Capstone Project 2
Stock Market Trend Predictions using Random Forests and Gradient Boosting

Introduction and Objective

1. Predict the price movement from 1 to 20 days ahead and decide how many days ahead has a better prediction

I am going to predict the stock market trend (up or down) as a starting point by looking at technical indicators and by using Random Forest and Gradient Boosting. Next, I will evaluate the results with accuracy scores and F1 scores, from which we can see how many days ahead would be a better stock movement prediction.

2. Compare the performance between Random Forest and Gradient Boost

One of the main advantages of the random forests model is that we do not need any data scaling or normalization before training it. Also the model does not strictly need parameter tuning such as in the case of support vector machine (SVM) and neural networks (NN) models.

In addition, since gradient boosting seems to be the most efficient algorithm in recent years, I will compare the result between random forest and gradient boosting.

3. Compare technology stocks and retail stocks

With the result of random forest and gradient boost, I would like to apply them to technology stocks (Apple, Microsoft) and retail stocks (Costco, JNJ) to see if they get similar results.

Target Audience and Why

My target audience will be investors who are interested in trading stocks. By looking at the results from technical indicators, we can have a better understanding of how stocks will move from the history records. Along with the personal stock knowledge, investors can decide to buy or sell the stocks at the right time to avoid loss.

Dataset Acquisition

Data was scraped from Yahoo Finance for the stocks from 1/1/1998 to 6/30/2016. The dataset comprises “closing prices”, “High”, “Low”, and “Volume” of all stocks.

Data Wrangling

Steps required to clean and modify the data into a necessary format for analysis.

1. Clean Data: Include only “Adj Close”, “High”, “Low”, “Volume” for each stock
2. Make sure the index is datetime.
3. Convert all columns to lowercase for future analysis.
4. Convert Volume to float type.
5. Use the TA-Lib package to get stock indicators.

Final Dataset:

All stocks have the same time frame and dataset.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4654 entries, 1998-01-02 to 2016-06-30
Data columns (total 6 columns):
high         4654 non-null float64
low          4654 non-null float64
open         4654 non-null float64
close        4654 non-null float64
volume       4654 non-null float64
adj close    4654 non-null float64
dtypes: float64(6)
memory usage: 254.5 KB
```

Exploratory Data Analysis

Most Of the initial data analysis was regarding the stocks movement and their associated counts. More advanced analysis will require supervised machine learning models to be created.

