Systematic Equity Research on AAPL: A Statistical and NLP-Enhanced Approach

Saxon Lee

12th of September 2023

Introduction and Prerequisite Work

This document outlines a proposed methodology for developing and evaluating machine learning models to predict short-to-medium term returns for Apple Inc. (AAPL) stock. This research builds upon prerequisite exploratory analysis performed on historical AAPL OHLCV data (1980-2022), which involved unsupervised learning techniques to identify market regimes and detect anomalies.

The prior work successfully applied:

- **Feature Engineering:** Calculation of log returns, rolling volatility, rolling momentum, and relative volume metrics using 60-day and 21-day windows.
- **Market Regime Detection:** K-Means clustering (exploring K=3, 4, 5, 6) based on volatility and momentum features to segment historical data into distinct market states.
- **Anomaly Detection:** Identification of statistically unusual trading days using Isolation Forest, Local Outlier Factor (LOF), and Z-Score methods on the engineered features.

Visualizations from this phase (regime plots, anomaly plots) provide context on the stock's historical behavior. The current proposal shifts focus to a supervised learning paradigm, aiming to leverage historical patterns for predictive forecasting.

1 Rigorous Problem Formulation

Objective: Develop and rigorously evaluate statistical and machine learning models capable of predicting future price movements, specifically N-day ahead returns, of AAPL stock, utilizing solely historical OHLCV data as input features.

Task Definition: Supervised Learning - Time Series Regression. The objective is to learn a mapping function $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ represents the high-dimensional feature space derived from historical data, and $\mathcal{Y}$ represents the target space of future returns.

Target Variable (y_t): Given the well-documented non-stationarity (presence of unit roots, stochastic trends, and time-varying variance) of financial price series P_t, direct price prediction is statistically challenging and often leads to spurious results. A more robust approach is to predict N-day ahead logarithmic returns, defined at time t as:

$$y_t = \log\left(\frac{P_{t+N}}{P_t}\right) \tag{1}$$

where P_t is the Adjusted Close price at the close of trading day t. Log returns approximate percentage changes for small values, exhibit more desirable statistical properties (closer to stationarity, though potentially exhibiting volatility clustering), and are additive over time. This formulation aligns with standard practices in quantitative finance and time series analysis *(Ref: Concepts of stationarity, differencing, and transformations in time series analysis, e.g., Box-Jenkins methodology; PMML Intro/Advanced).*

Prediction Horizon (N): The forecast lookahead period N will be explored for short-term horizons (e.g., $N = 1, N = 5$ trading days) where market inefficiencies or autocorrelations might yield predictive signals, and potentially medium-term horizons (e.g., $N = 21$ trading days, approx. 1 month), where predictability is expected to diminish significantly according to market efficiency theories.

Input Features (X_t): Define $X_t \in \mathbb{R}^d$ as the feature vector summarizing relevant information available strictly at or before time t. The dimensionality d will depend on the feature engineering choices. The model f aims to approximate the conditional expectation $E[y_t|X_t]$.

2 Comprehensive Feature Engineering & Selection

The construction of the feature set X_t is critical and aims to capture diverse aspects of past market dynamics. All features must be computable using information available only up to time t.

