```
!pip install yfinance torch torchvision
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.11/dist-packages (0.2.55)
Requirement already satisfied: torch in /usr/local/lib/python3.11/dist-packages (2.6.0+cu124)
Requirement already satisfied: torchvision in /usr/local/lib/python3.11/dist-packages (0.21.0+cu124)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.2.2)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.0.2)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.11/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: platformdirs>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (4.3.7)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2025.2)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.4.6)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.11/dist-packages (from yfinance) (3.17.9)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /usr/local/lib/python3.11/dist-packages (from yfinance) (4.13.3
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from torch) (3.18.0)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.11/dist-packages (from torch) (4.13.1
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (from torch) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from torch) (2025.3.2)
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch)
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cublas-cu12==12.4.5.8 (from torch)
  Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cufft-cu12==11.2.1.3 (from torch)
  Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-curand-cu12==10.3.5.147 (from torch)
  Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch)
  Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cusparse-cu12==12.3.1.170 (from torch)
  Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl.metadata (1.6 kB)
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib/python3.11/dist-packages (from torch) (0.
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/python3.11/dist-packages (from torch) (2.21.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch) (12.4.
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch)
  Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist-packages (from torch) (3.2.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist-packages (from torch) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from sympy==1.13.1->torch)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.11/dist-packages (from torchvision) (11.1
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4>=4.11.1->yf
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.3.0->yf
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.3.0->yfinance)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31-
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfina
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfina
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->torch) (3.0.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas>
Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl (363.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 4.1 MB/s eta 0:00:00
Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (13.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 48.0 MB/s eta 0:00:00
Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (24.6 MB)
```

```
# Import standard libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import datetime

# ----------------------------
# NLP and Sentiment Libraries
# ----------------------------
import nltk
# Download required NLTK data for VADER
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer

from transformers import pipeline

# ----------------------------
# Finance Data Extraction
# ----------------------------
import yfinance as yf
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

```
import pandas as pd
import yfinance as yf
import numpy as np
from sklearn.preprocessing import MinMaxScaler, StandardScaler
import torch
import torch.nn as nn
import matplotlib.pyplot as plt
from torch.autograd import Variable
from sklearn.metrics import mean_squared_error, mean_absolute_error
from keras.models import Sequential
from keras.layers import Dense, LSTM, Bidirectional, Dropout
import math
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import matplotlib.dates as mdates
from pandas.tseries.holiday import USFederalHolidayCalendar
from pandas.tseries.offsets import CustomBusinessDay
from datetime import datetime, timedelta
from collections import Counter
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import seaborn as sns
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# Past Financial data of GameStop
tickerSymbol = 'GME'
tickerData = yf.Ticker(tickerSymbol)
stock_prices_df = tickerData.history(period='1d', start='2021-01-04', end='2021-12-31')
columns = stock_prices_df.columns.tolist()
close_index = columns.index('Close')
columns = [columns[close_index]] + columns[:close_index] + columns[close_index + 1:]
stock_prices_df = stock_prices_df[columns]
stock_prices_df.head(3)
```

Date	Close	Open	High	Low	Volume	Dividends	Stock Splits
2021-01-04 00:00:00-05:00	4.3125	4.7500	4.775	4.2875	40090000	0.0	0.0
2021-01-05 00:00:00-05:00	4.3425	4.3375	4.520	4.3075	19846000	0.0	0.0
2021-01-06 00:00:00-05:00	4.5900	4.3350	4.745	4.3325	24224800	0.0	0.0

```
# Reddit Sentiment Data from Harvard
path = '/content/drive/MyDrive/rGME_dataset_features.csv'
reddit_harvard_df = pd.read_csv(path)
reddit_harvard_df.head()
```

```
<ipython-input-7-24b96faadf11>:3: DtypeWarning: Columns (8) have mixed types. Specify dtype option on import or set low_
  reddit_harvard_df = pd.read_csv(path)
```

	Unnamed: 0	id	title	url	score	author	num_comments	date	flair	c
0	0	kqfajb	You NEED to see this about GME 🚀🚀🚀🚀🚀🚀	https://www.reddit.com/r/GME/comments/kqfajb/y...	1.0	TitsDownOnly	9.0	2021-01-04	NaN	
1	1	kqjh2t	Short Squeeze Incoming 🚀🚀🚀🚀🚀🚀	/r/wallstreetbets/comments/kqcwdo/gamestops_gr...	1.0	zoomermoney	1.0	2021-01-04	NaN	
2	2	kqvp7l	THIS CONVINCED ME TO ALL IN 💰 GME (EXTREME PUMP...	https://www.reddit.com/r/GME/comments/kqvp7l/t...	1.0	TitsDownOnly	6.0	2021-01-05	NaN	
3	3	krcwch	You already know what we must do brothers and ...	/r/wallstreetbets/comments/kr98ym/gme_gang_we_...	1.0	dontforgettolive	4.0	2021-01-05	NaN	
4	4	krnthg	ICR conference (11th Jan)	https://www.reddit.com/r/GME/comments/krnthg/i...	1.0	nicky94	10.0	2021-01-06	NaN	

5 rows × 74 columns

```
# Reddit WallStreetBets Posts
path = '/content/drive/MyDrive/reddit_wsb.csv'
reddit_kaggle_df = pd.read_csv(path)
reddit_kaggle_df.head()
```

	title	score	id	url	comms_num	created	body	timestamp
0	It's not about the money, it's about sending a...	55	l6ulcx	https://v.redd.it/6j75regs72e61	6	1.611863e+09	NaN	2021-01-28 21:37:41
1	Math Professor Scott Steiner says the numbers ...	110	l6uibd	https://v.redd.it/ah50lyny62e61	23	1.611862e+09	NaN	2021-01-28 21:32:10
2	Exit the system	0	l6uhhn	https://www.reddit.com/r/wallstreetbets/commen...	47	1.611862e+09	The CEO of NASDAQ pushed to halt trading “to g...	2021-01-28 21:30:35
	NEW SEC FILING FOR GME! CAN			https://sec.report/Document/0001193125-21-				2021-01-28

```
# Print the dtypes to confirm the data types
print("Dataset dtypes:")
print(reddit_kaggle_df.dtypes)

# Combine the title and body into one new text column for our analysis
reddit_kaggle_df['text'] = reddit_kaggle_df['title'].fillna('') + " " + reddit_kaggle_df['body'].fillna('')
```

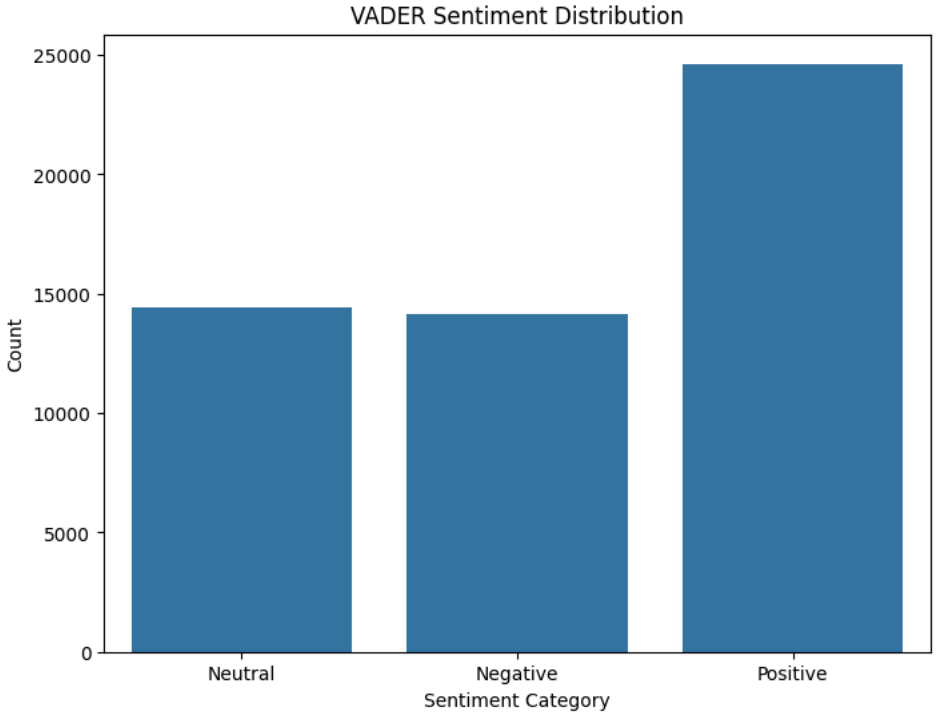
```
Dataset dtypes:
title         object
score          int64
id            object
url           object
comms_num      int64
created      float64
body          object
timestamp     object
dtype: object
```

```
# ============================
# 2. Data Preprocessing
# ============================

def clean_text(text):
    """
    Perform basic text cleaning:
      - Remove URLs.
      - Remove HTML tags.
```

```
      - Remove punctuation.
      - Convert to lowercase.
      - Remove extra whitespace.
    """
    # Remove URLs
    text = re.sub(r'http\S+', '', text)
    # Remove HTML tags
    text = re.sub(r'<.*?>', '', text)
    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Convert to lowercase and strip extra whitespace
    text = text.lower()
    text = re.sub(r'\s+', ' ', text).strip()
    return text

# Apply cleaning to the combined text column
reddit_kaggle_df['clean_text'] = reddit_kaggle_df['text'].apply(clean_text)
```

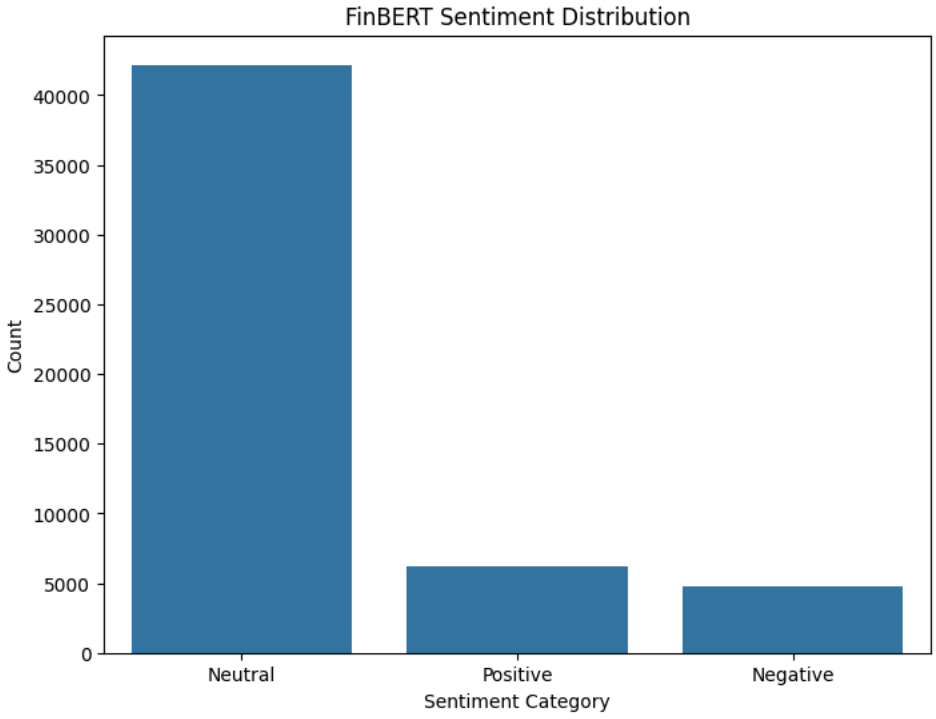
```
# ============================
# 3. Sentiment Analysis Using VADER
# ============================

