TITLE OF THE PROJECT :

Time Series Analysis For Bitcoin Price Prediction Using Prophet

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 a model 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 other cryptocurrencies.

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 develop new 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 model for 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 the historical price of BTC, the current market conditions, and news events.

Neural networks: Neural networks are a type of machine learning algorithm that 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 data and 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 collected from a variety of sources.

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

Model evaluation phase: In this phase, the developed models will be evaluated on 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 intended to 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 Prophet or other forecasting methods is a popular and interesting topic, there are several challenges and issues associated with it. It's important to note that the cryptocurrency market, including Bitcoin, is highly volatile and influenced by a variety 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. The market 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 analysis techniques.

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 and preprocessing 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 dynamics of 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 appropriate level 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 regulatory developments, 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 constraint for 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 right set of parameters can be a non-trivial task.

2.2 References

Bitcoin Forecasting Using ARIMA and PROPHET: Bitcoin Forecasting Using ARIMA 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, LSTM and XGBoost (ijraset.com)

GitHub - prasoon2510/Bitcoin_Price: GitHub - prasoon2510/Bitcoin_Price

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

EAI Endorsed Transactions - Time-Series Prediction of Cryptocurrency Market using 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 cryptocurrency market pose significant challenges for investors, traders, and analysts. Traditional financial markets often leverage time series analysis techniques to forecast 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 Prophet forecasting tool, to address the following challenges in Bitcoin price prediction:

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

considering factors such as market sentiment, regulatory changes, and macroeconomic 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's ability 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 historical data 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 are trying 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 strategies

Access to a user-friendly platform

Challenges:

Volatility of the cryptocurrency market

Difficulty in identifying patterns in historical data

Lack of reliable investment tools

Information overload from various sources

Pain Points:

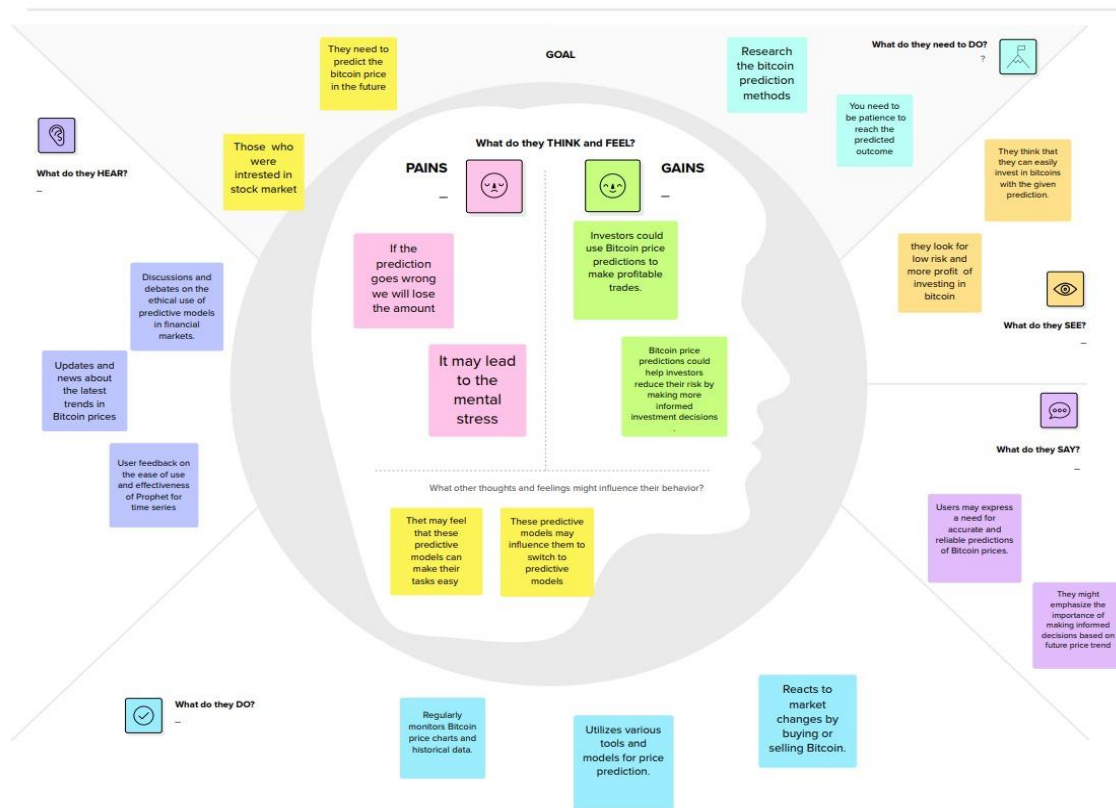Making investment decisions based on fear or emotions

Experiencing losses due to misinformed trading

Missing out on potential profits due to missed opportunities

Feeling overwhelmed by the vast amount of information available

Time Series Analysis For Bitcoin Price Prediction Using Prophet
This helps you to understand the the future predictions of the Bitcoin.
So, that you can take your next step on it.

