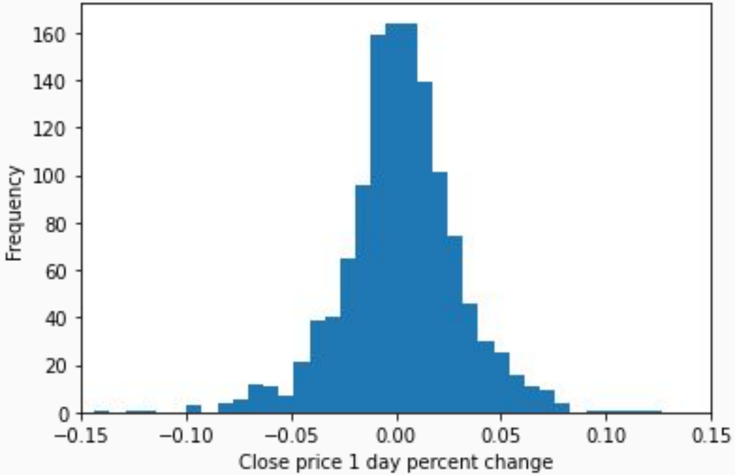
NVIDIA

CONTENTS

BACKGROUND

NVIDIA PRICE HISTORY

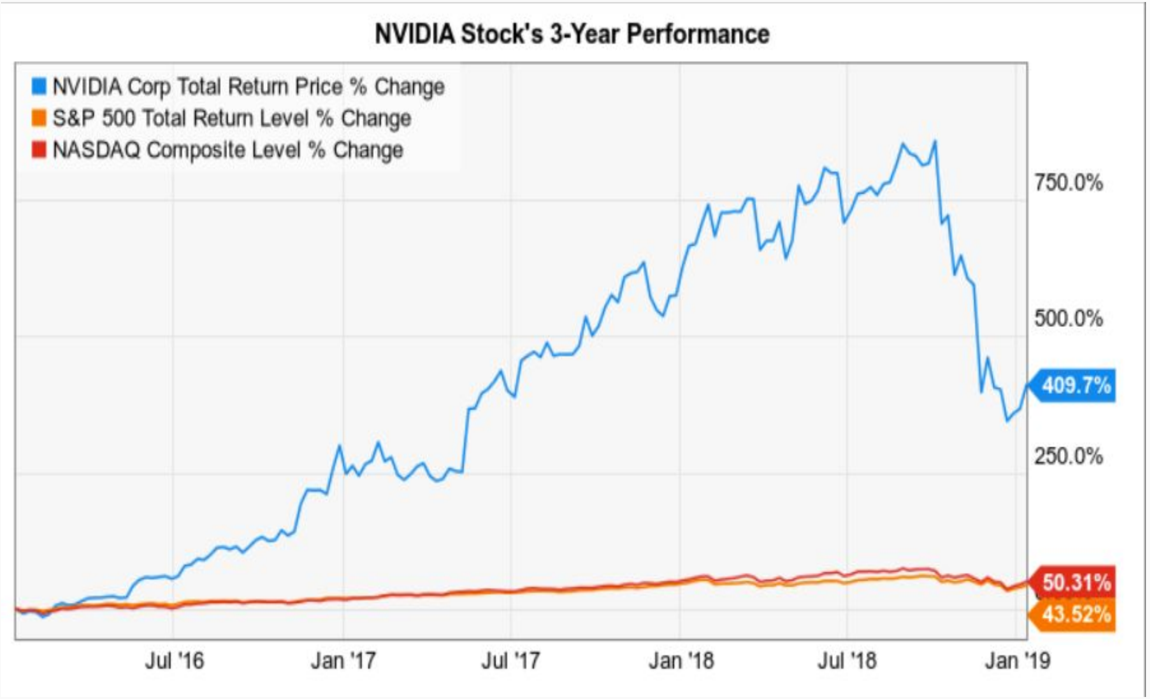


MAJOR EVENTS:

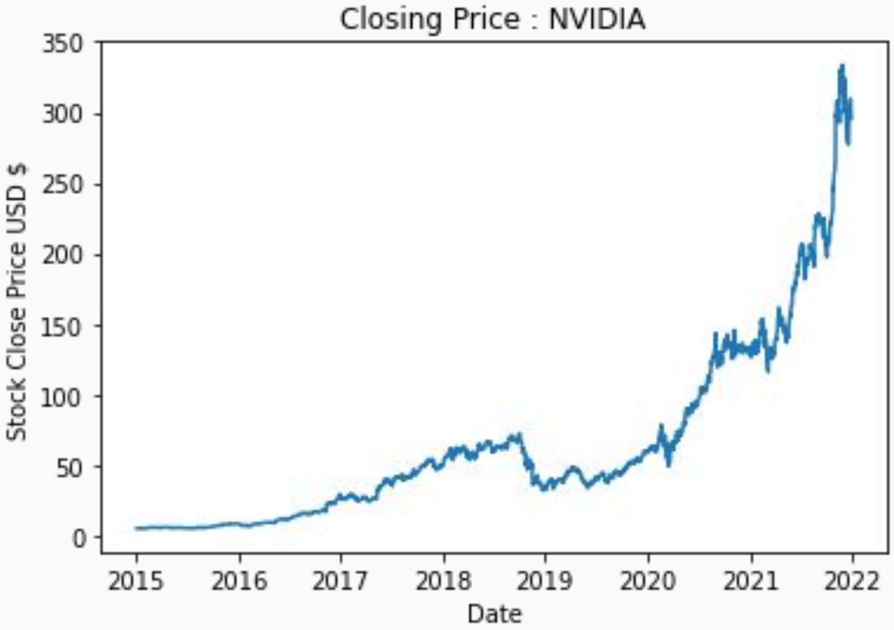
- **NVIDIA** went public in **January 1999**, with its shares priced at **$19.69** each.
- After returning **227%** and **82%** in **2016** and **2017**, respectively, shares of the graphics processing unit (GPU) specialist declined **30.8%** (including dividends) in **2018** due of The major market sell-off in October and "cryptocurrency hangover," in third Quarter.
- At start of **2020** there was a decline due to **covid-19** impact, but global **semiconductor shortage** served boon for NVDA and with strong data-centers, interest in gaming exploded in later in **2020**. By the end it increased by **121%.**
- NVDA underwent a 4-for-1 forward **stock split** in **July 2021**.
- In **2021**, Nvidia stock was on fire. The sharpest rise came in **October** last week.
  - Strong **market trend**
  - **Meta** - new corporate name and hype around “Metaverse”
  - **Meta** releasing their earnings and spending plans for its data centers & network infrastructure

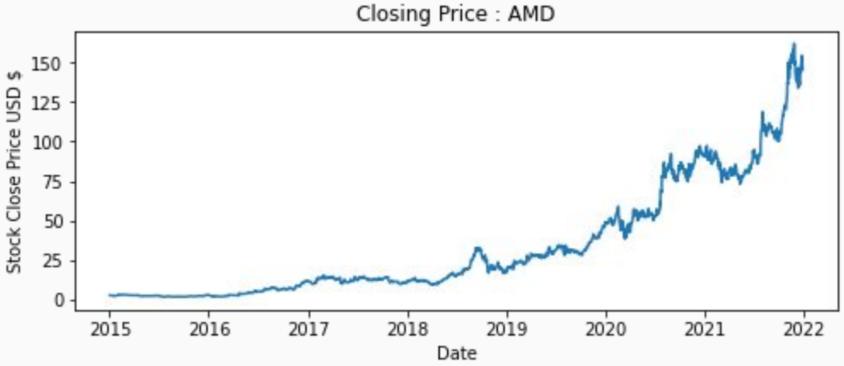
PERFORMANCE FROM 2016-2019

NVIDIA's stock gave excellent return from 2016-2019 becoming one of the favourite stocks for investors. NVIDIA's 3-year return were 409.7% which was well over the SPY's return of 50.31% in the period.



