A presentation by

THE SHORT-TERM CAPITAL GAINERS

On stock market prediction

INTRODUCTION & MOTIVATION

SPDR S&P 500 Trust ETF

STATE STREET GLOBAL ADVISORS.
SPDR

RELATED WORK

- *Short-term stock market price trend prediction using a comprehensive learning system*, J. Shen & O. Shafiq (2020). Journal of Big Data
- *How to apply Monte Carlo simulation to forecast Stock prices using Python*, I. Valchanov (2017)
- *ARIMA Model – Complete Guide to Time Series Forecasting in Python*, Prabhakaran, S. (2021)

METHODS

Brownian motion with drift, ARIMA, Logistic Regression, LSTM

BROWNIAN MOTION

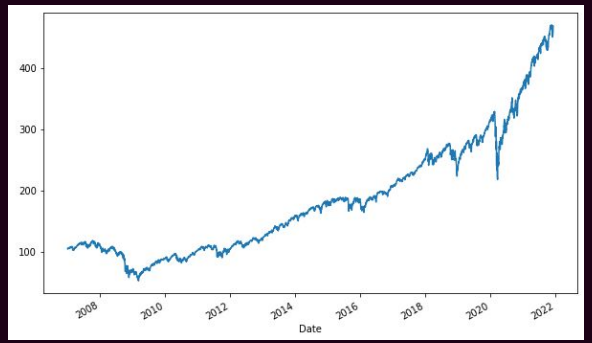
WITH DRIFT

WHAT IT IS

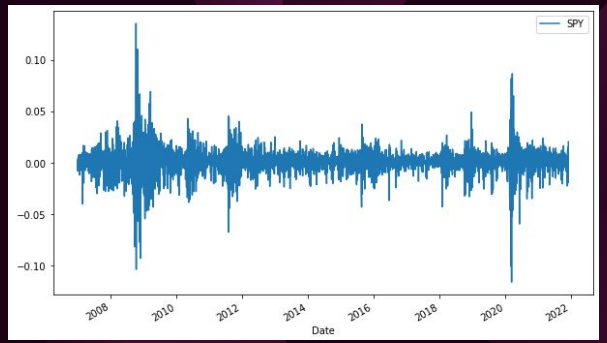
Instead of looking at prices,

look at the changes in price.

From this



To this



THE EQUATIONS

We define "drift" as $\mu - h \cdot \sigma^2$

We then simulate a change in price by $e^{drift + \sigma \cdot Z}$

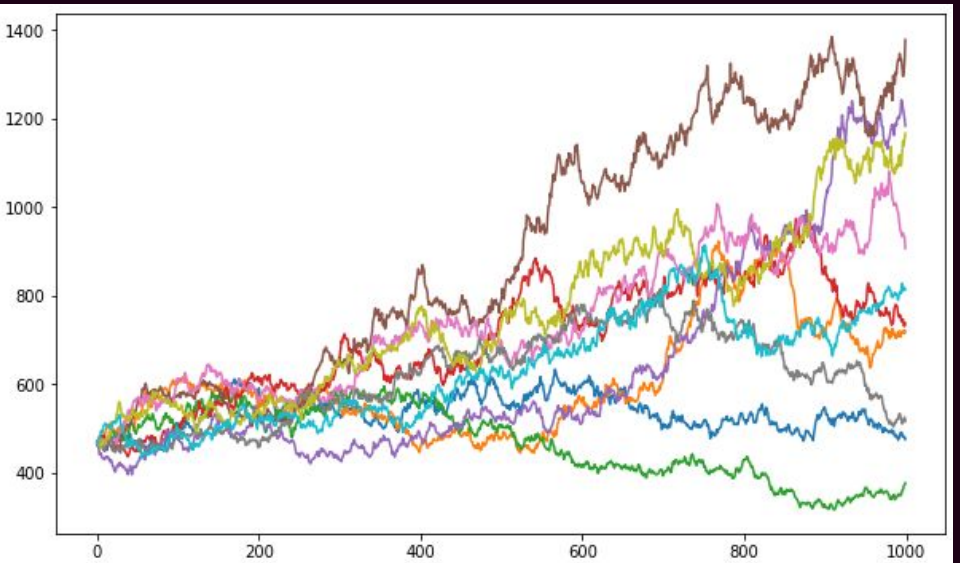
where Z is a standard normal random variable

But what is h ?

