Lesson 5.1: Machine Learning Foundations for Finance – Practice Exercises

Digital Finance Course

Exercise 1: Classification vs. Regression

Classify each financial problem as Classification (C) or Regression (R), and propose an appropriate evaluation metric:

1. Predicting whether a credit card transaction is fraudulent
2. Estimating the expected loss on a mortgage portfolio
3. Deciding whether to approve or deny a personal loan application
4. Forecasting next month's Value at Risk (VaR) for a trading desk
5. Detecting whether a customer will churn within 90 days
6. Predicting the settlement price of a commodity futures contract

For each problem, also answer:

- Is the data likely to be imbalanced?
- What is the cost asymmetry (which type of error is more expensive)?

Exercise 2: Confusion Matrix Calculations

A fraud detection model processes 50,000 transactions. The results are:

	Predicted Legitimate	**Predicted Fraud**
Actual Legitimate	48,200	500
Actual Fraud	120	1,180

Calculate:

1. Precision for the fraud class
2. Recall for the fraud class
3. F1-score for the fraud class
4. Overall accuracy
5. If each missed fraud costs $5,000 and each false alarm costs $50, what is the total cost of model errors?

Exercise 3: Bias-Variance Diagnosis

Three credit scoring models are evaluated. Diagnose each and recommend an action:

Model	Train AUC	Test AUC	Diagnosis?
Logistic Regression	0.72	0.70	?
Decision Tree (depth=30)	0.99	0.65	?
XGBoost (tuned)	0.91	0.88	?

For each model:

1. Classify as underfitting, overfitting, or good fit
2. Explain the relationship between bias and variance for this model
3. Recommend one specific action to improve performance
4. Which model would you deploy in production? Justify your choice.

Exercise 4: Feature Engineering for Anti-Money Laundering (AML)

You have a dataset of bank transactions with these raw fields:

`transaction_id, account_id, timestamp, amount, currency, sender_country, receiver_country, transaction_type, account_age_days, account_balance`

Design 8 engineered features that could help detect money laundering. For each feature:

1. Name the feature
2. Write the formula or logic
3. Explain why it captures suspicious behavior
4. Categorize it: ratio, rolling window, lag, aggregation, or time-since-event

Bonus: Which of your features would be most informative for detecting "structuring" (breaking large transfers into many small ones to avoid reporting thresholds)?

Exercise 5: Model Comparison – Tree-Based Methods

You are building a credit default prediction model. Compare three approaches:

Criterion	Decision Tree	Random Forest	XGBoost
How trees are built	?	?	?
Overfitting risk	?	?	?
Interpretability	?	?	?
Handles missing data	?	?	?
Training speed	?	?	?
Typical AUC range	?	?	?

Questions:

1. Fill in the table with appropriate descriptions
2. If a regulator requires full model explainability, which model would you choose and why?
3. Under what conditions might random forest outperform XGBoost?

Exercise 6: ROC Curve Interpretation

Two credit scoring models are evaluated on the same test set. Their ROC curves yield:

- Model A (Logistic Regression): AUC = 0.78
- Model B (XGBoost): AUC = 0.91

Questions:

1. What does it mean that Model B's AUC is 0.91? Express this in plain language.
2. At a False Positive Rate (FPR) of 5%, Model A achieves a True Positive Rate (TPR) of 45% and Model B achieves 82%. What does this mean for the bank?
3. The bank processes 100,000 loan applications per year, of which 3,000 eventually default. Using Model B at a 5% FPR operating point:
   - How many defaults will it correctly identify?
   - How many non-defaulters will be incorrectly flagged?
   - What is the precision at this operating point?
4. Under what business scenario would you choose Model A despite its lower AUC?

Exercise 7: Anomaly Detection vs. Supervised Classification

A fintech company processes payments and wants to build a fraud detection system. They have:

- 5 million labeled transactions from 2022–2023 (0.3% confirmed fraud)
- Evidence that fraudsters changed tactics significantly in late 2023
- A compliance team that can investigate 500 alerts per day

Questions:

1. Would you use supervised classification, anomaly detection, or both? Justify your architecture.
2. If you choose anomaly detection (Isolation Forest), what contamination parameter would you set and why?
3. The compliance team capacity is 500 alerts/day. If the model processes 200,000 transactions/day, what maximum false positive rate is acceptable?
4. Design a two-stage system that combines both approaches. Draw the data flow and explain each stage.

Exercise 8: End-to-End ML Pipeline Design

Design a complete ML pipeline for predicting customer churn at a neobank.

Given:

- 2 million customers, 18 months of data, 8% annual churn rate
- Available data: transaction history, app usage, demographics, customer service contacts
- Requirement: model must be explainable (EU AI Act compliance)

For each pipeline stage, specify your choice and justify it:

1. **Problem framing:** Classification or regression? Target variable definition?
2. **Feature engineering:** List 5 features you would create (with formulas)
3. **Validation strategy:** How would you split the data? Why?
4. **Model selection:** Which model(s) would you train? Why?
5. **Evaluation metric:** Which metric is primary? Why not accuracy?
6. **Deployment:** How would you monitor for model drift?