NVIDIA & IT'S COMPETITORS

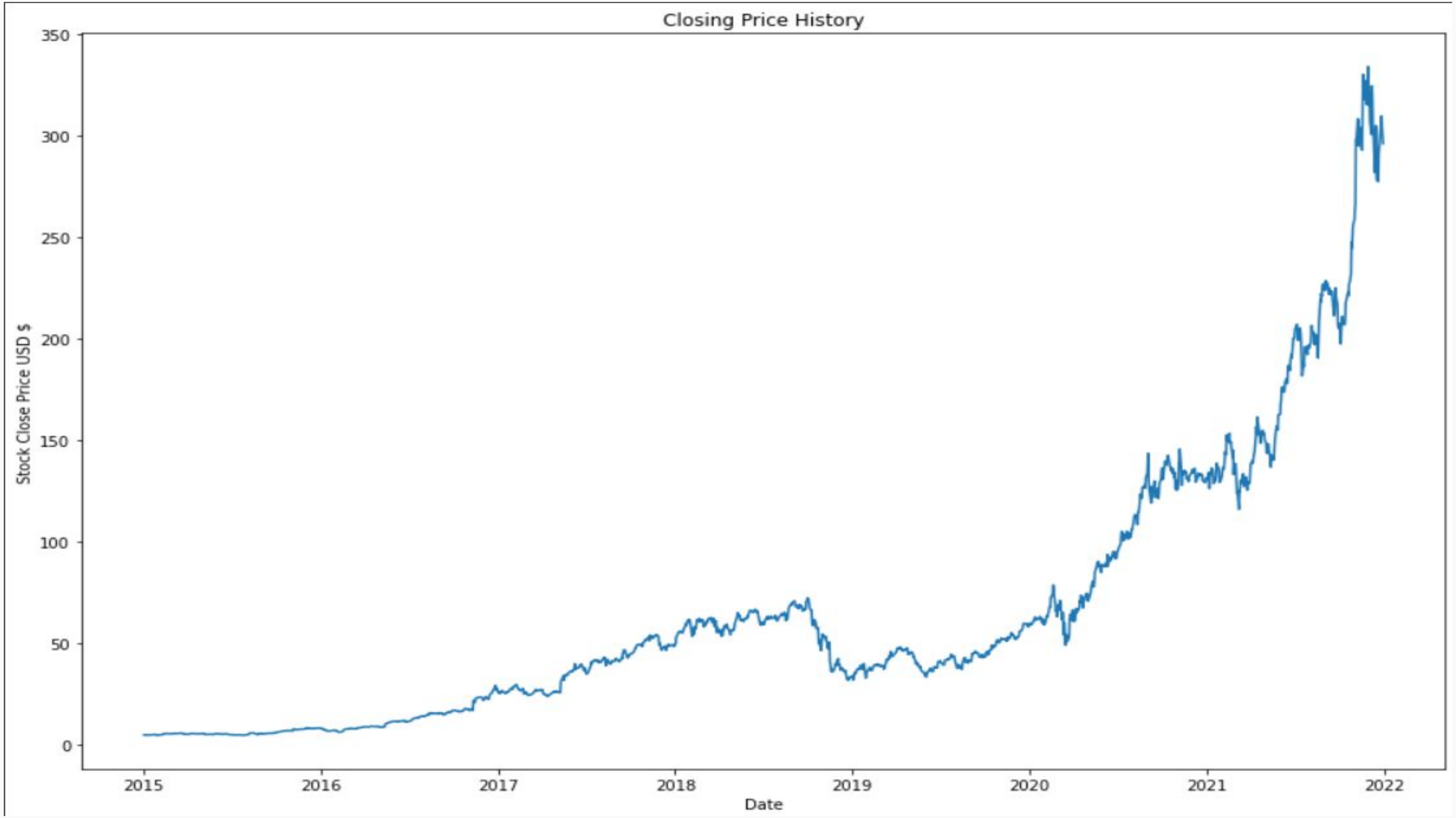




NVDA RETURN ANALYSIS

ORIGINAL TIME SERIES

Dates we took: January 1, 2017 to January 1, 2020



MODELS FOR RETURN

GARCH

The **Generalized ARCH** (GARCH) is a statistical model used in analyzing time-series data to describe volatility in financial markets.
Predicts volatility of returns on financial assets and forecasts it into the future
Responds to changes in data, it is dynamic

GARCH models are used by financial institutions to model asset risks over different holding periods.

GARCH aims to minimize errors in forecasting by accounting for errors in prior forecasting, thereby enhancing the accuracy of ongoing predictions. This is the main reason for the wide use of GARCH model in finance.

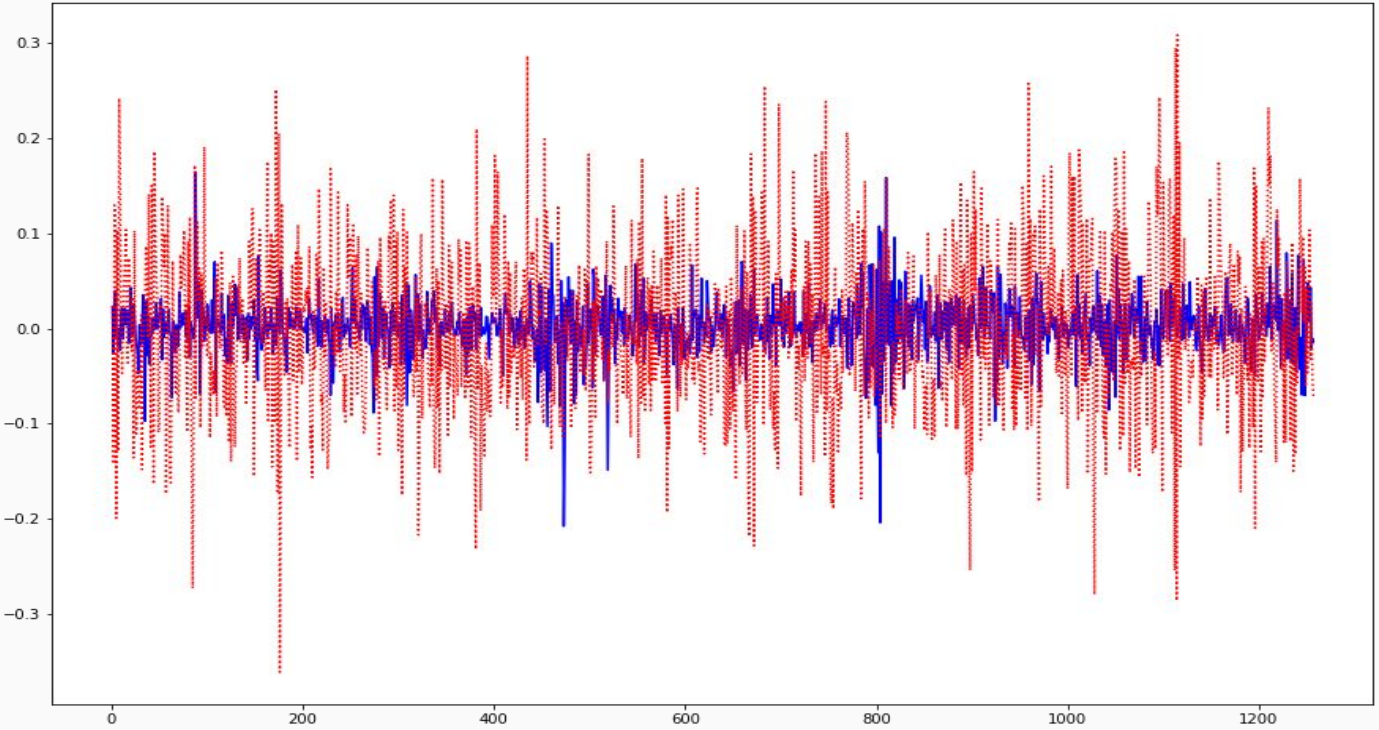
The general process for a GARCH model involves three steps :
The first is to estimate a best-fitting autoregressive model.
The second is to compute autocorrelations of the error term.
The third step is to test for significance.

GARCH (1,1)

sigma2[t] = omega + alpha*(Y[t-1]-mu)**2 + beta*sigma2[t-1]
Y[t] = mu + err[t]*sqrt(sigma2[t])
GARCH_t[t] = 0.5*(log(2*pi) + log(sigma2[t]) + (Y[t]-mu)**2/sigma2[t])

GARCH (1,1)

MU = 0 , OMEGA = 0.003, ALPHA = 0.3, BETA = 0.3



KALMAN FILTER

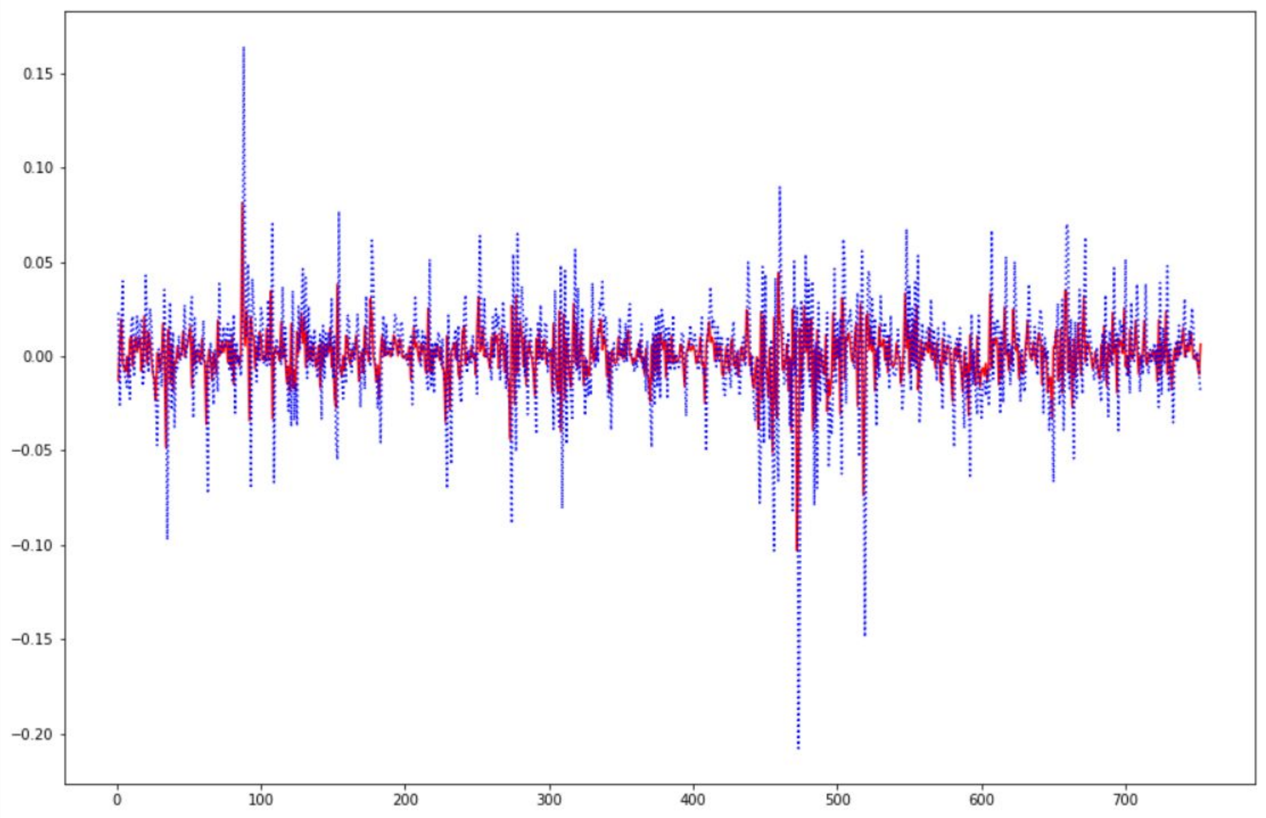
- An algorithm that provides estimates of some unknown variables given the measurements observed over time
- The Kalman filter process has two steps:

The prediction step, where the next state of the system is predicted given the previous measurements,

The update step, where the current state of the system is estimated given the measurement at that time

- A predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance when some presumed conditions are met

KALMAN FILTER RETURNS



NVDA PRICE ANALYSIS

MODELS FOR PRICE

01
OLS

$$Y_t = \alpha + \beta X_t + \epsilon_t,$$

02
LSTM

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

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

03
CAPM

$$ER_i = R_f + \beta_i(ER_m - R_f)$$

04
FAMA - FRENCH

$$r = r_f + \beta_1(r_m - r_f) + \beta_2(SMB) + \beta_3(HML) + \varepsilon$$

05
ARIMA

$$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_0 Y_0 + \epsilon_t$$

$$Y_{t-1} = \beta_1 Y_{t-2} + \beta_2 Y_{t-3} + .. + \beta_0 Y_0 + \epsilon_{t-1}$$

OLS MODEL

OLS MODEL

In data analysis, the most commonly applied econometric tool is least-squares estimation, also known as **regression**.

Ordinary least squares (OLS) regression is a statistical method of analysis that estimates the relationship between one or more independent variables and a dependent variable.

In a linear regression, the dependent variable $\boldsymbol{Y}$ is projected on a set of N predetermined independent variables, $\boldsymbol{X}$.

$$Y_t = \alpha + \beta X_t + \epsilon_t, \qquad t = 1, \ldots, T$$

where α is called the intercept, or constant. β is called the slope and ϵ is called error term.

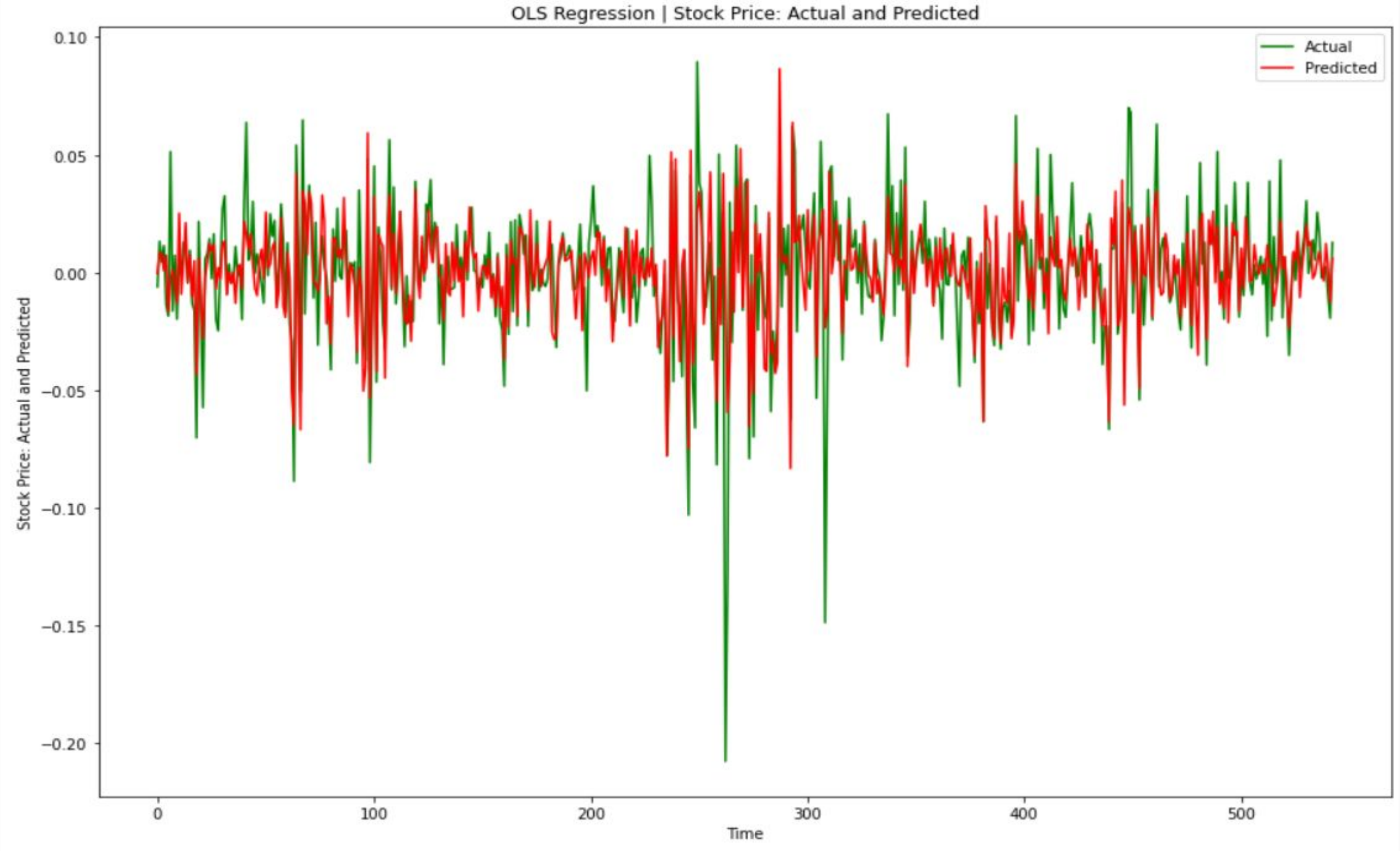
Here, Y is the regressand, and X the regressor/ predictor

```
beta              t_stat            p_val

[[2.00253538e-03 5.78304762e-02 4.76941831e-01]
 [4.33854689e-06 1.65457166e-02 4.93399515e-01]
 [1.28571955e+00 1.89550183e+00 2.90129745e-02]]

 Joint significance of all coefficients
 [0.04957036393447257, 1.0]
R-Square is
 0.7838839408442686
Adjusted R Square
 0.7819629092073288
Standard Error
 0.017255803845173064
```

OLS MODEL



OLS Regression | Stock Price: Actual and Predicted
Actual
Predicted
Stock Price: Actual and Predicted
Time
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
0
100
200
300
400
500

OLS MODEL

```
REGRESSION STATISTICS FOR AMD

-------------------------

beta                t_stat              p_val

[[ 2.70330234e-03  4.43166560e-02  4.82325998e-01]
 [-3.32616399e-06 -1.58187446e-02  5.06310503e-01]
 [ 1.53194236e+00  6.61971970e-01  2.53994593e-01]]

 Joint significance of all coefficients
 [0.05399193004775823 5, 1.0]
R-Square is
 0.305554608122844 2
Adjusted R Square
 0.302771259858807 5
Standard Error
 0.030453979565247057
```

```
REGRESSION STATISTICS FOR INTC

-------------------------

beta                t_stat              p_val

[[ 4.49828248e-05  1.61534398e-03  4.99355571e-01]
 [-4.46382563e-07 -4.65030575e-03  5.01855197e-01]
 [ 1.02096910e+00  9.66397451e-01  1.66922669e-01]]

 Joint significance of all coefficients
 [0.023981088132221367, 1.0]
R-Square is
 0.484115257862458 1
Adjusted R Square
 0.482047583545273 55
Standard Error
 0.013902666998501305
```

