NASDAQ Composite Index Stock Prediction

By Carol Ho, Jeffrey Cheng & Naomi Tsang

AGENDA

BRAINSTORMING
Financial Analysis

- What is the business value?
- Which stocks we want to predict?
- What data are the must for analysing?
- How do we get the data?

DATA COLLECTION & PREPROCESSING
Yahoo Finance & Market Insiders

Obtaining relevant information and generating insights via EDA.

MODEL CREATION
Classification & Time Series

Deploy Classification and Time Series models to predict future trend and stock price.

EVALUATION
Limitations and Improvements

Highlight limitations of the models and understand other factors that may affect the results.

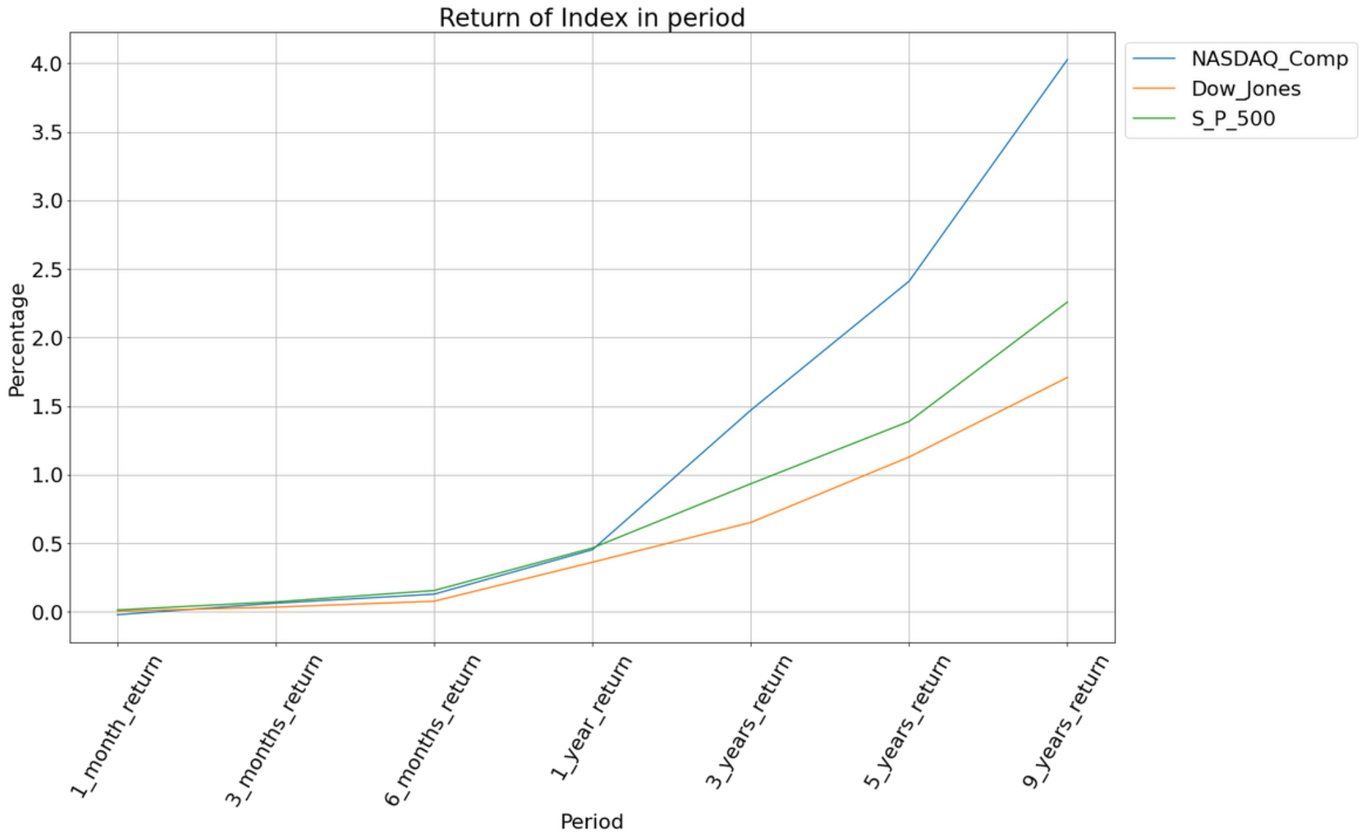
PROJECT AIM

PLANNING

BRAINSTORMING

Financial Analysis

- What is the business value?
- Which stocks we want to predict?
- What data are the must for analysing?
- How do we get the data?



