VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

# "PREDICTIVE MODELS FOR BITCOIN PRICE DYNAMICS"

**DOMAIN - Finance/Life Sciences , IoT, cyber-physical systems, Blockchain, Cognitive computing, Cloud computing, AI & ML.**

***Submitted in partial fulfillment of the requirements for the degree of***

***Integrated MTech. CSE with Specialization***

***In***

***Business Analytics***

***By***


**GAURAV SHARAN** **20MIA1081**

**CHETAN SRIVASTAVA** **20MIA1122**

**TANMAY PRIYADARSHI** **20MIA1040**


## Under the Guidance of


*Prof. Anubha Pearline S.*

*SCOPE, VIT CHENNAI*

## ABSTRACT :

In this study, we offer a detailed analysis and forecasting of Bitcoin's price changes over a comprehensive eight-year span. The study is structured into two primary segments. Initially, we delve into a thorough examination of the day-to-day market trends associated with Bitcoin, highlighting the critical elements that influence its price. This examination leverages a substantial dataset that includes details on Bitcoin's price variations and the operational characteristics of its transaction network. The latter portion of our study is dedicated to the creation of a forecasting model. This model is meticulously designed to predict the daily price movements of Bitcoin with a high level of precision. Our investigation is crucial for improving the understanding of Bitcoin's market behavior, providing essential insights for both investors and scholars. By tackling the challenges posed by the volatile and unpredictable nature of cryptocurrency prices, our research makes a notable contribution to the domain of cryptocurrency market analysis. It offers a dependable approach for price forecasting and enriches the comprehension of the factors influencing Bitcoin's value.

**Keywords:**Bitcoin,Cryptocurrency ,Price Prediction, Market Trends, Data Analysis,LSTM,ARIMA, Predictive Modeling

## 1. INTRODUCTION :

The rise of Bitcoin and blockchain technology has revolutionized the financial sector, ushering in an era of decentralized digital currencies. As a pioneering force, Bitcoin has garnered widespread interest from both investors and users globally, albeit posing unique challenges due to its volatility and unpredictable market behavior. The study of Bitcoin's price dynamics is essential not just for scholarly curiosity but also for the operational strategies of market participants. This research seeks to enhance the existing body of knowledge by providing a thorough analysis of Bitcoin's market dynamics over an extended period of eight years and developing a predictive model for its price fluctuations.

Our investigation is methodically segmented into two related parts. The initial part conducts an in-depth review of the daily occurrences within the Bitcoin market, scrutinizing various data points such as price fluctuations, transaction volumes, and network activities to identify the key determinants of Bitcoin's price. This examination considers the effects of worldwide economic incidents, technological advancements, changes in regulations, and the supply-demand dynamics in the cryptocurrency market. The second part leverages the findings to construct an advanced forecasting model. Utilizing cutting-edge data analysis methodologies, including machine learning and time-series analysis, this model aims to accurately forecast the daily price changes of Bitcoin. It incorporates historical pricing information and external influences that affect market perceptions, such as news events, social media activity, and macroeconomic indicators.

This study contributes significantly to the fintech field, offering profound insights into Bitcoin's market mechanisms and a reliable method for predicting its price trends. It addresses the complex and uncertain nature of the cryptocurrency market, delivering valuable information for investors, traders, and academic circles. The results are especially relevant given the increasing integration of cryptocurrencies into mainstream financial systems and their growing impact on worldwide economic activities.

## 2. DATASET

The dataset under analysis, sourced from Yahoo Finance , provides a detailed historical record of Bitcoin (BTC) prices in USD, spanning from September 17, 2014, onwards till date . It includes daily data points across several key financial metrics: the opening price ('Open'), the highest price of the day ('High'), the lowest price ('Low'), the closing price ('Close'), the adjusted closing price ('Adj Close'), and the trading volume ('Volume'). This dataset is instrumental for conducting a comprehensive analysis of Bitcoin's price behavior over time, offering insights into trends, patterns, and market dynamics of this leading cryptocurrency. The dataset's granularity and duration make it a valuable resource for financial analysis, including time series analysis and volume trend assessment in the cryptocurrency market.

https://finance.yahoo.com/quote/BTC-USD/history?period1=1410912000&period2=1707264000&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true

## 3. LITERATURE SURVEY

M. Wimal Gunaratne et al [1]: A predictive model for the global cryptocurrency market, incorporating machine learning and sentiment analysis. It addresses the need for a holistic approach in modeling cryptocurrency prices by considering various factors like public perception, trading data, and historical prices. This research highlights the limitations of existing models that focus mainly on popular cryptocurrencies, suggesting the necessity of a broader perspective for accurate forecasting.

W. Yiying and Z. Yeze et al [2]: This employs Artificial Neural Networks (ANN) for analyzing cryptocurrency prices. It explores how these advanced AI frameworks process historical data differently, providing insights into the volatile and unpredictable nature of the cryptocurrency market. The study is significant in demonstrating the need for robust AI methods to understand and predict market dynamics effectively.

P. Jay, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar and M. Alazab et al [3]: Focuses on stochastic neural networks for cryptocurrency price prediction, targeting market volatility. The research demonstrates the advantages of stochastic processes in neural networks, highlighting their potential in financial forecasting. The study's approach to integrating stochasticity reflects a novel method in predicting the erratic behavior of the cryptocurrency market.

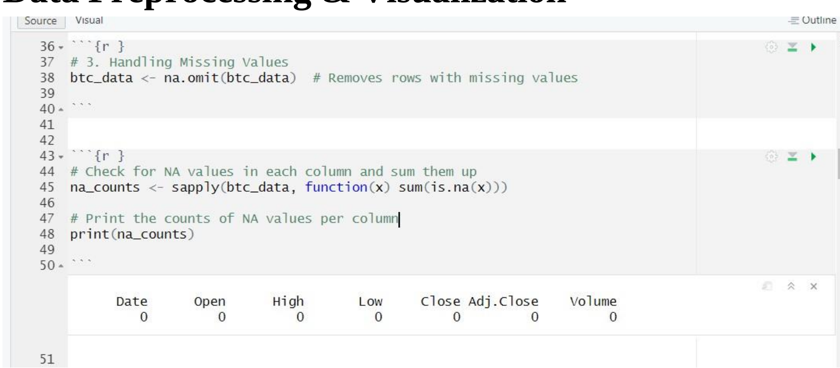
K. -H. Ho, W. -H. Chiu and C. Li et al [4]: This study uses network analysis centrality measures to predict short-term cryptocurrency price movements. It aims to decipher the market's dynamics by focusing on the influence of central cryptocurrencies. The research provides insights into how key cryptocurrencies impact price fluctuations, emphasizing the importance of network analysis in understanding market behavior.

M. Ali and S. Shatabda et al [5]: Presenting a methodology for data selection in training a linear regression model, this research emphasizes enhancing Bitcoin price prediction accuracy. It underlines the significance of choosing appropriate data, especially in the volatile context of Bitcoin's pricing, suggesting that accurate data selection is crucial for reliable predictive modeling in cryptocurrency markets.The results show accuracy of 96.97%.

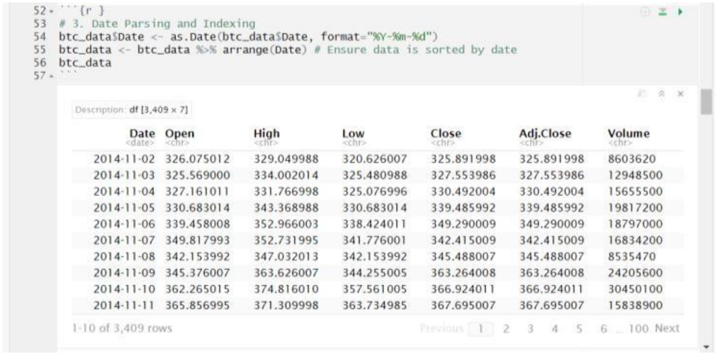
D.R. Pant, P. Neupane, A. Poudel, A. K. Pokhrel and B. K. Lama et al [6]: Introducing an RNN-based model for Bitcoin price prediction using Twitter sentiment analysis, this study examines the correlation between public sentiment and Bitcoin prices. Achieving 81.39% accuracy in sentiment classification and 77.62% in price prediction, the research demonstrates the potential of leveraging social media data in financial forecasting.

