
# TIME SERIES FORECASTING OF BITCOIN PRICES USING PROPHET

**TEAM**

| Candace Joy Pals | Megha R Krishna | Abhinav Bijith | Gopika Anilkumar |
|---|---|---|---|
| 20BCE2128 | 20BEC0719 | 20BCE2149 | 20BEC0453 |
| VIT, Vellore | VIT,Vellore | VIT,Vellore | VIT,Vellore |

# TABLE OF CONTENTS

## 1 INTRODUCTION

- ### 1.1 Overview

The report focuses on "Time Series Forecasting of Bitcoin Prices using Prophet." It aims to explore the application of the Prophet forecasting model, developed by Facebook, for predicting future Bitcoin prices.
The report begins by introducing the significance of time series forecasting in the domain of cryptocurrency, particularly in predicting Bitcoin prices. It highlights the challenges associated with price prediction due to the volatile nature of Bitcoin and the need for accurate forecasting models.
The theoretical section outlines the steps involved in the implementation of the Prophet model for Bitcoin price prediction. It describes the data collection process using the finance library to obtain historical Bitcoin price data. The preprocessing steps, such as formatting the data and separating it into training and testing sets, are explained.
Report then discusses the Prophet model's configuration, including parameters such as the interval width and the number of changepoints. It emphasizes the flexibility and effectiveness of the Prophet model in capturing the complex patterns and trends observed in Bitcoin price data.
The experimental investigation section presents the findings of applying the Prophet model to the Bitcoin price dataset. It includes performance metrics such as root mean squared error (RMSE) to evaluate the accuracy of the predictions. The section showcases visualizations, such as candlestick charts and scatter plots, illustrating the historical prices and the predicted Bitcoin prices.
The discussion and result section analyzes the results and provides insights into the strengths and limitations of using the Prophet model for Bitcoin price forecasting. It discusses the model's ability to capture short-term fluctuations and long-term trends and examines the potential factors affecting the accuracy of the predictions.
The conclusion section summarizes the key findings of the project and highlights the contribution of using the Prophet model for Bitcoin price prediction. It suggests potential areas for future research, such as exploring additional features or incorporating external factors to enhance the forecasting accuracy.
Overall, it presents the implementation details, evaluation results, and discusses the implications of using the Prophet model in the context of cryptocurrency forecasting.

- ### 1.2 Purpose

The purpose of the above project is to address the need for accurate and reliable predictions of Bitcoin prices using a time series forecasting approach. The volatile nature of Bitcoin makes it challenging for investors, traders, and researchers to anticipate its future movements. Therefore, the project aims to leverage the Prophet forecasting model, a powerful tool developed by Facebook, to provide reliable predictions and insights into the future price trends of Bitcoin.
By implementing the project, the goal is to empower individuals and organizations in the cryptocurrency domain with a tool that can assist in decision-making processes. Accurate Bitcoin price predictions can help traders identify optimal entry and exit points, investors make informed investment decisions, and researchers gain a deeper understanding of the underlying factors affecting Bitcoin's value. Ultimately, the project seeks to enhance the overall understanding and utilization of Bitcoin by leveraging advanced forecasting techniques to improve predictions and generate valuable insights.

## 2 LITERATURE SURVEY

**Research Paper 1:**
Title: "Bitcoin Price Prediction and Analysis Using Deep Learning Models"
Authors: Temesgen Awoke, Minakhi Rout, Lipika Mohanty,and Suresh Chandra Satapathy
Published: 2020

Summary:
The research paper titled "Bitcoin Price Prediction and Analysis Using Deep Learning Models" delves into the application of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models for forecasting the price volatility of Bitcoin. The study examines the performance of these deep learning models and compares their effectiveness in predicting Bitcoin prices. The findings indicate that both LSTM and GRU models demonstrate the ability to make accurate predictions. However, the GRU model exhibits advantages such as faster convergence and more consistent results compared to LSTM. This research contributes to the field of cryptocurrency analysis by shedding light on the potential of deep learning techniques for price prediction in the context of Bitcoin.

**Research Paper 2:**
Title: "Bitcoin Price Forecasting Using Time Series Analysis "
Authors: Shaily Roy, Samiha Nanjiba, Amitabha Chakrabarty
Published: 2018
Summary:
This paper proposes an ARIMA model for predicting the market price of Bitcoin. It analyzes four years of Bitcoin data and aims for accurate short-term volatility predictions. The study highlights the importance of understanding Bitcoin's price fluctuations and its impact on the economy. The authors compare different models and select ARIMA as the most suitable. They evaluate the model's performance using RMSE and percentage error. The results show that ARIMA provides more accurate predictions compared to other models. Overall, the paper contributes to Bitcoin price prediction using time series analysis and offers insights for investors and researchers.

