ML Solutions to Stock Price Prediction

GROUP 9, BRIAN HSU & JASON JI

A G E N D A

- Motivation
- Existing Works
- Data Source
- Preprocessing and Analysis
- Models & Results
- Next Steps
- Q&A

MOTIVATOIN

Recent studies found that machine learning models such as the long short-term memory (LSTM) model are good at capturing time series data such as stock price. We want to look deeper at using models we learned in class to predict stock prices and compare their accuracies against LSTM.

Stock Market Crashes Since 1950		
Dates	% Price Decline	Length in Days
8/2/1956–10/22/1957	-21.63	446
12/12/1961–6/26/1962	-27.97	196
2/9/1966–10/7/1966	-22.18	240
11/29/1968–5/26/1970	-36.06	543
1/11/1973–10/3/1974	-48.2	630
11/28/1980–8/12/1982	-27.11	622
8/25/1987–12/4/1987	-33.51	101
3/24/2000–9/21/2001	-36.77	546
1/4/2002–10/9/2002	-33.75	278
10/9/2007–11/20/2008	-51.93	408
1/6/2009–3/9/2009	-27.62	62
2/19/2020–3/23/2020	-33.92	33
Average	**-33.38%**	**342**

EXISTING WORKS

Linear Regression

Stock price prediction using machine learning on least-squares linear regression basis

https://iopscience.iop.org/article/10.1088/1742-6596/1734/1/012058/pdf

a root mean squared error of 0.512.

Random Forest

Stock Closing Price Prediction using Machine Learning Techniques.

RF		
RMSE	**MAPE**	**MBE**
1.29	1.14%	-0.0521
3.40	1.01%	0.0761
1.41	0.93%	-0.0313
1.53	0.75%	-0.0138
0.43	0.8%	-0.0155

LSTM

Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9257950

NO. of Epochs	LSTM RMSE	Time (min)	BI-LSTM RMSE	Time (min)
10	0.0011000	3	0.0007167	8
20	0.0007250	6	0.0006459	15
50	0.0004933	15	0.0004219	40
100	0.0004928	30	0.0004127	70
250	0.0031980	75	0.0003568	200

Our model differentiates by incorporating additional important features :
<u>competitor stock's performance</u>

DATA SOURCE

We will get the company stock dataset (APPLE) from Yahoo Finance using yfinance library with a range from 1980/12/12 to 2022/12/31.

Features include daily high, low, open, close, volume, moving average, index (ex. S&P500), and competitors/supplier's stock stats. Used next-day adjusted price as y to take dividends and splits into account.

10603 samples, 107 attributes.

yahoo!finance

```
            ORCL_Volume        MA_5       MA_21       MA_50       MA_100  \
Date
1980-12-12          NaN         NaN         NaN         NaN          NaN
1980-12-15          NaN         NaN         NaN         NaN          NaN
1980-12-16          NaN         NaN         NaN         NaN          NaN
1980-12-17          NaN         NaN         NaN         NaN          NaN
1980-12-18          NaN         NaN         NaN         NaN          NaN
...                 ...         ...         ...         ...          ...
2022-12-23    4129100.0  132.841998  141.015714  144.131800  150.515600
2022-12-27    4290700.0  132.373999  140.154762  143.964799  150.157800
2022-12-28    3794000.0  131.121999  139.289048  143.637399  149.764699
2022-12-29    3867800.0  129.953999  138.738572  143.354599  149.412100
2022-12-30    5375700.0         NaN         NaN         NaN          NaN

              MA_252
Date
1980-12-12       NaN
1980-12-15       NaN
1980-12-16       NaN
1980-12-17       NaN
1980-12-18       NaN
...              ...
2022-12-23  155.725238
2022-12-27  155.525635
2022-12-28  155.314325
2022-12-29  155.116825
2022-12-30       NaN

[10603 rows x 107 columns]
Size of the DataFrame: (10603, 107)
Number of rows: 10603
Number of columns: 107
```

DATA PREPROCESSING

1.Null Values / Missing Entries

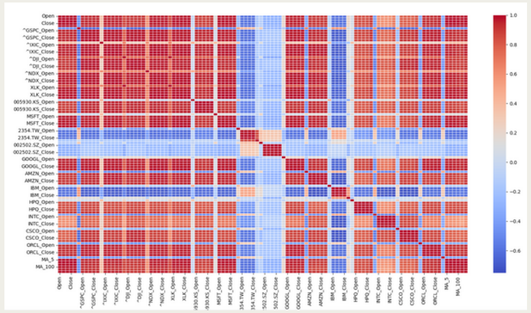
Because competitor company has different IPO time, our initial dataframe contains a substantial amount of null values.

