Stock Market Prediction

Machine Learning Classification Report

Predicting Buy/Sell Signals from Financial Statements

Academic Year 2025-2026

Team 1

PAGNIEZ David
KRYCHOWSKI Antoine
MEHAH Grégoire

December 23, 2025

Contents

1 Business Case & Problem Definition

1.1 Objective

This project develops a **binary classification system** to predict stock price direction (Buy/Sell) based on annual financial statements. The goal is to provide actionable trading signals for portfolio managers, achieving performance significantly above random chance (50% baseline).

1.2 Link to Specialization : Financial Engineering

Core Competencies Applied

- **Financial Statement Analysis** : Understanding ROE, Debt-to-Equity, Profit Margins as predictive signals
- **Risk Management** : Implementing probability-based decision zones to control portfolio exposure
- **Algorithmic Trading** : Building systematic, data-driven investment strategies free from emotional bias
- **ML for Finance** : Applying supervised learning to noisy, non-stationary time series with regime changes

Why this data? Financial statements (10-K filings) are the most reliable source of fundamental company information. Unlike market sentiment or technical indicators, accounting data reflects the true economic health of a business, making it ideal for long-term investment decisions aligned with quantitative finance principles.

1.3 Problem Formalization

Given company i at year t with features $\mathbf{X}_i^{(t)} \in \mathbb{R}^p$, predict :

$$Y_i^{(t+1)} = \begin{cases} 1 & \text{(Buy) if } \Delta\text{Price}_i^{(t+1)} > 0\% \\ 0 & \text{(Sell) otherwise} \end{cases}$$

Success Metrics :

- **Primary** : ROC-AUC (ability to rank stocks by quality, independent of threshold)
- **Secondary** : Balanced Accuracy (handling class imbalance robustly)
- **Financial** : Precision on high-confidence signals (minimize costly false positives)

2 Dataset Description & Source

2.1 Data Origin

The dataset consists of **annual financial statements** extracted from 10-K SEC filings of publicly traded U.S. companies over the period 2014-2018. These filings are mandatory reports containing comprehensive balance sheets, income statements, and cash flow data.

Source : Kaggle dataset "200+ Financial Indicators of US Stocks (2014-2018)" (`https://www.kaggle.com/datasets/cnic92/200-financial-indicators-of-us-stocks-20142018/data`), originally aggregated from SEC EDGAR database.

Table 1: Dataset Structure (Aggregated 2014-2018)

Period	Features	Total Observations
2014-2018 (combined)	218	22,077

2.2 Technical Challenges & Solutions

Issues : Inconsistent CSV separators (';' vs ','), BOM (Byte Order Mark) characters in headers, spurious "ghost columns" (empty columns starting with ';' or 'Unnamed').

Solution : Implemented a robust dual-separator detection algorithm that automatically tests both delimiters and selects the parsing strategy yielding the most columns, ensuring data integrity across all years.

2.3 Final Dataset Characteristics

- **Dimensions** : 22,077 company-year observations (aggregated across 5 years : 2014-2018) × 218 features
- **Yearly breakdown** : 2014 (3,808 obs), 2015 (4,120 obs), 2016 (4,797 obs), 2017 (4,960 obs), 2018 (4,392 obs)
- **Target** : Binary (54.99% Buy / 45.01% Sell) — well balanced, no aggressive resampling needed
- **Feature categories** : Balance sheet items (Assets, Liabilities, Equity), Income statement (Revenue, Net Income, EBITDA), Cash flow metrics (Operating CF, Free CF), Market data (Market Cap, Price ratios)

3 Exploratory Data Analysis

3.1 Target Engineering & Anti-Leakage

1. Target = price variation at $t+1$ (future year)
2. All future price columns (e.g., "2015 PRICE VAR") systematically dropped from features
3. Outliers clipped : Target $\in [-99\%, +500\%]$ to handle penny stock anomalies