- **Lagged Variables:**
  - *Lagged Returns:* Include recent past log returns: $r_{t-1}, r_{t-2}, \ldots, r_{t-L}$, where $r_t = \log(P_t/P_{t-1})$. The maximum lag L (e.g., 5, 10, 21) is a hyperparameter influencing the model's memory.
  - *Lagged Volatility:* Incorporate estimates of historical volatility, such as the rolling standard deviation of log returns over various windows $w \in \{5, 21, 60\}$: $\sigma_{t,w} = \sqrt{\frac{1}{w-1}\sum_{k=0}^{w-1}(r_{t-k} - \bar{r}_{t,w})^2}$.
  - *Other Lagged OHLCV Features:* Consider lagged raw prices (potentially detrended or normalized, e.g., price relative to a moving average $P_t/\text{SMA}_k(P_t)$), lagged volume, and intraday range measures (e.g., $(H_t - L_t)/C_t$).
- **Technical Indicators:** Compute standard indicators, ensuring calculations use data only up to time t.
  - *Momentum Oscillators:* Relative Strength Index (RSI(k)), Stochastic Oscillator (%K(k), %D(j)), Williams %R(k). Parameter k (e.g., 14) defines the lookback period.
  - *Trend Indicators:* Simple Moving Average (SMA(k)), Exponential Moving Average (EMA(k)) for various k; Moving Average Convergence Divergence (MACD(12, 26, 9)) signal line and histogram; Average Directional Index (ADX(14)).
  - *Volatility Channels/Indicators:* Bollinger Band Width ($\frac{\text{UpperBand}-\text{LowerBand}}{\text{MiddleBand}}$ for SMA(k=20, std=2)), Average True Range (ATR(k=14)).
- **Time-Based Features:** Encode potential calendar effects: Day of week, day of month, month of year, potentially quarter. Use appropriate encoding schemes like one-hot encoding or cyclical feature embedding (e.g., $\sin(2\pi \cdot \text{day}/7), \cos(2\pi \cdot \text{day}/7)$ for day of week).
- **(Optional) Unsupervised Features:** Integrate outputs from the prerequisite analysis:

  - Regime Labels: Categorical feature representing the market regime identified by K-Means (or GMM/HMM if implemented) at time t or $t-1$.
  - Anomaly Flags/Scores: Binary flag or continuous score from Isolation Forest/LOF indicating if time t or $t-1$ was anomalous. This tests the hypothesis that regime/anomaly status provides additional predictive information beyond standard technical features.

- **Feature Selection/Importance Analysis:** Given the potential for a high-dimensional feature space, apply techniques to mitigate multicollinearity, reduce noise, and identify the most salient predictors:
  - *Filter Methods:* Assess relevance using statistical measures like correlation (Pearson, Spearman), Mutual Information score between each feature and the target y_t.
  - *Wrapper Methods:* Employ techniques like Recursive Feature Elimination (RFE) using a base estimator (e.g., Linear Regression, Random Forest).
  - *Embedded Methods:* Utilize models with built-in feature selection, such as Lasso (L1 regularization) or leverage feature importance scores derived from tree-based ensembles (e.g., Gini importance from Random Forest, permutation importance). *(Ref: Feature selection methodologies; PMML Intro, UDL).*

3 Data Splitting & Preprocessing Strategy

Methodological rigor in data handling is crucial to prevent lookahead bias and obtain reliable estimates of generalization performance on unseen future data.

- **Temporal Splitting:** Data must be split chronologically. A fixed-origin approach will be used initially:
  - *Training Set:* Data from start date to T_{train} (e.g., 1980-12-12 to 2015-12-31). Used for model parameter estimation.
  - *Validation Set:* Data from $T_{\text{train}} + 1$ day to T_{val} (e.g., 2016-01-01 to 2018-12-31). Used exclusively for hyperparameter tuning and model selection.
  - *Test Set:* Data from $T_{\text{val}} + 1$ day to end date (e.g., 2019-01-01 to 2022-06-17). Used *only once* for final, unbiased performance evaluation of the selected model.

- **Walk-Forward Validation / Time Series Cross-Validation (Potential Extension):** For more robust hyperparameter tuning and assessment under potential concept drift, consider implementing walk-forward validation. This involves iteratively training on an expanding or rolling window and validating on the subsequent block of data. *(Ref: Model evaluation strategies for time series; PMML Intro, UDL).*

- **Feature Scaling:** Apply standardization (e.g., via `sklearn.preprocessing.StandardScaler` for zero mean, unit variance) or normalization (e.g., `MinMaxScaler`). **Crucially, the scaler must be fitted *only* on the training set data**. The same fitted scaler instance is then used to `transform` the validation and test sets. This prevents leakage of information (e.g., mean, variance) from future data distributions into the training process.