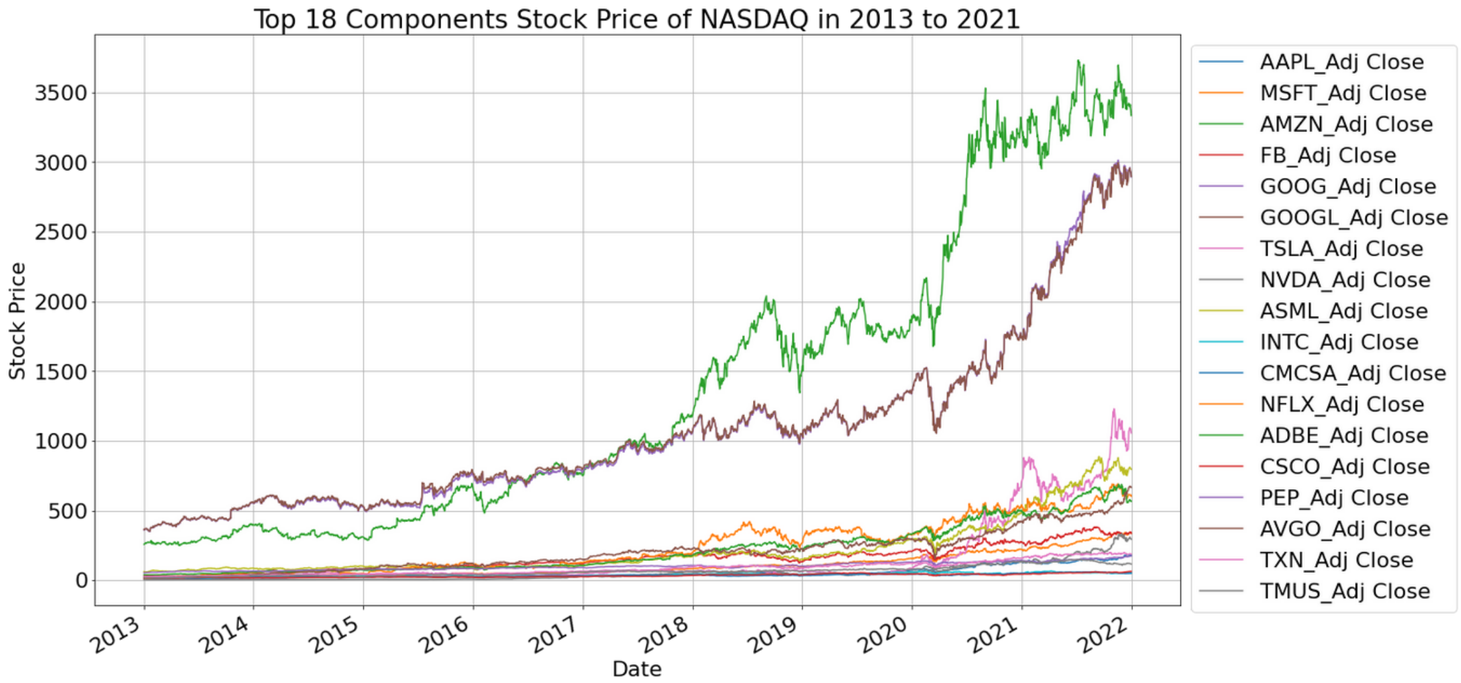
BUSINESS VALUE

POTENTIAL USAGE

Major Stakeholders	Potential applications
Private Investors	Find out the potential stocks with the predicted stock price of the next day
Finance Reporter	Find out predicted stock price of the next day to support their article research and plannings
Corporate Investors - Banks, Fund Houses	Find out the prediction of the stock price of the next day by the small private investor and do the opposite to earn money

Data Collection

Top 18 Components Stocks of NASDAQ



- PDD & PYPL were taken out as only too few record for analysis

Currency:

CNY to USD
GBP to USD
JPY to USD
EUR to USD
CAD to USD

Commodities:

Gold & Oil

Index:

NASDAQ Composite
Dow Jones
S&P 500

Treasury Bills

13 Week Treasury Bill, Treasury Yield 5 Years, Treasury Yield 10 Years, Treasury Yield 30 Years

Data Preprocessing & Cleaning

1) Understanding the data

2) Import libraries and all csv files

- pandas, re, matplotlib.pyplot, etc.

3) Create new useful columns per dataframe

- e.g. Daily_Change, Daily_Return

4) Renaming the column name per dataframe

- e.g. Daily_Change to APPL_Daily_Change

5) Combine all dataframes to a big dataframe

6) Convert Date column

- pd.to_datetime

7) Create useful columns for EDA

- e.g. Year, Month, DayOfWeek

8) Drop all rows with null

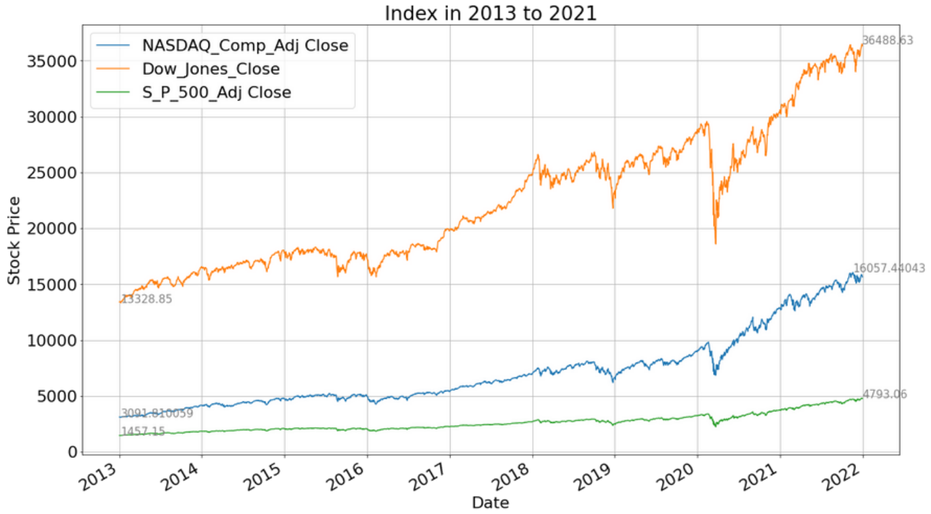
- e.g. the Treasury Bills data contain the date in Saturday with null data

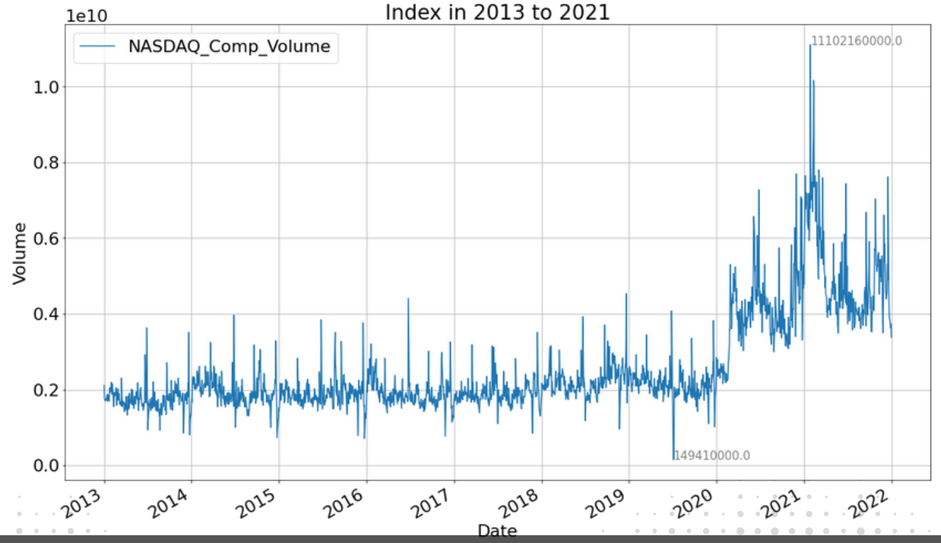
9) Prepare the real time data code for models creation with yfinance libraries