# Initialize the VADER sentiment analyzer
sid = SentimentIntensityAnalyzer()

def get_vader_sentiment(text):
    """
    Determine sentiment using NLTK's VADER.
    Return a tuple: (sentiment_label, compound_score).
    """
    scores = sid.polarity_scores(text)
    compound = scores['compound']
    if compound >= 0.05:
        sentiment = "Positive"
    elif compound <= -0.05:
        sentiment = "Negative"
    else:
        sentiment = "Neutral"
    return sentiment, compound

# Apply the VADER sentiment function to the cleaned text
reddit_kaggle_df[['vader_sentiment', 'vader_compound']] = reddit_kaggle_df['clean_text'].apply(
    lambda x: pd.Series(get_vader_sentiment(x))
)

# Display a sample of VADER sentiment results
print("\nVADER Sentiment Examples:")
print(reddit_kaggle_df[['vader_sentiment', 'vader_compound']].head())

# Visualize the distribution of VADER sentiments
plt.figure(figsize=(8,6))
sns.countplot(x='vader_sentiment', data=reddit_kaggle_df)
plt.title("VADER Sentiment Distribution")
plt.xlabel("Sentiment Category")
plt.ylabel("Count")
plt.show()
```

```
VADER Sentiment Examples:
  vader_sentiment  vader_compound
0         Neutral          0.0000
1        Negative         -0.6249
2        Negative         -0.6124
3        Negative         -0.2748
4        Positive          0.2235
```




```
# ============================
# 4. Sentiment Analysis Using FinBERT
# ============================

# Load the FinBERT sentiment model from Hugging Face
finbert = pipeline("sentiment-analysis",
                   model="yiyanghkust/finbert-tone",
                   tokenizer="yiyanghkust/finbert-tone")

def get_finbert_sentiment(text):
    """
    Use FinBERT to analyze sentiment.
    Input text is capped at 512 characters to avoid token-length issues.
    Returns a tuple: (sentiment_label, score)
    """
    try:
        result = finbert(text[:512])[0]
        label = result['label']
        score = result['score']
        return label, score
    except Exception as e:
        # In case of any error, return None values.
        return None, None

# Apply FinBERT sentiment analysis to the cleaned texts (processing all rows)
reddit_kaggle_df[['finbert_sentiment', 'finbert_score']] = reddit_kaggle_df['clean_text'].apply(
    lambda x: pd.Series(get_finbert_sentiment(x))
)

# Display a sample of FinBERT sentiment results
print("\nFinBERT Sentiment Examples:")
print(reddit_kaggle_df[['finbert_sentiment', 'finbert_score']].head())

# Visualize the distribution of FinBERT sentiments
plt.figure(figsize=(8,6))
sns.countplot(x='finbert_sentiment', data=reddit_kaggle_df)
plt.title("FinBERT Sentiment Distribution")
plt.xlabel("Sentiment Category")
plt.ylabel("Count")
plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens),
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100%                                                 533/533 [00:00<00:00, 13.3kB/s]
pytorch_model.bin: 100%                                           439M/439M [00:02<00:00, 161MB/s]
vocab.txt: 100%                                                   226k/226k [00:00<00:00, 3.69MB/s]
model.safetensors: 100%                                           439M/439M [00:02<00:00, 232MB/s]
Device set to use cuda:0
You seem to be using the pipelines sequentially on GPU. In order to maximize efficiency please use a dataset

FinBERT Sentiment Examples:
  finbert_sentiment  finbert_score
0           Neutral       0.997360
1           Neutral       0.740807
2           Neutral       0.626472
3           Neutral       0.999743
4           Neutral       0.812407
```




```
# Checking the missing values
missing_values = stock_prices_df.isnull().sum()
missing_values
```

	0
Close	0
Open	0
High	0
Low	0
Volume	0
Dividends	0
Stock Splits	0

dtype: int64

```
reddit_harvard_df['date'] = pd.to_datetime(reddit_harvard_df['date'])
sentiment_aggregated = reddit_harvard_df.groupby('date').agg({'compound': 'mean', 'neg': 'mean', 'neu': 'mean', 'pos': 'mean
sentiment_aggregated.head(3)
```

	date	compound	neg	neu	pos
0	2021-01-04	0.98890	0.0000	0.1340	0.8660
1	2021-01-05	0.11795	0.0575	0.8145	0.1280
2	2021-01-06	0.38885	0.0535	0.7580	0.1885

```
# Checking the missing values
missing_values = sentiment_aggregated.isnull().sum()
missing_values
```

	0
date	0
compound	0
neg	0
neu	0
pos	0

dtype: int64

```
#Time series forecasting model for financial data
# Splitting the data into train, validation and test sets

dataNum = 5
timesteps = 20
epochNum = 20

#Train
dataset_train = stock_prices_df[stock_prices_df.index < '2021-06-01']
print(dataset_train.shape)
training_set = dataset_train.iloc[:,1:dataNum+1].values
print(training_set.shape)

sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
print(training_set_scaled.shape)

X_train = []
Y_train = []
for i in range(timesteps, len(training_set_scaled)):
    X_train.append(training_set_scaled[i-timesteps:i, 0:dataNum])
    Y_train.append(training_set_scaled[i, 0])
X_train, Y_train = np.array(X_train), np.array(Y_train)
print(X_train.shape, Y_train.shape)
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], dataNum))
print(X_train.shape, Y_train.shape)

val_df = stock_prices_df[(stock_prices_df.index >= '2021-09-01') & (stock_prices_df.index <= '2021-12-31')]
validation_set = val_df.iloc[:, 1:dataNum+1].values

validation_set_scaled = sc.transform(validation_set)

X_val = []
Y_val = []
for i in range(timesteps, len(validation_set_scaled)):
    X_val.append(validation_set_scaled[i-timesteps:i, 0:dataNum])
    Y_val.append(validation_set_scaled[i, 0])
X_val, Y_val = np.array(X_val), np.array(Y_val)
X_val = np.reshape(X_val, (X_val.shape[0], X_val.shape[1], dataNum))

#Test
test_set = stock_prices_df[(stock_prices_df.index >= '2021-06-01') & (stock_prices_df.index < '2021-09-01')]
print(test_set.shape)
real_stock_price = test_set.iloc[:,1:dataNum+1].values
print(real_stock_price.shape)
lenOfReal = len(real_stock_price)
inputs = real_stock_price
inputs = sc.transform(inputs)
print(inputs.shape)

# Combine the last 20 days of training data with the test data
last_20_days_training = training_set_scaled[-timesteps:]
print('last_20_days_training', last_20_days_training.shape)
combined_test_data = np.concatenate((last_20_days_training, inputs))
print('inputs', inputs.shape)
print('combined_test_data', combined_test_data.shape)
```

```
#inputs_test using the combined data
inputs_test = []
for i in range(timesteps, len(inputs) + timesteps):
    inputs_test.append(combined_test_data[i-timesteps:i, 0:dataNum])
inputs_test = np.array(inputs_test)
inputs_test = np.reshape(inputs_test, (inputs_test.shape[0], inputs_test.shape[1], dataNum))
print('inputs_test', inputs_test.shape)
```

```
(102, 7)
(102, 5)
(102, 5)
(82, 20, 5) (82,)
(82, 20, 5) (82,)
(65, 7)
(65, 5)
(65, 5)
last_20_days_training (20, 5)
inputs (65, 5)
combined_test_data (85, 5)
inputs_test (65, 20, 5)
```

```
!pip install --upgrade tensorflow
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.11/dist-packages (2.18.0)
Collecting tensorflow
  Downloading tensorflow-2.19.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.1 kB)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (25.2.1
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tens
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from tensorflow) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in /usr/
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tensorflow) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.0.1)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (4.
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.71.0)
Collecting tensorboard~=2.19.0 (from tensorflow)
  Downloading tensorboard-2.19.0-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0)
Requirement already satisfied: numpy<2.2.0,>=1.26.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.0.2)
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.13.0)
Collecting ml-dtypes<1.0.0,>=0.5.1 (from tensorflow)
  Downloading ml_dtypes-0.5.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (21 kB)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from ten
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0->te
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.14.1
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensor
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorboard~=2.19.0->ten
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from te
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard~=2.19.0->ten
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tenso
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->
Downloading tensorflow-2.19.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (644.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 644.9/644.9 MB 1.6 MB/s eta 0:00:00
Downloading ml_dtypes-0.5.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.7/4.7 MB 95.5 MB/s eta 0:00:00
Downloading tensorboard-2.19.0-py3-none-any.whl (5.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.5/5.5 MB 62.6 MB/s eta 0:00:00
Installing collected packages: ml-dtypes, tensorboard, tensorflow
  Attempting uninstall: ml-dtypes
    Found existing installation: ml-dtypes 0.4.1
    Uninstalling ml-dtypes-0.4.1:
      Successfully uninstalled ml-dtypes-0.4.1
  Attempting uninstall: tensorboard
    Found existing installation: tensorboard 2.18.0
    Uninstalling tensorboard-2.18.0:
      Successfully uninstalled tensorboard-2.18.0
  Attempting uninstall: tensorflow
    Found existing installation: tensorflow 2.18.0
    Uninstalling tensorflow-2.18.0:
      Successfully uninstalled tensorflow-2.18.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviou
tensorflow-text 2.18.1 requires tensorflow<2.19,>=2.18.0, but you have tensorflow 2.19.0 which is incompatible.
tf-keras 2.18.0 requires tensorflow<2.19,>=2.18, but you have tensorflow 2.19.0 which is incompatible.
tensorflow-decision-forests 1.11.0 requires tensorflow==2.18.0, but you have tensorflow 2.19.0 which is incompatible.
Successfully installed ml-dtypes-0.5.1 tensorboard-2.19.0 tensorflow-2.19.0
```

```
print("X_train shape:", X_train.shape)
print("Y_train shape:", Y_train.shape)
```

```
X_train shape: (82, 20, 5)
Y_train shape: (82,)
```

```
!nvidia-smi
```

```
Mon Apr 14 13:12:29 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla T4                       Off |   00000000:00:04.0 Off |                    0 |
| N/A   66C    P0             31W /   70W |     168MiB /  15360MiB |      0%      Default |
```

```
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                             GPU Memory |
|        ID   ID                                                              Usage      |
|=======================================================================================|
+---------------------------------------------------------------------------------------+
```

```
X_train = X_train.astype('float32')
Y_train = Y_train.astype('float32')
X_val = X_val.astype('float32')
Y_val = Y_val.astype('float32')
```

```
print("X_train shape:", X_train.shape)
print("Y_train shape:", Y_train.shape)
```