GOAL

What do they HEAR? —

They need to predict the bitcoin price in the future

Those who were intrested in stock market

What do they THINK and FEEL?

PAINS —

GAINS —

Research the bitcoin prediction methods

What do they need to DO? ?

You need to be patience to reach the predicted outcome

They think that they can easily invest in bitcoins with the given prediction.

Discussions and debates on the ethical use of predictive models in financial markets.

Updates and news about the latest trends in Bitcoin prices

User feedback on the ease of use and effectiveness of Prophet for time series

If the prediction goes wrong we will lose the amount

It may lead to the mental stress

Investors could use Bitcoin price predictions to make profitable trades.

Bitcoin price predictions could help investors reduce their risk by making more informed investment decisions.

they look for low risk and more profit of investing in bitcoin

What do they SEE?

What other thoughts and feelings might influence their behavior?

Thet may feel that these predictive models can make their tasks easy

These predictive models may influence them to switch to predictive models

Users may express a need for accurate and reliable predictions of Bitcoin prices.

What do they SAY?

They might emphasize the importance of making informed decisions based on future price trend

What do they DO? —

Regularly monitors Bitcoin price charts and historical data.

Utilizes various tools and models for price prediction.

Reacts to market changes by buying or selling Bitcoin.

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 multiple machine 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 affecting BTC 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 the decision-making process and empower investors to make informed trades.

Offer educational resources and tutorials: We propose providing educational resources 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. This would foster a collaborative learning environment and provide valuable support for new investors.

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

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.
- 💡 Encourage wild ideas.
- 😊 Defer judgment.
- Listen to others.
- Go for volume.
- 👁 If possible, be visual.

Brainstorm

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

⏱ 10 minutes

Vishnu

Creating a ML model to predict

Identify and collect historical Bitcoin price data from reliable sources

Explore the inclusion of additional relevant data sources, such as social media

Rithwik

Train the model on large dataset to increase accuracy.

LSTM networks can be effective in capturing long-term dependencies in time series data.

FB Prophet model best suits for this time series data

Siddharth

Data Preprocessing clean the historical data to handle missing values,outliers,and normalize or scale the data to ensure consistency

Test the Bitcoin price predictionmodel in productiion and make necessary adjustments using evaluation metrics check accuracy of the model

Document the entire process ,including data preprocessing step,modeltraining ,and evaluuation methodologies.

Purushotham

Ensure that the deployed model can handle an increasing volume of data and requests

Implement a mechanism for real-time updates, allowing the model to adapt quickly to new data and market conditions

Consider deploying the model on scalable cloud infrastructure to accommodate fluctuations in user demand

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

Data Collection and Data Preprocessing

Model evaluation and model deployment

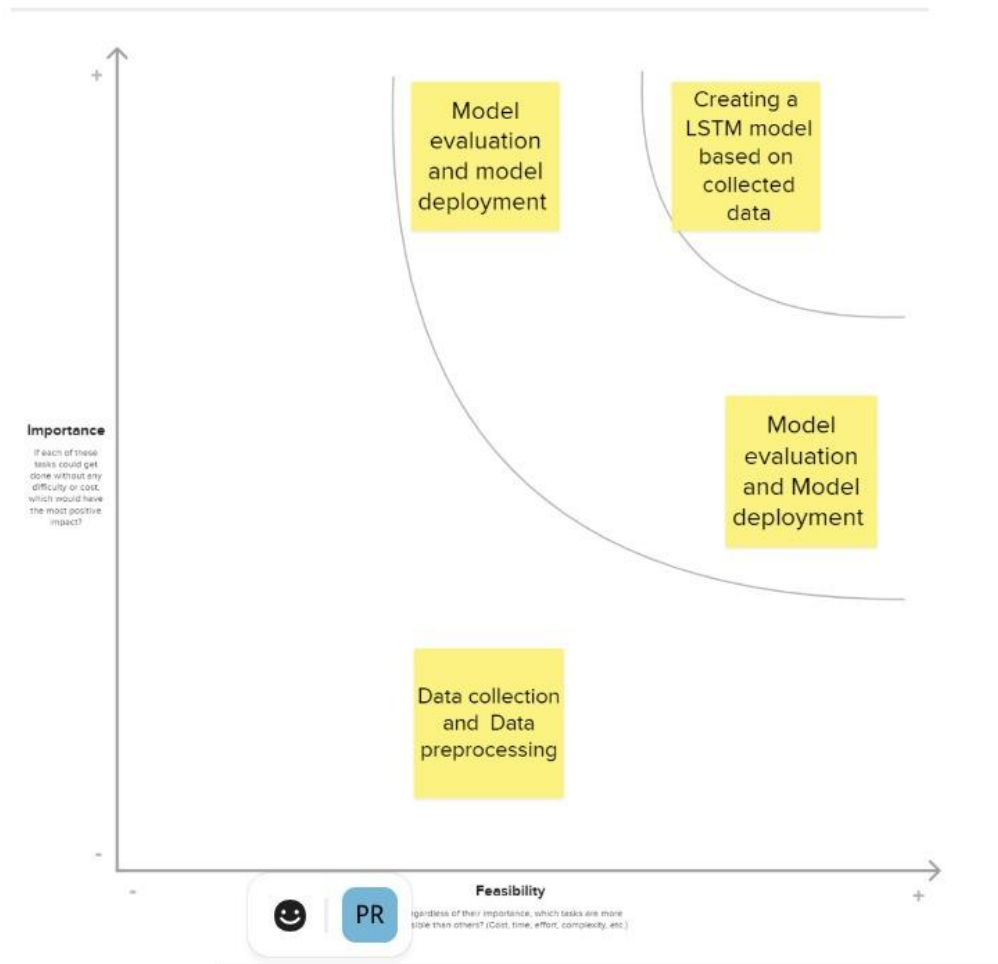Creating a LSTM model based on collected data