```
stocks['COST'].describe()
```

	high	low	open	close	volume	adj close
count	4654.000000	4654.000000	4654.000000	4654.000000	4.654000e+03	4654.000000
mean	66.636762	65.255954	65.938362	65.976239	3.818503e+06	53.177326
std	36.004317	35.755179	35.881064	35.894717	2.636948e+06	34.229855
min	21.750000	20.625000	21.093750	21.062500	1.033000e+05	15.323649
25%	40.405001	39.052500	39.724063	39.772500	2.234550e+06	28.975789
50%	53.720001	52.305000	53.014999	53.066250	3.253300e+06	39.647673
75%	83.675001	82.170002	83.009998	83.002502	4.704550e+06	65.669125
max	169.729996	165.869995	167.330002	168.869995	5.677350e+07	154.739960

Machine Learning

The primary goal of the machine learning (ML) section is to identify which algorithm can be used to properly predict stock movements. I will compare random forest and gradient boosting algorithms and will focus on testing and understanding key parameters as relates to the exploratory data analysis findings above.

Before we start, I would like to introduce the technical indicators that I will use in this project. The technical indicators are calculated with their default parameters settings using the TA-Lib python package. They are summarized in the table below where Pt is the closing price at the day t, Ht is the high price at day t, Lt is the low price at day t, HHn is the highest high during the last n days, LLt is the lowest low during the last n days, and EMA(n) is the exponential moving average.

Technical indicators:

Type	Abbreviation	Name
Overlap Studies	BBANDS	Bollinger Bands
	SMA	Simple Moving Average
	WMA	Weighted Moving Average
Momentum Indicators	MOM	Momentum
	MACD	Moving Average Convergence Divergence
	RSI	Relative Strength Index
	WILLR	Williams' Percentage R
	CCI	Commodity Channel Index
	ADX	Average Directional Movement Index
Volume Indicators	ADOSC	Chaikin A/D Oscillator
	OBV	On Balance Volume

Stock Indicator	Formula	Parameters
Simple Moving Average (SMA(5), SMA(10)) 5-days and 10-days	$\frac{P_t + P_{t-1} + \cdots + P_{t-(n-1)}}{n}$	n = 5, n = 10
Weighted Moving Average (WMA(5), WMA(10)) 5-days and 10-days	$\frac{nP_t + (n-1)P_{t-1} + \cdots + P_{t-(n-1)}}{n + (n-1) + \cdots + 1}$	n = 5, n = 10
Stochastic %K	$\frac{P_t - LL_n}{HH_n - LL_n}$ x 100	n = 5
Stochastic %D	$\frac{\sum_{i=0}^{n-1} \% K_{t-i}}{n}$	n = 3
10-days Momentum	$P_t + P_n$	n = 10
Moving Average Convergence Divergence (MACD)	$EMA(N_{fast})_t - EMA(N_{slow})_t$	$N_{fast} = 12,\ N_{slow} = 26$
Relative Strength Index (RSI)	$100 - \frac{100}{1 + \frac{sum\ of\ gains\ over\ the\ past\ n\ days}{sum\ of\ losses\ over\ the\ past\ n\ days}}$	n = 14
Williams' %R	$\frac{H_n - P_t}{H_n - L_n}$ x 100	n = 14
Commodity Channel Index (CCI)	$\frac{\frac{P + H + C}{3} - SMA(n)_t}{0.015 - \alpha(n)_t}$	n = 14
Chaikin A/D Oscillator	$EMA(N_{fast})_t\ of\ A/D\ line - EMA(N_{slow})_t$ of A/D line	$N_{fast} = 3,\ N_{slow} = 10$
On Balance Volume (OBV)	$OBV_t = OBV_{t-1} + volume,\ if\ close_t > close_{t-1}$ $OBV_t = OBV_{t-1} + 0,\ if\ close_t = close_{t-1}$ $OBV_t = OBV_{t-1} - volume,\ if\ close_t < close_{t-1}$	

Overlap Studies

SMA (Simple Moving Average)

A simple moving average (SMA) calculates the average of a selected range of prices, usually closing prices, by the number of periods in that range. The SMA is a technical indicator that can aid in determining if an asset price will continue or reverse a bull or bear trend.

WMA (Weighted Moving Average)

A Weighted Moving Average puts more weight on recent data and less on past data. This is done by multiplying each bar's price by a weighting factor. Because of its unique calculation, WMA will follow prices more closely than a corresponding Simple Moving Average.

A rising WMA tends to support the price action, while a falling WMA tends to provide resistance to price action. This strategy reinforces the idea of buying when price is near the rising WMA or selling when price is near the falling WMA.

Momentum Indicators

MOM (Momentum)

The Momentum (MOM) indicator compares the current price with the previous price from a selected number of periods ago. This indicator is similar to the “Rate of Change” indicator, but the MOM does not normalize the price, so different instruments can have different indicator values based on their point values.

MACD (Moving Average Convergence Divergence)

Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security’s price. The MACD is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA.