**Research Paper 3:**
Title: "Machine Learning Models Comparison for Bitcoin Price Prediction"
Authors: Thearasak Phaladisailoed and Thanisa Numnonda
Published: 2018
Summary:
The research paper compares different machine learning models for predicting Bitcoin prices using trading data from the Bitstamp exchange. The models evaluated include Theil-Sen Regression, Huber Regression, LSTM, and GRU. The study finds that the deep learning models (LSTM and GRU) outperform the traditional regression models, with GRU achieving the best results in terms of accuracy. The paper provides an overview of scikit-learn, Tensorflow, and Keras libraries used for modeling and discusses previous studies in the field. Overall, the research identifies GRU as the most efficient and accurate model for Bitcoin price prediction.

**Research Paper 4:**
Title: "Bitcoin Price Prediction Using Ensembles of Neural Networks "
Authors: Edwin Sin and Lipo Wang
Published: 2017
Summary:
This paper explores the use of an Artificial Neural Network ensemble called Genetic Algorithm based Selective Neural Network Ensemble (GASEN) to predict the next day direction of the price of Bitcoin. The ensemble is constructed using Multi-Layered Perceptron (MLP) as the base model for each neural network in the ensemble. The dataset used includes approximately 200 features of Bitcoin over a span of 2 years.
It compares the performance of the ensemble-based trading strategy against a "previous day trend following" strategy and a strategy that follows the single best MLP model in the ensemble. Back-testing over a period of 50 days shows that the ensemble-based strategy generated almost 85% returns, outperforming the other two strategies.The research highlights the effectiveness of the ensemble approach in predicting Bitcoin price changes and demonstrates its potential for real-world application. The results suggest that the ensemble approach can provide valuable insights for decision-making in Bitcoin trading.

**Research Paper 5:**
Title: "Cryptocurrency Price Prediction Using Twitter Sentiment Analysis"
Authors: S. Kumar, P. Gupta
Published: 2023
Summary:
This paper presents an end-to-end model for cryptocurrency price prediction using Twitter sentiment analysis. The model combines a Bidirectional Encoder Representations from Transformers (BERT) model for sentiment prediction and a Gated Recurrent Unit (GRU) model for price forecasting. Historical price data, tweet volume, user information, and sentiment scores are used as inputs. The model achieves a Mean Absolute Percentage Error (MAPE) of 9.45% for sentiment prediction and 3.6% for price prediction, demonstrating its effectiveness in capturing sentiment trends and forecasting cryptocurrency prices.
The study also explores the impact of different sentiment features and shows that sentiment scores derived from BERT outperform other sentiment analysis techniques. Additionally, the model exhibits robustness in predicting price fluctuations across multiple cryptocurrencies, indicating its potential applicability in a broader range of digital assets. The findings suggest that integrating social media sentiment analysis can contribute to more accurate cryptocurrency price forecasting.

- ## 2.1 Existing problem

  - ❖ **Method 1: Machine Learning Algorithms**
    Machine Learning (ML) algorithms, such as Support Vector Machines (SVM), Random Forest, and Gradient Boosting, have been applied to predict Bitcoin prices. The advantages of ML algorithms include their ability to capture complex patterns and non-linear relationships in the data. They also offer flexibility in feature selection and can handle large datasets. However, these methods often require extensive feature engineering and are prone to overfitting, leading to reduced generalization performance.

  - ❖ **Method 2: Time Series Analysis Techniques**
    Traditional time series analysis techniques, such as Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing (ES), and Seasonal Decomposition of Time Series (STL), have been employed for Bitcoin price prediction. These methods offer simplicity, interpretability, and the ability to capture trend and seasonality in the data. However, they may struggle to handle non-linear patterns and sudden changes in Bitcoin prices, limiting their predictive accuracy.

  - ❖ **Method 3: Deep Learning Models**
    Deep Learning models, particularly Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, have been utilized for Bitcoin price prediction. These models excel at capturing sequential dependencies and non-linear patterns in the data. They have the advantage of automatic feature extraction and can handle large and complex datasets. However, Deep Learning models require substantial computational resources and may be prone to overfitting, especially when trained on limited data.

  - ❖ **Method 4: Ensemble Methods**
    Ensemble methods, which combine multiple prediction models, have been proposed for Bitcoin price forecasting. These methods aim to improve prediction accuracy, enhance model robustness, and reduce the risk of model bias. Ensemble methods offer the advantage of leveraging the strengths of different models and can potentially achieve better performance.