- **Input Shaping for Sequence Models:** For models like LSTMs, GRUs, or Transformers, input features X_t need to be reshaped into sequences. Create overlapping windows of length k (sequence length, a hyperparameter): The input for predicting y_t would be the sequence $[X_{t-k+1}, \ldots, X_t]$. The output dimension corresponds to the prediction horizon N. The data shape becomes `[n_samples, sequence_length, n_features]`.

4 Model Selection, Training & Hyperparameter Tuning

Explore a hierarchy of models, from simple baselines to sophisticated deep learning architectures, to understand the complexity required to capture any predictive signal.

- **Baseline Models:** Essential for establishing minimum performance benchmarks.
  - *Historical Mean:* Predict $\hat{y}_t = \bar{y}_{\text{train}}$, the mean return observed in the training set.
  - *Naive / Persistence Forecast:* Predict $\hat{y}_t = r_t = \log(P_t/P_{t-1})$. Assumes the next N-day return equals the most recent 1-day return.
- **Linear Models:** Provide interpretability and robust baselines.
  - Linear Regression (OLS)
  - Ridge Regression (L2 Regularization): Controls model complexity, suitable when multicollinearity is present.
  - Lasso Regression (L1 Regularization): Performs implicit feature selection by shrinking some coefficients to zero.
- **Non-Linear Machine Learning Models:** Capture potential non-linear relationships in features.
  - Support Vector Regression (SVR): With various kernels (linear, polynomial, Radial Basis Function - RBF). Requires careful tuning of C (regularization) and kernel parameters (e.g., γ for RBF).
  - Tree-Based Ensembles: Random Forest Regressor, Gradient Boosting Machines (specifically XGBoost, LightGBM, CatBoost libraries known for performance and handling of features). Tune parameters like number of trees, tree depth, learning rate, subsampling rates.
- **Deep Learning Models:** Explore architectures designed for sequential data. *(Ref: UDL, Foundations of Deep Learning, PMML Advanced).*
  - *Recurrent Neural Networks (RNNs):* Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks are standard choices for modeling temporal dependencies. Architectures involve stacked layers, potentially bidirectional processing. Requires careful tuning of hidden units, activation functions, dropout rates (for regularization), optimizers (Adam, RMSprop), and learning rates.
  - *1D Convolutional Neural Networks (CNNs):* Can act as feature extractors on sequences, identifying local patterns (e.g., patterns over a few days). Often combined with RNNs (CNN-LSTM architecture) where CNNs process subsequences fed into an LSTM. Tune filter sizes, number of filters, pooling strategies.
  - *Transformers:* Attention-based architectures. While state-of-the-art in NLP, their application to financial time series requires careful consideration. May demand large datasets and significant tuning (embedding dimensions, number of attention heads, layers, feed-forward network size). Potential for capturing complex, long-range dependencies but risk of overfitting noisy financial data.
- **Hyperparameter Tuning:** Systematically optimize model hyperparameters using performance on the *validation set*. Avoid using the test set for tuning. Employ standard techniques:
  - Grid Search: Exhaustive search over a predefined grid of parameters.

- Randomized Search: Samples randomly from parameter distributions, often more efficient for high-dimensional search spaces.
- Bayesian Optimization: Uses probabilistic models to select promising parameter combinations, often converging faster.

Focus optimization on parameters controlling model complexity (e.g., tree depth, number of layers/units), regularization strength (e.g., α in Ridge/Lasso, dropout rate), and optimization algorithm parameters (e.g., learning rate). *(Ref: Model selection and hyperparameter optimization chapters; PMML Intro, UDL).*

5 Rigorous Evaluation

Evaluate the final selected model (chosen based on validation set performance) on the held-out *test set*. Performance metrics should cover both statistical accuracy and potential practical relevance.