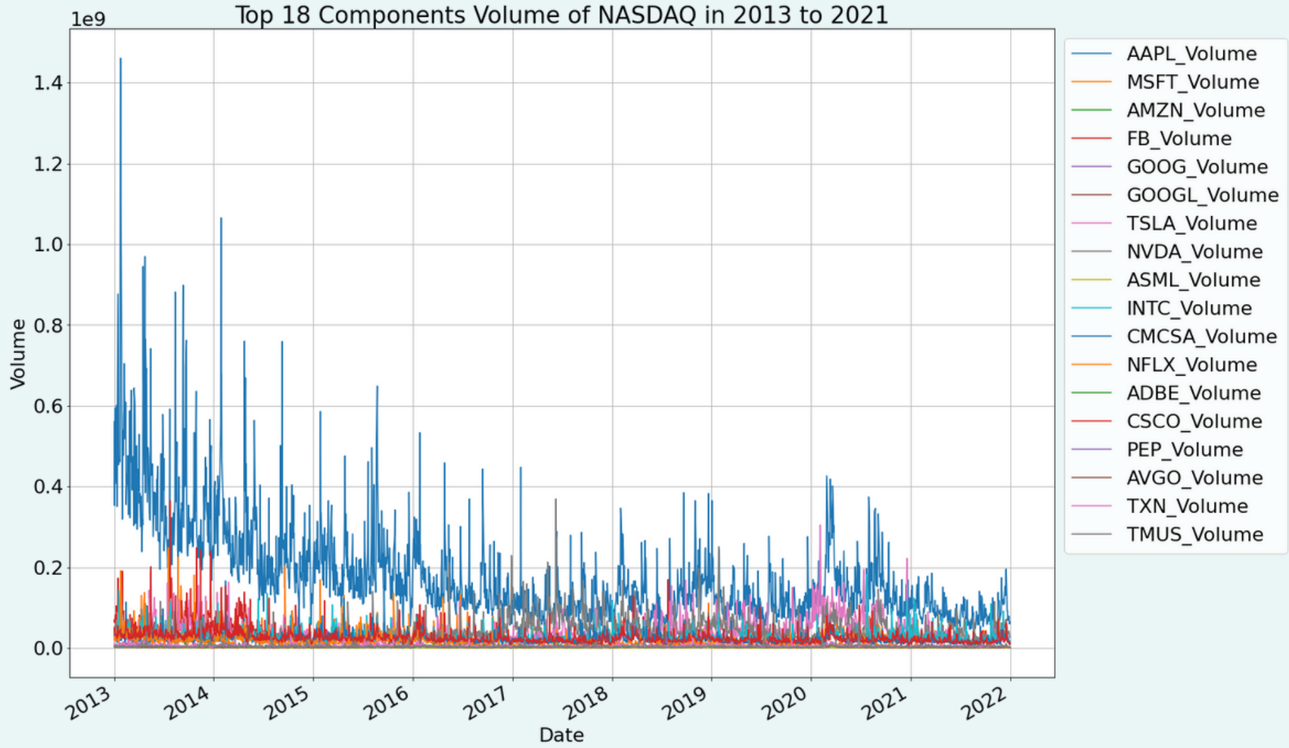
- repeat the data preprocessing steps

EDA

GENERAL VIEW



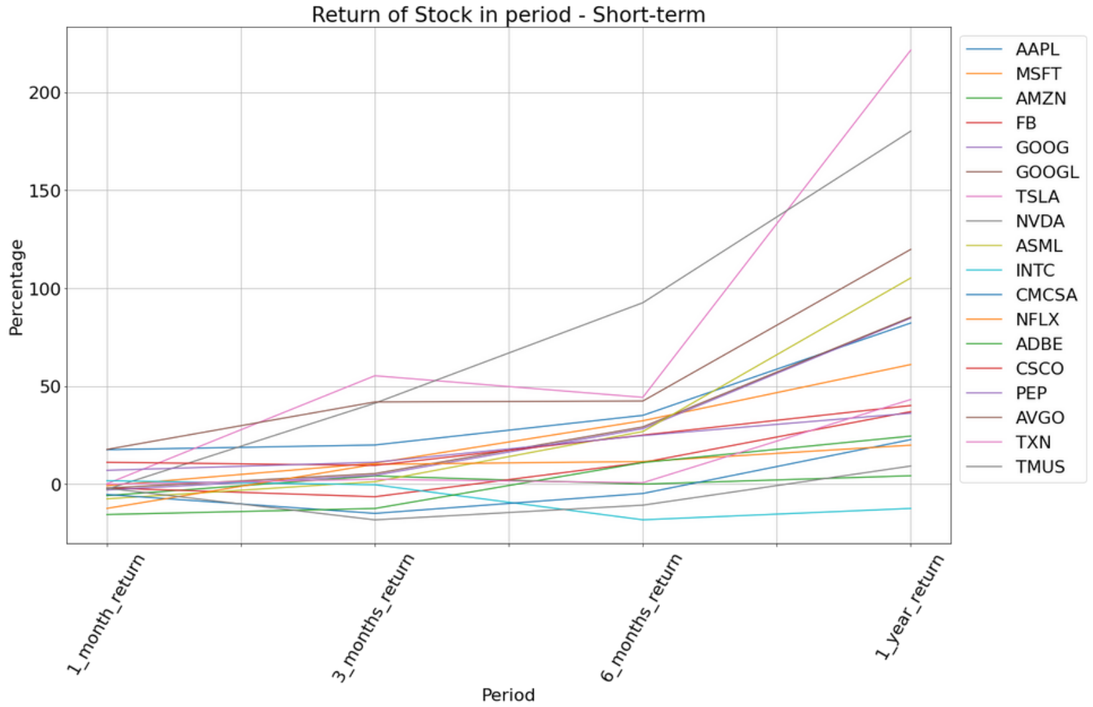


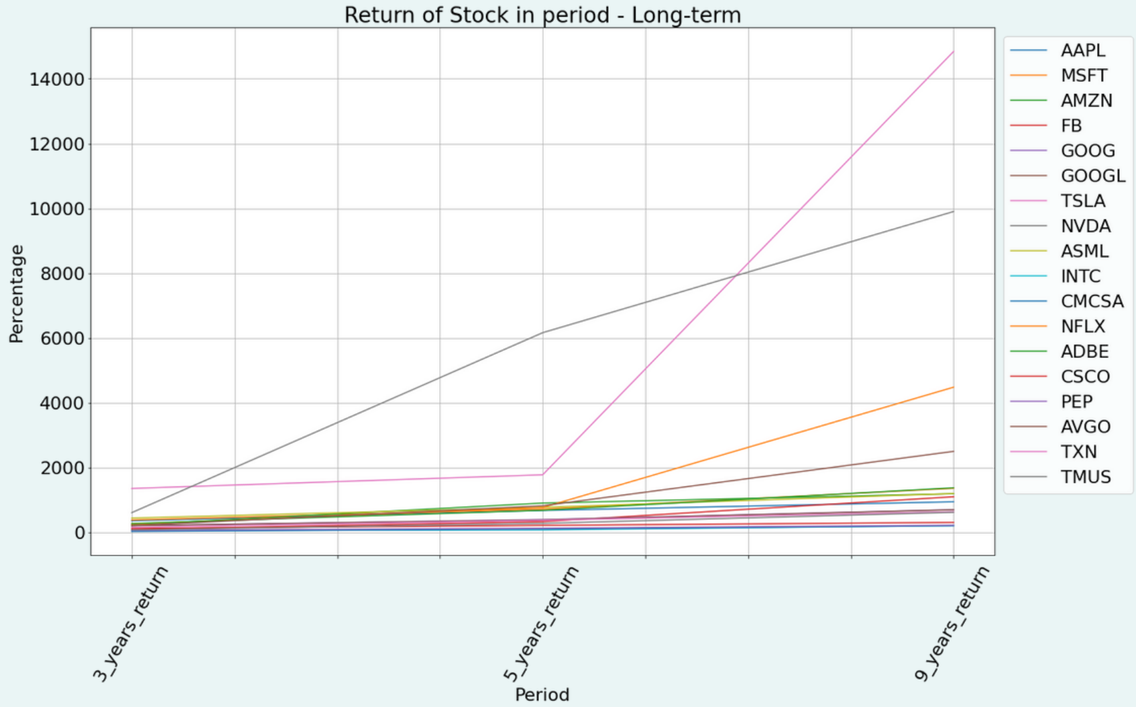


- All index showed an increasing trend from 2013 - 2021
- Volume was obviously increased since 2020
- Top Volume: APPL
- Second Volume: NVDA & TSLA