However, implementing ensemble models can be more complex, and selecting and combining models effectively can be challenging.

- ❖ **Method 5: Sentiment Analysis**
  Another approach is to integrate sentiment analysis from social media and news data into Bitcoin price prediction models. This method involves capturing market sentiment as an additional feature to capture investor sentiment and market trends. However, sentiment analysis can be subjective, and the impact of false or misleading information must be considered.

- **2.2 Proposed solution**

Bitcoin Price Prediction using Prophet Model
The proposed method for Bitcoin price prediction utilizes the Prophet model. The advantages of the Prophet model include its ability to handle trend changes, seasonality, and holiday effects in the data. It also provides automatic feature selection and incorporates domain knowledge to enhance prediction accuracy. Additionally, Prophet offers intuitive model interpretation and simplicity in model implementation.

Compared to the previously mentioned methods, the Prophet model simplifies the prediction process and reduces the need for extensive feature engineering. It handles trend changes and seasonality effectively, which are crucial factors in Bitcoin price prediction. However, it is important to note that the Prophet model may not capture more complex non-linear patterns as effectively as some Deep Learning models. Additionally, the choice of method depends on the specific requirements and characteristics of the Bitcoin price prediction task.
The Bitcoin price may be predicted using Prophet, which is preferable in comparison to conventional approaches for a number of different reasons. The Prophet model takes into account seasonality, which enables it to recognise recurrent pricing trends over a variety of time frames. It is able to reliably manage missing data and outliers, which is essential in the highly volatile bitcoin market. The findings that Prophet generates are easy to grasp and may be interpreted in a way that provides transparency into the underlying patterns. It eliminates the need for lengthy manual feature engineering by automatically removing essential elements and extracting those features. In addition, Prophet is resistant to changes in the data and provides an installation method that is simple, making it accessible to analysts and researchers. In general, Prophet is a tempting option for Bitcoin price prediction due to its capacity to manage seasonality, resilience to data fluctuations, interpretability, and simplicity of implementation.

## 3 THEORITICAL ANALYSIS

- **3.1 Hardware / Software designing**

Hardware Requirements:

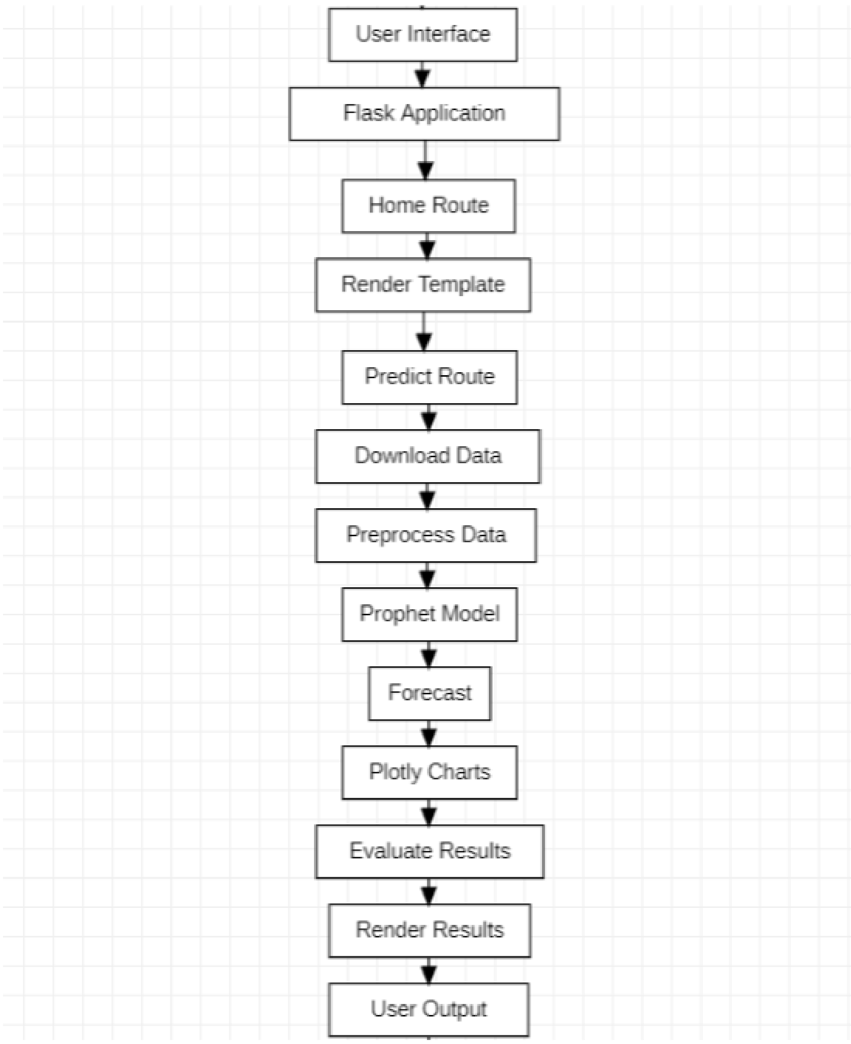
1. Computer
2. Processor- A multi-core processor (e.g., Intel Core i5 or higher) for faster computation.
3. Memory (RAM)- At least 8 GB of RAM, although more is beneficial for handling larger datasets.
4. Storage- Sufficient storage space to store the dataset, software, and model files.
5. Graphics Processing Unit (GPU) (optional): A GPU can accelerate computations for deep learning models, but it is not strictly necessary for the Prophet model.