- **Standard Regression Metrics:**
  - Mean Squared Error (MSE): $\frac{1}{M}\sum_{t\in\text{Test}}(y_t - \hat{y}_t)^2$. Sensitive to large errors.
  - Root Mean Squared Error (RMSE): $\sqrt{\text{MSE}}$. Interpretable in the units of log return.
  - Mean Absolute Error (MAE): $\frac{1}{M}\sum_{t\in\text{Test}}|y_t - \hat{y}_t|$. Less sensitive to outliers than MSE.
  - R-squared (R^2): $1 - \frac{\sum_{t\in\text{Test}}(y_t-\hat{y}_t)^2}{\sum_{t\in\text{Test}}(y_t-\bar{y}_{\text{Test}})^2}$. Proportion of variance explained relative to a simple mean prediction. Use with caution in time series due to potential non-stationarity and spurious trends. Out-of-sample R^2 can be negative.
- **Directional Accuracy (Hit Rate):** $\frac{1}{M}\sum_{t\in\text{Test}}\mathbb{I}(\text{sign}(y_t) = \text{sign}(\hat{y}_t))$, where $\mathbb{I}(\cdot)$ is the indicator function (1 if true, 0 otherwise). Measures the percentage of times the model correctly predicted the direction (up/down) of the return. Compare against a 50% benchmark.
- **Statistical Significance Testing:** Assess if the observed performance difference between the developed model and a baseline (e.g., naive forecast) is statistically significant. The Diebold-Mariano test is appropriate for comparing forecast accuracy, accounting for serial correlation in forecast errors.
- **Financial Backtesting (Simulated, Interpret with Extreme Caution):**
  - Define a simple, non-optimized trading rule based on predictions (e.g., long if $\hat{y}_t > \tau$, short if $\hat{y}_t < -\tau$, flat otherwise, where τ is a small threshold).
  - Simulate the execution of this strategy over the test period, incorporating realistic estimates for transaction costs (commission per trade + slippage estimate per trade).
  - Calculate standard portfolio performance metrics: Cumulative Return, Annualized Return (Geometric), Annualized Volatility (Standard Deviation of returns), Sharpe Ratio (risk-adjusted return: (AvgReturn − RiskFreeRate)/StdDevReturn), Maximum Drawdown (peak-to-trough decline).
  - **Disclaimer:** Backtesting is highly susceptible to overfitting (data snooping bias) and unrealistic assumptions. Results must be viewed as indicative only and not guarantees of future performance.

6 Addressing Potential Challenges

Acknowledge and proactively consider mitigation strategies for inherent difficulties in financial market forecasting.

- **Non-Stationarity & Concept Drift:** Financial markets are dynamic systems where statistical properties (volatility, correlations, trends) change over time. Models trained on past data may degrade.
  - *Mitigation:* Implement regular model retraining (e.g., using expanding or rolling windows for training data), potentially incorporate features designed to capture regime shifts (like the unsupervised regime labels), test model performance stability across different sub-periods of the test set.
- **Low Signal-to-Noise Ratio:** The underlying signal driving predictable returns is often very weak compared to random market noise.
  - *Mitigation:* Focus on robust feature engineering, utilize models with appropriate complexity (avoiding overly complex models prone to fitting noise), employ strong regularization techniques, prioritize directional accuracy over precise value prediction if applicable. *(Ref: Bias-variance tradeoff; PMML Intro, UDL, Foundations).*
- **Overfitting:** The risk of fitting the training data too closely, including its noise, leading to poor generalization on unseen data.
  - *Mitigation:* Strict use of validation sets for tuning, regularization (L1/L2, dropout, early stopping), cross-validation techniques (like walk-forward), feature selection.
- **Lookahead Bias:** Inadvertently incorporating information from the future into the model's training or feature calculation process.
  - *Mitigation:* Rigorous adherence to chronological data splitting, ensuring all feature calculations at time t use only data available up to t. Careful implementation of rolling window calculations.
- **Efficient Market Hypothesis (EMH):** Acknowledge the theoretical challenge posed by the EMH, particularly the semi-strong form, which posits that publicly available information (including past prices) is already reflected in current prices, making consistent abnormal profits from technical analysis difficult. Any observed predictability may be small, transient, or related to market frictions/behavioral factors not fully captured by the model.

