Analysis of stock market trends

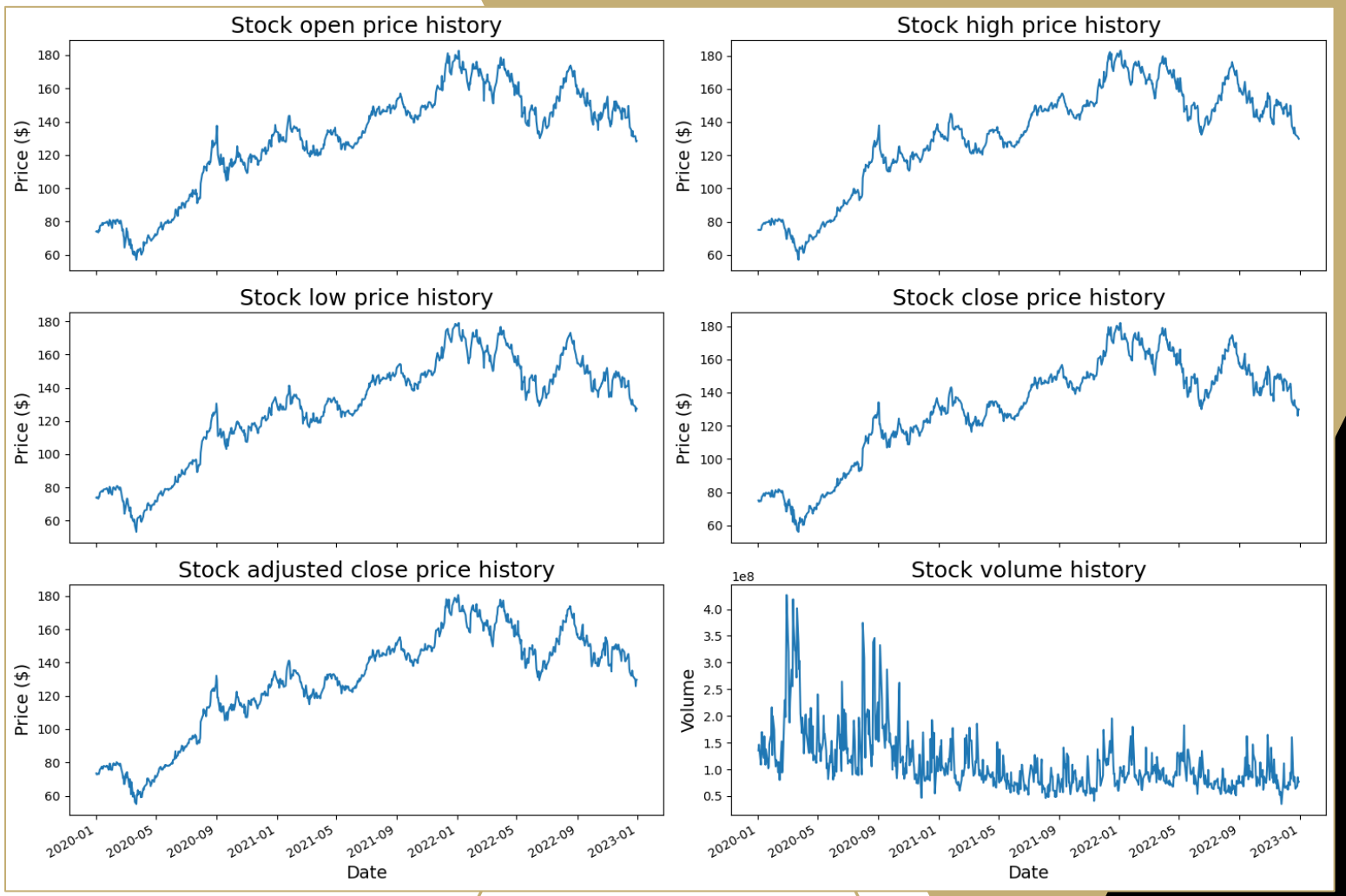
BEBEROK PATRYK

DWORACZYK MATEUSZ

22.05.2023

Main goals

- Analyzing the stock market data
- Loading the dataset into the program
- Making the neural network
- Trainning the neural network
- Getting the data from the neural network
- Making the report about the data

