```
from google.colab import files
uploaded = files.upload()
```

Choose files DJIA indices data.csv

- **DJIA indices data.csv**(text/csv) - 212038 bytes, last modified: 10/05/2025 - 100% done

```
Saving DJIA indices data.csv to DJIA indices data.csv
```

```
import pandas as pd

df = pd.read_csv("DJIA indices data.csv")
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)

# Show basic info
print("Shape:", df.shape)
print("Columns:", df.columns.tolist())
df.head()
```

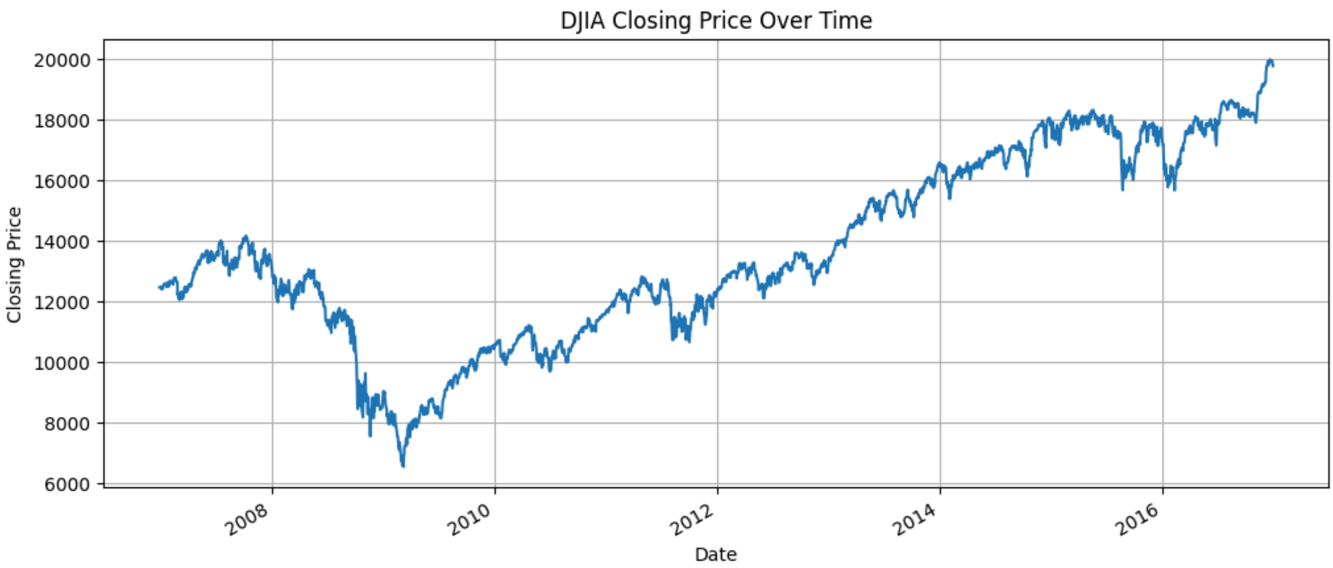
```
Shape: (2519, 6)
Columns: ['Open', 'High', 'Low', 'Close', 'Volume', 'Adj Close']
```

	Open	High	Low	Close	Volume	Adj Close
Date						
2016-12-30	19833.169922	19852.550781	19718.669922	19762.599609	271910000	19762.599609
2016-12-29	19835.460938	19878.439453	19788.939453	19819.779297	172040000	19819.779297
2016-12-28	19964.310547	19981.109375	19827.310547	19833.679688	188350000	19833.679688
2016-12-27	19943.460938	19980.240234	19939.800781	19945.039062	158540000	19945.039062
2016-12-23	19908.609375	19934.150391	19899.060547	19933.810547	158260000	19933.810547