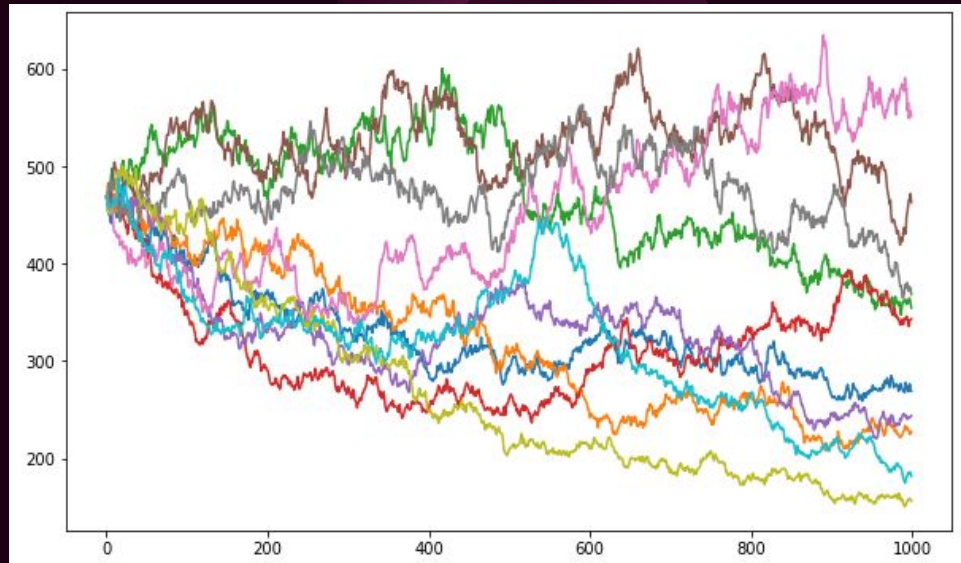
THE HYPERPARAMETER h

$$e^{\mu - h \cdot \sigma^2 + \sigma \cdot Z}$$

h = 0.05



h = 1



OPTIMIZING h

Reject the status quo

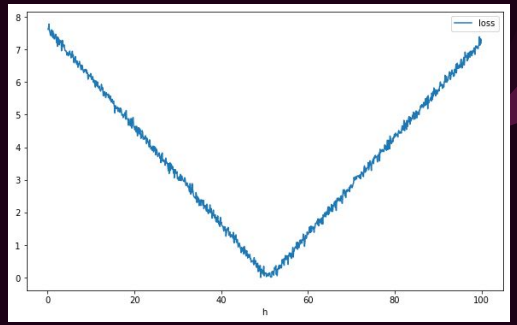
Find the h that minimizes loss

Pick the best h

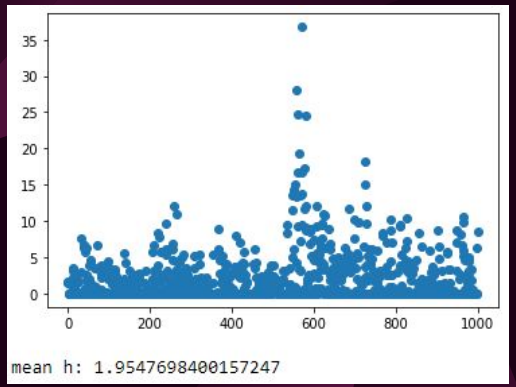
Ask the question, "why must I use ½ ?"