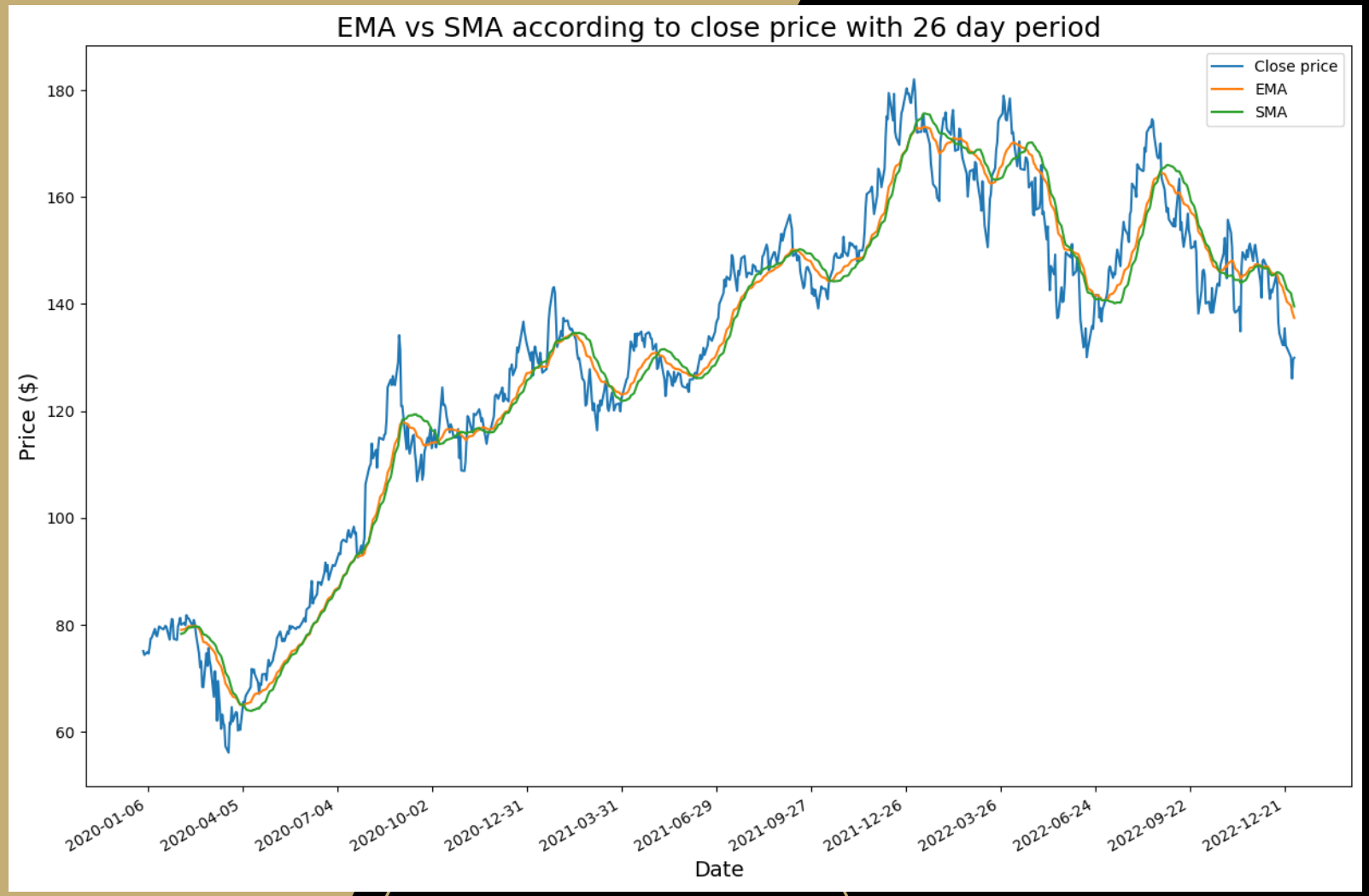


Loading dataset for visualization

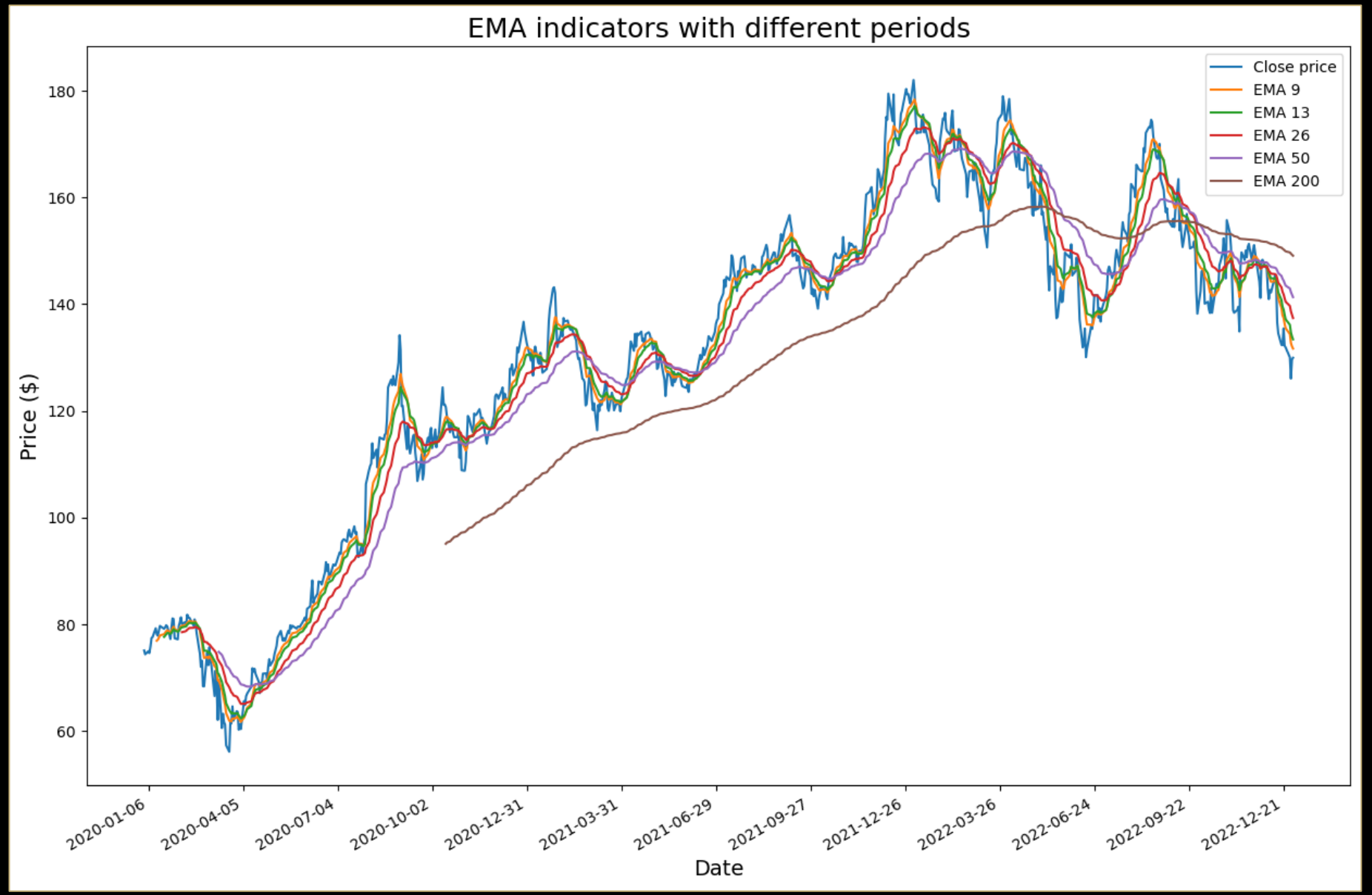
For the visualization we used the dataset from finance.yahoo.com.

This dataset contains the historical stock data from the AAPL and MSFT company.

