Capstone Project

StockMarket Predictions Final Report

## An Examination of AI and Machine Learning Techniques for Stock Market Predictions

[A graph on a screen

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/stock-banner.jpg)

**Author: Wayne H. Lee**

## Executive Summary

Being able to predict stock price movement can help inform investment decisions. If a stock is predicted to go up within a particular timeframe, it can be bought, held for that period, and then sold for a gain.

The goal of this project is to determine whether a machine learning model can be developed to predict the days on which a stock is expected to rise—using historical data—with better results than buying on random days. Ideally, if a trader buys only on days indicated by the model, they would expect a net gain over losses. In this scenario, only when the model predicts a daily price increase, trades are made each day after the market opens and then sold at closing.

The trader would like to maximize the number of Desired True Positives which represent days that the model predicted and in fact the stock went up.

The following should be minimized: "Costly Mistakes" because those are days where the model incorrectly indicated a purchase should be made but money was lost as the price went down. And (2) "Missed Opportunities" are days that the stock value increased but the model incorrectly did not indicate a purchase should be made.

[A blue squares with white text

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/IdealConfusionMatrix.png)

### Findings

Best Evaluated Model: Support Vector Machine

Of the classification type prediction models evaluated, the SVM/SVC model performed the best at predicting price increases. It achieved the highest F1 score of 0.6685, which takes into account missed opportunities, and a precision score of 0.62. The ratio of money-making predictions (TP: 121) to money-losing predictions (FP: 73) was 1.66 to 1.

Important Inputs for Prediction

An inspection revealed the most important historical and calculated features (inputs) as represented by one of the models:

1. Two-day horizon ratio (opening price divided by the average closing price for the past 2 days)
2. Yesterday's volatility index (VIX)
3. 2-day price change
4. 1-month price change
5. Number of days since the last dividend
6. Yesterday's volume

Inputs with High Correlation With Outcomes

As part of exploratory data analysis (EDA), it was discovered that the highest correlations with price changes were associated with the two-day horizon ratio (opening price divided by the average closing price of the previous two days) as well as the one-day and two-day price changes. This finding suggests some market momentum, where price increases on previous days can help predict continued upward movement.

[A close up of a screen

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/CorrelationHeatmapSummary.png)

Standard Distribution of Price Changes

Changes in the daily price (delta) appear to form an approximately bell-shaped curve. This indicates that small corrections in the stock value occur more frequently, whereas large price changes—either positive or negative—are relatively rare. Future research could leverage this distribution to estimate the potential magnitude of price changes offering additional insights for traders.

[A graph of a blue line with Ryugyong Hotel in the background

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/out/histogram_daily_close_change.png)

### Results and Conclusion

The highest performing model tested was a Support Vector Machine. For the stock examined in this project (Apple Computers - AAPL), the model resulted in a 1.66 to 1 ratio of money making trade vs money losing trades. This indicates that the model could beat out purchasing stocks on a random selection of days, which for this stock has a 1:13 to 1 ratio.

### Next Steps and Recommendations

While this preliminary investigation provides a useful model, additional enhancements can be made.

* Continue to tune the SVM and Decision Tree models. For example, exploring different the C value and kernels for SVM.
* Continued exploration in feature selection using PCA to see if better results can be achieved with fewer features.
* Additional performance metric features in the form of derived attributes.
* Investigate additional models such as neural networks.
* Attempting to understand the magnitude of price changes by using regression models.
* Using additional input data such as news sentiment analysis and other market indices.
* Analyze the stocks of other companies other than Apple Computer

## Rationale

Being able to predict stock price movement can help inform investment decisions. If a stock is predicted to go up within a particular timeframe, it can be bought, held for that period, and then sold for a gain.

## Research Question

Can stock price movement be predicted using historic stock market data?