$$h = \frac{1}{2}$$

For the loss function |predicted - actual|

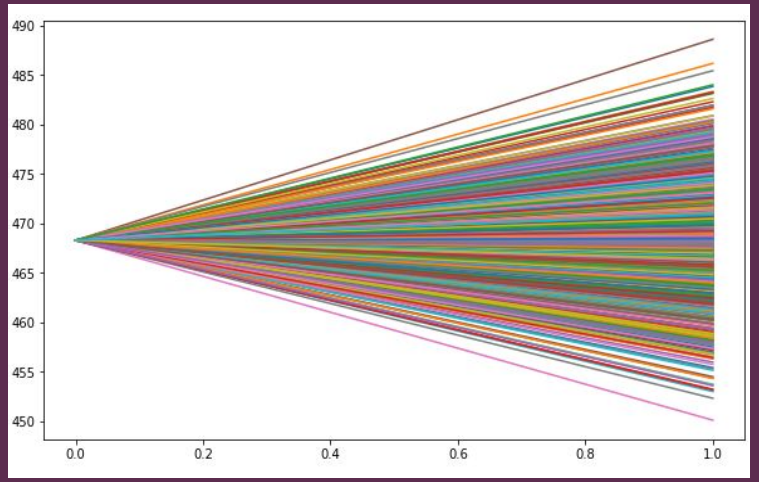


Backtest and take the average optimal h



PREDICTING PRICE

Using the h we found, predict tomorrow's price by simulation

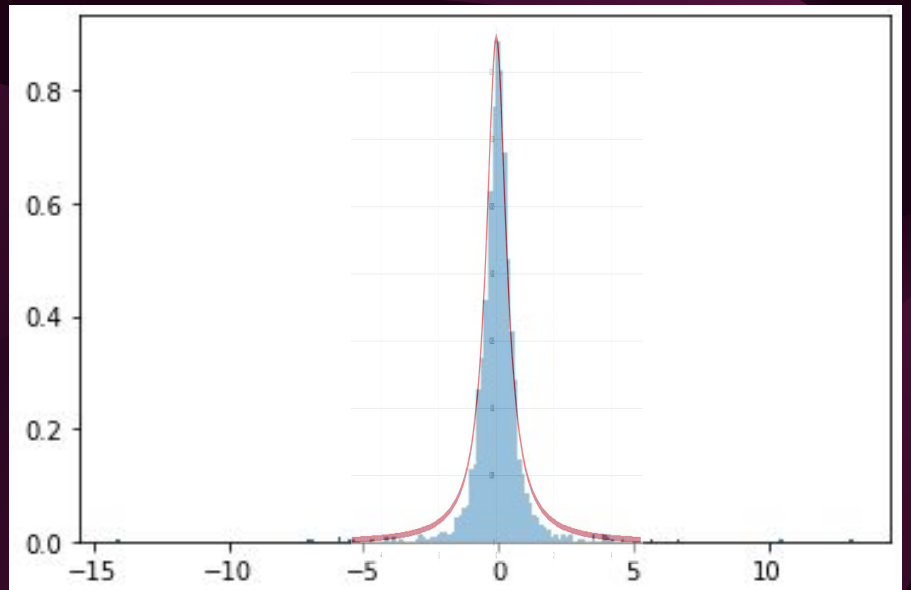


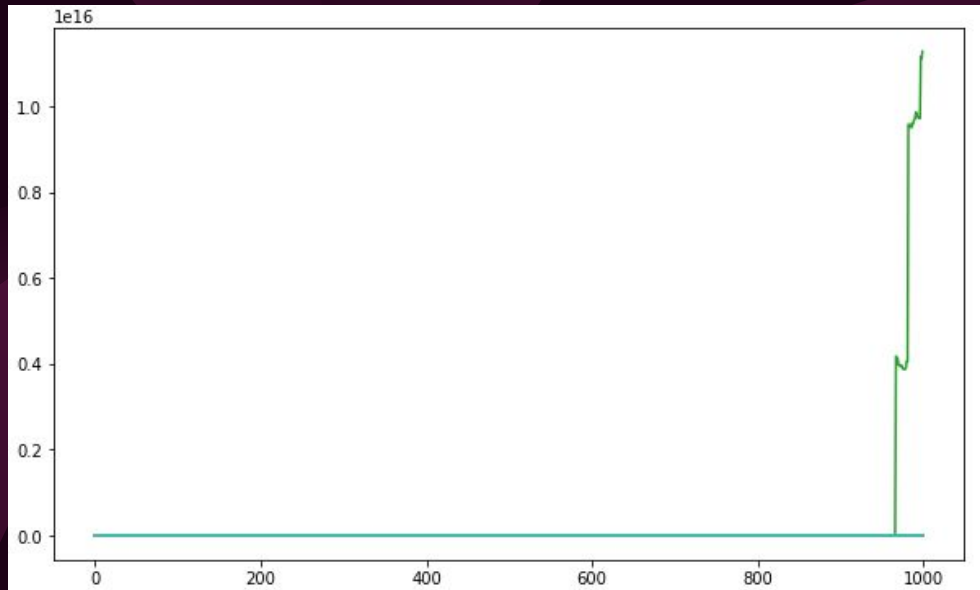
WHAT DOES MANDELBROT THINK?

WHAT WOULD MANDELBROT PROPOSE?

$$e^{drift+\sigma \cdot Z}$$

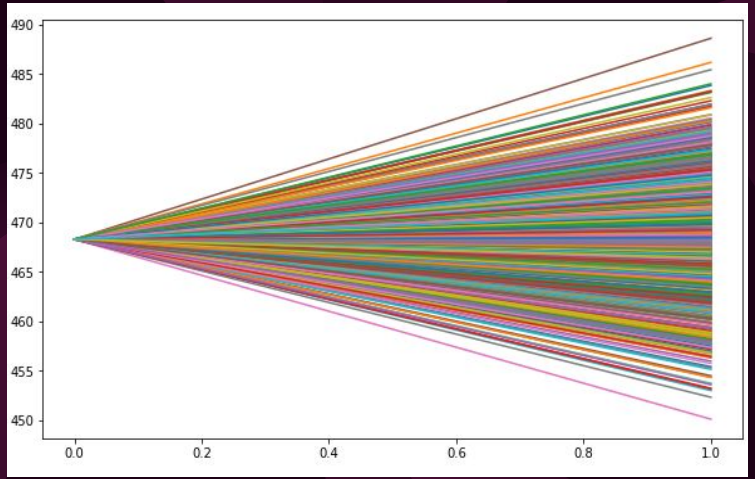
He would insist on using a Cauchy distribution for Z