```
X_train shape: (82, 20, 5)
Y_train shape: (82,)
```

```
import os
os.environ["TF_DISABLE_CUDNN_RNN"] = "1"
```

```
model = Sequential()
model.add(LSTM(units=500, input_shape=(X_train.shape[1], dataNum), return_sequences=True))
model.add(LSTM(units=400))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mean_squared_error'])
model.summary()

history = model.fit(X_train, Y_train, validation_data=(X_val, Y_val), batch_size=32, epochs=epochNum)
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(**kwargs)
```

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 20, 500)	1,012,000
lstm_1 (LSTM)	(None, 400)	1,441,600
dense (Dense)	(None, 1)	401

Total params: 2,454,001 (9.36 MB)
Trainable params: 2,454,001 (9.36 MB)
Non-trainable params: 0 (0.00 B)

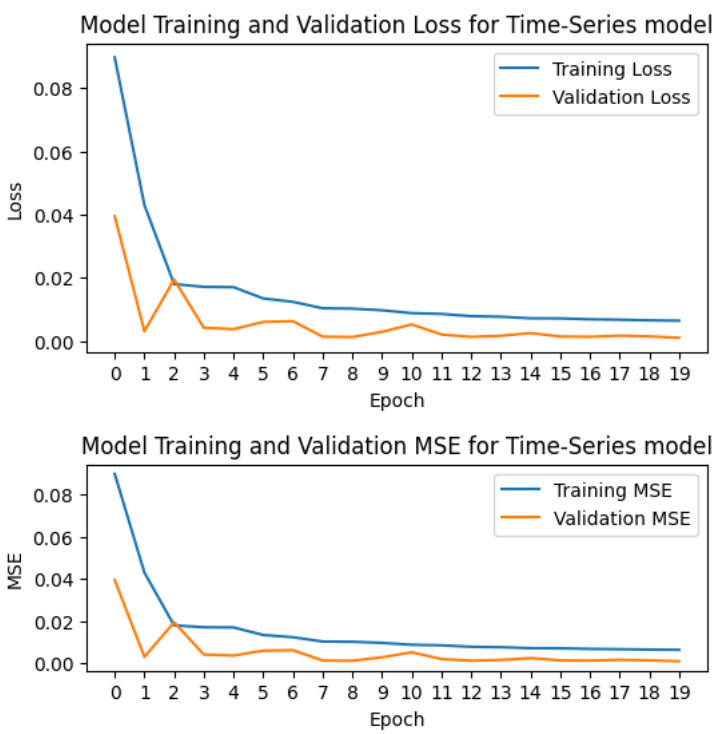
```
Epoch 1/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 4s 497ms/step - loss: 0.1078 - mean_squared_error: 0.1078 - val_loss: 0.0395 - val_mean_squared
Epoch 2/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.0441 - mean_squared_error: 0.0441 - val_loss: 0.0031 - val_mean_squared_
Epoch 3/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.0183 - mean_squared_error: 0.0183 - val_loss: 0.0194 - val_mean_squared_
Epoch 4/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0174 - mean_squared_error: 0.0174 - val_loss: 0.0042 - val_mean_squared_
Epoch 5/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0140 - mean_squared_error: 0.0140 - val_loss: 0.0037 - val_mean_squared_
Epoch 6/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0150 - mean_squared_error: 0.0150 - val_loss: 0.0060 - val_mean_squared_
Epoch 7/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.0125 - mean_squared_error: 0.0125 - val_loss: 0.0063 - val_mean_squared_
Epoch 8/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0099 - mean_squared_error: 0.0099 - val_loss: 0.0014 - val_mean_squared_
Epoch 9/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0103 - mean_squared_error: 0.0103 - val_loss: 0.0012 - val_mean_squared_
Epoch 10/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.0089 - mean_squared_error: 0.0089 - val_loss: 0.0029 - val_mean_squared_
Epoch 11/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step - loss: 0.0087 - mean_squared_error: 0.0087 - val_loss: 0.0052 - val_mean_squared_
Epoch 12/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0086 - mean_squared_error: 0.0086 - val_loss: 0.0020 - val_mean_squared_
Epoch 13/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.0092 - mean_squared_error: 0.0092 - val_loss: 0.0013 - val_mean_squared_
Epoch 14/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.0080 - mean_squared_error: 0.0080 - val_loss: 0.0016 - val_mean_squared_
Epoch 15/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0068 - mean_squared_error: 0.0068 - val_loss: 0.0025 - val_mean_squared_
Epoch 16/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.0071 - mean_squared_error: 0.0071 - val_loss: 0.0014 - val_mean_squared_
Epoch 17/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0080 - mean_squared_error: 0.0080 - val_loss: 0.0013 - val_mean_squared_
Epoch 18/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0070 - mean_squared_error: 0.0070 - val_loss: 0.0017 - val_mean_squared_
Epoch 19/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0061 - mean_squared_error: 0.0061 - val_loss: 0.0014 - val_mean_squared_
Epoch 20/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - loss: 0.0064 - mean_squared_error: 0.0064 - val_loss: 9.9092e-04 - val_mean_squa
```

```
history = model.fit(X_train, Y_train, validation_data=(X_val, Y_val), batch_size=32, epochs=epochNum)
```

```
Epoch 1/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 8s 833ms/step - loss: 0.1332 - mean_squared_error: 0.1332 - val_loss: 0.0295 - val_mean_squared
Epoch 2/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 497ms/step - loss: 0.0383 - mean_squared_error: 0.0383 - val_loss: 0.0072 - val_mean_squared
Epoch 3/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 548ms/step - loss: 0.0165 - mean_squared_error: 0.0165 - val_loss: 0.0308 - val_mean_squared
Epoch 4/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 472ms/step - loss: 0.0204 - mean_squared_error: 0.0204 - val_loss: 0.0017 - val_mean_squared
Epoch 5/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 542ms/step - loss: 0.0152 - mean_squared_error: 0.0152 - val_loss: 0.0090 - val_mean_squared
Epoch 6/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 767ms/step - loss: 0.0173 - mean_squared_error: 0.0173 - val_loss: 0.0013 - val_mean_squared
Epoch 7/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 962ms/step - loss: 0.0100 - mean_squared_error: 0.0100 - val_loss: 0.0092 - val_mean_squared
Epoch 8/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 4s 502ms/step - loss: 0.0119 - mean_squared_error: 0.0119 - val_loss: 0.0059 - val_mean_squared
Epoch 9/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 560ms/step - loss: 0.0108 - mean_squared_error: 0.0108 - val_loss: 0.0012 - val_mean_squared
Epoch 10/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 545ms/step - loss: 0.0105 - mean_squared_error: 0.0105 - val_loss: 0.0012 - val_mean_squared
Epoch 11/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 565ms/step - loss: 0.0100 - mean_squared_error: 0.0100 - val_loss: 0.0016 - val_mean_squared
Epoch 12/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 828ms/step - loss: 0.0105 - mean_squared_error: 0.0105 - val_loss: 0.0032 - val_mean_squared
Epoch 13/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 561ms/step - loss: 0.0087 - mean_squared_error: 0.0087 - val_loss: 0.0022 - val_mean_squared
Epoch 14/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 456ms/step - loss: 0.0074 - mean_squared_error: 0.0074 - val_loss: 0.0014 - val_mean_squared
Epoch 15/20
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 444ms/step - loss: 0.0070 - mean_squared_error: 0.0070 - val_loss: 0.0020 - val_mean_squared
Epoch 16/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 505ms/step - loss: 0.0073 - mean_squared_error: 0.0073 - val_loss: 0.0032 - val_mean_squared
Epoch 17/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 3s 718ms/step - loss: 0.0065 - mean_squared_error: 0.0065 - val_loss: 0.0013 - val_mean_squared
Epoch 18/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 4s 1s/step - loss: 0.0066 - mean_squared_error: 0.0066 - val_loss: 0.0011 - val_mean_squared_er
Epoch 19/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 2s 529ms/step - loss: 0.0071 - mean_squared_error: 0.0071 - val_loss: 0.0022 - val_mean_squared
Epoch 20/20
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 526ms/step - loss: 0.0072 - mean_squared_error: 0.0072 - val_loss: 0.0016 - val_mean_squared
```

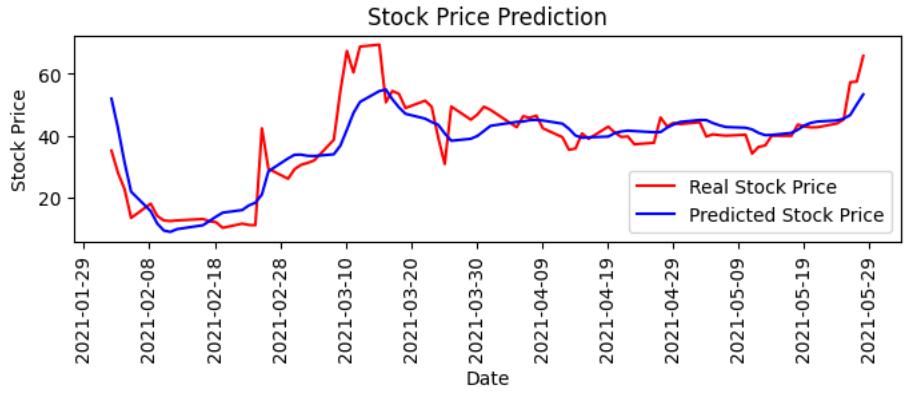
```
# Plot the training and validation loss
plt.figure(figsize=(6, 3))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Training and Validation Loss for Time-Series model')
plt.ylabel('Loss')
plt.xlabel('Epoch')
epoch_range = range(0, epochNum)
plt.xticks(epoch_range)
plt.legend()
plt.show()

# Plot the training and validation MSE
plt.figure(figsize=(6, 2))
plt.plot(history.history['mean_squared_error'], label='Training MSE')
plt.plot(history.history['val_mean_squared_error'], label='Validation MSE')
plt.title('Model Training and Validation MSE for Time-Series model')
plt.ylabel('MSE')
plt.xlabel('Epoch')
plt.xticks(epoch_range)
plt.legend()
plt.show()
```




```
# Prediction on train data based model

# Actual
real_train = dataset_train
real_train = dataset_train.iloc[timesteps:len(real_train)+1,1:2].values
train_dates = dataset_train.index.to_list()
adjusted_train_dates = train_dates[timesteps:len(real_train) + timesteps]

# predicted
predicted_train = model.predict(X_train)
predicted_train = np.pad(predicted_train,((0,0),(0,dataNum-1)),'constant')
predicted_train = sc.inverse_transform(predicted_train)
predicted_train = np.delete(predicted_train, [1, 2, 3, 4], axis=1)
np.savetxt('GME_pred_train' + '.csv', predicted_train, fmt="%.3f", delimiter=",")