EMA vs SMA

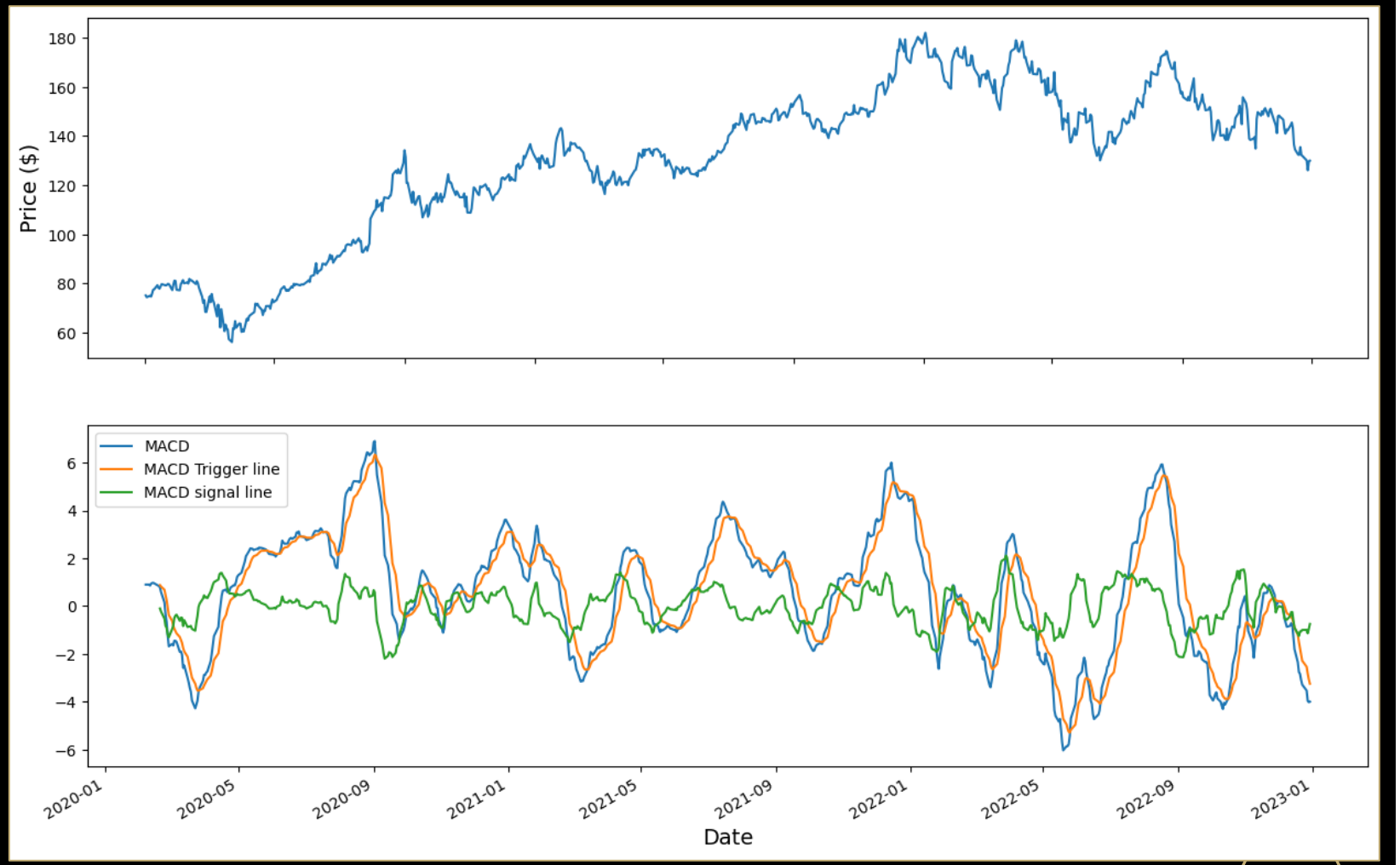


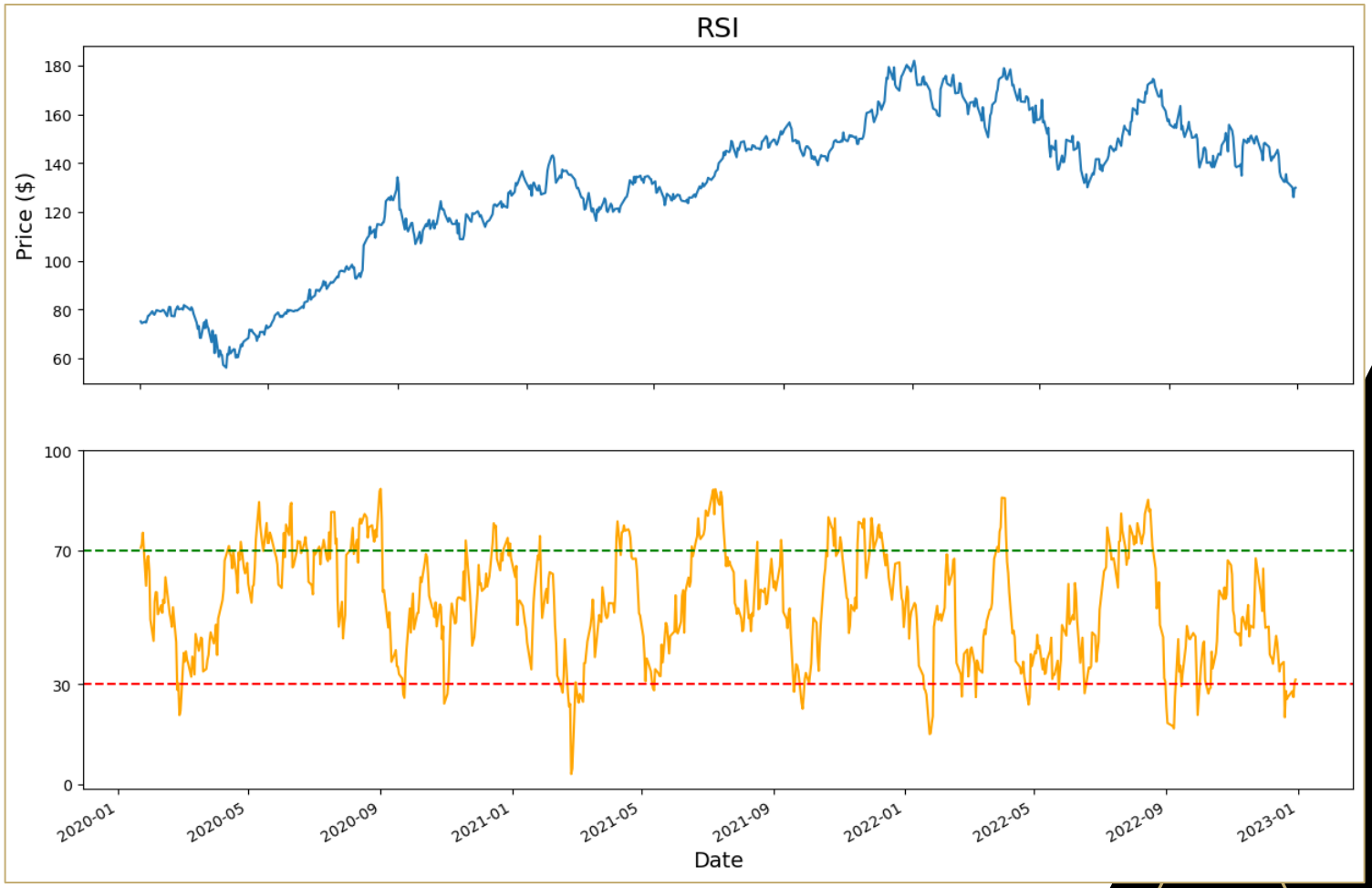
- The SMA is the most common type of average used by technical analysts and is calculated by dividing the sum of a set of prices by the total number of prices found in the series.
- The moving average (MA) helps to level the price data over a specified period by creating a constantly updated average price.
- SMA calculates the average of price data, while EMA gives more weight to current data.


EMA indicators with different periods
Close price
EMA 9
EMA 13
EMA 26
EMA 50
EMA 200
Price ($)
180
160
140
120
100
80
60
2020-01-06
2020-04-05
2020-07-04
2020-10-02
2020-12-31
2021-03-31
2021-06-29
2021-09-27
2021-12-26
2022-03-26
2022-06-24
2022-09-22
2022-12-21
Date

MACD

- First one, mostly known as MACD represents substracted value of EMA with long lookback period from EMA with short lookback period.
- Second one is trigger signal which is respresented by EMA with shorter lookback period than the one used in MACD signal.
- The last one is difference signal which is basically subtracted value of trigger signal from the MACD signal.





RSI

Relative Strength Index is a momentum indicator that represents magnitude of recent price changes.

It's mostly known for evaluation of either overbought or oversold conditions of shares.

It takes value from 0 to 100.

