



Stock Price Prediction

APMA 4990 Final Project

By Ho Hin Lee, Kevin Jin,, Tiancheng Li, Yuchen Li

Problem Statement

1

Stock prices can be very volatile and fluctuate daily.



2

Quantitative trading strategies are utilized by many financial institutions to develop trading strategies

3

Can use machine learning models to develop a simple web app that predicts future prices for the end-user to form a trading strategy



»» Dataset

1. On web app, user inputs NYSE stock code (GOOG, AAPL, etc...)



3. Python StockStats package is used to generate additional features from the closing price



2. Python pandas-datareader uses the Investors Exchange (IEX) API to automatically download the original dataset

»» Dataset

	close	volume	price+1	price+5	macd	rsi_12	volume_delta	MA 20	MA 50	Daily Change	Range
date											
2017-01-03	113.8474	28781865	113.7200	116.7488	1.277287	52.196938	-1804400.0	112.612895	110.465704	0.3430	1.368094
2017-01-04	113.7200	21118116	114.2983	117.3761	1.188925	49.671961	-7663749.0	112.951545	110.466010	0.1666	0.656558
2017-01-05	114.2983	22193587	115.5725	116.8860	1.152279	60.167694	1075471.0	113.277945	110.457402	0.6763	0.910283
2017-01-06	115.5725	31751900	116.6311	116.6801	1.212082	74.324962	9558313.0	113.615125	110.462576	1.1076	1.451020
2017-01-09	116.6311	33561948	116.7488	117.6211	1.329570	81.042392	1810048.0	113.951815	110.540802	1.0193	1.263300

$$\text{Moving Averages} = \frac{P_t + P_{t+1} + \dots + P_{t-(n-1)}}{n}$$

$$\text{Daily Change} = \text{Close} - \text{Open}$$

$$\text{Range} = \text{High} - \text{Low}$$

$$\text{MACD} = \text{EMA}_{n=12} - \text{EMA}_{n=26}$$

$$\text{RSI} = 100 - \frac{100}{1 + \frac{\text{Sum of gains from past } n}{\text{Sum of losses from past } n}}$$

$$\text{Volumn Delta} = \text{Volume}_n - \text{Volume}_{n-1}$$

»» Algorithm



01 Download
stock dataset

02 Generate
features

03 Choose model

04 Train and
validate model

05 Predict with
test data for 1
and 5 days
future
prediction

06 Output buy/sell decision
based on predicted price
compared to last price
point on the dataset

Predicted Price > $y[-1]$ ---BUY
Predicted Price < $y[-1]$ ---SELL



>>> Method



Models Evaluated:

- Random Forest Regression
- Recurrent Neural Network (LSTM)
- ARIMA Model

Used 1 test stock (AAPL) from March 2016 to May 1, 2018 data

Evaluated model for forecasting for 1, 5, and 10 days (Only ARIMA) into the future

- Model performance based on RMSE value for k-fold time series cross validation

Results

»» Models



Recurrent Neural Network (LSTM)

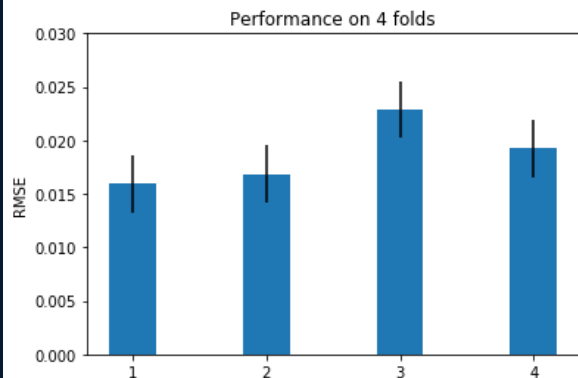
- Uses 90 epochs for training on a 10 layer model with 256 neurons
- Obtained good RMSE values for 1-day prediction.
- RMSE for 5-day prediction was slightly higher and 4 fold CV had higher variance even though the final recommendation was mostly correct in the test dataset
- Computationally slow

»» Models

Recurrent Neural Network

1 Day Results

Mean RMSE: 0.0187318399269
Standard Deviation: 0.00267976218822



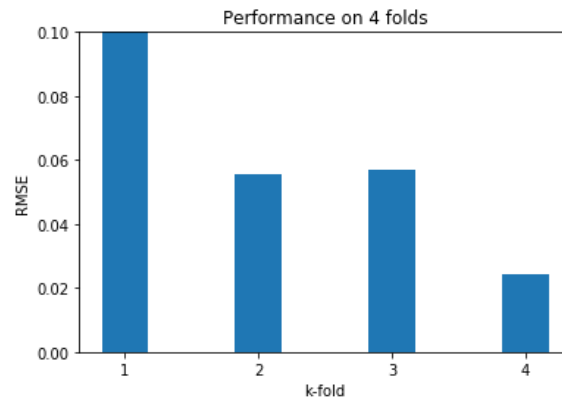
[[171.21844482]]