Scalability and Real-Time Updates

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 focus on what the system should do rather than how it should do it.

Core Functional Requirements:

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

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

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

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

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

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

Additional Functional Requirements:

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

Performance Tracking: The system should track the performance of prediction models 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 that best suits their needs.

Visualization and Reporting Tools: The system should provide visualization and reporting 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 focus on how the system should behave rather than what it should do.

Performance Requirements:

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

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

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

Reliability: The system should be reliable and operate consistently without frequent downtime or errors.

Security Requirements:

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

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

Privacy Protection: User privacy should be respected, and personal data should not be shared without explicit consent.

Usability Requirements:

Intuitive Interface: The interface should be intuitive, easy to navigate, and understandable 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 and provide alternative input and output methods.

Maintenance Requirements:

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

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

Documentation and Support: Comprehensive documentation and support should 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 model training. 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 is saved for future use.

Model Evaluation: The performance of the trained model is assessed using a separate 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 it accessible 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.


Data Flow Diagram

Time Series Analysis For Bitcoin Price Prediction Using Prophet
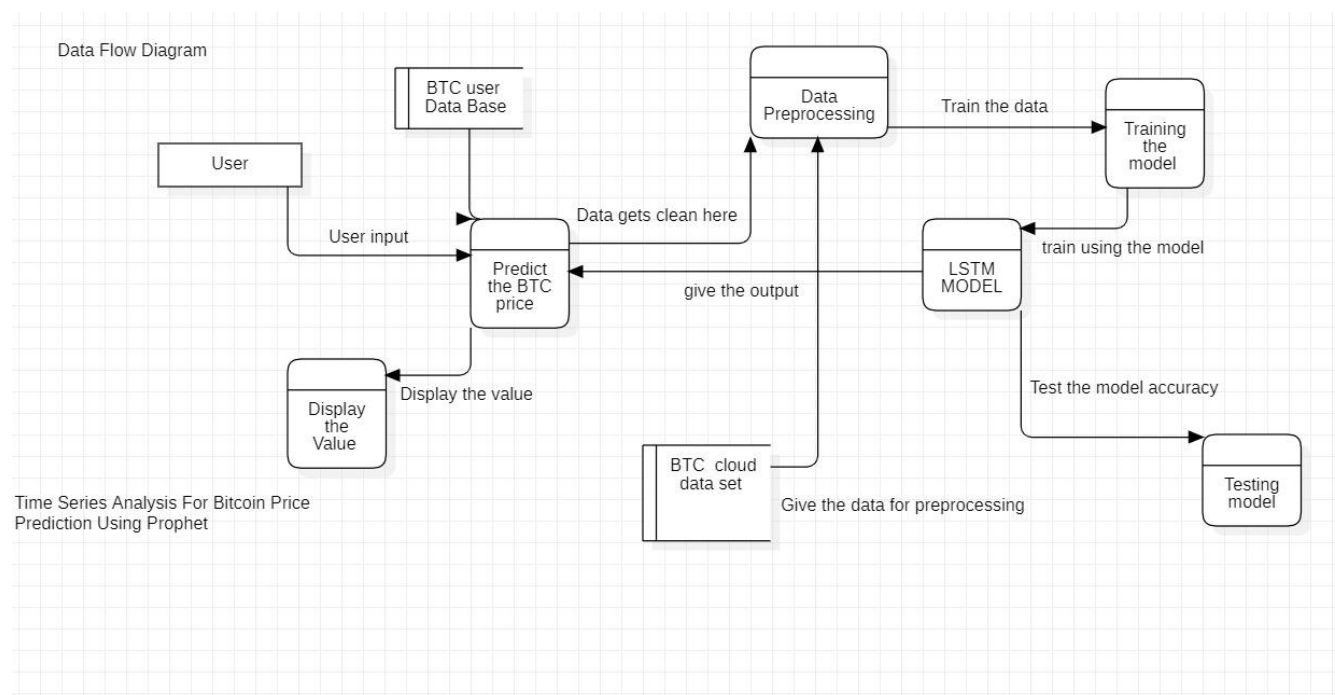
User Stories:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data collection	USN-1	Collect the datasets of the historical BTC price from different articles and research papers for building the deep learning model to forecast the BTC future predictions	2	High	Purushotham
Sprint-1	Validation of the data	USN-2	After collection of the data cleaning the data using proper methods eg (check for null values , encoding ,outliers and etc) of preprocessing and scaling the data	3	High	Purushotham
Sprint-2	Data visualization	USN-3	After the all preprocessing now will we visualized for understanding the data patterns and visualize for model selection	2	Low	Siddharth
Sprint-2	Model Selection	USN-4	After analysis of the data now we will select the appropriate deep learning model for training the data to give future predictions	3	High	Siddharth
Sprint-3	Model development	USN-5	Train the selected deep learning model using the preprocessed dataset and monitor its performance on the validation set.	3	High	Vishnu
Sprint-3	Evaluation metrics	USN-6	Finding the accuracy of the model using different techniques eg(f1 score ,confusion matrix etc...)	2	Low	Vishnu
Sprint-4	UI designing	USN-7	After testing creation of the friendly user interface and suitable for the model we have created	5	High	Ritvik
Sprint-4	Model deployment & Integration	USN-8	Deploy the trained deep learning model as an API or web service to make it accessible to BTC price prediction . Integrate the model's API into a user-friendly web interface for users to upload the data and get price forecast of the BTC	5	High	Ritvik