## Data Sources

Historical Apple Computer (AAPL) stock price data was gathered from Yahoo! Finance. This including historic prices, volume, and dividends. A market indicator, Volatility Index (VIX) was also retrieved. <https://finance.yahoo.com/>

A five year time window between March 16, 2020 to March 14, 2025 was selected to get sufficient data necessary for training of approximately 1000 daily records. A total of 1257 records were collected to ensure incomplete historic information at the beginning of the time series data could be discarded.

### Exploratory Data Analysis

The quality of the data from Yahoo! Finance is excellent. Data was provided for all days the market was open (weekdays excluding trading holidays).

It does not appear that closing prices of this stock follows a normal distribution. The standard deviation was 41.24.

[A graph showing a blue line

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/out/histogram_of_closing_prices.png)

Changes in the daily price (delta) appear to form an approximately bell-shaped curve. Standard Deviation was 2.80 and the maximum change on a particular day was $14.03. The mean of the distribution is slightly offset to the right (around 0.11), which aligns with the observation that this stock has generally been trending upward over the examined period.

[A graph of a blue line with Ryugyong Hotel in the background

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/out/histogram_daily_close_change.png)

### Cleaning and Preparation

Training features cannot include future data that is not available at the time the model is trained and used to predict future outcome. At this time, there were only numerical features present in the source training data.

The overall cleaned dataset will be split into a Training dataset used for training the models, and Dev Test dataset used to validate the model. Twenty-five percent of the dataset will be used for Dev Test, and the remaining 75% will be used for training.

The target classification (UP, DOWN) is obtained by subtracting the day's close from the previous day.

### Preprocessing

Relevant features for training are selected. Additionally feature engineering will need to be done for to shift values for time series analysis. Additionally, key metrics such as Horizon indexes can be calculated from the source data.

To perform the analysis, feature engineering is required to bring key historic and calculated metrics into one row of the DataFrame to train the model.

1. Previous Close to Open
2. Volume 1-day, 2-day
3. Close 1-day, 2-day, 3-day, 1-week, 2-week, 1-month
4. Delta 1-day, 2-day, 3-day, 1-week, 2-week, 1-month
5. VIX 1-day, 2-day, 3-day
6. Last Dividend Amount, Last Dividend Date, Days Since Dividend
7. Horizon Ratio 2-day, 1-week, 1-month
8. Increases 1-day, 2-day, 1-week, 1-month
9. Daily Close Change, Up or Down

### Final Dataset

Datasets are combined. Rows with NAs in calculated values due to missing historic values at the beginning of the dataset are removed. A total of 1218 records remain, meeting our training threshold of at least 1000 records.

## Methodology

For this application, an industry standard model called CRISP-DM is used. This process provides a framework for working through a data problem.

[Diagram of a diagram of data

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/crisp.png)

### Evaluation Metric

In this scenario where the trader only purchases on days where model predicts UP. The trader would like to maximize the number of True Positives which represent days that the model predicted and in fact the stock went up.

The following should be minimized: (1) False Positives can be labelled as "Costly Mistakes" because those are days where the model indicated a purchase should be made and money was lost as the price went down. (2) Missed Opportunities are represented by False Negatives as those are days that the stock value increased but the model did not indicate a purchase should be made.

True Negatives are irrelevant as in this scenario as a purchase would not have been made.

The desired confusion matrix for the scenario can be seen here. UP represents a day where the price increased, and DOWN represents a day where the price decreased.

[A blue squares with white text

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/IdealConfusionMatrix.png)

The following evaluation metrics are considered:

**Accuracy Score**

Accuracy is the number of correct predictions divided by the number of predictions and is a good starting point for general effectiveness of the model. However there are better metrics for this use case. $$ \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} $$

**Precision Score**

In this scenario, we only buy when the model tells us to. We want to minimize the number of false positives as false positives cause us to lose money. Recall is the ratio of true positives over true and false positives. This metric correctly penalizes false positives but there may be some missed opportunities in the form of false negatives. $$ \text{Precision} = \frac{TP}{TP + FP} $$

**Recall Score**

Defined as the ratio of true positives over true positives and false negatives, this metric helps us catch all opportunities as it penalizes missed opportunities (false negatives). But it misses on minimizing costly mistakes (false negatives).

