}(Y)$
- **Positive homogeneity:** $\text{CVaR}(\lambda X) = \lambda \text{CVaR}(X)$
- **Translation invariance:** $\text{CVaR}(X + c) = \text{CVaR}(X) + c$

Optimization Form:

$$\text{CVaR}_\alpha = \min_t \left\{ t + \frac{1}{\alpha} \mathbb{E}[\max(L - t, 0)] \right\}$$

Enables portfolio optimization with CVaR constraints

Why CVaR $\not\sim$ VaR: Advantages:

- Captures tail risk severity
- Coherent risk measure
- Sub-additive (diversification)
- Optimization-friendly

VaR Limitations:

- Not sub-additive
- Ignores tail losses
- Multiple local minima

Example: Portfolio Loss

- $\text{VaR}_{0.05}$: -\$1M (95th percentile)
- $\text{CVaR}_{0.05}$: -\$2.5M (avg beyond VaR)
- Tail risk: CVaR captures 2.5x worse scenarios

Regulatory Adoption:

- Basel III prefers ES over VaR
- Better capital adequacy
- Captures tail dependencies

Scenario Generation and Analysis

Traditional Stress Testing:

- Historical scenarios (2008 crisis, COVID-19)
- Hypothetical shocks (+3 std vol, -20% market)
- Reverse stress testing (find breaking point)

ML-Enhanced Approaches:

1. VAE Scenario Generation:

$$z \sim \mathcal{N}(0, I), \quad x = \text{Decoder}(z)$$

Generate plausible but unseen market scenarios

2. GAN Stress Scenarios:

- Train on crisis periods
- Generate extreme but realistic scenarios
- Capture tail dependencies

3. Random Forest Sensitivity:

- Feature importance for risk drivers
- Partial dependence plots
- Interaction effects

Stress Testing Framework:

Steps:

- ① Define risk factors (rates, spreads, FX)
- ② Generate stressed scenarios
- ③ Revalue portfolio under stress
- ④ Measure impact on P&L, VaR, capital

Example Scenarios:

| Scenario | Portfolio Loss |
|----------------------|----------------|
| Rate +200bp | -\$15M |
| Equity -30% | -\$45M |
| Credit spread +150bp | -\$22M |
| Combined (ML) | -\$68M |

ML Advantages:

- Discovers non-obvious correlations
- Generates tail scenarios
- Faster than Monte Carlo
- Learns from recent crises

Time-Varying Volatility Models

GARCH(1,1) Specification:

$$r_t = \mu + \epsilon_t, \quad \epsilon_t = \sigma_t z_t, \quad z_t \sim \mathcal{N}(0, 1)$$

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

Parameters:

- ω : Baseline variance (long-run vol)
- α : ARCH effect (shock impact)
- β : GARCH effect (persistence)
- Stationarity: $\alpha + \beta < 1$

Extensions:

- **EGARCH**: Exponential (leverage effects)
- **GJR-GARCH**: Asymmetric (bad news \downarrow good news)
- **DCC-GARCH**: Dynamic conditional correlation

Maximum Likelihood Estimation:

$$\ell(\theta) = \sum_{t=1}^T \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\sigma_t^2) - \frac{\epsilon_t^2}{2\sigma_t^2} \right]$$

ML Hybrid Approaches: Neural Network GARCH:

$$\sigma_t^2 = NN(\epsilon_{t-1}, \dots, \epsilon_{t-p}, \sigma_{t-1}, \dots, \sigma_{t-q})$$

Learns nonlinear volatility dynamics

LSTM-GARCH:

- Captures long-memory effects
- Handles regime switches
- Outperforms standard GARCH

Forecast Performance (S&P 500):

| Model | RMSE | MAE |
|------------|------|------|
| GARCH(1,1) | 2.8% | 1.9% |
| EGARCH | 2.6% | 1.8% |
| LSTM-GARCH | 2.2% | 1.5% |

Applications:

- Option pricing (implied vol)
- Risk management (VaR forecasting)
- Portfolio allocation (volatility timing)

Part 6: Algorithmic Trading ML on the Trading Floor

Predictive Trading Signals

Signal Definition:

$$\alpha_{i,t} = \mathbb{E}[R_{i,t+1} | \mathcal{F}_t] - r_f$$

Excess return prediction conditional on information set

Feature Engineering:

- **Technical:** Moving averages, RSI, MACD, volume
- **Fundamental:** P/E, P/B, EPS growth, margins
- **Alternative:** Sentiment, news, satellite data
- **Market micro:** Bid-ask spread, order imbalance

ML Models for Alpha:

- ① **Random Forest:** Feature interactions, non-linearity
- ② **XGBoost:** State-of-art for tabular data
- ③ **LSTM:** Sequential dependencies, momentum
- ④ **Transformer:** Attention over time series