# chart
```

```
plt.figure(figsize=(8, 2))
plt.plot(adjusted_train_dates, real_train, color = 'red', label = 'Real Stock Price')
plt.plot(adjusted_train_dates, predicted_train, color = 'blue', label = 'Predicted Stock Price')
plt.title('Stock Price Prediction')
plt.xlabel('Date')
plt.ylabel('Stock Price')
plt.legend()
plt.xticks(rotation=90)
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=10))
plt.show()
plt.savefig('pic1.png')
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 88ms/step
```



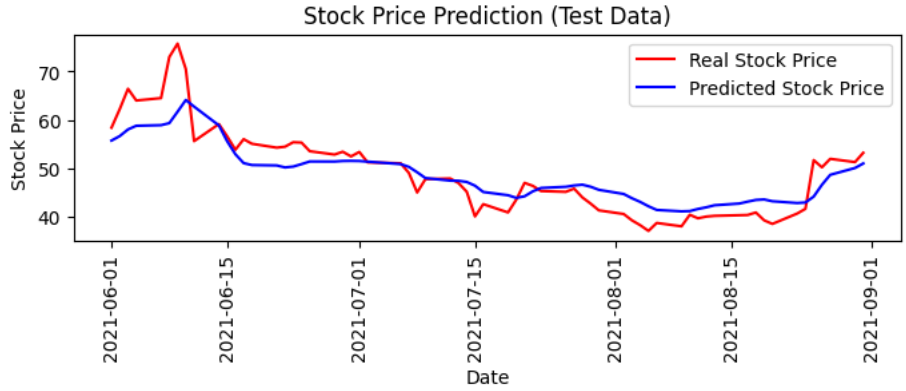

```
<Figure size 640x480 with 0 Axes>
```

```
train_df = pd.DataFrame({
    'Date': pd.to_datetime(adjusted_train_dates).date,
    'Real_Price_Train': real_train.flatten(),
    'Predicted_Price_Train': predicted_train.flatten()})
train_df.tail()
```

	Date	Real_Price_Train	Predicted_Price_Train
77	2021-05-24	43.962502	44.957954
78	2021-05-25	45.250000	45.497520
79	2021-05-26	57.250000	46.676220
80	2021-05-27	57.450001	50.077217
81	2021-05-28	65.742500	53.332935

```
# Prediction on test data

# Actual
print('test_set', test_set.shape)
real_test = test_set.iloc[0:lenOfReal+1,1:2].values # test_set.iloc[timesteps:lenOfReal+1,1:2].values
print('real_test', real_test.shape)
test_dates = test_set.index[0:lenOfReal+1] # test_set.index[timesteps:lenOfReal+1]
print('test_dates', test_dates.shape)

# predicted
print('inputs_test', inputs_test.shape)
predicted_test = model.predict(inputs_test)
print('predicted_test', predicted_test.shape)
predicted_test = np.pad(predicted_test,((0,0),(0,dataNum-1)),'constant')
predicted_test = sc.inverse_transform(predicted_test)
predicted_test = np.delete(predicted_test, [1, 2, 3, 4], axis=1)
print(inputs_test.shape)
predicted_test = predicted_test[:len(test_dates)] # NEW
print(inputs_test.shape)
np.savetxt('GME_pred_test' + '.csv', predicted_test, fmt="%.3f", delimiter=",")

## chart
plt.figure(figsize=(8, 2))
plt.plot(test_dates, real_test, color = 'red', label = 'Real Stock Price')
plt.plot(test_dates, predicted_test, color = 'blue', label = 'Predicted Stock Price')
plt.title('Stock Price Prediction (Test Data)')
plt.xlabel('Date')
plt.ylabel('Stock Price')
plt.xticks(rotation=90)
plt.legend()
plt.show()
plt.savefig('pic2.png')
```

```
test_set (65, 7)
real_test (65, 1)
test_dates (65,)
inputs_test (65, 20, 5)
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step
predicted_test (65, 1)
(65, 20, 5)
(65, 20, 5)
```



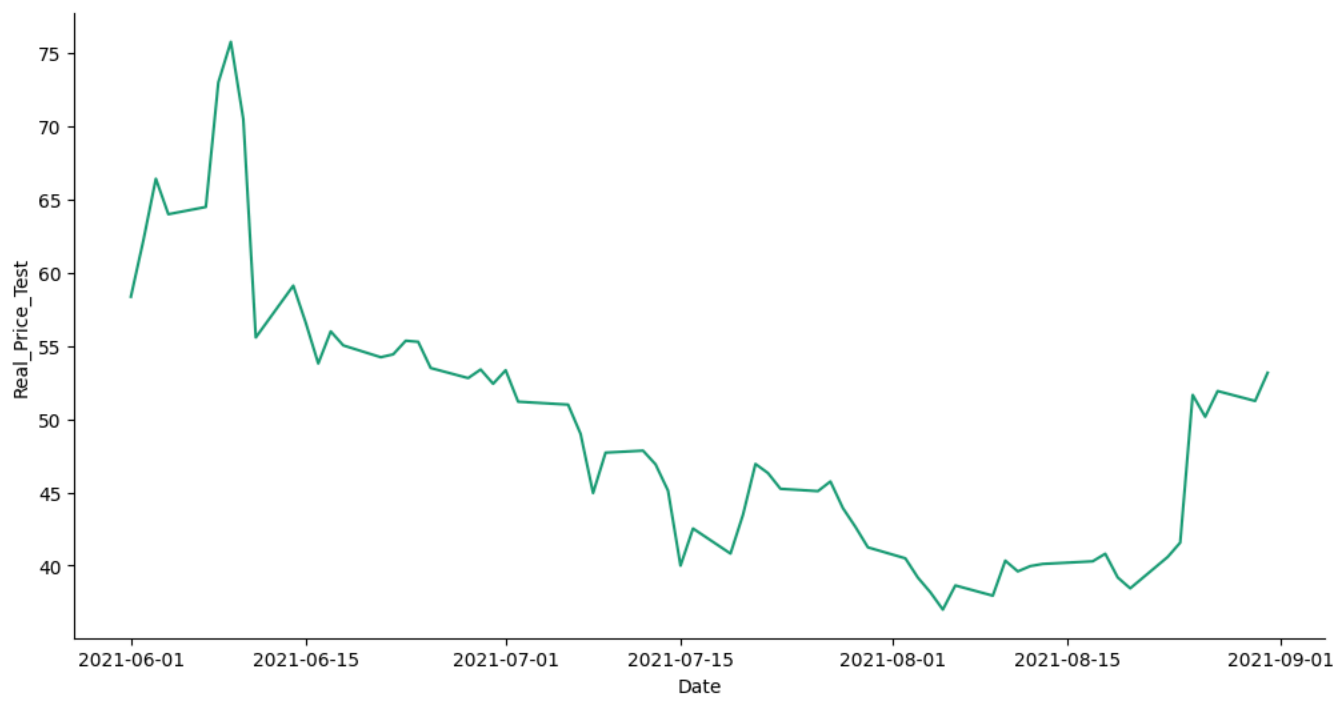

```
<Figure size 640x480 with 0 Axes>
```

```
test_df = pd.DataFrame({
    'Date': pd.to_datetime(test_dates).date,
    'Real_Price_Test': real_test.flatten(),
    'Predicted_Price_Test': predicted_test.flatten()})
test_df.head()
```

	Date	Real_Price_Test	Predicted_Price_Test
0	2021-06-01	58.369999	55.703697
1	2021-06-02	62.220001	56.632839
2	2021-06-03	66.427498	58.010185
3	2021-06-04	64.004997	58.774548
4	2021-06-07	64.500000	58.898399

Date vs Real_Price_Test

Show code




```
# Key stats

## train
```

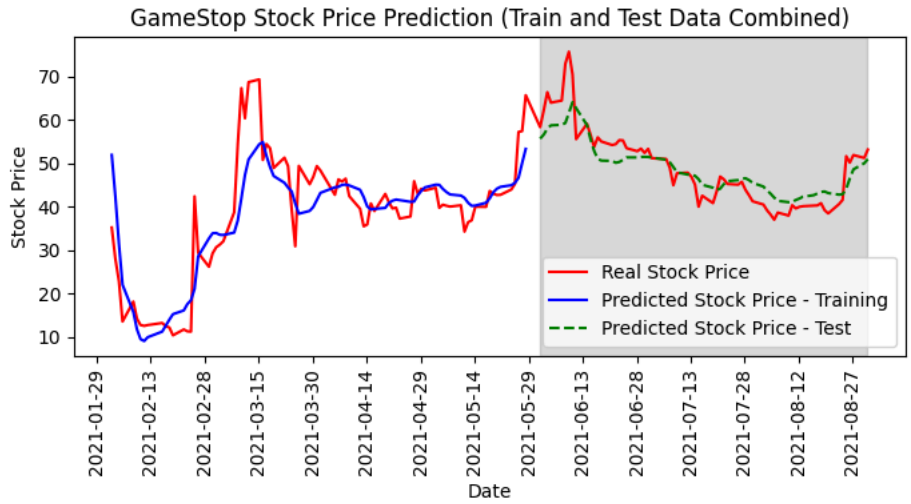
```
print('train_dates', len(train_dates))
print('adjusted_train_dates', len(adjusted_train_dates))
print('real_train', len(real_train))
print('predicted_train', len(predicted_train))

## test
print('\ntest_dates', len(test_dates))
print('real_test', len(real_test))
# print('predicted_test', len(predicted_test))
```

```
train_dates 102
adjusted_train_dates 82
real_train 82
predicted_train 82

test_dates 65
real_test 65
```

```
# Evaluation metrics

# train
mseTrain = round(mean_squared_error(real_train, predicted_train), 2)
rmseTrain = round(math.sqrt(mseTrain), 2)
maeTrain = round(mean_absolute_error(real_train, predicted_train), 2)

print("Training Data Metrics:")
print("MSE_train = " + str(mseTrain))
print("RMSE_train = " + str(rmseTrain))
print("MAE_train = " + str(maeTrain))

# test
mseTest = round(mean_squared_error(real_test, predicted_test), 2)
rmseTest = round(math.sqrt(mseTest), 2)
maeTest = round(mean_absolute_error(real_test, predicted_test), 2)

print("\nTesting Data Metrics:")
print("MSE_test = " + str(mseTest))
print("RMSE_test = " + str(rmseTest))
print("MAE_test = " + str(maeTest))
```

```
Training Data Metrics:
MSE_train = 51.43
RMSE_train = 7.17
MAE_train = 5.07

Testing Data Metrics:
MSE_test = 18.27
RMSE_test = 4.27
MAE_test = 3.29
```

```
# Combined chart

combined_dates = np.concatenate((adjusted_train_dates, test_dates)) # Combine the dates
combined_real = np.concatenate((real_train, real_test)) # Combine the real prices
combined_predicted = np.concatenate((predicted_train, predicted_test)) # Combine the predicted prices

# Create the chart
plt.figure(figsize=(7, 4))
plt.plot(combined_dates, combined_real, color='red', label='Real Stock Price') # Plot the real stock prices
plt.plot(adjusted_train_dates, predicted_train, color='blue', label='Predicted Stock Price - Training') # Plot the predictec
plt.plot(test_dates, predicted_test, color='green', linestyle='dashed', label='Predicted Stock Price - Test') # Plot the pre
plt.title('GameStop Stock Price Prediction (Train and Test Data Combined)')
plt.xlabel('Date')
plt.ylabel('Stock Price')
plt.legend()

start_date = pd.to_datetime('2021-06-01')
end_date = pd.to_datetime('2021-08-31')
plt.axvspan(start_date, end_date, color='grey', alpha=0.3)

plt.xticks(rotation=90)
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=15))
plt.tight_layout()
plt.show()
plt.savefig('pic3.png')
```




