TESLA, Inc

Agenda

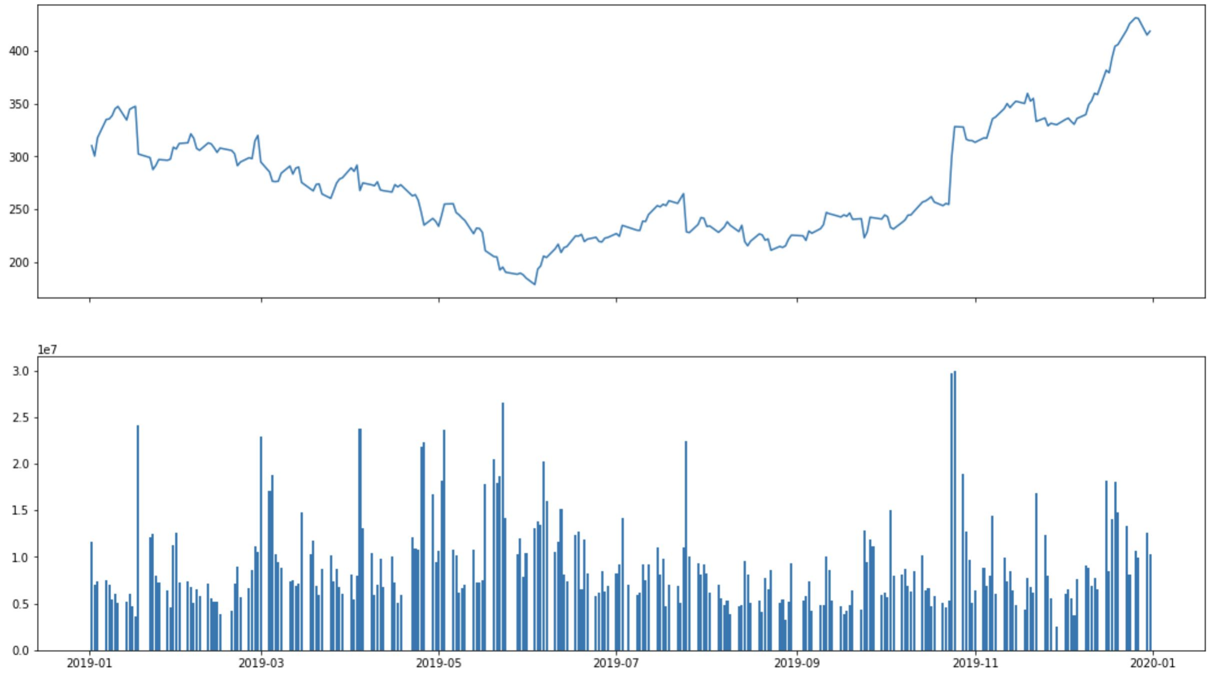
- General Data Overview Recap
- Models
  - CAPM
  - Fama French
  - APT
  - Pattern Recognition
  - Vol-GARCH
- Bootstrap result
- Next step

General Data Overview Recap

Yaksh Bhatt

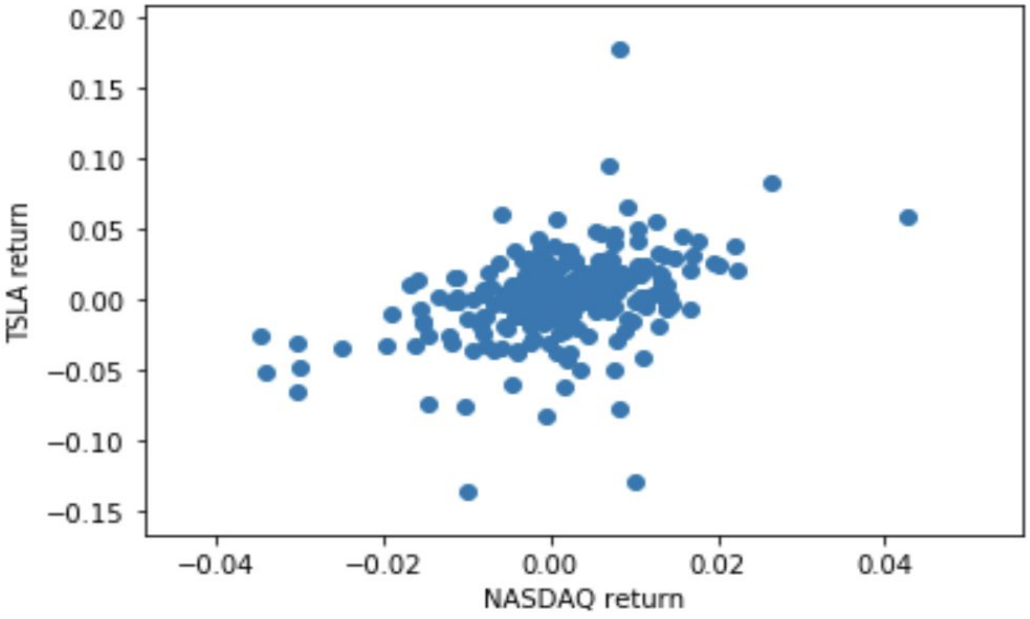
General Exploratory Data Analysis

- TSLA Price vs Volume: 20190101-20191231



General Exploratory Data Analysis Cont.