Recall=TPTP+FN

**F1 Score**

Because we want to maximize true positives and minimize missed opportunities(FN) and costly mistakes (FP), the F1 score is optimal. This metric also known as a threat risk score and optimizes detecting an event. By definition, the metric maximizes TP by putting it in the numerator and minimizes FN and FP by putting it in the denominator. TN is omitted as it is not required. $$ F\_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} $$

F1=2×TP2×TP+FP+FN

## Models

Four models were considered, tuned and evaluated.

**K Nearest Neighbors (KNN)** A KNeighborsClassifier was created inside a pipeline as a default model to use as a baseline. GridSearch Cross Validation was used with the F1 evaluation metric to find the optimal hyperparameter for 'k' representing nearest neighbos. When compared with accuracy as a scoring metric, the results were the simiar. The GridSearch CV algorithm found that 91 nearest neighbors was the most optimal.

**Decision Tree** Using default values, a DecisionTreeClassifier with the default 'gini' criterion was used and a tree of depth 17 was produced. The default setting produced a training accuracy of 1.0 suggesting some overfitting. Tuning the hyperparameter 'max\_depth' using GridSearchCV identified a flat tree with a max\_depth=1 as optimal with the training set. This reduced the training accuracy but increased test accuracy and F1 scores confirming that overfitting was previously happening.

**Random Forest Classifier** To avoid the overfitting issues seen in the decision tree, the RandomForestClassifier was used with n\_estimators = 100 and min\_samples\_split=100 to ensure a reasonable division of the training set.

**SVM (Support Vector Machine)** A Support Vector Classifier Model was used in a pipeline inline with a StandardScalar transformer.

## Model Evaluation and Results

With the Ideal Scenario Confusion Matrix above in mind, the confusion matrix for each model can be used to evaluate performance.

[A blue squares with white text

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/IdealConfusionMatrix.png)

[A graph of confusion matrix

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/out/confusion_matrix_for_svm.png)

[A graph of a forest classifier

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/out/confusion_matrix_for_random_forest_classifier.png)

[A blue squares with white text

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/out/confusion_matrix_for_tuned_decision_tree_tuning.png)

[A blue squares with white text

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/out/confusion_matrix_for_tuned_knn.png)

[A screenshot of a graph

AI-generated content may be incorrect.](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/images/CrossValidationResults.png)

**SVM (Support Vector Machine) Model** Of the models evaluated, the SVC model performed the best at predicting price increases. It had the highest F1 score at 0.6685 and a precision score of 0.62. It was also the best model at identifying true positives and minimizing missed opportunities. The ratio of money making predictions (TP: 121) to money losing predictions (FP: 73) was 1.66 to 1.

**Random Forest Classifier Model** The Random Forest Classifier is an ensemble model that uses multiple decision trees to improve predictive performance and reduce overfitting. It had a similar F1 performance at 0.6684 however was just a bit behind SVM in Precision at 0.61.

**Decision Tree Model** The tuned Decision Tree produced the highest Precision score of the models considered at 0.627. The result is the best ratio of money making trades to money losing trades at 1.69 to 1. However, it had only the third highest F1 score at 0.663 meaning that it had a higher rate of "missed opportunities."

**K Nearest Neighbor Model** After tuning, the KNN model had a F1 score of 0.617, a precision score of 0.574, and test accuracy of 0.544.

## Outline of Project

* Jupyter Notebook: [StockMarketPredictions.ipynb](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/StockMarketPredictions.ipynb)
* Download Data: [data.zip](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/data/data.zip)
* Final Report: [Stock Market Predictions Final Project Report.doc](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/Stock%20Market%20Predictions%20Final%20Project%20Report.docx)

## Contact and Further Information

Wayne Lee

Email: [wayneone@gmail.com](mailto:wayneone@gmail.com)

[LinkedIn](https://github.com/im360wayne/AI-ML-StockMarketPredictions/blob/main/linkedin.com/in/wayneone)