Next steps: Generate code with df View recommended plots New interactive sheet

```
import matplotlib.pyplot as plt

df['Close'].plot(figsize=(12, 5), title="DJIA Closing Price Over Time")
plt.xlabel("Date")
plt.ylabel("Closing Price")
plt.grid()
plt.show()
```



```
from sklearn.preprocessing import MinMaxScaler
import numpy as np

data = df[['Close']].values
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(data)

# Create sequences
def create_sequences(data, seq_length=60):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i:i+seq_length])
        y.append(data[i+seq_length])
    return np.array(X), np.array(y)

sequence_length = 60
X, y = create_sequences(scaled_data, sequence_length)

# Train/test split
split = int(len(X) * 0.8)
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping

X_train = X_train.reshape(-1, sequence_length, 1)
X_test = X_test.reshape(-1, sequence_length, 1)

model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(X_train.shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(50, return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(1))

model.compile(optimizer='adam', loss='mean_squared_error')
history = model.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.1, callbacks=[EarlyStopping(patience=10)], verbose=
```

```
Epoch 1/100
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200: UserWarning: Do not pass an `input_shape`/`input_dim` ar
  super().__init__(**kwargs)
56/56 ━━━━━━━━━━━━━━━━━━━━ 8s 81ms/step - loss: 0.1171 - val_loss: 0.0060
Epoch 2/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 3s 52ms/step - loss: 0.0045 - val_loss: 9.2841e-04
Epoch 3/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 3s 51ms/step - loss: 0.0045 - val_loss: 9.2320e-04
Epoch 4/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 9s 116ms/step - loss: 0.0035 - val_loss: 8.4996e-04
Epoch 5/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 4s 73ms/step - loss: 0.0033 - val_loss: 6.7072e-04
Epoch 6/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 4s 55ms/step - loss: 0.0034 - val_loss: 6.0075e-04
Epoch 7/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 5s 82ms/step - loss: 0.0027 - val_loss: 6.8141e-04
Epoch 8/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 3s 52ms/step - loss: 0.0030 - val_loss: 6.8087e-04
Epoch 9/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 5s 51ms/step - loss: 0.0034 - val_loss: 6.1279e-04
Epoch 10/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 5s 51ms/step - loss: 0.0030 - val_loss: 4.5406e-04
Epoch 11/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 3s 51ms/step - loss: 0.0031 - val_loss: 6.0384e-04
Epoch 12/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 6s 67ms/step - loss: 0.0028 - val_loss: 5.3742e-04
Epoch 13/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 3s 56ms/step - loss: 0.0028 - val_loss: 4.2488e-04
Epoch 14/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 5s 51ms/step - loss: 0.0029 - val_loss: 4.3897e-04
Epoch 15/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 3s 55ms/step - loss: 0.0024 - val_loss: 9.1103e-04
Epoch 16/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 4s 69ms/step - loss: 0.0028 - val_loss: 4.6551e-04
Epoch 17/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 4s 51ms/step - loss: 0.0025 - val_loss: 4.3385e-04
Epoch 18/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 6s 60ms/step - loss: 0.0028 - val_loss: 4.5353e-04
Epoch 19/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 4s 64ms/step - loss: 0.0024 - val_loss: 4.2402e-04
Epoch 20/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 5s 58ms/step - loss: 0.0024 - val_loss: 4.3398e-04
Epoch 21/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 3s 55ms/step - loss: 0.0024 - val_loss: 4.4693e-04
Epoch 22/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 5s 51ms/step - loss: 0.0021 - val_loss: 4.1109e-04
Epoch 23/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 5s 51ms/step - loss: 0.0023 - val_loss: 4.2892e-04
Epoch 24/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 6s 68ms/step - loss: 0.0024 - val_loss: 4.0759e-04
Epoch 25/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 4s 51ms/step - loss: 0.0023 - val_loss: 5.3406e-04
Epoch 26/100
```

```
56/56 ━━━━━━━━━━━━━━━━━━━━ 3s 50ms/step - loss: 0.0022 - val_loss: 4.2437e-04
Epoch 27/100
56/56 ━━━━━━━━━━━━━━━━━━━━ 4s 67ms/step - loss: 0.0018 - val_loss: 4.6963e-04
Epoch 28/100
```