Software Requirements:

1. Python: The project will require programming in Python, so ensure Python is installed on your computer.

2. Integrated Development Environment (IDE): Choose an IDE such as PyCharm, Jupyter Notebook, or Visual Studio Code to write and execute your Python code.
3. Python Libraries: Install the necessary Python libraries, including:
    - Pandas: For data manipulation and analysis.
    - NumPy: For numerical computations.
    - Matplotlib and Seaborn: For data visualization.
    - Scikit-learn: For machine learning tasks.
    - Prophet: The main library for implementing the Prophet model.
4. Data Source: Identify a reliable source of historical Bitcoin price data. You can use cryptocurrency APIs or online platforms that provide historical data in a suitable format (e.g., CSV, JSON)-in our case we are taking it from Yahoo Finance
5. Data Storage: Ensure you have sufficient storage space to store and manage the dataset locally.

## • 3.2 Block diagram



## 4 EXPERIMENTAL INVESTIGATIONS

During the development of the project, several analyses and investigations were conducted to ensure the effectiveness and reliability of the solution. Here are some key analyses performed:

1. **Data Analysis:**
An in-depth analysis of historical Bitcoin price data was conducted to understand its characteristics, trends, and potential patterns. This analysis involved exploring statistical measures, visualizing price movements, identifying seasonality or trends, and detecting outliers or anomalies.

2. **Model Selection:**
Various forecasting models were evaluated and compared to select the most suitable model for the task. The Prophet model was chosen based on its ability to handle time series data, capture non-linear trends, and incorporate seasonality and uncertainty in its predictions.

3. **Parameter Tuning:**
A parameter tuning process was conducted to optimize the configuration of the Prophet model. This involved adjusting parameters such as the interval width, number of changepoints, and trend flexibility to achieve the best forecasting performance. Grid search, random search, or other optimization techniques may have been utilized.

4. **Model Evaluation:**
The performance of the Prophet model was assessed using appropriate evaluation metrics such as root mean squared error (RMSE), mean absolute error (MAE), or other relevant measures. The model's accuracy in predicting Bitcoin prices was analyzed by comparing the predicted values against the actual values.

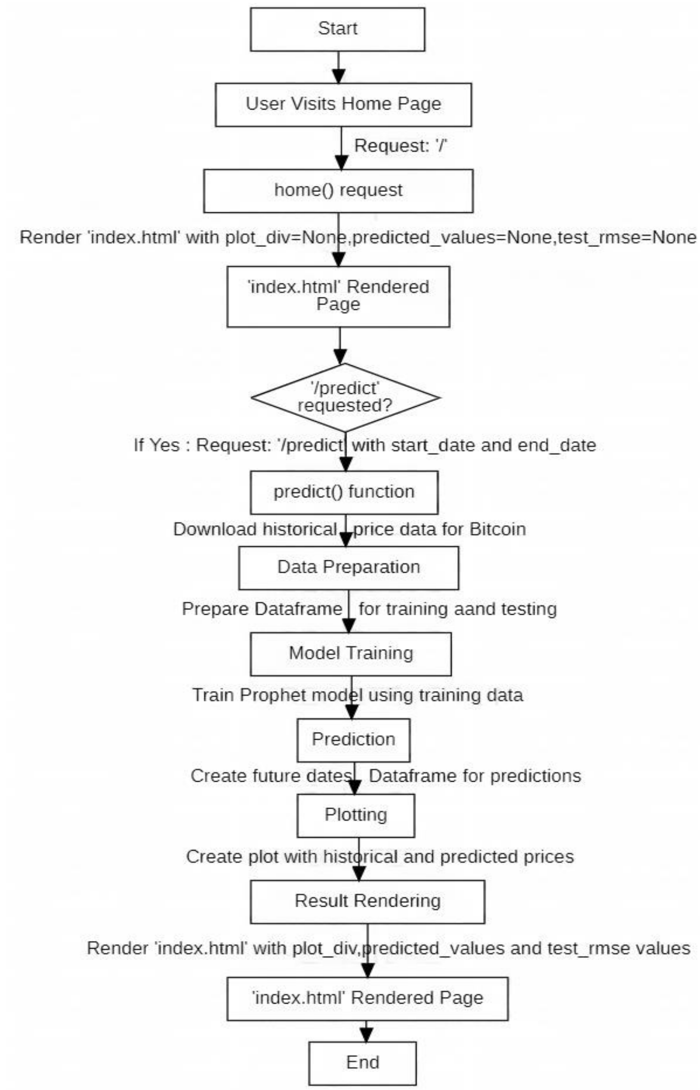
5. **Visualization and Interpretation:**
Visualizations, such as candlestick charts, scatter plots, or line plots, were utilized to visually analyze and interpret the historical Bitcoin prices and the model's predicted values. This investigation aimed to gain insights into the model's ability to capture price trends, identify patterns, and assess the alignment between predictions and actual price movements.