EDA

TOP 18 NASDAQ COMPONENTS STOCKS PERFORMANCE



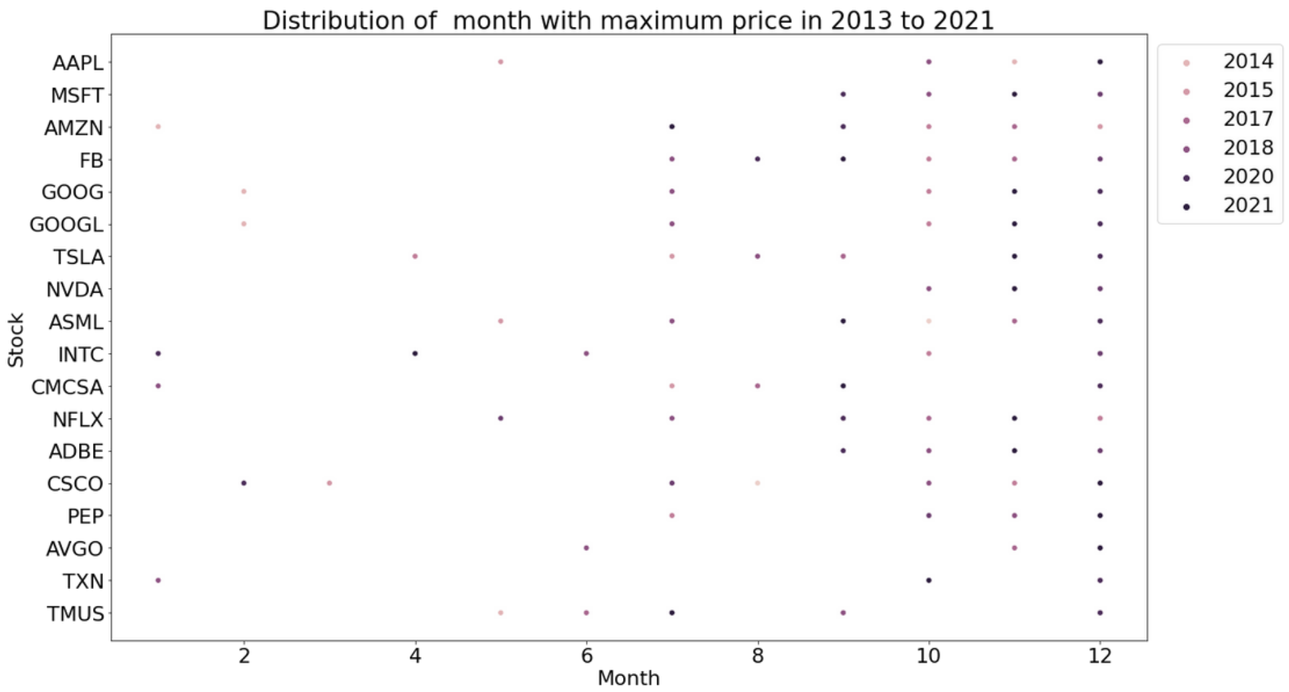


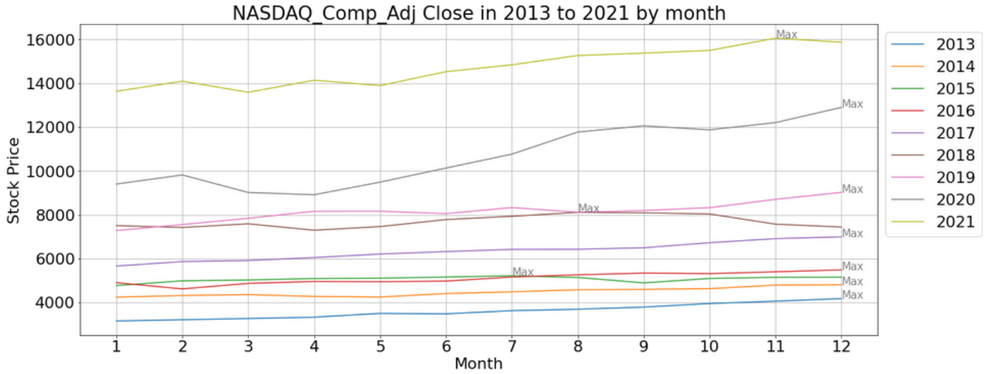
- 1 year return: TSLA >220%
- 6 months return: NVDA >90%
- 3 months return: TSLA >55%
- 1 month return: GOOGL >20%

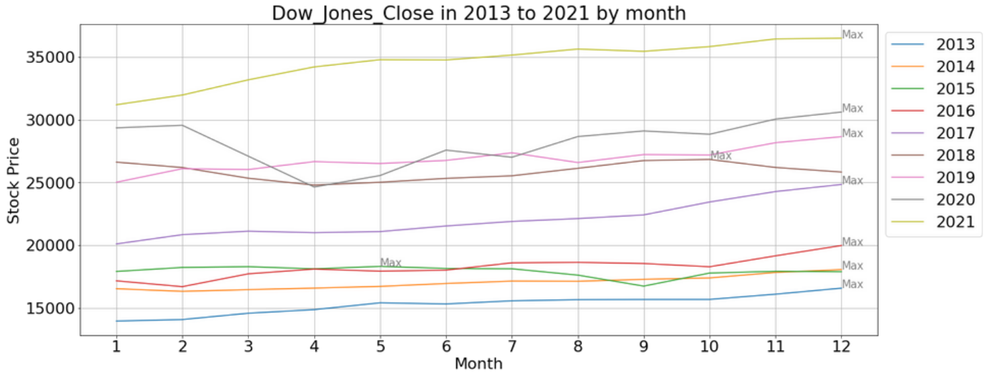
- 9 years return: TSLA >14000%
- 5 years return: NVDA >6000%
- 3 years return: TSLA >1600%

