Project Specification: Machine Learning for Stock Predictions

Your Quant Club Name

February 14, 2025

Objective

To design and implement a machine learning model to predict stock price movements or trends based on historical price data and engineered features such as technical indicators and external market factors.

Project Goals

1. Collect and preprocess stock market data.
2. Engineer meaningful features from the data.
3. Train and evaluate machine learning models for prediction.
4. Backtest predictions to assess profitability and performance.
5. Analyze and refine the strategy based on results.

Scope

- Focus on predicting **short-term movements** (e.g., next day or next week).
- Predictions will classify stock price movements as **up, down, or neutral**.
- Use **supervised learning techniques**, comparing multiple models for performance.

Deliverables

1. **Data Pipeline:**
    - Script to fetch historical stock data (e.g., Yahoo Finance API).
    - Feature engineering pipeline for technical indicators and market signals.
2. **Machine Learning Model:**

- ML models for classification (e.g., Logistic Regression, Random Forest, XGBoost, Neural Networks).
- Train/test split or cross-validation for model evaluation.

3. **Evaluation Metrics:**
   - Accuracy, precision, recall, F1-score, and Sharpe ratio for performance evaluation.
4. **Backtesting Framework:**
   - Simulate trading based on predictions to calculate profits/losses.
5. **Report:**
   - Document the data preprocessing, model results, and backtesting performance.

Steps

1. **Data Collection:**
   - Use Yahoo Finance (`yfinance`) or another source to download daily stock data.
   - Collect data for multiple stocks over at least 5 years for training/testing.
2. **Feature Engineering:**
   - Technical indicators:
     - Simple Moving Average (SMA)
     - Exponential Moving Average (EMA)
     - Relative Strength Index (RSI)
     - Bollinger Bands
     - Moving Average Convergence Divergence (MACD)
   - Additional features:
     - Momentum
     - Volatility
     - Volume changes
     - Lagged returns (past price movements)
   - Optional:
     - Sentiment data (e.g., financial news or social media sentiment).
3. **Model Selection:**
   - Train multiple models to compare performance:
     - Logistic Regression
     - Random Forest
     - Gradient Boosted Trees (e.g., XGBoost, LightGBM)

    - Neural Networks (simple feedforward or LSTMs for sequential data)
  - Experiment with hyperparameter tuning for each model.

4. **Model Evaluation:**
   - Evaluate models using:
     - Classification metrics (accuracy, precision, recall, F1-score).
     - Sharpe ratio for profitability analysis during backtesting.

5. **Backtesting Framework:**
   - Simulate a trading strategy using the predicted signals (e.g., buy when predicting "up," sell when predicting "down").
   - Evaluate performance in terms of:
     - Total returns
     - Drawdowns
     - Risk-adjusted returns (Sharpe and Sortino ratios).

6. **Deployment and Automation:**
   - Build a script to automate predictions for new data.
   - Optional: Integrate into a dashboard for real-time predictions.

Timeline

Week	Milestone
1	Define the scope and collect historical data.
2	Engineer features and preprocess the data.
3	Train initial machine learning models.
4	Evaluate models and optimize performance.
5	Implement backtesting framework.
6	Refine and finalize the strategy.
7	Create a report and present results.

Resources Needed

- **Python Libraries:** `pandas`, `numpy`, `scikit-learn`, `xgboost`, `tensorflow` or `pytorch`, `matplotlib`, `yfinance`.
- **Data:** Historical stock prices, volume data, optional sentiment/news data.
- **Team Skills:** Data preprocessing, feature engineering, ML model training and evaluation, Python coding, financial knowledge.

Evaluation Criteria

- **Model Accuracy:** Can the model correctly classify price movements?
- **Profitability:** Does the trading strategy based on predictions yield positive returns?
- **Scalability:** Can the model generalize across multiple stocks?
- **Interpretability:** Are the predictions explainable in financial terms?

Extensions (Optional)

- Experiment with **deep learning models** (e.g., LSTM for time series analysis).
- Incorporate **alternative data** like sentiment analysis or macroeconomic indicators.
- Deploy the model to make **real-time predictions**.