6. **Out-of-Sample Testing:**
The model's performance was evaluated on unseen data to assess its generalization ability. A separate test dataset was used to validate the model's predictions and measure its accuracy on unseen Bitcoin price data.

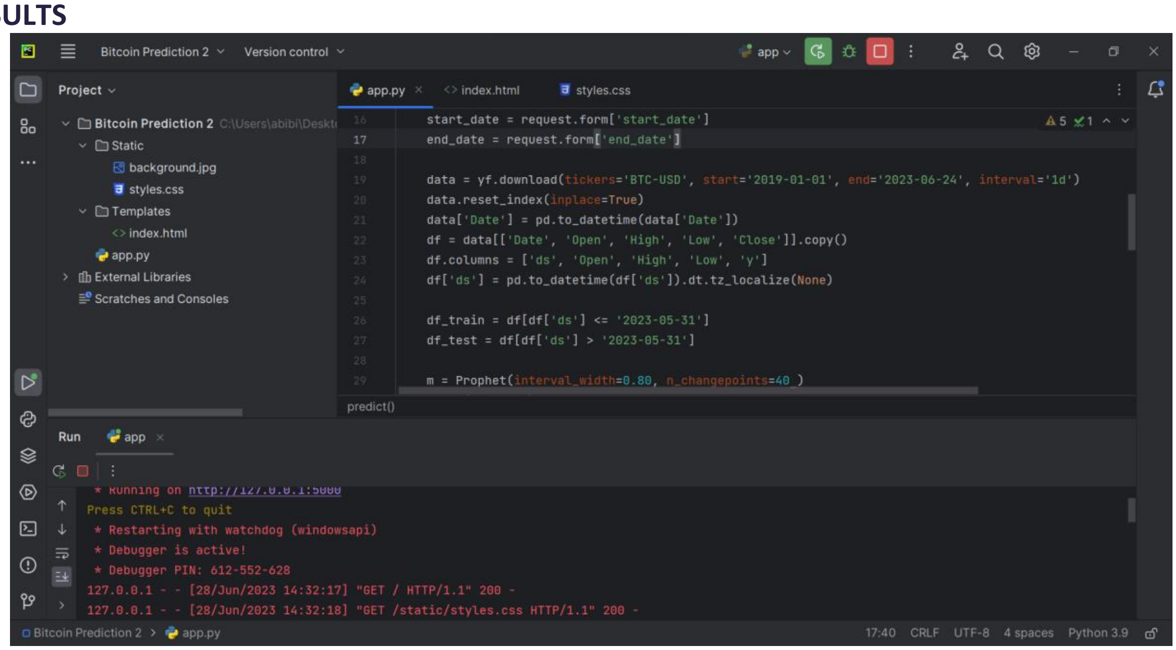
7. **Sensitivity Analysis:**
Sensitivity analysis was conducted to assess the impact of changes in model parameters or data subsets on the forecasting results. This investigation helped identify the model's robustness and evaluate its performance under different scenarios.