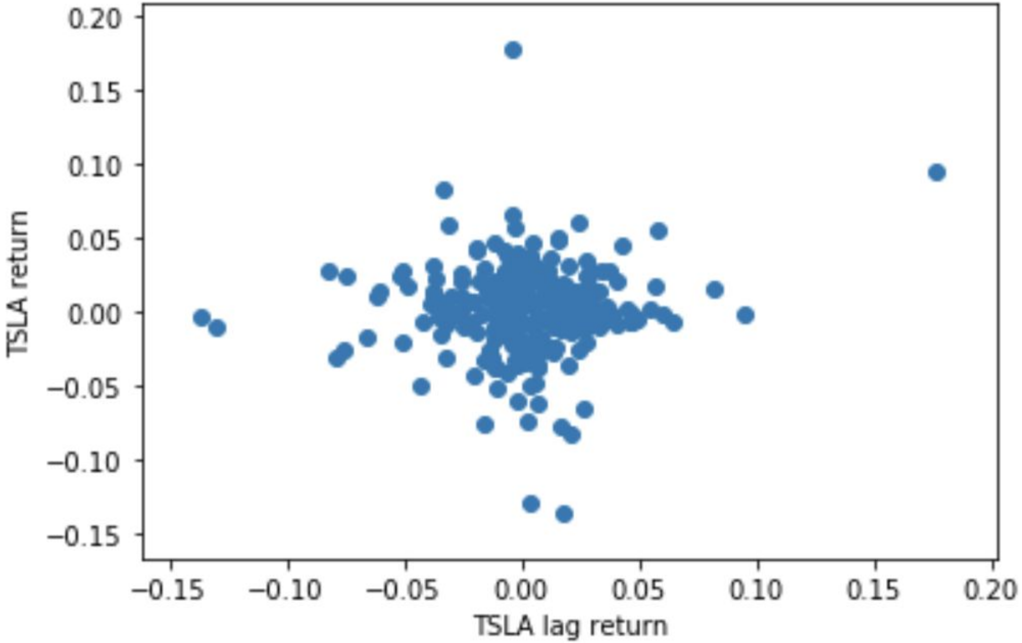
- Market return vs TSLA return (2019)



  - Positive trend between market return and tesla price return

General Exploratory Data Analysis Cont.

- Autocorrelation between TSLA return (2019) vs its first lag



    - No significant correlation

The Capital Asset Price Model (CAPM)

Ruoyu Xu

CAPM's Brief Intro

$$E(r_i) = r_f + \beta_{im}[E(r_m) - r_f]$$

- $E(R_i)$ is the expected return on the capital asset
- R_f is the risk-free rate of interest such as interest arising from government bonds
- β_i (beta) is the sensitivity of the expected excess asset returns to the expected excess market returns
- $E(R_m)$ is the expected return of the market
- $E(R_m) - R_f$ is sometimes known as the market premium (the difference between the expected market rate of return and the risk-free rate of return)
- $E(R_i) - R_f$ is also known as the risk premium
- $\rho_{i,m}$ denotes the correlation coefficient between the investment i and the market m
- σ_i is the standard deviation for the investment i
- σ_m is the standard deviation for the market m

Statistics

```
#bata
beta=np.linalg.inv(X.transpose()@X)@(X.transpose()@Y)
y_hat=X@beta
residuals=Y-y_hat

#sigma2
sig2=(1/X.shape[0])*(np.sum(residuals*residuals))
cov_beta=(sig2)*np.linalg.inv(X.transpose()@X)

#SD of betas
sd_beta=np.sqrt(np.diag(cov_beta))

#R squared
r_sq=1-residuals.transpose()@residuals/(X.shape[0]*np.var(Y))
adj_rsq=1-(1-r_sq)*(X.shape[0]-1)/(X.shape[0]-X.shape[1])

#t-stats
t_stat=beta.transpose()/sd_beta

#p-values
pvs=(1-norm.cdf(abs(t_stat)))*2
```

Statistics

R-sq	0.163

```
x_capm=capmdata['NASDAQ_rp'].values.reshape(-1,1)
y_capm=capmdata['TSLA_rp'].values.reshape(-1,1)
lrfit(x_capm,y_capm)
```

```
Betas:  [[1.35263396e-04 1.25922200e+00]]
sigma2:  0.0007916762023620447
SD of betas:  [0.00178809 0.18006932]
t-stats:  [[0.07564683 6.99298474]]
p-values:  [[9.39700077e-01 2.69095857e-12]]
R squared:  [[0.16305947]]
```

