Predicting the Stock Market

Modeling 30 Years of S&P 500 Trading

Aria Algamehand, Alec Anderson, Reed Oken
University of San Diego, MS Applied Artificial Intelligence
AAI 501: Introduction to Artificial Intelligence
Video presentation: https://youtu.be/I0vB-F7O164
Github repo: https://github.com/okenreed/aai501_final

Introduction

- The stock market is a key indicator of economic stability and a source of investment opportunities.
- Accurately predicting future trends in the stock market is a challenging and complex task that requires extensive data analysis and the use of advanced machine learning algorithms.
- The project goal is to analyze 30 years of S&P 500 stock data to build a system that can make accurate predictions about future trends in the market.
- The project will utilize a variety of machine learning algorithms such as regression, time series analysis, and deep learning to train models on historical data and predict future market trends.

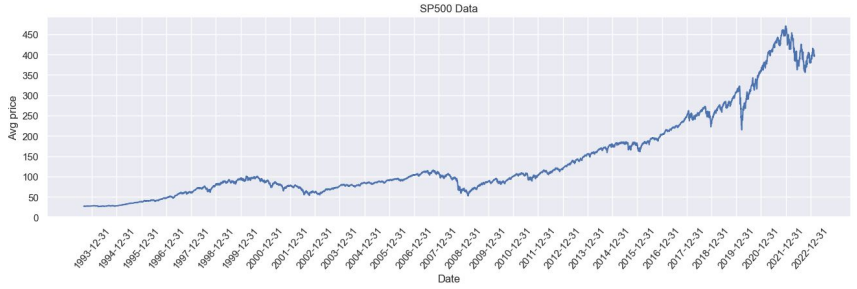
Data cleaning and preparation

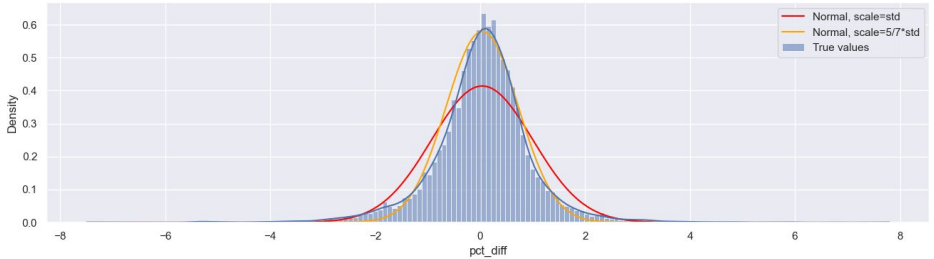
- Data is historically well maintained
  - Available on multiple platforms, Kaggle, Yahoo Finance
  - No null/NaN values
- Date column was an object rather than a date string

Date	Open	High	Low	Close	Volume	Day	Weekday	Week	Month	Year
1993-01-29	25.236158	25.236158	25.110605	25.218222	1003200	29	4	4	1	1993
1993-02-01	25.236146	25.397572	25.236146	25.397572	480500	1	0	5	2	1993
1993-02-02	25.379673	25.469354	25.325865	25.451418	201300	2	1	5	2	1993
1993-02-03	25.487270	25.738376	25.469334	25.720440	529400	3	2	5	2	1993
1993-02-04	25.810132	25.881876	25.523153	25.828068	531500	4	3	5	2	1993

Exploratory data analysis

- Focused on feature extraction
  - Including technical indicators in dataset
- Worked to identify a target feature





Technical indicators

- Relative strength index (RSI)
  - Momentum oscillator bounded from 0 to 100 over a time interval

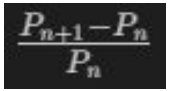
$$RSI_{\text{step one}} = 100 - \left[\frac{100}{1 + \frac{\text{Average gain}}{\text{Average loss}}}\right]$$

- Exponential moving averages
  - Moving average with greater weight on most recent data points
  - Exponential moving average with fixed timeframe (EMAF)
  - Exponential moving average with adaptive multiplier (EMAM)
    - Weighting adjusted based on asset volatility
  - Exponential moving average with support and resistance (EMAS)
    - Accounts for support and resistance levels

$$EMA_{\text{Today}} = \left(\text{Value}_{\text{Today}} * \left(\frac{\text{Smoothing}}{1 + \text{Days}}\right)\right) + EMA_{\text{Yesterday}} * \left(1 - \left(\frac{\text{Smoothing}}{1 + \text{Days}}\right)\right)$$

Identifying a target feature

- Average price for the next day
  - Average price calculated by averaging the high and low prices
  - Next day average was then shifted by one day in the dataframe
  - Does not account for inflation or general market trend
- Daily percent change
  - Price difference between one day and the next, as a percentage of the first day
  - Accounts for general market trend and inflation



Both features were considered as prospective features during modeling

Model selection and analysis

- Long short term memory (LSTM), linear regression, support vector machine (SVM), decision tree, gradient boosting, and random forest
- Models chosen are generally unable to account for external factors
  - General economic indicators, geopolitical events, overall market sentiment
  - These factors are not part of the dataset being used
- Model performance judged using performance metrics
  - Mean squared error (MSE)
  - Mean absolute error (MAE)
  - Root mean squared error/deviation (RMSE)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$