Based on previous value of: 169.1

Buy

5 Day Results

Mean RMSE: 0.0659452687027
Standard Deviation: 0.0376450931181



[[173.20153809]]

Based on previous value of: 169.1

Buy

»» Models

Random Forests

- Chose the best number of trees for the model by using GridSearchCV
- RMSE was similar to other models for 1 and 5 day forecasts in 4 fold CV.
- Predicted Value did not make sense, so web app cannot make accurate prediction

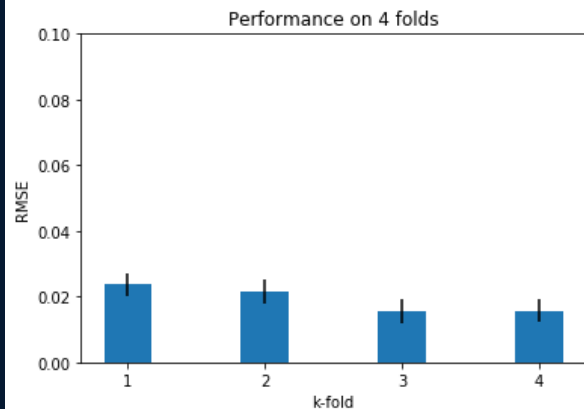
```
CV_rf.best_params_  
{ 'max_depth': 7, 'max_features': 'auto', 'n_estimators': 12 }
```

»» Models

Random Forests

Results: 1 Day

Mean RMSE: 0.0191139666001
Standard Deviation: 0.00357484590501

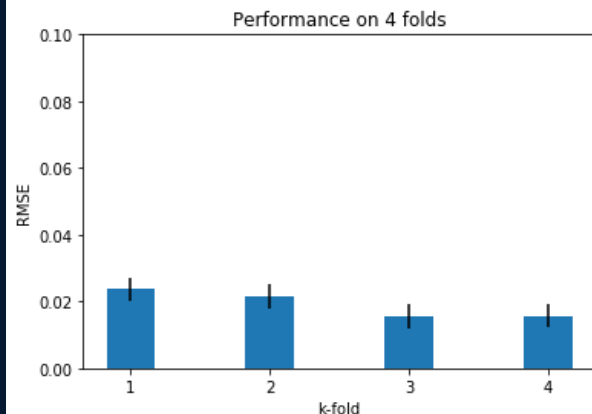


```
y_predict = (model.predict(X_test))*((y_max-y_min)+y_min)  
print(y_predict)
```

```
[ 162.73082379]
```

5 Day

Mean RMSE: 0.0191139666001
Standard Deviation: 0.00357484590501



```
y_predict = (model.predict(X_test))*((y_max-y_min)+y_min)  
print(y_predict[-1])
```

```
152.590944665
```

»» Models

ARIMA

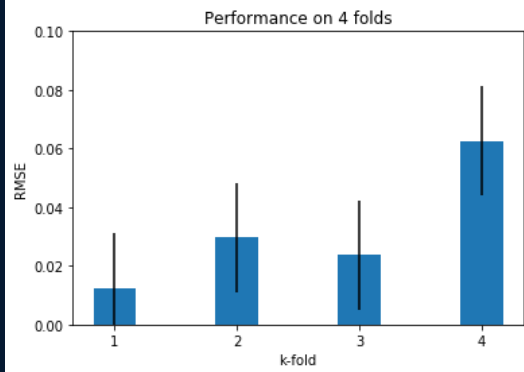
- Autoregressive integrated Moving Average model
- Uses only past closing price to forecast future prices
- Model uses 3 parameters: p , d , q
 - p : Number of time lags
 - d : Differencing
 - q : Moving Average order
- RMSE similar to LSTM and Random Forests for 1 day and 5 day, but performed much better for higher time steps.
- Drawback: Model cannot use any other features other than the previous prices, so no feature selection and optimization other than the 3 parameters can be performed