Solution: removal/replace with average.

```
L_Close  ORCL_Adj_Close
     NaN              NaN
     NaN              NaN
```

2. Pearson Correlation

Only kept attributes with high correlation.



3. Colinearity

Removed predictors that are highly correlated with others.

```
correlation_threshold = 0.8
```

```
.shape), k=1).astype(bool))
upper_triangle[column] > correlation_threshold)
```

4. Standard Scaler

Normalizing the data is important for the model performance.

$$X_{new} = \frac{X_i - X_{mean}}{\text{Standard Deviation}}$$

108 features --> 30 features

MODEL EVALUATION METRICS

All continuous numerical features

Mean Absolute Error (MAE)

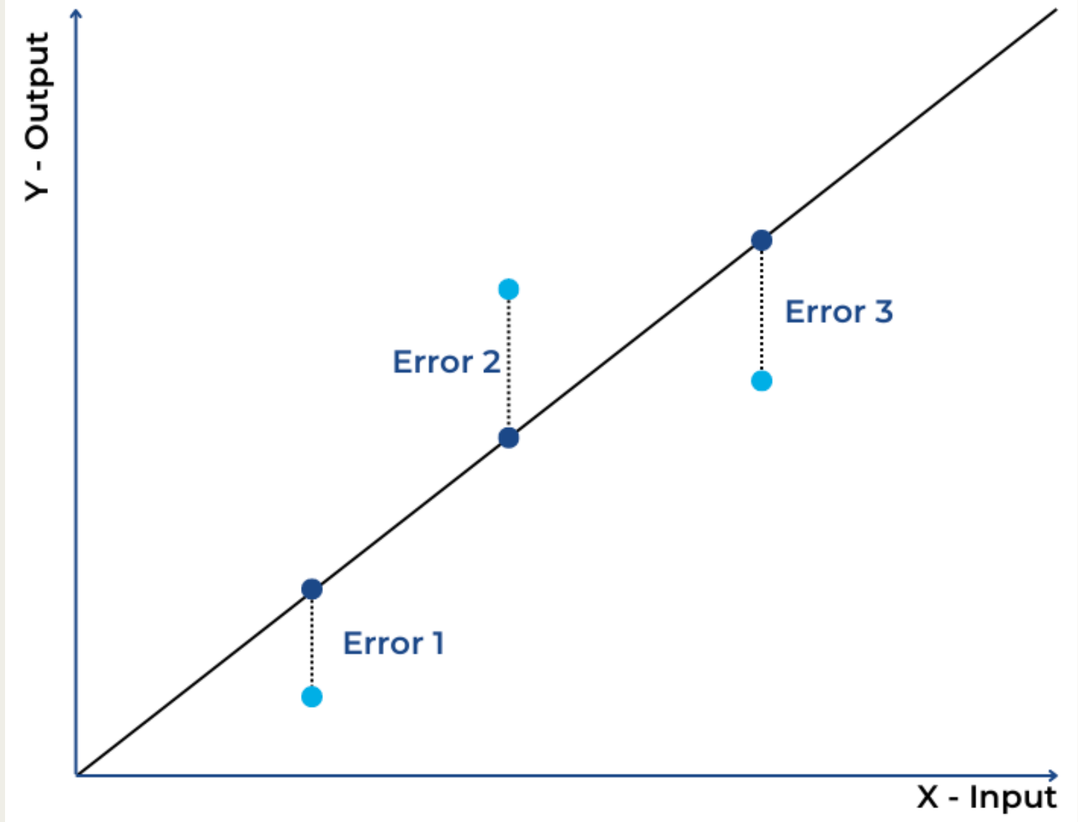
The average of the absolute errors between the predicted values and the actual values.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|,$$