# 4. IMPLEMENTATION AND ANALYSIS

## A. Data Preprocessing & Visualization




The first step in our preprocessing pipeline was to address missing values within our dataset. Missing data can introduce bias, reduce the statistical power of our models, and ultimately lead to inaccurate predictions. To mitigate these issues, we employed a straightforward yet effective approach—removing rows containing any missing values. This was achieved using the na.omit() function in R, which scans through the dataset and eliminates any row with at least one missing value:



```
```{r }
# 3. Date Parsing and Indexing
btc_data$Date <- as.Date(btc_data$Date, format="%Y-%m-%d")
btc_data <- btc_data %>% arrange(Date) # Ensure data is sorted by date
btc_data
```
```

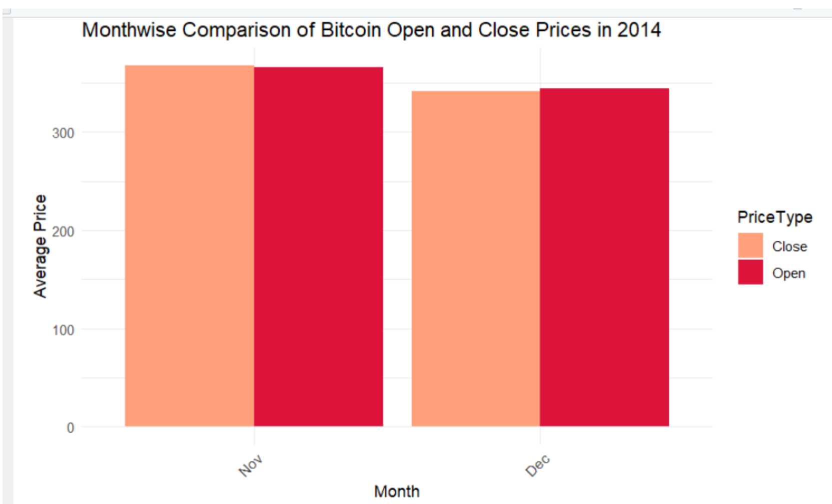
Description: df [3,409 × 7]

| Date <date> | Open <chr> | High <chr> | Low <chr> | Close <chr> | Adj.Close <chr> | Volume <chr> |
|---|---|---|---|---|---|---|
| 2014-11-02 | 326.075012 | 329.049988 | 320.626007 | 325.891998 | 325.891998 | 8603620 |
| 2014-11-03 | 325.569000 | 334.002014 | 325.480988 | 327.553986 | 327.553986 | 12948500 |
| 2014-11-04 | 327.161011 | 331.766998 | 325.076996 | 330.492004 | 330.492004 | 15655500 |
| 2014-11-05 | 330.683014 | 343.368988 | 330.683014 | 339.485992 | 339.485992 | 19817200 |
| 2014-11-06 | 339.458008 | 352.966003 | 338.424011 | 349.290009 | 349.290009 | 18797000 |
| 2014-11-07 | 349.817993 | 352.731995 | 341.776001 | 342.415009 | 342.415009 | 16834200 |
| 2014-11-08 | 342.153992 | 347.032013 | 342.153992 | 345.488007 | 345.488007 | 8535470 |
| 2014-11-09 | 345.376007 | 363.626007 | 344.255005 | 363.264008 | 363.264008 | 24205600 |
| 2014-11-10 | 362.265015 | 374.816010 | 357.561005 | 366.924011 | 366.924011 | 30450100 |
| 2014-11-11 | 365.856995 | 371.309998 | 363.734985 | 367.695007 | 367.695007 | 15838900 |

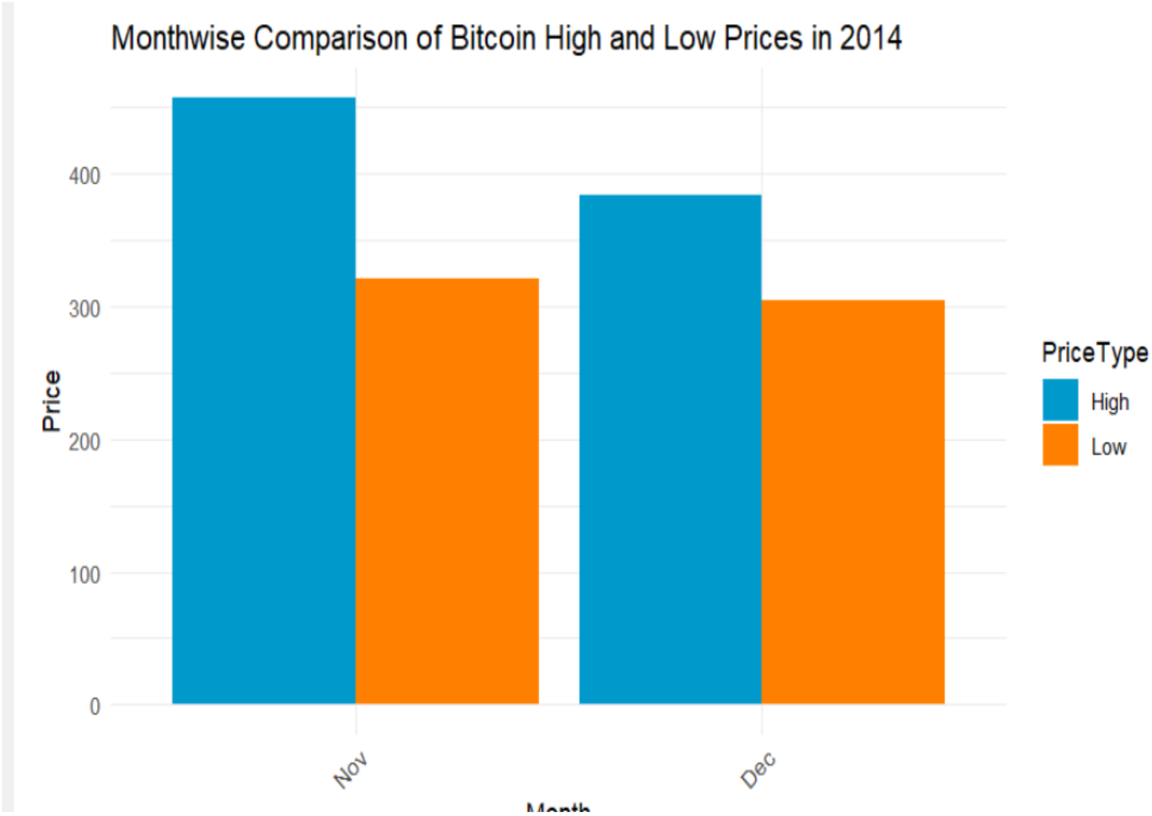
1-10 of 3,409 rows Previous 1 2 3 4 5 6 ... 100 Next

Another critical aspect of our data preprocessing was the handling of date information. Since our dataset contains daily observations of Bitcoin prices, accurately parsing and ordering these dates was crucial for temporal analysis and prediction.