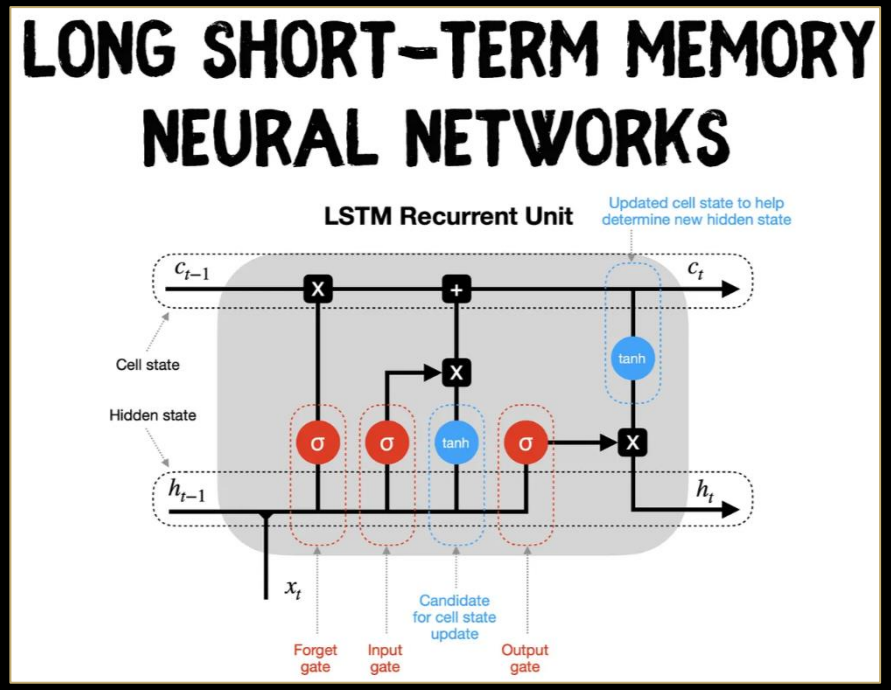
Model creation

In the process of developing our models we used three types of layers:

- LSTM
- Dense
- Dropout

```
# Default LSTM model
lstm = Sequential(
    layers=[
        LSTM(512, return_sequences=True, input_shape=(x_train.shape[1], 1)),
        LSTM(256),
        Dense(1)
    ]
)
lstm.compile(optimizer='adam', loss='mean_squared_error')
lstm.fit(x_train, y_train, batch_size=256, epochs=100)
```

```
# LSTM model with additional LSTM layer, dropout layers and additional dense layer
lstm_6 = Sequential(
    layers=[
        LSTM(512, return_sequences=True, input_shape=(x_train.shape[1], 1)),
        Dropout(0.2),
        LSTM(256, return_sequences=True),
        Dropout(0.2),
        LSTM(128),
        Dropout(0.2),
        Dense(10),
        Dense(1)
    ]
)
lstm_6.compile(optimizer='adam', loss='mean_squared_error')
lstm_6.fit(x_train, y_train, batch_size=256, epochs=100)
```



What contains LSTM?

Long short-term memory (LSTM) is an artificial neural network used in the fields of artificial intelligence and deep learning.

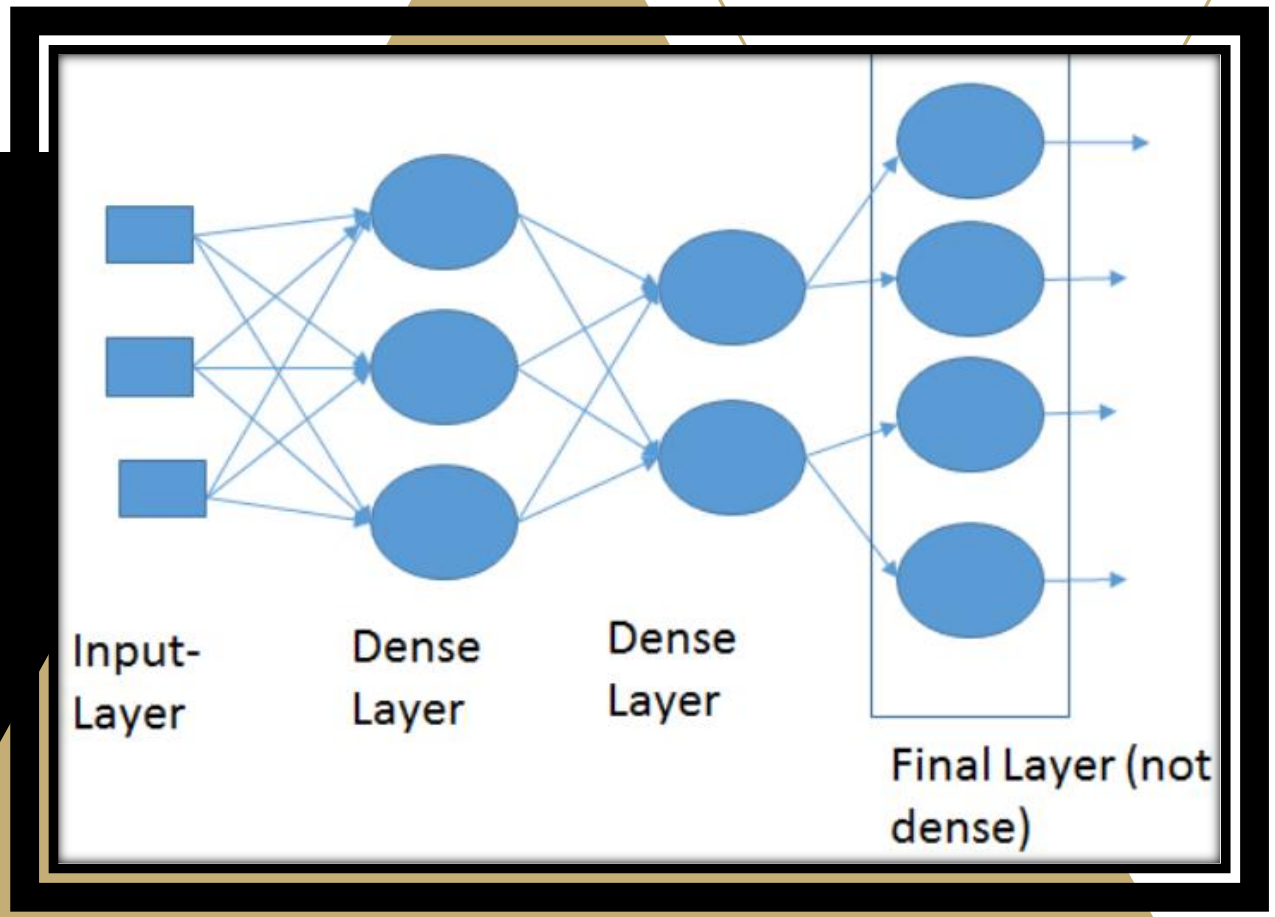
Unlike standard feedforward neural networks, LSTM has feedback connections. Such a recurrent neural network (RNN) can process not only single data points (such as images), but also entire sequences of data (such as speech or video).

This characteristic makes LSTM networks ideal for processing and predicting data.

LSTM recurrent unit is much more complex than that of RNN, which improves learning but requires more computational resources.

Dense

Dense Layer is simple layer of neurons in which each neuron receives input from all the neurons of previous layer, thus called as dense. Dense Layer is used to classify image based on output from convolutional layers.

