TITLE OF THE PROJECT :

Time Series Analysis For Bitcoin Price Prediction UsingProphet

1. Introduction

1.1 Project Overview

This project aims to predict the future price of Bitcoin (BTC) using a profit model. The project will use a variety of machine learning techniques to build amodel that can accurately predict the price of BTC. The project is expected to provide valuable insights into the future of BTC and help investors make informed decisions.

1.2 Purpose

The purpose of this project is to develop a reliable and accurate model for predicting the future price of BTC. The project is designed to address the need for better investment tools in the cryptocurrency market. The project is also intended to contribute to the growing body of knowledge about BTC and othercryptocurrencies.

The cryptocurrency market is a highly volatile market, and the price of BTC can fluctuate significantly in a short period of time. This volatility makes it difficult for investors to make informed decisions about when to buy and sell BTC. A reliable and accurate model for predicting the future price of BTC would be a valuable tool for investors. Additionally, such a model could be used to developnew trading strategies and to better understand the factors that affect the price of BTC.

This project will use a variety of machine learning techniques to build a modelfor predicting the future price of BTC. These techniques will include:

Support vector machines (SVMs): SVMs are a supervised learning algorithm that can be used to classify data. In this project, SVMs will be used to classify historical BTC price data into different categories, such as "upward trend" and"downward trend."

Random forests: Random forests are an ensemble learning algorithm that combines multiple decision trees. In this project, random forests will be used

to predict the future price of BTC based on a variety of factors, such as thehistorical price of BTC, the current market conditions, and news events.

Neural networks: Neural networks are a type of machine learning algorithmthat is inspired by the structure of the human brain. In this project, neural networks will be used to learn complex patterns in historical BTC price dataand to make predictions about the future price of BTC.

The project will be divided into three phases:

Data collection phase: In this phase, historical BTC price data will be collectedfrom a variety of sources.

Model development phase: In this phase, machine learning models will bedeveloped using the collected data.

Model evaluation phase: In this phase, the developed models will be evaluatedon their ability to predict the future price of BTC.

The project is expected to provide valuable insights into the future of BTC and to help investors make informed decisions. Additionally, the project is intendedto contribute to the growing body of knowledge about BTC and other cryptocurrencies.

2. LITERATURE SURVEY

2.1 Existing problem

While the use of time series analysis for Bitcoin price prediction using Prophetor other forecasting methods is a popular and interesting topic, there are several challenges and issues associated with it. It's important to note that thecryptocurrency market, including Bitcoin, is highly volatile and influenced by avariety of factors, making accurate predictions challenging. Here are some existing problems and challenges associated with time series analysis for

Bitcoin price prediction using Prophet or similar

methods:Volatility and Non-Linearity:

Cryptocurrency prices, including Bitcoin, are known for their high volatility. Themarket can be influenced by a wide range of factors, including regulatory changes, macroeconomic trends, market sentiment, and technological developments. The non-linear and unpredictable nature of these factors makes

it challenging to capture and model using traditional time series analysistechniques.

Data Quality and Noise:

Cryptocurrency markets can be susceptible to noise and manipulation. Low liquidity in some markets, coupled with the potential for sudden and unexpected events, can introduce noise into historical price data. Cleaning andpreprocessing data to remove outliers and noise is a critical step, and the quality of historical data can significantly impact the accuracy of predictions.

Changing Market Dynamics:

Cryptocurrency markets are relatively new and evolving rapidly. The dynamicsof the market can change over time due to factors such as increased adoption,regulatory changes, technological advancements, and shifts in investor sentiment. Models trained on historical data may struggle to adapt to these changing market conditions.

Overfitting and Model Complexity:

Overfitting occurs when a model is trained too closely on historical data, capturing noise rather than the underlying patterns. Choosing an appropriatelevel of model complexity and regularization is crucial to prevent overfitting and ensure the model generalizes well to new data.

External Factors:

Cryptocurrency markets are influenced by external factors such as regulatorydevelopments, security concerns, and macroeconomic trends. These factors are often difficult to incorporate into traditional time series models, and unexpected events can have a significant impact on prices.

Limited Historical Data:

The limited history of Bitcoin and other cryptocurrencies can be a constraintfor time series analysis. Traditional time series models may require a longer history to identify robust patterns, and the relatively short lifespan of cryptocurrencies makes long-term predictions more challenging.

Model Sensitivity to Parameters:

Models like Prophet have hyperparameters that need to be tuned appropriately for optimal performance. The sensitivity of these models to parameter choices can impact the accuracy of predictions, and finding the rightset of parameters can be a non-trivial task.

2.2 References

Bitcoin Forecasting Using ARIMA and PROPHET: Bitcoin Forecasting UsingARIMA and PROPHET | IEEE Conference Publication | IEEE Xplore

Univariate Time Series Analysis of Cryptocurrency Data using Prophet: Univariate Time Series Analysis of Cryptocurrency Data using Prophet, LSTMand XGBoost (ijraset.com)

GitHub - prasoon2510/Bitcoin_Price: GitHub - prasoon2510/Bitcoin_Price

A novel cryptocurrency price time series hybrid prediction model via ...: A novelcryptocurrency price time series hybrid prediction model via machine learning with MATLAB/Simulink | SpringerLink

EAI Endorsed Transactions - Time-Series Prediction of Cryptocurrency Marketusing Machine Learning Techniques (eudl.eu)

bitcoin-price-prediction · GitHub Topics · GitHub: bitcoin-price-prediction · GitHub Topics · GitHub

2.3 Problem Statement Definition

The volatility and unpredictability of Bitcoin prices in the cryptocurrencymarket pose significant challenges for investors, traders, and analysts.
Traditional financial markets often leverage time series analysis techniques toforecast future prices and trends. However, the unique characteristics of the cryptocurrency market, particularly Bitcoin, require specialized approaches.
This study aims to employ time series analysis, specifically utilizing the Prophetforecasting tool, to address the following challenges in Bitcoin price prediction:

Volatility and Non-Linearity: Develop a robust time series model capable ofcapturing the non-linear and volatile nature of Bitcoin price movements,

considering factors such as market sentiment, regulatory changes, andmacroeconomic trends.

Data Quality and Noise: Investigate methods for preprocessing historical Bitcoin price data to mitigate the impact of noise, outliers, and potential manipulation, ensuring the accuracy and reliability of the forecasting model.

Changing Market Dynamics: Explore the adaptability of time series models to changing market conditions in the cryptocurrency space. Evaluate the model'sability to identify and respond to shifts in investor sentiment, technological developments, and other dynamic factors.

Overfitting and Model Complexity: Assess the risk of overfitting in time series models and determine optimal levels of model complexity and regularization to enhance generalization to unseen data and improve prediction accuracy.

External Factors: Examine the incorporation of external factors, such as regulatory changes and macroeconomic indicators, into the time series analysis to enhance the model's predictive power and broaden its applicability.

Limited Historical Data: Investigate the impact of the relatively short historicaldata available for Bitcoin on the accuracy of time series models. Explore strategies to overcome the challenges associated with limited data for long- term price predictions.

Model Sensitivity to Parameters: Analyze the sensitivity of the Prophet forecasting tool to its hyperparameters and explore parameter tuning strategies to optimize the model's performance in predicting Bitcoin prices.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

To better understand the needs and challenges faced by investors in the cryptocurrency market, we created an empathy map canvas. The empathy map canvas is a tool used to visualize and understand the thoughts, feelings,and behaviors of a particular group of people.