5.2 Solution Architecture

Solution Architecture Diagram:
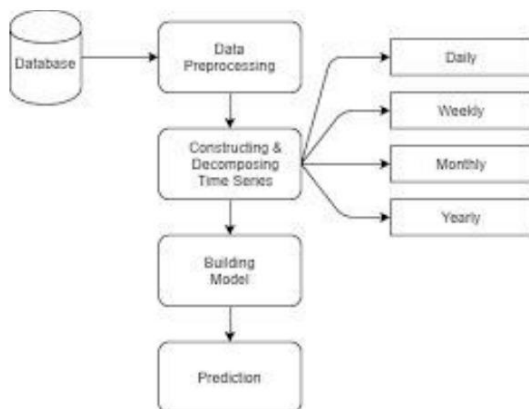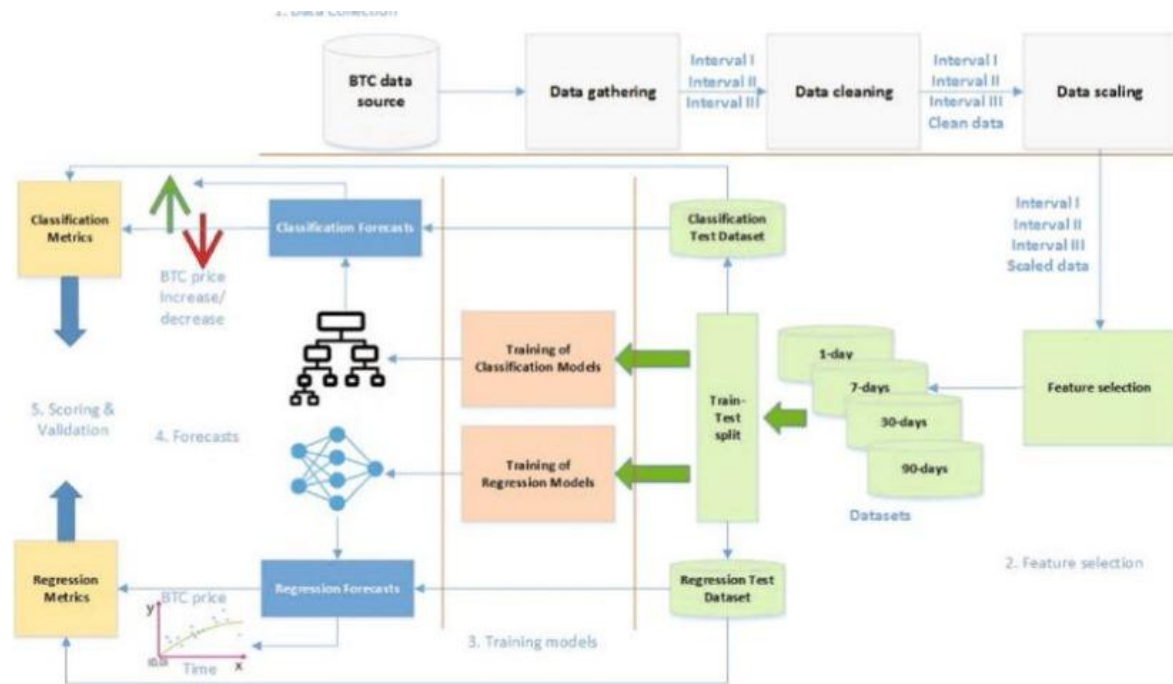
6. PROJECT PLANNING & SCHEDULING
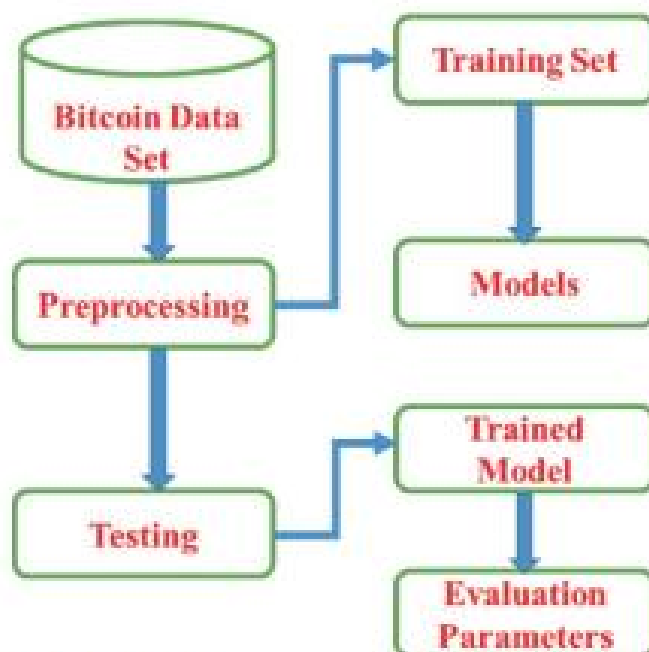
6.1 Technical Architecture

