Analysis of Tesla

Yongyuan Qu

yongyuaq@uci.edu

And

Jiajun Gu

jiajung2@uci.edu

Abstract

Tesla is a popular company in recent years because of their innovative products, such as Model 3, Model X, Model Y, etc. The notion of Tesla is encouraging zero-carbon emission transportation and focusing on replacing fossil fuels with cleaning energies, like electricity. So, the stock price of Tesla increasing a lot and many investors care about this company. However, due to the COVID-19, the delivery of new vehicles is postponed, and the economy of the US is not very good. it will lead to the decline of the stock price of Tesla and many people begin to lose confidence. In 2022, the business of Tesla begins to rebound, and our team wants to create serval models to predict the future price of Tesla by using the historical volatility and factors. In order to ensure the validity of our models, we would also build risk models: Value at Risk to measure the different risks. Therefore, we want to investigate the best relationship between **Future Stock prices** with **proper risk**. Based on this situation, several models are established: Model I: Multi-Factor Model; Model II: Multi-Regression Model; Model III: LSTM Model.

For Model I, we try to select several **features** to create the multi-factor models. We choose SMB, HML, and Momentum as the three main factors to create the classic Fama-French three factors model. Small Minus Big (SMB) is a size effect based on the market capitalization of a company. High Minus Low (HML) is a value premium; it represents the spread in returns between companies with a high book-to-market value ratio and companies with a low book-to-market value ratio. Momentum (Mom) is "Winner minus loser", with stocks that have outperformed in the past tend to exhibit strong returns going forward.

For Model II, Time Series Model is related to predicting future outcomes, understanding past outcomes, making policy suggestions, and much more. These general goals of time series modeling don't vary significantly from modeling cross-**sectional or panel data**. Time series forecasting occurs when you make scientific predictions based on historical time stamped data. It involves building models through historical analysis and using them to make observations and drive future strategic decision-making.

For Model III, long short-term memory (LSTM) normally applies **recurrent neural network** (RNN) as a basic recurrent unit. However, conventional LSTM assumes that the state at the current time step depends on the previous time step. This assumption constraints the time dependency modeling capability.

For sensitivity analysis, we use the rolling mean and standard deviation of the previous returns to generate the return of the current day and we can get the percentage change of the **rolling window**, which can help us to find the sensitivity to change in the current returns.

After three models, we will make a conclusion for readers and propose our results by interpretations.

Keywords: Feature Engineering, Sectional or Panel Data, Recurrent neural network, Rolling window

Contents

1 Introduction

1.1 Background of Tesla

Tesla was founded in 2003 and the company wants to fully achieve the self-drive technology and get rid of the traditional cars which depend on fossil fuels. Tesla believes that electric vehicles would have better performance than traditional cars. In recent years, Tesla is a more and more popular product in the world and many people begin to choose tesla as their choice.

(a) Tesla Model X

(b）Elon Musk

Figure 1: Target: (a) **Tesla Model X**: Model X is the best SUV to drive, and the best SUV to be driven in. Clean, powerful yet invisible cabin conditioning; (b) **Elon Musk**: He is the founder, CEO, and Chief Engineer at SpaceX; angel investor, CEO, and Product Architect of Tesla, Inc.; founder of The Boring Company; and co-founder of Neuralink and OpenAI.

1.2 Approach

To forecast the future performance of Tesla, we need to clean the data first, then backtest the volatility of the past by applying time series. Finally, we built models to forecast and create risk management models.

1) Standardize the dataset with time series.
2) Backtest the volatility
3) Apply ARCH, GARCH, and multi-factor models
4) Apply LSTM model to predict the price
5) Combine the regression with the validity of the model
6) Create Value at Risk to measure the different risks
7) Based on the results of our mode, we make a conclusion for readers

2 Our Assumption and Model Overview

Since the forecasting stock price always is affected by many factors, we want to simplify this process and the followings are our assumptions:

- **Assumption 1:** Assume there is no transaction cost during trading and all information is frictionless.

  Justification: We would assume transaction cost is zero in our model because it would a dummy variable and will causes the mean squared error. Information is also frictionless, which each investor can get the same news of the stock and prevent the effects of insider trading to cause the fluctuation of the stock.

- **Assumption 2:** ignore any political policy in the market that may dramatically affect the price fluctuation of Tesla.

  Justification: We believe that policy is an unstable factor that can severely affect our models.

- **Assumption 3:** Assume that there is no worldwide economic crisis that may affect the price of Tesla from 2022.5 to 2023.5.

  Justification: Economic crisis will lead to a big fluctuation in the price of bitcoin and gold and other dummy variables will appear.

- **Assumption 4:** Assume that there are some inter-relationships between the price of volatility and stock price.

  Justification: Such that we can predict the volatility of the stock using some of the information of Tesla.

- **Assumption 5:** There exists a potential linear relationship between the prior volatility of Tesla and the stock price between the current volatility of Tesla and its stock price.

  Justification: This assumption is just for the regression model.

We firstly use the ARIMA model to make our data flatten, then apply multi-factor models to find the volatility and the expected price. Next, we would apply regression test to check the validity and build our risk management techniques. Finally, we would forecast the future price of Tesla and make backtest.