Follow-ups

$$e^{drift+\sigma \cdot Z}$$



ARIMA

AutoRegressive Integrated Moving Average

AR

Uses the relationship between an observation and some predefined number of lagged observations.

p is the number of lag observations

I

Subtracts an observation from the previous **d** observations to make the data stationary.

d is the degree of differencing

MA

Uses the relationship between an observation and its lag errors.

q is the size/width of the moving average window

ARIMA

- I → Differencing
- AR → $Y_t = \alpha + \gamma_1 Y_{t-1} + \gamma_2 Y_{t-2} + ... + \gamma_p Y_{t-p} + \epsilon_t$
- MA → $Y_t = \beta + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + ... + \phi_q \epsilon_{t-q}$
- ARIMA

$$Y_t = \kappa + \gamma_1 Y_{t-1} + \gamma_2 Y_{t-2} + ... + \gamma_p Y_{t-p} + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + ... + \phi_q \epsilon_{t-q}$$

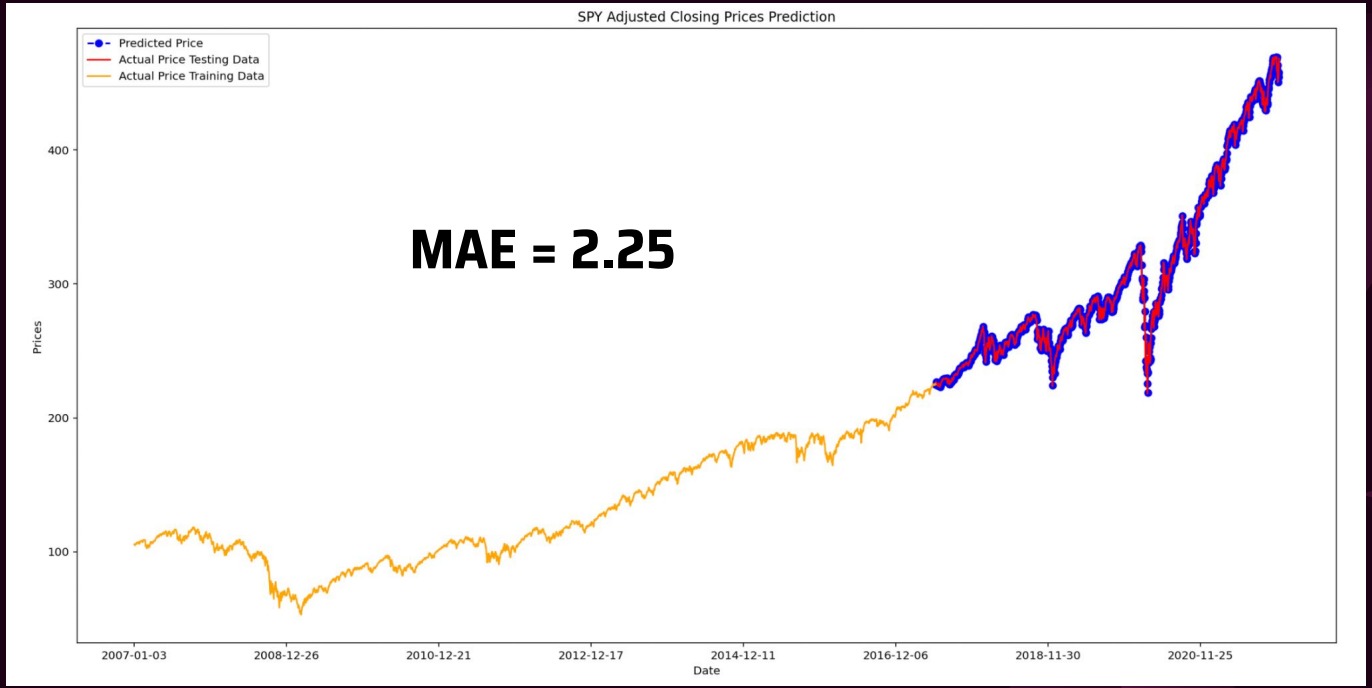
AR MA

p is the number of lag observations

d is the degree of differencing

q is the size/width of the moving average window

ARIMA RESULT



LINEAR REGRESSION

By: Jacqueline Gonzalez

RELATED WORK:

- https://www.akademiabaru.com/doc/ARBMSV14_N1_P35_41.pdf
- https://www.alpharithms.com/predicting-stock-prices-with-linear-regression-214618/
- https://byuistats.github.io/BYUI_M221_Book/Lesson22.html