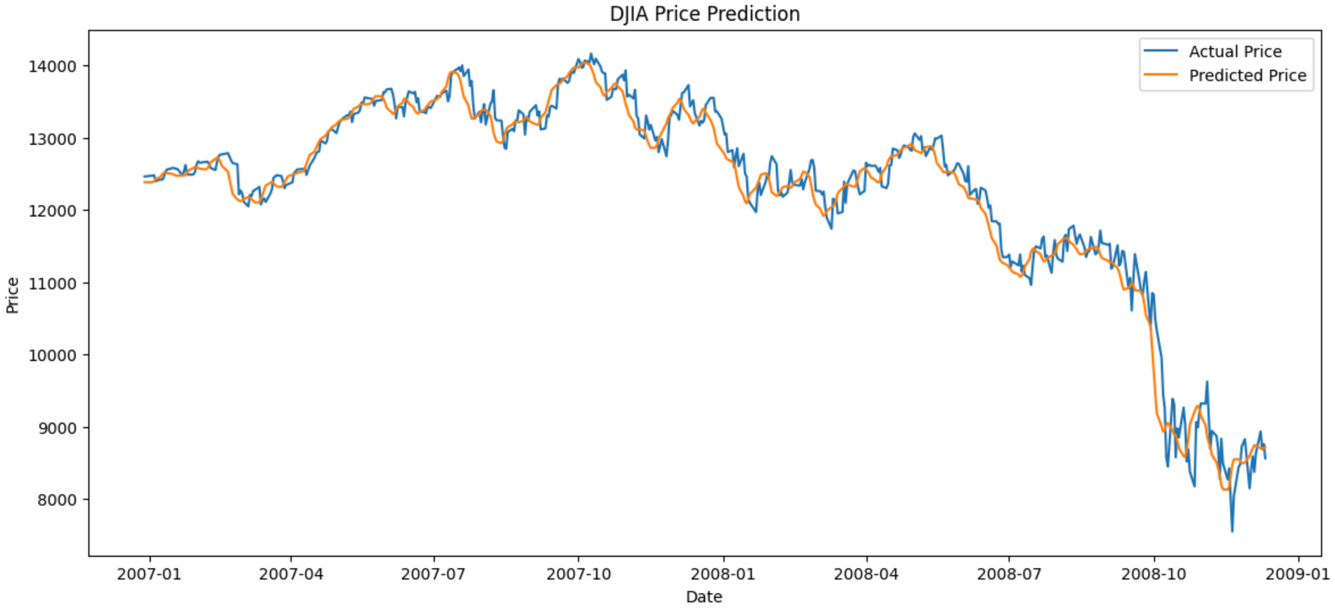
```
from sklearn.metrics import mean_squared_error

y_pred = model.predict(X_test)
y_pred_inv = scaler.inverse_transform(y_pred)
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))

plt.figure(figsize=(14,6))
plt.plot(df.index[-len(y_test):], y_test_inv, label='Actual Price')
plt.plot(df.index[-len(y_test):], y_pred_inv, label='Predicted Price')
plt.title("DJIA Price Prediction")
plt.xlabel("Date")
plt.ylabel("Price")
plt.legend()
plt.show()

print("MSE:", mean_squared_error(y_test_inv, y_pred_inv))
```

```
16/16 ━━━━━━━━━━━━━━━━━━━━ 2s 59ms/step
```



```
MSE: 61399.6941069892
```

```
# Start with the last sequence
last_sequence = scaled_data[-sequence_length:]
predictions = []

for _ in range(7):
    input_seq = last_sequence.reshape((1, sequence_length, 1))
    next_price_scaled = model.predict(input_seq)[0][0]
    predictions.append(next_price_scaled)

    # Update the sequence with the new predicted value
    last_sequence = np.append(last_sequence[1:], [[next_price_scaled]], axis=0)

# Convert predictions back to original scale
future_prices = scaler.inverse_transform(np.array(predictions).reshape(-1, 1))

# Display results
for i, price in enumerate(future_prices, 1):
    print(f"Day +{i}: {price[0]:.2f}")
```

