Project Proposal
Title: Stock Price Movement Prediction Using NYSE Dataset
Course: CSCE 5300
Group Members:

SATYA TEJ KAMMILI (satyatejkammili@my.unt.edu) 11911161
MERLA GANESH REDDY (merlaganeshreddy@my.unt.edu) 11797294
RUTHVIKA MUCHALA (ruthvikamuchala@my.unt.edu) 11906895
KOUSHIK REDDY SUDIREDDY (koushikreddysudireddy@my.unt.edu) 11885863

Index

1. Introduction
2. Problem Statement
3. Proposed Methods and Dataset
4. Models
5. Evaluation Approach
6. Anticipated Challenges
7. Work-Split
8. Planned Deliverables

Introduction

Stock market behavior is shaped by many factors, including a company's financial performance, investor sentiment, and overall economic conditions. Predicting short-term price changes is difficult because markets are highly volatile and contain a lot of noise. This project aims to forecast whether a stock's price will rise or fall on the next trading day using historical market data. By applying machine learning and time-series analysis, we seek to identify patterns that may inform trading decisions, and we will evaluate the models using both predictive accuracy and relevant financial performance metrics.

Problem Statement

Stock market movements are shaped by many different elements, such as a company's financial health, investor behavior, and broader economic trends. Because of these influences, short-term price prediction is particularly difficult the market tends to be unpredictable and doesn't always follow a linear pattern.

In this project, our goal is to forecast whether a stock's price will rise or fall on the following trading day. To do this, we will use historical records from the New York Stock Exchange (NYSE), which provide daily trading details for a variety of companies.

By building machine learning models on this dataset, we want to see if it's possible to identify patterns that can guide investment decisions and perform better than simple approaches, like guessing randomly or sticking to a basic buy-and-hold strategy.

1. Methods and Dataset

Dataset:

We use the NYSE dataset from the files you uploaded (Dataset.zip). I inspected the data and computed the exact sizes:

- Main price file used: prices-split-adjusted.csv (also included prices.csv, fundamentals.csv, securities.csv).
- Price table (after labelling for next-day movement): 850,763 labeled rows (rows with a computable next-day close), covering 501 distinct tickers.
- Date range in the price data: 2010-01-04 through 2016-12-29.

Additional tables:

- Fundamentals.csv: 1,781 rows × 79 columns (company fundamentals that can be merged on ticker/date for additional features).
- securities.csv: 505 rows (static security metadata such as sector, HQ).

Target / labeling:

- Binary target target_up = 1 if next day's close > today's close, else 0 (ties included in 0).
- After labeling: 438,931 up (51.59%) and 411,832 down (48.41%) — overall fairly balanced.
- Per-symbol balance: mean fraction of "up" ≈ 0.5158 (std ≈ 0.0145, min ≈ 0.4668, max ≈ 0.556). Only 3 of 501 tickers have strong imbalance (>0.55 or <0.45).

2. Method

Task definition

- This is a binary classification task: predict whether the stock closes higher the next trading day (Up=1) or not (Down/Flat=0).

Feature plan

- Price-driven: previous-day returns, lagged returns (1,2,3,5 days), rolling means (5/10/20-day), rolling std (volatility).
- Volume: day-over-day volume change, rolling average volume.
- Technical indicators: RSI, MACD, Bollinger Bands, momentum.
- Calendar: day-of-week, month, quarter, earnings-day flag (where available).
- Firm fundamentals & sector metadata merged from fundamentals.csv and securities.csv when applicable.

Models

- Baselines: naive (previous-day sign), moving-average crossover.
- Classical ML: Logistic Regression (with class weights), Random Forest, XGBoost.
- Optional extension: LSTM (sequence models) — only if enough contiguous time-series per ticker and after careful overfitting controls.

Evaluation & metrics

- Primary classification metric: F1-score. Report macro-F1 (treats classes equally) and weighted-F1 (weights by class support).
- Also report: accuracy, precision, recall, ROC-AUC, and confusion matrices.
- Primary practical metrics: cumulative returns, Sharpe ratio, and maximum drawdown computed from a realistic backtest (account for transaction costs & slippage).
- Why F1? F1 balances precision and recall — appropriate when we care about having reasonably few false signals and capturing true movements, and it's more meaningful than raw accuracy for trading signals.

Validation protocol

1. Walk-forward (rolling) validation with an expanding or sliding training window to avoid lookahead bias.

- Example: train on 2010–2012 → validate on 2013; roll forward in 6-month or 1-year steps.
- Hyperparameter tuning performed inside each training window using nested time-series cross validation (no leakage).

2. Evaluate model both per-ticker and on a portfolio-level aggregated backtest (construct a simple long-short or directional portfolio from predictions to compute economic metrics).

- Baseline to compare: previous-day sign and a moving-average strategy. Models must beat these baselines on both F1 and financial metrics to be considered useful.

Anticipated Challenges

Non-stationarity / regime shifts

- Mitigation: rolling retraining (retrain periodically), use shorter training windows and/or exponential weighting of recent data; monitor model degradation and trigger re-training when performance drops. Consider regime detection (volatility clusters) and condition models on regime.

High noise / weak signals

- Mitigation: robust feature engineering (smoothing, aggregating multiple indicators), use ensembles to average noisy predictions, and focus on risk-adjusted returns rather than raw accuracy. Use calibration and probability thresholds to generate trading signals only when the model is confident.

Data leakage / lookahead bias

- Mitigation: build a disciplined pipeline that computes features using only past data in each fold, perform all normalization/scaling inside the training fold (apply parameters to validation/test), and never use forward-looking columns (e.g., next day volume).

Imbalance or uneven class distribution for some tickers

- Mitigation: although overall the dataset is balanced (~51.6% up), some tickers are a bit imbalanced — use class weights in models, consider stratified sampling per time-window, or apply resampling (SMOTE) restricted to the training fold only. Report per-ticker F1 to check worst-case behavior.

Overfitting (especially with deep models)

- Mitigation: prefer simpler models first; use regularization (L1/L2), early-stopping for boosting/NNs, pruning for trees, and nested CV to tune hyperparameters. Validate on an out-of-time holdout (final test period) to measure generalization.

Survivorship bias / dataset coverage

- Mitigation: carefully use securities.csv “date first added” fields to avoid backtesting companies before they joined the index; report coverage and remove tickers with insufficient history.

Practical trading issues: transaction costs, slippage, and execution

- Mitigation: include realistic transaction cost assumptions in backtest and measure net returns; use conservative position sizing and apply a minimum confidence filter to avoid over-trading.

Computational / reproducibility

- Mitigation: code in Colab/Jupyter with pinned package versions; provide a requirements.txt and a reproducible seed; use vectorized pandas/numpy and limit model ensembles size for speed.

Work-Split

SATYATEJ KAMMILI - Collect NYSE dataset, ML Methods (Logistic, RF, XGBoost) and BackTesting
MERLA GANESH REDDY - Compute financial metrics and Presentation slides
RUTHVIKA MUCHALA - Evaluation and Modeling with baseline
KOUSHIK REDDY SUDIREDDY- Report writing and GitHub

Planned Deliverables

1. Final Report (PDF) - A complete document covering the methodology, results, analysis, and discussion of findings.
2. Colab / Jupyter Notebook - Full code implementation with a reproducible workflow so that others can replicate the analysis.
3. GitHub Repository - All scripts, references to datasets, and accompanying documentation for easy access and version control.
4. Presentation Slides - A concise summary of the project, highlighting methodology, results, and key conclusions.