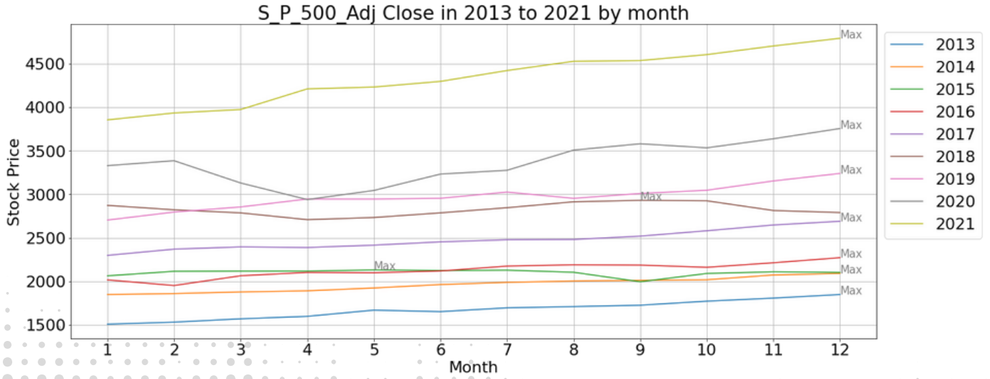
EDA

DISTRIBUTION OF THE MAXIMUM STOCK PRICE BY MONTH IN 2013 - 2021

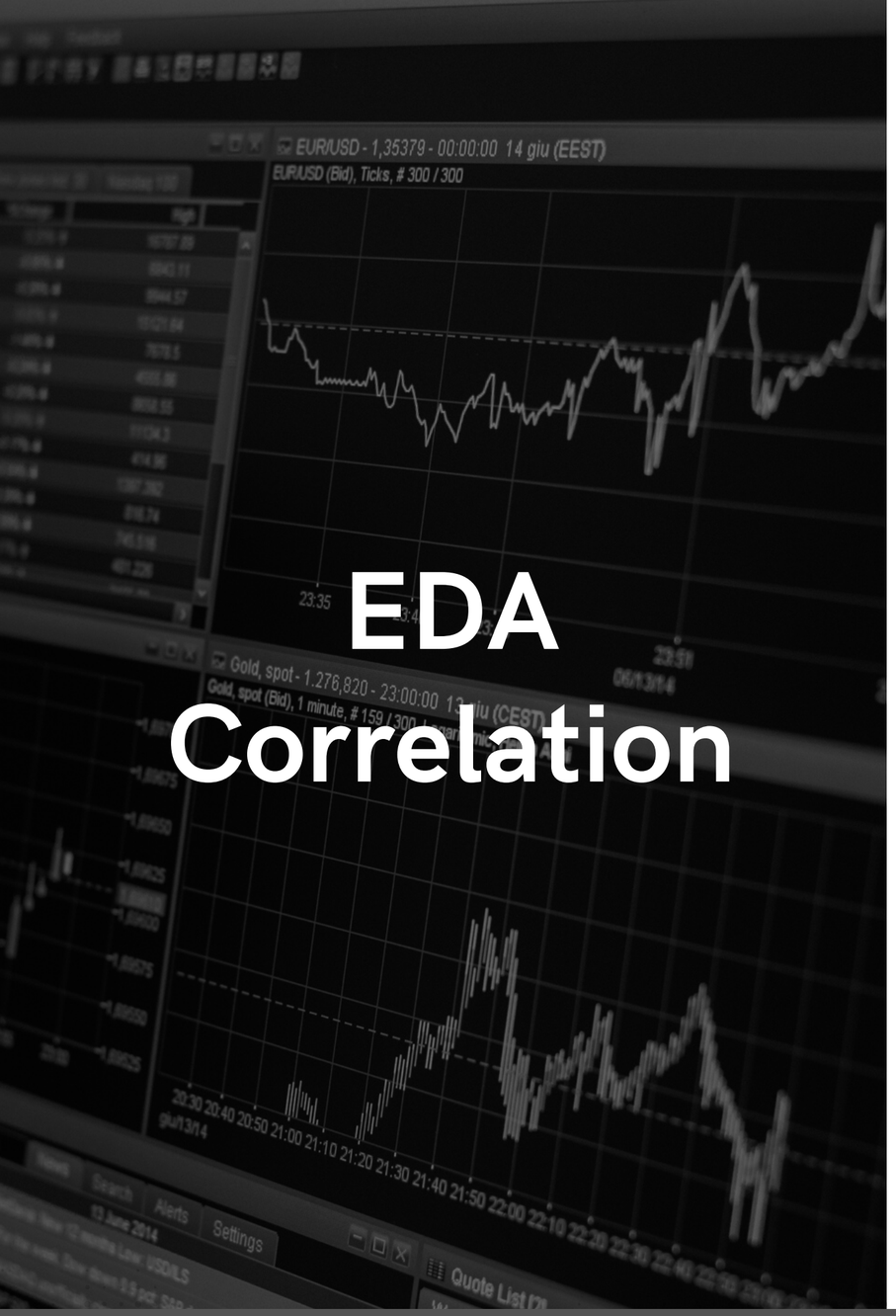




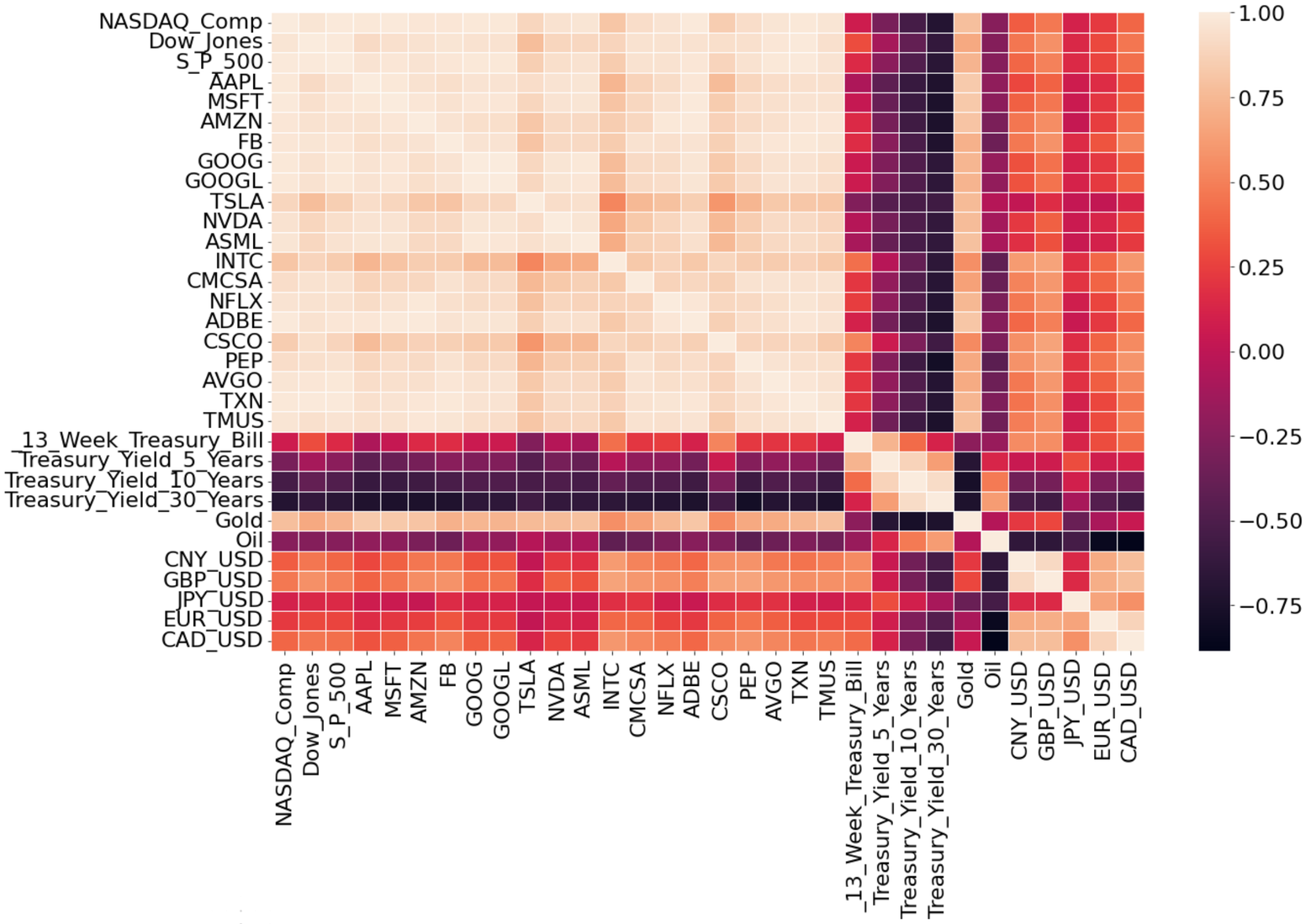




- Top Month: December
- 2nd Month: October
- 3rd Months: July & November



EDA
Correlation



- **Only real time data of Index, Stocks, Currency and Treasury Bills are chosen for the models by import yfinance library**

OUR MODELS

Classification - Trend of NASDAQ Index

BEST MODEL

Model 2

Random Forest

Accuracy:55.1%

Classify the trend by using the bagging method

Model 1

Logistic Regression

Accuracy:54.2%

Classify the trend by using the Logistic Regression Model

Model 3

XGBoost

Accuracy:52.6%

Classify the trend by using the boosting method

Time Series - Price Prediction

BEST MODEL

Model 2

LSTM

Mean Absolute % Error: 1.4%

Fitting a model with LSTM for Price Prediction

Model 1

ARIMA

Mean Absolute % Error: 6.86%

Fitting a model with ARIMA for Price Prediction

Model Creation - Classification

PREDICT THE TREND OF NASDAQ COMPOSITE INDEX OF THE NEXT DAY

1. Library Installation
   -sklearn, lazypredict, yfinance
2. Make Dummies
   -year, month, day of week
3. Feature Selection
   -open, close, adj. close, daily change %
4. Feature Scaling
   -StandardScaler
5. Train-Test Split
   -75:25 Ratio
6. Lazy Predict*
7. Model Deployment
   -Logistic Regression, Random Forest, XGboost

Best Model

Classification - Trend of NASDAQ Index

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

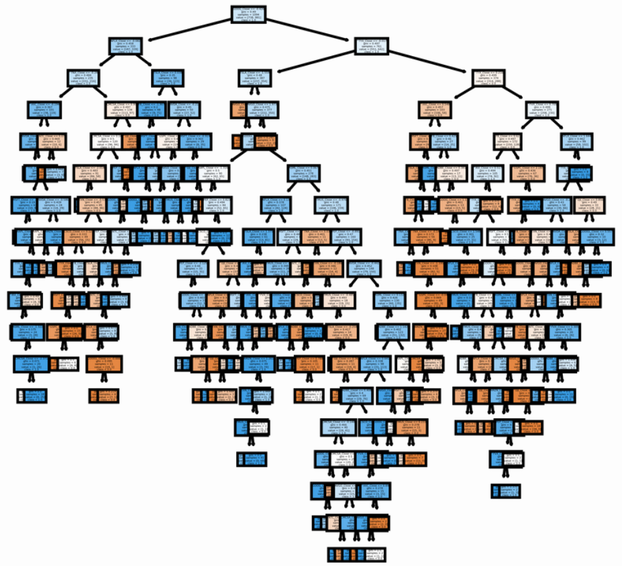
```
rf = RandomForestClassifier(n_estimators=100)
rf.fit(x_train,y_train)
y_pred_rf = rf.predict(x_test)
```

Random Forest

Accuracy:55.1%

Description:

Features Used:
All the features (including open, close, adjacent close, daily change %)



Hyperparameter Optimization

Classification - Trend of NASDAQ Index

Randomized Search

Accuracy:57.0%

Best Estimator:
max_depth=60
min_samples_split=5
n_estimators=1400

Randomized Search

```
from sklearn.model_selection import RandomizedSearchCV
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]

max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)