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 evnironment	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	Data collection	USN-1	Collect the datasets of the historical BTC price from different articles and research papers for building the deep learning model to forecast the BTC future predictions	2	High	Purushotham
Sprint-1	Validation of the data	USN-2	After collection of the data cleaning the data using proper methods eg (check for null values , encoding ,outliers and etc) of preprocessing and scaling the data	3	High	Purushotham
Sprint-2	Data visualization	USN-3	After the all preprocessing now will we visualized for understanding the data patterns and visualize for model selection	2	Low	Siddharth
Sprint-2	Model Selection	USN-4	After analysis of the data now we will select the appropriate deep learning model for training the data to give future predictions	3	High	Siddharth
Sprint-3	Model development	USN-5	Train the selected deep learning model using the preprocessed dataset and monitor its performance on the validation set.	3	High	Vishnu
Sprint-3	Evaluation metrics	USN-6	Finding the accuracy of the model using different techniques eg(f1 score ,confusion matrix etc…)	2	Low	Vishnu
Sprint-4	UI designing	USN-7	After testing creation of the friendly user interface and suitable for the model we have created	5	High	Ritvik
Sprint-4	Model deployment & Integration	USN-8	Deploy the trained deep learning model as an API or web service to make it accessible to BTC price prediction . Integrate the model's API into a user-friendly web interface for users to upload the data and get price forecast of the BTC	5	High	Ritvik

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:

```python
from prophet import Prophet
from prophet.plot import plot_plotly
import pandas as pd, matplotlib.pyplot as plt, yfinance as yf
```

```python
df=yf.download('BTC-USD')
```

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

```python
df
```

Date	Open	High	Low	Close	Adj Close	Volume
2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600
2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100
...	...	...	...	...	...	...
2023-11-16	37879.980469	37934.625000	35545.472656	36154.769531	36154.769531	26007385366
2023-11-17	36164.824219	36704.484375	35901.234375	36596.683594	36596.683594	22445028430
2023-11-18	36625.371094	36839.281250	36233.312500	36585.703125	36585.703125	11886022717

```python
df.reset_index(inplace=True)
df=df[["Date","Adj Close"]]
```

```python
df
```

