Comparative Analysis of Regression vs. Classification in Financial Market Forecasting

Fan Yang

Motivation & Problem Statement

Why Forecast the Market?

- Daily prices are noisy and hard to predict
- Small signals may still be exploitable
- Better forecasts → better trading performance

Two Ways to Predict the Market

- Regression**:** predict return size (e.g., +0.3%)
- Classification**:** predict direction (Up / Down)

Key Question

Which approach gives more reliable trading signals?

Data

- Data source: daily OHLCV from Stooq
- Univserse: 20 large US stocks (e.g., AAPL, MSFT, AMZN, META, GOOGL, etc.)
- Example:

Date	Open	High	Low	Close	Volume
1984-12-03	0.0926865	0.09329722	0.0914741	0.0914742	116303790
1984-12-04	0.0932972	0.09537683	0.0932972	0.0932972	142860926
1984-12-05	0.0977914	0.0977914	0.0977914	0.0977914	312013193

- Basic cleaning:
  - Sort by date, drop missing values, align features and targets

Feature Engineering

- **Categories of Features**
- **Momentum**
  - Recent price moves (e.g., 1-day & 5-day returns)
- **Volatility**
  - Risk and price fluctuation (rolling std, high/low range)
- **Trend Indicators**
  - Moving averages and spread (10-day vs 50-day)
- **Drawdown**
  - Selling pressure and fear(maximum drop over the last 60 days)
- **Seasonality**
  - Day-of-week and month effects
- **Key Design Choice**
- All features rely **only on past data** → no look-ahead bias

Models

- **Goal of Modeling:** Use **two parallel families of models** to answer the key question:
  Does regression or classification lead to better trading performance?

Regression	Classification
Linear Regression	Logistic Regression
Ridge Regression	Random Forest Classifier
Random Forest Regressor	Gradient Boosting Classifier
Gradient Boosting Regressor	SVM Classifier

- Controlled variables: same features, same rolling windows, same trading rules, same transaction costs.

Rolling Walk-Forward Backtest

- **How it works**

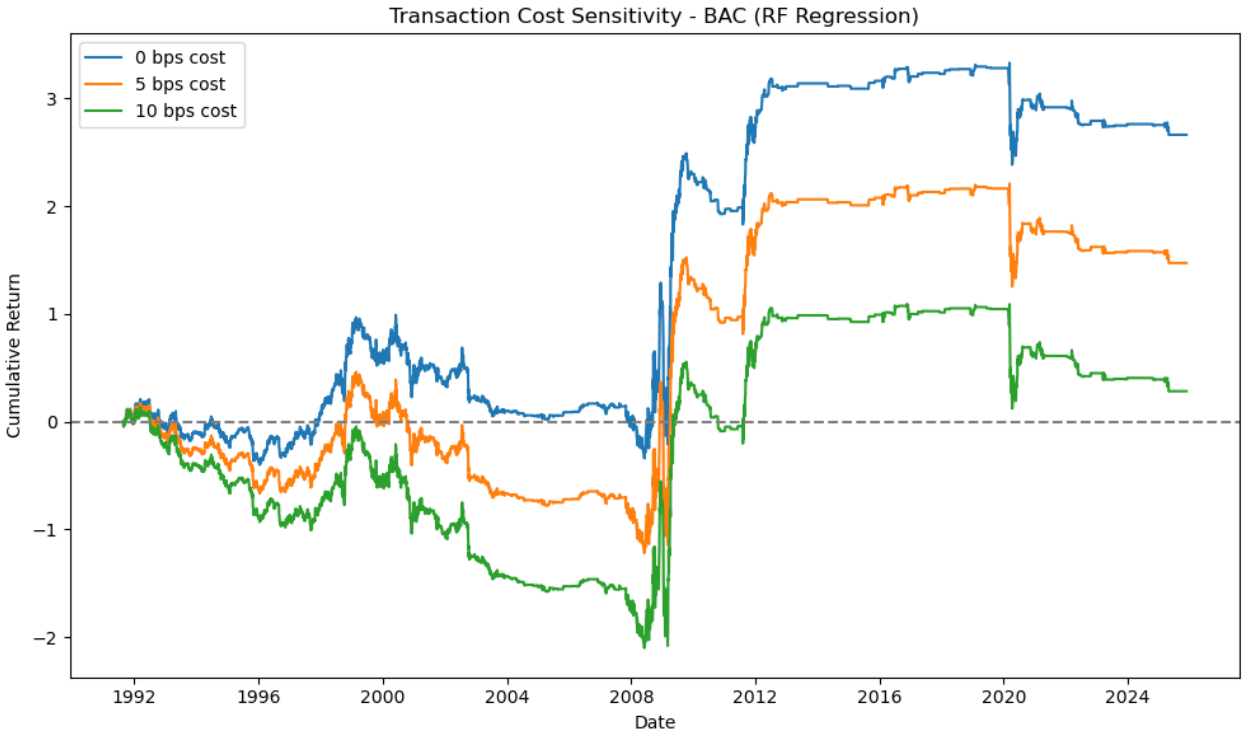
1. Train on the first **5** years of data
2. Test on the next **1** year
3. Roll forward by **1** year
4. Re-train and repeat