LSTM MODEL

LSTM MODEL

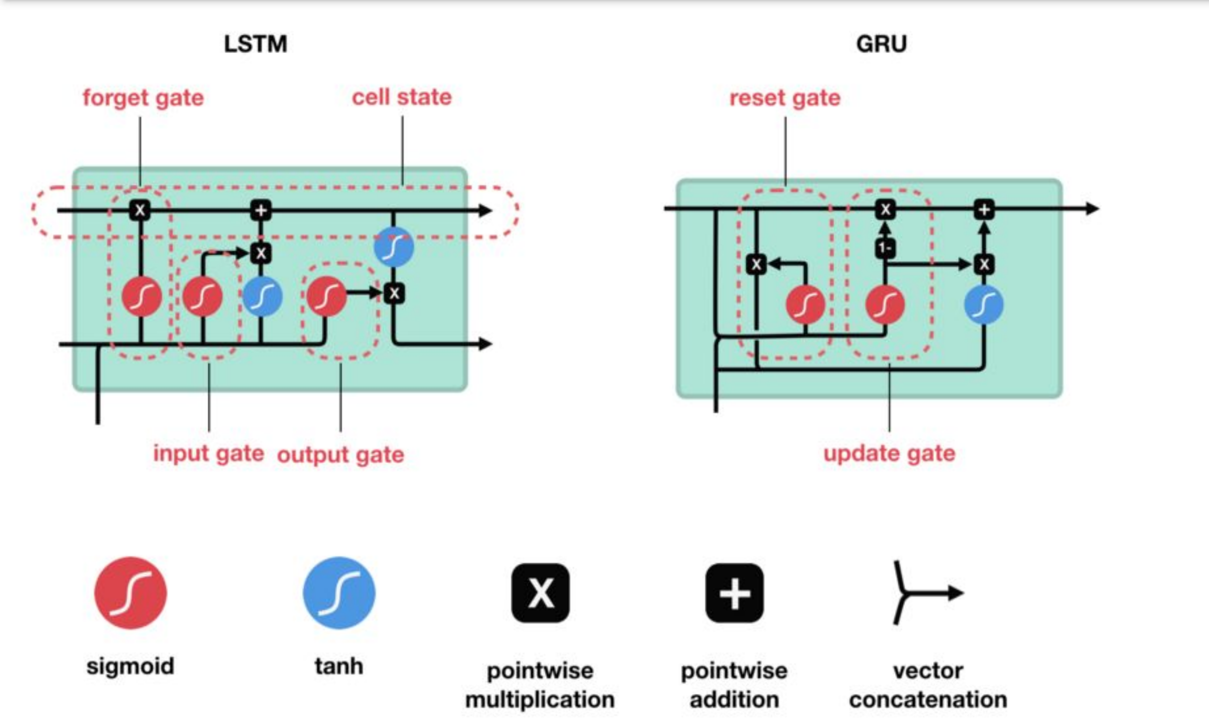
Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

These gates can learn which data in a sequence is important to keep or throw away.

By doing that, it can pass relevant information down the long chain of sequences to make predictions.

Almost all state of the art results based on recurrent neural networks are achieved with these two networks.

LSTM's and GRU's can be found in speech recognition, speech synthesis, and text generation. You can even use them to generate captions for videos



LSTM MODEL


```
### Calculate RMSE performance metrics
import math
from sklearn.metrics import mean_squared_error
math.sqrt(mean_squared_error(y_train,train_predict))
```

```
38.39841077893767
```

```
### Test Data RMSE
math.sqrt(mean_squared_error(ytest,test_predict))
```

```
144.03820944435554
```

CAPM MODEL

CAPM MODEL

CAPM gives relationship between risk and predicted return of assets in certain stocks.

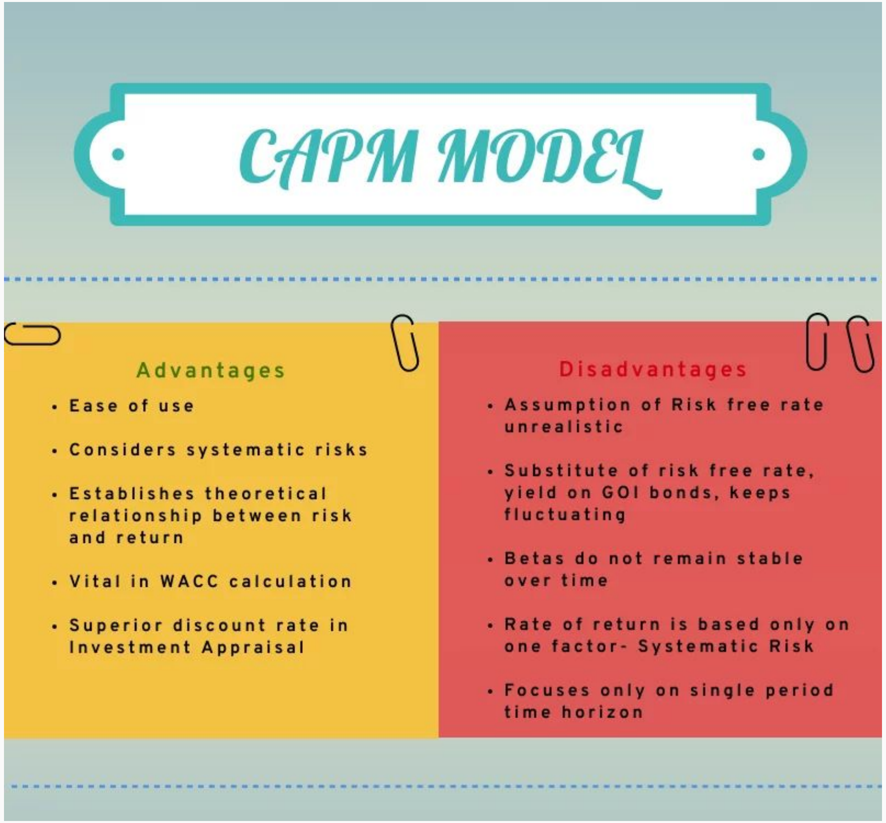
There must be a relationship between stock and market performance

We have taken TQQQ here

We calculate the alpha and beta values by using stats package and call the linear regression of it

After calculating alpha and beta values we find that **alpha=**0.0008 and **beta=**1.0849

Beta value is really low here so can infer there is not much relation between the stock and the market.

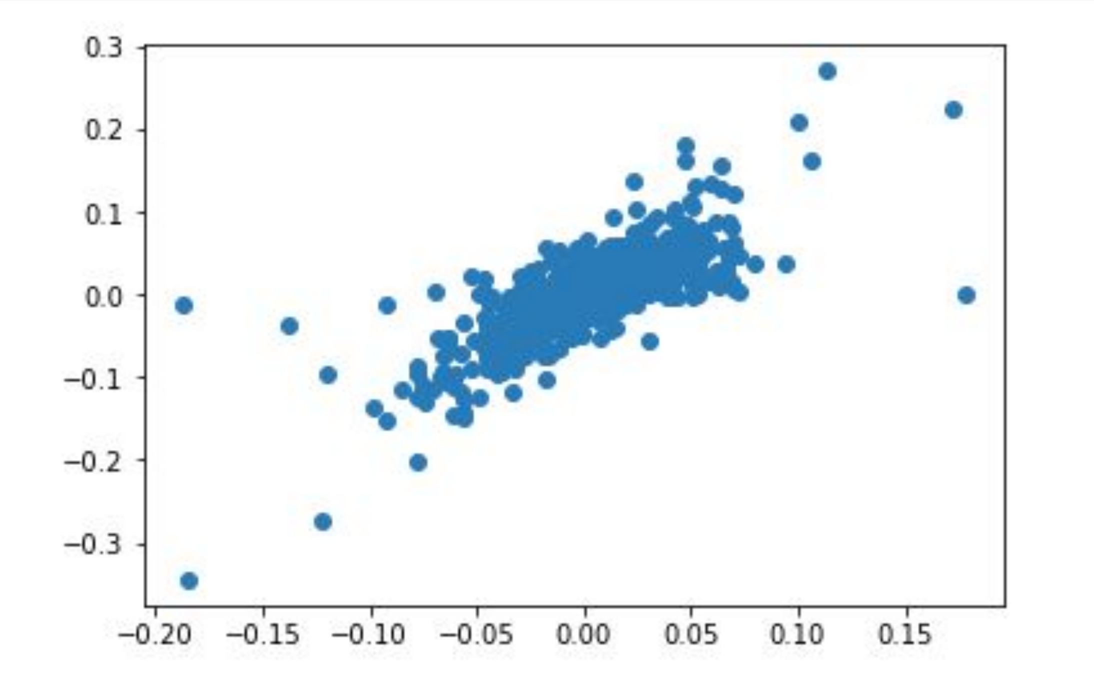


CAPM MODEL

According to CAPM, there should be some relation between the stock performance and market performance

CAPM MODEL

Scatter plot for daily returns of stock and market



CAPM FOR NVIDIA AND COMPETITORS

	NVIDIA	AMD	INTL
Alpha	0.0008	0.0012	0.0023
Beta	1.0849	0.6552	1.3772
r_val	0.7537	0.5663	0.7135
p_val	2.8241	2.0620	2.4984
std_err	0.0298	0.0300	0.0426

ARIMA
MODEL

ARIMA MODEL

Autoregressive Integrated Moving Average

ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends.

A statistical model is autoregressive if it predicts future values based on past values. For example, an ARIMA model might seek to predict a stock's future prices based on its past performance or forecast a company's earnings based on past periods.

A **ARIMA** model is classified as an "**ARIMA(p,d,q)**" model, where:

- p is the number of autoregressive terms,
- d is the number of non seasonal differences needed for stationarity, and
- q is the number of lagged forecast errors in the prediction equation.