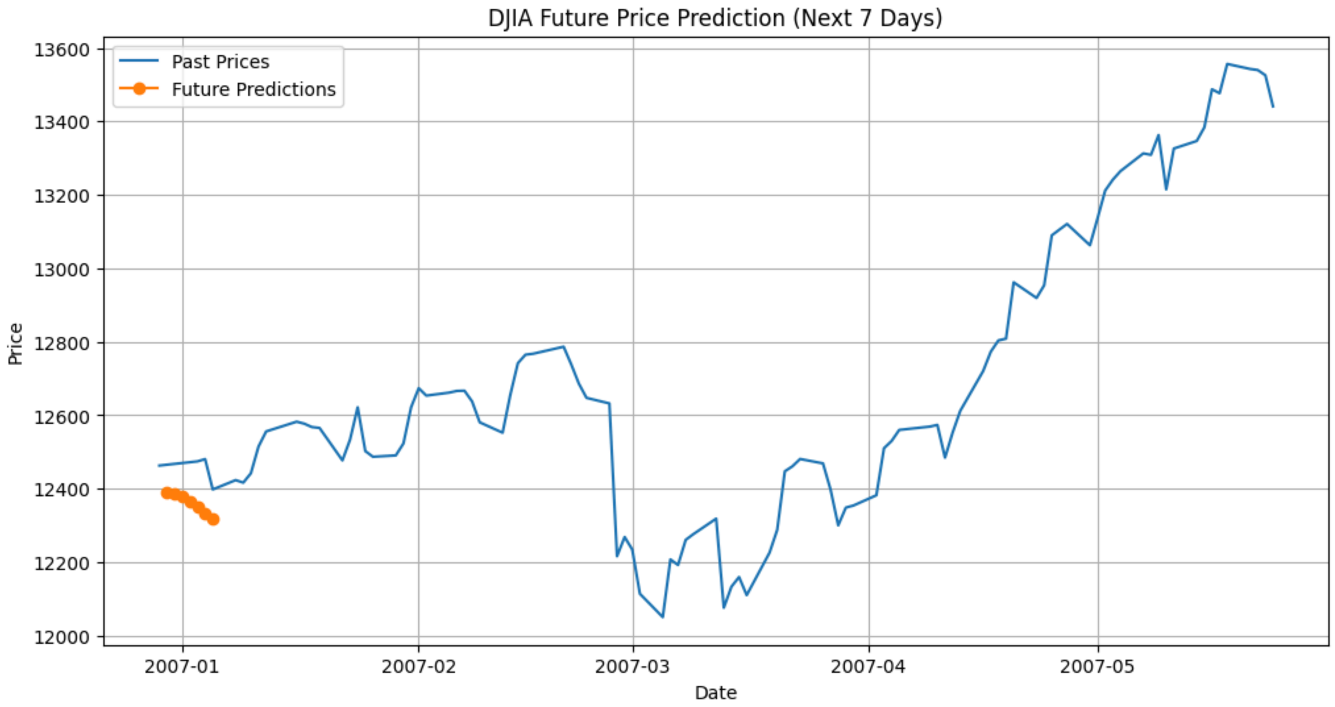
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 310ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 103ms/step
Day +1: 12391.08
Day +2: 12387.55
Day +3: 12378.08
Day +4: 12365.09
Day +5: 12350.11
Day +6: 12334.07
Day +7: 12317.52
```

```
import datetime

# Get index for plotting
last_date = df.index[-1]
future_dates = [last_date + datetime.timedelta(days=i) for i in range(1, 8)]

# Plot
plt.figure(figsize=(12, 6))
plt.plot(df.index[-100:], df['Close'].values[-100:], label="Past Prices")
plt.plot(future_dates, future_prices, label="Future Predictions", marker='o')
plt.xlabel("Date")
plt.ylabel("Price")
plt.title("DJIA Future Price Prediction (Next 7 Days)")
plt.legend()
plt.grid(True)
plt.show()
```



```
!pip install gradio

import gradio as gr

def predict_next(prices):
    if len(prices) != sequence_length:
        return "Please input exactly 60 prices"
    scaled = scaler.transform(np.array(prices).reshape(-1, 1))
    scaled = np.array(scaled).reshape((1, sequence_length, 1))
    pred_scaled = model.predict(scaled)
    pred = scaler.inverse_transform(pred_scaled)
    return f"Predicted next closing price: {pred[0][0]:.2f}"