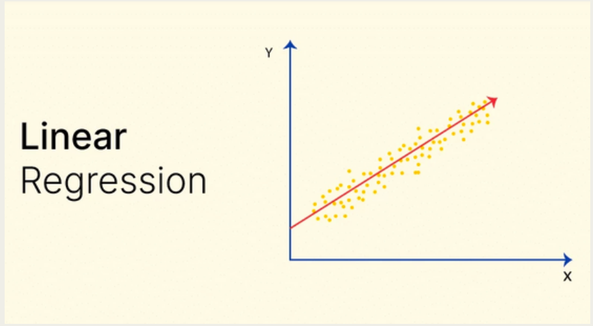
Root Mean Squared Error (RMSE)

The square root of the average of the squared differences between the predicted values and the actual values.
Penalize larger errors more.

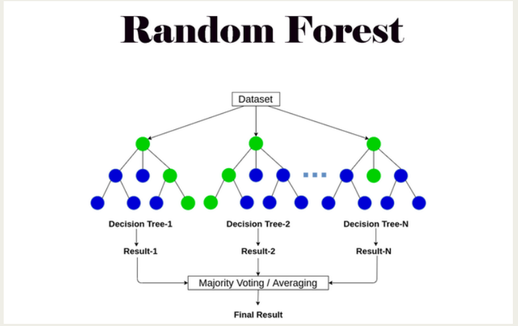
$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}.$$