```
<Figure size 640x480 with 0 Axes>
```

```
# Prepare financial data for merging with the sentiment data
stock_prices_df2 = stock_prices_df[['Close', 'Open', 'High', 'Low', 'Volume']].copy()
stock_prices_df2.reset_index(inplace=True)
stock_prices_df2.rename(columns={'Date': 'date'}, inplace=True)
stock_prices_df2['date'] = stock_prices_df2['date'].dt.tz_localize(None)
stock_prices_df2['date'] = stock_prices_df2['date'].dt.strftime('%Y-%m-%d')
stock_prices_df2['date'] = pd.to_datetime(stock_prices_df2['date'])
stock_prices_df2.head(3)
```

	date	Close	Open	High	Low	Volume
0	2021-01-04	4.3125	4.7500	4.775	4.2875	40090000
1	2021-01-05	4.3425	4.3375	4.520	4.3075	19846000
2	2021-01-06	4.5900	4.3350	4.745	4.3325	24224800

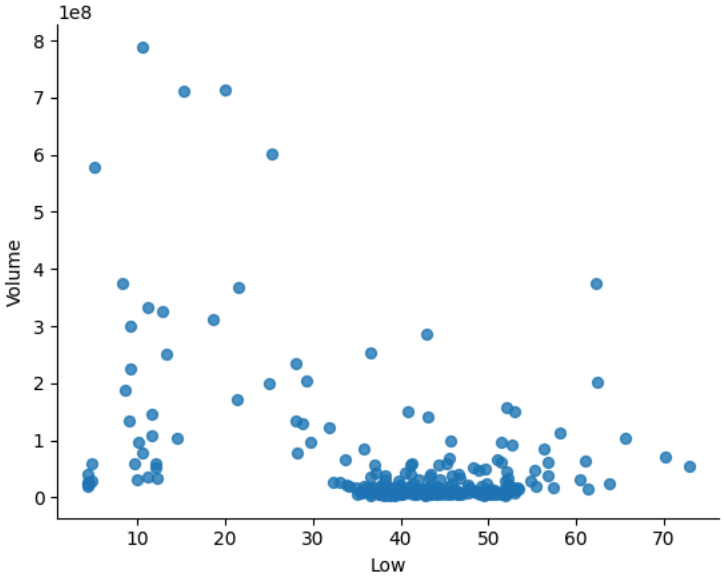
```
# Merge financial data with the sentiment data
combined_df = stock_prices_df2.merge(sentiment_aggregated, on='date', how='left')
combined_df.fillna(method='ffill', inplace=True)
combined_df.set_index('date', inplace=True)
print(combined_df.shape)
combined_df.head(3)
```

```
(251, 9)
<ipython-input-32-b531b6e6dc08>:3: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a futur
  combined_df.fillna(method='ffill', inplace=True)
```

	Close	Open	High	Low	Volume	compound	neg	neu	pos
date									
2021-01-04	4.3125	4.7500	4.775	4.2875	40090000	0.98890	0.0000	0.1340	0.8660
2021-01-05	4.3425	4.3375	4.520	4.3075	19846000	0.11795	0.0575	0.8145	0.1280
2021-01-06	4.5900	4.3350	4.745	4.3325	24224800	0.38885	0.0535	0.7580	0.1885

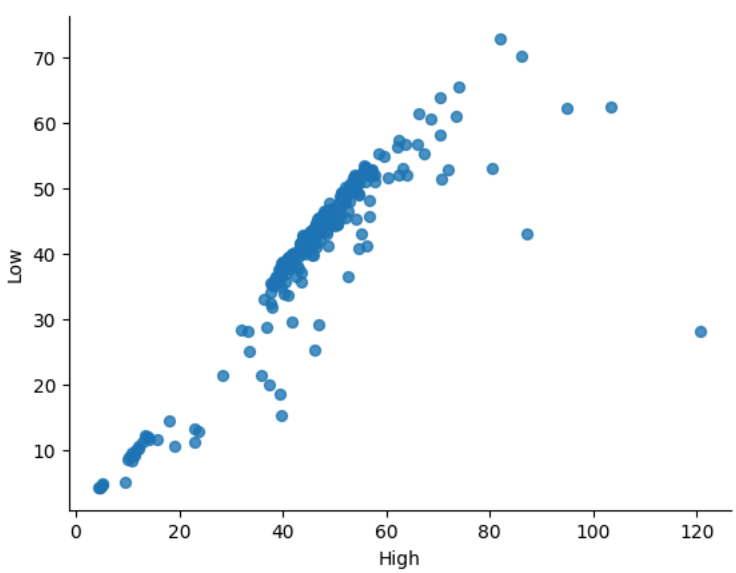
Low vs Volume

Show code



High vs Low

Show code




```
combined_df.columns
```

```
Index(['Close', 'Open', 'High', 'Low', 'Volume', 'compound', 'neg', 'neu',
       'pos'],
      dtype='object')
```

```
# Splitting the data into train/val/test

# Config
dataNum = 8
timesteps = 10
epochNum = 300

# Train
dataset_train = combined_df[combined_df.index < '2021-06-01']
print(dataset_train.shape)
training_set = dataset_train.iloc[:,1:dataNum+1].values
print(training_set.shape)

# Feature scaling
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
print(training_set_scaled.shape)

#Splitting into inputs/targets
X_train = []
Y_train = []
```

```
for i in range(timesteps, len(training_set_scaled)):
    X_train.append(training_set_scaled[i-timesteps:i, 0:dataNum])
    Y_train.append(training_set_scaled[i, 0])
X_train, Y_train = np.array(X_train), np.array(Y_train)
print(X_train.shape, Y_train.shape)
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], dataNum))
print(X_train.shape, Y_train.shape)

# Validation
val_df = combined_df[(combined_df.index >= '2021-09-01') & (combined_df.index <= '2021-12-31')]
validation_set = val_df.iloc[:, 1:dataNum+1].values

# Feature scaling
validation_set_scaled = sc.transform(validation_set)

#Splitting into inputs/targets
X_val = []
Y_val = []
for i in range(timesteps, len(validation_set_scaled)):
    X_val.append(validation_set_scaled[i-timesteps:i, 0:dataNum])
    Y_val.append(validation_set_scaled[i, 0])
X_val, Y_val = np.array(X_val), np.array(Y_val)
X_val = np.reshape(X_val, (X_val.shape[0], X_val.shape[1], dataNum))

# Test
test_set = combined_df[(combined_df.index >= '2021-06-01') & (combined_df.index < '2021-09-01')]
print(test_set.shape)
real_stock_price = test_set.iloc[:,1:dataNum+1].values
print(real_stock_price.shape)
lenOfReal = len(real_stock_price)
inputs = real_stock_price
inputs = sc.transform(inputs)
print(inputs.shape)

# Combine the last 20 days of training data with the test data
last_20_days_training = training_set_scaled[-timesteps:]
print('last_20_days_training', last_20_days_training.shape)
combined_test_data = np.concatenate((last_20_days_training, inputs))
print('inputs', inputs.shape)
print('combined_test_data', combined_test_data.shape)

inputs_test = []
for i in range(timesteps, len(inputs) + timesteps):
    inputs_test.append(combined_test_data[i-timesteps:i, 0:dataNum])
inputs_test = np.array(inputs_test)
inputs_test = np.reshape(inputs_test, (inputs_test.shape[0], inputs_test.shape[1], dataNum))
print('inputs_test', inputs_test.shape)
```

```
(102, 9)
(102, 8)
(102, 8)
(92, 10, 8) (92,)
(92, 10, 8) (92,)
(65, 9)
(65, 8)
(65, 8)
last_20_days_training (10, 8)
inputs (65, 8)
combined_test_data (75, 8)
inputs_test (65, 10, 8)
```

```
# Define the model
model_combined = Sequential()

model_combined.add(LSTM(units=800, input_shape=(X_train.shape[1], dataNum), return_sequences=True))
model_combined.add(Dropout(0.05))
model_combined.add(LSTM(units=700))

model_combined.add(Dense(units=1))
model_combined.compile(optimizer='adam', loss='mean_squared_error', metrics=['mean_squared_error'])
model_combined.summary()

# Train the model and save the history
history_combined = model_combined.fit(X_train, Y_train, validation_data=(X_val, Y_val), batch_size=20, epochs=epochNum)
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(**kwargs)
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
lstm_5 (LSTM)	(None, 10, 800)	2,588,800
dropout (Dropout)	(None, 10, 800)	0
lstm_6 (LSTM)	(None, 700)	4,202,800
dense_1 (Dense)	(None, 1)	701