In this case, we focused on the needs and challenges of investors who aretrying to make informed decisions about when to buy and sell BTC. We identified the following key points:

Needs:

Accurate and reliable price predictions

Insights into factors affecting BTC price

Tools to develop effective trading

strategiesAccess to a user-friendly

platform Challenges:

Volatility of the cryptocurrency market

Difficulty in identifying patterns in historical

dataLack of reliable investment tools

Information overload from various

sourcesPain Points:

Making investment decisions based on fear or

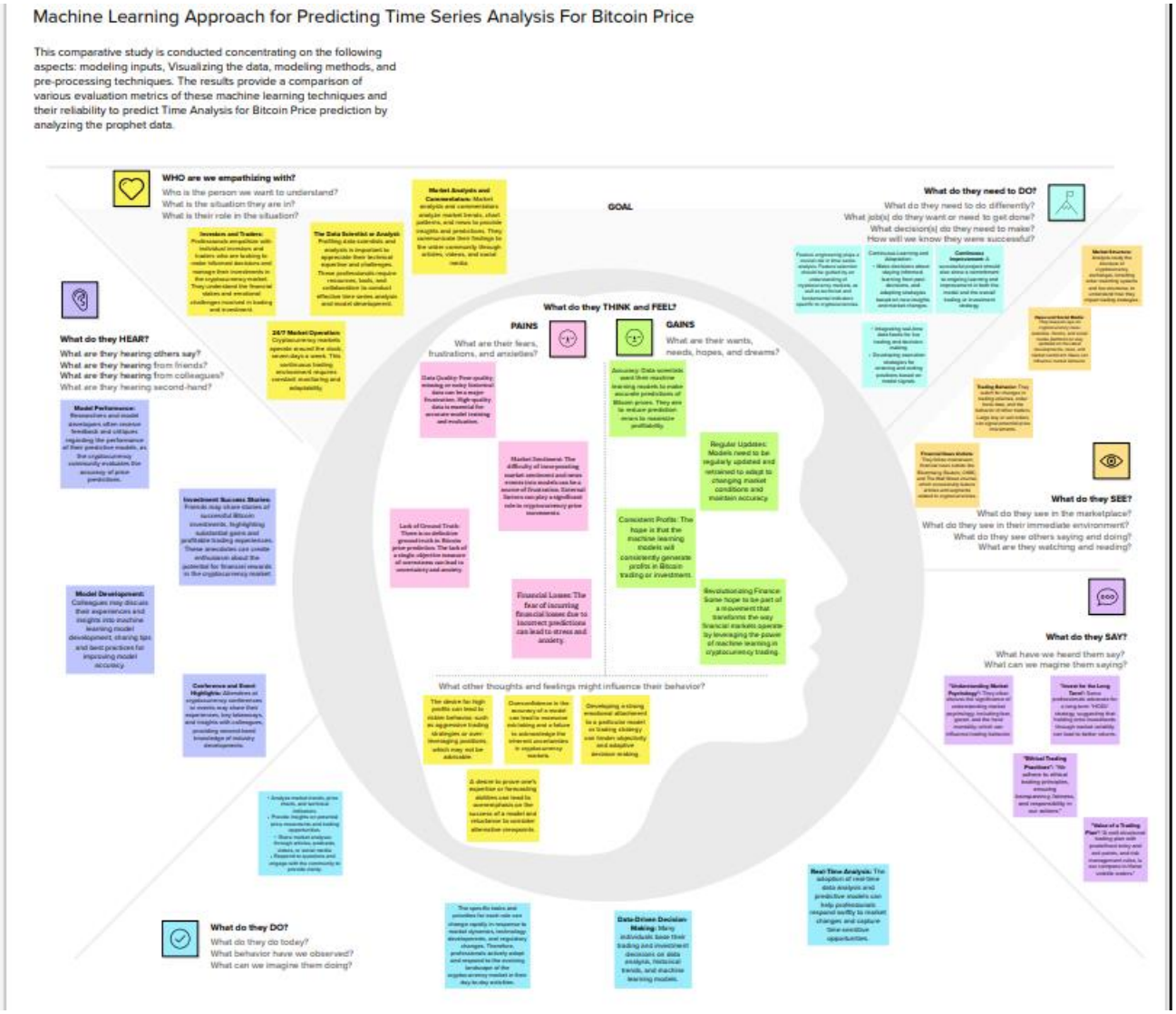
emotionsExperiencing losses due to misinformed

trading

Missing out on potential profits due to missed opportunities

Feeling overwhelmed by the vast amount of information

available



Link to our mural :
https://app.mural.co/t/pranay9391/m/pranay9391/1699192930524/b8aecb07605e6d4a91ec89cd8da26ce777f9de27?sender=udbc24384af8e6eaf8bfd4310

3.2 Ideation & Brainstorming

Based on the insights gathered from the empathy map canvas, we conducted an ideation and brainstorming session to generate potential solutions for addressing the needs and challenges of BTC investors. We encouraged participants to think creatively and come up with innovative ideas.

Here are some of the key ideas that emerged from the brainstorming session:

Develop a multi-algorithm prediction model: We propose combining multiplemachine learning algorithms, such as SVMs, random forests, and neural networks, to create a more robust and accurate prediction model. This approach would leverage the strengths of each algorithm and reduce the overall error rate.

Incorporate fundamental analysis factors: In addition to historical price data,we suggest incorporating fundamental analysis factors, such as news events,market sentiment, and economic indicators, into the prediction model. This would provide a more comprehensive understanding of the factors affectingBTC price and improve prediction accuracy.

Create a user-friendly prediction platform: We envision a user-friendly platform that provides investors with real-time price predictions, risk assessments, and potential trading strategies. This platform would simplify thedecision-making process and empower investors to make informed trades.

Offer educational resources and tutorials: We propose providing educationalresources and tutorials to help investors gain a better understanding of cryptocurrency market dynamics, technical analysis, and risk management strategies. This would enhance their overall knowledge and decision-making capabilities.

Establish a community forum and support system: We suggest creating a community forum and support system where investors can share insights, discuss trading strategies, and seek guidance from experienced traders. Thiswould foster a collaborative learning environment and provide valuable support for new investors.

By implementing these proposed solutions, we aim to address the needs andchallenges faced by BTC investors, empowering them to make informed decisions and achieve their investment goals.

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Gandla Vishnu Kailash
Narayana Reddy Gari Siddharth Reddy
Jakka Subramanya Rithwik
Yamasani Purushotham Reddy

B **Set the goal**
Time Series Analysis For Bitcoin Price Prediction Using Prophet

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

PR

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

The cryptocurrency market, particularly Bitcoin, is known for its high volatility and dynamic nature. Traders, investors, and financial analysts often face challenges in accurately predicting the future price movements of Bitcoin. Traditional financial models struggle to capture the intricate patterns and sudden shifts in the cryptocurrency market, making it crucial to explore advanced techniques for forecasting.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Defer judgment.
- Go for volume.
- Encourage wild ideas.
- Listen to others.
- If possible, be visual.

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

Who:

Traders, investors, and cryptocurrency enthusiasts

What:

Time series analysis for Bitcoin Price Prediction

When:

The project can be ongoing, continuously updating the model with new data to make short-term and long-term predictions.

Where:

The predictions are applicable in the context of the cryptocurrency market, which is accessible online through various cryptocurrency exchanges.

Why:

- accurate price predictions can help traders make profitable decisions, mitigate risks, and time their trades effectively.
- investors can use these predictions to assess potential long-term value and make informed decisions about buying and holding Bitcoin.

PROBLEM

"Develop a time series analysis model for accurate short-term Bitcoin price prediction, leveraging historical price data and relevant external factors, to help cryptocurrency traders and investors make informed decisions in a highly volatile market."

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Person 1	Person 2	common idea	Person 3
Data collection and preprocessing	**Feature Engineering and Selection**	**Model Development and Training**	**Evaluation and Deployment**
Focus on collecting relevant datasets	Work on feature engineering, selecting the most informative variables for prediction	Build and train the machine learning models for Time Series analysis for Bitcoin Price prediction	Focus on evaluating model performance, conducting testing, and preparing the model for deployment
Ensure data quality, and clean, preprocess, and format the data for ML model input	involves identifying relevant features, handling missing data, and creating new variables if needed	Explore various ML algorithms such as logistic regression, decision trees, random forests, support vector machines or neural networks	consider ethical and regulatory considerations for deploying a Time Series analysis for Bitcoin Price prediction