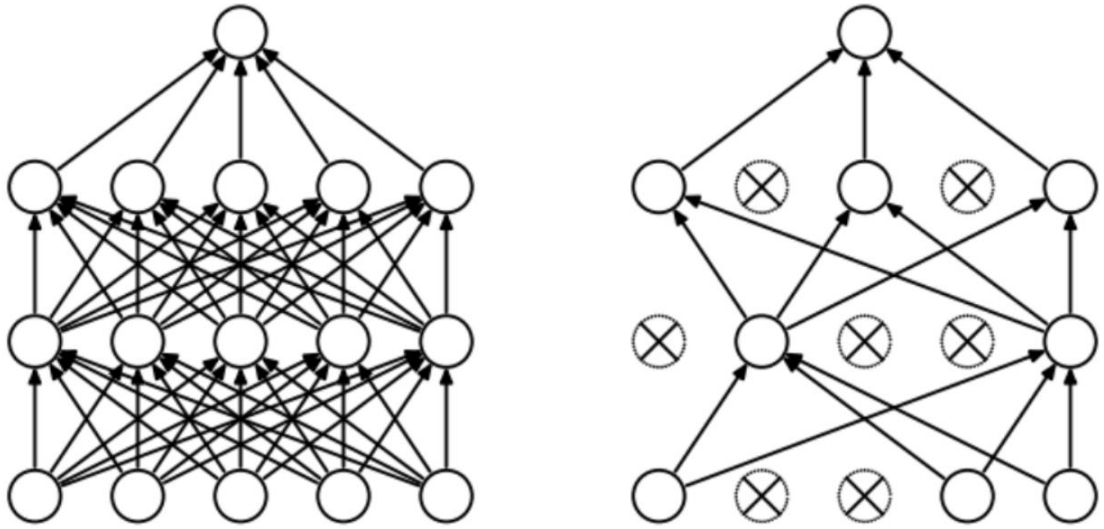


Dropout

WHAT IS A DROPOUT?

The term “dropout” refers to dropping out the nodes (input and hidden layer) in a neural network. All the forward and backwards connections with a dropped node are temporarily removed, thus creating a new network architecture out of the parent network. The nodes are dropped by a dropout probability of p.

SCHEMATIC



(a) Standard Neural Net

(b) After applying dropout.

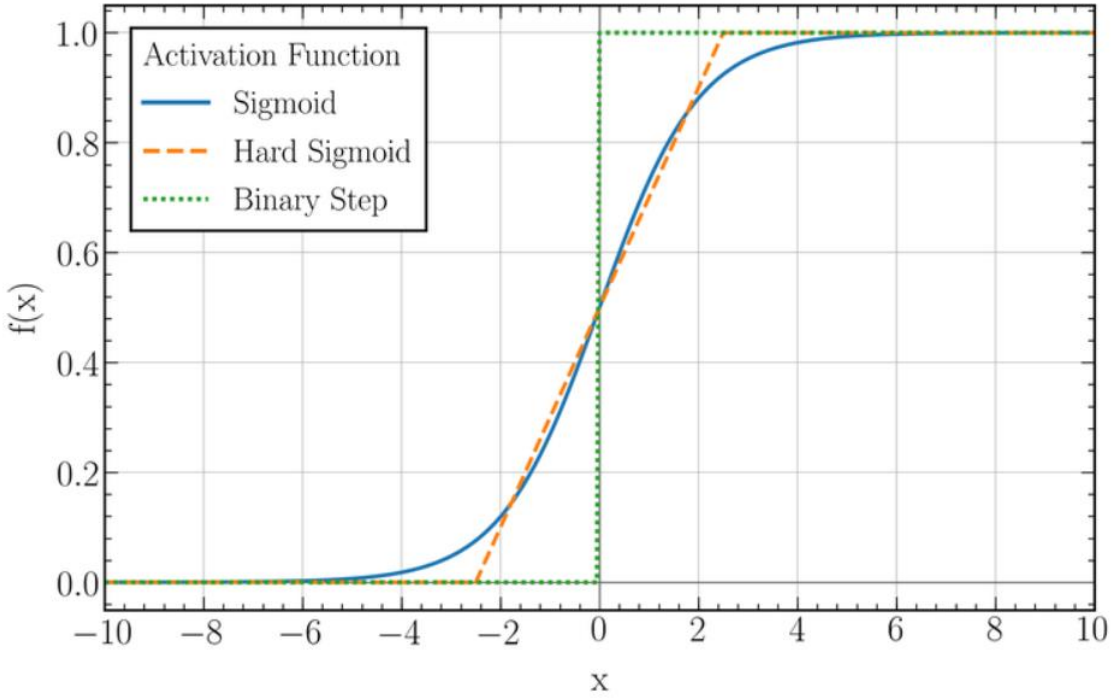
Sigmoid and hard sigmoid

The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output.

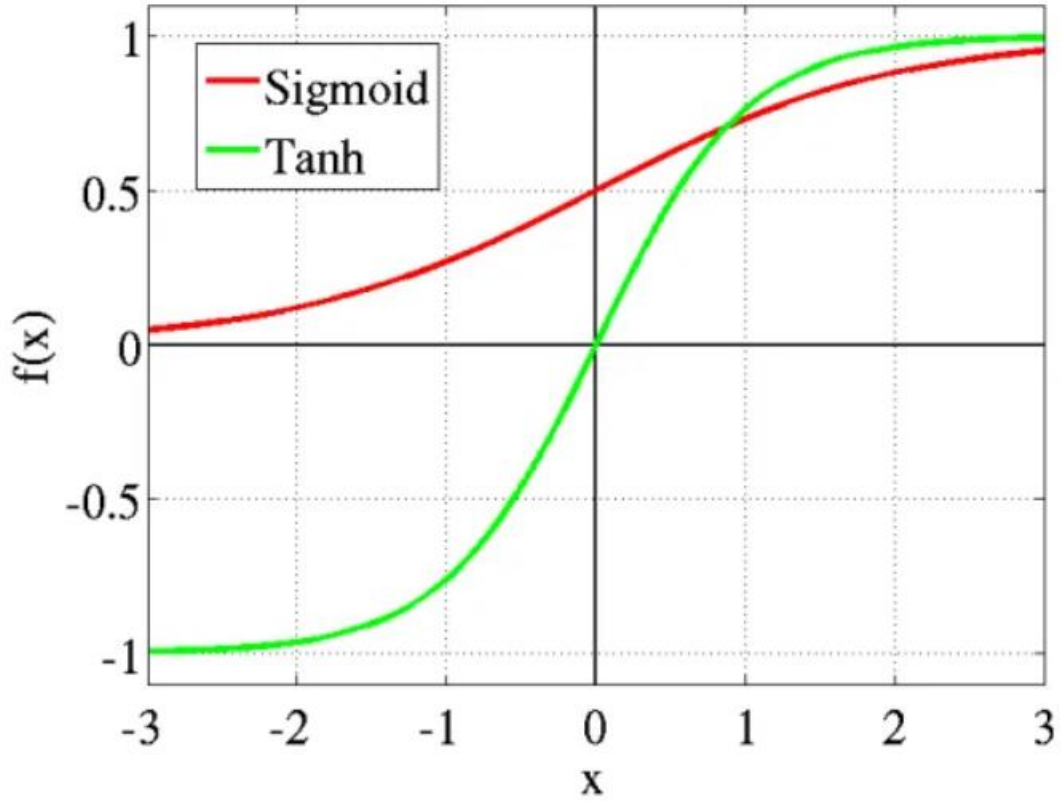
$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} = 1 - S(-x)$$

Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

Hard sigmoid: $f(x) = \max\left(0, \min\left(1, \frac{(x+1)}{2}\right)\right)$



Tanh activation function



The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph.