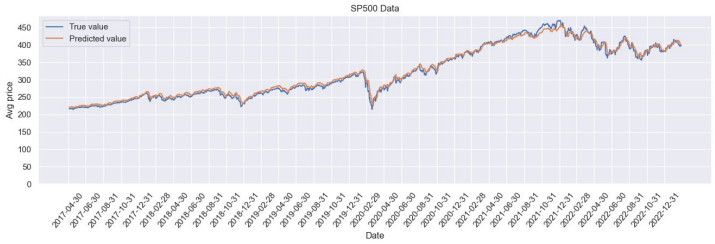
$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n}$$

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}.$$

Long short term memory (LSTM)

- Simple single layer LSTM with 64 nodes
- 20% dropout layer
- 58.72 MSE, 6.09 MAE, 7.66 RMSE

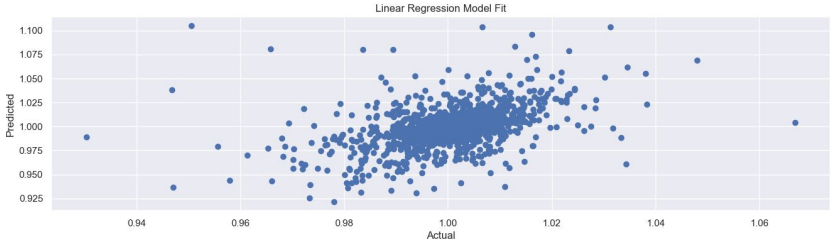
Percentage modeling



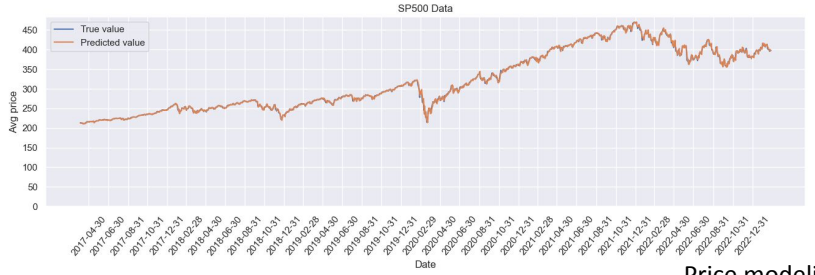
Price modeling

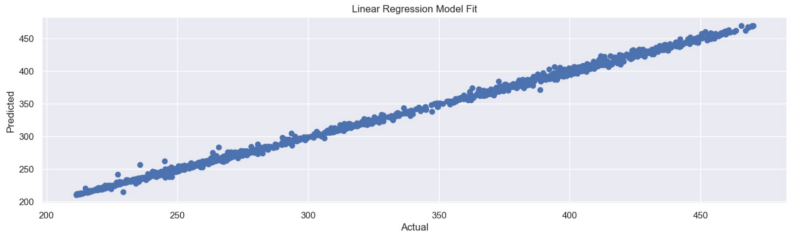
Linear regression

- Much better performance predicting target price over percentage
- Highly accurate, with R2 score of .99
- 7.87 MSE, 1.85 MAE, 2.80 RMSE



Percentage modeling

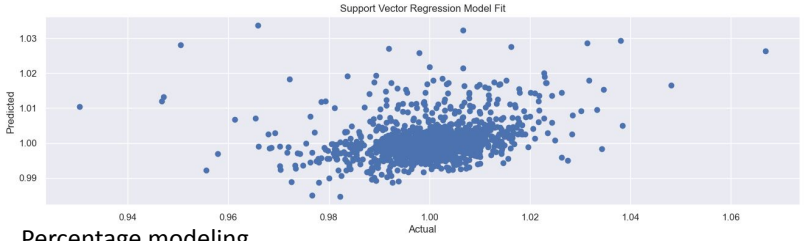




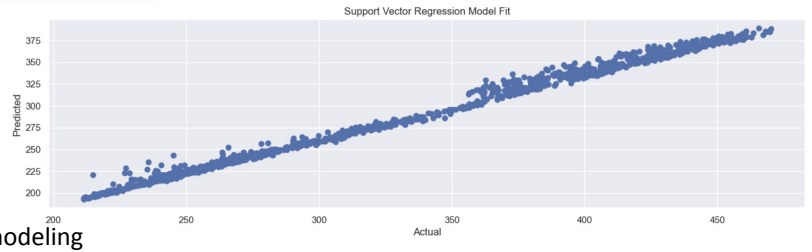
Price modeling

Support vector machine (SVM)

- Much better performance predicting target price over percentage
- Consistently underestimates market performance
- 2345.79 MSE, 44.34 MAE, 48.43 RMSE