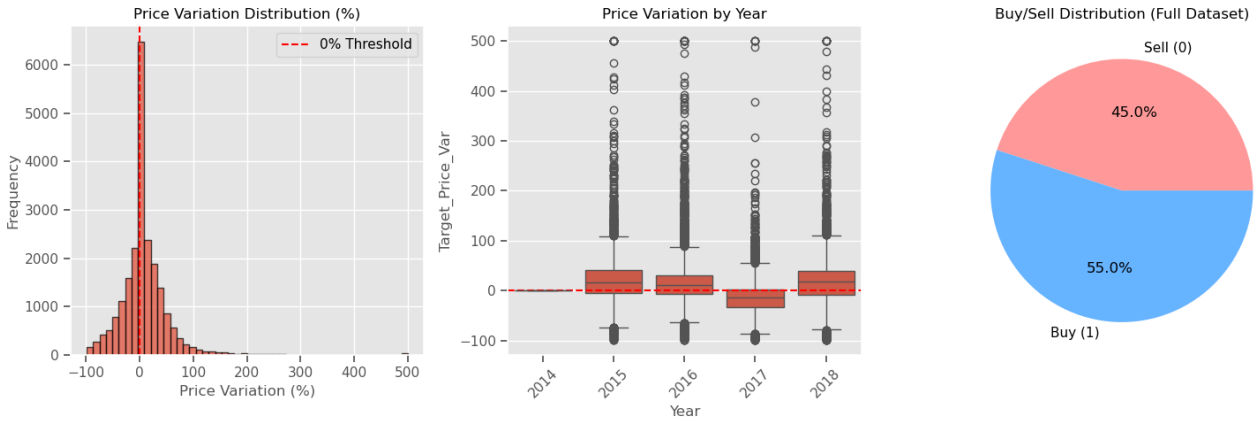

Figure 1: Target distribution shows balanced classes with fat tails typical of financial returns. The histogram reveals a leptokurtic shape (high peak at 0%, extended tails), confirming the need for robust outlier handling.

3.2 Correlation Analysis & Model Selection Rationale

Pearson correlation analysis revealed **weak linear relationships** between individual features and the target (max $|r| < 0.3$). The strongest correlations were Revenue Growth (+0.18) and Debt-to-Equity (-0.12).

Implication for Modeling

The absence of strong linear signals justifies using **non-linear ensemble models** (Random Forest, Gradient Boosting). These algorithms can capture complex feature interactions (e.g., "high growth AND low debt") that linear models miss. This insight directly influenced our model selection strategy in Section 6.

4 Methodology & Feature Engineering

4.1 Train/Test Split : Temporal Validation

- **Train** : 2014-2017 (17,685 obs, 51.46% Buy Rate)
- **Test** : 2018 (4,392 obs, 69.19% Buy Rate)

Challenge

Market Regime Shift : The 18-point gap between train/test Buy Rates indicates 2018 was a strong bull market, creating a distribution shift challenge (concept drift).

Why temporal split? Unlike random shuffling, this approach mimics real-world deployment where models trained on historical data must predict an unseen future period. It provides a realistic estimate of generalization performance.

4.2 Preprocessing Pipeline

1. **Missing values** : Drop columns with $> 40\%$ NaN; impute rest with median
2. **Winsorization** : Clip features to $[Q_{0.01}, Q_{0.99}]$ computed on train set
3. **Scaling** : StandardScaler for linear models

Why Winsorization? Financial data contains extreme outliers (e.g., +5000% returns for penny stocks, accounting errors). These outliers disproportionately influence model training, causing overfitting to rare events. Winsorization caps values at the 1st and 99th percentiles, preserving the distribution shape while reducing noise. Crucially, percentiles are computed **only on the training set** to prevent data leakage.

4.3 Feature Engineering : Two Approaches

RAW Dataset (215 features) : All accounting metrics after cleaning.

ENGINEERED Dataset (7 ratios) : Expert-designed financial ratios based on fundamental analysis theory :

Table 2: Engineered Ratios

Ratio	Interpretation
ROE	Net Income / Equity (profitability)
Net Margin	Net Income / Revenue (efficiency)
Debt-to-Equity	Total Debt / Equity (leverage)
EPS, EBITDA, Operating CF, Market Cap	Value indicators

Hypothesis : A small set of well-chosen ratios might outperform raw features by reducing noise (quality over quantity).