The ARIMA forecasting equation for a stationary time series is a linear equation in which the predictors consist of lags of the dependent variable and/or lags of the forecast errors.

ARIMA Model Results

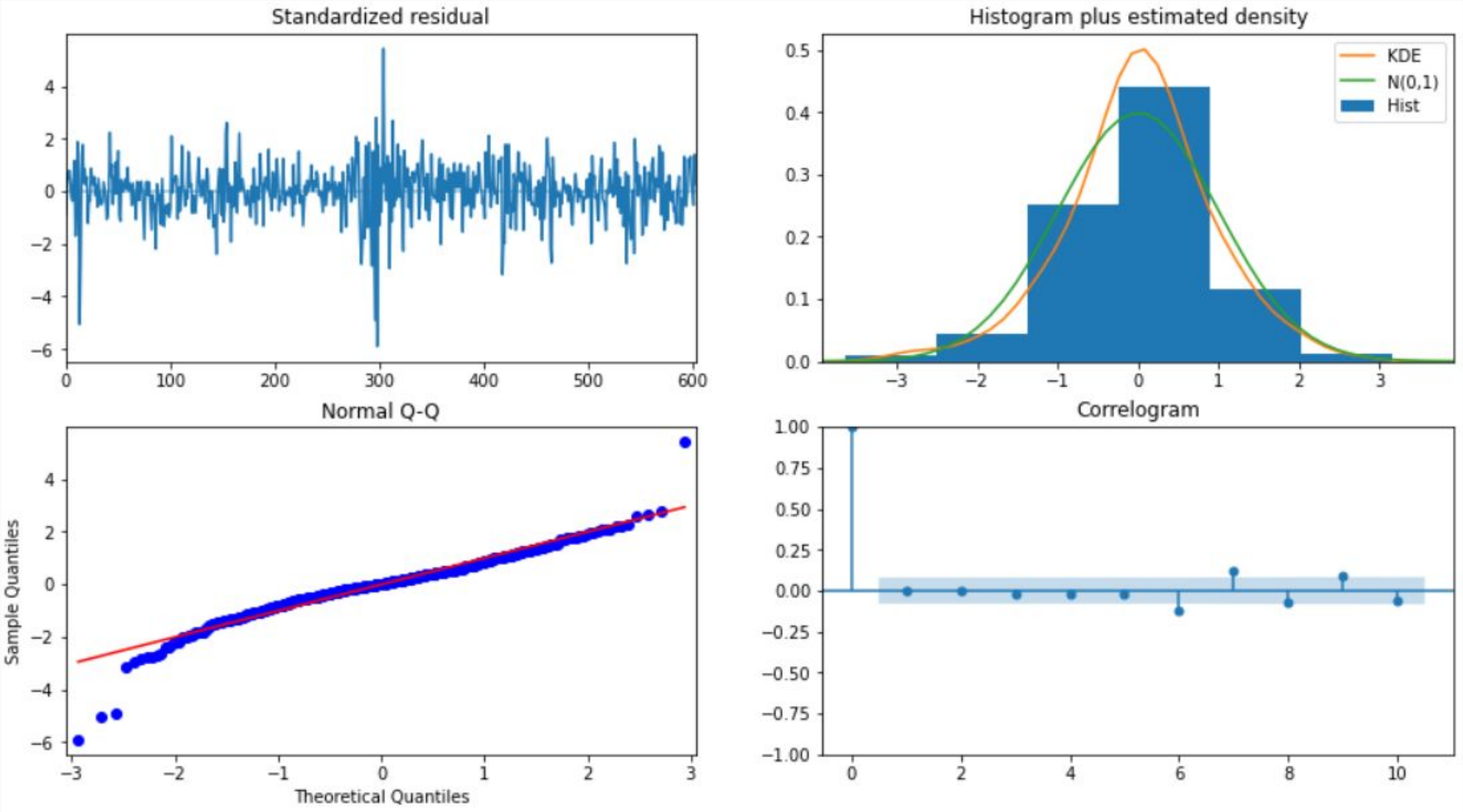
Dep. Variable:	D.Close	No. Observations:	605
Model:	ARIMA(2, 1, 0)	Log Likelihood	1258.365
Method:	css-mle	S.D. of innovations	0.030
Date:	Sat, 02 Apr 2022	AIC	-2508.730
Time:	10:13:06	BIC	-2491.109
Sample:	1	HQIC	-2501.873

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0026	0.001	2.321	0.020	0.000	0.005
ar.L1.D.Close	-0.1848	0.041	-4.554	0.000	-0.264	-0.105
ar.L2.D.Close	0.0672	0.041	1.657	0.098	-0.012	0.147

Roots

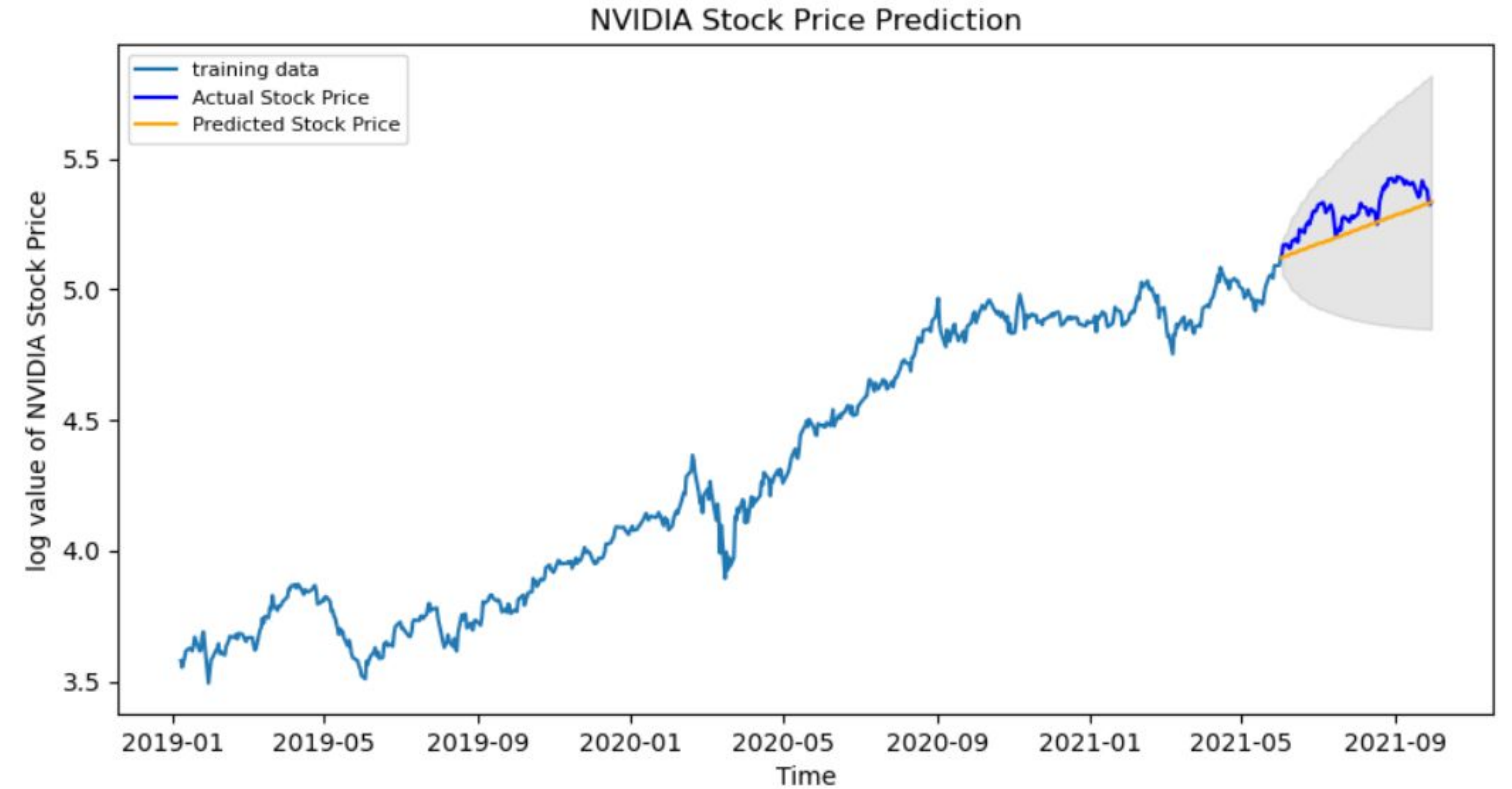
	Real	Imaginary	Modulus	Frequency
AR.1	-2.7206	+0.0000j	2.7206	0.5000
AR.2	5.4709	+0.0000j	5.4709	0.0000

ARIMA MODEL



Standardized residual
Histogram plus estimated density
KDE
N(0,1)
Hist
Normal Q-Q
Sample Quantiles
Theoretical Quantiles
Correlogram

ARIMA MODEL


NVIDIA Stock Price Prediction
training data
Actual Stock Price
Predicted Stock Price
log value of NVIDIA Stock Price
5.5
5.0
4.5
4.0
3.5
2019-01
2019-05
2019-09
2020-01
2020-05
2020-09
2021-01
2021-05
2021-09
Time

FAMA-FRENCH MODEL

FAMA-FRENCH MODEL

The **Fama French** Model is an asset pricing model

It expands on the capital asset pricing model (CAPM) by adding size risk and value risk factors to the market risk factor in CAPM

This model considers the fact that value and small-cap stocks outperform markets on a regular basis and model adjusts for this outperforming tendency, by including these two additional factors.

