

```

import streamlit as st

import pandas as pd

import pickle

from sklearn.ensemble import RandomForestRegressor


# load the trained model

model_path = 'random_forest_model.pkl'

with open(model_path, 'rb') as file:

    model_rf = pickle.load(file)


# Function to predict based on user inputs

def predict_btc_price(input_data):

    # Make prediction using the model

    prediction = model_rf.predict(input_data)

    return prediction[0]    #Assuming model returns a single prediction


def main():

    # Title of your web app

    st.title('Predict BTC Close Price')


    # Sidebar for user inputs

    st.sidebar.title('Input Features')


    # Inputs for USDT, BNB closing prices and volumes

    usdt_close = st.sidebar.number_input("USDT Close Price", min_value=0.0, format="%.2f")

    usdt_volume = st.sidebar.number_input('USDT Volume', min_value=0.0, format="%.2f")

    bnb_close = st.sidebar.number_input('BNB Close Price', min_value=0.0, format="%.2f")

    bnb_volume = st.sidebar.number_input('BNB Volume', min_value=0.0, format="%.2f")


    # Create input dataframe

```

```
input_data = pd.DataFrame({  
    'USDT_Close': [usdt_close],  
    'USDT_Volume': [usdt_volume],  
    'BNB_Close': [bnb_close],  
    'BNB_Volume': [bnb_volume]  
})
```

```
# Button to trigger prediction
```

```
if st.button('Predict BTC Close Price'):  
    predicted_price = predict_btc_price(input_data)  
    st.write('Predicted BTC Close Price:', predicted_price)
```

```
if __name__ == '__main__':  
    main()
```