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

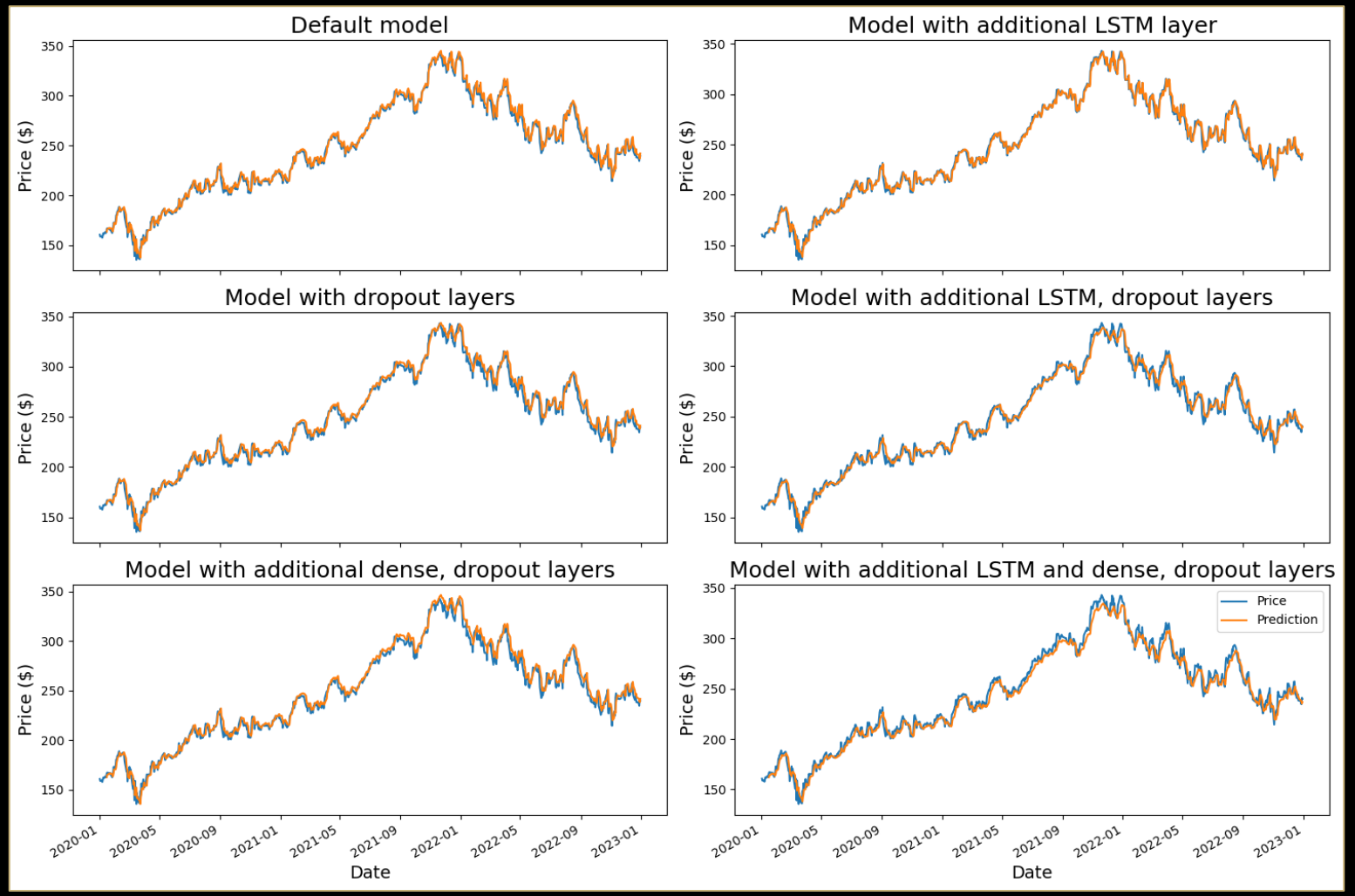
How we tested our model?

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

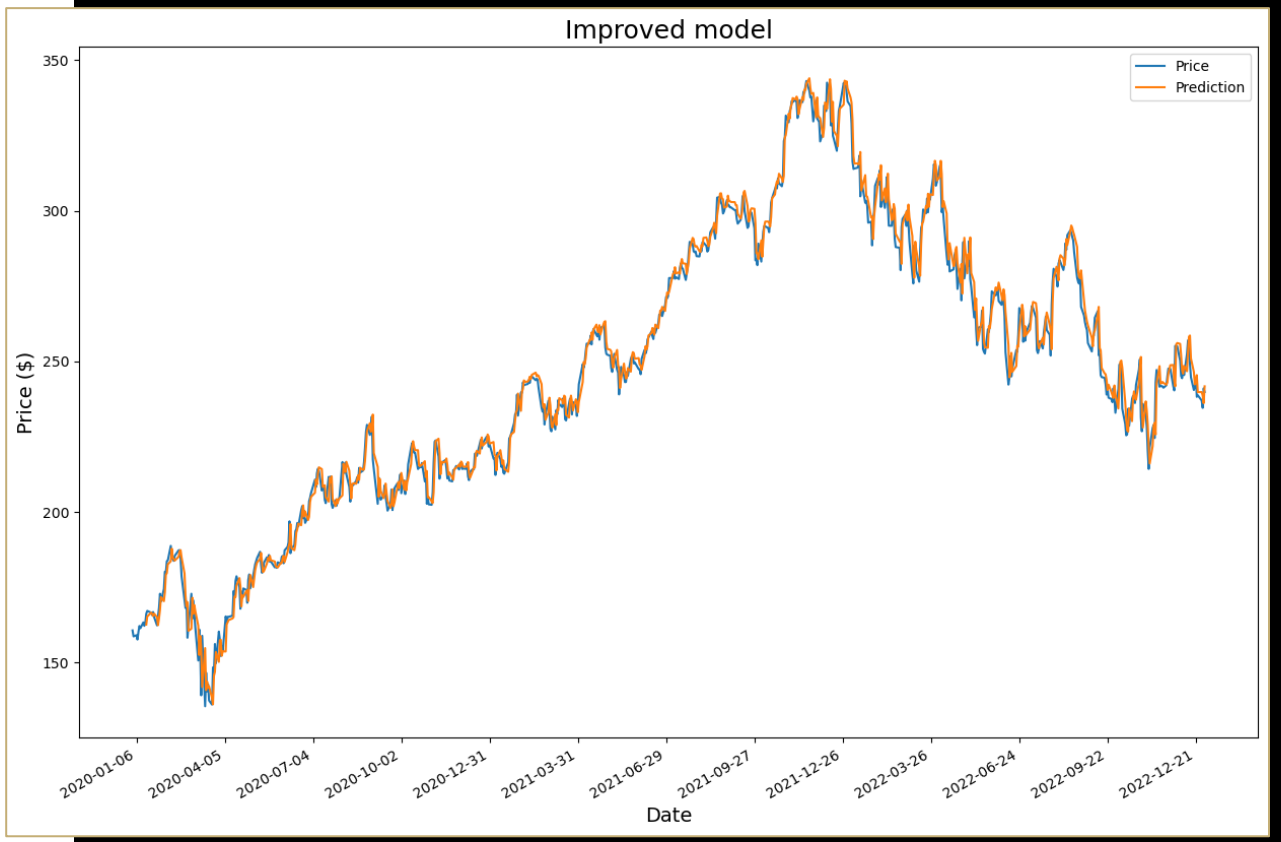
Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.

```
24/24 [==============================] - 1s 21ms/step
Default model RMSE: 5.115447304890235
24/24 [==============================] - 1s 20ms/step
Model with dropout layers RMSE: 5.551426500222034
24/24 [==============================] - 1s 21ms/step
Model with additional dense, dropout layers RMSE: 5.862157263279812
24/24 [==============================] - 1s 23ms/step
Model with additional LSTM layer RMSE: 4.825531329114918
24/24 [==============================] - 1s 22ms/step
Model with additional LSTM, dropout layers RMSE: 5.598058541048898
24/24 [==============================] - 1s 23ms/step
Model with additional LSTM and dense, dropout layers RMSE: 6.127154677104754
```

