**IDC6940 - Capstone Proposal**

Don Krapohl

**Team**

I will be working independently on this capstone project.

**Topic**

I’ve selected to study the use of Random Forest (RF) models supplemented with basic Autoregressive Moving Average (ARMA) techniques to predict the direction (up or down) of the stock market 20 days into the future. The objective is to use my course work, current literature, and intent on future research to classify the market movement as either upward or downward. Because Random Forest has no memory I will use both machine learning and time series modeling to identify optimal lags and moving averages and introduce these variables during feature engineering.

**Methods**

The methodology will combine traditional time series analysis with machine learning classification. The time series aspect will be limited to the addition of engineered features generated based on the outcomes from determining time series parameters of the data. It isn’t clear from the literature if this is an appropriate strategy but additional readings should validate this approach. There are additional approaches using non-overlapping windows that can capture subtrends within the data. If time allows I will add this to my model and evaluate the value it brings.

The first step in my analysis will require building basic ARMA models and to interpret autocorrelation functions (ACF) and partial autocorrelation functions (PACF) in the stock market time series data. I will use the Augmented Dickey-Fuller test to check stationarity and determine if I need to difference the data to remove S&P500 closing price by day trend, fit a basic Ordinary Least Squares (OLS) model, generate the ACF and PACF plots, check Akaike and Bayesian Information Criteria (AIC and BIC respectively) for optimal lags, and generate features for use in my RF model for the lag values from the ACF/PACF and AIC/BIC determinations. These measures may also lead to seasonality features and potential advice on moving average windows.

Once my time series features are engineered I will train a Random Forest classifier to predict the binary target, which is 20-day future market direction (up or down). The choice of Random Forest algorithm was made to take advantage of the “wisdom of the crowd” phenomenon in which multiple moderately successful models can vote to gain an outcome superior to any of the individual models. The RF model also is able to handle non-linear relationships, requires no label encoding, requires no feature reduction, reduces the risk of overfitting, and advises on the most important factors discovered.

Because I have only a few thousand samples I will likely not do a train/test split. I will use 10-fold cross validation with Bootstrap Aggregation (bagging) and Out-Of-Bag (OOB) sampling to do testing and get accuracy measures using samples outside those trained on the specific tree being measured.

After interpreting the ADF results to determine if a trend must be eliminated, I might remove any trend using the first-order differencing (d=1) with second-order differencing applied if warranted. The formula for first-order differencing is:

Yt′​=Y\_t​−Y\_t−1​

Where Y\_t is the original value, Y\_t-1 is the value at lag=1, and Y\_t’ is the differenced series.

The ACF drawn from a fitted ARCH model to identify significant lag structures together with AIC and BIC to determine the optimal lag length for pricing data. ACF is calculated by the arch library as:

ρ(k)= Cov(X\_t​,X\_t−k​)​ / Var(X\_t​)

Where :

X\_t is the value at time t

X\_t-k is the value at time k lagged by k periods

Cov(X\_t​,X\_t−k​)​ is the covariance between the current and lagged value

Var(X\_t​) is the variance of the series

AIC and BIC formulas are:

AIC = 2k – 2\*ln(L)

BIC = k\*ln(n) – 2\*ln(L)

Where:

k is the number of parameters in the model

L is the maximum log-likelihood of the model

n is the sample size

During training I will use the Gini Impurity Index to guide node splitting within each tree. For my predictions I will have two classes to predict at 20 days: price went up or price went down. The Gini index will range from 0 (all samples in the node belong to 1 class) and the maximum possible value for the number of classes in the model if the model is evenly split among multiple classes. The formula for the Gini Impurity Index is:

G = 1 - sum\_{i=1}^{C=2} p\_i^2

Where:

G is the Gini impurity measure at the node

C=2 indicates there are two classes in the response variable

p\_i is the proportion of samples in the node that belong to class i

I will evaluate model performance using accuracy, precision, recall, F1-score, and ROC-AUC and will generate a confusion matrix to visualize how misclassifications are distributed. I will conduct feature importance analysis to understand which features, engineered lags, and moving averages contribute most significantly to predictions. I may reduce the feature set based on the results of my model quality testing.

**Programming**

I’ll code in Python, using libraries such as pandas for data manipulation, statsmodels for ARMA modeling, scikit-learn for Random Forest implementation, and matplotlib+seaborn for visualization.

**Data**

The dataset will be built from historical daily Standard and Poor’s 500 (S&P 500) stock market index data covering January 1, 1990 to February 16, 2024. The beginning date was selected to ensure the data captured recent price patterns without factoring in historical anomalies. The end date was selected as it is when the Kaggle 34-year dataset ends. Data will be sourced from Yahoo Finance and Kaggle and include key variables such as date, open, high, low, close, and trading volume. Data will be accessed through Python APIs (such as yfinance) or downloaded CSV files and stored locally for analysis. The dataset will be a few dozen kilobytes composed of ~30 features and ~9000 samples.