The Fama and French model has three factors: the size of firms, book-to-market values, and excess return on the market.These factors used are SMB (small minus big), HML (high minus low) and the portfolio's return less the risk free rate of return.

The model is essentially the result of an econometric regression of historical stock prices.

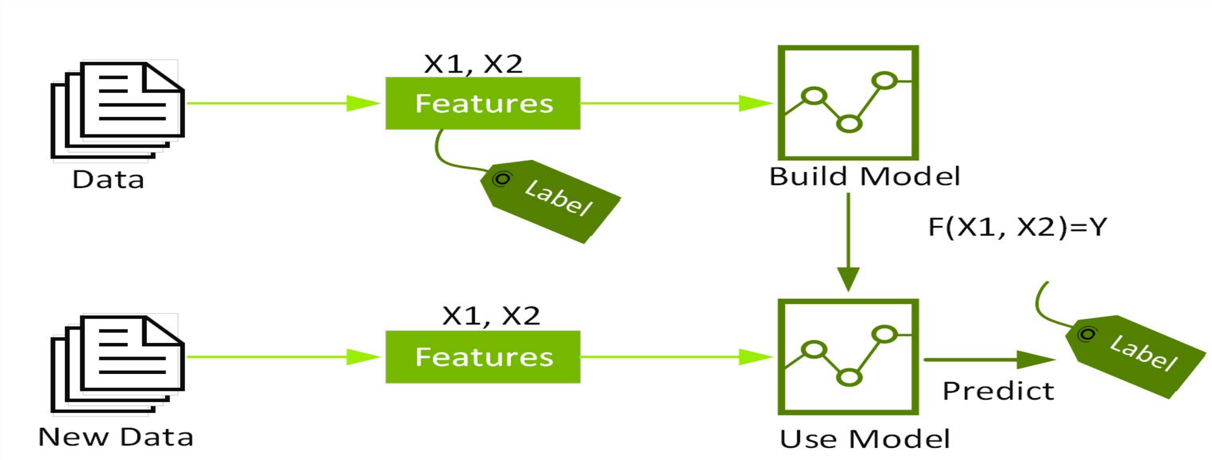
```
==============================================================================
Dep. Variable:                      y   R-squared:                       0.357
Model:                            OLS   Adj. R-squared:                  0.081
Method:                 Least Squares   F-statistic:                     1.294
Date:                Thu, 24 Mar 2022   Prob (F-statistic):              0.349
Time:                        20:00:37   Log-Likelihood:                 12.195
No. Observations:                  11   AIC:                            -16.39
Df Residuals:                       7   BIC:                            -14.80
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0547      0.055      0.999      0.351      -0.075       0.184
Mkt-RF         0.9426      0.484      1.948      0.092      -0.201       2.087
SMB           -0.9504      1.410     -0.674      0.522      -4.285       2.384
HML           -0.4428      1.275     -0.347      0.739      -3.459       2.573
==============================================================================
Omnibus:                        1.696   Durbin-Watson:                   1.995
Prob(Omnibus):                  0.428   Jarque-Bera (JB):                1.222
Skew:                           0.667   Prob(JB):                        0.543
Kurtosis:                       2.059   Cond. No.                         58.9
==============================================================================
```

XGBOOST

XGBOOST

XGBoost stands for Extreme Gradient Boosting, which is a scalable distributed gradient boosting machine learning library

XGBoost is decision-tree based ensemble ML algorithm, and decision-tree based algorithms are considered best in class right now for small to medium structured/tabular data



MODEL SYNOPSIS

CHALLENGES

Initially, the team brainstormed and dug deep in data. We explored a lot of resources to get dataset with wide variety of attributes. After several visualization and analysis, we finalized on XGBoost

PROCESS

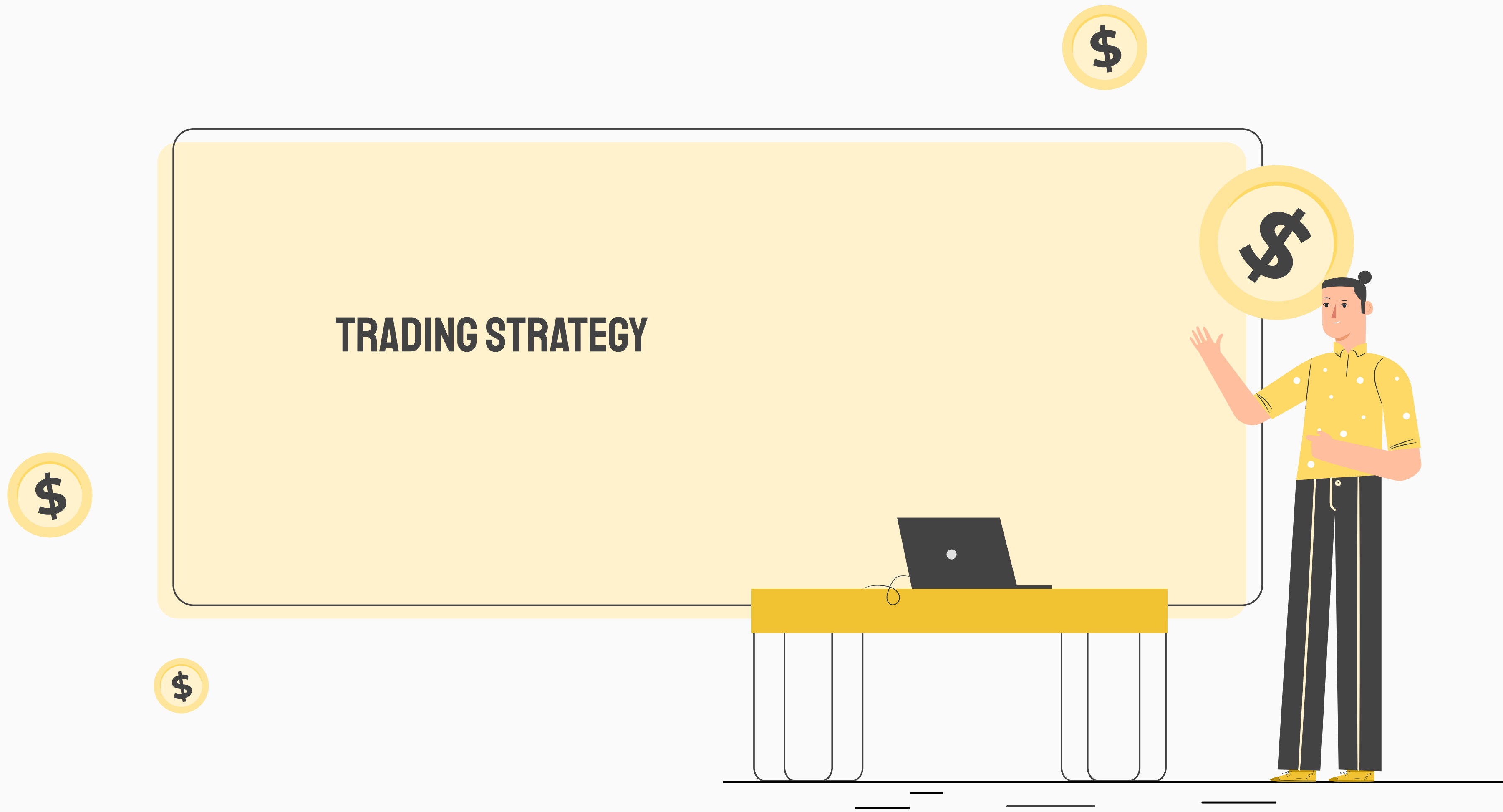
Starting off with **Technical Indicators** such as SMA, EMA, RSI and MACD. Proceeded to feature engineering for the model and fine tuning the parameters.