	Coefficient	Standard Deviation	T-Stat	P-Value
Intercept	1.35263396e-04	0.00178809	0.07564683	9.39700077e-01
Nasdaq Ret - Risk free	1.259222	0.18006932	6.99298474	2.69095857e-12

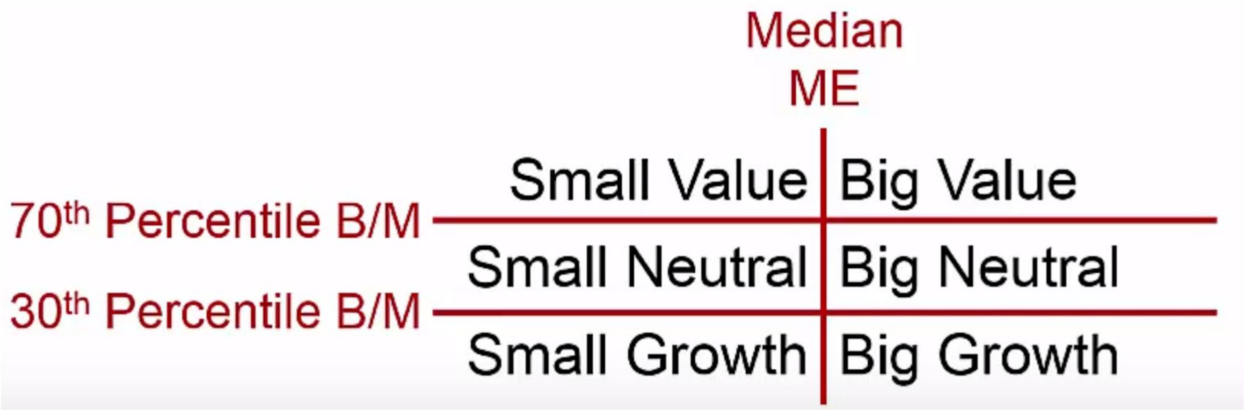
Fama-French Three Factor Model

Rushabh Nisher

Fama-French's Brief Intro

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

- $E(r_i)$ is the expected rate of return
- r_f is the risk-free rate
- $(r_m - r_f)$ is the market risk premium
- SMB(Small Minus Big) is the historic excess returns of small-cap companies over large-cap companies
- HML(High Minus Low) is the historic excess of value stocks(high book-to-price ratio) over the growth stocks (low book-to-price ratio)
- $\beta_{1,2,3}$ is the factor's Coefficient
- ε is the market risk



- SMB = $\frac{\text{(Small Value + Small Neutral + Small Growth)}}{3} - \frac{\text{(Big Value + Big Neutral + Big Growth)}}{3}$ i.e. Small Cap - Big Cap

- HML = $\frac{\text{(Small Value + Big Value)}}{2} - \frac{\text{(Small Growth + Big Growth)}}{2}$ i.e. Value Stocks – Growth Stocks

Fama-French Model Assumptions

- The Fama-French model aims to describe stock returns through three factors:

1. Market risk,
2. Outperformance of small-cap companies relative to large-cap companies
3. Outperformance of high book-to-market value companies versus low book-to-market value companies.

- The rationale behind the model is that high value and small-cap companies tend to regularly outperform the overall market.

Statistics

	Coefficient		Std Error		T-statistic		P-Value	
	OLS	Linear Algebra	OLS	Linear Algebra	OLS	Linear Algebra	OLS	Linear Algebra
Intercept	0.0005	0.0005	0.002	0.002	0.281	0.281	0.779	0.779
Market-excess	1.1649	1.1732	0.231	0.231	5.046	5.082	0.000	0.000
SMB	0.9680	0.9599	0.408	0.408	2.370	2.350	0.019	0.020
HML	-0.2194	-0.2238	0.315	0.315	-0.695	-0.709	0.487	0.479

OLS
R-squared: 0.147
Adj. R-squared: 0.136

Linear Algebra
R-squared: 0.14627
Adj. R-squared: 0.13939

Observations

1. HML - not significant for the year of 2019
2. When considering the data for multiple years (2015 to 2019) SMB is insignificant
3. But when we consider the values from before 2015 (2011 to 2019), All the factors SMB, HML, and mkt excess are significant (p-values < 0.000)
4. For a long term period (over 6 years), Fama-French Model can be useful for predicting our daily returns
5. But for short term periods (less than 5 years), Fama-French Model is most likely not going to work with these 3 factors (SMB, HML & mkt excess)

The Arbitrage Pricing Theory (APT)