Testing results



Improvement of the best model

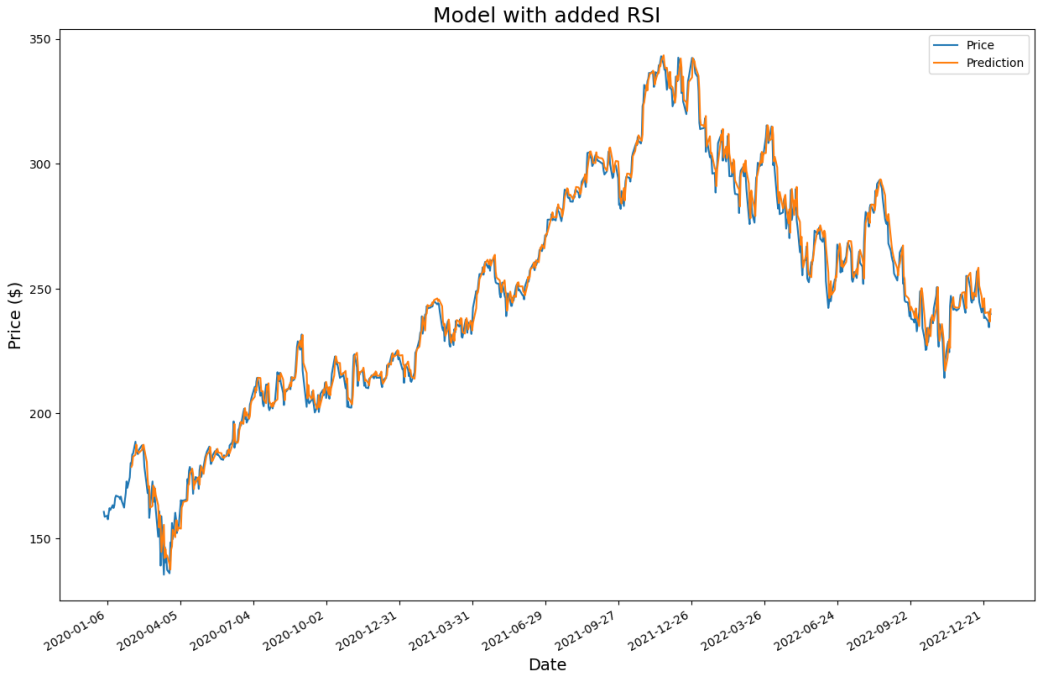



```
Epoch 500/500
42/42 [==============================] - 9s 213ms/step - loss: 2.8746e-05
24/24 [==============================] - 1s 23ms/step
Model with additional LSTM layer RMSE: 4.969659945219601
```

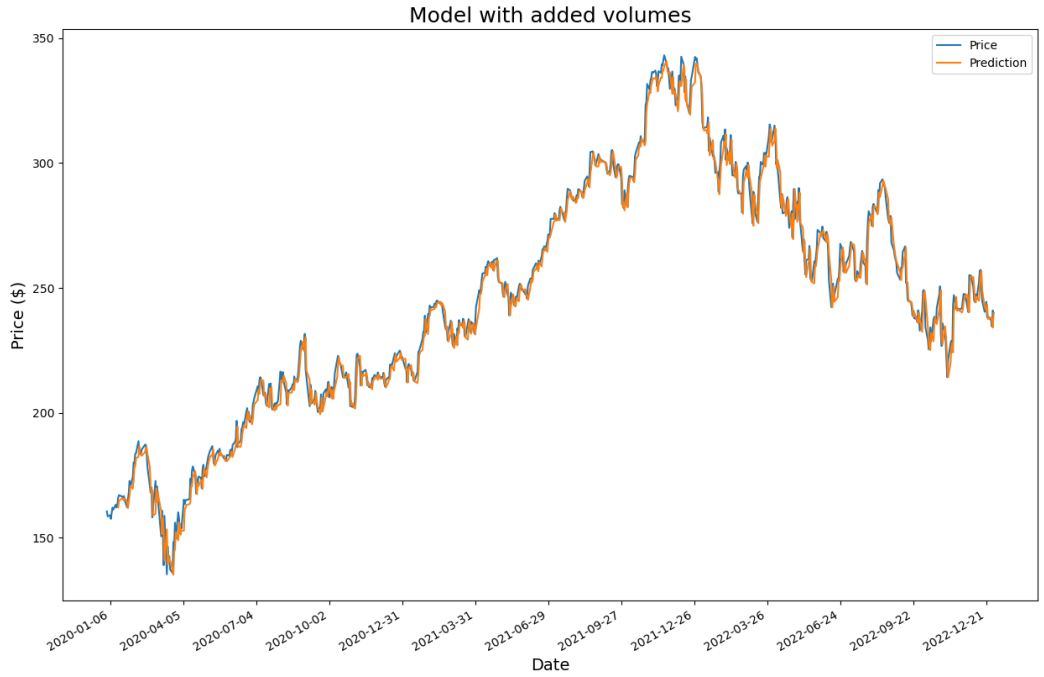
```python
# LSTM model with additional LSTM layer
lstm_4 = Sequential(
    layers=[
        LSTM(512, return_sequences=True, input_shape=(x_train.shape[1], 1)),
        LSTM(256, return_sequences=True),
        LSTM(128),
        Dense(25),
        Dense(1)
    ]
)
lstm_4.compile(optimizer='adam', loss='mean_squared_error')
lstm_4.fit(x_train, y_train, batch_size=256, epochs=500)

predictions_4 = lstm_4.predict(x_test)
predictions_4 = scaler.inverse_transform(predictions_4)
rmse_4=np.sqrt(np.mean((predictions_4 - y_test) ** 2))
print(f'Model with additional LSTM layer RMSE: {rmse_4}')
```