RESULT

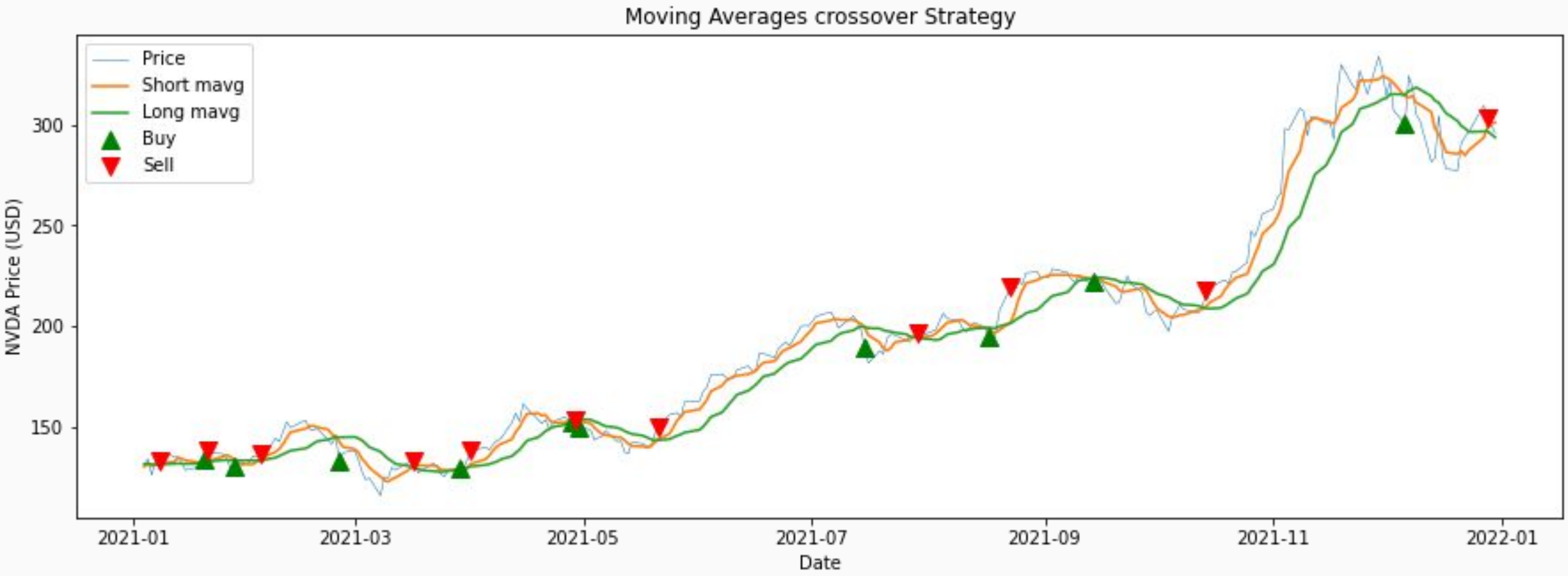
Successfully, built a model which would predict proximating 'Close Price' of the stock price on the Test data. Model results in MSE to be as low as 5.567 which conveniently provides line of best fit.

```
[89] y_pred = model.predict(X_test)
     print(f'y_true = {np.array(y_test)[:5]}')
     print(f'y_pred = {y_pred[:5]}')
```

```
y_true = [42.18000031 41.22999954 40.29750061 37.69749832 38.08750153]
y_pred = [43.36597  41.951378 41.119476 40.56402  39.48251 ]
```

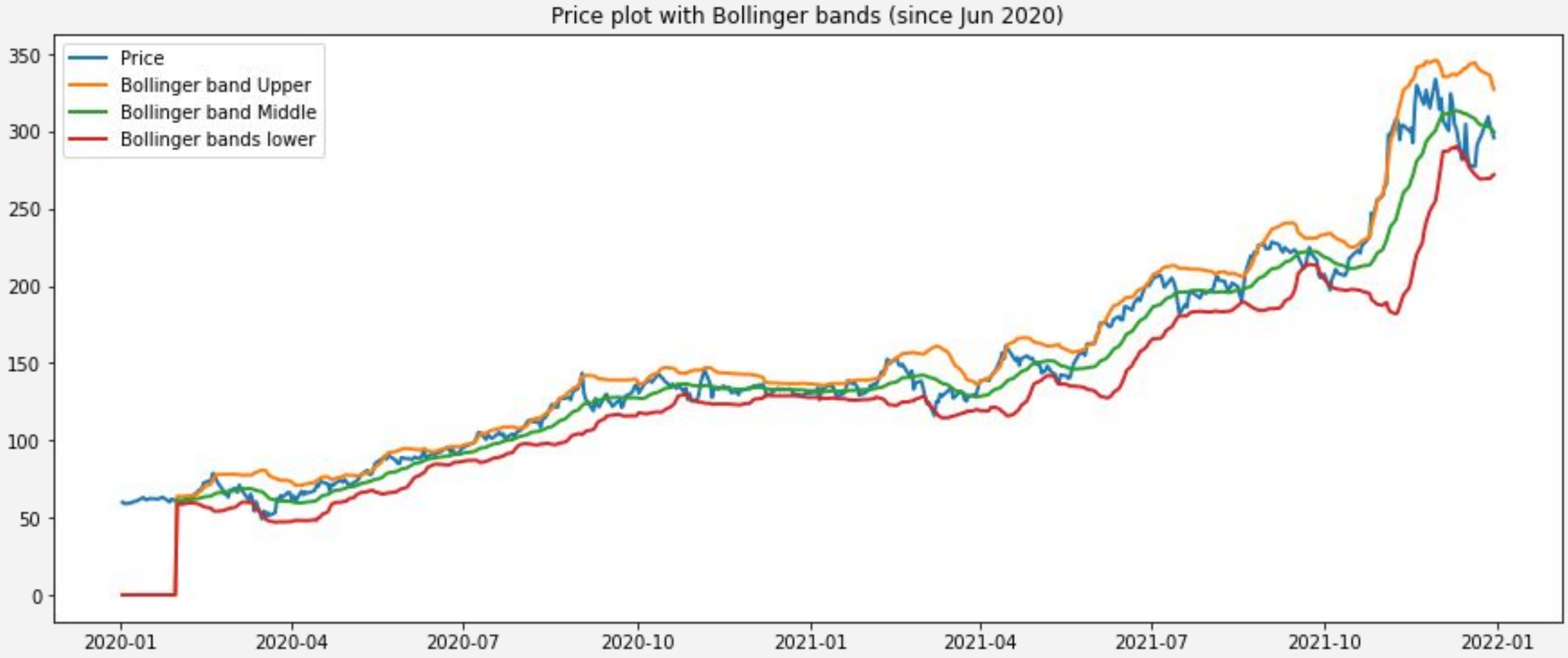

TRADING STRATEGY

BACKGROUND

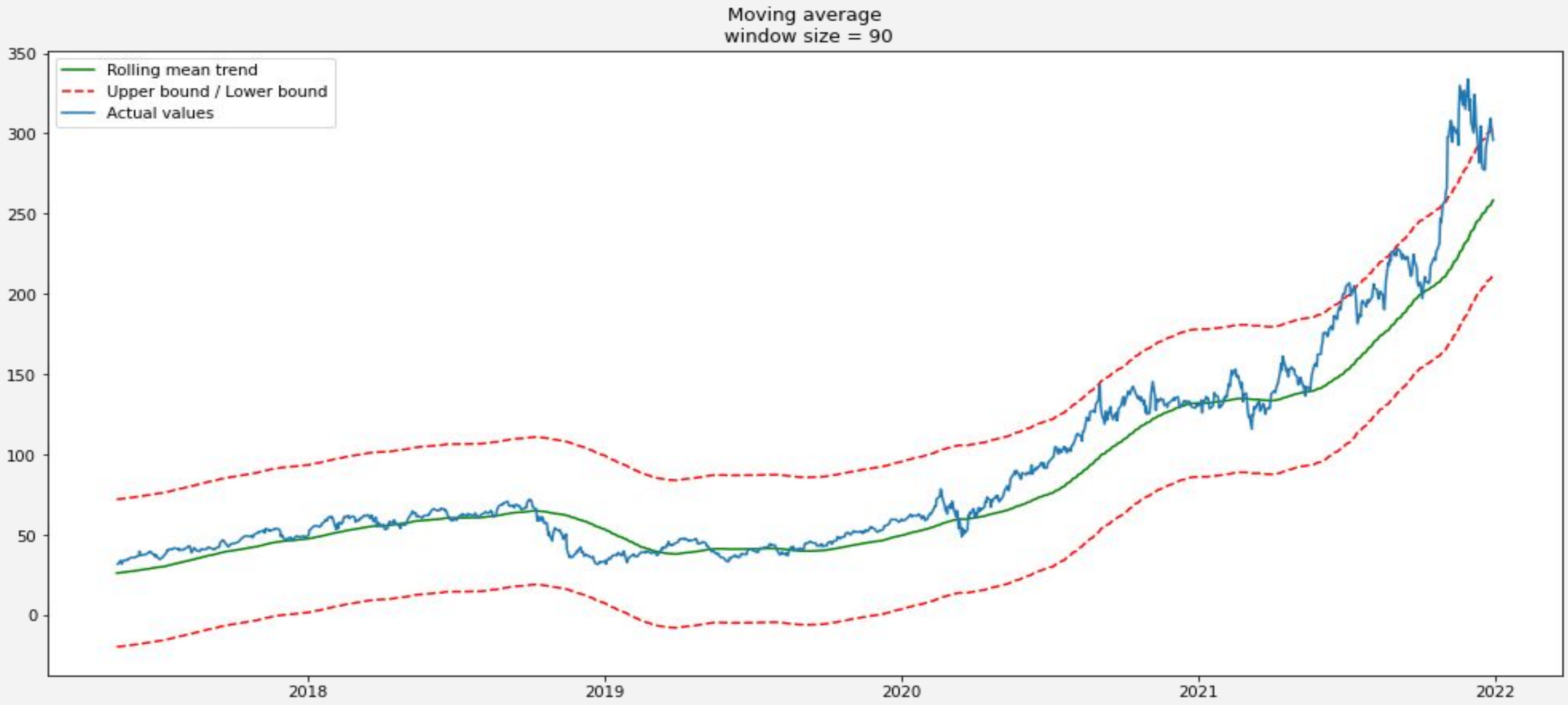


Moving Averages crossover Strategy
Price
Short mavg
Long mavg
Buy
Sell
NVDA Price (USD)
300
250
200
150
2021-01
2021-03
2021-05
2021-07
2021-09
2021-11
2022-01
Date