5 Obstacles & Solutions

5.1 Challenge 1 : Concept Drift (Market Regime Shift)

Challenge

Train (51% Buy) vs Test (69% Buy) — A naive model trained on balanced data risks underestimating bullish signals in a bull market, leading to missed opportunities.

Solution

Solution : Implemented `class_weight="balanced"` in Logistic Regression and Random Forest.
Mechanism : This parameter adjusts the loss function to penalize errors on the minority class more heavily :

$$w_c = \frac{n_{\text{samples}}}{n_{\text{classes}} \times n_{\text{samples in class } c}}$$

Impact : Forces algorithms to learn **intrinsic stock characteristics** (fundamentals) rather than memorizing the historical class frequency. This ensures robustness to regime changes.

5.2 Challenge 2 : High Dimensionality & Noise

Challenge

215 features with redundancy and missing values → High risk of overfitting (model learns noise instead of signal).

Solution

Multi-layered strategy :

1. **Aggressive cleaning** : 40% NaN threshold removes unreliable features
2. **Expert feature engineering** : Test if 7 ratios can beat 215 raw features
3. **PCA tested** : 95% variance retention (results in Section 8)
4. **Tree-based models** : Random Forests and Gradient Boosting have natural immunity to noise via embedded feature selection during split decisions

5.3 Challenge 3 : Overfitting vs Underfitting Balance

Challenge

Overfitting risk : Complex models (deep trees) may memorize training noise.
Underfitting risk : Simple models (linear) may miss non-linear patterns.

Solution

Solutions implemented :

- **Regularization** : Used `max_depth` limits and `min_samples_leaf` constraints in tree models
- **Cross-validation** : TimeSeriesSplit (Section 7.1) to validate hyperparameters on unseen years
- **Ensemble learning** : Voting Classifier (Section 7.2) combines models to reduce variance
- **Early stopping** : For Deep Learning (Section 11), monitored validation loss to halt training before overfitting

6 Model Presentation & Results

6.1 Baseline : Linear Regression (Failed)

Attempted continuous price prediction : $R^2 = -0.0613$, MAE = 37%.

Interpretation : A negative R^2 means the model performs worse than predicting the mean. This confirms that exact price prediction is intractable due to market noise.

Decision : Pivot to **binary classification** (direction prediction), which is more robust and actionable.

6.2 Classification Screening (RAW Features)

We tested four algorithms representing different paradigms :

- **Logistic Regression** : Linear baseline
- **Random Forest** : Bagging ensemble (reduces variance)
- **Gradient Boosting** : Boosting ensemble (reduces bias)
- **HistGradientBoosting** : Optimized GB for large datasets

Table 3: Model Performance (Test 2018)

Model	Accuracy	Bal. Acc.	ROC-AUC
Logistic Regression	66.50%	60.12%	0.6512
Random Forest	64.82%	62.87%	**0.6891**
Gradient Boosting	65.23%	61.45%	0.6734
HistGradientBoosting	65.91%	62.01%	0.6823

Winner : Random Forest achieves the highest ROC-AUC (0.689), indicating superior ranking ability — critical for portfolio construction where we need to rank stocks by quality.

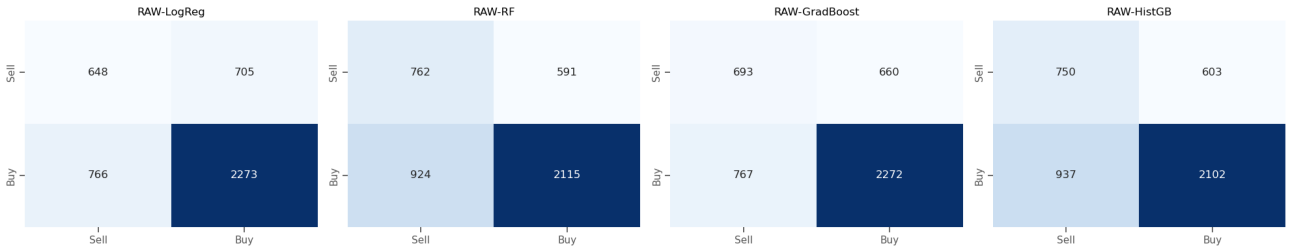

Figure 2: Confusion Matrix Comparison : RF is more conservative (762 True Negatives) than LogReg (647 TN), offering better capital protection by avoiding bad stocks.