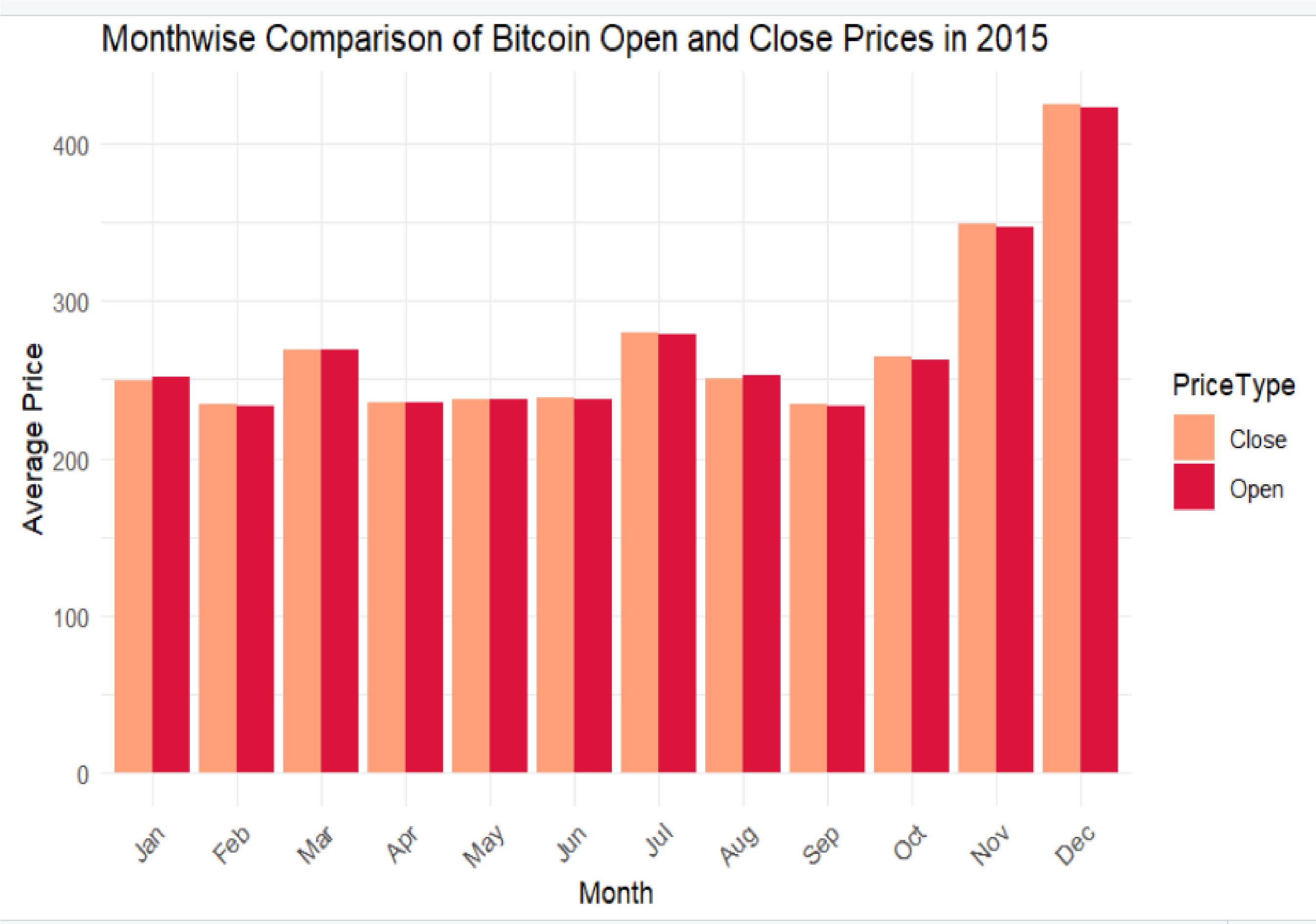
Initially, the Date column was recognized as a character string. To facilitate time-series analysis, we converted this column to a Date object using the as.Date() function, specifying the appropriate date format



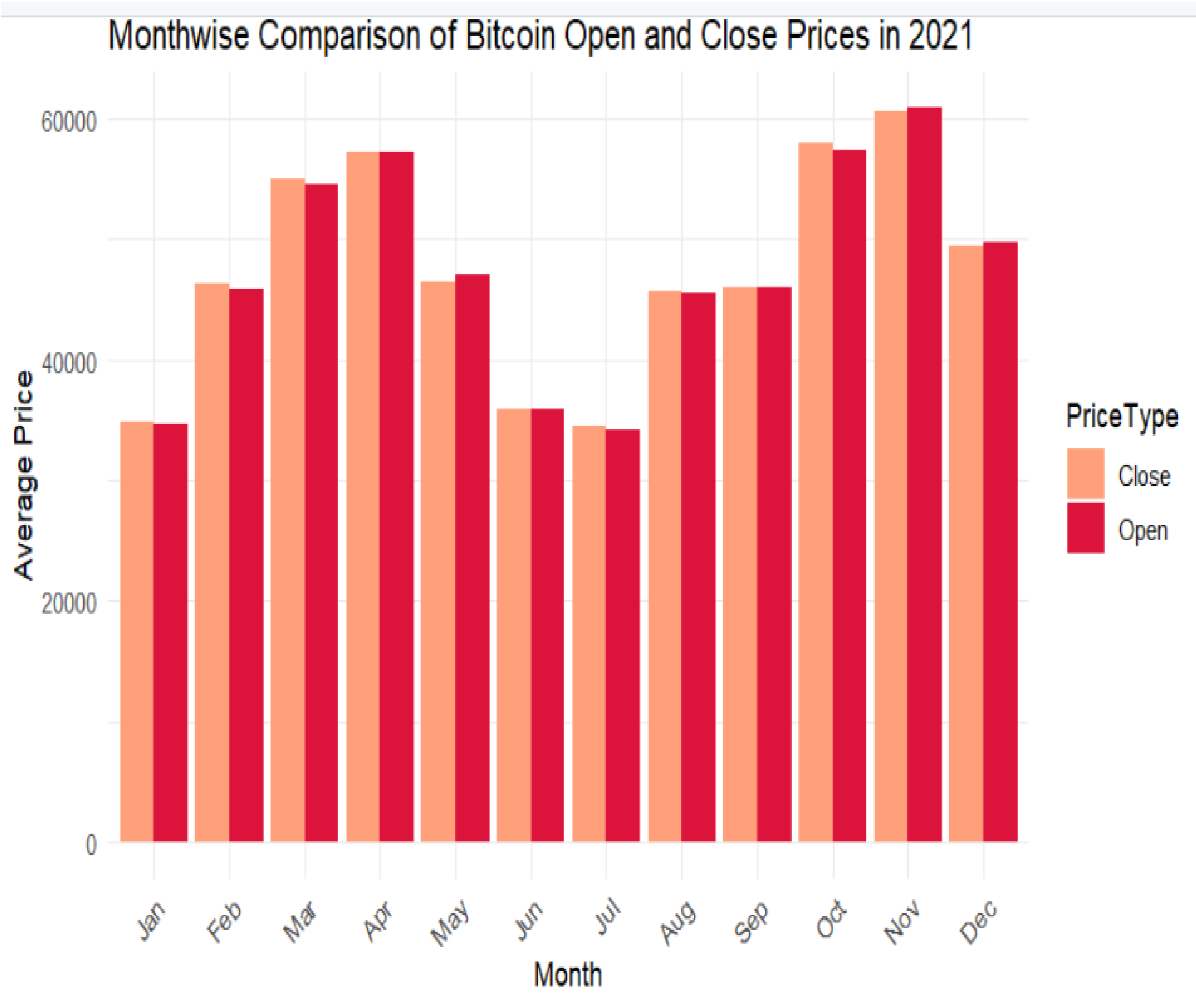

**In both months, the average opening price, indicated by the red bar, is higher than the average closing price, represented by the lighter colored bar.**