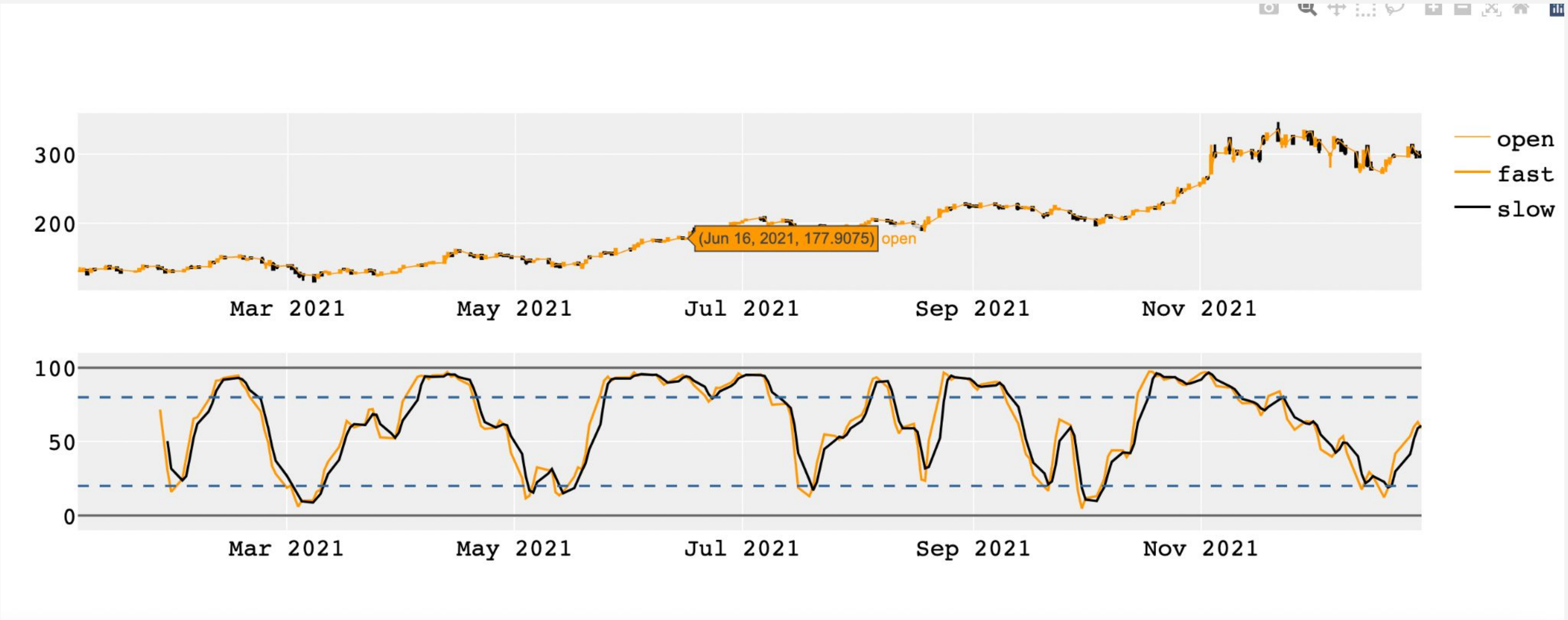
BOLLINGER BAND



MOVING AVERAGE

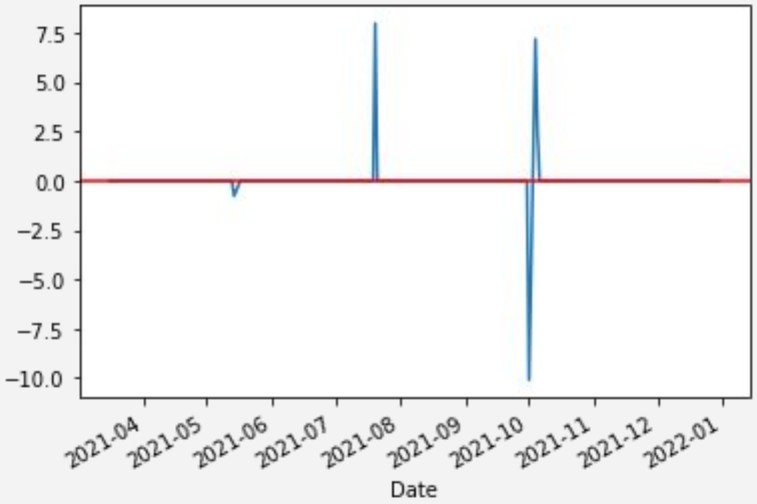


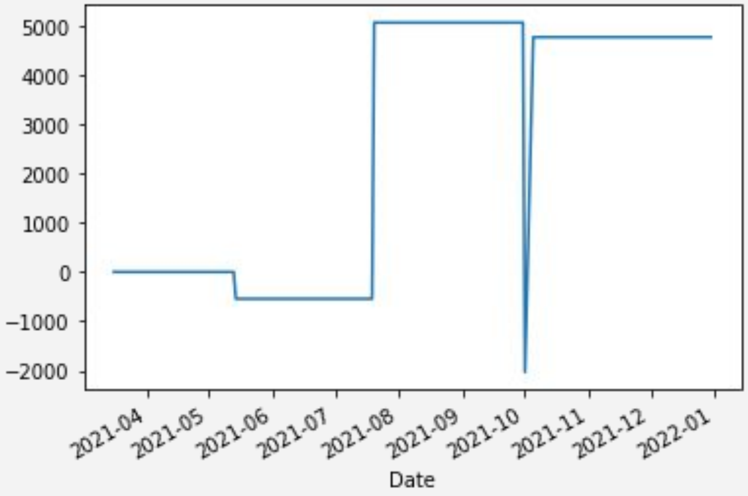
TRADING STRATEGY



STRATEGY I

```
conditions=[
            (NVDA['stochk_14_3_3'] > 80) & (NVDA['stochd_14_3_3']) > 80 & (NVDA['stochk_14_3_3'] < NVDA['stochd_14_3_3'])
            (NVDA['stochk_14_3_3'] < 20) & (NVDA['stochd_14_3_3']< 20) & (NVDA['stochk_14_3_3'] > NVDA['stochd_14_3_3'])
]
```

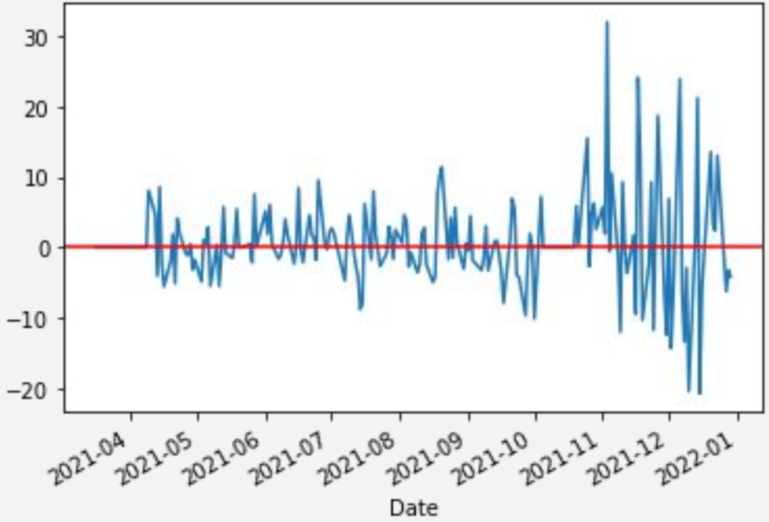


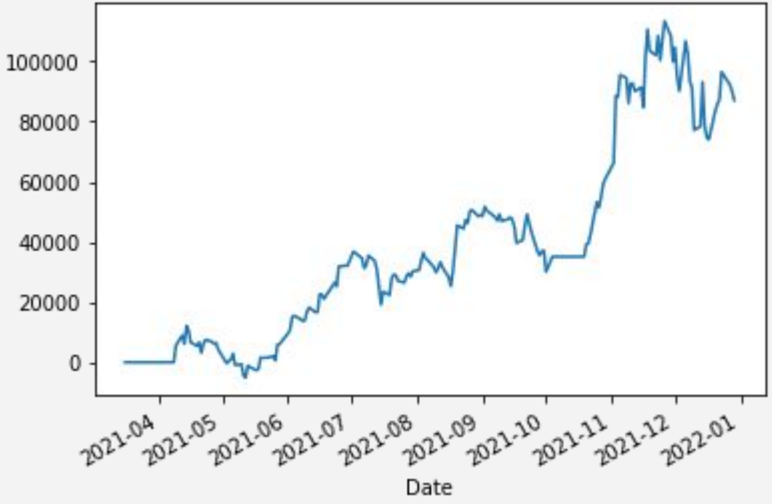


Investment = 100000
Return at the end of 2021 = 47%

STRATEGY II

```
conditions = [NVDA['MA10'] > NVDA['MA50'],
              NVDA['MA10'] < NVDA['MA50']]
```





Investment = 100000
Return at the end of 2021 = 87%


THANK YOU!