	Date	Adj Close
0	2014-09-17	457.334015
1	2014-09-18	424.440002
2	2014-09-19	394.795990
3	2014-09-20	408.903992
4	2014-09-21	398.821014
...	...	...
3347	2023-11-16	36154.769531
3348	2023-11-17	36596.683594
3349	2023-11-18	36585.703125
3350	2023-11-19	37386.546875
3351	2023-11-20	37141.347656

3352 rows × 2 columns

```
df.columns=["ds","y"]
df
```

	ds	y
0	2014-09-17	457.334015
1	2014-09-18	424.440002
2	2014-09-19	394.795990
3	2014-09-20	408.903992
4	2014-09-21	398.821014
...	...	...
3347	2023-11-16	36154.769531
3348	2023-11-17	36596.683594
3349	2023-11-18	36585.703125
3350	2023-11-19	37386.546875
3351	2023-11-20	37141.347656

3352 rows × 2 columns

Building the model:

```
model=Prophet()
```

```
model.fit(df)
```

```
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
DEBUG:cmdstanpy:input tempfile: /tmp/tmp3yc2p55v/d82i9t8y.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmp3yc2p55v/bmiet7xo.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.10/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=79991', 'data', 'file=/tmp/tmp3yc2p55v/d82i9t8y.json', 'i
12:14:29 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
12:14:30 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
<prophet.forecaster.Prophet at 0x7e50c409acb0>
```

```
model.component_modes
```

```
{'additive': ['yearly',
  'weekly',
  'additive_terms',
  'extra_regressors_additive',
  'holidays'],
 'multiplicative': ['multiplicative_terms', 'extra_regressors_multiplicative']}
```

```
df.head()
```

```
# Creating date for the 2 months # 60 days for the prediction of the bitcoin for better analysis
future_dates=model.make_future_dataframe(periods=60)
```
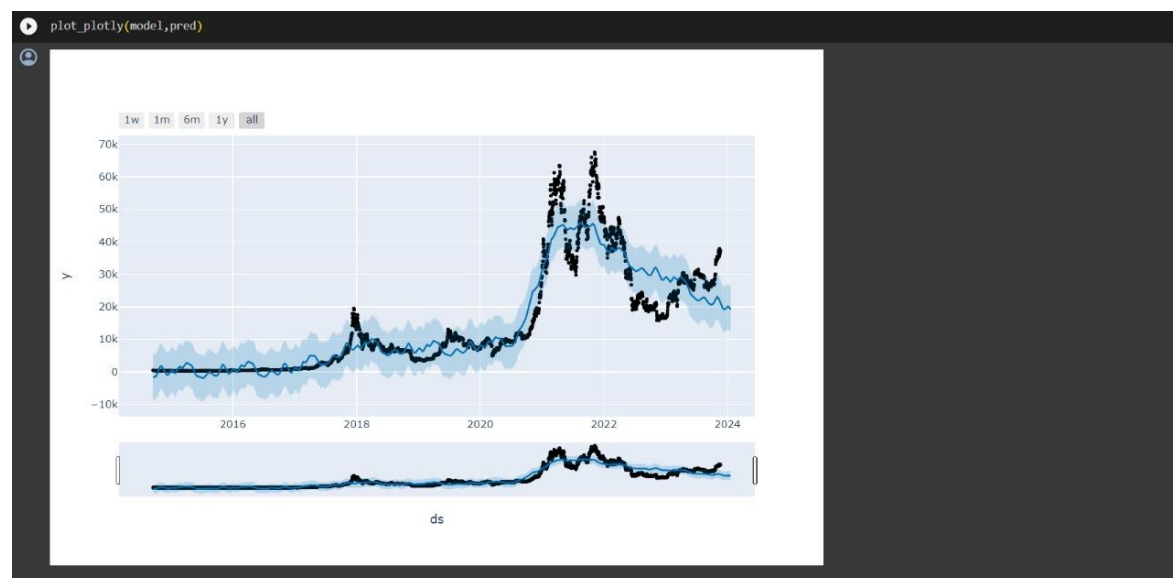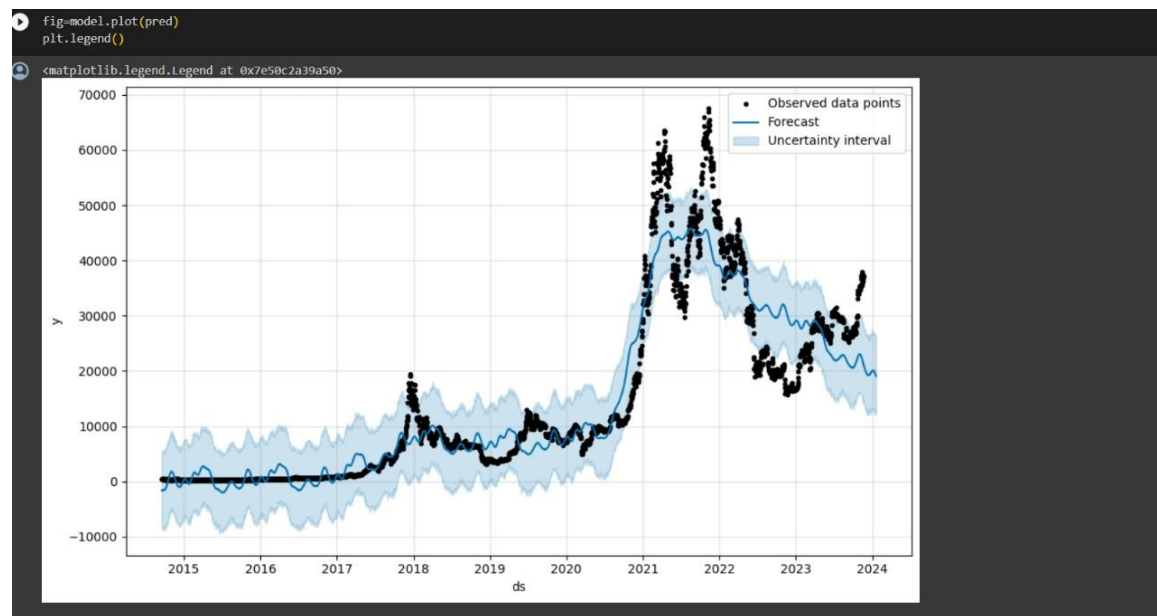
```
future_dates.tail()
```