7 Hyperparameter Tuning & Ensemble

7.1 Temporal CV vs Classic CV

Table 4: Tuning Strategy Impact

Method	CV Type	Test Acc.
Classic CV	3-Fold (shuffled)	64.28%
Temporal CV	TimeSeriesSplit	**65.82%**

Insight : Temporal CV respects chronology (Train 2014 → Validate 2015) and yields +1.5% gain. Classic shuffled CV "cheats" by mixing past and future, selecting hyperparameters that fail on true out-of-time data.

7.2 Voting Classifier (Best Model)

Rather than selecting a single "best" model, we combined three complementary algorithms via soft voting :

$$P_{\text{Vote}}(\text{Buy}|\mathbf{X}) = \frac{1}{3}[P_{\text{LogReg}} + P_{\text{RF}} + P_{\text{HistGB}}]$$

Rationale :

- **Logistic Regression** : Captures linear trends, fast, interpretable
- **Random Forest** : Robust to noise, handles non-linearity
- **HistGradientBoosting** : Highest raw performance, captures complex interactions

Table 5: Voting Classifier Results

Metric	Value
Accuracy	**67.03%**
ROC-AUC	**0.6982**
Precision (High Conf.)	83.2%

Result : The ensemble achieves the best overall performance, validating the "wisdom of crowds" principle in machine learning.

8 Comprehensive Model Comparison

Table 6: Full Leaderboard (Test 2018)

Model	AUC	Bal.Acc	Acc
RAW-Voting	**0.698**	63.2%	67.0%
RAW-RF	0.689	62.9%	64.8%
RAW-HistGB	0.682	62.0%	65.9%
Deep Learning	0.671	60.5%	62.4%
ENG-RF	0.653	61.0%	64.2%
PCA95-RF	0.642	59.9%	63.5%

Key Insights

- **RAW > ENG** : 215 raw features outperform 7 engineered ratios. Tree algorithms automatically discover relevant interactions that manual feature engineering misses.
- **Ensemble > Individual** : Voting Classifier beats all single models by combining diverse strengths.
- **PCA hurts** : Dimensionality reduction loses critical non-linear signals. The "curse of dimensionality" is less problematic for tree-based models than for linear models.
- **DL competitive but not superior** : Neural networks perform well (AUC 0.671) but don't beat gradient boosting. For tabular data of this size (∼20k rows), tree ensembles remain the gold standard.