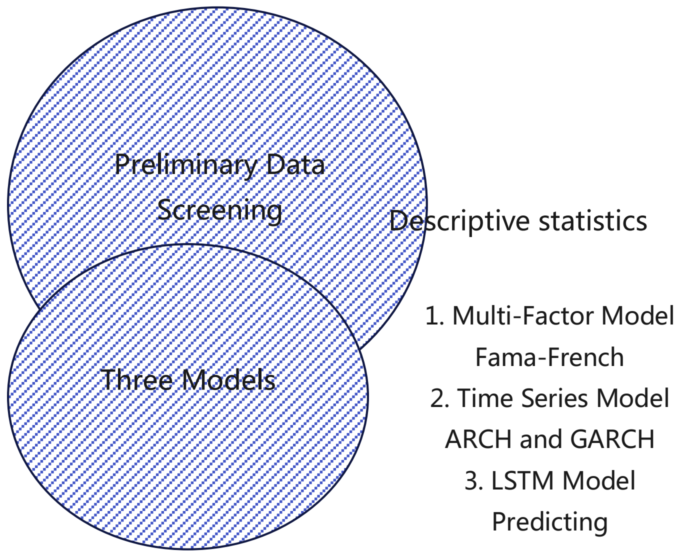


Figure 2: Model Overview

3 Model Preparation

3.1 Notation

Table 1: Notations

Symbol	Description
Y_i-Tesla	ith return of Tesla
S_i-Tesla	Skew of Tesla
$\widehat{Y}_i$-Tesla	Predicted ith return of Tesla
K_i-Tesla	Kurtosis of Tesla
Xstd_Tesla	Standard deviation of Tesla
Wt	Weight of Tesla

Table 2: Other Symbols

Symbol	Description
MAE	Mean absolute Error
RMSE	Root Mean Squared error
MSE	Mean Squared error

Notes: we will also discuss other variables in the specific parts.

3.2 Data Overview

3.2.1 Data Collection

Since we are going to sacrifice some datasets during training the data, we are going to collect the relevant stock price from Yahoo Finance.

Table 3: Data Collection

Database	Data Sources	Data Type
Tesla	NASDAQ	Float
Benchmark	S&P 500	Float

3.2.2 Data Screening

In this problem, we mainly use the **Expected Return, Standard Deviation, Skewness, and Kurtosis** of Tesla to see the preliminary result. We also have calculated these indicators of the databases.

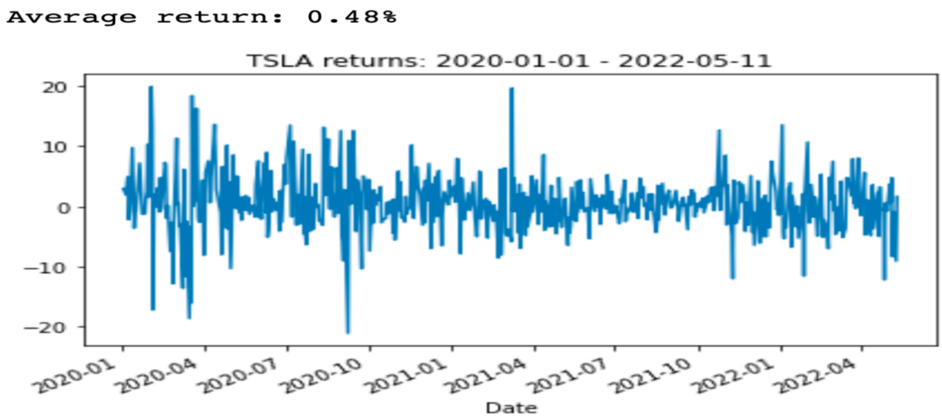


Figure 3: Distribution of the stock price

First, it is important to standardize the data and get the annual expected return of Tesla is 1.15 and the annual standard deviation is 1.13. The skewness and kurtosis of Tesla are 0.088 and 3.05 which is leptokurtic with fat tails.

$$\text{Skewness} \approx \left(\frac{1}{n}\right) \frac{\sum_{i=1}^{n} (X_i - \overline{X})^3}{s^3}.$$

Skewness is a quantitative measure of skew (lack of symmetry), a synonym of skew. It is computed as the average cubed deviation from the mean standardized by dividing by the standard deviation cubed

$$K_E \approx \left[\left(\frac{1}{n}\right) \frac{\sum_{i=1}^{n} (X_i - \overline{X})^4}{s^4} \right] - 3.$$

Kurtosis is a measure of the combined weight of the tails of distribution relative to the rest of the distribution.

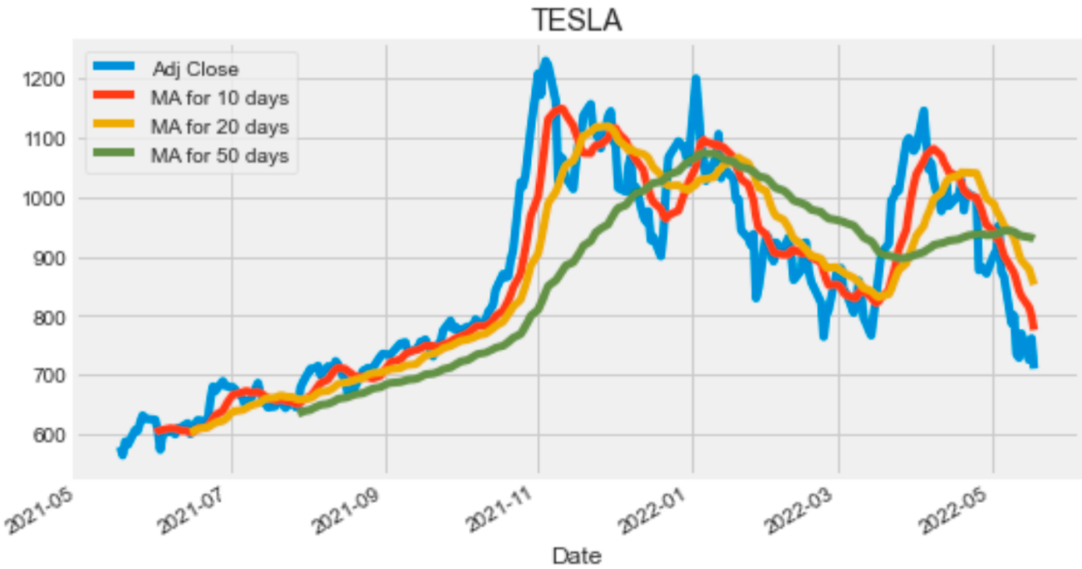


Figure 4: Time Series Stock Price Visualization of Tesla

Moving averages: it is usually a useful tool to calculate to identify the trend direction of a stock or to determine its support and resistance levels. It is a trend-following—or lagging—indicator because it is based on past prices. The longer the period for the moving average, the greater the lag.