min_samples_split = [2, 5, 10]

min_samples_leaf = [1, 2, 4]

bootstrap = [True, False]

random_grid = {'n_estimators': n_estimators,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
print(random_grid)
```

```
{'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000], 'max_depth': [10, 20, 30, 40, 50, 60, 70,
80, 90, 100, 110, None], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'bootstrap': [True, False]}
```

```
RandomForestClassifier(max_depth=60, min_samples_split=5, n_estimators=1400)
Model Performance
Accuracy = 0.570.
              precision    recall  f1-score   support

         0.0       0.52      0.21      0.30       252
         1.0       0.58      0.85      0.69       322

    accuracy                           0.57       574
   macro avg       0.55      0.53      0.50       574
weighted avg       0.56      0.57      0.52       574
```

Evaluation and Comparison

Feature Selection

Close Price

Accuracy:
53.1%

Features Used:
Close Price of all Items

Model Used:
Random Forest

All Features

Accuracy:
55.1%

Features Used:
All the features

Model Used:
Random Forest

Daily Change %

Accuracy:
53.6%

Features Used:
Daily Change % of all Items

Model Used:
Random Forest

OUR MODELS

Classification - Trend of NASDAQ Index

BEST MODEL

Model 2

Random Forest

Accuracy:55.1%

Classify the trend by using the bagging method

Model 1

Logistic Regression

Accuracy:54.2%

Classify the trend by using the Logistic Regression Model

Model 3

XGBoost

Accuracy:52.6%

Classify the trend by using the boosting method

Time Series - Price Prediction

BEST MODEL

Model 2

LSTM

Mean Absolute % Error: 1.4%

Fitting a model with LSTM for Price Prediction

Model 1

ARIMA

Mean Absolute % Error: 6.86%

Fitting a model with ARIMA for Price Prediction

Model Creation - Time Series

PREDICT THE PRICE OF NASDAQ COMPOSITE INDEX OF THE NEXT DAY

1. Library Installation
   -sklearn, keras, yfinance
2. Scaling
   -MinMaxScaler
3. Feature Selection
   -open, close, adj. close, daily change %
4. Transforming the Data
   -for neural model processing
5. Train-Test Split
   -90:10 Ratio
6. Training the Model
7. Model Deployment