- **Why we do this:** Models are **never** trained on future data and gives a realistic measure of how performance would evolve over time

Trading Strategy

- Regression:
  - If predicted > +0.0005 , then go long
  - If predicted < -0.0005, then go short
  - Else, stay flat
- Classification:
  - If p(up) > 0.5+0.05, then go long
  - If p(up) < 0.5- 0.05, then go short

  Else, stay flat
- Transaction Costs Included:
  - Every time position changes -> cost deducted
  - small/uncertain signals become unprofitable
  - cost_per_side = 0.0005



difference between 0, 5, 10 bps cost on Ticker MSFT using RF regression

Evaluation Methods

Metric	What it measures	Why it matters
Sharpe Ratio	Return / Volatility	Higher = more return per unit of risk
Max Drawdown	Worst peak-to-trough drop	Captures downside risk & stress
Total Return	Overall profit over the full period	Raw profitability, not adjusted

- Comparison:
  - For each stock pick:
  - Best regression model(highest Sharpe)
  - Best classification model ((highest Sharpe)

  Then compare per-stock winners

  Statistical Test to test p-values

Best Hyperparameters

Model	Key Hyperparameters Selected
Linear Regression	*(no tunable hyperparameters)*
Ridge Regression	α = **100.0**, fit_intercept = **False**, solver = **svd**
Random Forest Regressor	n_estimators = **100**, max_depth = **3**, max_features = **'log2'**, min_samples_leaf = **20**, bootstrap = **False**
Gradient Boosting Regressor	n_estimators = **100**, learning_rate = **0.01**, max_depth = **2**, subsample = **0.8**, loss = **squared_error**
Logistic Regression	C = **0.1**, penalty = **l2**, class_weight = **balanced**, fit_intercept = **False**
Random Forest Classifier	n_estimators = **100**, max_depth = **3**, max_features = **None**, min_samples_leaf = **5**, bootstrap = **False**
Gradient Boosting Classifier	n_estimators = **100**, learning_rate = **0.01**, max_depth = **3**, subsample = **1.0**, loss = **log_loss**
SVM Classifier	C = **0.1**, gamma = **scale**, kernel = **poly**, class_weight = **balanced**

Result

- 10 stocks:

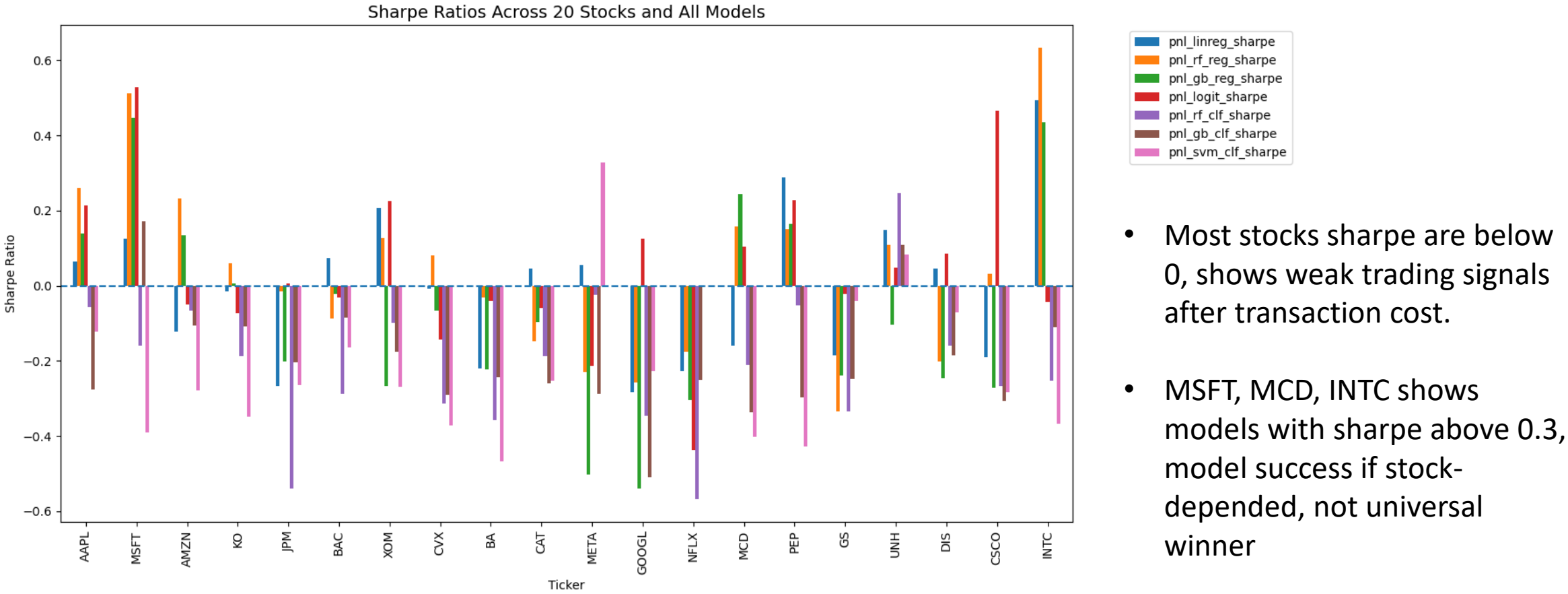
stat = 9.0 , p-value = **0.064**
classification appears slightly better
but results are **not statistically significant** at the 5% level.

- 20 stocks:

stat = 87.0, p-value = **0.522**

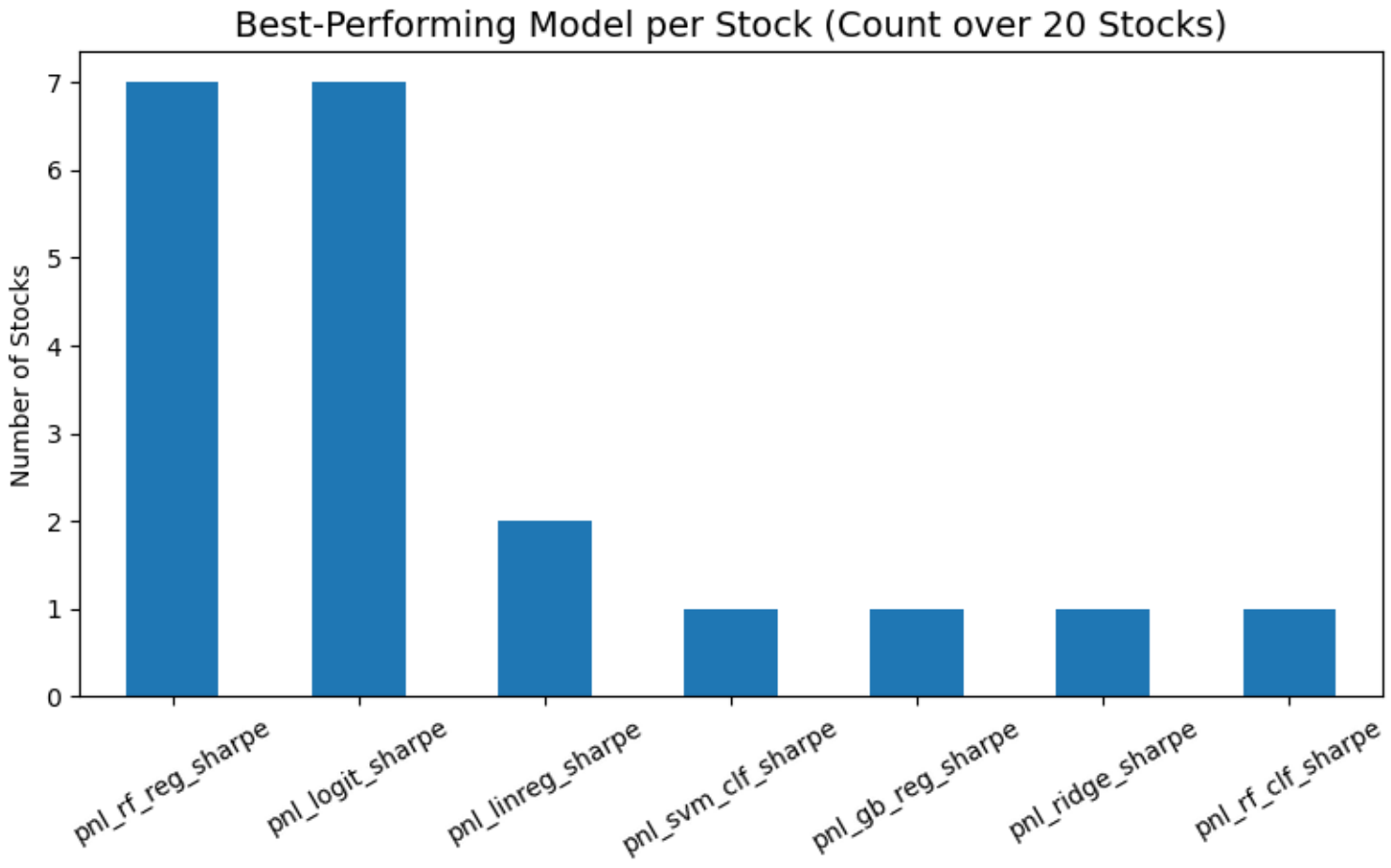
no statistical difference, regression and classification deliver statistically similar performance