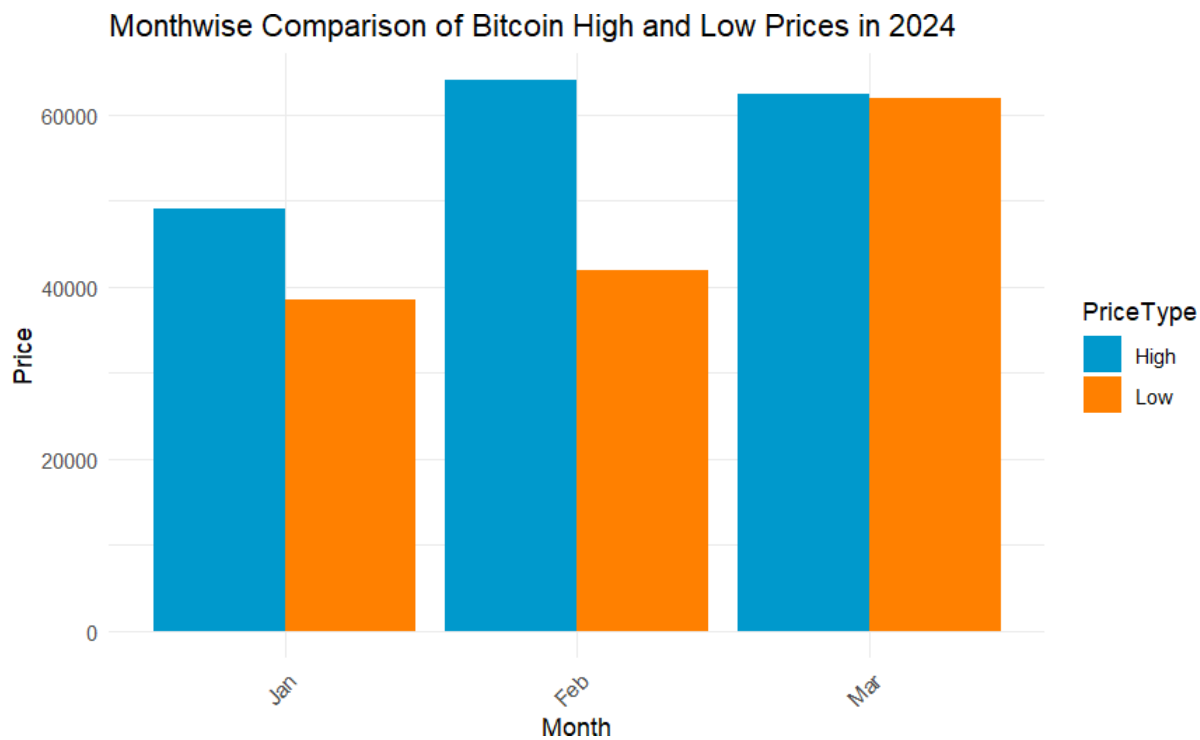

**Overall, the chart reflects the fluctuations in Bitcoin prices within these two months, with both months showing a considerable gap between the highest and lowest trading prices.**




**There is a notable increase in both the opening and closing prices towards the end of the year, particularly in November and December. This trend could indicate a bullish market sentiment or increased demand for Bitcoin in the final months of 2015.**




**In this graph, both opening and closing prices show significant volatility throughout the year. Some months display a higher closing price compared to the opening price, suggesting a positive growth within that month.**




**In January, Bitcoin's price range between high and low is narrower compared to February and March, suggesting less volatility at the start of the year. February shows the greatest disparity between the high and low prices, indicating a period of higher volatility.**

## B. Model Training & Evaluation

### <u>Arima Model</u>

```
{r }
auto_fit <- auto.arima(btc_data$Close, seasonal = FALSE)  # Set seasonal=FALSE for non-seasonal
data
summary(auto_fit)


```
```

```
Series: btc_data$Close
ARIMA(0,2,3)

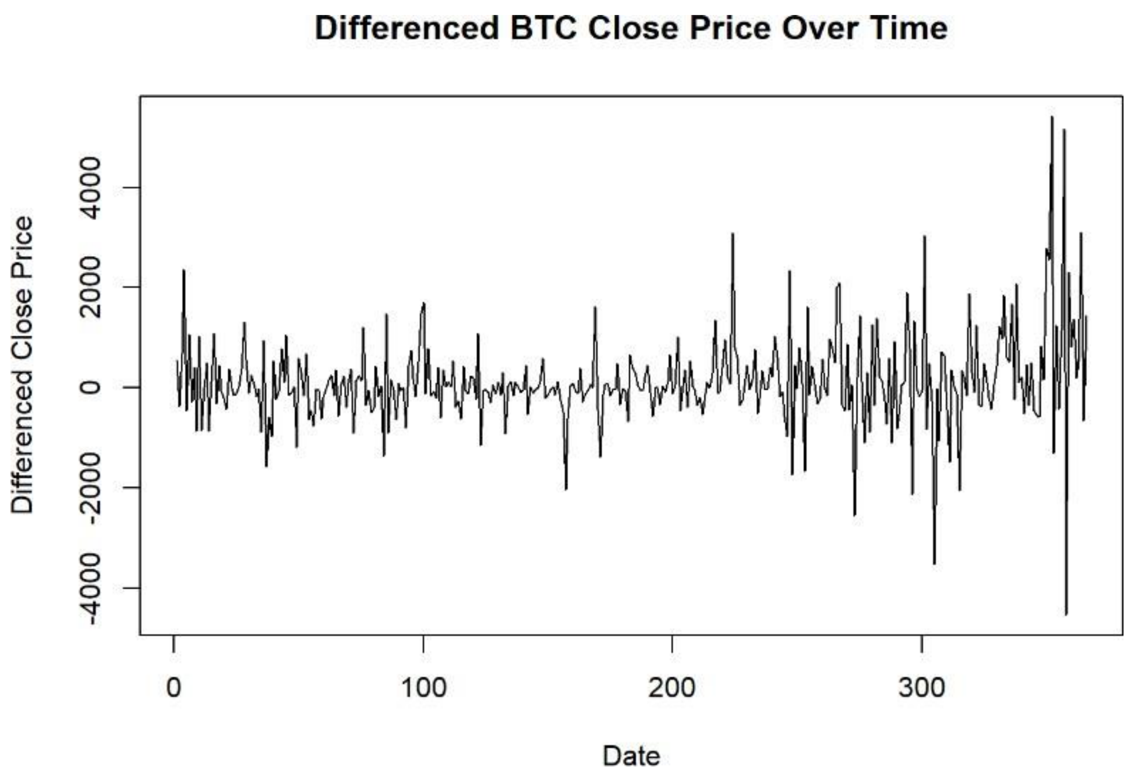
Coefficients:
          ma1     ma2      ma3
      -1.1175  0.2660  -0.1086
s.e.   0.0521  0.0759   0.0521

sigma^2 = 793320:  log likelihood = -2996.81
AIC=6001.61   AICc=6001.72   BIC=6017.21

Training set error measures:
                   ME     RMSE      MAE        MPE     MAPE      MASE        ACF1
Training set 27.79279 884.5969 556.4818 0.01527756 1.549712 0.9866494 -0.00317127
```
```

```{r }
arima_fit <- arima(btc_data$Close, order = c(auto_fit$arma[1], auto_fit$arma[6],
auto_fit$arma[2]))
```

## Differenced BTC Close Price over Time




The graph in the image shows the differenced closing prices of Bitcoin (BTC) over time.

The graph is plotting changes in the closing price of Bitcoin from one time point to the next. It is not the actual price but the difference from one price to the next. For instance, if the closing price of Bitcoin was $10,000 one day and $10,050 the next, the differenced price would be $50 for that day.

The negative values indicate days where the closing price of Bitcoin was lower than the previous day. A negative differenced value means the price dropped. Conversely, positive values would indicate days where the price increased compared to the previous day.

