# Problem description

# Data set

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Open | High | Low | Close | Adj Close | Volume |
| 3/20/25 | 5646.92 | 5711.15 | 5632.33 | 5672.83 | 5672.83 | 1257423003 |
| 3/19/25 | 5632.37 | 5715.33 | 5622.20 | 5675.29 | 5675.29 | 4660090000 |
| 3/18/25 | 5654.53 | 5654.53 | 5597.76 | 5614.66 | 5614.66 | 4666270000 |
| 3/17/25 | 5635.60 | 5703.52 | 5631.12 | 5675.12 | 5675.12 | 5031770000 |
| 3/14/25 | 5563.85 | 5645.27 | 5563.85 | 5638.94 | 5638.94 | 4863180000 |
| 3/13/25 | 5594.45 | 5597.78 | 5504.65 | 5521.52 | 5521.52 | 5018980000 |
| 3/12/25 | 5624.84 | 5642.19 | 5546.09 | 5599.30 | 5599.30 | 5219830000 |
| 3/11/25 | 5603.65 | 5636.30 | 5528.41 | 5572.07 | 5572.07 | 6221240000 |
| 3/10/25 | 5705.37 | 5705.37 | 5564.02 | 5614.56 | 5614.56 | 6409370000 |
| 3/7/25 | 5726.01 | 5783.01 | 5666.29 | 5770.20 | 5770.20 | 5705140000 |
| 3/6/25 | 5785.87 | 5812.08 | 5711.64 | 5738.52 | 5738.52 | 5165080000 |

Shape of the DataFrame: data: (3827, 7)

## Visualization:

SPX price by day

A graph showing the growth of a stock market

Description automatically generated

# Challenges:

# Strategy :

# Result:

## Model 1: LSTM + Encoder-Decoder model

### code:

def LSTMEncoder\_Decoder(epochs, batch\_size, X\_train, y\_train, X\_test, y\_test) :

# LSTM Encoder-Decoder Model

latent\_dim = 50

# Encoder

encoder\_inputs = Input(shape=(past\_days, 1))

encoder = tf.keras.layers.LSTM(latent\_dim, activation="relu", return\_sequences=False)(encoder\_inputs)

encoded = RepeatVector(future\_days)(encoder)

# Decoder

decoder = tf.keras.layers.LSTM(latent\_dim, activation="relu", return\_sequences=True)(encoded)

decoder\_outputs = TimeDistributed(Dense(1))(decoder)

# Define model

model = Model(encoder\_inputs, decoder\_outputs)

model.compile(optimizer="adam", loss="mse")

# Train model

with tf.device('/gpu:0'):

history = model.fit(X\_train, y\_train, epochs=epochs, batch\_size=batch\_size, validation\_data=(X\_test, y\_test))

return history, model

### Model summary()

A screenshot of a computer

Description automatically generated

## Hyperparameters

|  |  |  |
| --- | --- | --- |
| Hyper parameter | Value | Comments |
| Encoder |  |  |
| LSTM layer | 1 |  |
| Decoder |  |  |
| LSTM layer | 1 |  |
| Optimizer | Adam |  |
| Loos | MSE |  |
| EPOCHS | 100 |  |
| Batch\_size | 64 |  |

## Loss

A graph of a training loss

Description automatically generated

From the result of loss function by each epoch in above chart, the model fitted well, neither overfitting nor under fitting.

## Prediction

A graph showing the price of a stock market

Description automatically generated

Compare the predicted price and the actual price, the predicated price match with the acctual value well. The error is small by 0.00017429136216920926 (MSE)

## Zoom-in

A graph with blue and orange lines

Description automatically generated

## Model 2: LSTM stack model

### Code

def LSTM(epochs, batch\_size, X\_train, y\_train, X\_test, y\_test) :

latent\_dim = 50

model = Sequential([

tf.keras.layers.LSTM(latent\_dim, activation="relu", input\_shape=(past\_days, 1), return\_sequences=True),

tf.keras.layers.LSTM(latent\_dim, activation="relu", input\_shape=(latent\_dim, 1), return\_sequences=False),

Dense(25, activation="relu"),

Dense(future\_days) #

])

model.compile(optimizer="adam", loss="mse")

# Train model

with tf.device('/gpu:0'):

history = model.fit(X\_train, y\_train, epochs=epochs, batch\_size=batch\_size, validation\_data=(X\_test, y\_test))

### Model.summary()

A screenshot of a computer

Description automatically generated

### Hyperparameters

|  |  |  |
| --- | --- | --- |
| Hyper parameter | Value | Comments |
| LSTM layer | 2 | Not big impact when add the second layer of LSTM |
| Optimizer | Adam |  |
| Loos | MSE |  |
| EPOCHS | 100 |  |
| Batch\_size | 64 |  |

### Loss function result

A graph of a training loss

Description automatically generated

From the result of loss function by each epoch in above chart, the model fitted well, neither overfitting nor under fitting.

### Prediction result

A graph showing the price of a stock market

Description automatically generated

Compare the predicted price and the actual price, the predicated price match with the acctual value well. The error is small by 0.12997036388668592 (MSE)

### Zoom-in

A graph of a graph showing the price of a stock market

Description automatically generated

Zoom in the prediction chart shows the gap between prediction and actual data.

### MSE value:

MSE: 0.12997036388668592