Including technical indicators to our models

RSI

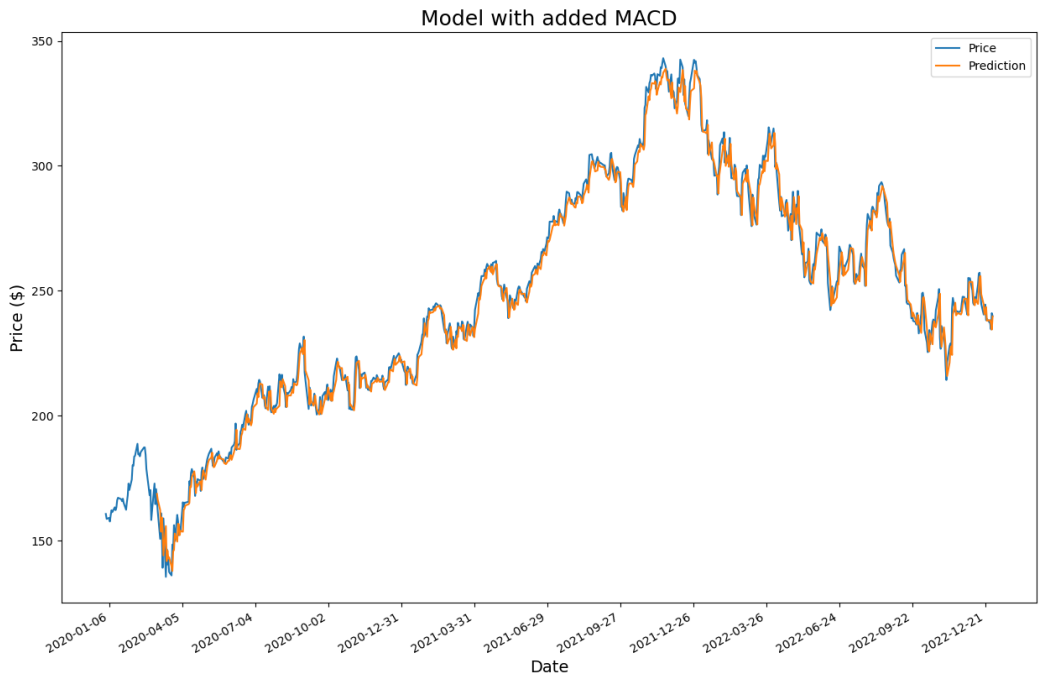


VOLUMES

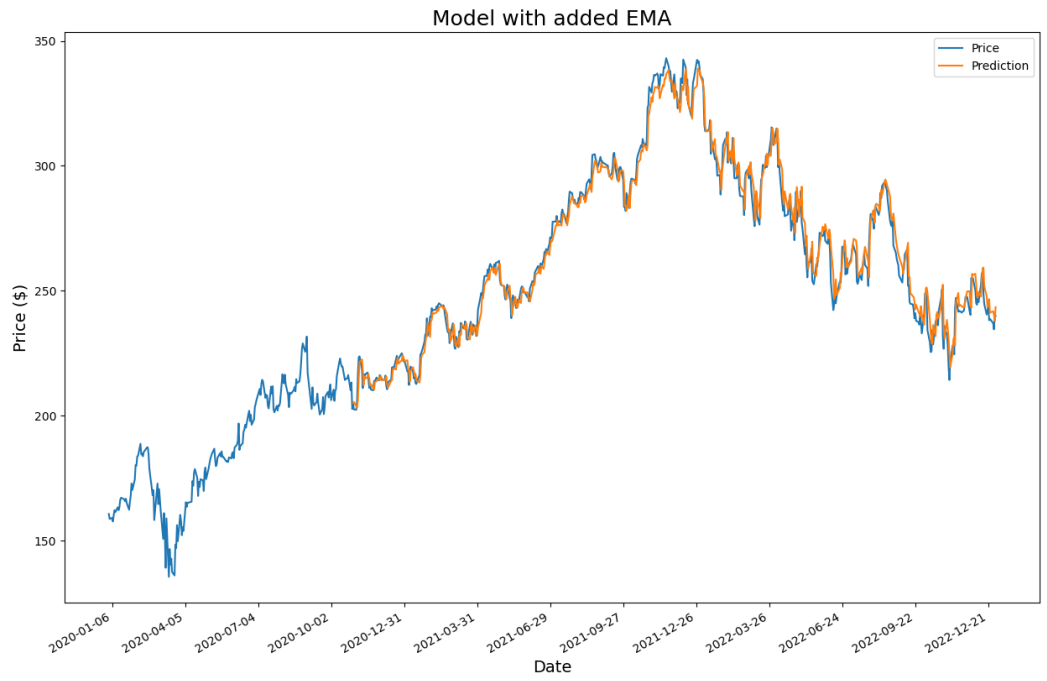


Including technical indicators to our models

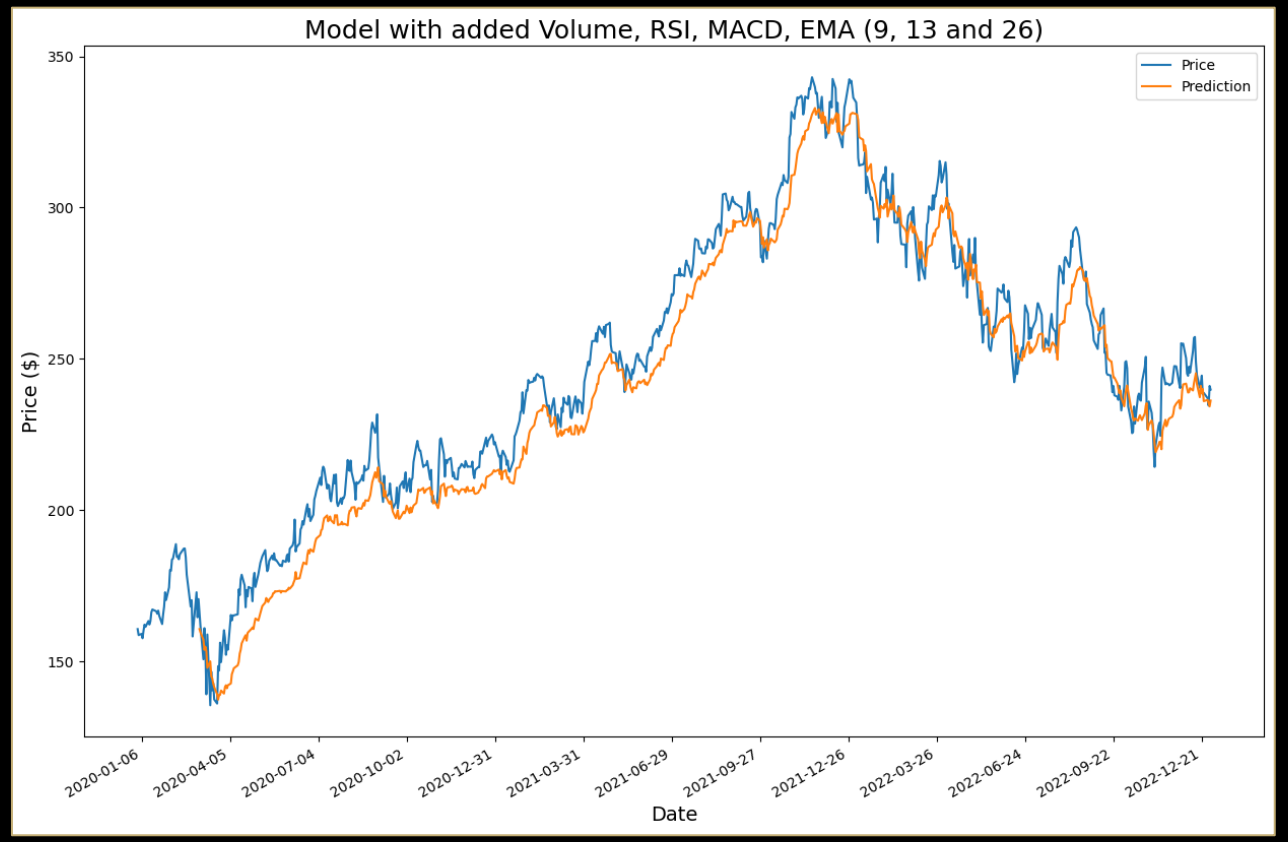
MACD



EMA



Including technical indicators to our models



Bibliography

https://finance.yahoo.com/

https://towardsdatascience.com/

https://machinelearningmastery.com

Thank you for your attention!

matedwo224@student.polsl.pl

patrbeb641@student.polsl.pl