According to the graph, we can see the price of Tesla very fluctuates and it reaches its highest point in November 2021 and begin to go down in the following two months. It has a little bit of rebound but still has a decreasing tendency.

4. Model I: Multi-Factor Model

4.1 Model Establishment

In this model, we are trying to select several features from our price datasets and find out if there is a relationship between these features and the ultimate prices. We are going to construct a Fama-French model to solve this problem. We choose SMB, HML, and Momentum as three factors. The first step is the data collection, since we are going to sacrifice some datasets during training the data, we are going to collect the relevant price from Yahoo Finance.

4.2 Test the Model

4.2.1. Fama-French Three Factors Model

Small Minus Big (SMB) is a size effect based on the market capitalization of a company. SMB measures the historic excess of small-cap companies over big-cap companies and It is a factor in the Fama/French stock pricing model that says whether smaller companies outperform larger ones over the long term. When small stocks have good performance than large stocks, the number will be positive, vice versa.

High Minus Low (HML) is a value premium; it represents the spread in returns between companies with a high book-to-market value ratio and companies with a low book-to-market value ratio.

Momentum (Mom) is "Winner minus loser", with stocks that have outperformed in the past tend to exhibit strong returns going forward. A momentum strategy is grounded in relative returns from three months to a one-year time frame.

4.2.2 Model Evaluation

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.4432	0.168	2.632	0.009	0.113	0.774
Mkt-RF	1.4595	0.103	14.212	0.000	1.258	1.661

Omnibus:	62.140	**Durbin-Watson:**	2.039
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	329.467
Skew:	0.287	**Prob(JB):**	2.86e-72
Kurtosis:	6.693	**Cond. No.**	1.64

Figure 9: One Factor Model Data

	const	Mkt-RF	SMB	HML	Mom
TSLA	0.443201	1.439264	0.952868	-0.659393	0.271124
^GSPC	-0.003844	0.989218	-0.148967	0.035644	-0.003403
^IRX	0.527907	0.365937	-13.691798	1.496068	-0.728339

Figure 10: Multi-Factor Model

From the above result, it is easy for us to see that one matter under the one-factor model (CAPM) and Fama-French three factors model, tesla has a higher risk level when compared with the market. From the four factors analysis, we can see the result is a little bit different from only including one factor of market risk premium because it is more accurate when incorporating other factors inside and that will cause some deviation as well. By estimating past returns during the period of covid, it has a constant coefficient greater than 0 and it is a way of measuring risk-adjusted performance (Tesla has 0.4432% abnormal return over the market per day), indicating that it performs well over the market. For the coefficient of risk premium, they display high volatility than the market (greater than 1) which makes sense due to a higher return and it displays a higher accuracy when we have incorporated more factors, we have a more accurate approximation of beta relative to the market. Tesla's small minus big factor is negative, suggesting it is a stock with a small market capitalization that performs well in comparison to the market overall. In high minus low coefficients, Tesla shows the excess return is from the company's small book-to-market equity value, behaving more like one of a growth stock.

4.3 Model Backtest

Multiple Regression Assumptions:

1st Durbin-Watson serial correlation test (no serial correlation in the residual)

2nd Breusch Pagan Heteroskedasticity (Error variances are all equal)

3rd Check of Multicollinearity Problem (no indication of perfect multicollinearity problem)

Table 6: Critical Statistic

Statistics indicators	Value
Annual_return_Tesla	1.123118
Daily_Average_return_Tesla	0.004492
Annual_Average_risklevel_Tesla	1.130734
Daily_Average_risklevel_Tesla	0.046695
skew_TSLA	0.090437
Kurt_TSLA	3.027286
Prob_negative_TSLA_return	0.46167718

From the above annual statistics indicators, we can have a clear sense that Tesla's return distribution follows an approximate normal pattern. After the occurrence of the covid pandemic, the tesla performance level is not decreasing, but on the contrary, it follows a relatively high return together with high risks.