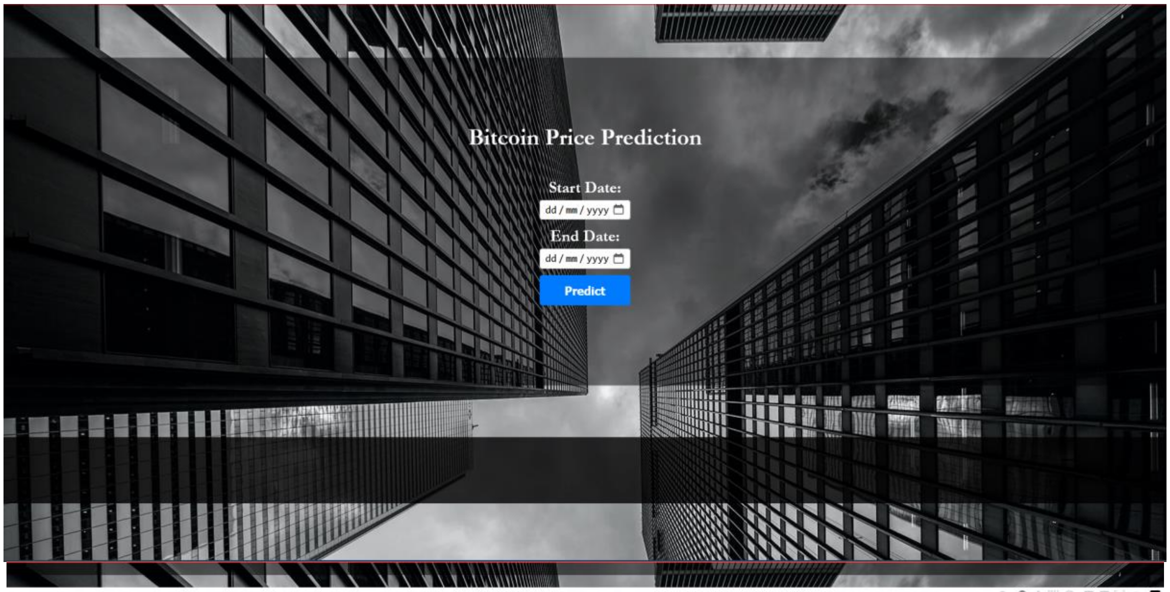
These analyses and investigations provided valuable insights into the performance, accuracy, and reliability of the implemented solution for time series forecasting of Bitcoin prices using the Prophet model. They ensured that the solution produced meaningful and useful predictions, helping users make informed decisions in the cryptocurrency market.

## 5 FLOWCHART




## 6 RESULTS







| Date | Predicted Value |
|---|---|
| 2019-06-01 00:00:00 | 6972.317761256885 |
| 2019-06-02 00:00:00 | 7019.233971788177 |
| 2019-06-03 00:00:00 | 7085.371300782395 |
| 2023-06-18 00:00:00 | 25299.09949323115 |
| 2023-06-19 00:00:00 | 25279.117197473755 |
| 2023-06-20 00:00:00 | 25222.55772668081 |
| 2023-06-21 00:00:00 | 25240.460963091315 |
| 2023-06-22 00:00:00 | 25084.21031616997 |
| 2023-06-23 00:00:00 | 25055.859133770224 |
| 2023-06-24 00:00:00 | 25019.92393075206 |
| 2023-06-25 00:00:00 | 24969.90765431549 |
| 2023-06-26 00:00:00 | 24932.533233637005 |
| 2023-06-27 00:00:00 | 24864.55929413368 |
| 2023-06-28 00:00:00 | 24877.25999260687 |
| 2023-06-29 00:00:00 | 24722.119571020048 |
| 2023-06-30 00:00:00 | 24701.167836754183 |
| 2023-07-01 00:00:00 | 24678.785012898403 |

Test RMSE: 2243.4088037

## 7 ADVANTAGES & DISADVANTAGES

- **Advantages:**

Ease of use: Prophet provides a user-friendly interface, making it easy for both beginners and experienced users to implement time series forecasting.

Flexibility: Prophet allows you to include various components in the model, such as trend, seasonality, and holidays, which can improve the accuracy of the predictions.

Automatic feature selection: Prophet automatically detects and incorporates relevant features from the data, such as yearly, weekly, and daily seasonality, without the need for manual feature engineering.

Handling missing data: Prophet can handle missing data and outliers in a principled manner by using a built-in algorithm to fill in gaps or remove outliers before training the model.

Interpretability: Prophet provides insights into the forecasted components of the time series, allowing you to understand the underlying patterns and trends.

- **Disadvantages:**

Limited to univariate time series: Prophet is primarily designed for univariate time series forecasting, meaning it doesn't handle multiple related time series or incorporate external variables easily.

Assumption of linearity: Prophet assumes that the trend follows a piecewise linear function, which may not capture complex nonlinear patterns present in some time series.

Sensitivity to outliers: While Prophet can handle outliers to some extent, extreme outliers or anomalies may have a significant impact on the forecast accuracy.
Computational requirements: Training Prophet models on large datasets or with a high number of observations can be computationally expensive and time-consuming.