Best Model

Time Series - Price Prediction for NASDAQ

Multivariate LSTM Model

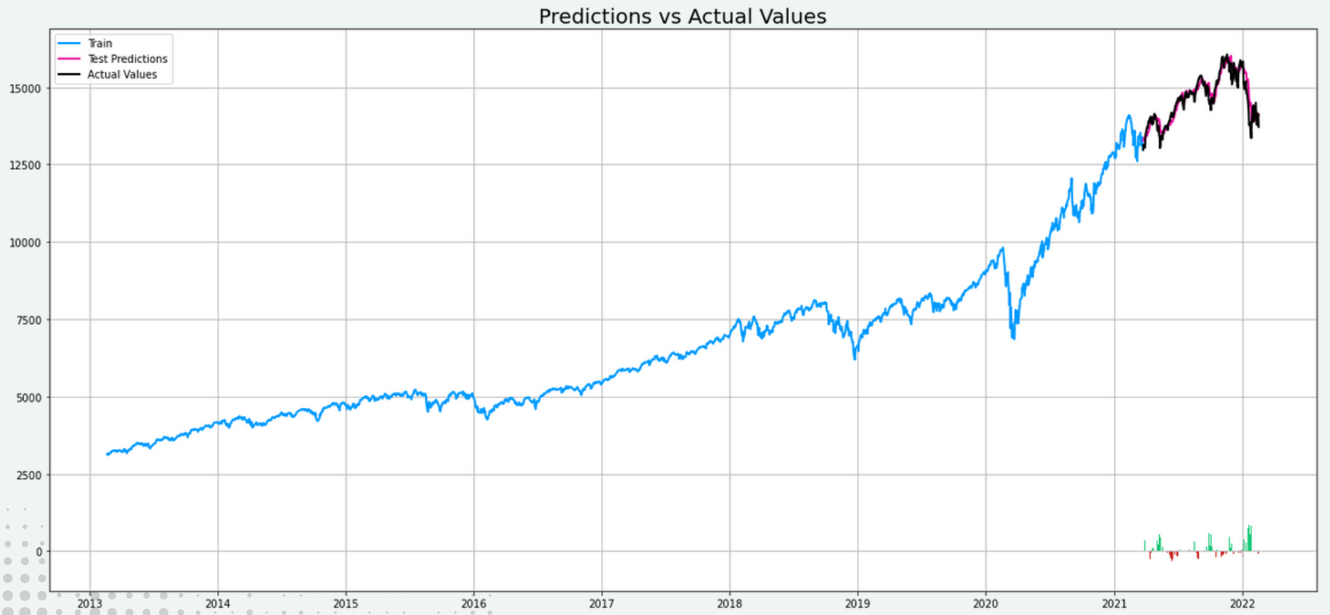
Mean Absolute Error (MAE): 202.06
Mean Absolute Percentage Error (MAPE): 1.4 %
Median Absolute Percentage Error (MDAPE): 0.99 %

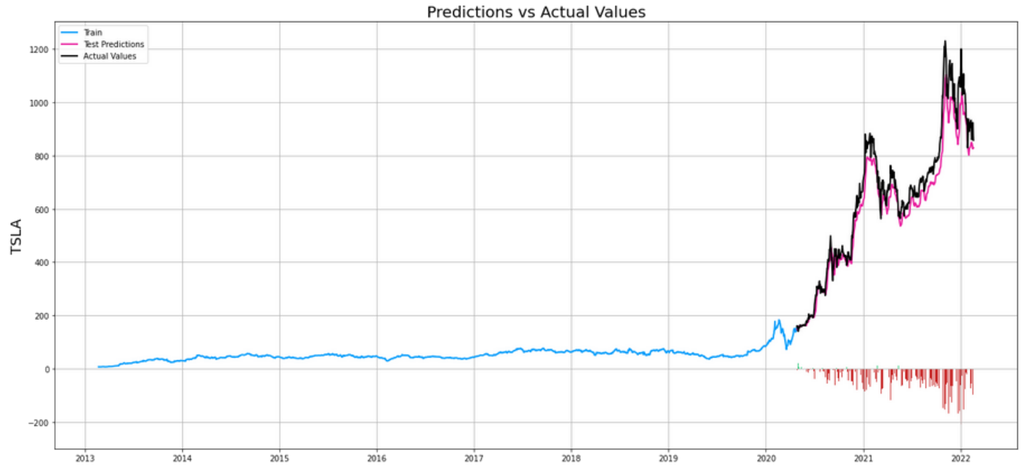
Feature Used:

- NASDAQ Close
- Daily Change % of 18 stocks
- Daily Change % of Treasury Bill

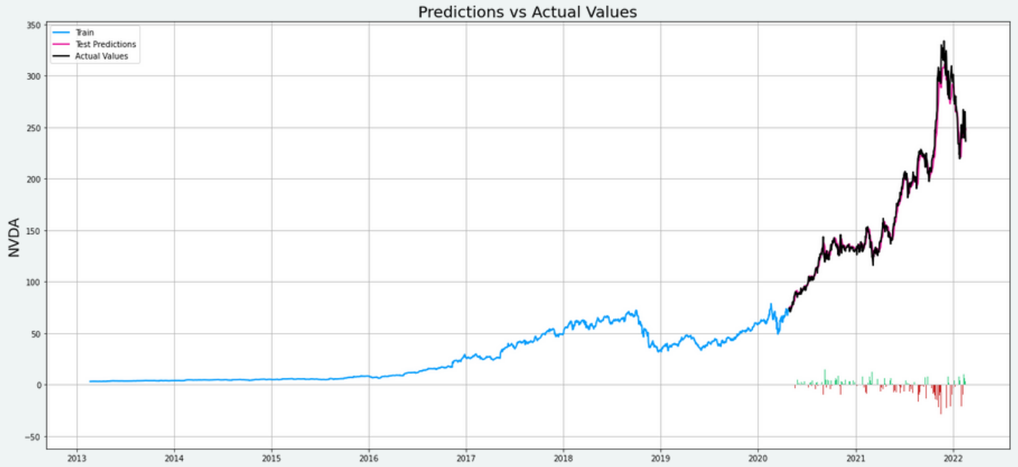
```
# Configure the neural network model
model = Sequential()

# Model with n_neurons = inputshape Timestamps, each with x_train.shape[2] variables
n_neurons = x_train.shape[1] * x_train.shape[2]
print(n_neurons, x_train.shape[1], x_train.shape[2])
model.add(LSTM(n_neurons, return_sequences=True, input_shape=(x_train.shape[1], x_train.shape[2])))
model.add(LSTM(n_neurons, return_sequences=False))
model.add(Dense(5))
model.add(Dense(1))

# Compile the model
model.compile(optimizer='adam', loss='mse')
```

```
# Training the model
epochs = 10
batch_size = 16
early_stop = EarlyStopping(monitor='loss', patience=5, verbose=1)
history = model.fit(x_train, y_train,
                    batch_size=batch_size,
                    epochs=epochs,
                    validation_data=(x_test, y_test),

                    callbacks=[early_stop])
```