MODELS

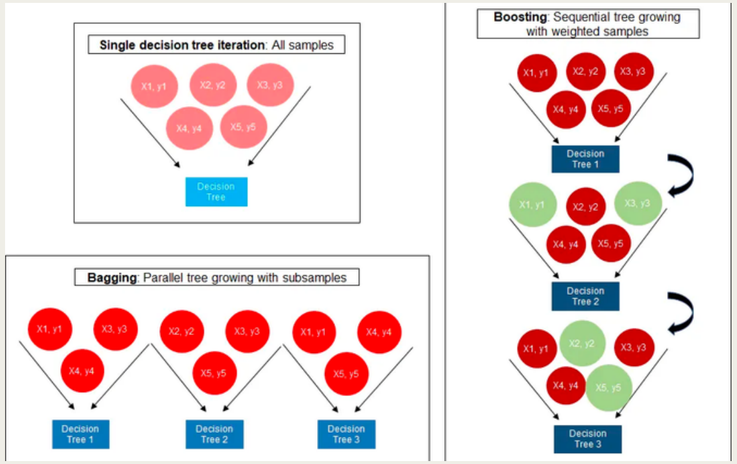


Linear Regression



Random Forest

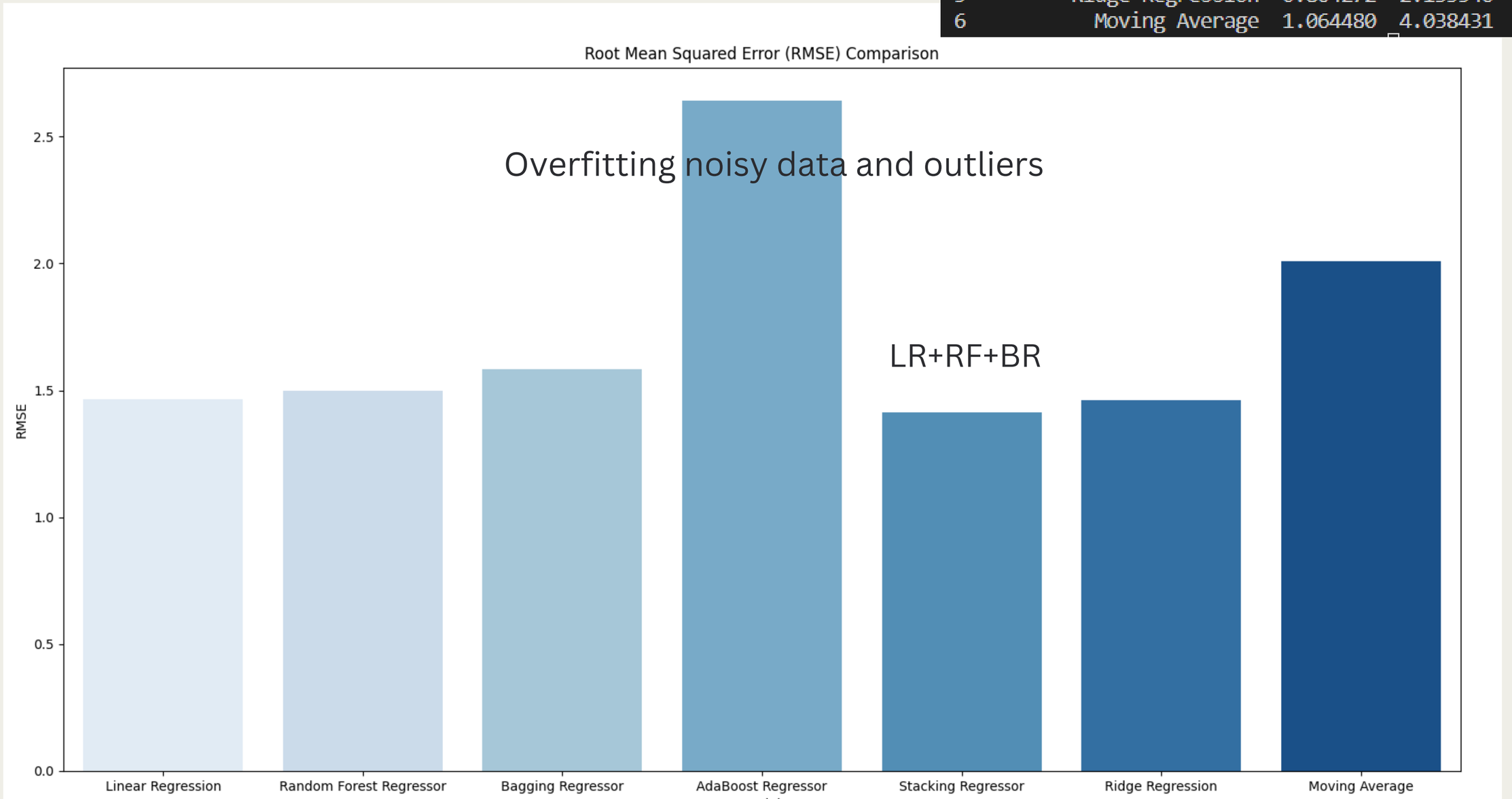
Moving Average

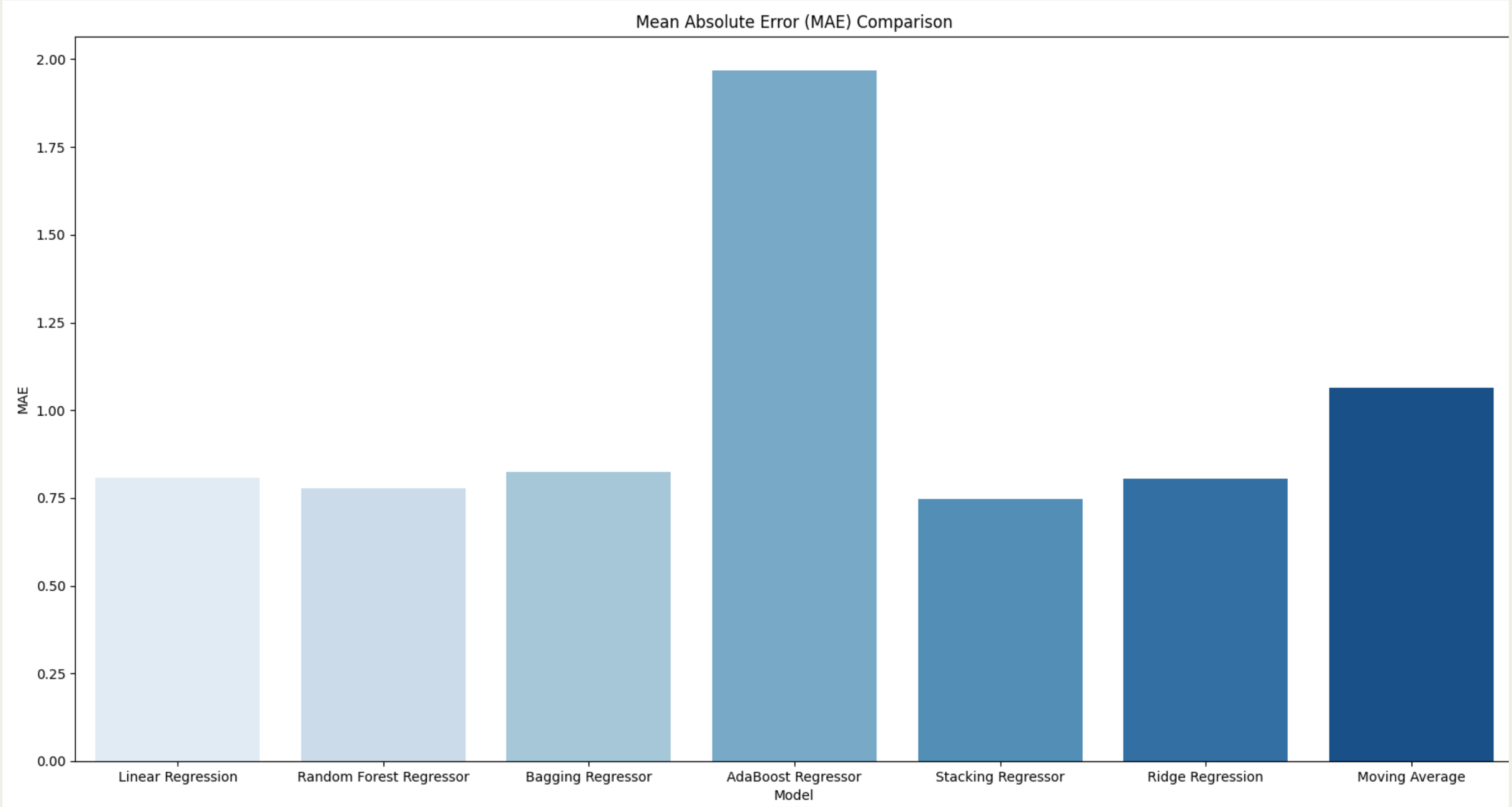