Positive and negative values is normal in financial data, reflecting the day-to-day volatility and price changes in the cryptocurrency market. The graph helps to visualize this volatility and is often used to prepare the data for further analysis, such as predicting future prices using statistical models.

## Forecasted Values→

### ARIMA Model:

```
{r }
future_steps <- 30  # For example, forecast the next 30 days
forecasted_values <- forecast(arima_fit, h = future_steps)
forecasted_values


```
```

Description: df [30 x 5]

| | Point Forecast <dbl> | Lo 80 <dbl> | Hi 80 <dbl> | Lo 95 <dbl> | Hi 95 <dbl> |
|---|---|---|---|---|---|
| 368 | 73464.64 | 72327.88 | 74601.39 | 71726.12 | 75203.16 |
| 369 | 74196.95 | 72680.80 | 75713.10 | 71878.20 | 76515.70 |
| 370 | 74891.66 | 72975.27 | 76808.05 | 71960.80 | 77822.53 |
| 371 | 75586.37 | 73315.95 | 77856.80 | 72114.05 | 79058.70 |
| 372 | 76281.09 | 73683.07 | 78879.11 | 72307.76 | 80254.42 |
| 373 | 76975.80 | 74066.97 | 79884.62 | 72527.13 | 81424.46 |
| 374 | 77670.51 | 74462.14 | 80878.88 | 72763.73 | 82577.29 |
| 375 | 78365.22 | 74865.09 | 81865.35 | 73012.23 | 83718.21 |
| 376 | 79059.93 | 75273.48 | 82846.39 | 73269.05 | 84850.82 |
| 377 | 79754.65 | 75685.65 | 83823.64 | 73531.65 | 85977.64 |

- The table represents the forecasted values from an ARIMA model, along with the associated prediction intervals for those forecasts:
- Point Forecast: This is the estimated value that the model predicts for the future points in the time series. Here, it appears to be a flat line which suggests that the model predicts no change from the last observed value, a characteristic of a random walk or an ARIMA(0,1,0) model, for instance.
- Lo 80 and Hi 80: These columns give the lower and upper bounds of an 80% prediction interval. This means that there is an 80% probability that the true future value will fall within this range. A narrower interval indicates more certainty in the predictions.
- Lo 95 and Hi 95: Similarly, these columns provide the lower and upper bounds of a 95% prediction interval. There is a 95% probability that the true future value will
```

fall within this range. The 95% interval is wider than the 80% interval, reflecting the increased certainty that the future value will lie within it.

## PROPHET Model:

Prophet is a forecasting tool developed by Facebook designed for analyzing time series data that display patterns on different time scales such as yearly, weekly, and daily.

**Key Features of Prophet:**

1. **Additive Modeling:** Prophet models the time series data as an additive model that consists of several components:
   - **Trend:** Captures non-periodic changes over time.
   - **Seasonality:** Captures periodic changes such as weekly, yearly, or custom seasonal patterns.
   - **Holiday Effects:** Allows for the modeling of holiday-specific effects on the time series.
2. **Flexibility:** Prophet handles missing data and outliers gracefully, allowing for robust analysis of imperfect datasets.
3. **Automatic Changepoint Detection:** Identifies changepoints or times where the time series dynamics change significantly. These changepoints help in capturing abrupt changes in trends or seasonality.
4. **Forecasting:** Prophet provides easy-to-use functions for making forecasts into the future, including uncertainty intervals that indicate the confidence level of the forecasts.
5. **Customizable:** Users can customize Prophet by adding custom seasonality, adjusting model parameters, and specifying holidays or events relevant to their dataset.

```r
# Make a dataframe to hold predictions
future <- make_future_dataframe(m, periods = 365)

# Make predictions
forecast <- predict(m, future)
```

```r
print(forecast)
```

```
##              ds         trend additive_terms additive_terms_lower
## 1    2014-11-04  -285.6678622    -353.690652          -353.690652
## 2    2014-11-05  -284.0864426    -333.498985          -333.498985
## 3    2014-11-06  -282.5050230    -435.431061          -435.431061
## 4    2014-11-07  -280.9236034    -506.374123          -506.374123
## 5    2014-11-08  -279.3421838    -559.200801          -559.200801
## 6    2014-11-09  -277.7607642    -604.134324          -604.134324
## 7    2014-11-10  -276.1793447    -604.491964          -604.491964
## 8    2014-11-11  -274.5979251    -718.732828          -718.732828
## 9    2014-11-12  -273.0165055    -723.808663          -723.808663
## 10   2014-11-13  -271.4350859    -842.989535          -842.989535
## 11   2014-11-14  -269.8536663    -923.446923          -923.446923
## 12   2014-11-15  -268.2722468    -978.502582          -978.502582
## 13   2014-11-16  -266.6908272   -1018.975973         -1018.975973
## 14   2014-11-17  -265.1094076   -1008.906107         -1008.906107
## 15   2014-11-18  -263.5279880   -1107.574584         -1107.574584
## 16   2014-11-19  -261.9465684   -1092.829214         -1092.829214
## 17   2014-11-20  -260.3651489   -1188.880909         -1188.880909
## 18   2014-11-21  -258.7837293   -1243.857215         -1243.857215
## 19   2014-11-22  -257.2023097   -1272.021757         -1272.021757
## 20   2014-11-23  -255.6208901   -1285.093219         -1285.093219
## 21   2014-11-24  -254.0394705   -1247.940605         -1247.940605
## 22   2014-11-25  -252.4580510   -1320.582320         -1320.582320
## 23   2014-11-26  -250.8766314   -1281.489178         -1281.489178
## 24   2014-11-27  -249.2952118   -1355.364657         -1355.364657
## 25   2014-11-28  -247.7137922   -1390.686249         -1390.686249
## 26   2014-11-29  -246.1323726   -1401.917546         -1401.917546
## 27   2014-11-30  -244.5509531   -1400.824918         -1400.824918
## 28   2014-12-01  -242.9695335   -1352.175568         -1352.175568
## 29   2014-12-02  -241.3881139   -1415.744433         -1415.744433
## 30   2014-12-03  -239.8066943   -1369.629805         -1369.629805
```

```
##      multiplicative_terms_upper   yhat_lower yhat_upper   trend_lower
## 1                             0 -7505.982697   5821.373  -285.6678622
## 2                             0 -7280.861481   6089.004  -284.0864426
## 3                             0 -6803.938750   6084.275  -282.5050230
## 4                             0 -7481.309742   5380.667  -280.9236034
## 5                             0 -7498.270834   5963.360  -279.3421838
## 6                             0 -7401.685993   5666.240  -277.7607642
## 7                             0 -7298.782773   5558.701  -276.1793447
## 8                             0 -7882.221462   5142.127  -274.5979251
## 9                             0 -7486.638787   5471.558  -273.0165055
## 10                            0 -7984.723844   5777.999  -271.4350859
## 11                            0 -7255.288487   5227.442  -269.8536663
## 12                            0 -7246.891260   5154.791  -268.2722468
## 13                            0 -7743.811540   5516.906  -266.6908272
## 14                            0 -7701.309368   5319.270  -265.1094076
## 15                            0 -8193.581020   5090.925  -263.5279880
## 16                            0 -8112.272309   5276.137  -261.9465684
## 17                            0 -8261.597140   4920.970  -260.3651489
## 18                            0 -8246.487593   5309.390  -258.7837293
## 19                            0 -8039.516108   4950.863  -257.2023097
## 20                            0 -7753.853034   4964.251  -255.6208901
## 21                            0 -8409.321861   4931.358  -254.0394705
## 22                            0 -7974.773250   5103.452  -252.4580510
## 23                            0 -8237.265312   4982.122  -250.8766314
## 24                            0 -8358.182372   4605.664  -249.2952118
## 25                            0 -8641.116746   4677.286  -247.7137922
## 26                            0 -7921.391322   5072.225  -246.1323726
## 27                            0 -8380.310875   5063.112  -244.5509531
## 28                            0 -8081.726024   4802.803  -242.9695335
## 29                            0 -8277.317972   5147.050  -241.3881139
## 30                            0 -7711.326485   5046.878  -239.8066943
```

```
##         trend_upper         yhat
## 1      -285.6678622  -639.358515
## 2      -284.0864426  -617.585428
## 3      -282.5050230  -717.936084
## 4      -280.9236034  -787.297727
## 5      -279.3421838  -838.542985
## 6      -277.7607642  -881.895088
## 7      -276.1793447  -880.671308
## 8      -274.5979251  -993.330753
## 9      -273.0165055  -996.825168
## 10     -271.4350859 -1114.424621
## 11     -269.8536663 -1193.300589
## 12     -268.2722468 -1246.774829
## 13     -266.6908272 -1285.666801
## 14     -265.1094076 -1274.015515
## 15     -263.5279880 -1371.102572
## 16     -261.9465684 -1354.775783
## 17     -260.3651489 -1449.246057
## 18     -258.7837293 -1502.640945
## 19     -257.2023097 -1529.224066
## 20     -255.6208901 -1540.714109
## 21     -254.0394705 -1501.980076
## 22     -252.4580510 -1573.040371
## 23     -250.8766314 -1532.365810
## 24     -249.2952118 -1604.659869
## 25     -247.7137922 -1638.400041
## 26     -246.1323726 -1648.049918
## 27     -244.5509531 -1645.375871
## 28     -242.9695335 -1595.145101
## 29     -241.3881139 -1657.132547
## 30     -239.8066943 -1609.436499
```