Signal Quality Metrics: Information Coefficient (IC):

$$IC = \text{Corr}(\alpha_t, R_{t+1})$$

Information Ratio:

$$IR = \frac{\text{Mean}(\alpha)}{\text{Std}(\alpha)} = IC \times \sqrt{\text{Breadth}}$$

Typical Performance:

| Metric | Value |
|--------------|-----------|
| IC (monthly) | 0.05-0.08 |
| IR (annual) | 0.8-1.2 |
| Sharpe | 1.5-2.0 |
| Hit rate | 52-54% |

Signal Combination:

$$\alpha_{\text{combined}} = \sum_i w_i \alpha_i$$

Ensemble of multiple signals improves robustness

Minimizing Market Impact

Execution Cost Components:

$$\text{Total Cost} = \text{Spread} + \text{Impact} + \text{Timing Risk}$$

VWAP (Volume-Weighted Average Price):

$$\text{VWAP} = \frac{\sum_t P_t V_t}{\sum_t V_t}$$

Match historical volume profile

TWAP (Time-Weighted Average Price):

$$q_t = \frac{Q}{T}, \quad \text{for } t = 1, \dots, T$$

Uniform time slicing

Implementation Shortfall (Almgren-Chriss):

$$\min_{\{q_t\}} \mathbb{E}[\text{Cost}] + \lambda \text{Var}[\text{Cost}]$$

Optimal Trade Schedule:

ML-Enhanced Execution: Deep Reinforcement Learning:

- State: Order book, inventory, time
- Action: Trade size at each time step
- Reward: -Cost - $\lambda \times \text{Risk}$
- Policy: DDPG, PPO algorithms

Performance vs Benchmarks:

| Strategy | Slippage (bps) |
|---------------------|----------------|
| VWAP | 8.2 |
| TWAP | 9.5 |
| IS (Almgren-Chriss) | 6.8 |
| Deep RL | 5.4 |

Advantages:

- Adapts to market conditions
- Learns optimal urgency
- Handles constraints dynamically

Providing Liquidity for Profit

Market Making Problem:

$$\max_{\delta^{bid}, \delta^{ask}} \mathbb{E}[PnL] - \lambda \text{Var}[PnL]$$

subject to inventory constraints

Bid-Ask Spread Optimization:

$$\delta^{bid} = S_0 - P^{bid}, \quad \delta^{ask} = P^{ask} - S_0$$

Inventory Management:

$$q_t^* = -\frac{\alpha}{\gamma\sigma^2}$$

where α = alpha signal, γ = risk aversion

Order Book Forecasting:

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

ML Features:

- Order book imbalance
- Weighted mid-price

High-Frequency Strategies: Statistical Arbitrage:

- Mean reversion at millisecond scale
- Cointegration pairs
- Lead-lag relationships
- Cross-venue arbitrage

ML Techniques:

- LSTM: Order flow prediction
- CNN: Limit order book patterns
- RL: Optimal quoting strategies
- Transformers: Multi-asset dependencies

Performance Metrics:

| Metric | Value |
|---------------------------|-----------|
| Sharpe Ratio | 5-15 |
| Latency | <1ms |
| Prediction R ² | 0.15-0.25 |
| Fill rate | 85-95% |

Part 7: Credit & Fraud Protecting Financial Systems

Probability of Default Modeling

Traditional FICO Score:

- Payment history: 35%
- Amounts owed: 30%
- Length of history: 15%
- Credit mix: 10%
- New credit: 10%

Linear scorecard: 300-850 range

Logistic Regression Baseline:

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

where X includes debt-to-income, payment history, utilization

ML Enhancements:

- **Random Forest:** Non-linear interactions
- **XGBoost:** Best predictive performance
- **Neural Networks:** Deep feature learning
- **Ensemble:** Combine multiple models

Model Comparison:

| Model | AUC | Gini |
|---------------|------|------|
| FICO | 0.72 | 0.44 |
| Logistic | 0.78 | 0.56 |
| Random Forest | 0.84 | 0.68 |
| XGBoost | 0.87 | 0.74 |
| Deep NN | 0.86 | 0.72 |

Alternative Data:

- Mobile phone usage patterns
- Social media activity
- Transaction history
- Geolocation data

Regulatory Considerations:

- Model explainability (GDPR, ECOA)
- Fairness constraints
- Adverse action notices
- Validation requirements

Expected Loss Framework

Expected Loss Decomposition:

$$EL = EAD \times PD \times LGD$$

- **EAD:** Exposure at Default
- **PD:** Probability of Default
- **LGD:** Loss Given Default ($1 - \text{Recovery Rate}$)

LGD Modeling Challenges:

- Bimodal distribution (full loss or high recovery)
- Limited default data
- Collateral valuation uncertainty
- Economic cycle dependencies

ML Approaches:

- ① **Two-stage:** Logistic (default?) + Regression (LGD)
- ② **Beta regression:** $LGD \in (0,1)$ naturally
- ③ **Quantile regression:** Estimate LGD distribution
- ④ **Survival analysis:** Time-to-recovery modeling

Feature Engineering: Borrower Characteristics:

- Industry sector
- Firm size, leverage
- Credit rating history
- Management quality

Loan Characteristics:

- Seniority (secured vs unsecured)
- Collateral type and quality
- Loan-to-value ratio
- Covenants and protections

Macroeconomic Conditions:

- GDP growth, unemployment
- Interest rate environment
- Sectoral stress indicators

Typical LGD Estimates:

| Loan Type | LGD |
|----------------|--------|
| Senior secured | 25-35% |

Real-Time Transaction Monitoring

Fraud Detection Pipeline:

- ① Feature extraction from transaction
- ② Real-time scoring ($< 100\text{ms}$)
- ③ Rule engine + ML model ensemble
- ④ Human review for high-risk cases

Key Features:

- **Transaction:** Amount, location, merchant, time
- **Behavioral:** Deviation from normal patterns
- **Historical:** Past fraud, dispute history
- **Network:** Graph-based (connected accounts)

Class Imbalance Problem:

- Fraud rate: 0.1-0.5% of transactions
- SMOTE oversampling for minority class
- Cost-sensitive learning (FP vs FN)
- Precision-recall optimization

ML Techniques: Supervised Models:

- Random Forest: Feature interactions
- XGBoost: Best precision-recall
- Neural Networks: Deep patterns

Unsupervised Anomaly:

- Isolation Forest
- One-class SVM
- Autoencoders (reconstruction error)

Performance Metrics:

| Model | Precision | Recall |
|------------|-----------|--------|
| Rules only | 45% | 62% |
| Logistic | 68% | 75% |
| XGBoost | 82% | 84% |
| Ensemble | 85% | 86% |

Business Impact:

- Reduced false positives (better CX)
- Faster detection (lower losses)
- Adaptive to new fraud patterns

Part 8: Ethics & Regulation

Responsible AI in Finance

Federal Reserve Supervisory Guidance

SR 11-7 Requirements:

- ① **Model Development:** Documentation, assumptions, limitations
- ② **Model Validation:** Independent review, backtesting
- ③ **Model Governance:** Approval, monitoring, controls

Three Lines of Defense:

- **1st Line:** Model developers and users
- **2nd Line:** Model risk management function
- **3rd Line:** Internal audit

Validation Components:

- Conceptual soundness review
- Ongoing monitoring (champion-challenger)
- Outcomes analysis (backtesting)
- Stress testing and scenario analysis

ML-Specific Challenges:

Black Box Problem:

- Deep networks lack interpretability
- Regulatory explainability requirements
- Need for model-agnostic explanations

Mitigation Strategies:

- SHAP values for feature importance
- LIME for local explanations
- Attention visualization
- Simplified surrogate models

Documentation Requirements:

- Training data provenance
- Hyperparameter selection rationale
- Performance metrics (in/out-sample)
- Limitations and failure modes
- Monitoring thresholds

Annual Review Mandate:

- Performance degradation checks

MiFID II, GDPR, and ECOA Compliance

Right to Explanation (GDPR Article 22):

- Automated decisions require human review
- Meaningful information about logic
- Consequences of processing
- Applies to EU customers

Equal Credit Opportunity Act (ECOA):

- Prohibits discrimination on protected attributes
- Adverse action notices required
- Must provide specific reasons for denial

Fairness Metrics:

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

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

Trade-offs: Impossible to satisfy all fairness criteria simultaneously (Chouldechova impossibility theorem)

Explainability Techniques: Global Explanations:

- Feature importance rankings
- Partial dependence plots
- Model-agnostic summaries

Local Explanations:

- SHAP values for individual predictions
- LIME: Local linear approximations
- Counterfactual examples

Bias Mitigation:

- ① Pre-processing: Reweighting, resampling
- ② In-processing: Fairness constraints
- ③ Post-processing: Threshold optimization

Model Cards:

- Intended use and limitations
- Training data characteristics
- Performance across subgroups
- Fairness metrics

MiFID II and Market Abuse Regulation

MiFID II Requirements:

- Algo trading registration and authorization
- Pre- and post-trade controls
- Kill switches and circuit breakers
- Audit trail for 5 years
- Annual self-assessment

Market Abuse Regulation (MAR):

- Prohibition of market manipulation
- Layering, spoofing illegal
- Insider trading detection
- Suspicious transaction reporting (STR)

Testing Requirements:

- Development environment testing
- Conformance testing (exchange rules)
- Stress testing under extreme conditions
- Kill switch functionality

ML Compliance Challenges:

Risks:

- Inadvertent market manipulation
- Flash crashes from feedback loops
- Unintended discrimination
- Model brittleness

Best Practices:

- Robust validation framework
- Continuous monitoring
- Human oversight requirements
- Graceful degradation

Regulatory Technology (RegTech):

- Automated compliance monitoring
- NLP for regulatory change detection
- ML for suspicious activity detection
- Real-time reporting automation

Future Trends:

- Global regulatory harmonization

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):

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

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