I have done basic exploration to ensure my data sources are viable and I can generate some of the features common to the papers I have reviewed. My current intent is to generate the following columns:

Source: K=Kaggle, Y=yahoo, TA=python technical analysis, EF=engineered feature

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Meaning** | **Type** | **Source** |
| Date | Date of observation in YYYY-MM-DD format | Index | K |
| vix | VIX (Volatility Index) - market volatility. | Volatility | K |
| sp500\_volume | Daily trading volume for the S&P 500 | Volume | K |
| djia | Dow Jones Industrial Average (DJIA) price | Index | K |
| djia\_volume | Daily trading volume for the DJIA. | Volume | K |
| hsi | Hang Seng Index-Hong Kong stock market | Index | K |
| Ads | Aruoba-Diebold-Scotti (ADS) business cond. | Macro | K |
| Us3m | U.S. Treasury 3-month bond yield | Macro | K |
| Joblessness | U.S. unemployment rate | Macro | K |
| epu | Economic Policy Uncertainty Index | Macro | K |
| GPRD5 | Geopolitical Risk Index (Daily) | Geopol. | K |
| Prev\_day | Previous day’s S&P 500 closing value. | Price | k |
| sp500\_open | Open price (USD) | Price | Y |
| sp500\_high | High price for the day | Price | Y |
| sp500\_low | Low price for the day | Price | Y |
| sp500\_close | Closing price for the day | Price | Y |
| sp500\_adj\_close | Unknown. Came with Kaggle dataset. | Price | Y |
| sp500\_ohlc\_volume | Day trading volume | Volume | Y |
| 1d\_return | 1-day gain or loss | Return | EF |
| direction\_1d | Up/down direction from previous close | Direction | EF |
| direction\_5d | Up/down direction over last 5 days | Direction | EF |
| direction\_20d | Up/down direction over last 20 days | Direction | EF |
| direction\_50d | Up/down direction over last 50 days | Direction | EF |
| direction\_200d | Up/down direction over last 200 days | Direction | EF |
| macd | Moving Average Convergence Divergence | Indicator | TA |
| macd\_signal | Moving Average Convergence Divergence signal | Indicator | TA |
| roc | Rate of Change | Indicator | TA |
| rsi | Relative Strength Indicator | Indicator | TA |
| stoch\_k | Stochastic k value | Indicator | TA |
| stoch\_d | Stochastic d value | Indicator | TA |
| adx | Average Directional Index | Indicator | TA |
| obv | On Balance Volume | Indicator | TA |
| atr | Average True Range | Indicator | TA |
| bb\_upper | Bollinger Bands Upper value | Indicator | TA |
| bb\_middle | Bollinger Bands Middle value | Indicator | TA |
| bb\_lower | Bollinger Bands Lower value | Indicator | TA |
| ema\_12 | 12-day exponential moving average | Indicator | TA |
| ema\_26 | 26 day exponential moving average | Indicator | TA |
| sma\_20 | 20 day simple moving average | Indicator | TA |
| sma\_50 | 50 day simple moving average | Indicator | TA |
| sma\_200 | 200 day simple moving average | Indicator | TA |
| 1d\_return\_pct | 1 day price gain in percent | Return | EF |
| return\_lag\_1 | Previous day close (lag=1) | Return | EF |
| return\_lag\_2 | Close 2 days ago (lag=2) | Return | EF |
| return\_lag\_3 | Close 3 days ago (lag=3) | Return | EF |
| return\_lag\_11 | Close 11 days ago (lag=11) | Return | EF |
| roll\_mean\_5 | 5-day rolling mean | Return | EF |
| roll\_std\_5 | 5-day rolling standard deviation | Return | EF |
| roll\_mean\_20 | 20-day rolling mean | Return | EF |
| roll\_std\_20 | 20-day rolling standard deviation | Return | EF |

**References (tentative)**

Sadorsky, P. (2021). A random forests approach to predicting clean energy stock prices. Energy Economics, 100, 105339.

Basak, S., Kar, S., Saha, S., & Khaidem, L. (2019). Predicting the direction of stock market prices using tree-based classifiers. North American Journal of Economics and Finance, 47, 552–567

Ghosh, P., Neufeld, A., & Sahoo, J. K. (2022). Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. Finance Research Letters, 46, 102280. <https://doi.org/10.1016/j.frl.2021.102280>

Lohrmann, C., & Luukka, P. (2019). Classification of intraday stock market movements using multiple time frames. Applied Intelligence, 49(12), 4296–4311. <https://doi.org/10.1007/s10489-019-01505-1>

Tratkowski, G. (2020). Identification of nonlinear determinants of stock indices derived by random forest algorithm. International Journal of Management and Economics, 56(3), 209–217.

Demir, C. (2019). Macroeconomic determinants of stock market fluctuations: The case of BIST- 100. Economies, 7(1), 8

Yahoo Finance and Kaggle datasets for historical market data.

Brockwell, P. J., & Davis, R. A. (2016). Introduction to Time Series and Forecasting (3rd ed.). Springer.