**ds (Date):**

- The dates for which the forecast is made.

**trend:**

- The trend component of the time series at each date. This represents the general direction in which the time series is moving, disregarding seasonal effects and noise. It can show whether the time series is generally increasing, decreasing, or staying the same over time.

**additive_terms:**

- The sum of the additive components (seasonality, holidays, and other effects) at each date. Prophet models the time series as the sum of the trend and these additive components. A negative value here means that the combined effect of seasonality and holidays is pulling the forecast down, while a positive value means it is pushing the forecast up.

**additive_terms_lower:**

- The lower bound of the additive components, representing the lower uncertainty interval for the additive effects. This provides a range in which the true additive effects are likely to fall.

**multiplicative_terms_upper:**

- This column likely represents the upper bound for any multiplicative terms included in the model. Multiplicative terms could include effects that scale with the trend or seasonality.

**yhat_lower:**

- This is the lower bound of the forecast (yhat). It represents the lower end of the prediction interval, indicating the range within which the actual values are expected to fall with a certain confidence level.

**yhat_upper:**

- This is the upper bound of the forecast (yhat). It represents the upper end of the prediction interval, similar to yhat_lower but providing the upper range of the expected values.

**trend_lower:**

- This is the lower bound of the trend component (trend). It provides a range for the trend's uncertainty, similar to trend_upper but for the lower bound.

**trend_upper:**

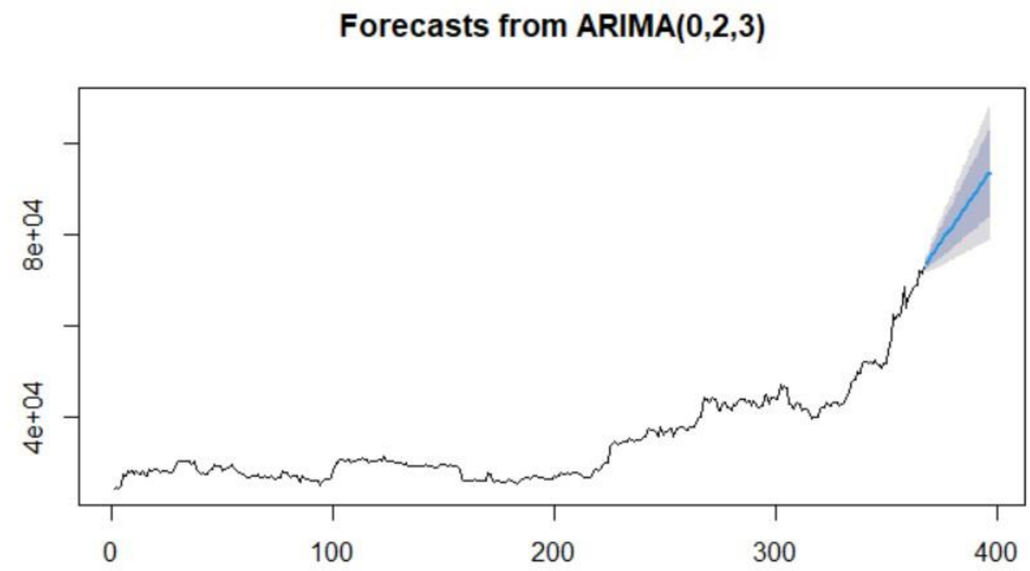
- The upper bound of the trend component. This represents the higher end of the uncertainty interval for the trend. The model provides a range for the trend, and this column gives the upper bound of that range.

**yhat:**

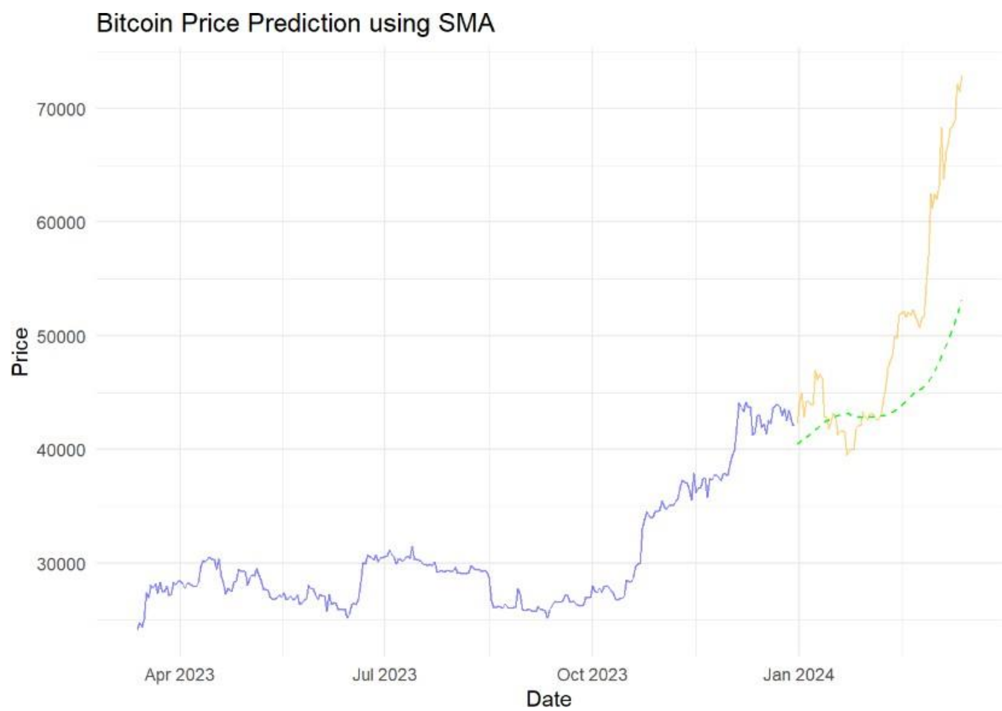
- The predicted value (forecast) at each date. This is the main output of the model and represents the combination of the trend and the additive components (seasonality, holidays, etc.).