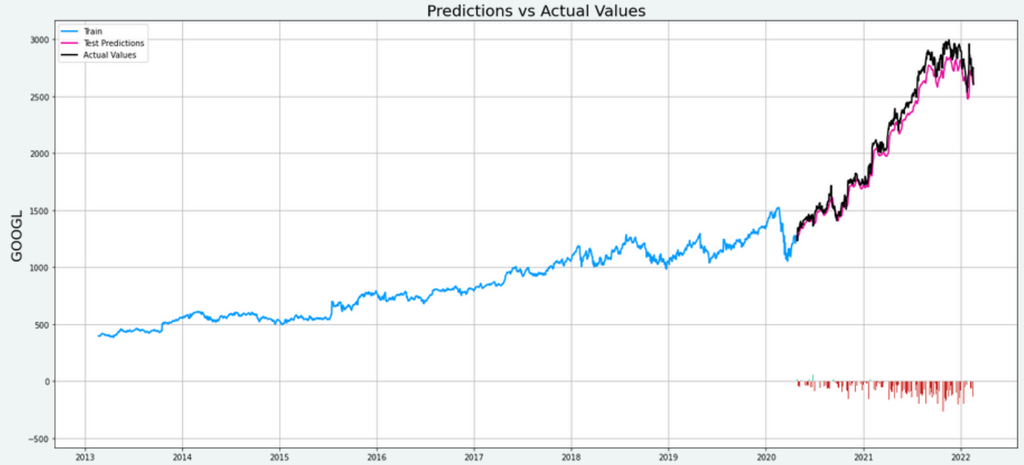




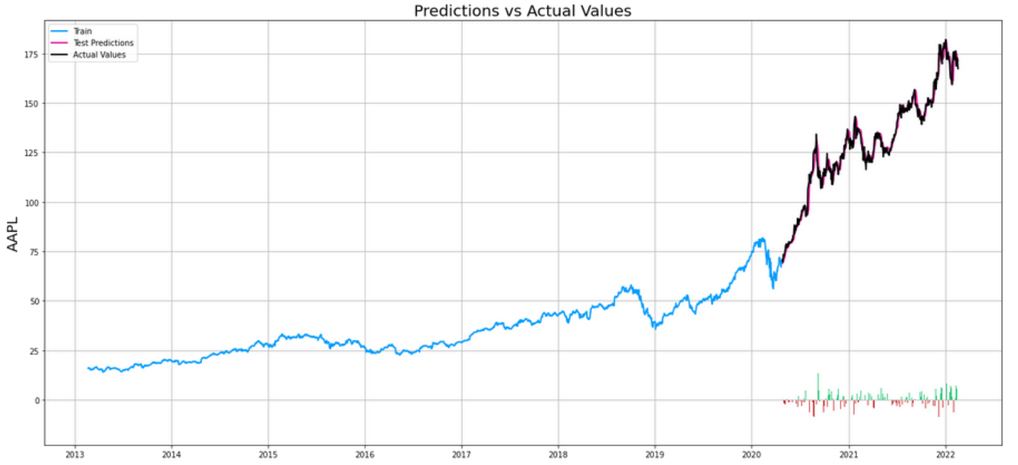
The close price for TSLA at 2022-02-18 was 856.98
The predicted close price is 819.58(-4.56%)
MAE: 48.1 MAPE: 6.92% MDAPE: 6.54%



The close price for NVDA at 2022-02-18 was 236.42
The predicted close price is 243.61 (+2.95%)
MAE: 5.51 MAPE: 3.01% MDAPE: 2.5%



The close price for GOOGL at 2022-02-18 was 2608.06
The predicted close price is 2575.73 (-1.26%)
MAE: 80.56 MAPE: 3.55% MDAPE: 3.5%



The close price for AAPL at 2022-02-18 was 167.3
The predicted close price is 170.49 (+1.87%)
MAE: 2.61 MAPE:2.02% MDAPE: 1.5%

Evaluation and Comparison

Feature Selection / Hyper Parameter Optimization

Close Price

MAE:
2.32%

Features Used:
Close Price of all Items

Close Price

w/o Commodities & Currency

MAPE:
2.35%

Features Used:
Close Price of all items except Gold, Oil, and Currencies

Daily Change %

MAPE:
1.4%

Features Used:
NASDAQ Close, Daily Change % of all items, except Gold, Oil, and Currencies

Evaluation

- More indicator doesn't necessarily mean better results
- LSTM very time consuming

- Optimize model with:
  - Feature Selection Techniques
  - Feature Engineering
  - Grid Search

Limitations

How to improve the accuracy

Company News

Industries or Company

e.g. Tesla cars accidents, coronavirus affecting Netflix

News may cause the company or whole industry stock prices boosting or dropping within a short period.

Current Affairs

Government Policies or Wars

e.g. Government subsidies to electric cars, Ukraine, U.S. and Russia Wage Signaling War

News may cause the company or whole industry stock prices boosting or dropping within a short period.

Limited Data Sources

Not only Stock Prices

e.g. Amount of Call and Put of the stock, trade details data

Currently, our models were based on previous stock prices only, the Derivative of a stock is absolutely a key affecting the stock price.

Future Improvement

How to improve the accuracy

Tracking News

Industries or Company

Webscrapping / APIs to obtain stock related news / updates that may affect stock price.

eg. Sentiment analysis on news articles

Option Data

Every Stock

Put into account Option Calls Data in Machine Learning Models

Feature Scaling / Engineering

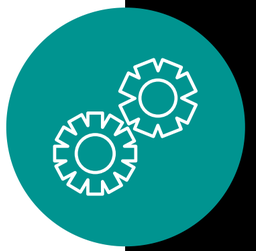
Every Stock / Index

Put into account Option Calls Data in Machine Learning Models

Conclusion

Technology industries may be a good choice for investment than other industries

NASDAQ Composite Index shown the fastest grown comparing with Dow Jones Index & S&P 500 Index



Prediction of the stock price is more accurate than the trend prediction by our models?

The stock price prediction is more accurate than the prediction of the trend in our case.

Only with the previous stock price is far not enough to predict the future stock price and trend

Company news, government policies and markets of derivatives will affect the stock price.

Thank you!

The above models and analysis are only for educational use.
Feel free to reach out to us if you have any questions.

Carol Ho
Model: Time Series

Jeffrey Cheng
Model: Classification

Naomi Tsang
Data Preparation & Analysis
Model: Time Series - Stock Price