METHODS

1. Building a data set

DataSet

```
ticker = 'SPY'
data = wb.DataReader(ticker, data_source='yahoo', start='2007-1-1')

data.reset_index(inplace=True)
```

```
#Summary Statistics
data.head()
```

	Date	High	Low	Open	Close	Volume	Adj Close
0	2007-01-03	142.860001	140.570007	142.250000	141.369995	94807600.0	105.446762
1	2007-01-04	142.050003	140.610001	141.229996	141.669998	69620600.0	105.670479
2	2007-01-05	141.399994	140.380005	141.330002	140.539993	76645300.0	104.827637
3	2007-01-08	141.410004	140.250000	140.820007	141.190002	71655000.0	105.312469
4	2007-01-09	141.600006	140.399994	141.309998	141.070007	75680100.0	105.222977

METHODS

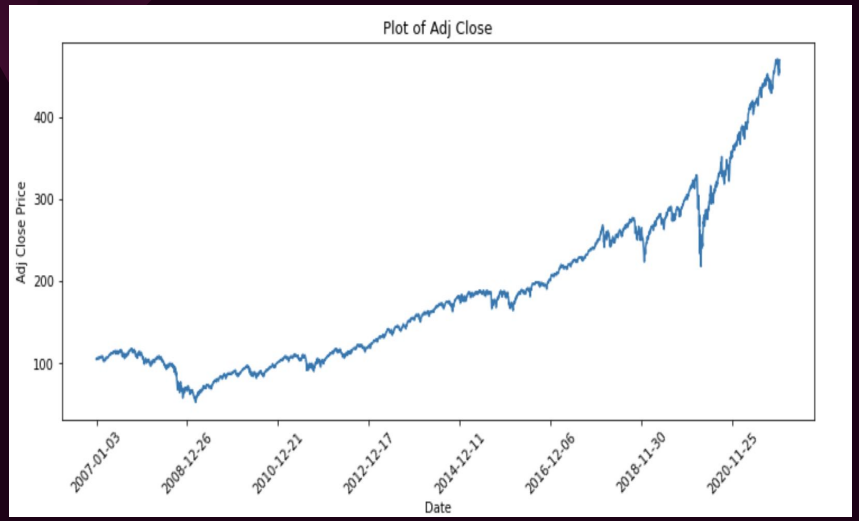
Linear Regression is a linear model

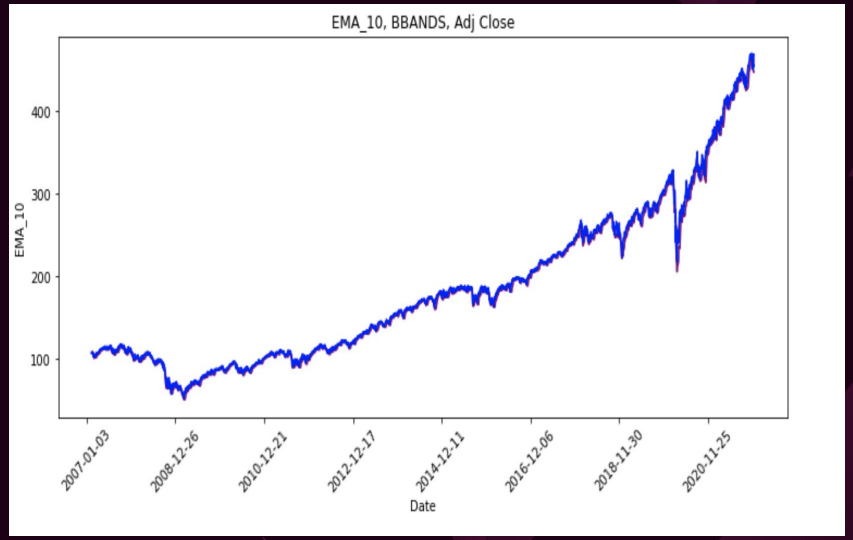
$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \ldots + b_p X_p ,$$