demo = gr.Interface(
    fn=predict_next,
    inputs=gr.Textbox(label="Enter last 60 closing prices (comma-separated)"),
    outputs="text",
    examples=[",".join(map(str, df['Close'].values[-60:]))]
)
```

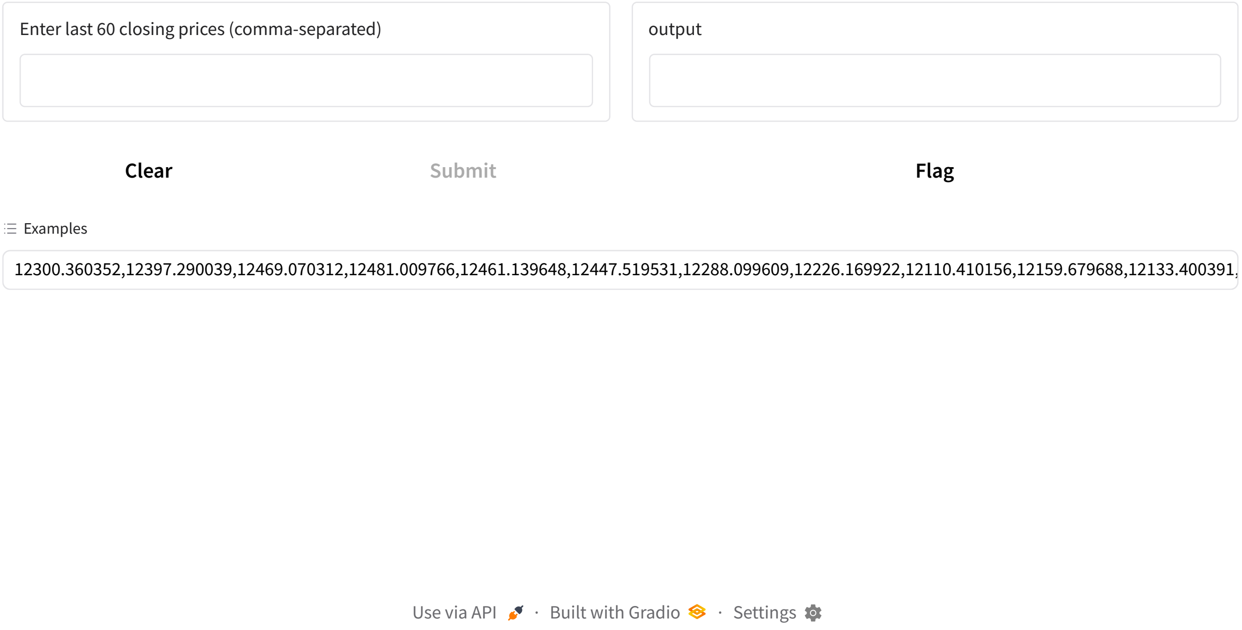
```
    demo.launch()
```

```
Collecting gradio
  Downloading gradio-5.29.0-py3-none-any.whl.metadata (16 kB)
Collecting aiofiles<25.0,>=22.0 (from gradio)
  Downloading aiofiles-24.1.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.9.0)
Collecting fastapi<1.0,>=0.115.2 (from gradio)
  Downloading fastapi-0.115.12-py3-none-any.whl.metadata (27 kB)
Collecting ffmpy (from gradio)
  Downloading ffmpy-0.5.0-py3-none-any.whl.metadata (3.0 kB)
Collecting gradio-client==1.10.0 (from gradio)
  Downloading gradio_client-1.10.0-py3-none-any.whl.metadata (7.1 kB)
Collecting groovy~=0.1 (from gradio)
  Downloading groovy-0.1.2-py3-none-any.whl.metadata (6.1 kB)
Requirement already satisfied: httpx>=0.24.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.28.1)
Requirement already satisfied: huggingface-hub>=0.28.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.31.1)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.1.6)
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.0.2)
Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.0.2)
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.10.18)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from gradio) (24.2)
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.2.2)
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (11.2.1)
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.11.4)
Collecting pydub (from gradio)
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting python-multipart>=0.0.18 (from gradio)
  Downloading python_multipart-0.0.20-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (6.0.2)
Collecting ruff>=0.9.3 (from gradio)
  Downloading ruff-0.11.9-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (25 kB)
Collecting safehttpx<0.2.0,>=0.1.6 (from gradio)
  Downloading safehttpx-0.1.6-py3-none-any.whl.metadata (4.2 kB)
Collecting semantic-version~=2.0 (from gradio)
  Downloading semantic_version-2.10.0-py2.py3-none-any.whl.metadata (9.7 kB)
Collecting starlette<1.0,>=0.40.0 (from gradio)
  Downloading starlette-0.46.2-py3-none-any.whl.metadata (6.2 kB)
Collecting tomlkit<0.14.0,>=0.12.0 (from gradio)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.15.3)
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.13.2)
Collecting uvicorn>=0.14.0 (from gradio)
  Downloading uvicorn-0.34.2-py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->gradio) (2025.3.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->gra
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (2025.4.26)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx>=0.24.1->gradio) (
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (3.18.0
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (2.32.3
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (4.
Requirement already satisfied: hf-xet<2.0.0,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gra
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradi
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gra
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (8.1.8)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (1.5
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (13.9.4)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas<3.0,>=1.0
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0,>=
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0,
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hu
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.2
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->
Downloading gradio-5.29.0-py3-none-any.whl (54.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.1/54.1 MB 16.5 MB/s eta 0:00:00
Downloading gradio_client-1.10.0-py3-none-any.whl (322 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 322.9/322.9 kB 21.1 MB/s eta 0:00:00
Downloading aiofiles-24.1.0-py3-none-any.whl (15 kB)
Downloading fastapi-0.115.12-py3-none-any.whl (95 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 95.2/95.2 kB 8.0 MB/s eta 0:00:00
Downloading groovy-0.1.2-py3-none-any.whl (14 kB)
Downloading python_multipart-0.0.20-py3-none-any.whl (24 kB)
Downloading ruff-0.11.9-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.5/11.5 MB 77.1 MB/s eta 0:00:00
Downloading safehttpx-0.1.6-py3-none-any.whl (8.7 kB)
Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Downloading starlette-0.46.2-py3-none-any.whl (72 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 72.0/72.0 kB 6.3 MB/s eta 0:00:00
Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
Downloading uvicorn-0.34.2-py3-none-any.whl (62 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.5/62.5 kB 5.1 MB/s eta 0:00:00
Downloading ffmpy-0.5.0-py3-none-any.whl (6.0 kB)
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub, uvicorn, tomlkit, semantic-version, ruff, python-multipart, groovy, ffmpy, aiofiles, starle
Successfully installed aiofiles-24.1.0 fastapi-0.115.12 ffmpy-0.5.0 gradio-5.29.0 gradio-client-1.10.0 groovy-0.1.2 pydub-0.25.1
```