Yujin Xiang

Brief intro

Arbitrage pricing theory (APT) is a multi-factor asset pricing model based on the idea that an asset's returns can be predicted using the linear relationship between the asset's expected return and a number of macroeconomic variables that capture systematic risk

APT model formula

$$\mathrm{E(R)_i} = E(R)_z + (E(I) - E(R)_z) \times \beta_n$$

where:

$\mathrm{E(R)_i}$ = Expected return on the asset

R_z = Risk-free rate of return

β_n = Sensitivity of the asset price to macroeconomic factor n

Ei = Risk premium associated with factor i

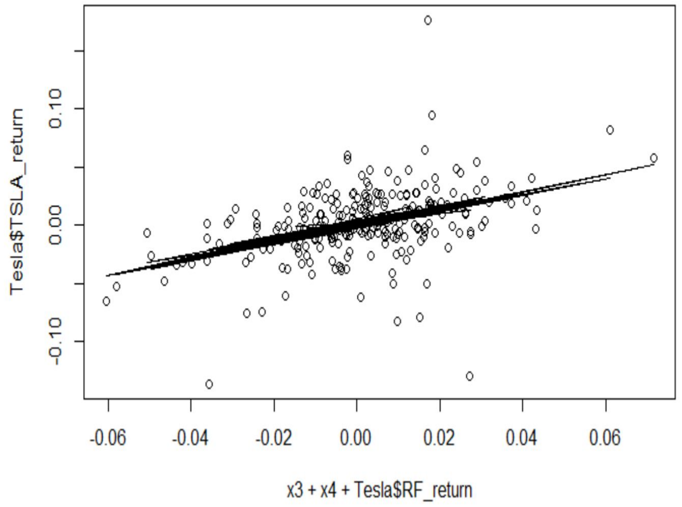
- Parameters to input:
  - Gas Price
  - NASDAQ index
  - Exchange rate
  - Automobile Index

Statistics

```
Residuals:
      Min        1Q    Median        3Q       Max
-0.148465 -0.011536  0.000079  0.013616  0.164296

Coefficients: (1 not defined because of singularities)
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      0.0004425  0.0017972   0.246 0.805734
x1               0.0567732  0.0954162   0.595 0.552389
x2              -0.0737628  0.7266580  -0.102 0.919229
x3               0.8564362  0.2461573   3.479 0.000595 ***
x4               0.4952330  0.2201181   2.250 0.025347 *
Tesla$RF_return         NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02815 on 245 degrees of freedom
Multiple R-squared:  0.1822,    Adjusted R-squared:  0.1689
F-statistic: 13.65 on 4 and 245 DF,  p-value: 4.629e-10
```



Update

Because two factors are not significant(Gas price and Exchange rate), so I need to find some new factors. I find lithium battery and energy sector which might be significant factors. After implementing the new data, there are four significant factors.

- Parameters update
  - Gas price
  - NASDAQ index
  - Lithium battery ETF
  - Energy Sector Index

Statistics

r-sq	0.2486
Adjusted r-sq	0.2363

	coefficient	Standard deviation	t-value	p-value
Intercept	0.0006714	0.0017321	0.388	0.69864
Gas Price	0.2586621	0.1150838	2.248	0.02549
Nasdaq Index	0.7021175	0.2665838	2.634	0.00898
Lithium Battery	1.1035328	0.2175283	5.073	7.74e-07
Energy Sector	-0.7146801	0.2501757	-2.857	0.00465

