Finance Applications of ML - Basic Handout

Machine Learning for Smarter Innovation

1 Finance Applications of ML - Basic Handout

Target Audience: Finance professionals with no ML background **Duration**: 30 minutes reading **Level**: Basic (no math, practical focus)

1.1 ML in Finance: Overview

Machine learning transforms finance by: - Automating decisions that humans make slowly - Finding patterns in massive datasets - Making predictions based on historical data - Managing risk more precisely

---

1.2 Key Applications

1.2.1 1. Credit Scoring

Problem: Decide who gets a loan **ML Solution**: Predict default probability from application data

Benefits: - Faster decisions (seconds vs days) - More consistent than human judgment - Can process more data points - Reduces discrimination when done right

Example Features: Income, employment history, existing debts, payment history

1.2.2 2. Fraud Detection

Problem: Find suspicious transactions **ML Solution**: Flag anomalies that deviate from normal patterns

Benefits: - Real-time detection - Adapts to new fraud patterns - Reduces false positives over time - Scales to millions of transactions

Example: Credit card company blocks transaction in foreign country you've never visited

1.2.3 3. Algorithmic Trading

Problem: Execute trades optimally **ML Solution**: Predict price movements, optimize execution

Applications: - High-frequency trading (milliseconds) - Portfolio rebalancing - Market making - Sentiment-based trading

Note: Most retail investors shouldn't compete here - institutions have advantages

1.2.4 4. Portfolio Management

Problem: Allocate assets to maximize returns for given risk **ML Solution**: Optimize portfolios using predicted returns and correlations

Robo-Advisors: Automated portfolio management (Betterment, Wealthfront) - Low fees - Tax-loss harvesting - Automatic rebalancing

1.2.5 5. Risk Management

Problem: Quantify potential losses **ML Solution**: Better estimate of Value at Risk (VaR) and stress scenarios

Applications: - Market risk (price changes) - Credit risk (defaults) - Operational risk (fraud, errors) - Liquidity risk (can't sell assets)

1.3 Key Concepts

1.3.1 Value at Risk (VaR)

"What's the maximum I could lose in a bad day?"

Example: "95% VaR of $1M means there's a 5% chance of losing more than $1M"

1.3.2 Portfolio Optimization

"How do I balance risk and return?"

Key Idea: Diversification - don't put all eggs in one basket

1.3.3 Backtesting

"Would this strategy have worked in the past?"

Warning: Past performance doesn't guarantee future results. Overfitting is a major risk.

1.4 Regulatory Requirements

1.4.1 SR 11-7 (Federal Reserve)

- Model risk management
- Independent validation
- Ongoing monitoring
- Documentation requirements

1.4.2 MiFID II (Europe)

- Algorithmic trading controls
- Best execution requirements
- Transparency rules

1.4.3 Basel III

- Capital requirements
- Risk-weighted assets
- Stress testing

Key Point: Models in finance are heavily regulated. You can't just deploy any ML model.

1.5 When ML Works in Finance

1.5.1 Good Fit:

- Large historical datasets available
- Patterns are relatively stable
- Decisions are frequent and similar
- Speed matters
- Human bias is a concern

1.5.2 Poor Fit:

- Unprecedented events (black swans)
- Data is sparse or unreliable
- Regulations require human judgment
- Explainability is critical
- Market conditions change fundamentally

1.6 Common Pitfalls

1.6.1 1. Overfitting to Historical Data

- Strategy worked in backtest but fails live
- Solution: Out-of-sample testing, walk-forward validation

1.6.2 2. Look-Ahead Bias

- Using information that wouldn't have been available
- Solution: Strict temporal separation

1.6.3 3. Survivorship Bias

- Only analyzing companies that still exist
- Solution: Include delisted/bankrupt companies

1.6.4 4. Data Snooping

- Testing many strategies, keeping only winners
- Solution: Pre-register hypotheses, adjust for multiple testing

1.6.5 5. Ignoring Transaction Costs

- Strategy profitable on paper, loses money in practice
- Solution: Include realistic costs in backtests

1.7 Getting Started Checklist

1.7.1 For Credit/Risk Models:

☐ Understand regulatory requirements
☐ Document data sources and preprocessing
☐ Establish baseline (simple model)
☐ Plan for model monitoring
☐ Prepare explainability reports

1.7.2 For Trading Strategies:

☐ Use realistic backtesting
☐ Account for transaction costs
☐ Test on out-of-sample data
☐ Start with paper trading
☐ Size positions conservatively

1.8 Tools and Platforms

1.8.1 Python Libraries:

- **pandas**: Data manipulation
- **numpy**: Numerical computing
- **scikit-learn**: General ML
- **statsmodels**: Statistical models
- **QuantLib**: Derivatives pricing

1.8.2 Platforms:

- **Bloomberg Terminal**: Market data, analytics
- **Refinitiv**: Financial data
- **Alpaca**: Algorithmic trading API
- **QuantConnect**: Backtesting platform

1.9 Key Terms

Term	Definition
VaR	Maximum expected loss at confidence level
Sharpe Ratio	Risk-adjusted return measure
Alpha	Excess return above benchmark
Beta	Sensitivity to market movements

Term	Definition
Drawdown	Peak-to-trough decline
Backtesting	Testing strategy on historical data

1.10 Ethics in Finance ML

1.10.1 Fair Lending:

- Models must not discriminate on protected characteristics
- Even indirect discrimination (proxy variables) is problematic
- Regular fairness audits required

1.10.2 Market Manipulation:

- Some algorithmic strategies may be illegal
- Spoofing, layering, and front-running are prohibited
- Ensure compliance with market rules

1.10.3 Systemic Risk:

- Many algorithms using similar strategies can amplify crashes
- Flash crashes have occurred
- Consider market impact

1.11 Next Steps

1. **Learn**: Take a quantitative finance course
2. **Practice**: Use paper trading to test ideas
3. **Read**: Follow financial ML research
4. **Comply**: Understand regulatory requirements
5. **Proceed**: Read intermediate handout for implementation

In finance, the stakes are real money. Always validate thoroughly, comply with regulations, and remember that models can fail.