The result of that calculation is the MACD line. A nine-day EMA of the MACD called the "signal line," is then plotted on top of the MACD line, which can function as a trigger for buy and sell signals. Traders may buy the security when the MACD crosses above its signal line and sell - or short - the security when the MACD crosses below the signal line. Moving Average Convergence Divergence (MACD) indicators can be interpreted in several ways, but the more common methods are crossovers, divergences, and rapid rises/falls.

RSI (Relative Strength Index)

The relative strength index (RSI) is a momentum indicator that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset. The RSI is displayed as an oscillator (a line graph that moves between two extremes) and can have a reading from 0 to 100.

Traditional interpretation and usage of the RSI are that values of 70 or above indicate that a security is becoming overbought or overvalued and may be primed for a trend reversal or corrective pullback in price. An RSI reading of 30 or below indicates an oversold or undervalued condition.

Williams %R (William′s Percentage Range)

Williams %R, also known as the Williams Percent Range, is a type of momentum indicator that moves between 0 and -100 and measures overbought and oversold levels. The Williams %R may be used to find entry and exit points in the market. The indicator is very similar to the Stochastic oscillator and is used in the same way.

A reading above -20 is overbought. A reading below -80 is oversold.

CCI (Commodity Channel Index)

The Commodity Channel Index (CCI) is a momentum-based oscillator used to help determine when an investment vehicle is reaching a condition of being overbought or oversold. It is also used to assess price trend direction and strength. This information allows traders to determine if they want to enter or exit a trade, refrain from taking a trade, or add to an existing position. In this way, the indicator can be used to provide trade signals when it acts in a certain way.

ADX (Average Directional Movement Index)

ADX can be used to measure the overall strength of a trend. The ADX indicator is an average of expanding price range values.

ADX Value	Trend Strength
0-25	Absent or Weak Trend
25-50	Strong Trend
50-75	Very Strong Trend
75-100	Extremely Strong Trend

When the ADX turns down from high values, then the trend may be ending. You may want to do additional research to determine if closing open positions is appropriate for you.

If the ADX is declining, it could be an indication that the market is becoming less directional, and the current trend is weakening. You may want to avoid trading trend systems as the trend changes.

Volume Indicators

Chaikin Oscillator

The Chaikin Oscillator is a third-derivative technical analysis indicator – an indicator of an indicator, the latter of which is derived from the stock price. The oscillator builds on the concept of moving average convergence divergence or MACD. MACD is derived from the moving average, which is the mean price of an issue over a certain period. The Chaikin Indicator applies MACD to the accumulation-distribution line rather than closing price.

A cross above the accumulation-distribution line indicates that market players are accumulating shares, securities or contracts, which is typically bullish.

Random Forest

One of the advantages of random forests is that it does not strictly need parameter tuning. Random forests is an aggregation of another weaker machine learning model, decision trees.

First, a bootstrapped sample is taken from the training set. Then, a random number of features are chosen to form a decision tree. Finally, each tree is trained and grown to the fullest extend possible without pruning. Those three steps are repeated n times for random decision trees.

Each tree gives a classification and the classification that has the most votes is chosen. For the number of trees in the random forests, I chose 300 trees. I could go for a higher number but according to research, a larger number of trees does not always give better performance and only increases the computational cost. Since we will not be tuning the model's parameters, we are only going to split the data to train and test set (no validation set).

To evaluate the model, I will use the accuracy score and the f1 score. The accuracy score is simply the percentage (or fraction) of the instances correctly classified. The f1 score is calculated by below.

$$\mathrm{F1} = 2\,\frac{precision \times recall}{precision + recall}$$

$$precison = \frac{tp}{\mathrm{tp} + fp}$$

$$recall = \frac{tp}{\mathrm{tp} + fn}$$