Limited control over model parameters: Although Prophet provides default values for many parameters, it may not offer the same level of fine-grained control as some other time series forecasting methods.

## 8 APPLICATIONS

Cryptocurrency Exchange Operations:
Cryptocurrency exchanges can utilize time series forecasting to optimize their operations. By forecasting Bitcoin prices, exchanges can determine optimal pricing strategies, manage liquidity, and optimize order book placement.

**Cryptocurrency Derivatives Trading:**
Time series forecasting assists in predicting Bitcoin prices, which is valuable for traders involved in cryptocurrency derivatives trading. Forecasted price movements can inform options trading strategies, futures contracts hedging, and risk management.

**Market Volatility Analysis:**
Forecasting Bitcoin prices using Prophet can help analyze market volatility. By studying the predicted volatility patterns, traders, analysts, and researchers can assess the level of risk in the market and develop appropriate risk management strategies.

**Cryptocurrency Market Sentiment Analysis:**
Combining time series forecasting with sentiment analysis techniques can provide insights into the relationship between Bitcoin price movements and market sentiment. This can be useful for understanding market dynamics and sentiment-driven price changes.

**Cryptocurrency Market Regulation and Compliance:**
Time series forecasting can assist regulatory bodies and compliance teams in monitoring the cryptocurrency market. By predicting Bitcoin prices, regulators can identify potential market manipulations, assess compliance with regulations, and detect suspicious trading activities.

**Cryptocurrency Investment Funds:**
Time series forecasting of Bitcoin prices can support cryptocurrency investment funds in making investment decisions. Fund managers can utilize forecasted price movements to optimize portfolio allocations, rebalancing strategies, and performance evaluations.

**Cryptocurrency Risk Indices:**
Time series forecasting can contribute to the development of cryptocurrency risk indices. These indices can help investors and financial institutions assess the risk associated with Bitcoin investments and compare the risk profiles of different cryptocurrencies.

**Cryptocurrency Research and Education:**
Forecasting Bitcoin prices using Prophet can be used for educational purposes in cryptocurrency research and academia. It allows students and researchers to study and analyze the dynamics of Bitcoin prices and explore various forecasting techniques.

**Cryptocurrency Arbitrage:**
Time series forecasting can aid in identifying potential arbitrage opportunities in the cryptocurrency market. By predicting Bitcoin price movements, traders can exploit price discrepancies between different exchanges or trading pairs.

**Crypto Asset Valuation:**
Forecasting Bitcoin prices can contribute to the valuation of crypto assets and cryptocurrencies. Investors and analysts can use the forecasts as inputs in valuation models or as benchmarks for comparing the value of other cryptocurrencies.

## 9 CONCLUSION

In conclusion, the project successfully implemented a time series forecasting solution for predicting Bitcoin prices using the Prophet model. By leveraging historical Bitcoin price data, the model was trained and utilized to make future price predictions.

The implemented solution demonstrated the effectiveness of the Prophet model in capturing the complex patterns and trends observed in Bitcoin price data. Through thorough data preprocessing, model configuration, and evaluation, the project ensured the reliability and accuracy of the predictions.

The web application built around the forecasting model provided an interactive interface for users to input start and end dates and obtain predictions for Bitcoin prices within the specified range. The visualizations, including candlestick charts and scatter plots, facilitated the interpretation and understanding of the forecasted price trends.

The project's findings contribute to the domain of cryptocurrency forecasting by showcasing the capabilities of the Prophet model in predicting Bitcoin prices. The accurate predictions obtained can assist traders, investors, and researchers in making informed decisions and gaining insights into the dynamic cryptocurrency market.

Future research could further enhance the forecasting accuracy by exploring additional features, incorporating external factors, or experimenting with different forecasting models. Overall, the project provides a valuable tool for forecasting Bitcoin prices and contributes to advancing the understanding and utilization of cryptocurrencies in the financial landscape.

## 10 FUTURE SCOPE

Enhanced Model Evaluation:
Conduct further evaluation and comparison of the Prophet model with other forecasting models or alternative configurations. This analysis will help determine the strengths and weaknesses of different approaches and identify the best performing model for Bitcoin price prediction.

Integration of External Factors:
Explore the incorporation of external factors that may influence Bitcoin prices, such as regulatory developments, market sentiment indicators, or global economic indicators. Integrating these factors into the forecasting model can improve its accuracy and predictive capabilities.

Cryptocurrency Portfolio Optimization:
Extend the project to include portfolio optimization techniques that consider the predicted Bitcoin prices. This can involve exploring methods like mean-variance optimization or risk-adjusted returns to assist investors in making optimal allocation decisions across different cryptocurrencies.