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

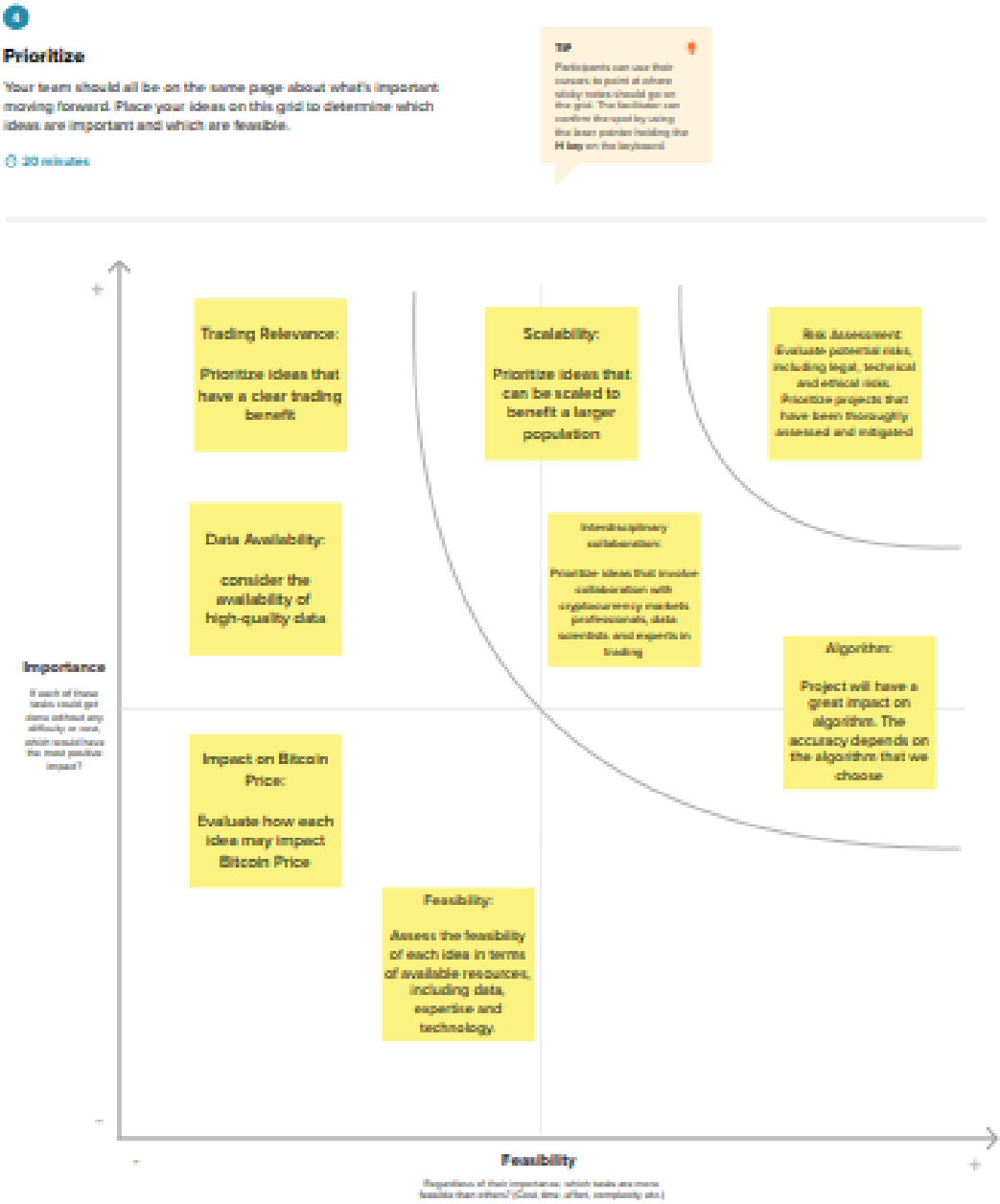
TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Project Management and collaboration

Documentation and Reporting

Quality Assurance and Ethical Considerations



4. Requirement Analysis

4.1 Functional Requirements

Functional requirements define the specific functions and capabilities that the prediction system must provide to meet user needs. These requirements focuson what the system should do rather than how it should do it.

Core Functional Requirements:

Accurate and Reliable Price Predictions: The system should generate accurateand reliable predictions of the future price of BTC.

Real-time Price Updates: The system should provide real-time updates on thecurrent price of BTC.

Historical Price Data Access: The system should allow users to access andanalyze historical BTC price data.

Risk Assessment and Analysis: The system should provide risk assessment toolsto help users evaluate the potential risks associated with BTC trading.

Potential Trading Strategy Generation**:** The system should suggest potentialtrading strategies based on prediction models and risk assessments.

User-friendly Interface: The system should provide a user-friendly interfacethat is easy to navigate and understand, even for non-technical users.

Additional Functional Requirements:

Customizable Prediction Settings**:** Users should be able to customize predictionsettings, such as time horizons and risk tolerance.

Performance Tracking: The system should track the performance of predictionmodels over time and provide insights into their accuracy and reliability.

Multiple Algorithm Support: The system should support multiple machine learning algorithms for prediction, allowing users to select the algorithm thatbest suits their needs.

Visualization and Reporting Tools: The system should provide visualization andreporting tools to help users analyze historical data and understand price trends.

4.2 Non-Functional Requirements

Non-functional requirements define the overall characteristics and performance expectations of the prediction system. These requirements focuson how the system should behave rather than what it should do.

Performance Requirements:

Scalability: The system should be scalable to handle increasing data volumesand user traffic.

Responsiveness: The system should provide quick response times and real-time updates.

Accuracy: The prediction accuracy should be consistently high and withinacceptable error margins.

Reliability: The system should be reliable and operate consistently withoutfrequent downtime or errors.

Security Requirements:

Data Security: User data and historical price data should be protected fromunauthorized access and breaches.

Transaction Security: Trading transactions should be secure and protectedfrom fraud or manipulation.

Privacy Protection: User privacy should be respected, and personal data shouldnot be shared without explicit consent.

Usability Requirements:

Intuitive Interface: The interface should be intuitive, easy to navigate, andunderstandable for users with varying levels of technical expertise.

Clear Instructions and Documentation: Clear instructions, documentation, and tutorials should be provided to guide users through the system's functionalities.

Accessibility: The system should be accessible to users with disabilities andprovide alternative input and output methods.

Maintenance Requirements:

Easy Updateability: The system should be easy to update with new featuresand bug fixes.

Log Monitoring: The system should maintain comprehensive logs for monitoring performance, identifying errors, and facilitating troubleshooting.

Documentation and Support: Comprehensive documentation and supportshould be provided for system administrators and users.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

Data Collection: This phase entails gathering a diverse dataset of Bitcoin from sources such as articles, or other means. The collected data are then stored in a raw data repository.

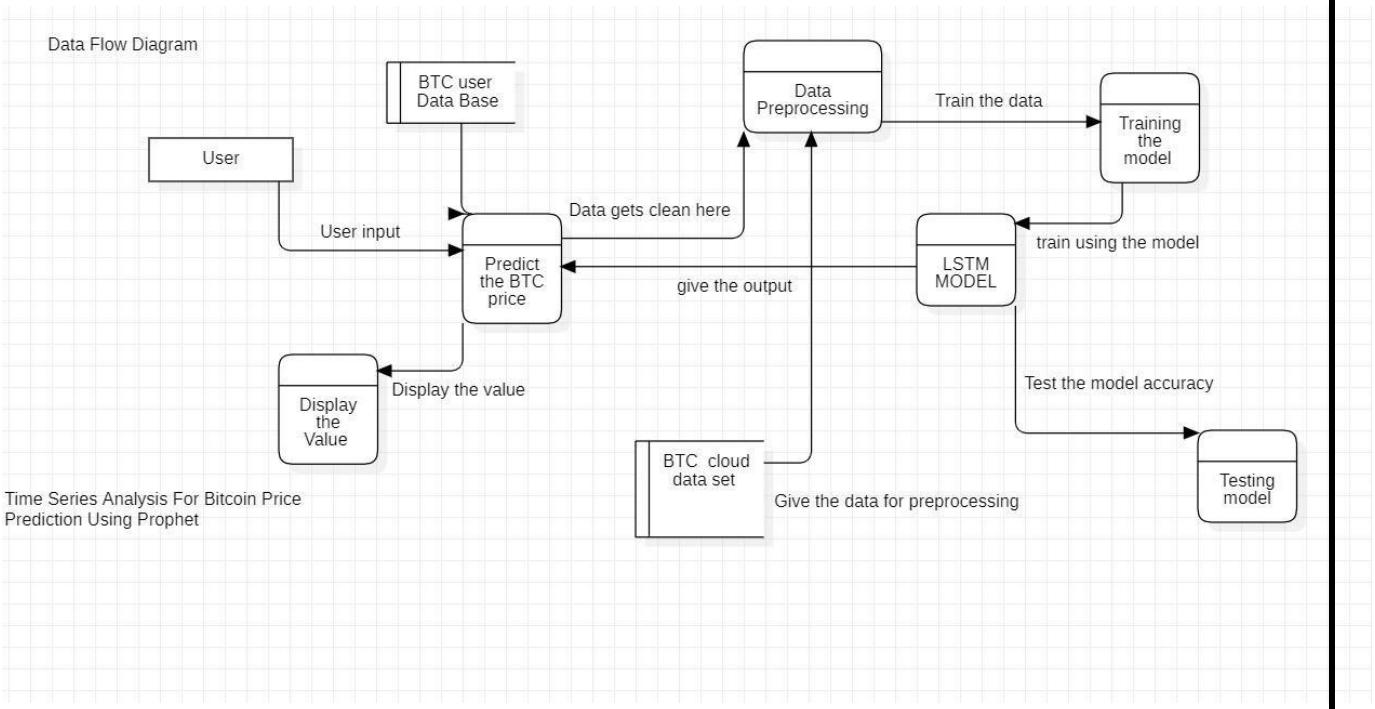
Data Pre-processing: Raw data are pre-processed to prepare them for modeltraining. This can include removing nulls, and applying data augmentation techniques to enhance dataset diversity.