Sharpe ratio across 20 stocks and all models



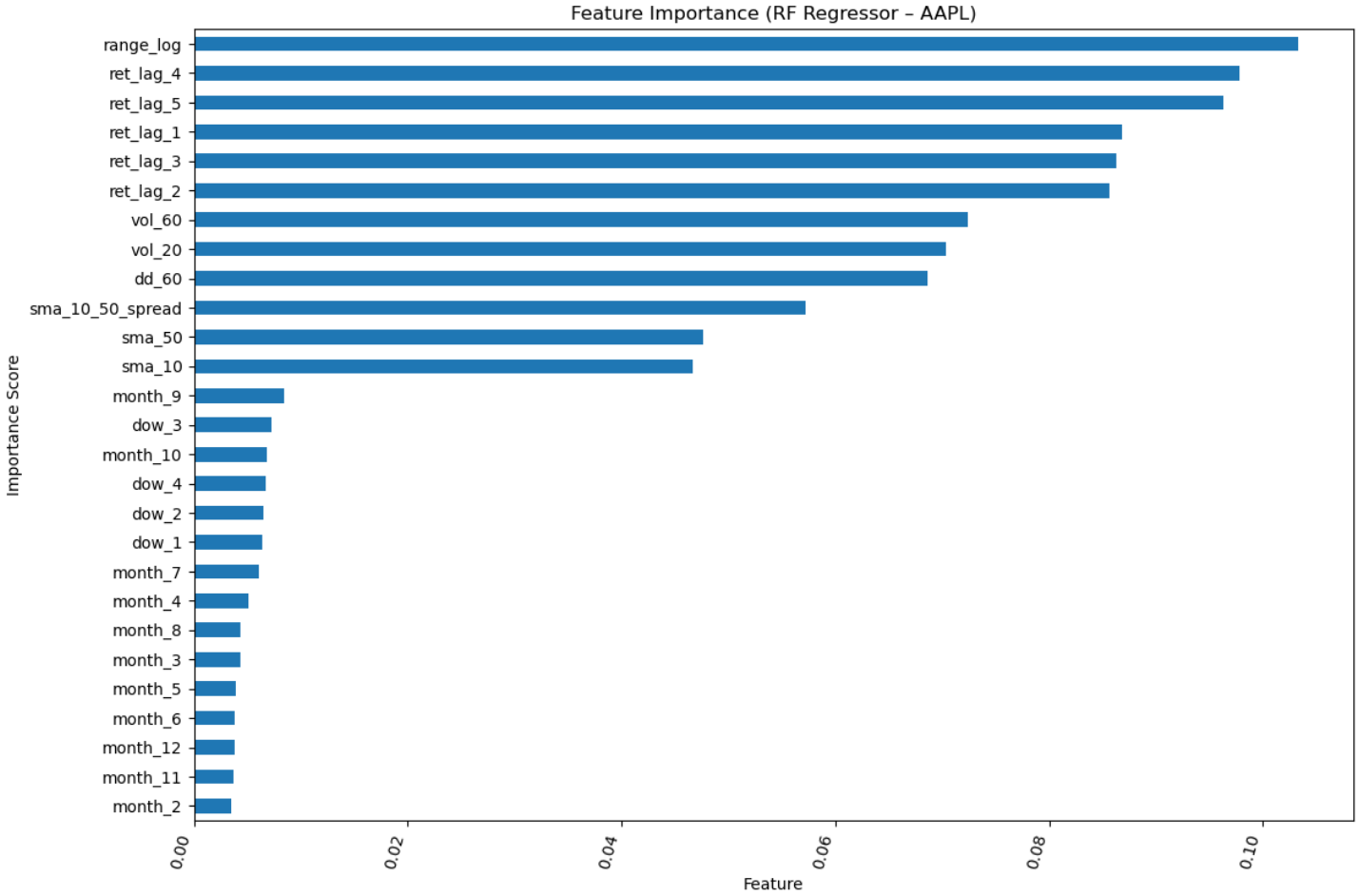
- Most stocks sharpe are below 0, shows weak trading signals after transaction cost.
- MSFT, MCD, INTC shows models with sharpe above 0.3, model success if stock-depended, not universal winner

Best performing model per stock



Two models dominate the wins
Other models are less consistent

Feature importance – APPL(RF)



- What drives the predictions?
  - Short-term return lags(ret_lag 1-5)
  - Intraday volatility(range_log)
  - Medium-term volatility (vol_20, vol_60, dd_60)
- What matters less?
  - Moving averages(minor contribution)
  - Calendar feature(days of week, month of year)

Conclusion

- **Main Findings**
- On a small sample (10 stocks), classification showed a **weak edge**, but not statistically significant
- With more assets (20 stocks), **no meaningful performance difference** between regression and classification
- Individual model winners are **stock-dependent**
  → Random Forest Regressor and Logistic Regression dominate equally

Future Work & Improvements

Expand Data Coverage

- More stocks, longer history, intraday features

Enhance Signal Quality

- Add sentiment analysis from news, earning releases and social media
- Explore **ensemble strategies** combining regression & classification outputs

Thanks

- Any questions?