Price = b_0+ b_1EMA + b_2BBANDS

METHODS

2. Adding Technical Indicators

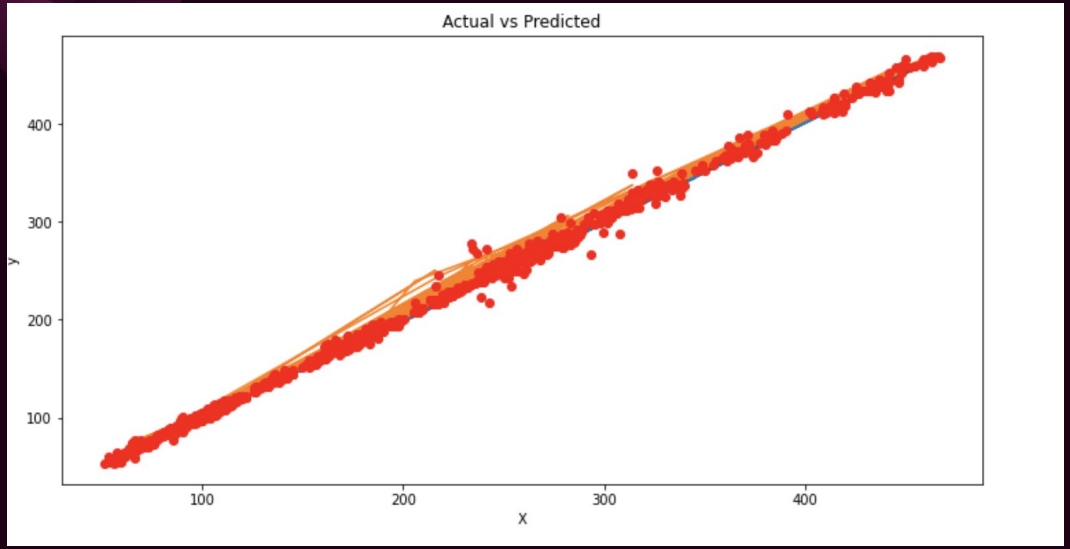




METHODS

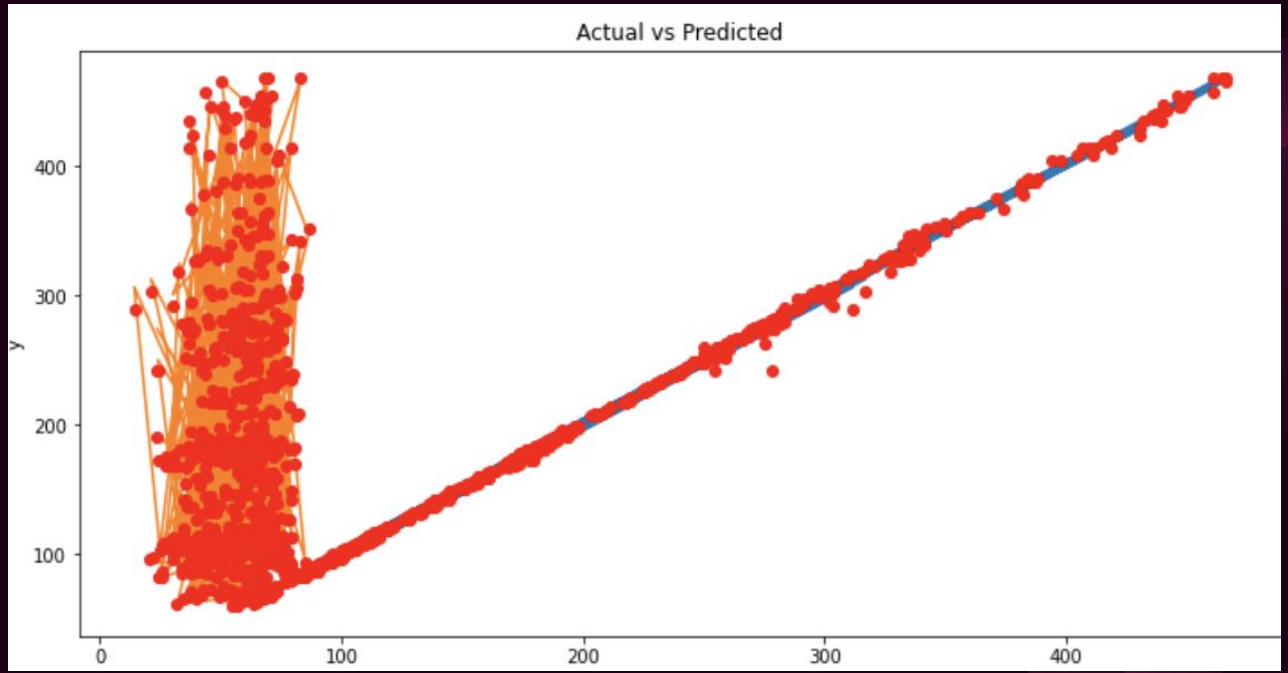
3. Training the Model

RESULTS



Price = 0.8818*EMA + 0.12568*BBANDS - .4180714

PREVIOUS PROBLEMS

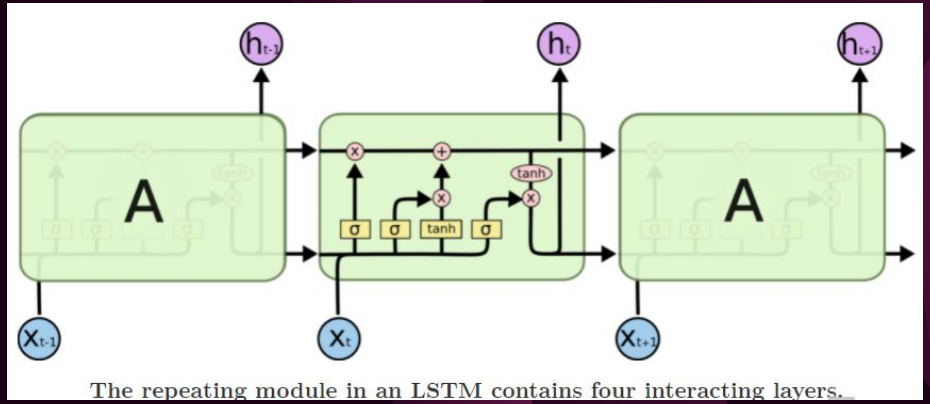


LONG SHORT-TERM MEMORY MODEL (LSTM)

RELATED WORK

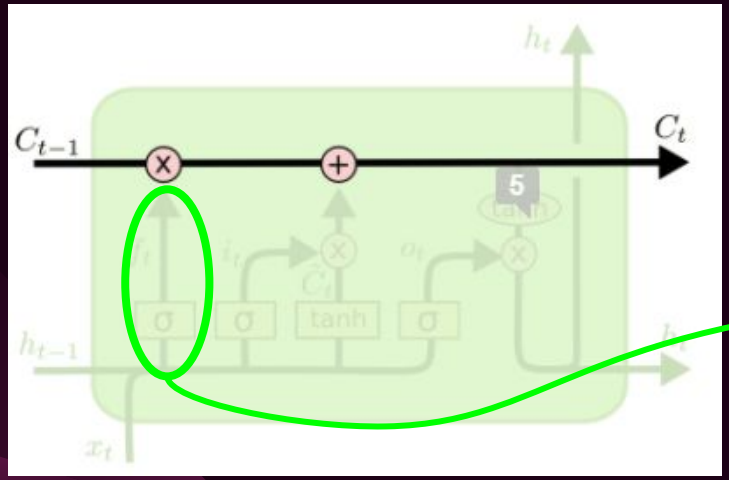
- *Loukas, S. (2021, October 8). LSTM time-series forecasting: Predicting stock prices using an LSTM model. Medium.*
- *Olah, C. (2015). Understanding LSTM networks. Understanding LSTM Networks -- colah's blog.*

LSTM - INTUITION