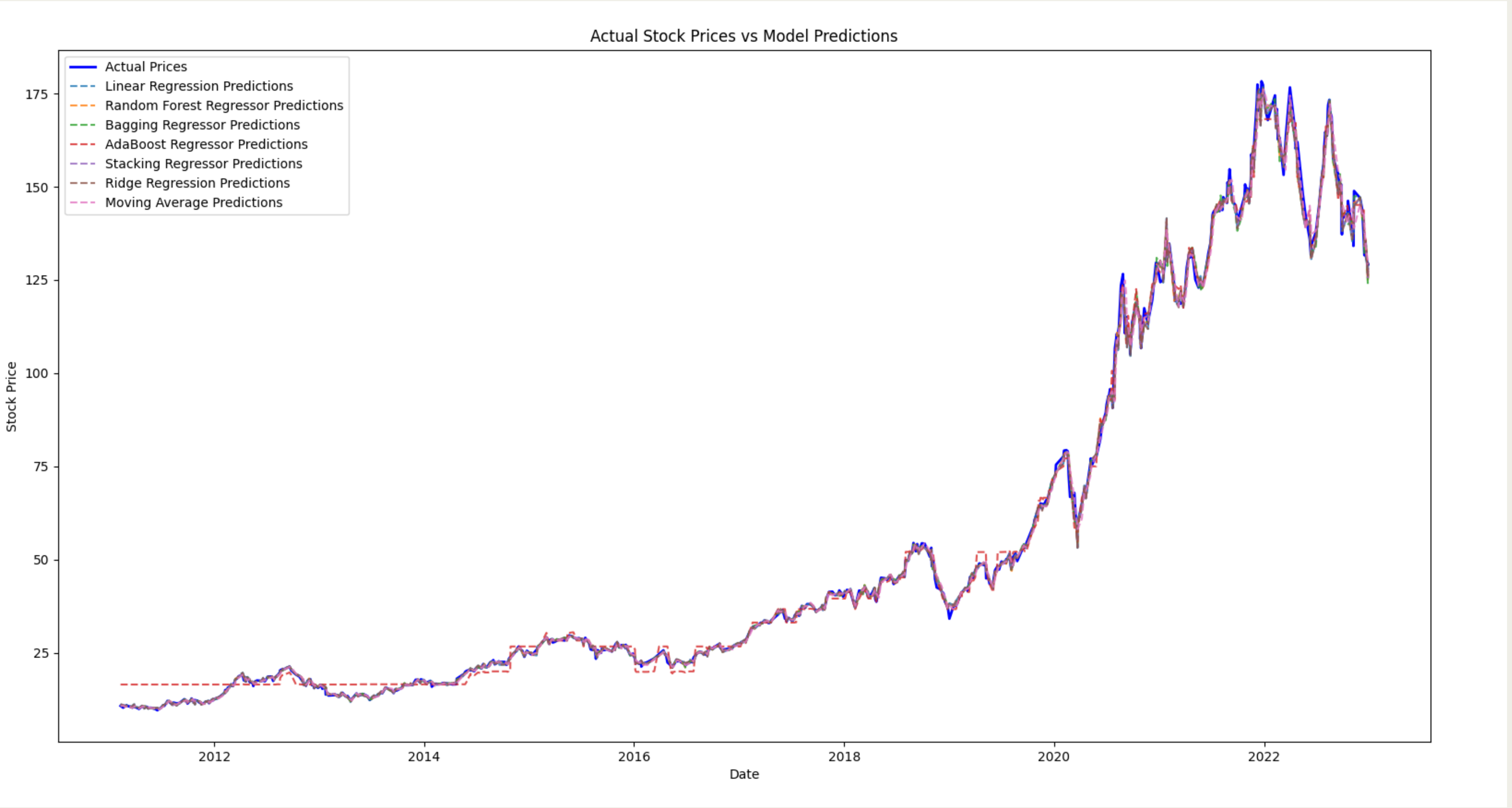
Ensemble Methods

RESULT

	Model	MAE	MSE	RMSE
0	Linear Regression	0.807925	2.141947	1.463539
1	Random Forest Regressor	0.786603	2.285413	1.511758
2	Bagging Regressor	0.842873	2.506118	1.583072
3	AdaBoost Regressor	1.991721	7.325366	2.706541
4	Stacking Regressor	0.756476	2.041915	1.428956
5	Ridge Regression	0.804272	2.135546	1.461351
6	Moving Average	1.064480	4.038431	2.009585