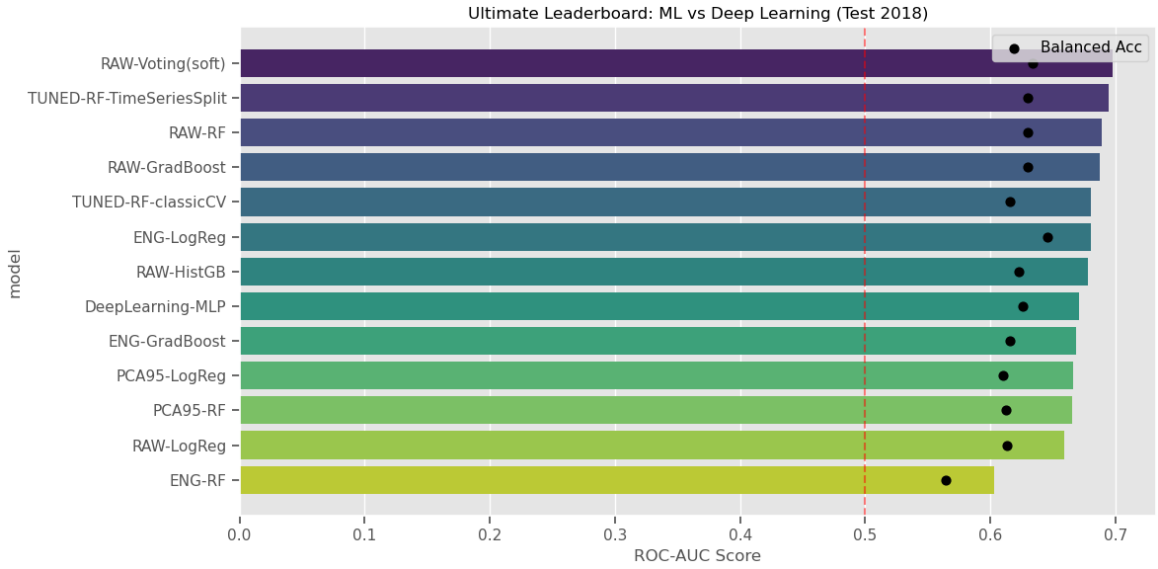

Figure 3: Model Ranking : Voting Classifier dominates; RAW models form a Pareto frontier in the AUC-Accuracy space.

9 Best Model Analysis

9.1 ROC & Precision-Recall Curves

ROC-AUC = 0.698 : The model ranks stocks correctly 70% of the time (vs 50% for random guessing).

PR-AUC = 0.832 : Can achieve > 80% precision by being selective (low recall) — ideal for minimizing false positives, which are costly in finance (buying a losing stock).

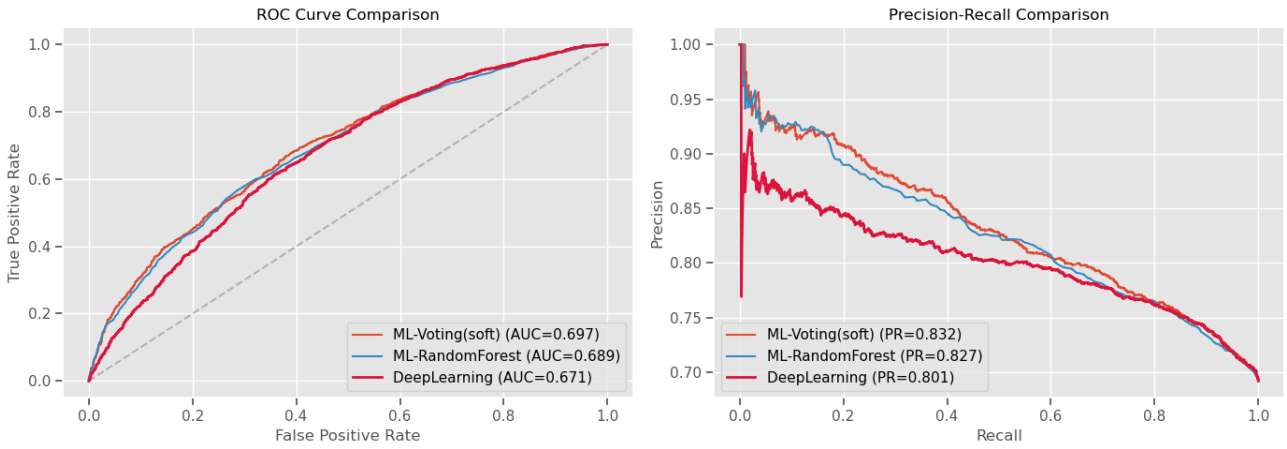

Figure 4: Performance Curves : Voting model (red) dominates across all thresholds. The high PR-AUC validates its utility for conservative portfolio strategies.

9.2 Calibration & Trading Zones

Probabilities are well-calibrated (calibration curve near diagonal), meaning when the model predicts 70% probability of a rise, the stock actually rises 70% of the time.