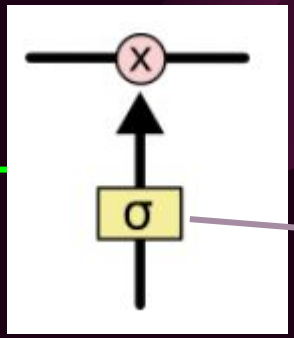


The repeating module in an LSTM contains four interacting layers.

Cell State - One Layer



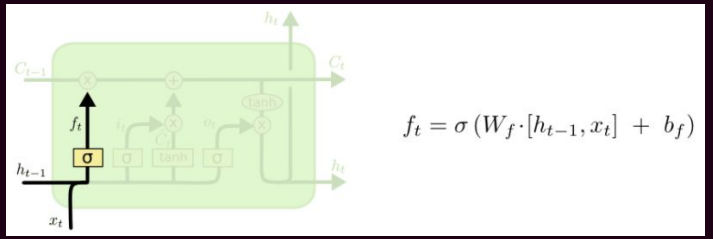
The Gates



Sigmoid layer

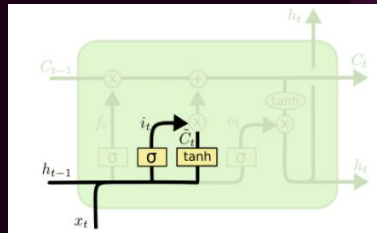
LSTM - STEPS

Step 1 - Decide what info to remove



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

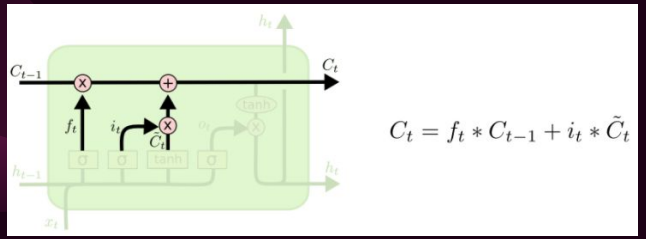
Step 2 - Decide what info to add



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

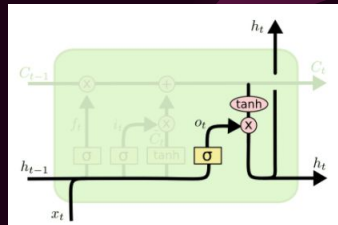
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Step 3 - Update old cell state to new