Pattern Recognition & LSTM Model

Shreya Ganesh

Introduction

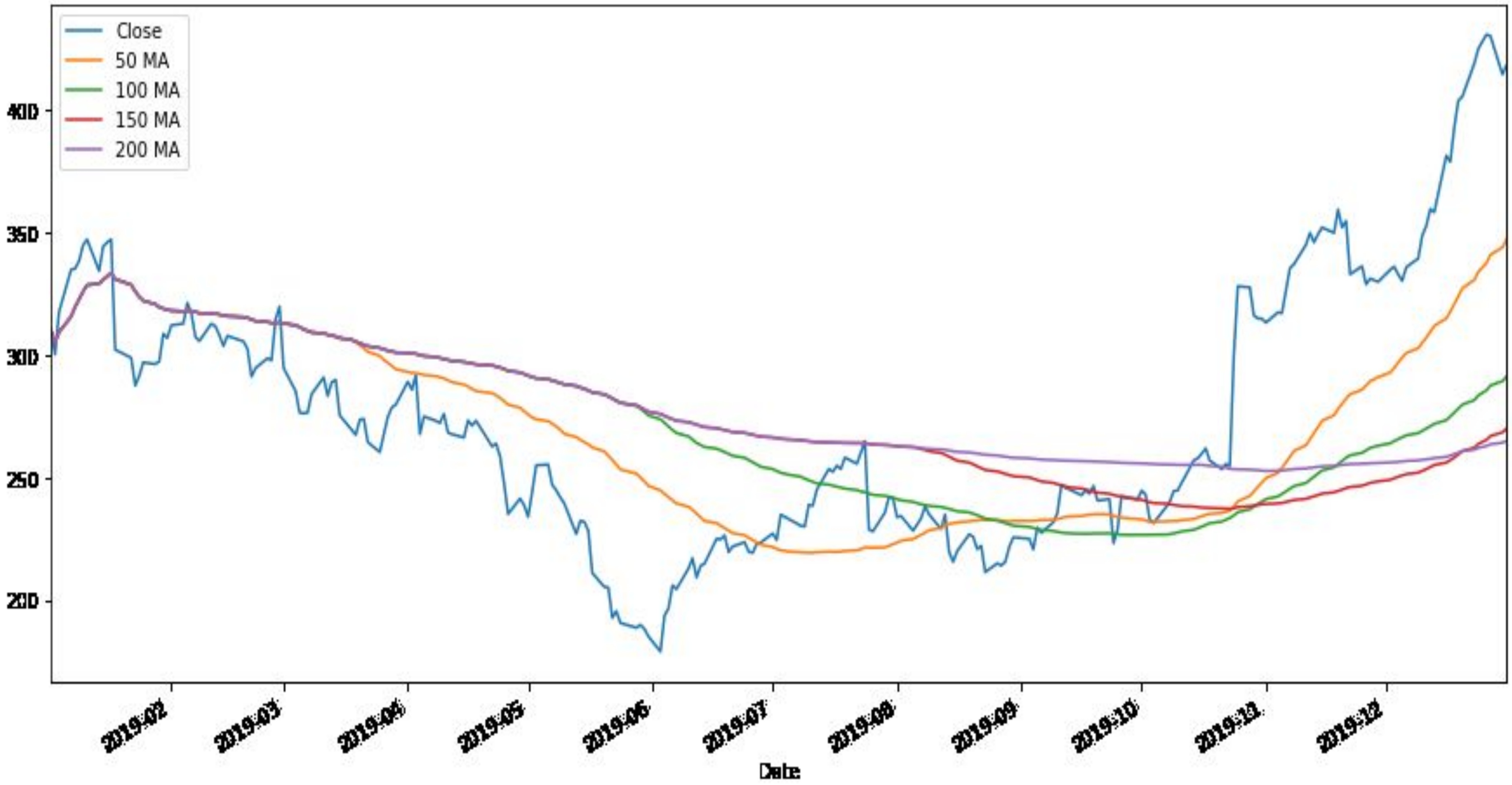
- Pattern recognition techniques are the methods that have been used by traders throughout time to find the opportune time to buy or sell a stock through identifying the recurring sequences/patterns
- There are several different pattern recognition techniques that are used by individual traders, here with Tesla’s stock we have tried to implement two such techniques namely Moving Averages & Candlestick Chart

Moving Averages

- Moving Averages is a simple pattern recognition technique that smooths out the price data by creating a constantly updated average price
- The average here is taken over any period of time that the trader chooses eg: 30 days, 30 weeks etc.
- The main advantage of using this particular technique is that it easily helps in identifying the trend and understanding which way the price is going/moving

Calculation

- A simple moving average is calculated by computing average price of a security over a period of time
- Eg: To calculate a SMA for a 5 day period the trader can take either the closing price or low to compute the moving average, by first computing the sum of 5 day close prices and then dividing it with the time duration of 5 days
- Additionally, the moving average also has a tendency to act as a support or resistance eg: in uptrend it acts as a support while in downtrend it acts as resistance



Close
50 MA
100 MA
150 MA
200 MA
400
350
300
250
200
2019-02
2019-03
2019-04
2019-05
2019-06
2019-07
2019-08
2019-09
2019-10
2019-11
2019-12
Date

Candlestick Charts

- The candlestick charts are used by the traders to determine possible price movement based on past patterns and are thereby particularly advantageous since they show the high, low, open and close prices all together for any time period that the trader specifies
- The candlestick also has a wide component called the the ‘real body’ which represents the price difference between the open and close in a day’s trading
- Just above and below the real body are the ‘shadows’ which show the high and low prices of that day

Basic Candlestick Patterns

- The candlestick are created by up and down movement of prices
- There are many candlestick patterns but they are broadly divided into two patterns namely bearish and bullish patterns
- Bearish pattern indicates that the price is likely to rise, while the bullish indicates that the price is likely to fall
- Additionally, no patterns work all the time as they are only the tendencies not the guarantees about any stock price