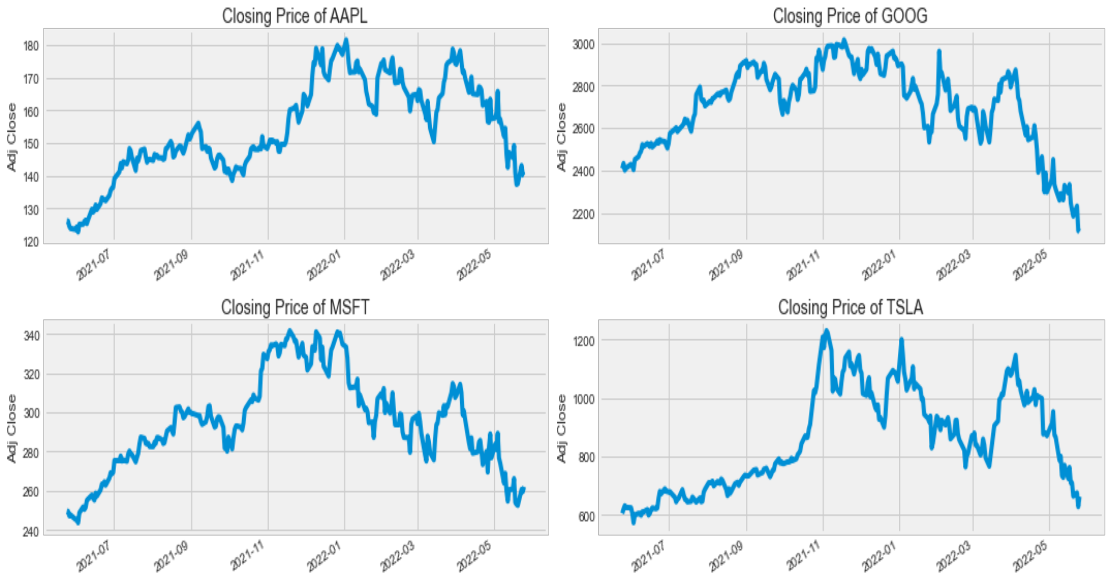


Figure 11: Different Stock price prediction

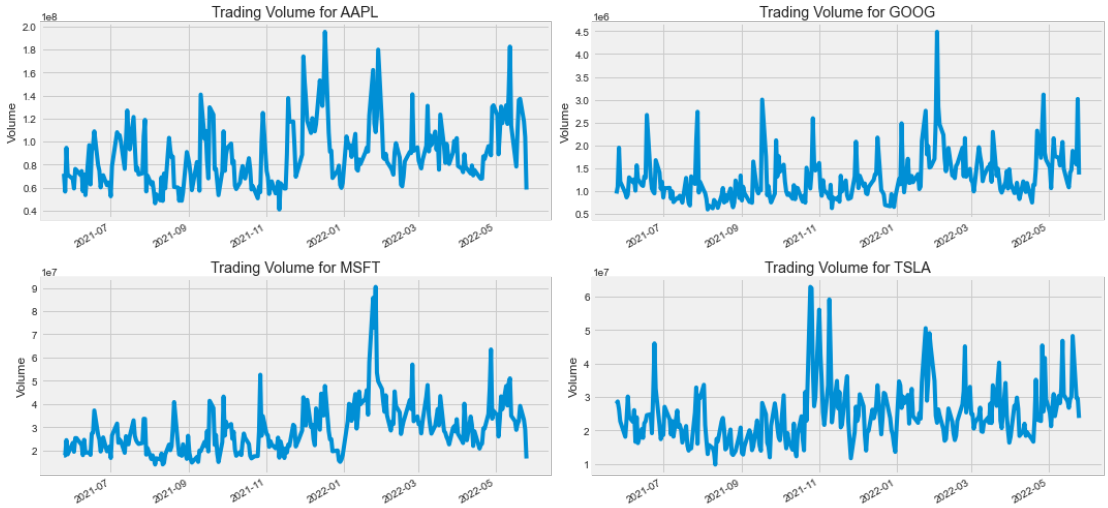


Figure 12: Different Stock Trading Volume

Other indicators in comparison with stocks within the high-tech industry

After basic descriptive statistics analysis, we can shift our focus to other indicators to broadly acknowledge the overall performance level of Tesla company.

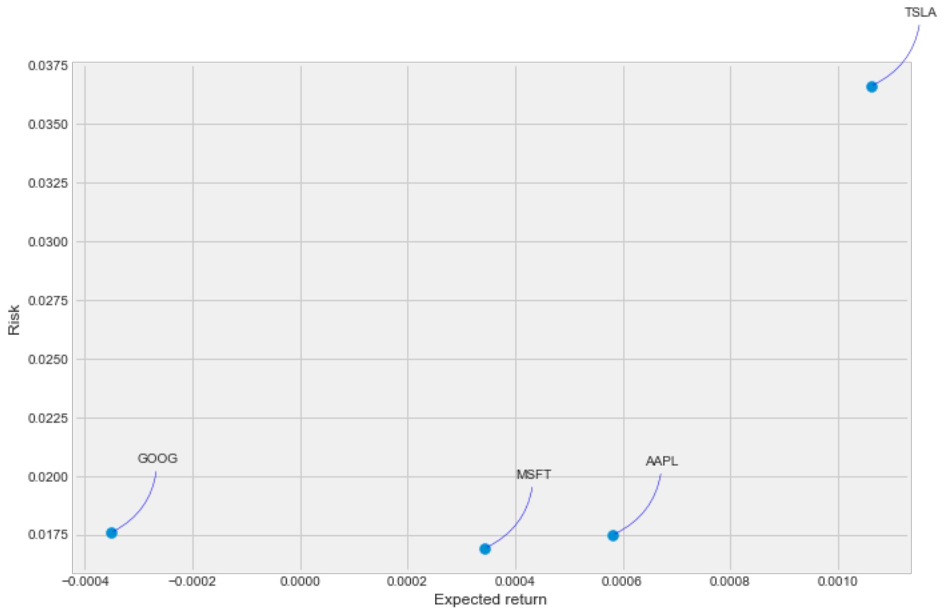


Figure 13: Different Stock Expected Return relative to risk level

When compared within the industry, we can see that Tesla has the second-highest closing stock price with strong volatility which indicates Tesla is relatively a risky stock susceptible to market that is suitable for risky inclined investors. Also, Tesla has the highest trading volume maintained with high-frequency trading each day even after covid, indicating its potential investors to its future growth perspective. However, from the matrix table of the return and risks, we can see that Tesla has an average higher expected return with a higher risk level daily, which also confirms our initial descriptive statistics that tesla is a high growth stock but also filled with potential risks as well.