Model Training: In this stage, the pre-processed data is utilized to train a deep learning model, which learns to classify Bitcoin prediction. The trained model issaved for future use.

Model Evaluation: The performance of the trained model is assessed using aseparate dataset not used in training, to gauge its accuracy, sensitivity, and specificity in Bitcoin prediction.

Model Deployment: This step involves deploying the trained model to make itaccessible for Bitcoin prediction classification applications, either on local devices or in the cloud.

User Interaction: End-users interact with the deployed model through a user-friendly application or API.



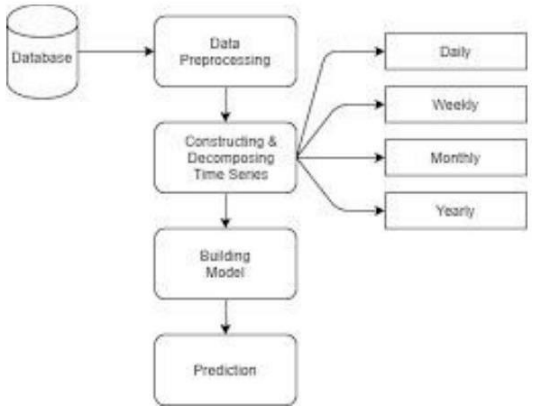
User Stories

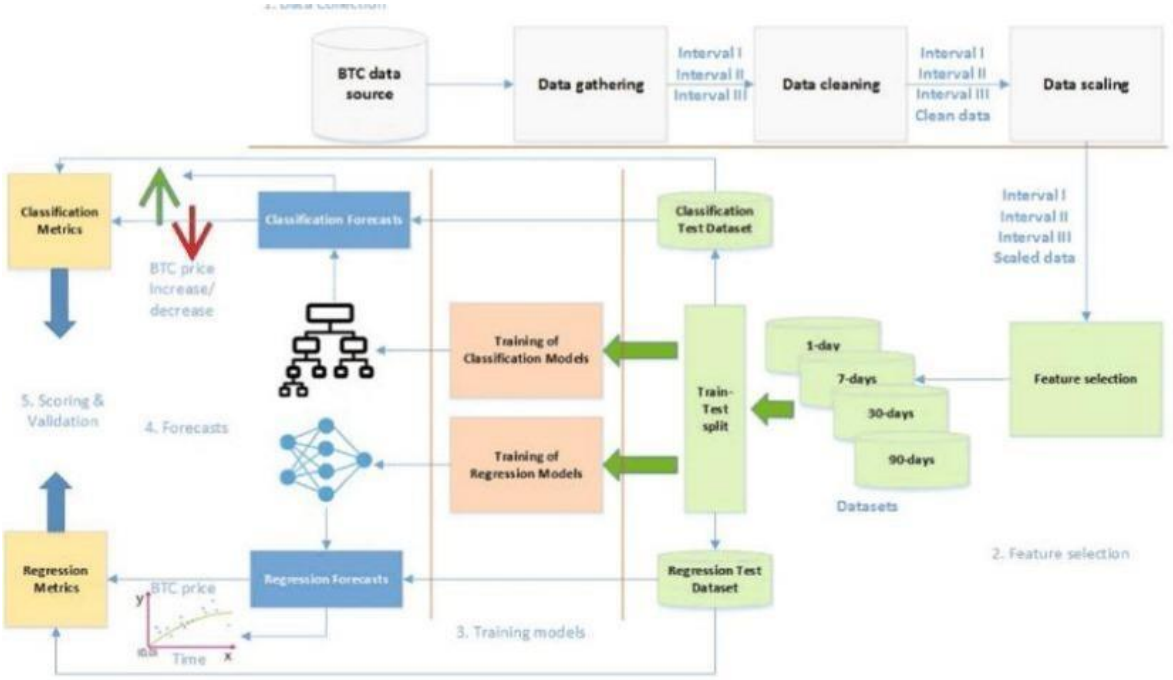
Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration Required	USN-1	As a user,I can visit the website and get to know about the current price of that bitcoin	I can access the current price of the bitcoin	High	Sprint-1
Customer	Registration Required	USN-2	As a user, I can log into the application by entering email & password	I can access the currentprice of the bitcoin	Medium	Sprint-1
	Login	USN-3	As a trader, I want the website to provide additional market data and analysis related to Bitcoin's price to help me make informed decisions.	Integration with news or analysis sections related to Bitcoin's market trends and updates.	low	Sprint 2
	Login	USN-4	As an investor, I want to see the historical price trends of Bitcoin on the website, allowing me to analyze its performance over time.	The website should include a graph or chart displaying the historical price trends of Bitcoin over different timeframes (day, week, month, year).	low	Sprint 3
Customer (Web user)						
Customer Care Executive						
Administrator						

Solution Architecture

Solution Architecture Diagram:





6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Python
6.	Cloud Database	Type of dataframe taken using yticker	Took by using yticker
7.	File Storage	Stored in colab environment	Colab
10.	Machine Learning Model	Time series forecasting with Prophet	Facebook Prophet library
11.	Infrastructure (Server / Cloud)	Application Deployment Cloud Local Server	Colab

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Python's Flask
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryption
3.	Scalable Architecture	Justify the scalability of architecture	Yticker, tensorflow, matplotlib, datetime
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Evaluation metrics to check performace	MAE, MSE, R2, RMSE

6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story	Number User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration Required	USN-1	As a user,I can visit the website and get to know about the current price of that bitcoin	8	High	AKILI HARSHA VARDHAN
Sprint-1	Registration Required	USN-2	As a user, I can log into the application by entering email & password	5	Medium	GOLLA PRANAY
Sprint-2	Login	USN-3	As a trader, I want the website to provide additional market data and analysis	7	Low	K. LOKESH REDDY

			related to Bitcoin's price to help me make informed decisions.			
Sprint-3	Login	USN-4	As an investor, I want to see the historical price trends of Bitcoin on the website, allowing me to analyze its performan ce over time.	5	Low	K.LOKESH REDDY

6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	5	3 Days	04 Oct 2023	06 Nov 2023	5	06 Nov 2023
Sprint-2	5	3 Days	07 Oct 2023	09 Nov 2023	5	09 Nov 2023
Sprint-3	5	4 Days	10 Nov 2023	13 Nov 2023	5	13 Nov 2023
Sprint-4	10	6 Days	14 Nov 2023	19 Nov 2023	10	19 Nov 2023

7. CODING & SOLUTIONING

Data Preprocessing:

```
import pandas as pd
import yfinance as yf
from datetime import datetime
from datetime import timedelta
import plotly.graph_objects as go
from prophet import Prophet
from prophet.plot import plot_plotly,plot_components_plotly
import warnings

warnings.filterwarnings('ignore')

pd.options.display.float_format='${:,.2f}'.format
```

1] ✓ 2.1s Python

```
c:\Users\akhil\AppData\Local\Programs\Python\Python312\Lib\site-packages\tqdm\auto.py:21: TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/s
  from .autonotebook import tqdm as notebook_tqdm
```

```
# import dataset

today=datetime.today().strftime('%Y-%m-%d')
start_date='2016-01-01'

df=yf.download('BTC-USD',start_date,today)
```

2] ✓ 1.6s P

```
[*********************100%%**********************]  1 of 1 completed
```

C: > Users > akhil > Desktop > Bit coin price prediction using FB Prophet_Harsha.ipynb > df.head()

+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ... Python 3.12.0

```
df
```

[3] ✓ 0.0s Python

Date	Open	High	Low	Close	Adj Close	Volume
2016-01-01	$430.72	$436.25	$427.52	$434.33	$434.33	36278900
2016-01-02	$434.62	$436.06	$431.87	$433.44	$433.44	30096600
2016-01-03	$433.58	$433.74	$424.71	$430.01	$430.01	39633800
2016-01-04	$430.06	$434.52	$429.08	$433.09	$433.09	38477500
2016-01-05	$433.07	$434.18	$429.68	$431.96	$431.96	34522600
...	...	...	...	...	...	...
2023-11-17	$36,164.82	$36,704.48	$35,901.23	$36,596.68	$36,596.68	22445028430
2023-11-18	$36,625.37	$36,839.28	$36,233.31	$36,585.70	$36,585.70	11886022717
2023-11-19	$36,585.77	$37,509.36	$36,414.60	$37,386.55	$37,386.55	12915986553
2023-11-20	$37,374.07	$37,756.82	$36,882.53	$37,476.96	$37,476.96	20888209068
2023-11-21	$37,469.16	$37,631.14	$35,813.81	$35,813.81	$35,813.81	25172163756