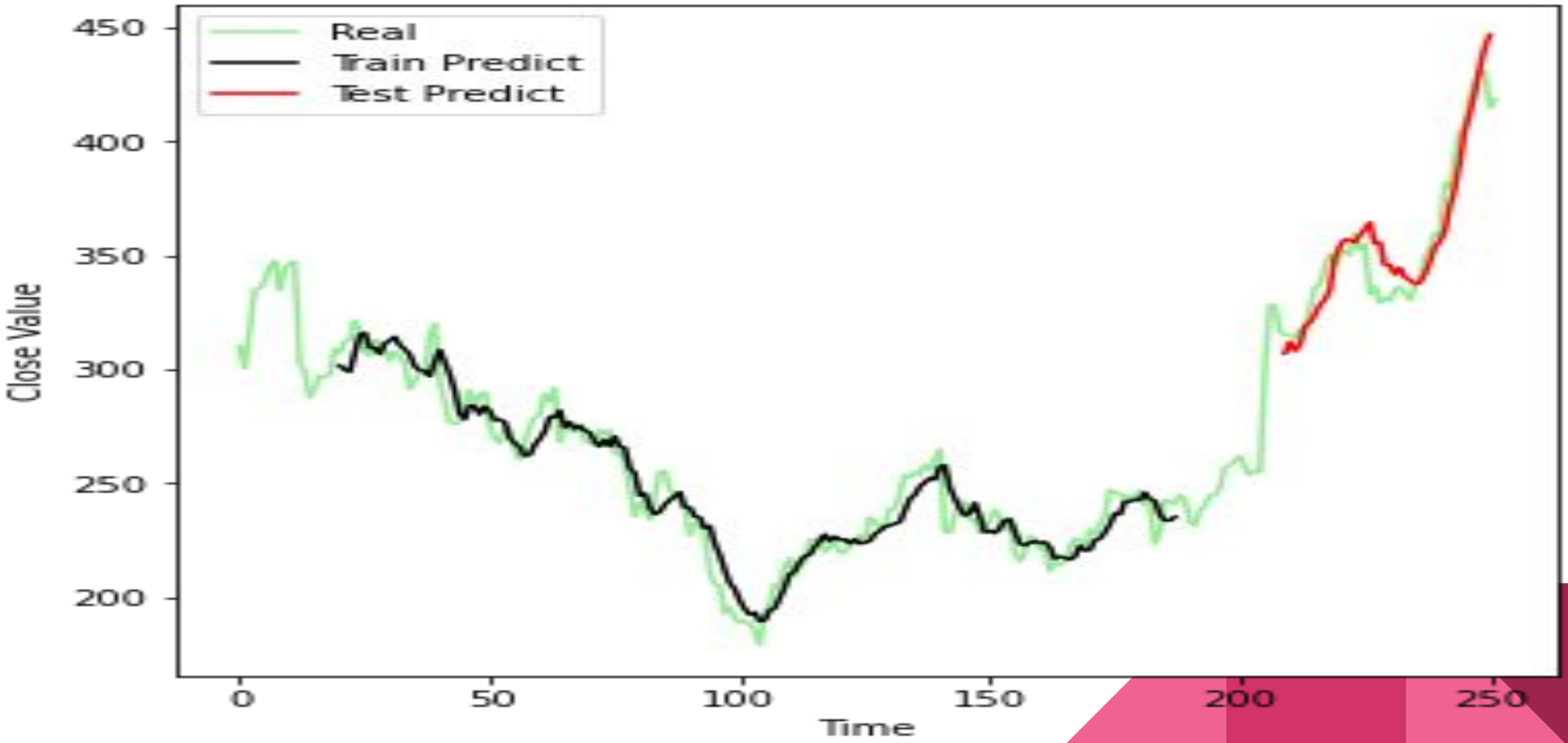
Candlestick for a month of December 2019

Candlestick for a year (2019)

Long Short Term Memory Neural Network

- One of the very useful methods to predict the stock prices is by using Long Short Term Memory Neural Network or LSTM
- LSTM's are improvised Recurrent Neural Networks and perform better than conventional RNN's since they are able to retain information over a long period of time unlike RNN which retain only very little past information
- This ability of LSTM to retain better, quintessentially makes LSTM's perfect fit for predicting the stock prices