In the above calculation, tp is the number of positive instances classified as positive, fp is the number of negative instances classified as positive, and fn is the number of positive instances classified as negative. Because of the randomness of the model, each train set is trained 5 times and the average of the scores on the test set is the final score. All of the calculations were done by python's scikit-learn library.

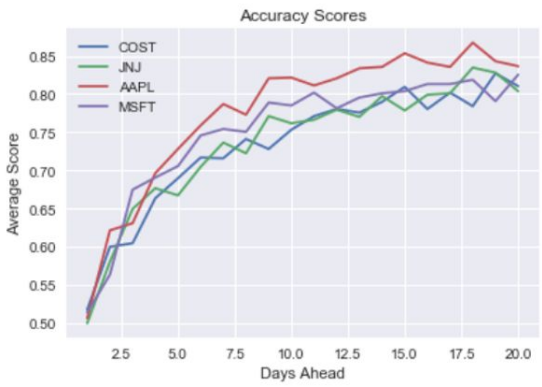
Random Forest Results:

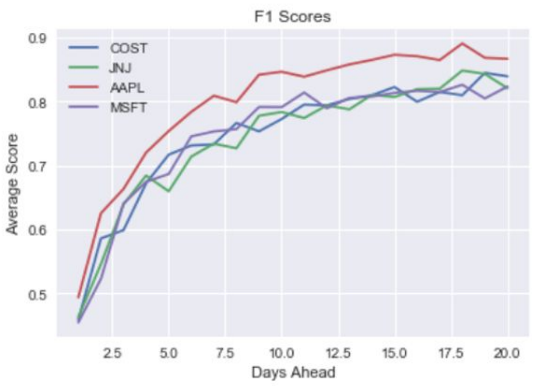
Accuracy Table

	AAPL	MSFT	COST	JNJ
1	0.506542	0.514019	0.517757	0.5
2	0.621495	0.563551	0.6	0.580374
3	0.630841	0.674766	0.604673	0.649533
4	0.696262	0.690654	0.663551	0.676636
5	0.728037	0.705607	0.68972	0.66729
6	0.758879	0.745794	0.716822	0.704673
7	0.786916	0.754206	0.715888	0.736449
8	0.772897	0.750467	0.741121	0.72243
9	0.820561	0.788785	0.728037	0.771028
10	0.821495	0.785047	0.753271	0.761682
11	0.811215	0.801869	0.771028	0.766355
12	0.820561	0.781308	0.780374	0.779439
13	0.833645	0.795327	0.775701	0.770093
14	0.835514	0.800935	0.78972	0.797196
15	0.853271	0.803738	0.809346	0.778505
16	0.841121	0.813084	0.780374	0.799065
17	0.835514	0.813084	0.801869	0.800935
18	0.86729	0.818692	0.784112	0.834579
19	0.842991	0.790654	0.827103	0.828037
20	0.836449	0.825234	0.81028	0.803738

F1 Score Table

	AAPL	MSFT	COST	JNJ
1	0.494253	0.454927	0.459119	0.46339
2	0.625347	0.522983	0.586074	0.546006
3	0.663257	0.640496	0.599052	0.639076
4	0.720069	0.674533	0.672131	0.684307
5	0.753599	0.686567	0.716724	0.659656
6	0.783557	0.745318	0.731145	0.713768
7	0.808725	0.753052	0.732865	0.733962
8	0.798674	0.756609	0.766245	0.726771
9	0.841584	0.791128	0.753181	0.777475
10	0.846092	0.790909	0.772414	0.783347
11	0.8384	0.813708	0.794979	0.77396
12	0.848341	0.789189	0.793679	0.793706
13	0.8576	0.805333	0.803601	0.787565
14	0.864615	0.807588	0.809483	0.810149
15	0.872668	0.813167	0.8223	0.806846
16	0.870427	0.816514	0.799317	0.818871
17	0.864407	0.814471	0.814685	0.819644
18	0.890263	0.825853	0.809563	0.848069
19	0.867925	0.804538	0.844407	0.842735
20	0.86631	0.823084	0.839017	0.820513





Extreme Gradient Boosting (XG Boost)

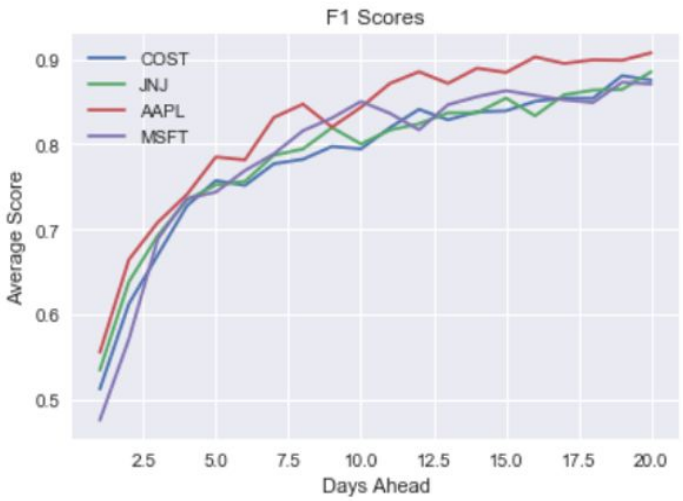
XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework.

XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solves many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples.

It has gained much popularity and attention recently as it was the algorithm of choice for many winning teams of many machine learning competitions.

F1 Score Table

	AAPL	MSFT	COST	JNJ
1	0.555556	0.47591	0.512351	0.534653
2	0.66443	0.570878	0.612457	0.638413
3	0.708402	0.688827	0.670103	0.692927
4	0.74092	0.736185	0.727572	0.734375
5	0.78515	0.743891	0.757377	0.752228
6	0.781628	0.769094	0.751634	0.756432
7	0.83153	0.788707	0.777152	0.787252
8	0.847148	0.815878	0.782324	0.794425
9	0.820233	0.83085	0.797417	0.819618
10	0.84343	0.850258	0.794544	0.800334
11	0.871483	0.836331	0.819536	0.816736
12	0.885246	0.816927	0.841017	0.824104
13	0.871525	0.84648	0.828685	0.836892
14	0.889401	0.855925	0.838174	0.836903
15	0.884375	0.862882	0.83927	0.854288
16	0.902602	0.857391	0.850482	0.833471
17	0.89497	0.852204	0.854183	0.858091
18	0.899248	0.849073	0.854098	0.863524
19	0.898638	0.87276	0.880562	0.864379
20	0.907449	0.870968	0.875294	0.885167



Conclusion

I extracted the highest F1 score of 1-20 days ahead from the random forest algorithm and Extreme Gradient Boosting algorithm as the F1 Score Comparison table below. With the table, we can answer our questions from the very beginning.

Highest F1 Score Comparison

	AAPL	MSFT	COST	JNJ
Random Forest	0.8902	0.8258	0.8444	0.8480
Extreme Gradient Boosting	0.9074	0.8727	0.8805	0.8851

1. **Predict the price movement from 1 to 20 days ahead and decide how many days ahead has a better prediction**

   The random forest and XG Boost algorithms with ten technical indicators show better results when predicting the price movement mostly within 18 to 20 days ahead with scores ranging from 0.80 to 0.91 for f1 scores. We probably could have better results if we add more features that include fundamentals and macro economic variables.

2. **Compare the performance between Random Forest and Gradient Boost**

   The highest F1 score of extreme gradient boosting are higher than that of random forest. Therefore, we can conclude that the extreme gradient boosting performs better than random forest.

3. **Compare technology stocks and retail stocks**
   The highest F1 score has no huge difference between technology stocks and retail stocks. Therefore, we can conclude that the these two algorithms can be applied to either technology stocks or retail stocks.