2882 rows × 6 columns

```
df.head()
```

[4] ✓ 0.0s Python

Date	Open	High	Low	Close	Adj Close	Volume
2016-01-01	$430.72	$436.25	$427.52	$434.33	$434.33	36278900
2016-01-02	$434.62	$436.06	$431.87	$433.44	$433.44	30096600
2016-01-03	$433.58	$433.74	$424.71	$430.01	$430.01	39633800
2016-01-04	$430.06	$434.52	$429.08	$433.09	$433.09	38477500
2016-01-05	$433.07	$434.18	$429.68	$431.96	$431.96	34522600

```
PS C:\Users\akhil> python -u "c:\Users\akhil\Desktop\fe.py"
[*********************100%%**********************]  1 of 1 completed
22:35:59 - cmdstanpy - INFO - Chain [1] start processing
22:36:00 - cmdstanpy - INFO - Chain [1] done processing
[*********************100%%**********************]  1 of 1 completed
36508.8984375 36904.13838197588
PS C:\Users\akhil> python -u "c:\Users\akhil\Desktop\fe.py"
[*********************100%%**********************]  1 of 1 completed
22:36:22 - cmdstanpy - INFO - Chain [1] start processing
22:36:22 - cmdstanpy - INFO - Chain [1] done processing
[*********************100%%**********************]  1 of 1 completed
36524.9609375 36904.13838197588
PS C:\Users\akhil>
```

```
df.tail()
```

[5] ✓ 0.0s

Date	Open	High	Low	Close	Adj Close	Volume
2023-11-17	$36,164.82	$36,704.48	$35,901.23	$36,596.68	$36,596.68	22445028430
2023-11-18	$36,625.37	$36,839.28	$36,233.31	$36,585.70	$36,585.70	11886022717
2023-11-19	$36,585.77	$37,509.36	$36,414.60	$37,386.55	$37,386.55	12915986553
2023-11-20	$37,374.07	$37,756.82	$36,882.53	$37,476.96	$37,476.96	20888209068
2023-11-21	$37,469.16	$37,631.14	$35,813.81	$35,813.81	$35,813.81	25172163756

```
df.describe()
```

[6] ✓ 0.0s

	Open	High	Low	Close	Adj Close	Volume
count	$2,882.00	$2,882.00	$2,882.00	$2,882.00	$2,882.00	$2,882.00
mean	$16,535.11	$16,924.96	$16,112.79	$16,545.82	$16,545.82	$19,170,779,928.23
std	$16,165.23	$16,559.96	$15,716.88	$16,163.12	$16,163.12	$19,397,147,454.42
min	$365.07	$374.95	$354.91	$364.33	$364.33	$28,514,000.00
25%	$4,071.54	$4,137.98	$3,988.74	$4,074.10	$4,074.10	$3,707,515,000.00
50%	$9,533.77	$9,708.22	$9,341.73	$9,537.46	$9,537.46	$15,724,882,633.50
75%	$26,826.20	$27,225.60	$26,466.69	$26,846.32	$26,846.32	$29,853,300,820.00
max	$67,549.73	$68,789.62	$66,382.06	$67,566.83	$67,566.83	$350,967,941,479.00

```
df.info()
```

✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2882 entries, 2016-01-01 to 2023-11-21
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Open       2882 non-null   float64
 1   High       2882 non-null   float64
 2   Low        2882 non-null   float64
 3   Close      2882 non-null   float64
 4   Adj Close  2882 non-null   float64
 5   Volume     2882 non-null   int64
dtypes: float64(5), int64(1)
memory usage: 157.6 KB
```

```
df.isnull().any()
```

✓ 0.0s Python

```
Open         False
High         False
Low          False
Close        False
Adj Close    False
Volume       False
dtype: bool
```

```
df.isnull().sum()
```

✓ 0.0s Python

```
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

```
df.reset_index(inplace=True)
df.columns
```

✓ 0.0s Python

```
Index(['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
```

Building the model:

```
# model 1

model=Prophet(
    seasonality_mode="multiplicative"
)

model.fit(df1)
```

✓ 1.9s

```
23:11:30 - cmdstanpy - INFO - Chain [1] start processing
23:11:32 - cmdstanpy - INFO - Chain [1] done processing

<prophet.forecaster.Prophet at 0x299021265a0>
```

```
future=model.make_future_dataframe(periods=365)
future.tail()
```

✓ 0.0s

	ds
3242	2024-11-16
3243	2024-11-17
3244	2024-11-18
3245	2024-11-19
3246	2024-11-20

```
forecast=model.predict(future)
forecast
```

✓ 0.6s

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	multiplicative_terms	multiplicative_terms_lower	multiplicative_terms_upper	weekly	weekly_lower	weekly_upper	yearly	yearly_lower	yearly_upper	additive_terms	additive_terms_lower	additive_terms_upper	yhat
0	2016-01-01	$3.95	$-5,153.50	$4,897.84	$3.95	$3.95	$-0.00	$-0.00	$-0.00	$-0.00	$-0.00	$-0.00	$-0.00	$-0.00	$-0.00	$0.00	$0.00	$0.00	$3.94
1	2016-01-02	$6.71	$-4,449.42	$4,821.31	$6.71	$6.71	$0.00	$0.00	$0.00	$-0.00	$-0.00	$-0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$6.73
2	2016-01-03	$9.47	$-4,774.68	$4,928.37	$9.47	$9.47	$0.01	$0.01	$0.01	$-0.00	$-0.00	$-0.00	$0.01	$0.01	$0.01	$0.00	$0.00	$0.00	$9.55
3	2016-01-04	$12.23	$-4,956.14	$4,897.45	$12.23	$12.23	$0.01	$0.01	$0.01	$0.00	$0.00	$0.00	$0.01	$0.01	$0.01	$0.00	$0.00	$0.00	$12.39
4	2016-01-05	$14.99	$-4,845.47	$4,722.94	$14.99	$14.99	$0.02	$0.02	$0.02	$0.00	$0.00	$0.00	$0.01	$0.01	$0.01	$0.00	$0.00	$0.00	$15.22
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3242	2024-11-16	$27,157.53	$6,357.75	$50,063.33	$6,780.55	$46,475.07	$0.07	$0.07	$0.07	$-0.00	$-0.00	$-0.00	$0.07	$0.07	$0.07	$0.00	$0.00	$0.00	$28,935.01
3243	2024-11-17	$27,159.13	$6,097.82	$48,934.31	$6,627.67	$46,571.00	$0.06	$0.06	$0.06	$-0.00	$-0.00	$-0.00	$0.06	$0.06	$0.06	$0.00	$0.00	$0.00	$28,843.15
3244	2024-11-18	$27,160.74	$5,596.74	$49,792.28	$6,567.90	$46,661.08	$0.06	$0.06	$0.06	$0.00	$0.00	$0.00	$0.06	$0.06	$0.06	$0.00	$0.00	$0.00	$28,756.65
3245	2024-11-19	$27,162.34	$6,881.23	$49,784.96	$6,497.00	$46,792.81	$0.05	$0.05	$0.05	$0.00	$0.00	$0.00	$0.05	$0.05	$0.05	$0.00	$0.00	$0.00	$28,627.96
3246	2024-11-20	$27,163.95	$6,140.84	$50,485.60	$6,431.83	$46,930.12	$0.05	$0.05	$0.05	$-0.00	$-0.00	$-0.00	$0.05	$0.05	$0.05	$0.00	$0.00	$0.00	$28,441.48

3247 rows × 19 columns

```
forecast[['ds','yhat','yhat_lower','yhat_upper']].tail()
```

✓ 0.0s

	ds	yhat	yhat_lower	yhat_upper
3242	2024-11-16	$28,935.01	$6,357.75	$50,063.33
3243	2024-11-17	$28,843.15	$6,097.82	$48,934.31
3244	2024-11-18	$28,756.65	$5,596.74	$49,792.28
3245	2024-11-19	$28,627.96	$6,881.23	$49,784.96
3246	2024-11-20	$28,441.48	$6,140.84	$50,485.60

```
next_day=(datetime.today()+timedelta(days=1)).strftime('%Y-%m-%d')

print(f"Predicted Bitcoin price for {next_day} is {forecast[forecast['ds']==next_day]['yhat'].item()}")
```