LSTM Model for Tesla Close Price



Network details and results

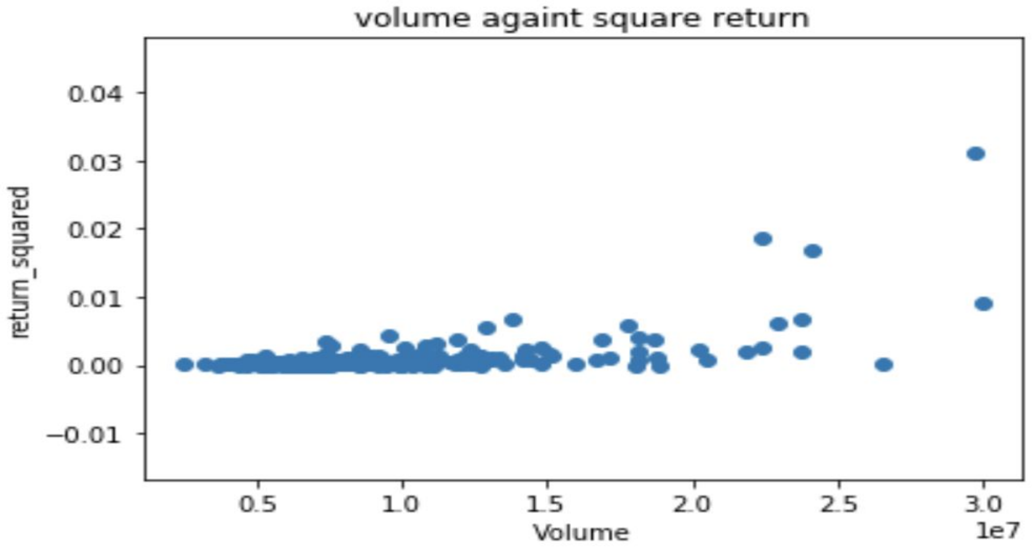
- Train Score: 9 RMSE
- Test Score: 12 RMSE
- Time steps = 20
- Train-test split (75-25 ratio)
- Epoch= 20

GARCH with Volume (Vol-GARCH)

Xinyue Cai

Why GARCH with Volume

- Correlation between return-squared vs Volume
  - E(mean price return) = 0



  - The higher the volume, the larger the volatility

Statistics

- Fit model: $$r_t^{\,2} = \beta_0 + \beta_1 * Volume_{t-1}$$

R-sq	0.345

	Coefficient	**Standard Deviation**	**T-Stat**	**P-Value**
Intercept	-2.13841967e-03	3.02744624e-04	-7.06344392	1.62425629e-12
Volume	3.37423810e-10	2.93523093e-11	11.49564781	0.00000000e+00

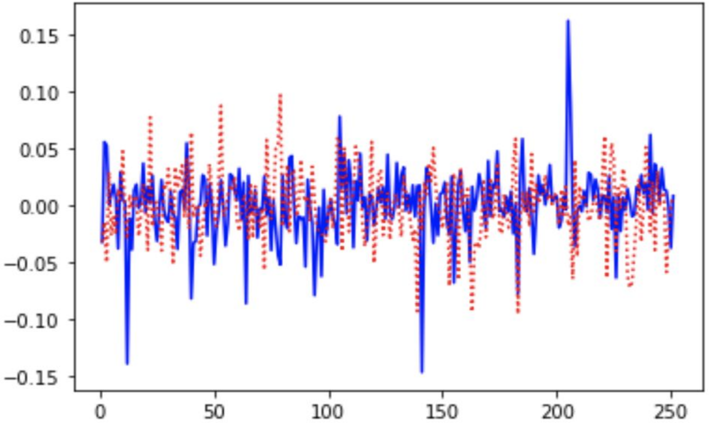
Brief Introduction

- Current GARCH(p,q):

$$r_t = \mu + \sigma_t \varepsilon_t$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i (r_{t-i} - \mu)^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2$$

- Simulated results with GARCH(p,q):



Brief Introduction Cont.

$$r_t = \mu + \sigma_t \varepsilon_t$$

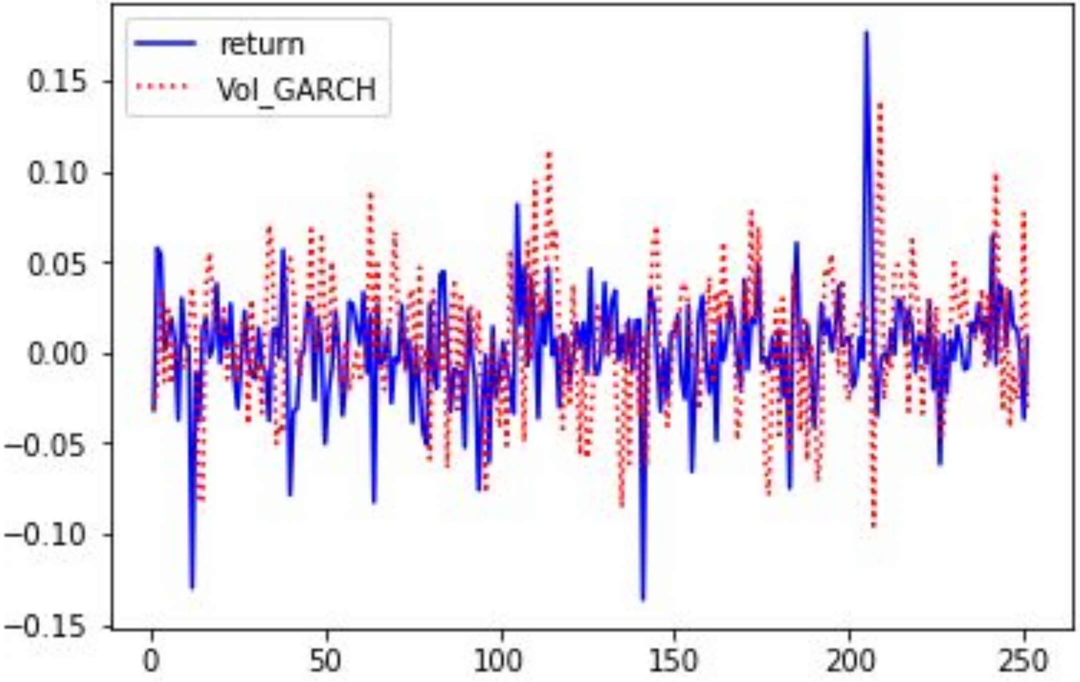
- Current GARCH(p,q): $\sigma_t^2 = \boxed{\omega} + \sum_{i=1}^{q} \alpha_i (r_{t-i} - \mu)^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2$