5 Model II: ARCH & GARCH

5.1 ARCH Model- Autoregressive Conditional Heteroskedasticity

Assumptions: The expected value of all error terms when squared is the same at any given point

Conditions: p=1, o=0, q=0

Autoregressive Conditional Heteroskedasticity (ARCH) is a method that explicitly models the change in variance over time in a time series, specifically, it models the variance at a time step as a function of the residual errors from a mean process.

```
                         Zero Mean - ARCH Model Results
==============================================================================
Dep. Variable:          asset_returns   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.002
Vol Model:                       ARCH   Log-Likelihood:               -1747.86
Distribution:                  Normal   AIC:                           3499.72
Method:            Maximum Likelihood   BIC:                           3508.49
                                        No. Observations:                  593
Date:                Sun, May 22 2022   Df Residuals:                      593
Time:                        23:02:26   Df Model:                            0
                              Volatility Model
==========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
--------------------------------------------------------------------------
omega         19.1156      2.119      9.022  1.839e-19   [ 14.963, 23.268]
alpha[1]       0.1202  5.859e-02      2.052  4.020e-02 [5.372e-03,  0.235]
==========================================================================

Covariance estimator: robust
```

Figure 14: ARCH Model Result

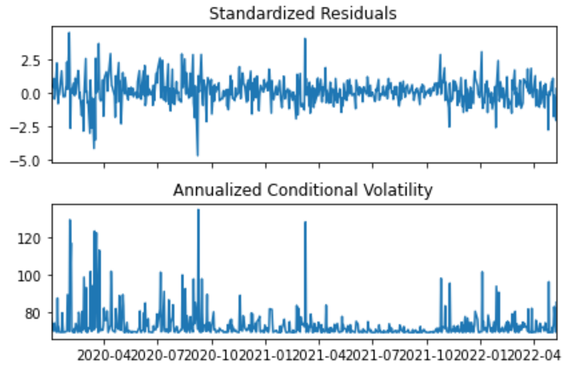


Figure 15: Annualized Conditional Volatility

5.2 GARCH Model- Generalized Autoregressive Conditional Heteroskedasticity

Conditions: p=1, o=0, q=0

Assumptions: The expected value of all error terms when squared is the same at any given point

Generalized Autoregressive Conditional Heteroskedasticity (GARCH) is an extension of the ARCH model that incorporates a moving average component together with the autoregressive component. Specifically, the model includes lag variance terms (e.g. the observations if modeling the white noise residual errors of another process), together with lag residual errors from a mean process.

The moving average component allows the model to both models the conditional change in variance over time as well as changes in the time-dependent variance. Examples include conditional increases and decreases invariance.

p: The number of lag variances to include in the GARCH model.

q: The number of lag residual errors to include in the GARCH model.

Alpha (ARCH term) represents how volatility reacts to new information

Beta (GARCH Term) represents the persistence of the volatility

omega (the variance intercept)

Formula Equation:

$r_t = \mu + \epsilon_t$

$\sigma^2 = \omega + \alpha\epsilon^2 + \beta\sigma^2$

$_{t\ t-1\ t-1}$ $\epsilon_t = \sigma_t e_t,\ e_t \sim N(0,1)$

```
                       Zero Mean - GARCH Model Results
==============================================================================
Dep. Variable:          asset_returns   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.002
Vol Model:                      GARCH   Log-Likelihood:               -1709.91
Distribution:                  Normal   AIC:                           3425.82
Method:            Maximum Likelihood   BIC:                           3438.97
                                        No. Observations:                  593
Date:                Sun, May 22 2022   Df Residuals:                      593
Time:                        23:02:27   Df Model:                            0
                              Volatility Model
===========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
---------------------------------------------------------------------------
omega          0.4306      0.519      0.830      0.407    [ -0.586,  1.448]
alpha[1]       0.0936  5.033e-02      1.859  6.307e-02 [-5.097e-03,  0.192]
beta[1]        0.8907  6.135e-02     14.519  9.161e-48    [  0.770,  1.011]
===========================================================================

Covariance estimator: robust
```

Figure 16: GARCH Model Result

From the GARCH model results, we can see that alpha and beta coefficients are both significant at a 5% significance level, we can see that the overall volatility level of tesla, As the GARCH coefficient value is higher than the ARCH coefficient value, we can conclude that the volatility is highly persistent and clustering. Also, the graph at the bottom can better illustrate the volatility level of tesla stock as well.

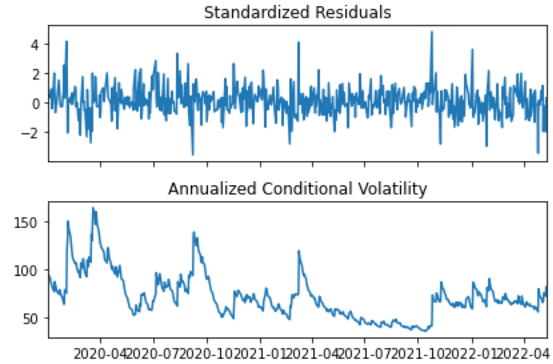


Figure 17: Annualized Conditional Volatility

5.3 Results：

From both reflected ARCH and GARCH model that incorporates a moving average and autoregressive component, we can both see that the volatility level is relatively high since the sum of alpha and beta coefficient approaches one, and Tesla's return performance is not stable overall the pandemic period, where an investor should be paid attention for future investment

6. Model III: LSTM Model

6.1 Model Structure

Long short-term memory (LSTM) units (or blocks) are a building unit for layers of a recurrent neural network (RNN). An RNN composed of LSTM units is often called an LSTM network. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM. Each of the three gates can be thought of as a "conventional" artificial neuron, as in a multi-layer (or feedforward) neural network: that is, they compute an activation (using an activation function) of a weighted sum. Intuitively, they can be thought as regulators of the flow of values that goes through the connections of the LSTM; hence the denotation "gate". There are connections between these gates and the cell.