»» Models

ARIMA

Results: 1 Day

Mean RMSE: 0.0320930258521
Standard Deviation: 0.0187016710195

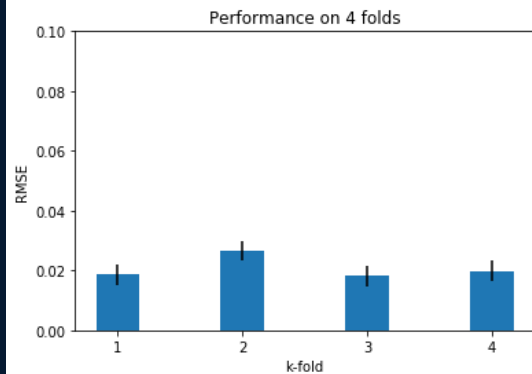


174.644665477

Based on previous value of: 169.1
Buy

5 Days

Mean RMSE: 0.0207892073386
Standard Deviation: 0.00340893242366

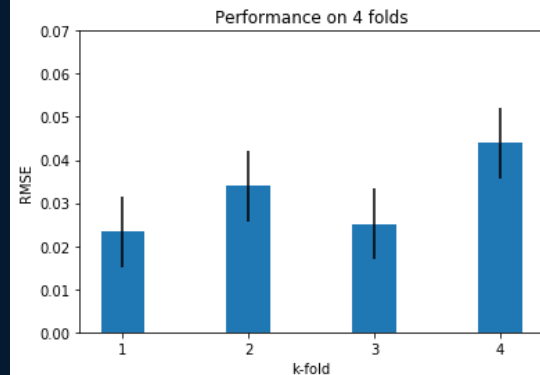


172.420250077

Based on previous value of: 169.1
Buy

10 Days

Mean RMSE: 0.031592558121
Standard Deviation: 0.00818385225831



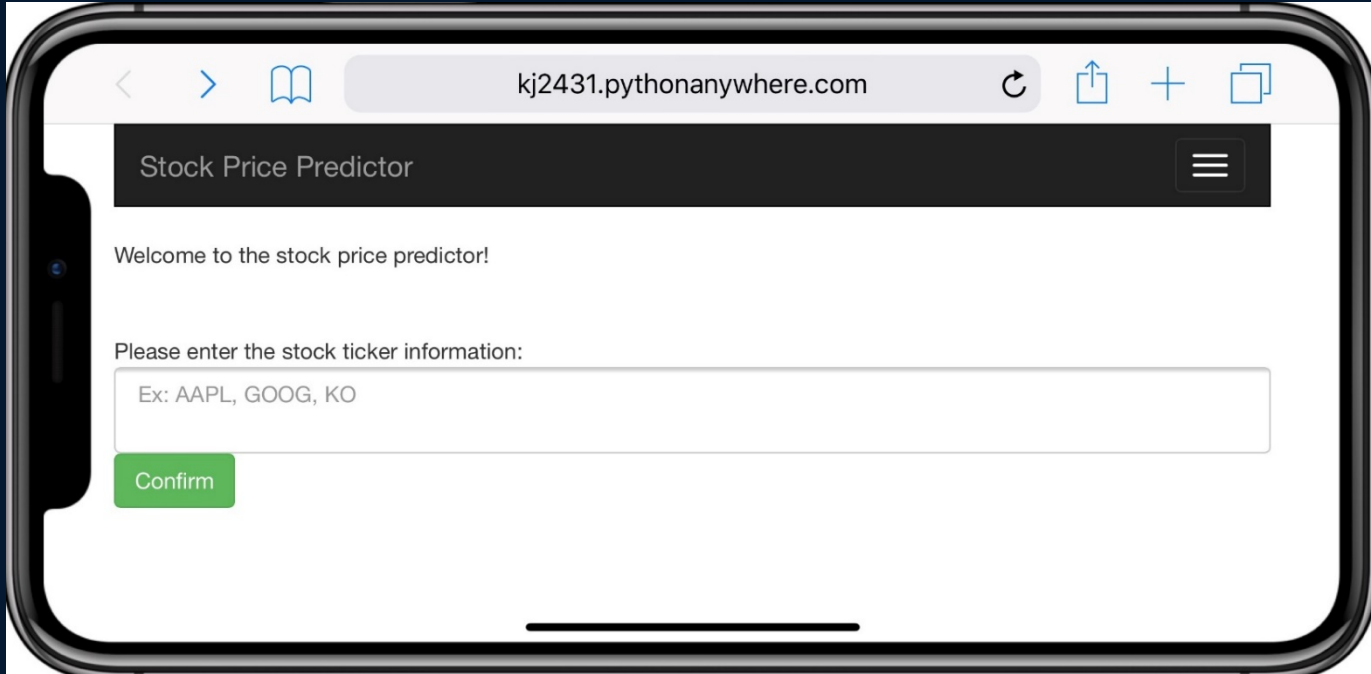
174.097563051

Based on previous value of: 169.1
Buy

»» *Final Model Selection*

- ARIMA for 10 day forecast
- RNN for 1 and 5 day forecast

»» Web app



Demo with web-app: kj2431.pythonanywhere.com

- Backup: <http://35.196.7.153/>

Thanks for listening!

The last day's (2018-05-09) closing price for GOOG is: 1053.91

Using the RNN model:

The next day prediction is: 1065.67

Recommendation: Buy

The 5-day prediction is: 1067.16

Recommendation: Buy

Using the ARIMA model for 10 day prediction:

1079.6600918184909

Recommendation: Buy

The last day's (2018-05-09) closing price for KO is: 41.81

Using the RNN model:

The next day prediction is: 43.03

Recommendation: Buy

The 5-day prediction is: 43.6

Recommendation: Buy

Using the ARIMA model for 10 day prediction:

42.59

Recommendation: Buy

The last day's (2018-05-09) closing price for BABA is: 196.31

Using the RNN model:

The next day prediction is: 183.9

Recommendation: Sell

The 5-day prediction is: 183.04

Recommendation: Sell

Using the ARIMA model for 10 day prediction:

194.23747082608932

Recommendation: Sell