import numpy as np

import pandas as pd

import yfinance as yf

from keras.models import load\_model

from sklearn.preprocessing import MinMaxScaler

import streamlit as st

# Load the trained model

model = load\_model('C:/Users/varni/OneDrive/Desktop/bitcoin/Bitcoin\_Price\_prediction\_Model.keras')

# Streamlit App Header

st.header('Bitcoin Price Prediction Model')

# Fetch Bitcoin Data

st.subheader('Bitcoin Price Data')

data = pd.DataFrame(yf.download('BTC-USD', '2015-01-01', '2024-04-08'))

# Flatten MultiIndex columns (if any)

if isinstance(data.columns, pd.MultiIndex):

    data.columns = [col[0] for col in data.columns]

# Reset index to have a proper Date column

data = data.reset\_index()

st.write(data)

# Plot Bitcoin Price Line Chart

st.subheader('Bitcoin Line Chart')

st.line\_chart(data[['Date', 'Close']].set\_index('Date'))

# Preprocessing the Data

data.drop(columns=['Date', 'Open', 'High', 'Low', 'Volume'], inplace=True, errors='ignore')

train\_data = data[:-100]

test\_data = data[-200:]

# Scaling the Data

scaler = MinMaxScaler(feature\_range=(0, 1))

train\_data\_scaled = scaler.fit\_transform(train\_data)

test\_data\_scaled = scaler.transform(test\_data)

# Preparing Test Data

base\_days = 100

x\_test, y\_test = [], []

for i in range(base\_days, test\_data\_scaled.shape[0]):

    x\_test.append(test\_data\_scaled[i - base\_days:i])

    y\_test.append(test\_data\_scaled[i, 0])

x\_test, y\_test = np.array(x\_test), np.array(y\_test)

x\_test = np.reshape(x\_test, (x\_test.shape[0], x\_test.shape[1], 1))

# Predicting the Prices

st.subheader('Predicted vs Original Prices')

pred = model.predict(x\_test)

pred = scaler.inverse\_transform(pred)

# Convert to DataFrame for Display

pred\_df = pd.DataFrame(pred, columns=['Predicted Price'])

y\_actual = scaler.inverse\_transform(y\_test.reshape(-1, 1))

actual\_df = pd.DataFrame(y\_actual, columns=['Original Price'])

chart\_data = pd.concat((pred\_df, actual\_df), axis=1)

st.write(chart\_data)

# Plot Predicted vs Original Prices

st.subheader('Predicted vs Original Prices Chart')

st.line\_chart(chart\_data)

# Future Predictions

st.subheader('Predicted Future Bitcoin Prices')

m = y\_test

future\_predictions = []

future\_days = 5

for \_ in range(future\_days):

    m = m.reshape(-1, 1)

    latest\_data = m[-base\_days:].reshape(1, base\_days, 1)

    pred = model.predict(latest\_data)

    m = np.append(m, pred)

    future\_predictions.append(pred)

# Reshape and Inverse Transform Future Predictions

future\_predictions = np.array(future\_predictions).reshape(-1, 1)

future\_predictions = scaler.inverse\_transform(future\_predictions)

# Plot Future Predictions

st.line\_chart(pd.DataFrame(future\_predictions, columns=['Future Predicted Price']))