The expression long short-term refers to the fact that LSTM is a model for short-term memory which can last for a long period of time. An LSTM is well-suited to classify, process, and predict time series given time lags of unknown size and duration between important events. LSTMs were developed to deal with the exploding and vanishing gradient problem when training traditional RNNs.

6.2 Predicting the price

For the Tesla stock, we choose 95% as the training row

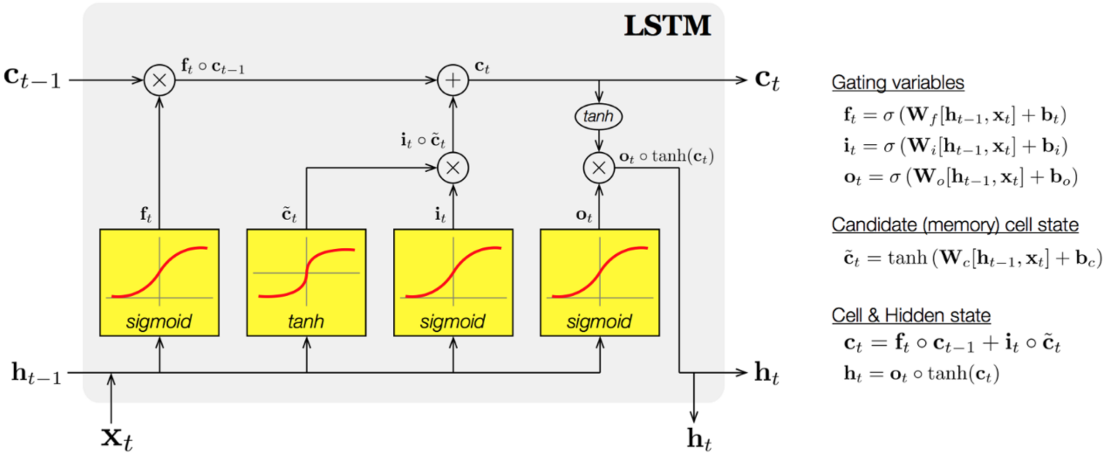


Figure 18: Training Row

Figure 19: Closing Price History

Figure 20: Closing Price Prediction within two months

Figure 21: Tesla Stock price before and after covid

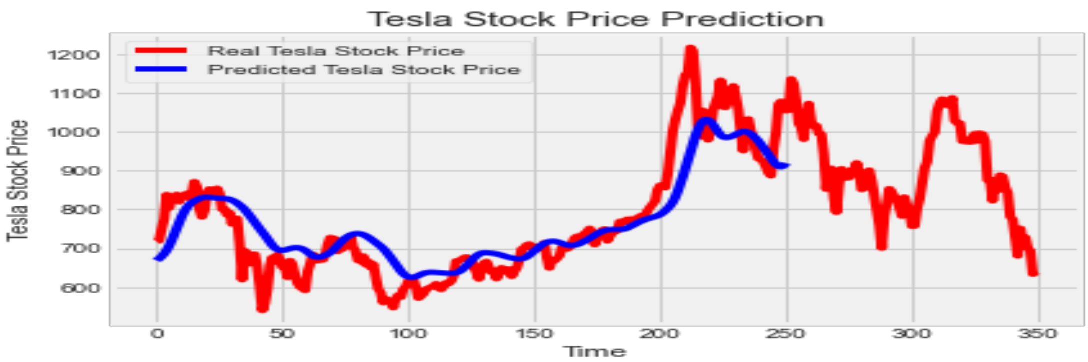


Figure 22: Predicted stock price over time

From the LSTM model, we select the data from within 60 days as the trained objects, and we can see that the predicted value is closer to the actual closing price, which suggests that the TSLM model does not fit very well due to the overall volatility level and its actual growing stages.

7. Risk Management

7.1 Value at Risk

The market changes over time and we can't depend on a single model or strategy to do the investments. In addition, we need to consider the risk management of our portfolio. We believe that we can apply VaR (Value at Risk) and ES (Expected Shortfall) to further measure our tail risk of the portfolio. According to Extreme Value Theory, tails can be Fréchet, Gumbel, and Weibull distribution and we could base on the corresponding distributions to backtest the VaR. If we make sure the specific VaR, we can map VaR to the stock, which can assist us to understand the risk metrics.

For the new energy vehicles industry, we choose NIO as a benchmark to compare with the return of Tesla and calculate the correlation is 0.59. We choose 1 year as time horizon.