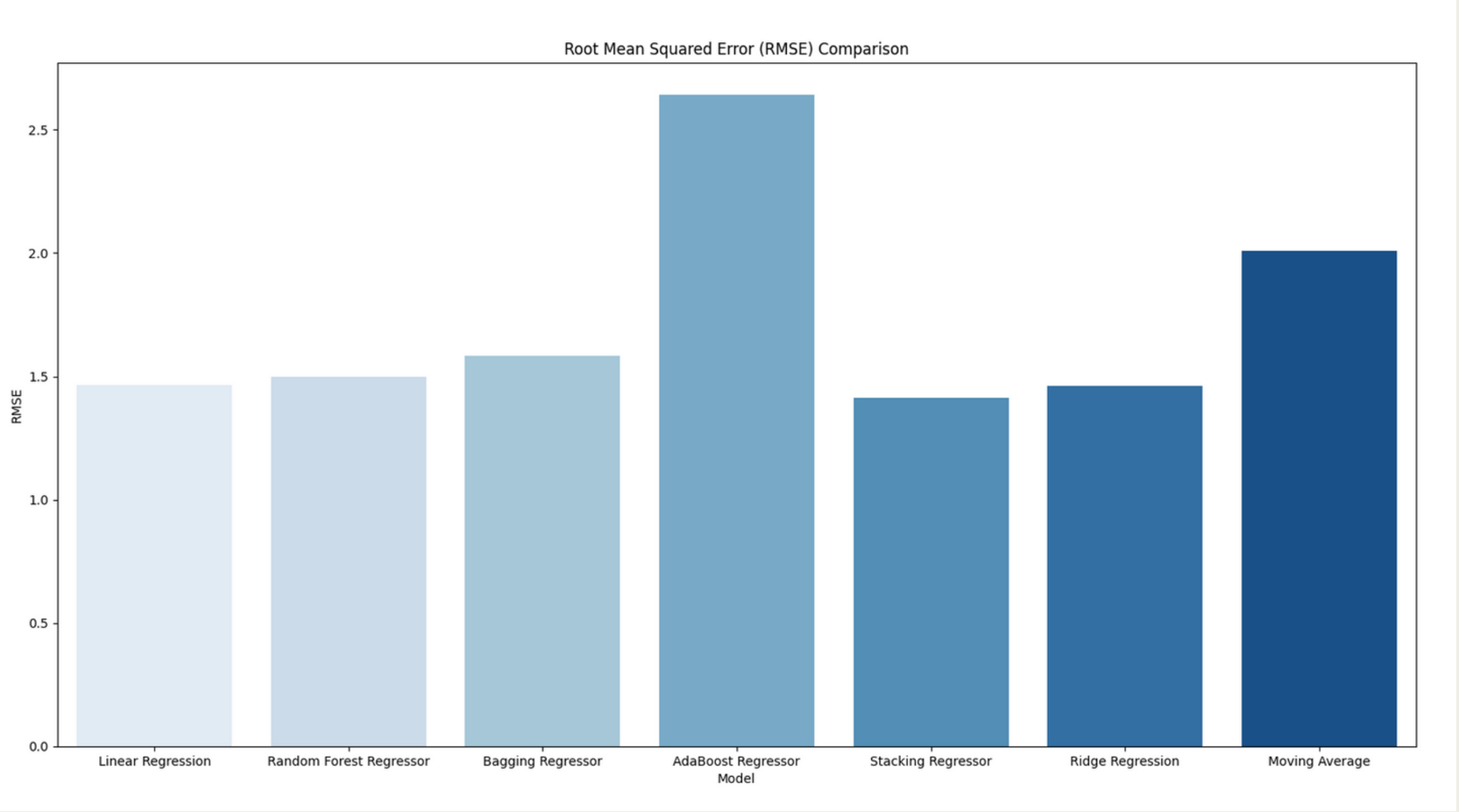

Mean Absolute Error (MAE) Comparison
MAE
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Linear Regression
Random Forest Regressor
Bagging Regressor
AdaBoost Regressor
Stacking Regressor
Ridge Regression
Moving Average
Model


Actual Stock Prices vs Model Predictions
Actual Prices
Linear Regression Predictions
Random Forest Regressor Predictions
Bagging Regressor Predictions
AdaBoost Regressor Predictions
Stacking Regressor Predictions
Ridge Regression Predictions
Moving Average Predictions
Stock Price
175
150
125
100
75
50
25
2012
2014
2016
2018
2020
2022
Date

COMPARISON TO EXISTING WORKS

a root mean squared error of 0.512.

RF		
RMSE	**MAPE**	**MBE**
1.29	1.14%	-0.0521
3.40	1.01%	0.0761
1.41	0.93%	-0.0313
1.53	0.75%	-0.0138
0.43	0.8%	-0.0155



NEXT STEPS

Principle Component Analysis

Further dimensionality reduction. Simpler model (variance reduction).
Improve Performance.

Hyperparameter Tunning

Grid Search.
For the best-performing model.
Ex: Random Forest (Best criterion and max_depth)

LSTM

Using our dataset that contains additional competitor features.

Thank you!

ANY QUESTIONS?