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Step 4 - Decide output



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

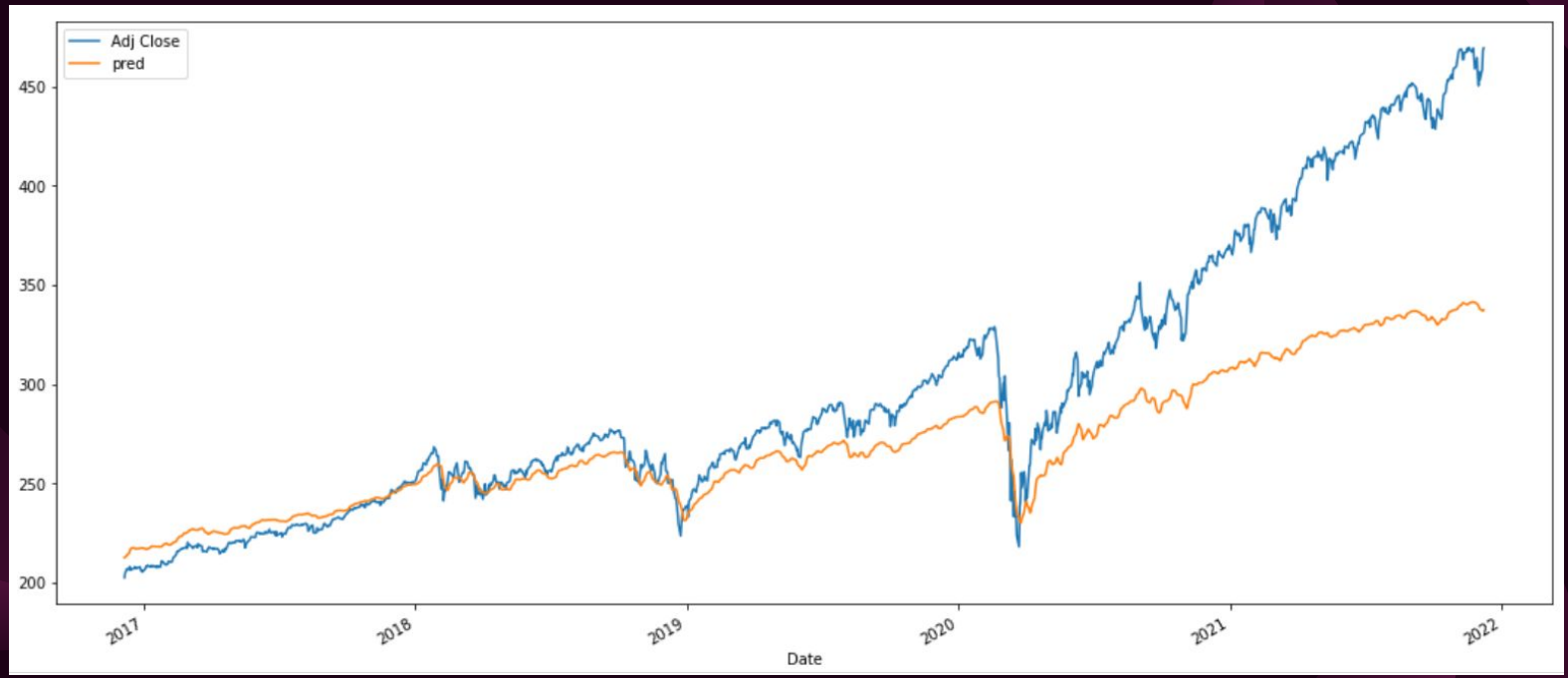
$$h_t = o_t * \tanh\left(C_t\right)$$

BUILDING THE MODEL

- Data preparation & cleaning (SPY)
  - Normalize inputs with MinMaxScaler
- Train LSTM model using years 2007-2017
  - Hyperparameter setting: 50 neurons and 4 hidden layers
- Test data on years 2018-2021
- Make Predictions

RESULTS

SPY Actual Price vs Predicted Price 2017-2021



RESULTS

Compare Absolute Error and Tomorrow's Adj. Closing Price

RESULTS

Closing price: 469.55

	Brownian	ARIMA	Linear Regression	LSTM*
Tomorrow's Adj. Closing Price (12/08/21)	$468.38	$468.97	$468.82	$337.37
Absolute error	$1.14	$0.55	$0.70	$132.15

***LSTM is not scaled to fit data.*

Thank you :)

DISCUSSION

DISCUSSION

- ARIMA works well for one stock market at a time (to apply to different markets, need to retrain the hyperparameters)
- What did you learn? What could you do better? (What would you have done next if you had more time)?..... Why do you think it didn't work if it didn't?
- If everything worked perfectly, what next steps would you suggest for follow-up work.

HELLO!

I am Ethan Chuang

I am here because I love to give presentations.

You can find me at @username