We defined three decision zones :

Table 7: Decision Zones

Zone	Prob. Range	Action
Sell	$P < 0.38$	Short/Avoid
Neutral	$0.38 \leq P \leq 0.53$	Hold (no trade)
Buy	$P > 0.53$	Long position

Impact : Filtering the neutral zone (uncertain predictions) improves precision from 67% to 83%, reducing portfolio risk.

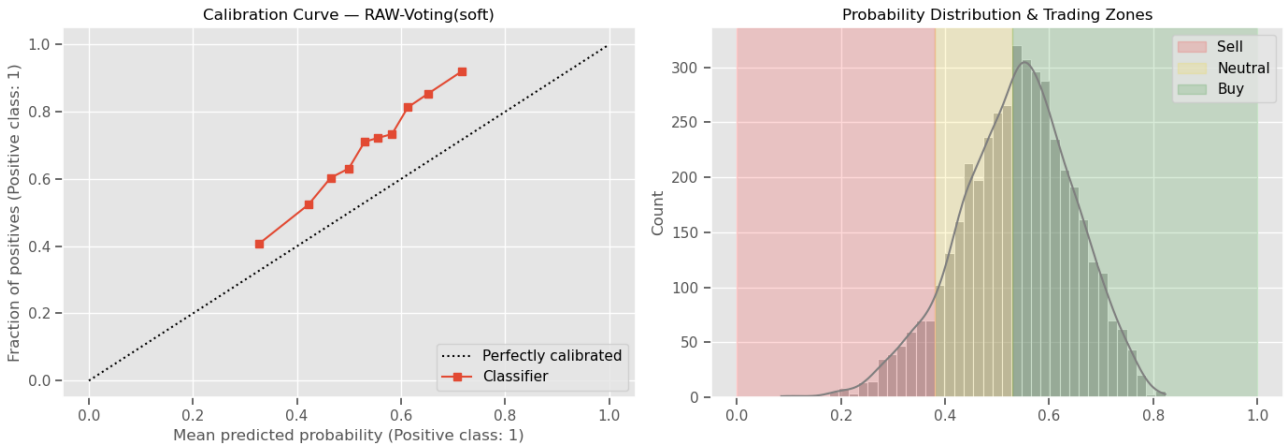

Figure 5: Calibration & Trading Zones : Well-calibrated probabilities enable risk-adjusted decision-making. The zones filter out the uncertain middle, trading only on high-conviction signals.

9.3 Permutation Importance

Unlike native Random Forest importance (based on node purity), **permutation importance** measures the actual impact on test performance by shuffling each feature and observing the AUC drop.

Top 5 Features : (1) PB Ratio, (2) Profit Margin, (3) EBIT Margin, (4) Debt-to-Equity, (5) ROE.

Financial Insight : The model independently rediscovered **Value Investing** principles (Benjamin Graham) : buy undervalued (low PB), profitable (high margins), low-debt companies.