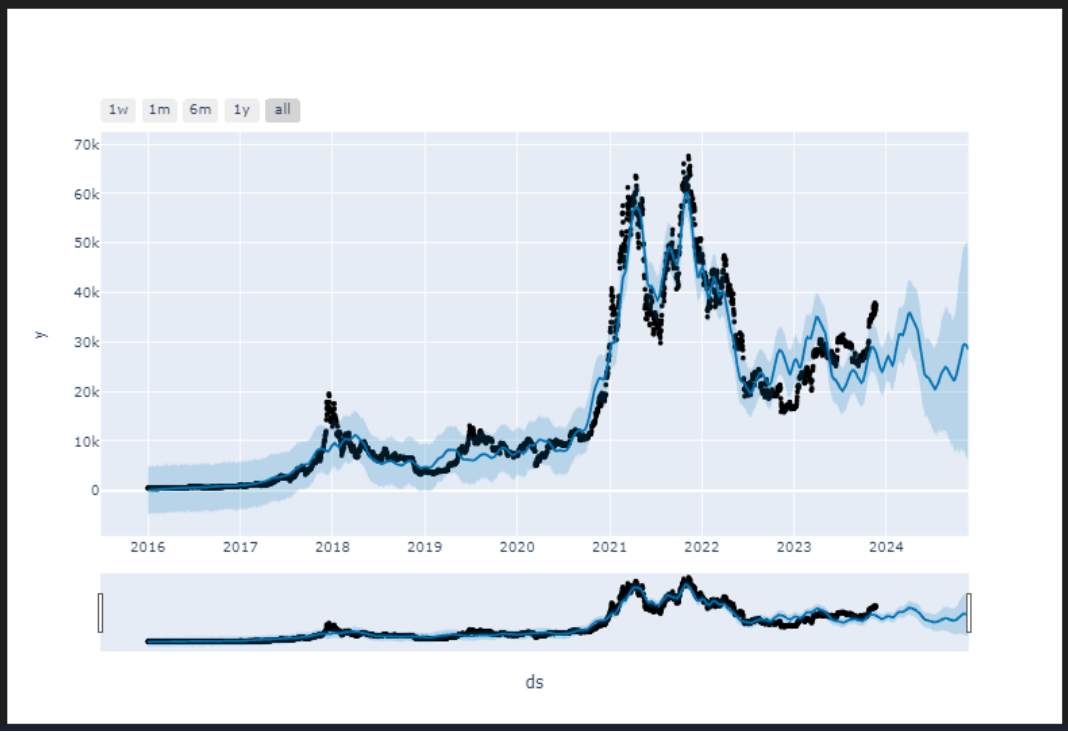
[21] ✓ 0.0s

```
Predicted Bitcoin price for 2023-11-23 is 27545.325940273993
```

Model predicted graph

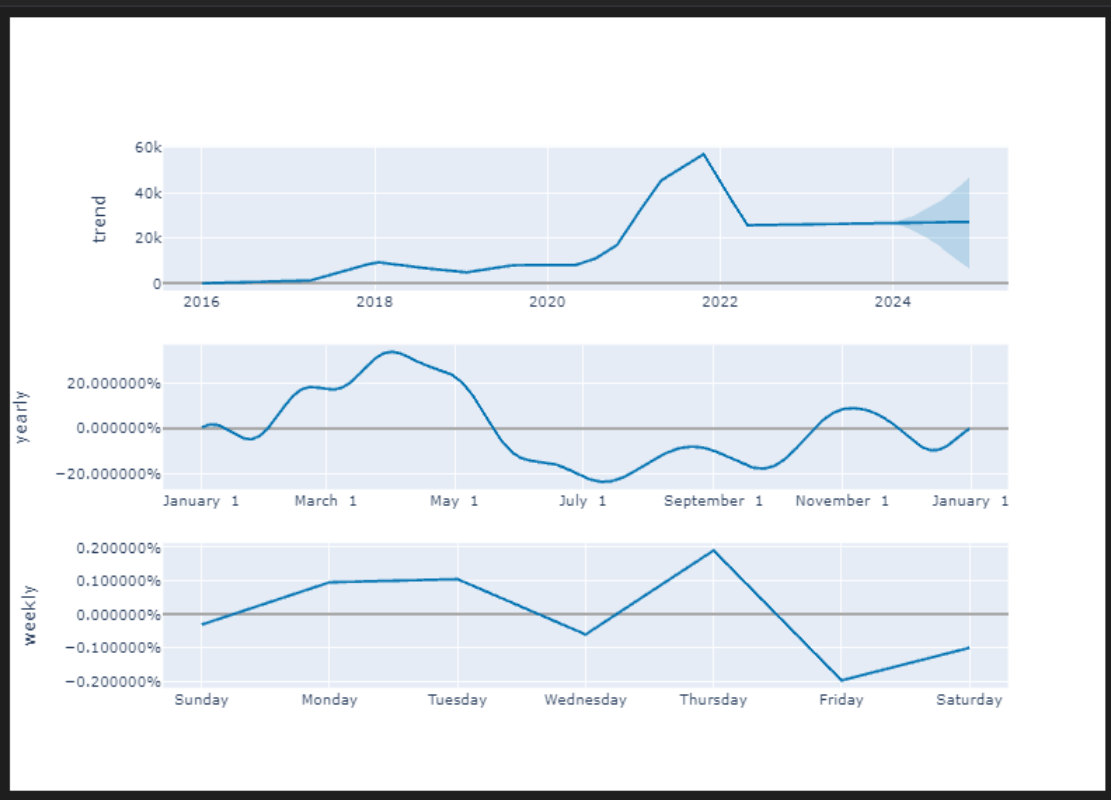
```
plot_plotly(model,forecast)
```

[22] ✓ 0.2s



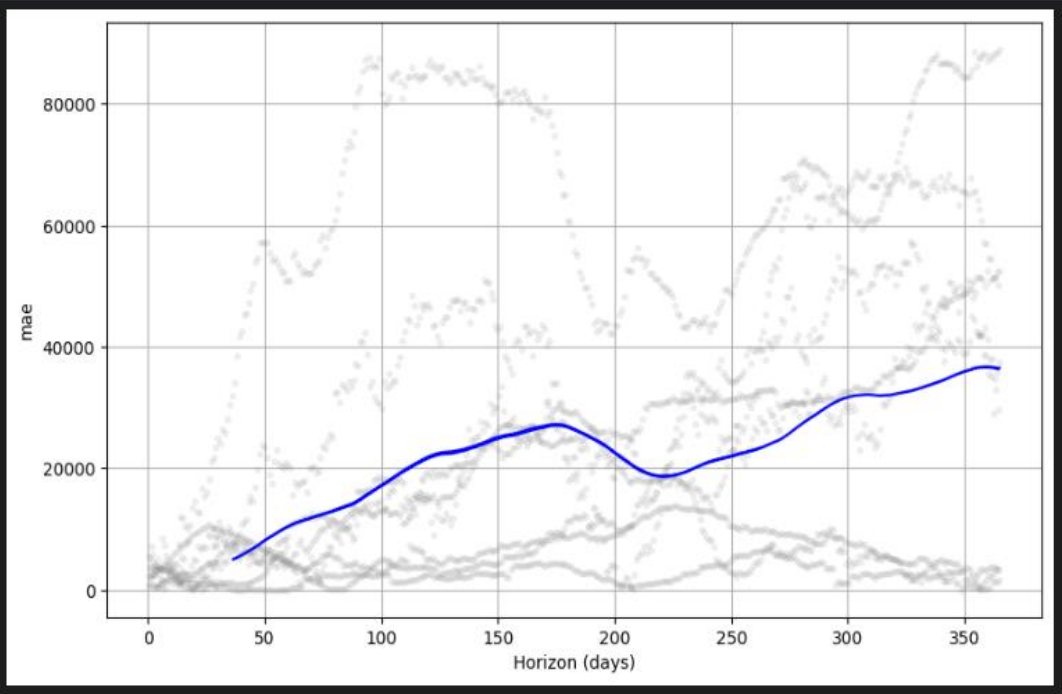

```
plot_components_plotly(model,forecast)
```