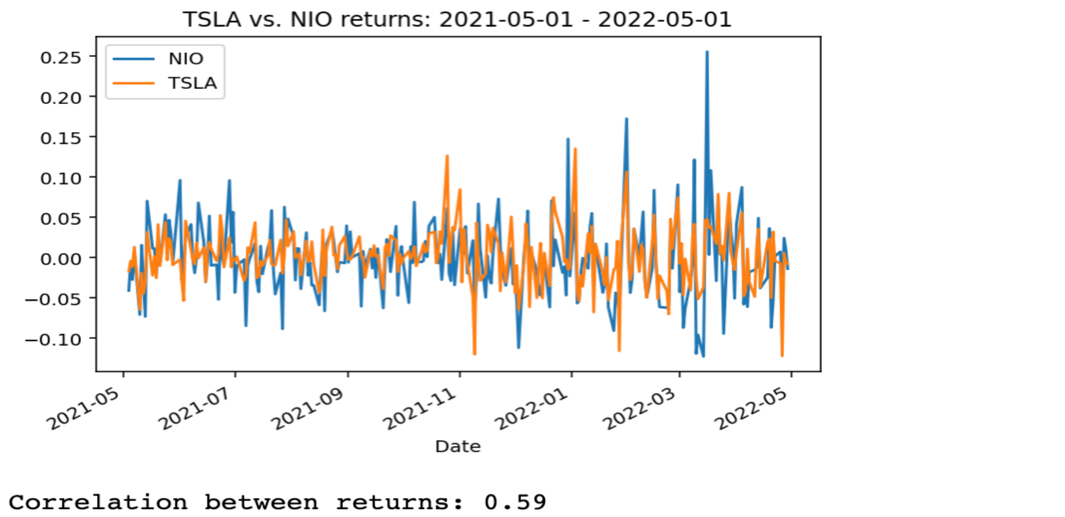


Figure 23: TSLA vs. NIO returns

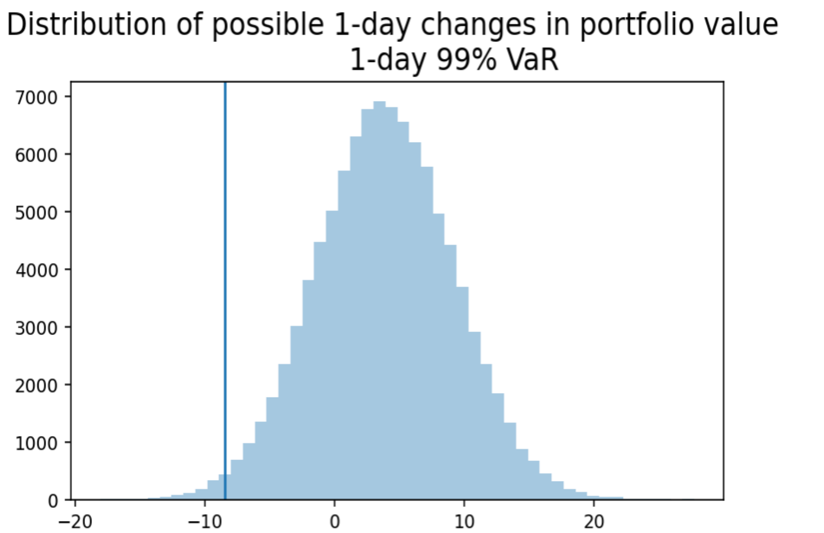


Figure 24: 1-day 99% VaR

Then, we use Covariance Matrix and Cholesky inequality to calculate VaR. We can see 1-day VaR with 99%, 99.9%, and 99.99% confidence intervals are 8.45$, 12.37$, and 15.26$. For the 1-day 99% VaR, it means when you invest 100$ in Tesla, there is a 1 percent probability that the loss will be greater than 8.45$ in one day. Therefore, we encourage investors to use other financial instruments to hedge the exposure.

7.2. Sensitivity Analysis

We conduct sensitivity analysis on our models. Since, for the multi-factor and the time series model, we only have the data for the everyday return. Therefore, our input variable is the rolling mean and standard deviation of the previous returns, and our target variable is the return of the current day. Since we are using rolling windows to make predictions, we can say that the percentage change of the rolling window (i.e., the input variable) will have a percentage change in the same direction on the current return (i.e., the target variable). Therefore, the sensitivity analysis indicates that the previous rolling window of returns is highly sensitive to changes in current returns.

$$Y_{Asset} = \propto_{ret-5-Asset} * \ ret-5-asset + \propto_{std-5-Asset} * \ std-5-asset + \propto_{ret-30-Asset} * \ ret-30-asset + \propto_{std-30-Asset} * \ std-30-asset + \beta_{asset}$$

8 Conclusion

8.1 Strengths

- After three models' testing, it will give us insights into the prediction of investment strategy from past data.
- We have established a relationship between the proportion invested of different assets and respective volatility and return and consider many kinds of situations in line with the actual situation. Therefore, the model has excellent universality and flexibility
- We have developed a series of models and suggestions on the effectiveness of testing the performance of the stock.

8.2 Weakness

- The time series model is strongly relying on past data performance, and it could have interference by harming the overall data accuracy
- The commission fee is not a fixed amount and depends on each transaction amount during trading while given by the fixed percentage, so it can influence the factors behind applying directly to three models

References

1] https://github.com/PacktPublishing/Python-for-Finance-Cookbook/blob/master/Chapter%2003/chapter_3.ipynb

[2] https://github.com/PacktPublishing/Python-for-Finance-Cookbook/blob/master/Chapter%2005/chapter_5.ipynb

[3] https://github.com/PacktPublishing/Python-for-Finance-Cookbook/blob/master/Chapter%2003/chapter_3.ipynb

[4] https://www.kaggle.com/code/faressayah/stock-market-analysis-prediction-using-lstm/notebook

[5] https://blogs.oracle.com/connect/post/python-deep-learning-understand-data

[6] https://analyzingalpha.com/interpret-arima-results

[7] https://corporatefinanceinstitute.com/resources/knowledge/other/durbin-watson-statistic/

[8] https://machinelearningmastery.com/develop-arch-and-garch-models-for-time-series-forecasting-in-python/

[9] https://www.kaggle.com/code/thebrownviking20/intro-to-recurrent-neural-networks-lstm-gru/notebook

Appendices

Appendix 1

Prediction Stock price value

Date	Close	Predictions
2022-04-11	975.929993	1086.705933
2022-04-12	986.950012	1071.481934
2022-04-13	1022.369995	1055.916504
2022-04-14	985.000000	1045.652222
2022-04-18	1004.289978	1035.076538
2022-04-19	1028.150024	1028.001831
2022-04-20	977.200012	1026.076294
2022-04-21	1008.780029	1021.440918
2022-04-22	1005.049988	1019.725098
2022-04-25	998.020020	1019.076904
2022-04-26	876.419983	1018.347534
2022-04-27	881.510010	1004.399109
2022-04-28	877.510010	986.232849
2022-04-29	870.760010	967.079834
2022-05-02	902.940002	948.693848
2022-05-03	909.250000	936.228638
2022-05-04	952.619995	928.825073
2022-05-05	873.280029	929.725098
2022-05-06	865.650024	926.209595
2022-05-09	787.109985	920.362061
2022-05-10	800.039978	905.276611
2022-05-11	734.000000	888.538818
2022-05-12	728.000000	865.320312
2022-05-13	769.590027	840.678467
2022-05-16	724.369995	822.201660
2022-05-17	761.609985	803.966248
2022-05-18	709.809998	792.021545
2022-05-19	709.419983	779.046814
2022-05-20	663.900024	767.055969
2022-05-23	671.599976	751.907227