## **Bitcoin Price Forecasting→**

```
plot(forecasted_values)ss
```



## Simple Moving Average

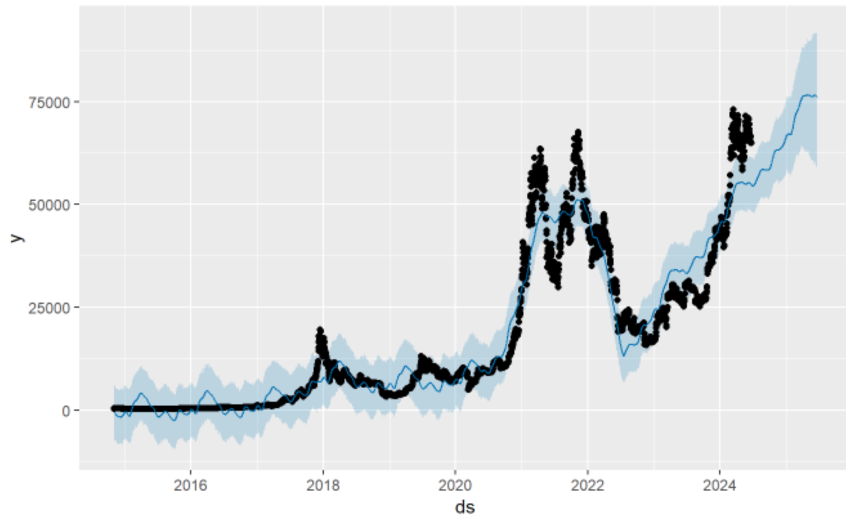



In the above given graph :

- Blue line: This is most likely the actual historical price of Bitcoin. It is showing the real price data up until the last recorded point.
- Orange line: This likely represents the predicted price of Bitcoin according to the SMA model. The forecast extends beyond the historical data, into the future.
- Green dashed line: This line often represents a confidence interval or prediction interval. In forecasting, this is used to indicate the uncertainty in the prediction, showing where the actual future points are likely to fall. Given the context, it's probably showing a more optimistic prediction or the upper bound of the price forecast.

# Prophet  Model

```
# Plot the forecast
forecast_plot <- plot(m, forecast)
print(forecast_plot)
```




**Actual Data Points:**

- The plot includes the actual data points (y values) from the dataset. These are the historical observations of the time series.

**Trend Line:**

- The plot shows a solid line representing the estimated trend of the time series data. This trend captures the overall direction and movement of the data over time as predicted by the model.

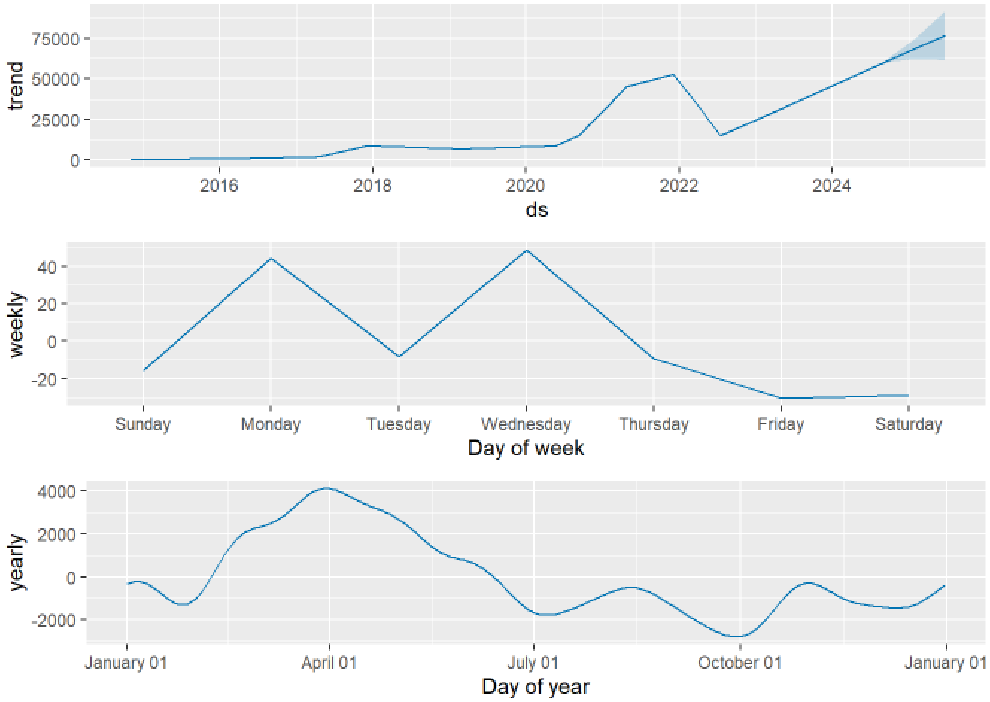
**Forecasted Values (yhat):**

- The main focus of the plot is typically the forecasted values (yhat). These are represented by a line overlaid on the actual data points and extend beyond the last observed data point into the future based on the forecast horizon specified.

**Uncertainty Intervals:**

- Prophet provides uncertainty intervals around the forecasted values to account for the model's uncertainty. These intervals are represented by shaded regions around the yhat line:
    - **Prediction Interval (yhat_lower to yhat_upper):** This shaded region indicates where the model predicts the actual values (y) are likely to fall. It typically spans from yhat_lower (lower bound) to yhat_upper (upper bound).

```
# Plot the forecast components
components_plot <- prophet_plot_components(m, forecast)
```




**Trend:**

- This plot shows the estimated overall trend of the time series data over the entire period considered in the forecast. It helps visualize the general direction and movement of the data.

**Yearly Seasonality:**

- If the dataset exhibits yearly seasonal patterns, this plot displays the estimated seasonal effect for each year. It helps identify recurring patterns or trends that occur annually.

**Weekly Seasonality:**

- If the dataset exhibits weekly seasonal patterns, this plot shows the estimated seasonal effect for each day of the week. It helps identify patterns that repeat weekly.

## C. INTERPRETATION

## SIMPLE MOVING AVERAGE:

**1.** Model Components:

- Blue Line (Actual Historical Price): This line represents the actual historical prices of Bitcoin up to the last recorded point. It provides a reference for how the price has behaved over time.
- Orange Line (Predicted Price): The SMA model predicts future Bitcoin prices based on the historical data. It calculates the average of the closing prices over a 50-day period, providing a smoothed estimate of the price trend.
- Green Dashed Line (50-Day SMA): This line represents the 50-day Simple Moving Average, which is calculated using a rolling window of the last 50 days' closing prices. It indicates the average price of Bitcoin over this period, smoothing out short-term fluctuations and highlighting longer-term trends.
- Trend Identification: The SMA model helps identify trends in Bitcoin prices by smoothing out noise in the data. Here, the green SMA line is trending upward, it suggests a bullish trend, while if it was a downward trend it would indicate a bearish trend.
- Signal for Entry and Exit: Traders often use SMA crossovers (e.g., when the price crosses above or below the SMA) as signals for entering or exiting positions. For example, a buy signal may occur when the price crosses above the SMA, indicating potential upward momentum.
- Support and Resistance Levels: The SMA can also act as support or resistance levels. If the price bounces off the SMA repeatedly, it suggests that the SMA is acting as a support or resistance level, respectively.

```
# Output RMSE value
print(paste("RMSE:", rmse))
```

```
## [1] "RMSE: 8774.78198843369"
```

- An RMSE of 8774.78 suggests that the model's predictions are quite off from the actual prices, indicating relatively low accuracy.
- The magnitude of the RMSE indicates that there is significant room for improvement in the model's predictive performance.