```
It looks like you are running Gradio on a hosted a Jupyter notebook. For the Gradio app to work, sharing must be enabled. Automat

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://2fa80f616dbc315c80.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the work
```



Enter last 60 closing prices (comma-separated)

output

12300.360352,12397.290039,12469.070312,12481.009766,12461.139648,12447.519531,12288.099609,12226.169922,12110.410156,12159.679688,12133.400391,

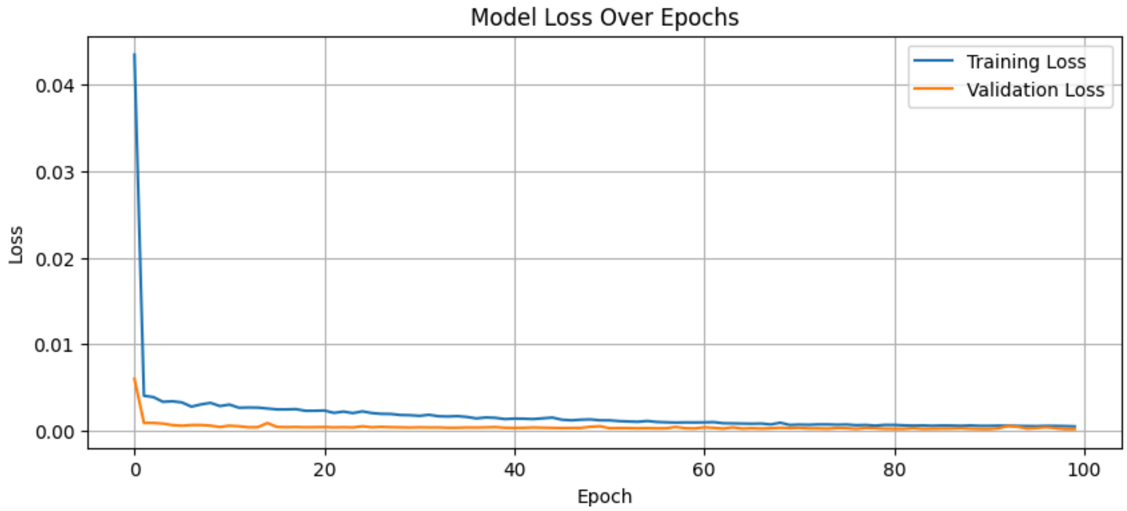
```
from sklearn.metrics import mean_absolute_error, r2_score

mae = mean_absolute_error(y_test_inv, y_pred_inv)
r2 = r2_score(y_test_inv, y_pred_inv)

print(f"📉 Mean Absolute Error (MAE): {mae:.2f}")
print(f"📈 R² Score: {r2:.4f}")
```

```
📉 Mean Absolute Error (MAE): 180.51
📈 R² Score: 0.9688
```

```
plt.figure(figsize=(10, 4))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss Over Epochs')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)
plt.show()
```



```
profits = y_pred_inv - y_test_inv  # simple difference
gain_days = (profits > 0).sum()
```