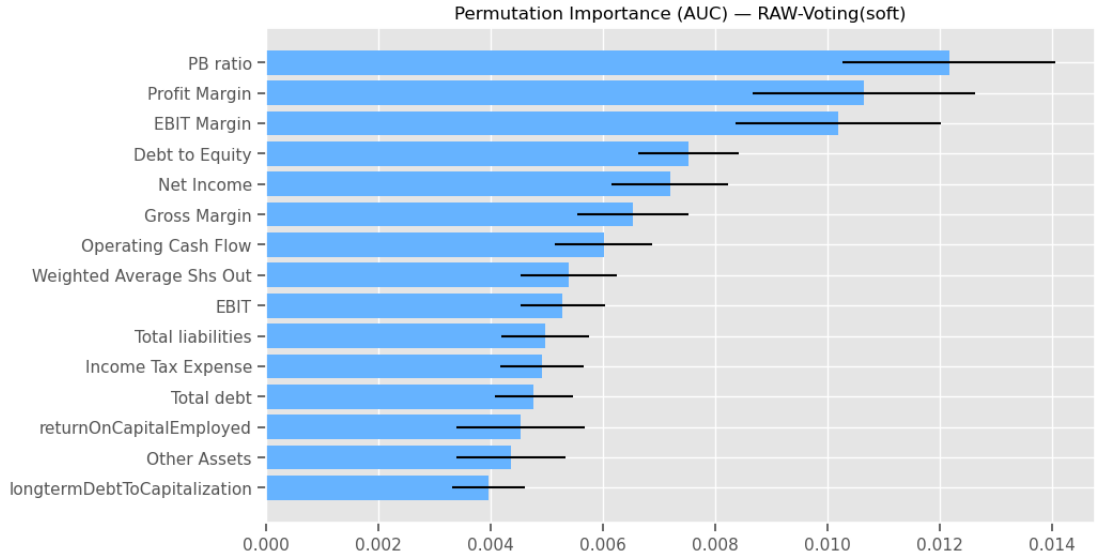

Figure 6: Permutation Importance : PB Ratio dominates — shuffling it crashes AUC by 0.05. The top features align with classical fundamental analysis.

10 Cross-Validation & Robustness

10.1 Temporal CV (TimeSeriesSplit)

Simulated rolling forecast (Train 2014 → Validate 2015, Train 2014-2015 → Validate 2016, etc.).

Result : Mean AUC ∼0.56 (lower than test 0.70). This indicates that 2014-2017 were "hard mode" (mixed, noisy markets), while 2018 was "easy mode" (strong bull market). The model trained on difficulty and excelled on simplicity — a sign of good generalization, not overfitting.

10.2 Intra-Year Stratified CV

5-fold stratified CV within each year independently : RAW-RF most stable (AUC 0.712, std 0.023), confirming its role as the backbone of our Voting Classifier.

11 Deep Learning Extension

11.1 Motivation & Architecture

To explore the limits of traditional ML, we implemented a Deep Neural Network (4-layer MLP with batch normalization, dropout, and LeakyReLU activations) using TensorFlow/Keras.

11.2 Results

- AUC = 0.671
- Accuracy = 62.4%
- Precision (Active Trades) = 76.2%

Conclusion : DL achieves respectable performance but doesn't beat the Voting Classifier (0.698). For ∼20k tabular samples, **tree ensembles remain superior** due to :

- Better sample efficiency (less data needed)

- Natural handling of feature interactions without manual engineering
- Robustness to scaling issues (trees are invariant to monotonic transformations)