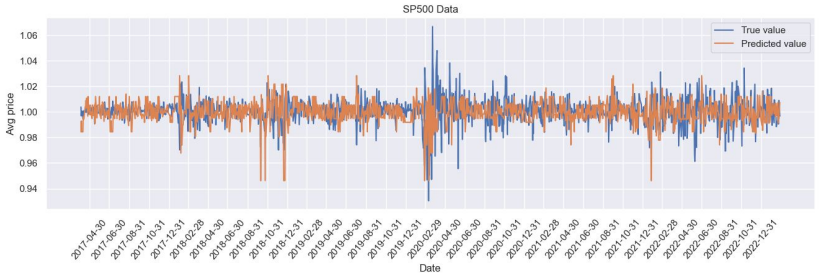
Percentage modeling



Price modeling

Decision tree

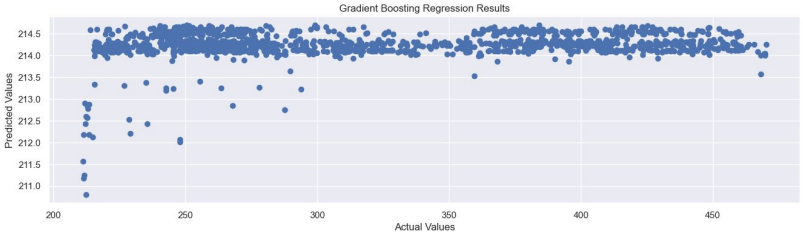
- Did not fit well when target feature was price
- Much better fit for target feature of percent change
    - Performance scores are not comparable across different target features



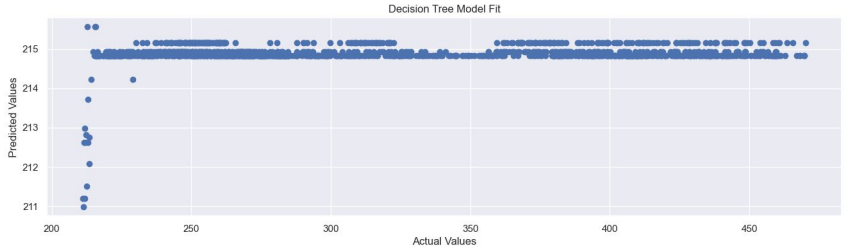
Percentage modeling

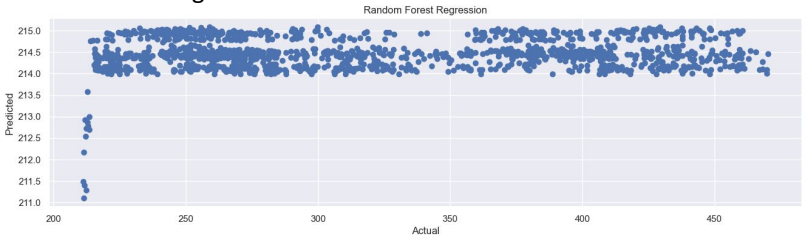
Other models

- Decision tree, gradient boosting, random forest
- Generally did not fit the data well for price target
  - Predicted all target scores within a ~5 value range
- 16000+ MSE, 100+ MAE, 125+ RMSE



Price modeling





Conclusion

Model	Target	MSE	MAE	MAPE	RMSE
LSTM	Price	58.718952	6.094461	0.019569	7.662829
LR	Price	7.866433	1.845412	0.005730	2.804716
SVM	Price	2345.789956	44.343462	0.132029	48.433356
DT	Price	16734.418125	105.162524	0.291265	129.361579
GB	Price	16868.664059	105.799066	0.293362	129.879421
RF	Price	16824.357744	105.579710	0.292627	129.708742
LSTM	Pct	0.000102	0.007081	0.007084	0.010084
LR	Pct	0.000305	0.011236	0.011247	0.017451
SVM	Pct	0.000097	0.006564	0.006584	0.009862
DT	Pct	0.000177	0.009426	0.009424	0.013321
GB	Pct	0.000105	0.006814	0.006811	0.010253
RF	Pct	0.000100	0.006946	0.006946	0.010005

- Linear regression performed the best
  - MSE: 7.87
  - MAE: 1.85
  - RMSE: 2.80
- LSTM performed reasonably well for basic implementation
- SVM consistently underestimated market performance
  - Tracked market movements relatively well otherwise
- Decision tree, gradient boosting and random forest regression models were included for comparison purposes, but performed poorly

Recommendations

- Alternative data sources such as news and social media could be added to include market sentiment to our models. The addition of market sentiment can provide insights to why the S&P 500 increased or deceased in a given time period.
- Hybrid models could be implemented as an addition to linear regression. Example: creating a hybrid model that appends linear regression + support vector or linear regression + decision tree to increase the accuracy of our models.
- Scaling techniques such as standardization or normalization can be used to ensure the model is not biased towards features with a higher range of values.
- Cross-validation techniques such as K-fold cross-validation should be used to test the model on different sets of data to help identify whether the model is generalizing well and assist in selecting the best hyperparameters.

Data source

Kaggle:

https://www.kaggle.com/datasets/gkitchen/s-and-p-500-spy?resource=download