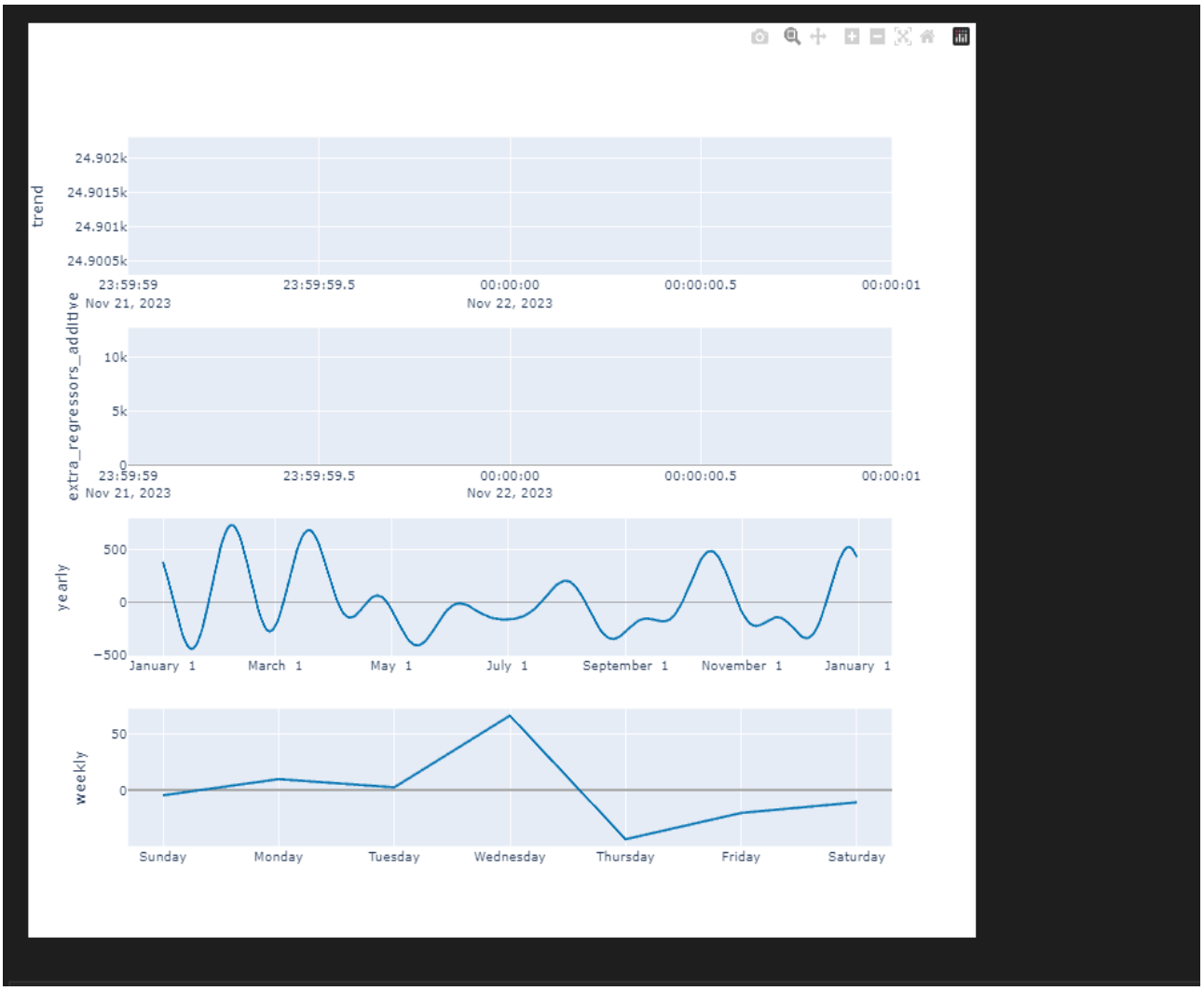
[23] ✓ 0.7s



```
from prophet import plot
fig2=plot.plot_cross_validation_metric(df_cv,metric='mae')
```

[30] ✓ 0.3s





Testing

```
# Next day Bit coin price prediction

next_day=(datetime.today()+timedelta(days=1)).strftime('%Y-%m-%d')

print(f"Predicted Bitcoin price for {next_day} is {forecast2[forecast2['ds']==next_day]['yhat'].item()}")
```

[33] ✓ 0.0s

```
Predicted Bitcoin price for 2023-11-23 is 25492.096309776058
```

Testing accuracy

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np
from prophet.diagnostics import performance_metrics

forecast = model.predict(future)
y_true = df1['y']
forecast_upto_today = forecast[forecast['ds'] < datetime.today().strftime('%Y-%m-%d')]
y_pred = forecast_upto_today['yhat']
mae = mean_absolute_error(y_true, y_pred)
print(f"Mean Absolute Error (MAE): {mae}")
mse = mean_squared_error(y_true, y_pred)
print(f"Mean Squared Error (MSE): {mse}")
r2 = r2_score(y_true, y_pred)
print(f"R-squared (R2): {r2}")
rmse = np.sqrt(mse)
print(f"Root Mean Squared Error (RMSE): {rmse}")
```

1] ✓ 1.4s

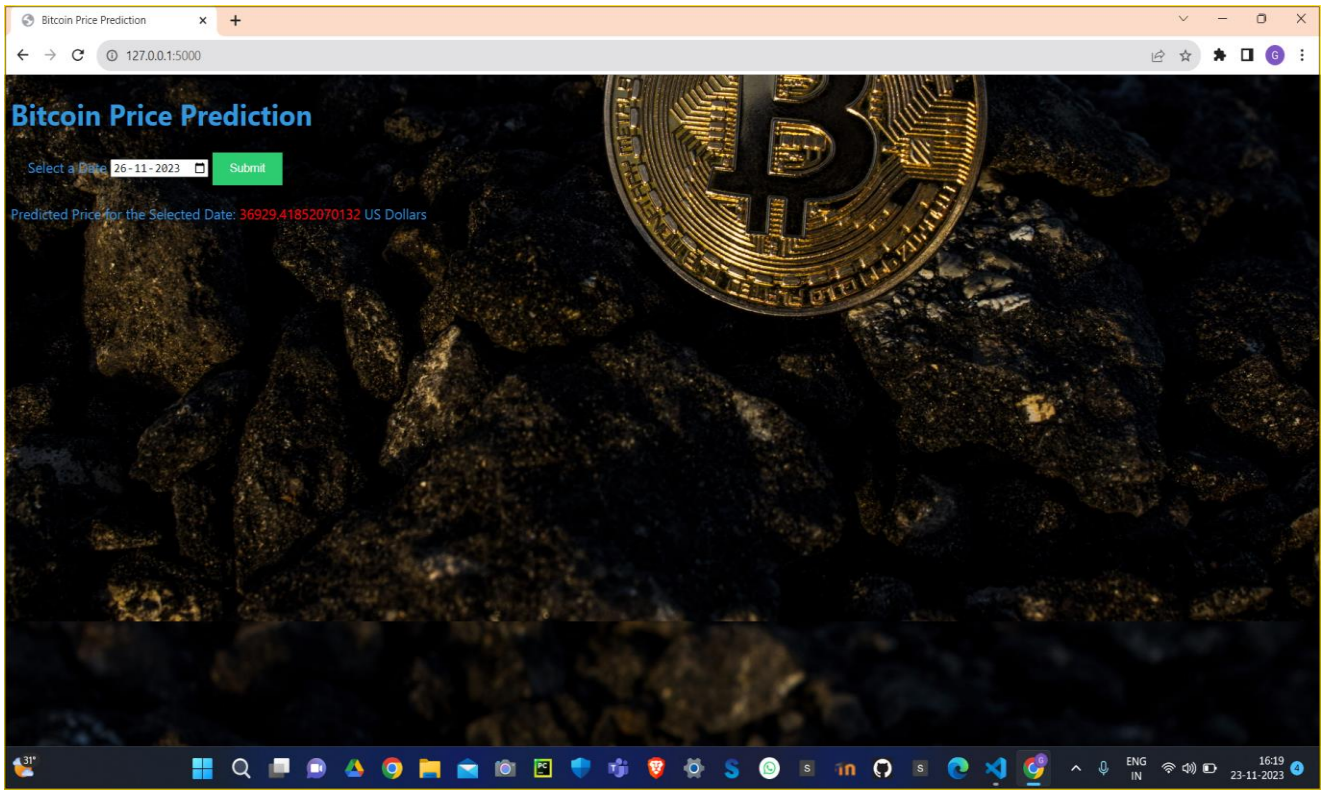
```
from prophet import Prophet