Appendix 2

VaR

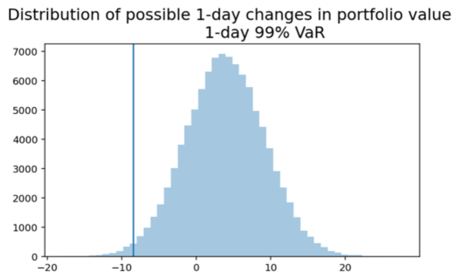
In [62]:
```
P_diff_sorted = np.sort(P_diff)
percentiles = [0.01, 0.1, 1.]
var = np.percentile(P_diff_sorted, percentiles)

for x, y in zip(percentiles, var):
    print(f'1-day VaR with {100-x}% confidence: {-y:.2f}$')
```

```
1-day VaR with 99.99% confidence: 15.26$
1-day VaR with 99.9% confidence: 12.37$
1-day VaR with 99.0% confidence: 8.45$
```

In [64]:
```
ax = sns.distplot(P_diff, kde=False)
ax.set_title('''Distribution of possible 1-day changes in portfolio value
             1-day 99% VaR''', fontsize=16)
ax.axvline(var[2], 0, 10000)

plt.tight_layout()
plt.show()
```



In [65]:
```
var = np.percentile(P_diff_sorted, 5)
expected_shortfall = P_diff_sorted[P_diff_sorted<=var].mean()

print(f'The 1-day 95% VaR is {-var:.2f}$, and the accompanying Expected Shortfall is {-expected_shortfall:.2f
```

```
The 1-day 95% VaR is 4.81$, and the accompanying Expected Shortfall is 7.02$.
```

Appendices 3: Important Code

Economic model Prediction

First part-Volatility Forecasting

In [58]:
```
#Company performance stock price
```

Autoregressive conditional heteroskedasticity (ARCH) is a statistical model used to analyze volatility in time series in order to forecast future volatility.

In the ARCH(q) process the conditional variance is specified as a linear function of past sample variances only, whereas the GARCH(p, q) process allows lagged conditional variances to enter as well.

In [59]:
```
# Volatility Forecasting Explaining stock returns' volatility with ARCH models
```

In [60]:
```
model = arch_model(returns, mean='Zero', vol='ARCH', p=1, o=0, q=0)
model_fitted = model.fit(disp='off')
print(model_fitted.summary())
```

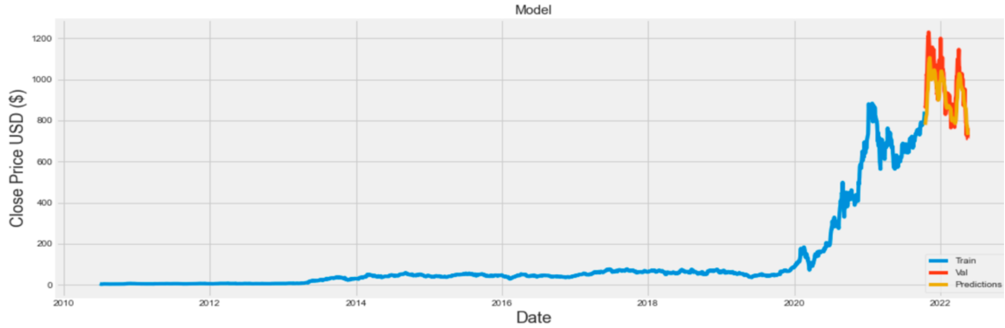
```
                        Zero Mean - ARCH Model Results
==============================================================================
Dep. Variable:          asset_returns   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.002
Vol Model:                       ARCH   Log-Likelihood:               -1747.86
Distribution:                  Normal   AIC:                           3499.72
Method:            Maximum Likelihood   BIC:                           3508.49
                                        No. Observations:                  593
Date:                Thu, May 19 2022   Df Residuals:                      593
Time:                        16:08:34   Df Model:                            0
                             Volatility Model
==========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
--------------------------------------------------------------------------
omega         19.1156      2.119      9.022  1.839e-19   [ 14.963, 23.268]
alpha[1]       0.1202  5.859e-02      2.052  4.020e-02 [5.372e-03,  0.235]
==========================================================================

Covariance estimator: robust
```

In [156]:
```
# Plot the data
train = data[:training_data_len]
valid = data[training_data_len:]
valid['Predictions'] = predictions
# Visualize the data
plt.figure(figsize=(16,6))
plt.title('Model')
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price USD ($)', fontsize=18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Val', 'Predictions'], loc='lower right')
plt.show()
```

```
<ipython-input-156-0cdd5e68a3a9>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  valid['Predictions'] = predictions
```



In [185]:
```
# Create the testing data set
# Create a new array containing scaled values from index 1543 to 2002
test_data = scaled_data[training_data_len - 60: , :]
# Create the data sets x_test and y_test
x_test = []
y_test = dataset[training_data_len:, :]
for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0])

# Convert the data to a numpy array
x_test = np.array(x_test)

# Reshape the data
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1 ))

# Get the models predicted price values
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)

# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
rmse
```

```
1/1 [==============================] - 1s 863ms/step
```

Out[185]:
```
202.70353657330176
```