- Extend GARCH(p,q) with Volume:

$$\sigma_t^2 = \boxed{\omega * Volume_{t-1}} + \sum_{i=1}^{q} \alpha_i (r_{t-i} - \mu)^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2$$

Implementation

Simulated Result of Vol-GARCH(1,1)



Bootstrap Result

Yaksh Bhatt

Block Bootstrap

- We have a fixed time horizon, h.
- We pick a random start date, t
- We fit model from t to t+h and get the coefficients/parameters
- We repeat for n times with replacement
- We aggregate the coefficients/parameters
    - We averaged out the coefficients/parameters equally because the block is same for each strap

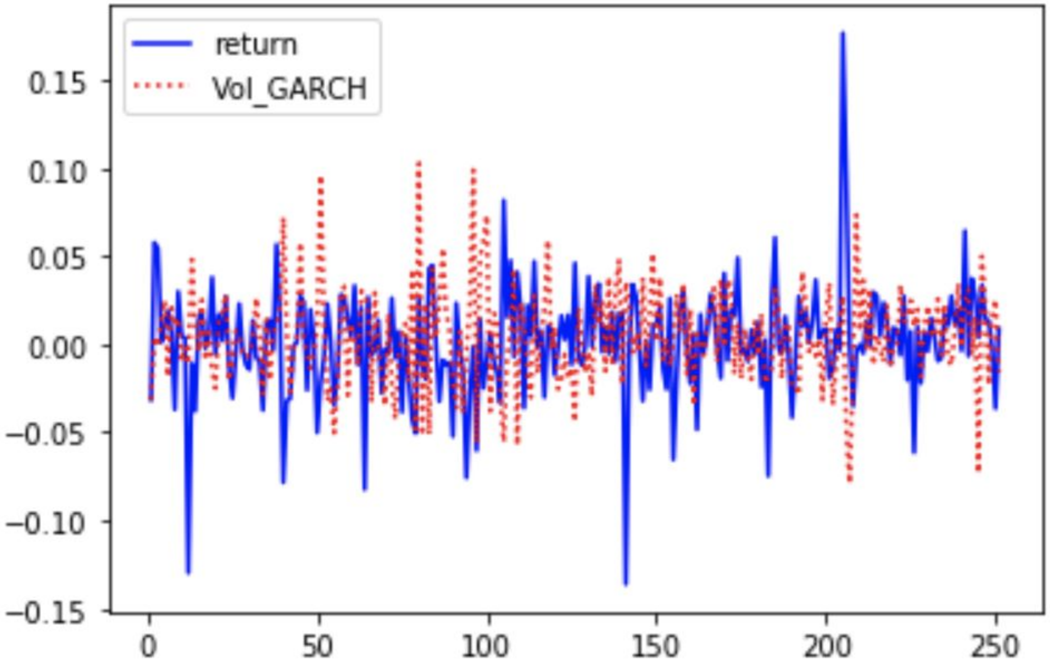
Results for Linear Models

- Bootstrap result:

	Mkt-Rf	SMB	HML	Oil Price	Lithium Price	Energy Price	Intercept
CAPM	1.20455385	-	-	-	-	-	6.9328481e-05
Fama-French	1.07306943	6.563e-03	4.7e-04	-	-	-	6.1894473e-04
APT	6.3702e-01	-	-	3.871184e-01	1.09589399e+00	-7.88479587e-01	4.2525501e-04

Results for Vol-GARCH

- Bootstrap result:



Next Steps

Yaksh Bhatt

Next Steps

- Expand time horizon to be ten years
  - Apply the first 8 years to be training data and apply with bootstrapping
  - expand the block accordingly
- Add LSTM into bootstrapping process
- Run for testing data