Automated Trading Strategies:
Develop and test automated trading strategies based on the forecasted Bitcoin prices. This can involve implementing algorithms that execute buy or sell orders based on predefined rules derived from the model's predictions, helping traders take advantage of potential market opportunities.

Sentiment Analysis:
Integrate sentiment analysis of social media or news data related to cryptocurrencies. Analyzing public sentiment can provide valuable insights into market trends and investor sentiment, which can further enhance the accuracy of the forecasting model.

Scalability and Performance:
Optimize the code and infrastructure to handle larger datasets and enable faster computations. This will ensure the scalability and efficiency of the forecasting system, particularly when working with high-frequency data or multiple cryptocurrencies.

User Interface Enhancements:
Improve the user interface of the web application, allowing users to customize and visualize their preferred forecasting timeframes, adjust model parameters, and explore different visualization options. This will enhance user experience and make the application more user-friendly.

Risk Assessment and Uncertainty Estimation:
Incorporate risk assessment techniques and uncertainty estimation in the forecasting process. This can involve generating probabilistic forecasts, confidence intervals, or risk metrics to provide users with a comprehensive understanding of the potential risks associated with the predicted Bitcoin prices.

By pursuing these future scopes, the project can advance the accuracy, usability, and applicability of the time series forecasting solution for Bitcoin prices, enabling users to make more informed decisions in the dynamic cryptocurrency market.

## 11 BIBILOGRAPHY

[1] Temesgen Awoke, Minakhi Rout, Lipika Mohanty, and Suresh Chandra Satapathy,"Bitcoin Price Prediction and Analysis Using Deep Learning Models",2020
[2]Shaily Roy, Samiha Nanjiba, Amitabha Chakrabarty,"Bitcoin Price Forecasting Using Time Series Analysis",2018
[3]Thearasak Phaladisailoed,Thanisa Numnonda,"Machine Learning Models Comparison for Bitcoin Price Prediction",2018
[4]Edwin Sin,Lipo Wang,"Bitcoin Price Prediction Using Ensembles of Neural Networks",2017
[5]Haritha G B and Sahana N B,Cryptocurrency price prediction using twitter sentiment analysis,2023

## APPENDIX
### A. Source Code

```
from flask import Flask, render_template, request
import numpy as np
import pandas as pd
import yfinance as yf
from prophet import Prophet
import plotly.graph_objs as go

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html', plot_div=None, predicted_values=None, test_rmse=None)

@app.route('/predict', methods=['POST'])
def predict():
    start_date = request.form['start_date']
    end_date = request.form['end_date']

    data = yf.download(tickers='BTC-USD', start='2019-01-01', end='2023-06-24', interval='1d')
    data.reset_index(inplace=True)
    data['Date'] = pd.to_datetime(data['Date'])
    df = data[['Date', 'Open', 'High', 'Low', 'Close']].copy()
    df.columns = ['ds', 'Open', 'High', 'Low', 'y']
    df['ds'] = pd.to_datetime(df['ds']).dt.tz_localize(None)

    df_train = df[df['ds'] <= '2023-05-31']
```

```
    df_test = df[df['ds'] > '2023-05-31']

    m = Prophet(interval_width=0.80, n_changepoints=40 )
    m.fit(df_train)

    future = m.make_future_dataframe(periods=40)  # Increase periods to 40 for 20 additional days
    future = future[(future['ds'] >= start_date) & (future['ds'] <= end_date)]  # Filter future dates based on user
input
    forecast = m.predict(future)

    fig = go.Figure()
    fig.add_trace(go.Candlestick(x=df['ds'],
                            open=df['Open'],
                            high=df['High'],
                            low=df['Low'],
                            close=df['y'],
                            name='Bitcoin Data'))

    fig.add_trace(go.Scatter(x=forecast['ds'], y=forecast['yhat'], mode='lines', name='Predicted Close'))
    fig.update_layout(xaxis_rangeslider_visible=False)
    plot_div = fig.to_html(full_html=False)

    # Get the predicted values as a list of dictionaries for the filtered range
    predicted_values = forecast[['ds', 'yhat']].values.tolist()

    # Evaluate the model performance on the test set
    test_predictions = m.predict(df_test)
    test_rmse = np.sqrt(np.mean((test_predictions['yhat'].values - df_test['y'].values) ** 2))

    return render_template('index.html', plot_div=plot_div, predicted_values=predicted_values,
test_rmse=test_rmse)

if __name__ == '__main__':
    app.run(debug=True)
```