Total params: 6,792,301 (25.91 MB)
Trainable params: 6,792,301 (25.91 MB)
Non-trainable params: 0 (0.00 B)

```
Epoch 1/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 4s 117ms/step - loss: 0.2386 - mean_squared_error: 0.2386 - val_loss: 0.1272 - val_mean_squared
Epoch 2/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 36ms/step - loss: 0.0838 - mean_squared_error: 0.0838 - val_loss: 0.0219 - val_mean_squared_
Epoch 3/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0446 - mean_squared_error: 0.0446 - val_loss: 0.0325 - val_mean_squared_
Epoch 4/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0407 - mean_squared_error: 0.0407 - val_loss: 0.0019 - val_mean_squared_
Epoch 5/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0364 - mean_squared_error: 0.0364 - val_loss: 0.0037 - val_mean_squared_
Epoch 6/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0340 - mean_squared_error: 0.0340 - val_loss: 0.0142 - val_mean_squared_
Epoch 7/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0215 - mean_squared_error: 0.0215 - val_loss: 0.0053 - val_mean_squared_
Epoch 8/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0246 - mean_squared_error: 0.0246 - val_loss: 0.0042 - val_mean_squared_
Epoch 9/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0222 - mean_squared_error: 0.0222 - val_loss: 0.0028 - val_mean_squared_
Epoch 10/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0111 - mean_squared_error: 0.0111 - val_loss: 0.0024 - val_mean_squared_
Epoch 11/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0229 - mean_squared_error: 0.0229 - val_loss: 0.0029 - val_mean_squared_
Epoch 12/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0179 - mean_squared_error: 0.0179 - val_loss: 0.0011 - val_mean_squared_
Epoch 13/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0123 - mean_squared_error: 0.0123 - val_loss: 6.7266e-04 - val_mean_squa
Epoch 14/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0114 - mean_squared_error: 0.0114 - val_loss: 8.2595e-04 - val_mean_squa
Epoch 15/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0118 - mean_squared_error: 0.0118 - val_loss: 0.0022 - val_mean_squared_
Epoch 16/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0141 - mean_squared_error: 0.0141 - val_loss: 0.0022 - val_mean_squared_
Epoch 17/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0109 - mean_squared_error: 0.0109 - val_loss: 0.0015 - val_mean_squared_
Epoch 18/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0107 - mean_squared_error: 0.0107 - val_loss: 0.0010 - val_mean_squared_
Epoch 19/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0100 - mean_squared_error: 0.0100 - val_loss: 0.0013 - val_mean_squared_
Epoch 20/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0075 - mean_squared_error: 0.0075 - val_loss: 0.0026 - val_mean_squared_
Epoch 21/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0103 - mean_squared_error: 0.0103 - val_loss: 0.0036 - val_mean_squared_
Epoch 22/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0089 - mean_squared_error: 0.0089 - val_loss: 0.0031 - val_mean_squared_
Epoch 23/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0089 - mean_squared_error: 0.0089 - val_loss: 0.0031 - val_mean_squared_
Epoch 24/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0073 - mean_squared_error: 0.0073 - val_loss: 8.4015e-04 - val_mean_squa
Epoch 25/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0100 - mean_squared_error: 0.0100 - val_loss: 6.9598e-04 - val_mean_squa
Epoch 26/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0085 - mean_squared_error: 0.0085 - val_loss: 6.1800e-04 - val_mean_squa
Epoch 27/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0100 - mean_squared_error: 0.0100 - val_loss: 4.5712e-04 - val_mean_squa
Epoch 28/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0069 - mean_squared_error: 0.0069 - val_loss: 6.8607e-04 - val_mean_squa
Epoch 29/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 0.0093 - mean_squared_error: 0.0093 - val_loss: 7.2777e-04 - val_mean_squa
Epoch 30/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0080 - mean_squared_error: 0.0080 - val_loss: 5.6901e-04 - val_mean_squa
Epoch 31/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0075 - mean_squared_error: 0.0075 - val_loss: 0.0012 - val_mean_squared_
Epoch 32/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 0.0055 - mean_squared_error: 0.0055 - val_loss: 5.3331e-04 - val_mean_squa
Epoch 33/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0093 - mean_squared_error: 0.0093 - val_loss: 6.2148e-04 - val_mean_squa
Epoch 34/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 0.0054 - mean_squared_error: 0.0054 - val_loss: 0.0018 - val_mean_squared_
Epoch 35/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0079 - mean_squared_error: 0.0079 - val_loss: 0.0040 - val_mean_squared_
Epoch 36/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 0.0082 - mean_squared_error: 0.0082 - val_loss: 0.0018 - val_mean_squared_
Epoch 37/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - loss: 0.0085 - mean_squared_error: 0.0085 - val_loss: 0.0017 - val_mean_squared_
```

```
Epoch 38/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 0.0062 - mean_squared_error: 0.0062 - val_loss: 0.0013 - val_mean_squared_
Epoch 39/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 0.0070 - mean_squared_error: 0.0070 - val_loss: 0.0015 - val_mean_squared_
Epoch 40/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0060 - mean_squared_error: 0.0060 - val_loss: 8.3671e-04 - val_mean_squa
Epoch 41/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0073 - mean_squared_error: 0.0073 - val_loss: 5.1562e-04 - val_mean_squa
Epoch 42/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0068 - mean_squared_error: 0.0068 - val_loss: 5.6429e-04 - val_mean_squa
Epoch 43/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0079 - mean_squared_error: 0.0079 - val_loss: 5.0292e-04 - val_mean_squa
Epoch 44/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0074 - mean_squared_error: 0.0074 - val_loss: 7.7451e-04 - val_mean_squa
Epoch 45/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0056 - mean_squared_error: 0.0056 - val_loss: 6.1023e-04 - val_mean_squa
Epoch 46/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0045 - mean_squared_error: 0.0045 - val_loss: 5.9031e-04 - val_mean_squa
Epoch 47/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0049 - mean_squared_error: 0.0049 - val_loss: 4.2361e-04 - val_mean_squa
Epoch 48/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0064 - mean_squared_error: 0.0064 - val_loss: 7.6457e-04 - val_mean_squa
Epoch 49/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0052 - mean_squared_error: 0.0052 - val_loss: 0.0012 - val_mean_squared_
Epoch 50/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0081 - mean_squared_error: 0.0081 - val_loss: 0.0013 - val_mean_squared_
Epoch 51/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0065 - mean_squared_error: 0.0065 - val_loss: 8.7021e-04 - val_mean_squa
Epoch 52/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0054 - mean_squared_error: 0.0054 - val_loss: 4.5850e-04 - val_mean_squa
Epoch 53/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0057 - mean_squared_error: 0.0057 - val_loss: 5.5022e-04 - val_mean_squa
Epoch 54/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0071 - mean_squared_error: 0.0071 - val_loss: 5.9832e-04 - val_mean_squa
Epoch 55/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0062 - mean_squared_error: 0.0062 - val_loss: 6.9814e-04 - val_mean_squa
Epoch 56/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0050 - mean_squared_error: 0.0050 - val_loss: 0.0019 - val_mean_squared_
Epoch 57/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0053 - mean_squared_error: 0.0053 - val_loss: 0.0021 - val_mean_squared_
Epoch 58/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0082 - mean_squared_error: 0.0082 - val_loss: 5.0561e-04 - val_mean_squa
Epoch 59/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0047 - mean_squared_error: 0.0047 - val_loss: 4.9253e-04 - val_mean_squa
Epoch 60/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0053 - mean_squared_error: 0.0053 - val_loss: 6.0286e-04 - val_mean_squa
Epoch 61/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0048 - mean_squared_error: 0.0048 - val_loss: 0.0023 - val_mean_squared_
Epoch 62/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0064 - mean_squared_error: 0.0064 - val_loss: 0.0014 - val_mean_squared_
Epoch 63/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0083 - mean_squared_error: 0.0083 - val_loss: 4.2748e-04 - val_mean_squa
Epoch 64/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0050 - mean_squared_error: 0.0050 - val_loss: 0.0015 - val_mean_squared_
Epoch 65/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0065 - mean_squared_error: 0.0065 - val_loss: 0.0014 - val_mean_squared_
Epoch 66/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0046 - mean_squared_error: 0.0046 - val_loss: 4.8931e-04 - val_mean_squa
Epoch 67/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0037 - mean_squared_error: 0.0037 - val_loss: 0.0010 - val_mean_squared_
Epoch 68/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0036 - mean_squared_error: 0.0036 - val_loss: 0.0013 - val_mean_squared_
Epoch 69/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0055 - mean_squared_error: 0.0055 - val_loss: 4.7840e-04 - val_mean_squa
Epoch 70/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - loss: 0.0047 - mean_squared_error: 0.0047 - val_loss: 6.9664e-04 - val_mean_squa
Epoch 71/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0042 - mean_squared_error: 0.0042 - val_loss: 8.5681e-04 - val_mean_squa
Epoch 72/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0065 - mean_squared_error: 0.0065 - val_loss: 8.2256e-04 - val_mean_squa
Epoch 73/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0038 - mean_squared_error: 0.0038 - val_loss: 6.7764e-04 - val_mean_squa
Epoch 74/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0045 - mean_squared_error: 0.0045 - val_loss: 5.0971e-04 - val_mean_squa
Epoch 75/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 0.0047 - mean_squared_error: 0.0047 - val_loss: 5.0568e-04 - val_mean_squa
Epoch 76/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0058 - mean_squared_error: 0.0058 - val_loss: 7.5465e-04 - val_mean_squa
Epoch 77/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - loss: 0.0037 - mean_squared_error: 0.0037 - val_loss: 4.7907e-04 - val_mean_squa
Epoch 78/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 0.0039 - mean_squared_error: 0.0039 - val_loss: 8.4831e-04 - val_mean_squa
Epoch 79/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 0.0040 - mean_squared_error: 0.0040 - val_loss: 0.0010 - val_mean_squared_
Epoch 80/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - loss: 0.0036 - mean_squared_error: 0.0036 - val_loss: 4.8394e-04 - val_mean_squa
Epoch 81/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 0.0044 - mean_squared_error: 0.0044 - val_loss: 0.0010 - val_mean_squared_
Epoch 82/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0053 - mean_squared_error: 0.0053 - val_loss: 0.0025 - val_mean_squared_
Epoch 83/300
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0054 - mean_squared_error: 0.0054 - val_loss: 0.0011 - val_mean_squared_
Epoch 84/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0053 - mean_squared_error: 0.0053 - val_loss: 6.1922e-04 - val_mean_squa
Epoch 85/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0062 - mean_squared_error: 0.0062 - val_loss: 0.0011 - val_mean_squared_
Epoch 86/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 0.0049 - mean_squared_error: 0.0049 - val_loss: 0.0011 - val_mean_squared_
Epoch 87/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0041 - mean_squared_error: 0.0041 - val_loss: 0.0011 - val_mean_squared_
Epoch 88/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 0.0046 - mean_squared_error: 0.0046 - val_loss: 4.6100e-04 - val_mean_squa
Epoch 89/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 0.0039 - mean_squared_error: 0.0039 - val_loss: 4.9307e-04 - val_mean_squa
Epoch 90/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 0.0032 - mean_squared_error: 0.0032 - val_loss: 6.9717e-04 - val_mean_squa
Epoch 91/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0030 - mean_squared_error: 0.0030 - val_loss: 7.9356e-04 - val_mean_squa
Epoch 92/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 0.0045 - mean_squared_error: 0.0045 - val_loss: 0.0017 - val_mean_squared_
Epoch 93/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0042 - mean_squared_error: 0.0042 - val_loss: 0.0010 - val_mean_squared_
Epoch 94/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0067 - mean_squared_error: 0.0067 - val_loss: 4.9684e-04 - val_mean_squa
Epoch 95/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0039 - mean_squared_error: 0.0039 - val_loss: 0.0011 - val_mean_squared_
Epoch 96/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0052 - mean_squared_error: 0.0052 - val_loss: 7.2541e-04 - val_mean_squa
Epoch 97/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0036 - mean_squared_error: 0.0036 - val_loss: 9.8922e-04 - val_mean_squa
Epoch 98/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0026 - mean_squared_error: 0.0026 - val_loss: 4.5026e-04 - val_mean_squa
Epoch 99/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0038 - mean_squared_error: 0.0038 - val_loss: 5.0901e-04 - val_mean_squa
Epoch 100/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0036 - mean_squared_error: 0.0036 - val_loss: 9.3046e-04 - val_mean_squa
Epoch 101/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0032 - mean_squared_error: 0.0032 - val_loss: 0.0012 - val_mean_squared_
Epoch 102/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0038 - mean_squared_error: 0.0038 - val_loss: 4.9882e-04 - val_mean_squa
Epoch 103/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0034 - mean_squared_error: 0.0034 - val_loss: 4.8040e-04 - val_mean_squa
Epoch 104/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0033 - mean_squared_error: 0.0033 - val_loss: 8.0506e-04 - val_mean_squa
Epoch 105/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0028 - mean_squared_error: 0.0028 - val_loss: 6.8462e-04 - val_mean_squa
Epoch 106/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0021 - mean_squared_error: 0.0021 - val_loss: 4.4697e-04 - val_mean_squa
Epoch 107/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0029 - mean_squared_error: 0.0029 - val_loss: 0.0028 - val_mean_squared_
Epoch 108/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0038 - mean_squared_error: 0.0038 - val_loss: 0.0013 - val_mean_squared_
Epoch 109/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0034 - mean_squared_error: 0.0034 - val_loss: 8.3157e-04 - val_mean_squa
Epoch 110/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0035 - mean_squared_error: 0.0035 - val_loss: 5.0996e-04 - val_mean_squa
Epoch 111/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0030 - mean_squared_error: 0.0030 - val_loss: 0.0011 - val_mean_squared_
Epoch 112/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0058 - mean_squared_error: 0.0058 - val_loss: 7.5363e-04 - val_mean_squa
Epoch 113/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0029 - mean_squared_error: 0.0029 - val_loss: 4.7029e-04 - val_mean_squa
Epoch 114/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0032 - mean_squared_error: 0.0032 - val_loss: 7.1604e-04 - val_mean_squa
Epoch 115/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0022 - mean_squared_error: 0.0022 - val_loss: 5.4650e-04 - val_mean_squa
Epoch 116/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0032 - mean_squared_error: 0.0032 - val_loss: 4.3877e-04 - val_mean_squa
Epoch 117/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0040 - mean_squared_error: 0.0040 - val_loss: 0.0016 - val_mean_squared_
Epoch 118/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0044 - mean_squared_error: 0.0044 - val_loss: 4.2978e-04 - val_mean_squa
Epoch 119/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0028 - mean_squared_error: 0.0028 - val_loss: 4.3515e-04 - val_mean_squa
Epoch 120/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0033 - mean_squared_error: 0.0033 - val_loss: 5.1438e-04 - val_mean_squa
Epoch 121/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0027 - mean_squared_error: 0.0027 - val_loss: 8.1355e-04 - val_mean_squa
Epoch 122/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0027 - mean_squared_error: 0.0027 - val_loss: 5.5815e-04 - val_mean_squa
Epoch 123/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0024 - mean_squared_error: 0.0024 - val_loss: 8.2906e-04 - val_mean_squa
Epoch 124/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0030 - mean_squared_error: 0.0030 - val_loss: 4.1520e-04 - val_mean_squa
Epoch 125/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0025 - mean_squared_error: 0.0025 - val_loss: 4.5835e-04 - val_mean_squa
Epoch 126/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0025 - mean_squared_error: 0.0025 - val_loss: 5.9623e-04 - val_mean_squa
Epoch 127/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0016 - mean_squared_error: 0.0016 - val_loss: 5.7907e-04 - val_mean_squa
Epoch 128/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0021 - mean_squared_error: 0.0021 - val_loss: 4.1660e-04 - val_mean_squa
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0021 - mean_squared_error: 0.0021 - val_loss: 4.1660e-04 - val_mean_squa
Epoch 129/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0024 - mean_squared_error: 0.0024 - val_loss: 4.3292e-04 - val_mean_squa
Epoch 130/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0027 - mean_squared_error: 0.0027 - val_loss: 5.6497e-04 - val_mean_squa
Epoch 131/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - loss: 0.0023 - mean_squared_error: 0.0023 - val_loss: 4.7081e-04 - val_mean_squ
Epoch 132/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 157ms/step - loss: 0.0022 - mean_squared_error: 0.0022 - val_loss: 7.8056e-04 - val_mean_squ
Epoch 133/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 131ms/step - loss: 0.0024 - mean_squared_error: 0.0024 - val_loss: 4.7616e-04 - val_mean_squ
Epoch 134/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 48ms/step - loss: 0.0035 - mean_squared_error: 0.0035 - val_loss: 4.8290e-04 - val_mean_squa
Epoch 135/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0015 - mean_squared_error: 0.0015 - val_loss: 4.1593e-04 - val_mean_squa
Epoch 136/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 0.0021 - mean_squared_error: 0.0021 - val_loss: 8.0355e-04 - val_mean_squa
Epoch 137/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0021 - mean_squared_error: 0.0021 - val_loss: 0.0010 - val_mean_squared_
Epoch 138/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0021 - mean_squared_error: 0.0021 - val_loss: 4.9952e-04 - val_mean_squa
Epoch 139/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0015 - mean_squared_error: 0.0015 - val_loss: 4.4189e-04 - val_mean_squa
Epoch 140/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0020 - mean_squared_error: 0.0020 - val_loss: 3.8891e-04 - val_mean_squa
Epoch 141/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0019 - mean_squared_error: 0.0019 - val_loss: 4.3376e-04 - val_mean_squa
Epoch 142/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0018 - mean_squared_error: 0.0018 - val_loss: 5.8863e-04 - val_mean_squa
Epoch 143/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0020 - mean_squared_error: 0.0020 - val_loss: 4.8634e-04 - val_mean_squa
Epoch 144/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0026 - mean_squared_error: 0.0026 - val_loss: 5.6144e-04 - val_mean_squa
Epoch 145/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0018 - mean_squared_error: 0.0018 - val_loss: 4.0832e-04 - val_mean_squa
Epoch 146/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0028 - mean_squared_error: 0.0028 - val_loss: 4.0611e-04 - val_mean_squa
Epoch 147/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0031 - mean_squared_error: 0.0031 - val_loss: 3.9573e-04 - val_mean_squa
Epoch 148/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0019 - mean_squared_error: 0.0019 - val_loss: 3.9462e-04 - val_mean_squa
Epoch 149/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0015 - mean_squared_error: 0.0015 - val_loss: 3.9646e-04 - val_mean_squa
Epoch 150/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 0.0018 - mean_squared_error: 0.0018 - val_loss: 8.0367e-04 - val_mean_squa
Epoch 151/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0024 - mean_squared_error: 0.0024 - val_loss: 5.3822e-04 - val_mean_squa
Epoch 152/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0028 - mean_squared_error: 0.0028 - val_loss: 0.0015 - val_mean_squared_
Epoch 153/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0031 - mean_squared_error: 0.0031 - val_loss: 0.0012 - val_mean_squared_
Epoch 154/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0031 - mean_squared_error: 0.0031 - val_loss: 4.3374e-04 - val_mean_squa
Epoch 155/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0027 - mean_squared_error: 0.0027 - val_loss: 9.0177e-04 - val_mean_squa
Epoch 156/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0033 - mean_squared_error: 0.0033 - val_loss: 4.4959e-04 - val_mean_squa
Epoch 157/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0022 - mean_squared_error: 0.0022 - val_loss: 3.8985e-04 - val_mean_squa
Epoch 158/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - loss: 0.0029 - mean_squared_error: 0.0029 - val_loss: 0.0013 - val_mean_squared_
Epoch 159/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0025 - mean_squared_error: 0.0025 - val_loss: 4.9169e-04 - val_mean_squa
Epoch 160/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0018 - mean_squared_error: 0.0018 - val_loss: 3.9542e-04 - val_mean_squa
Epoch 161/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - loss: 0.0034 - mean_squared_error: 0.0034 - val_loss: 3.6088e-04 - val_mean_squa
Epoch 162/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0016 - mean_squared_error: 0.0016 - val_loss: 5.3012e-04 - val_mean_squa
Epoch 163/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0027 - mean_squared_error: 0.0027 - val_loss: 3.5058e-04 - val_mean_squa
Epoch 164/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - loss: 0.0017 - mean_squared_error: 0.0017 - val_loss: 3.4333e-04 - val_mean_squa
Epoch 165/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0026 - mean_squared_error: 0.0026 - val_loss: 3.6601e-04 - val_mean_squa
Epoch 166/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0020 - mean_squared_error: 0.0020 - val_loss: 3.7572e-04 - val_mean_squa
Epoch 167/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0022 - mean_squared_error: 0.0022 - val_loss: 6.0299e-04 - val_mean_squa
Epoch 168/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0017 - mean_squared_error: 0.0017 - val_loss: 7.3487e-04 - val_mean_squa
Epoch 169/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 87ms/step - loss: 0.0016 - mean_squared_error: 0.0016 - val_loss: 7.3028e-04 - val_mean_squa
Epoch 170/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - loss: 0.0025 - mean_squared_error: 0.0025 - val_loss: 4.5062e-04 - val_mean_squa
Epoch 171/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0015 - mean_squared_error: 0.0015 - val_loss: 6.2679e-04 - val_mean_squa
Epoch 172/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 88ms/step - loss: 0.0020 - mean_squared_error: 0.0020 - val_loss: 6.8942e-04 - val_mean_squa
Epoch 173/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 93ms/step - loss: 0.0022 - mean_squared_error: 0.0022 - val_loss: 4.3034e-04 - val_mean_squa
```

```
Epoch 174/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - loss: 0.0032 - mean_squared_error: 0.0032 - val_loss: 3.4796e-04 - val_mean_squa
Epoch 175/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0022 - mean_squared_error: 0.0022 - val_loss: 4.7924e-04 - val_mean_squa
Epoch 176/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 0.0027 - mean_squared_error: 0.0027 - val_loss: 3.4874e-04 - val_mean_squa
Epoch 177/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - loss: 0.0020 - mean_squared_error: 0.0020 - val_loss: 0.0018 - val_mean_squared_
Epoch 178/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0037 - mean_squared_error: 0.0037 - val_loss: 7.1292e-04 - val_mean_squa
Epoch 179/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0026 - mean_squared_error: 0.0026 - val_loss: 3.7989e-04 - val_mean_squa
Epoch 180/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0017 - mean_squared_error: 0.0017 - val_loss: 3.6442e-04 - val_mean_squa
Epoch 181/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 0.0017 - mean_squared_error: 0.0017 - val_loss: 4.0721e-04 - val_mean_squa
Epoch 182/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 0.0025 - mean_squared_error: 0.0025 - val_loss: 8.8482e-04 - val_mean_squa
Epoch 183/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - loss: 0.0022 - mean_squared_error: 0.0022 - val_loss: 7.8318e-04 - val_mean_squa
Epoch 184/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - loss: 0.0020 - mean_squared_error: 0.0020 - val_loss: 4.8324e-04 - val_mean_squa
Epoch 185/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0015 - mean_squared_error: 0.0015 - val_loss: 5.7434e-04 - val_mean_squa
Epoch 186/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0015 - mean_squared_error: 0.0015 - val_loss: 5.7807e-04 - val_mean_squa
Epoch 187/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 3.3643e-04 - val_mean_squa
Epoch 188/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 3.3160e-04 - val_mean_squa
Epoch 189/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 4.5547e-04 - val_mean_squa
Epoch 190/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 3.4724e-04 - val_mean_squa
Epoch 191/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 6.6898e-04 - val_mean_squa
Epoch 192/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 4.1463e-04 - val_mean_squa
Epoch 193/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 4.8118e-04 - val_mean_squa
Epoch 194/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0014 - mean_squared_error: 0.0014 - val_loss: 5.7190e-04 - val_mean_squa
Epoch 195/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0015 - mean_squared_error: 0.0015 - val_loss: 0.0010 - val_mean_squared_
Epoch 196/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 3.3658e-04 - val_mean_squa
Epoch 197/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0014 - mean_squared_error: 0.0014 - val_loss: 3.0069e-04 - val_mean_squa
Epoch 198/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 4.8694e-04 - val_mean_squa
Epoch 199/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 3.0345e-04 - val_mean_squa
Epoch 200/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 8.3762e-04 - val_mean_squa
Epoch 201/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0017 - mean_squared_error: 0.0017 - val_loss: 3.5280e-04 - val_mean_squa
Epoch 202/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0021 - mean_squared_error: 0.0021 - val_loss: 2.9250e-04 - val_mean_squa
Epoch 203/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 3.4962e-04 - val_mean_squa
Epoch 204/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 3.1782e-04 - val_mean_squa
Epoch 205/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 2.8484e-04 - val_mean_squa
Epoch 206/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 5.1173e-04 - val_mean_squa
Epoch 207/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 0.0012 - val_mean_squared_
Epoch 208/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0019 - mean_squared_error: 0.0019 - val_loss: 7.5070e-04 - val_mean_squa
Epoch 209/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0018 - mean_squared_error: 0.0018 - val_loss: 5.9354e-04 - val_mean_squa
Epoch 210/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 6.9121e-04 - val_mean_squa
Epoch 211/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0017 - mean_squared_error: 0.0017 - val_loss: 5.5977e-04 - val_mean_squa
Epoch 212/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 7.7862e-04 - val_mean_squa
Epoch 213/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0018 - mean_squared_error: 0.0018 - val_loss: 4.6165e-04 - val_mean_squa
Epoch 214/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0014 - mean_squared_error: 0.0014 - val_loss: 2.7127e-04 - val_mean_squa
Epoch 215/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 2.9073e-04 - val_mean_squa
Epoch 216/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 3.3714e-04 - val_mean_squa
Epoch 217/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 2.9000e-04 - val_mean_squa
Epoch 218/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 2.7512e-04 - val_mean_squa
Epoch 219/300
```

