Detailed Modeling Steps

Step 1: Data Loading and Preparation

1. **Load Data**: Load five years of Bitcoin price data (a CSV) into Python.
2. **Convert Date**: Ensure have a date column and convert it to the `datetime` type.
3. **Sort Data**: **(Most important step)** ensure our data is strictly sorted in chronological order (from earliest to latest).
4. **Set Index**: Set the date column as the DataFrame's index. This makes time-based slicing much easier.

Step 2: Feature Engineering

This step answers previous question ("which data features do we want to use as a basis for predictions?"). For time series, the most fundamental and important features are **"Lag Features"—"past prices"**

- **Target (y)**: The value we want to predict, typically tomorrow's "Close Price".
- **Features (X)**: The data we use to make the prediction.
    - **Lag Features**: For example, using the price from 1 day ago, 2 days ago... 7 days ago to predict today's price.
    - **Other Features**: our data might also include "Open", "High", "Low", and "Volume". These are also very useful features.

Step 3: Train-Test Split (Core)

For time series, we **must not** shuffle the data randomly. We must split it chronologically:

1. Calculate the split point: Find the index that marks 80% of the total data.
2. **Training Data**: Select the **first 80%** of the data.
3. **Test Data**: Select the **last 20%** of the data.

The model will learn patterns from the "past" (the first 80%) and then be tested on the "future" (the last 20%).

Step 4: Model Selection and Training

Just examples:

1. **Linear Regression**
2. **Machine Learning Models (Random Forest, XGBoost)**
3. **Time Series Models (ARIMA, Prophet)**:
4. **Deep Learning (LSTM, GRU)**:

Step 5: Prediction and Evaluation ("Comparing Prices")

1. **Predict**: Use the trained model to make predictions on the test set, `X_test`, which gives you `y_pred`.
2. **Evaluate**:
    - **"Compare Prices"**: This is where we compare `y_pred` (Predicted Price) against `y_test` (Actual Price).
    - **Visualize**: Plot `y_pred` and `y_test` on the same graph to see how well the lines fit.
    - **Metrics**: Calculate metrics to quantify the error, such as MAE (Mean Absolute Error) or RMSE (Root Mean Squared Error).