

BTC_APP IN R

Libraries used:

1. **dplyr**: Part of the `tidyverse` collection, used for data manipulation (filtering, selecting, and transforming datasets) with the pipe operator (`%>%`).
2. **tidyverse**: A collection of R packages (including `ggplot2`, `dplyr`, `tidyr`, etc.) designed for data science, making data manipulation and visualization easier.
3. **lubridate**: Helps with working with date and time data, including parsing and converting formats.
4. **forecast**: A package used for time series forecasting and modeling.
5. **ggplot2**: The primary package for creating advanced visualizations and graphs in R.
6. **prophet**: A library developed by Facebook for time series forecasting. It's particularly useful for handling seasonality and irregular time series data.

Explanation of the code:

1. **Data Loading and Cleaning**: We start by loading the BTC hourly data (`BTC-Hourly.csv`) into a dataframe using `read.csv()`. The column `date` is converted from a string format to `POSIXct` using `lubridate` for easier time-based operations. Then, we use `dplyr` functions to remove missing values and select only the relevant columns (`date`, `open`, `high`, `low`, `close`, `Volume.BTC`, `Volume.USD`).

Data Visualization: Using `ggplot2`, we create a line chart to visualize the trend of Bitcoin's closing price over time. The `geom_line()` function is used to create the line, and labels are added for clarity.

Time Series Forecasting: The `prophet` library is used to create a time series forecasting model. We rename the `date` and `close` columns to `ds` and `y` (the required format for `prophet`). The model is trained on the historical data, and predictions are made for the next 30 days.

Model Evaluation: The data is split into training and testing sets. We train the `prophet` model using the training data, make predictions, and compare them with the test set. Finally, we compute the Root Mean Squared Error (RMSE) to evaluate the accuracy of the model.