Answer Key (1/3)

Exercise 1: 1) C – F1 or Recall; imbalanced (yes); missing fraud costlier. 2) R – RMSE or MAE; not imbalanced; underestimating loss costlier. 3) C – AUC (ranking); moderate imbalance; denying good applicants loses revenue, approving bad costs default losses. 4) R – RMSE; not imbalanced; underestimating VaR means insufficient capital reserves. 5) C – F1 or AUC; moderate imbalance (10–20% churn); missing churners means lost LTV. 6) R – RMSE or MAE; not imbalanced; symmetric cost usually.

Exercise 2: 1) Precision $= \frac{1180}{1180+500} = \frac{1180}{1680} = 70.2\%$ 2) Recall $= \frac{1180}{1180+120} = \frac{1180}{1300} = 90.8\%$ 3) $F_1 = 2 \cdot \frac{0.702 \times 0.908}{0.702+0.908} = 2 \cdot \frac{0.638}{1.610} = 79.2\%$ 4) Accuracy $= \frac{48200+1180}{50000} = 98.76\%$ 5) Cost $= 120 \times \$5{,}000 + 500 \times \$50 = \$600{,}000 + \$25{,}000 = \$625{,}000$. Missed fraud dominates total cost (96% of error cost).

Answer Key (2/3)

Exercise 3: 1) LR: Underfitting (both AUC low, small gap). DT: Severe overfitting (huge gap). XGBoost: Good fit (high AUC, small gap). 2) LR: High bias, low variance. DT: Low bias, very high variance. XGBoost: Balanced. 3) LR: Add nonlinear features or use a more complex model. DT: Reduce max_depth or add min_samples_leaf. XGBoost: Already good; fine-tune learning rate. 4) Deploy XGBoost: best generalization (0.88 test AUC) with manageable complexity. Use SHAP for explainability.

Exercise 4 (sample features): 1) `txn_count_30d`: count of transactions in last 30 days (rolling window). Detects sudden activity spikes. 2) `amt_to_balance_ratio`: amount / account_balance (ratio). Large transfers relative to balance are suspicious. 3) `cross_border_pct_7d`: % of cross-border transactions in 7 days (rolling window). Sudden international activity. 4) `avg_amt_vs_median`: transaction amount / median amount for account (ratio). Deviation from normal behavior. 5) `time_since_last_txn`: hours since previous transaction (time-since-event). Rapid-fire transactions. 6) `unique_receivers_7d`: count distinct receivers in 7 days (aggregation). Many recipients = distribution pattern. 7) `just_below_threshold`: flag if amount is 90–99% of reporting threshold (ratio). Classic structuring indicator. 8) `round_amount_flag`: binary flag if amount is perfectly round (derived). Suspicious pattern. Bonus: `just_below_threshold` directly targets structuring behavior.

Answer Key (3/3)

Exercise 5: Decision Tree: single tree (greedy splits); high overfitting risk; fully interpretable; imputation needed; fast; AUC 0.60–0.75. Random Forest: parallel trees on bootstrap samples; low overfitting; partial interpretability; handles missing (some implementations); moderate; AUC 0.80–0.90. XGBoost: sequential trees correcting errors; moderate risk (needs tuning); partial (SHAP); native missing handling; slower; AUC 0.85–0.95. Regulator requirement: Decision tree (or logistic regression). RF beats XGBoost when: little tuning time, noisy data, or when XGBoost hyperparameters are poorly chosen.

Exercise 6: 1) 91% probability that Model B ranks a random defaulter higher than a random non-defaulter. 2) At 5% FPR: Model A catches 45% of defaults, Model B catches 82%. Bank misses fewer defaults with B. 3) Defaults caught: $0.82 \times 3{,}000 = 2{,}460$. False flags: $0.05 \times 97{,}000 = 4{,}850$. Precision: $\frac{2460}{2460+4850} = 33.7\%$. 4) If explainability is paramount (regulation) and LR is inherently interpretable, or if computational resources are limited.

Exercise 7: 1) Both: supervised for known patterns, anomaly detection for novel fraud. 2) Contamination $\approx$ 0.003–0.005 (slightly above known fraud rate to catch unknowns). 3) Max FPR: $\frac{500}{200{,}000} = 0.25\%$. 4) Stage 1: Anomaly detection flags top 0.5% unusual transactions. Stage 2: Supervised classifier re-scores flagged transactions. Combined output sent to compliance.

Exercise 8: 1) Binary classification; target = churned within 90 days (yes/no). 2) Features: avg_monthly_txn_count, days_since_last_login, complaint_count_90d, balance_trend_slope, pct_recurring_payments. 3) Time-based split: train on months 1–12, validate 13–15, test 16–18. Never shuffle temporal data. 4) XGBoost (primary), logistic regression (baseline, explainable). Use SHAP for XGBoost explanations. 5) F1-score or AUC. Not accuracy because 92% non-churn baseline inflates accuracy. 6) Monitor: weekly AUC on recent predictions, feature distribution drift (PSI), churn rate shift. Retrain quarterly.