# model 2

model2=Prophet(
    seasonality_mode="multiplicative",

    # Hyperparameter tuning
    changepoint_prior_scale=0.5,
    seasonality_prior_scale=10,
)

model2.fit(df1)

forecast2 = model2.predict(future)
y_true = df1['y']
forecast_upto_today2 = forecast2[forecast2['ds'] < datetime.today().strftime('%Y-%m-%d')]
y_pred = forecast_upto_today2['yhat']
mae = mean_absolute_error(y_true, y_pred)
print(f"Mean Absolute Error (MAE): {mae}")
mse = mean_squared_error(y_true, y_pred)
print(f"Mean Squared Error (MSE): {mse}")
r2 = r2_score(y_true, y_pred)
print(f"R-squared (R2): {r2}")
rmse = np.sqrt(mse)
print(f"Root Mean Squared Error (RMSE): {rmse}")
```

[32] ✓ 3.3s

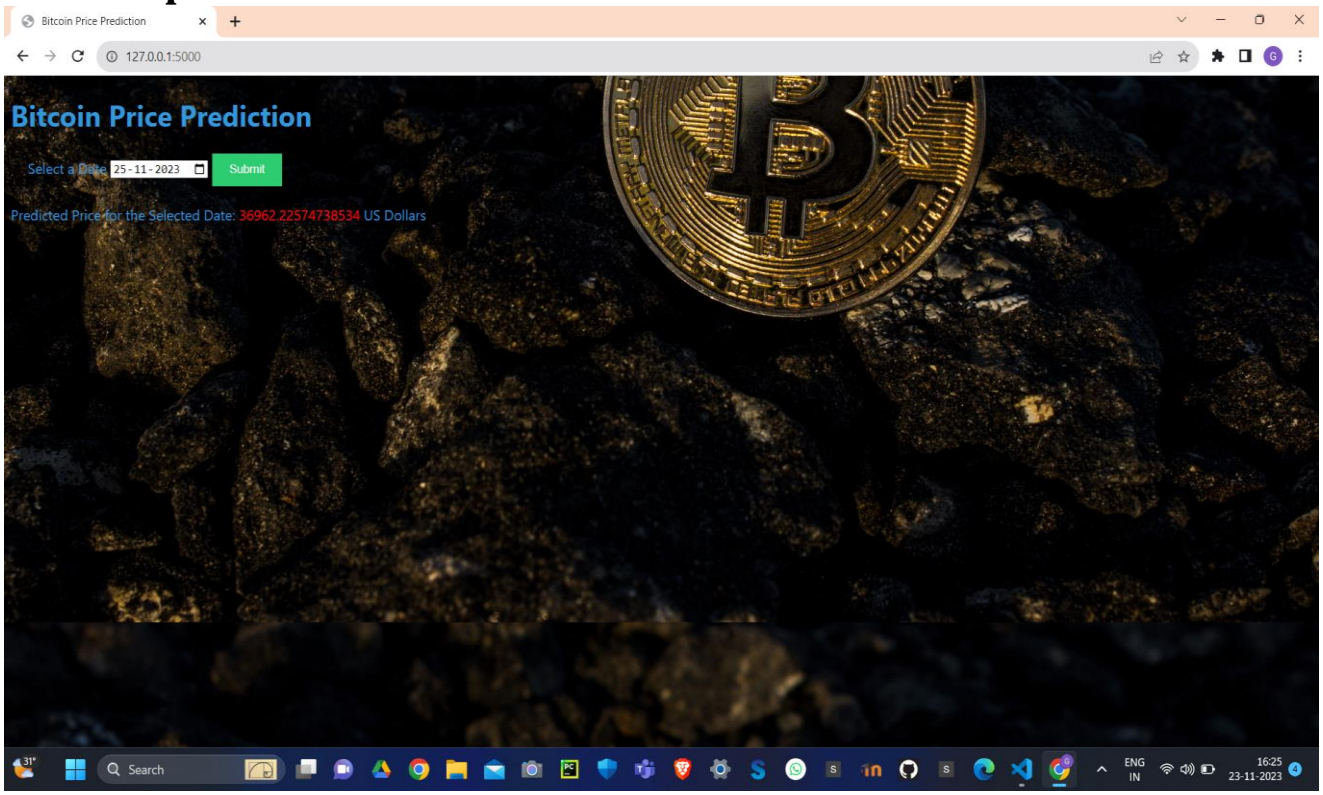
```
23:11:44 - cmdstanpy - INFO - Chain [1] start processing
23:11:47 - cmdstanpy - INFO - Chain [1] done processing
Mean Absolute Error (MAE): 1664.8419407420868
Mean Squared Error (MSE): 6742530.846332288
R-squared (R2): 0.9741887067494618
Root Mean Squared Error (RMSE): 2596.63837419312
```

9. Result

9.1. Output1


Bitcoin Price Prediction
127.0.0.1:5000
Bitcoin Price Prediction
Select a Date: 26-11-2023
Submit
Predicted Price for the Selected Date: 36929.41852070132 US Dollars

9.2. Output2



10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

Ease of Use:

Prophet is designed to be user-friendly, making it accessible to both expertsand non-experts in time series analysis. Its intuitive interface and default settings simplify the modeling process.

Handling Missing Data:

Prophet can effectively handle missing data and outliers, which are common challenges in time series analysis. This robustness contributes to more accuratepredictions.

Incorporation of Seasonality:

Prophet is well-suited for capturing seasonality patterns in time series data, which is crucial in the context of Bitcoin price prediction where periodic trendsmay be present.

Flexible Trend Components:

The model allows for flexible specification of trend components, including logistic growth and saturating minimums, making it adaptable to various typesof time series data.

Automatic Holiday Effects:

Prophet has the ability to incorporate holiday effects automatically, which canbe advantageous in cryptocurrency markets where price movements might beinfluenced by holidays or specific events.

Interpretability:

The transparency of Prophet's model components allows users to interpret thecontributions of different factors to the overall prediction, aiding in model understanding.

Open-Source and Active Community:

Prophet is an open-source tool developed and maintained by Facebook. It benefits from a vibrant community, resulting in ongoing improvements, bugfixes, and a wealth of resources for users.

Disadvantages :

Sensitivity to Model Parameters:

Prophet's performance can be sensitive to the choice of hyperparameters.Users need to carefully tune these parameters to achieve optimal results, which can be a non-trivial task.

Limited Handling of External Factors:

While Prophet allows the inclusion of custom seasonality components, it maynot handle external factors (e.g., regulatory changes, geopolitical events) as effectively as more complex models designed for such considerations.

Assumption of Additive Components:

Prophet assumes that different components, such as seasonality and holidays,are additive. In cases where interactions between components are not strictlyadditive, this assumption may limit the model's accuracy.

Long-Term Predictions:

Prophet may face challenges when making long-term predictions, especially indynamic markets like cryptocurrencies. The model's reliance on historical datamight limit its ability to capture longer-term trends.

Limited Control Over Feature Engineering:

While Prophet handles missing data well, users have limited control over feature engineering compared to more customizable models. This can be adisadvantage when dealing with complex time series patterns.

Resource Intensive for Large Datasets:

For large datasets, the computational requirements of Prophet can be resource-intensive. This may pose challenges in terms of both processing timeand memory usage.

Not Necessarily the Best Fit for All Time Series:

While Prophet is effective for many time series forecasting tasks, it may not bethe best fit for all scenarios. Complex or highly irregular time series data mightbe better suited to more advanced models.

11. CONCLUSION

This project aims to leverage time series analysis and the Prophet model to make accurate predictions of Bitcoin prices. The structured approach throughdifferent phases ensures a thorough exploration of data, model selection, andproper evaluation before finalizing predictions. Continuous refinement and documentation throughout the project contribute to its success and maintainability.

12. FUTURE SCOPE

The scope of this project is to develop a time series BTC price prediction modelusing machine learning techniques and a user-friendly platform to assist investors and traders in making informed decisions about BTC investments.
Collect and cleanse historical BTC price data from credible sources.

Develop and evaluate various machine learning algorithms for predicting BTC price movements. Construct a multi-algorithm prediction model that combinesthe strengths of multiple machine learning algorithms.
Implement the prediction model in a user-friendly web application.
Provide real-time BTC price updates, historical price data analysis, risk assessment tools, potential trading strategies, and educational resources.

The project will focus on the following aspects:

Accurate and reliable price predictions: The prediction model should generatepredictions that are accurate and reliable within acceptable error margins.
User-friendly interface: The web application should have a user-friendly interface that is easy to navigate and understand, even for non-technical users.

Performance and scalability:

The prediction model and web application should be scalable to handleincreasing data volumes and user traffic.

Security and privacy:

The system should implement robust security measures to protect user dataand historical price data from unauthorized access and breaches.

The project will exclude the following:

Trading execution:

The project will not directly involve trading BTC or providing trading signals. Itwill focus on providing prediction tools and analytics to facilitate informed trading decisions.

Consulting or advising:

The project will not provide personalized consulting or advising services toindividual investors or traders. It will focus on developing a generalized prediction model and web application for broader use.

Prediction for specific time horizons:

The project will focus on developing a general prediction model that can beused for various time horizons, but it may not be optimized for specific time frames. The project will be completed within a timeframe of 6 months. The specific timeline will be divided into the following phases:

Data Collection and Preprocessing :

Collect historical BTC price data from various sources, clean and prepare thedata for analysis.

Model Development and Evaluation :

Develop and evaluate various machine learning algorithms, optimize the multi-algorithm prediction model, and evaluate its performance.

Web Application Development and Deployment :

Develop the user-friendly web application, integrate the prediction model, anddeploy the application to a cloud platform.