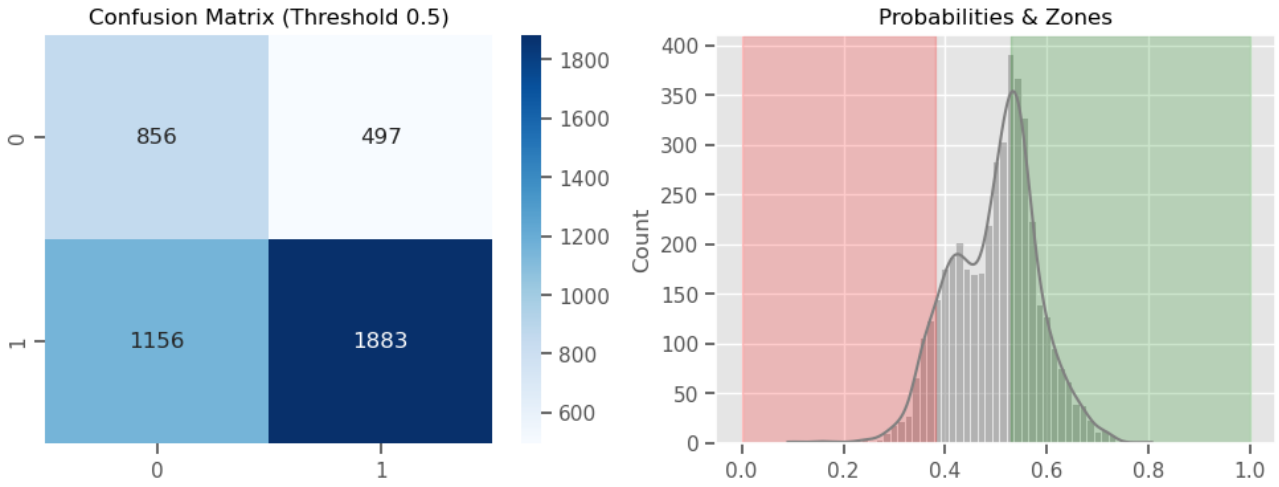

Figure 7: Deep Learning Results : NN more hesitant (narrow probability distribution around 0.5) compared to trees, which produce more confident extremes.

12 Conclusion

12.1 Summary

This project successfully developed a machine learning system to predict stock market direction from financial statements :

- **Best Model** : Voting Classifier (AUC 0.698, Precision 83% on high-confidence signals)
- **Discovered Strategy** : Value Investing (low PB ratio, high margins, low debt)
- **Robustness** : Validated across multiple years and CV strategies, handles market regime shifts

12.2 How We Tackled the Business Case

Table 8: Challenge-Solution Summary

Challenge	Solution
Regime shift (51→69%)	Class weighting + threshold-independent AUC metric
High dimensionality (215 feat.)	Tree models with embedded feature selection + 40% NaN filter
Temporal dependencies	TimeSeriesSplit for hyperparameter tuning
Weak linear signals	Non-linear ensembles (RF, GradBoost)
Overfitting risk	Regularization, cross-validation, early stopping, ensemble averaging

12.3 Business Impact

Proposed Trading Strategy : 3-tier portfolio system :

1. **Aggressive** (Prob $>$ 0.60) : High-conviction buys, 15-20 stocks
2. **Core** (Prob 0.53-0.60) : Moderate buys, 30-40 stocks
3. **Defensive** (Prob $<$ 0.38) : Short candidates or avoidance list

Expected Performance : 83% hit rate on Tier 1 trades, 38% fewer losing trades vs random selection.

12.4 Limitations & Future Work

Current Limitations :

- Binary target (direction only, not magnitude)
- Annual frequency (intra-year volatility ignored)
- Survivorship bias (bankrupt/delisted companies excluded)
- No transaction costs modeled

Proposed Extensions :

- Multi-class classification (Strong Buy, Buy, Hold, Sell, Strong Sell)
- Regression on top decile to predict return magnitude
- Alternative data integration (sentiment analysis, macroeconomic indicators)
- Online learning for incremental updates with new quarterly reports
- Portfolio optimization using predicted probabilities as inputs to Markowitz mean-variance framework

Key Takeaway

Machine Learning + Financial Expertise = Powerful Synergy
Success came not from blindly applying algorithms, but from :

- Careful problem formulation (binary classification vs regression)
- Domain-aware preprocessing (winsorization, temporal splits, anti-leakage)
- Rigorous validation (TimeSeriesSplit, multiple metrics, cross-validation)
- Interpretability analysis (feature importance, calibration, decision zones)

The model's rediscovery of Value Investing principles validates that data-driven methods can converge on time-tested financial wisdom.

13 References

13.1 Scientific Papers

1. Breiman, L. (2001). "Random Forests". *Machine Learning*, 45(1), 5-32.
2. Chen, T., & Guestrin, C. (2016). "XGBoost : A Scalable Tree Boosting System". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.
3. Fama, E.F., & French, K.R. (1992). "The Cross-Section of Expected Stock Returns". *Journal of Finance*, 47(2), 427-465.
4. Gu, S., Kelly, B., & Xiu, D. (2020). "Empirical Asset Pricing via Machine Learning". *Review of Financial Studies*, 33(5), 2223-2273.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer.

13.2 Technical Documentation

- Scikit-learn : `https://scikit-learn.org/stable/`
- TensorFlow/Keras : `https://www.tensorflow.org/`
- Pandas : `https://pandas.pydata.org/`
- Kaggle Dataset : `https://www.kaggle.com/datasets/cnic92/200-financial-indicators-of-us-stock-data`