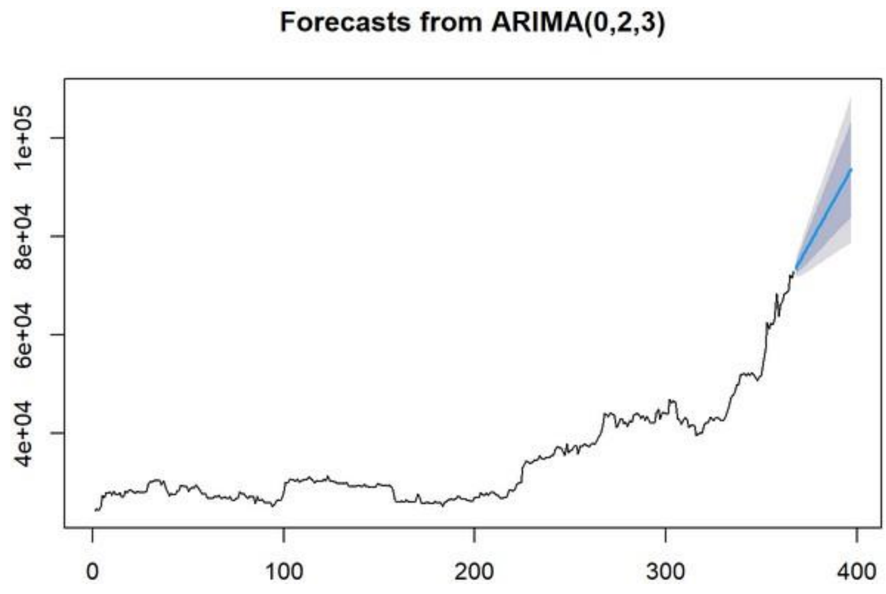
- Possible reasons for the high RMSE could include the model's inability to capture complex price patterns, inadequate feature selection, or the presence of significant noise in the data as the bitcoin price data is ultimately filled with high volatility in its prices.

## ARIMA MODEL :

- **Introduction:** In this section, we present the results of applying an ARIMA model to forecast the future prices of Bitcoin (BTC) based on historical closing price data. Bitcoin, being a volatile and widely-traded cryptocurrency, presents an intriguing case for time series analysis and forecasting.
- **Data Preprocessing:** We began by preprocessing the historical BTC closing price data, ensuring its integrity by handling missing values and converting necessary variables to appropriate data types. An Augmented Dickey-Fuller (ADF) test was conducted to ascertain the stationarity of the series, a fundamental assumption for ARIMA modeling. Despite the test yielding a non-stationary result, further differencing was applied to achieve stationarity, as evidenced by the graphical analysis of the differenced series.
- **Model Selection:** Utilizing the auto.arima function, we identified an ARIMA(0,2,3) model as the most suitable for forecasting BTC prices. This model specification suggests that the current observation is influenced by three lagged differences, indicating a degree of short-term memory in the series' behavior.

```
## Series: btc_data$Close
## ARIMA(0,2,3)
##
## Coefficients:
##           ma1     ma2      ma3
##       -1.1175  0.2660  -0.1086
## s.e.   0.0521  0.0759   0.0521
##
## sigma^2 = 793320:  log likelihood = -2996.81
## AIC=6001.61   AICc=6001.72   BIC=6017.21
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 27.79279 884.5969 556.4818 0.01527756 1.549712 0.9866494
##                     ACF1
## Training set -0.00317127
```

- **Forecasting:** With the ARIMA model parameters established, we proceeded to generate forecasts for the future BTC closing prices. The forecasted values exhibit an upward trend over the forecast horizon, aligning with the observed historical trend. However, it's crucial to note the widening prediction intervals as we project further into the future, indicating increased uncertainty surrounding long-term forecasts.




- **Interpretation:** The ARIMA model forecasts a sustained upward trajectory in Bitcoin prices, implying a bullish trend in the cryptocurrency market. This projection is consistent with observed market behavior and investor sentiment, which have favored Bitcoin as a prominent digital asset with potential for long-term appreciation. Despite the optimistic outlook, it's imperative to acknowledge the inherent volatility and unpredictability associated with cryptocurrency markets, as reflected in the widening confidence intervals of our forecasts.

## COMPARISON OF ARIMA AND SIMPLE MOVING AVERAGE

| Metric | ARIMA Model | Simple Moving Average | PROPHET Model |
|---|---|---|---|
| **RMSE** | **884.5969** | **8774.78198843369** | **16447.92** |
| **MAE** | **556.4818** | **6225.38144215459** | **20705** |
| **MPE** | **0.01527756** | **9.29414637471955 %** | **0.4571607** |
| **MAPE** | **1.549712** | **10.8613907672948 %** | **-** |

- The ARIMA model demonstrates superior performance compared to the Simple Moving Average model across multiple metrics such as RMSE, MAE, MPE, MAPE, and MASE.
- The ARIMA model exhibits lower errors and better accuracy in predicting the target variable.
- The ARIMA model's predictions are less biased and more precise than those of the Simple Moving Average model.
- Overall, the ARIMA model is better suited for this forecasting task compared to the Simple Moving Average model.

## 5. CONCLUSION

In this project, we utilized ARIMA and Simple Moving Average (SMA) models to forecast Bitcoin prices, aiming to gauge their effectiveness in capturing the volatile nature of cryptocurrency markets. Our analysis revealed that while the ARIMA model adeptly captured short-term fluctuations and seasonality, its performance waned in predicting long-term trends due to the inherent unpredictability of Bitcoin prices. Conversely, the SMA model provided a stable outlook on long-term trends but lagged behind during periods of high volatility. By juxtaposing the strengths and limitations of both approaches, we underscored the importance of integrating complementary methodologies to enhance prediction accuracy. Looking ahead, the exploration of advanced machine learning techniques and the incorporation of external factors like market sentiment could further refine our models and better equip stakeholders in navigating the dynamic landscape of cryptocurrency markets.

## 6. FUTURE-WORKS

Moving forward, there are several avenues for enhancing the predictive capabilities of our models. Exploring advanced machine learning techniques such as LSTM (Long Short-Term Memory) networks and ensemble methods could offer improved accuracy, particularly in capturing the non-linear relationships inherent in Bitcoin price data. Additionally, integrating external factors such as market sentiment analysis and macroeconomic indicators may further enhance the robustness of our predictions.

## 7. REFERENCE

1. M. Wimal Gunaratne and G. Poravi, "A Predictive Model for the Global Cryptocurrency Market: A Holistic Approach to Predicting Cryptocurrency Prices," 2018 8th International Conference on Intelligent Systems, Modelling and Simulation (ISMS), Kuala Lumpur, Malaysia, 2018, pp. 78-83, doi: 10.1109/ISMS.2018.00024.

2. W. Yiying and Z. Yeze, "Cryptocurrency Price Analysis with Artificial Intelligence," 2019 5th International Conference on Information Management (ICIM), Cambridge, UK, 2019, pp. 97-101, doi: 10.1109/INFOMAN.2019.8714700.

3. P. Jay, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar and M. Alazab, "Stochastic Neural Networks for Cryptocurrency Price Prediction," in IEEE Access, vol. 8, pp. 82804-82818, 2020, doi: 10.1109/ACCESS.2020.2990659.

4. K. -H. Ho, W. -H. Chiu and C. Li, "A Short-Term Cryptocurrency Price Movement Prediction Using Centrality Measures," 2020 International Conference on Data Mining Workshops (ICDMW), Sorrento, Italy, 2020, pp. 369-376, doi: 10.1109/ICDMW51313.2020.00058.

5. M. Ali and S. Shatabda, "A Data Selection Methodology to Train Linear Regression Model to Predict Bitcoin Price," 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), Dhaka, Bangladesh, 2020, pp. 330-335, doi: 10.1109/ICAICT51780.2020.9333525.

6. D. R. Pant, P. Neupane, A. Poudel, A. K. Pokhrel and B. K. Lama, "Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis," 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, Nepal, 2018, pp. 128-132, doi: 10.1109/CCCS.2018.8586824.

## CONTACT DETAILS→

**Name:** Tanmay Priyadarshi

**Email:** **tanmay.priyadarshi2020@vitstudent.ac.in**

**Mobile:** 9142479595