Epoch 219/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 9.4047e-04 - mean_squared_error: 9.4047e-04 - val_loss: 2.7826e-04 - val_m
Epoch 220/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 9.3563e-04 - mean_squared_error: 9.3563e-04 - val_loss: 3.0035e-04 - val_m
Epoch 221/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 9.1582e-04 - mean_squared_error: 9.1582e-04 - val_loss: 3.7033e-04 - val_m
Epoch 222/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0016 - mean_squared_error: 0.0016 - val_loss: 3.5261e-04 - val_mean_squa
Epoch 223/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 7.0331e-04 - val_mean_squa
Epoch 224/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 0.0016 - mean_squared_error: 0.0016 - val_loss: 5.4007e-04 - val_mean_squa
Epoch 225/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 2.6395e-04 - val_mean_squa
Epoch 226/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 2.9848e-04 - val_mean_squa
Epoch 227/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 4.5983e-04 - val_mean_squa
Epoch 228/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 5.8660e-04 - val_mean_squa
Epoch 229/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 9.2949e-04 - mean_squared_error: 9.2949e-04 - val_loss: 7.1001e-04 - val_m
Epoch 230/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 0.0014 - mean_squared_error: 0.0014 - val_loss: 2.9670e-04 - val_mean_squa
Epoch 231/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 2.6021e-04 - val_mean_squa
Epoch 232/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 2.8985e-04 - val_mean_squa
Epoch 233/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - loss: 9.2446e-04 - mean_squared_error: 9.2446e-04 - val_loss: 2.5847e-04 - val_m
Epoch 234/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 2.7656e-04 - val_mean_squa
Epoch 235/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 3.4828e-04 - val_mean_squa
Epoch 236/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 2.7440e-04 - val_mean_squa
Epoch 237/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 9.9085e-04 - mean_squared_error: 9.9085e-04 - val_loss: 2.6265e-04 - val_m
Epoch 238/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 9.2742e-04 - mean_squared_error: 9.2742e-04 - val_loss: 2.8948e-04 - val_m
Epoch 239/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 8.6301e-04 - mean_squared_error: 8.6301e-04 - val_loss: 3.7442e-04 - val_m
Epoch 240/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 5.1192e-04 - val_mean_squa
Epoch 241/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 2.4033e-04 - val_mean_squa
Epoch 242/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 7.8779e-04 - mean_squared_error: 7.8779e-04 - val_loss: 3.0176e-04 - val_m
Epoch 243/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 2.9501e-04 - val_mean_squa
Epoch 244/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 7.9525e-04 - mean_squared_error: 7.9525e-04 - val_loss: 3.2394e-04 - val_m
Epoch 245/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0010 - mean_squared_error: 0.0010 - val_loss: 0.0012 - val_mean_squared_
Epoch 246/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 7.1841e-04 - val_mean_squa
Epoch 247/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0016 - mean_squared_error: 0.0016 - val_loss: 7.3474e-04 - val_mean_squa
Epoch 248/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 3.1836e-04 - val_mean_squa
Epoch 249/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 2.6069e-04 - val_mean_squa
Epoch 250/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 9.4618e-04 - mean_squared_error: 9.4618e-04 - val_loss: 2.7523e-04 - val_m
Epoch 251/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 9.8281e-04 - mean_squared_error: 9.8281e-04 - val_loss: 2.3795e-04 - val_m
Epoch 252/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 3.0061e-04 - val_mean_squa
Epoch 253/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 4.6247e-04 - val_mean_squa
Epoch 254/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 9.9541e-04 - mean_squared_error: 9.9541e-04 - val_loss: 2.4052e-04 - val_m
Epoch 255/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 9.5614e-04 - mean_squared_error: 9.5614e-04 - val_loss: 7.6070e-04 - val_m
Epoch 256/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 9.9740e-04 - mean_squared_error: 9.9740e-04 - val_loss: 4.2687e-04 - val_m
Epoch 257/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 8.6651e-04 - mean_squared_error: 8.6651e-04 - val_loss: 7.5181e-04 - val_m
Epoch 258/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 6.1987e-04 - val_mean_squa
Epoch 259/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 2.3829e-04 - val_mean_squa
Epoch 260/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 6.6666e-04 - val_mean_squa
Epoch 261/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 4.9983e-04 - val_mean_squa
Epoch 262/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0017 - mean_squared_error: 0.0017 - val_loss: 4.5956e-04 - val_mean_squa
Epoch 263/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 8.0420e-04 - mean_squared_error: 8.0420e-04 - val_loss: 4.2195e-04 - val_m
Epoch 264/300