This plan provides a detailed roadmap for the predictive modeling phase of the research, emphasizing methodological soundness and realistic evaluation criteria.

7 Future Work Expansion (Private Repository)

While the preceding sections outline a comprehensive approach based solely on historical OHLCV data, financial markets are significantly influenced by external information, corporate events, macroeconomic narratives, and shifts in investor sentiment often conveyed through textual sources. Purely quantitative models based on past price action may fail to capture the impact of such qualitative or event-driven factors. This section outlines a more ambitious, research-intensive direction for future work, intended for private development due to data sourcing and complexity considerations, focusing on integrating information extracted from unstructured textual data, specifically equity research reports.

Rationale: Equity research reports published by investment banks (e.g., UBS, Redburn, JP Morgan, Morgan Stanley) contain detailed analyses, forward-looking statements, price target revisions, and qualitative assessments (e.g., "Buy", "Hold", "Sell" ratings, discussions of competitive landscape, management changes, product cycles) that directly influence market perception and potentially future price movements of specific stocks like AAPL. Incorporating sentiment and key information extracted from these reports could provide orthogonal signals to the technical features derived from price/volume data, potentially improving prediction accuracy, especially around earnings announcements or major corporate events.

Approach 1: NLP-Driven Sentiment Feature Integration This approach focuses on extracting quantifiable sentiment signals from research reports and integrating them as features into the supervised learning framework described in Section 4.

1. **Data Acquisition and Preprocessing:**
   - Obtain a corpus of historical equity research reports specifically covering AAPL from target sources (UBS, Redburn, JPM, MS, etc.). This presents a significant challenge regarding access, licensing, and potentially cost.
   - Implement robust PDF parsing pipelines to extract textual content, handling diverse formats, tables, and images. Optical Character Recognition (OCR) might be necessary for scanned documents.
   - Develop methods for report segmentation (e.g., identifying summary, thesis, valuation sections) and associating reports accurately with their publication dates.
2. **Sentiment Analysis Model Selection and Application:** Explore various NLP techniques to quantify sentiment, ranging from simpler lexicon-based methods to sophisticated transformer models:
   - *Lexicon-based (Baseline):* Utilize VADER (Valence Aware Dictionary and sEntiment Reasoner) or potentially Loughran-McDonald financial sentiment dictionaries to generate polarity scores based on word counts. These are computationally efficient but may lack contextual understanding.
   - *Pre-trained Transformer Models:* Leverage models pre-trained on large text corpora and potentially fine-tuned on financial text:
     - **FinBERT:** A family of BERT models specifically pre-trained on large financial corpora (e.g., SEC filings, analyst reports), often showing strong performance on financial NLP tasks like sentiment analysis. Requires fine-tuning on a specific sentiment classification/regression task using labeled financial sentences if available, or can be used for zero-shot classification with appropriate prompts.
     - **General Domain Models (BERT, RoBERTa, etc.):** Can be fine-tuned on financial sentiment datasets (if available) or used within few-shot/zero-shot learning paradigms. Performance might be slightly lower than domain-specific models without extensive fine-tuning.
     - **Proprietary Models (e.g., BloombergGPT):** If accessible, large language models trained extensively on financial data could offer superior performance but come with access restrictions and computational costs.
   - *Aspect-Based Sentiment Analysis (Advanced):* Move beyond document-level sentiment to identify sentiment towards specific aspects (e.g., "iPhone sales", "management strategy", "valuation"). This requires more complex NLP pipelines (aspect extraction + sentiment classification).