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 9.7738e-04 - mean_squared_error: 9.7738e-04 - val_loss: 2.2166e-04 - val_m
Epoch 265/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 7.5250e-04 - mean_squared_error: 7.5250e-04 - val_loss: 2.1844e-04 - val_m
Epoch 266/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0012 - mean_squared_error: 0.0012 - val_loss: 3.6701e-04 - val_mean_squa
Epoch 267/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 0.0010 - mean_squared_error: 0.0010 - val_loss: 2.3733e-04 - val_mean_squa
Epoch 268/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 7.7880e-04 - mean_squared_error: 7.7880e-04 - val_loss: 2.2579e-04 - val_m
Epoch 269/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 5.8556e-04 - val_mean_squa
Epoch 270/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 9.6738e-04 - mean_squared_error: 9.6738e-04 - val_loss: 5.6953e-04 - val_m
Epoch 271/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 8.0666e-04 - mean_squared_error: 8.0666e-04 - val_loss: 5.3235e-04 - val_m
Epoch 272/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 2.4169e-04 - val_mean_squa
Epoch 273/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 9.0943e-04 - mean_squared_error: 9.0943e-04 - val_loss: 2.2402e-04 - val_m
Epoch 274/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 9.6797e-04 - mean_squared_error: 9.6797e-04 - val_loss: 3.7424e-04 - val_m
Epoch 275/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 6.1093e-04 - mean_squared_error: 6.1093e-04 - val_loss: 4.9382e-04 - val_m
Epoch 276/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - loss: 8.2263e-04 - mean_squared_error: 8.2263e-04 - val_loss: 4.4256e-04 - val_m
Epoch 277/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 9.2043e-04 - mean_squared_error: 9.2043e-04 - val_loss: 2.4733e-04 - val_m
Epoch 278/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 4.8757e-04 - mean_squared_error: 4.8757e-04 - val_loss: 2.1752e-04 - val_m
Epoch 279/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 6.7678e-04 - mean_squared_error: 6.7678e-04 - val_loss: 2.4112e-04 - val_m
Epoch 280/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - loss: 7.1439e-04 - mean_squared_error: 7.1439e-04 - val_loss: 2.7491e-04 - val_m
Epoch 281/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 5.8067e-04 - mean_squared_error: 5.8067e-04 - val_loss: 2.9784e-04 - val_m
Epoch 282/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 5.7876e-04 - mean_squared_error: 5.7876e-04 - val_loss: 2.3505e-04 - val_m
Epoch 283/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - loss: 8.6251e-04 - mean_squared_error: 8.6251e-04 - val_loss: 2.3108e-04 - val_m
Epoch 284/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - loss: 5.0669e-04 - mean_squared_error: 5.0669e-04 - val_loss: 3.0309e-04 - val_m
Epoch 285/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 9.7881e-04 - mean_squared_error: 9.7881e-04 - val_loss: 2.7511e-04 - val_m
Epoch 286/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step - loss: 0.0011 - mean_squared_error: 0.0011 - val_loss: 2.2170e-04 - val_mean_squa
Epoch 287/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 7.1005e-04 - mean_squared_error: 7.1005e-04 - val_loss: 2.3258e-04 - val_m
Epoch 288/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 9.0627e-04 - mean_squared_error: 9.0627e-04 - val_loss: 3.8797e-04 - val_m
Epoch 289/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 9.1291e-04 - mean_squared_error: 9.1291e-04 - val_loss: 2.3486e-04 - val_m
Epoch 290/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 6.5909e-04 - mean_squared_error: 6.5909e-04 - val_loss: 6.9082e-04 - val_m
Epoch 291/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 0.0010 - mean_squared_error: 0.0010 - val_loss: 6.2376e-04 - val_mean_squa
Epoch 292/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step - loss: 7.5154e-04 - mean_squared_error: 7.5154e-04 - val_loss: 2.3570e-04 - val_m
Epoch 293/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 8.6920e-04 - mean_squared_error: 8.6920e-04 - val_loss: 4.3131e-04 - val_m
Epoch 294/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.0013 - mean_squared_error: 0.0013 - val_loss: 2.3717e-04 - val_mean_squa
Epoch 295/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 0.0014 - mean_squared_error: 0.0014 - val_loss: 2.2775e-04 - val_mean_squa
Epoch 296/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step - loss: 8.9058e-04 - mean_squared_error: 8.9058e-04 - val_loss: 2.3862e-04 - val_m
Epoch 297/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 7.8110e-04 - mean_squared_error: 7.8110e-04 - val_loss: 2.1780e-04 - val_m
Epoch 298/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 5.3338e-04 - mean_squared_error: 5.3338e-04 - val_loss: 2.6529e-04 - val_m
Epoch 299/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step - loss: 6.5596e-04 - mean_squared_error: 6.5596e-04 - val_loss: 3.2890e-04 - val_m
Epoch 300/300
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step - loss: 5.7512e-04 - mean_squared_error: 5.7512e-04 - val_loss: 5.1558e-04 - val_m
```