3. **Feature Generation:** Convert NLP model outputs into numerical features for integration at time t:
    - Aggregate sentiment scores (e.g., average polarity, net positive minus negative mentions) from reports published within a recent window (e.g., last 1, 5, 21 days).
    - Generate features representing changes in sentiment or rating compared to previous reports from the same source.
    - Create features capturing the volume or intensity of analyst coverage/commentary.
4. **Model Integration:** Include the derived sentiment features S_t into the feature vector $X_t = [X_t^{\text{technical}}, S_t]$ used in the predictive models (Section 4). Evaluate the marginal contribution of sentiment features using feature importance analysis and ablation studies (comparing model performance with and without sentiment features). Consider interaction terms between technical and sentiment features.

Approach 2: Reinforcement Learning for Adaptive Signal Weighting (Exploratory) This more complex approach explores using RL to dynamically determine the importance or weighting of sentiment signals relative to technical signals, potentially adapting to changing market conditions where sentiment might be more or less influential.

1. **Problem Framing (Markov Decision Process - MDP):**
    - *State (s_t):* Represents the current market context, including technical features ($X_t^{\text{technical}}$) and processed sentiment signals (S_t). May also include information about recent prediction errors or market volatility.
    - *Action (a_t):* Could represent a weight $w_t \in [0, 1]$ assigned to the sentiment signal when making a prediction, or a choice between different predictive models (e.g., one technical-only, one sentiment-augmented).
    - *Reward (R_{t+1}):* Defined based on the accuracy of the subsequent prediction made using the chosen action/weight, or based on the profitability of a simulated trade resulting from the action. Designing an effective, non-myopic reward function is critical.
    - *Policy ($\pi(a_t|s_t)$):* The RL agent learns a policy that maps states to actions (or distributions over actions) to maximize expected cumulative future rewards.
2. **RL Algorithm Selection:** Consider algorithms suitable for continuous or discrete action spaces, such as Deep Q-Networks (DQN) for discrete actions or Actor-Critic methods (e.g., DDPG, PPO) for continuous weighting actions. *(Ref: RL concepts; Sutton and Barto, PMML Intro/Advanced, UDL).*
3. **Implementation Challenges:** Requires careful state representation design, reward shaping to encourage desired behavior, significant computational resources for training, and robust simulation environments for evaluation. The inherent noise and non-stationarity of financial markets make stable RL training particularly difficult.

Expected Outcome and Rigor: The integration of NLP-derived sentiment aims to capture information related to fundamental analysis, corporate events, and forward-looking expectations often absent in pure technical analysis. Success would be measured by a statistically significant improvement in predictive accuracy (e.g., lower MSE/MAE, higher directional accuracy) on the test set compared to models using only technical features, particularly during periods driven by news or earnings. The RL approach, while more speculative, seeks to create an adaptive prediction mechanism. Rigor demands careful handling of data biases (e.g., selection bias in

reports), robust NLP model evaluation, and statistically sound comparison of predictive model performance with and without the added textual features.

Disclaimer: Project Origin and Context

This research project was initiated on August 15th, 2023. The foundational work and conceptualization occurred during the author's internship tenure on an equities desk at an established, though unnamed, asset management firm.

The analysis, methodologies, and code presented herein represent the author's original work, and all intellectual property rights associated with this specific research plan and its potential implementation belong solely to the author.

The project initially formed part of a paper trading strategy developed for an internal competition involving interns at the aforementioned asset manager, which switched to another involving participants from three unnamed hedge funds based in London. This competition involved tracking performance metrics such as risk-adjusted returns, Sharpe ratio, and maximum drawdown across various self-selected asset classes (the author focused on equities and equity options). Trading strategies and specific positions were not disclosed between participants, High-Frequency Trading (HFT) strategies were disallowed, and standardized assumptions regarding transaction costs and slippage were applied where relevant.

The decision to keep the involved firms and individuals anonymous is deliberate, respecting the privacy and confidentiality of former colleagues and peers.