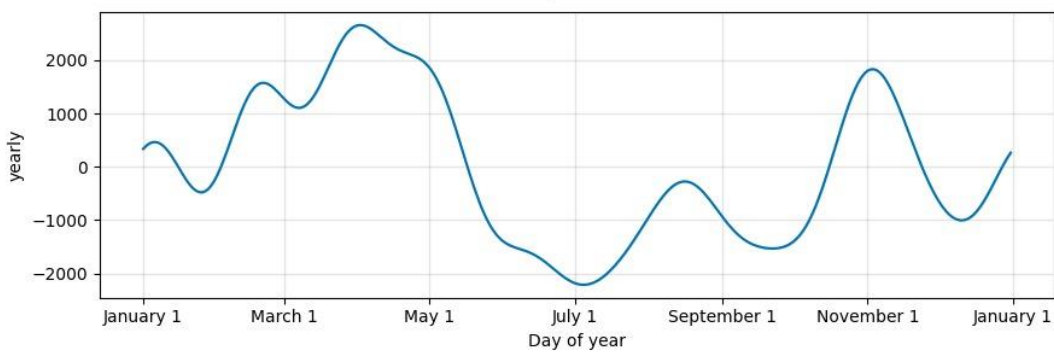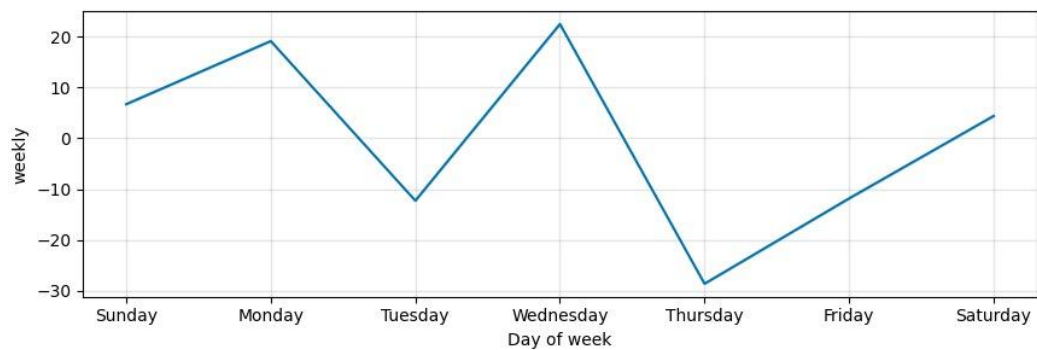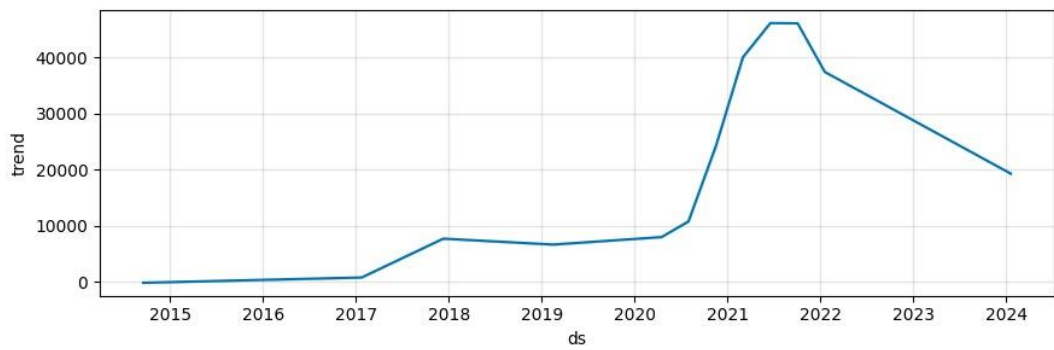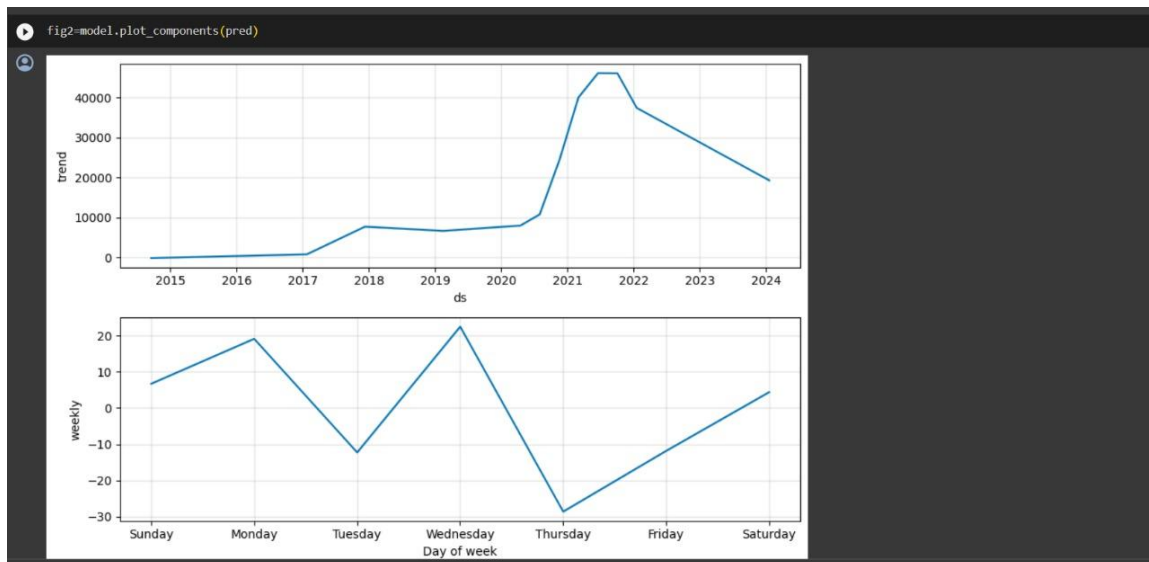|	ds
3407	2024-01-15
3408	2024-01-16
3409	2024-01-17
3410	2024-01-18
3411	2024-01-19

```
pred=model.predict(future_dates)
```

```
pred
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	weekly	weekly_lower	weekly_upper	
0	2014-09-17	-99.905663	-8371.381492	5182.057764	-99.905663	-99.905663	-1479.545535	-1479.545535	-1479.545535	22.507236	22.507236	22.507236	-1502.
1	2014-09-18	-98.838200	-8565.242079	5005.406025	-98.838200	-98.838200	-1541.169358	-1541.169358	-1541.169358	-28.648653	-28.648653	-28.648653	-1512.

Model predicted graph

```
fig=model.plot(pred)
plt.legend()
```

<matplotlib.legend.Legend at 0x7e50c2a39a50>


```
plot_plotly(model,pred)
```

```
fig2=model.plot_components(pred)
```

Testing

```
from datetime import datetime, timedelta
import pandas as pd
next_day = (datetime.today() + timedelta(days=1)).strftime('%Y-%m-%d') # format is important yyyy-mm-dd
future_df = model.make_future_dataframe(periods=1, include_history=False)
future_df['ds'] = pd.to_datetime(future_df['ds'])
future_df.loc[0, 'ds'] = pd.to_datetime(next_day)
forecast = model.predict(future_df)
predicted_value = forecast.loc[0, 'yhat']
print(f"Predicted value for {next_day}: {predicted_value}")
```

```
Predicted value for 2023-11-21: 21112.70109433989
```

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_dates)
y_true = df_cv['y']
y_pred = df_cv['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}")
```

```
Mean Absolute Error (MAE): 5361.4917615898685
Mean Squared Error (MSE): 76388899.93362887
R-squared (R2): 0.7071455611123004
Root Mean Squared Error (RMSE): 8740.074366596024
```

9. Result

9.1. Output1
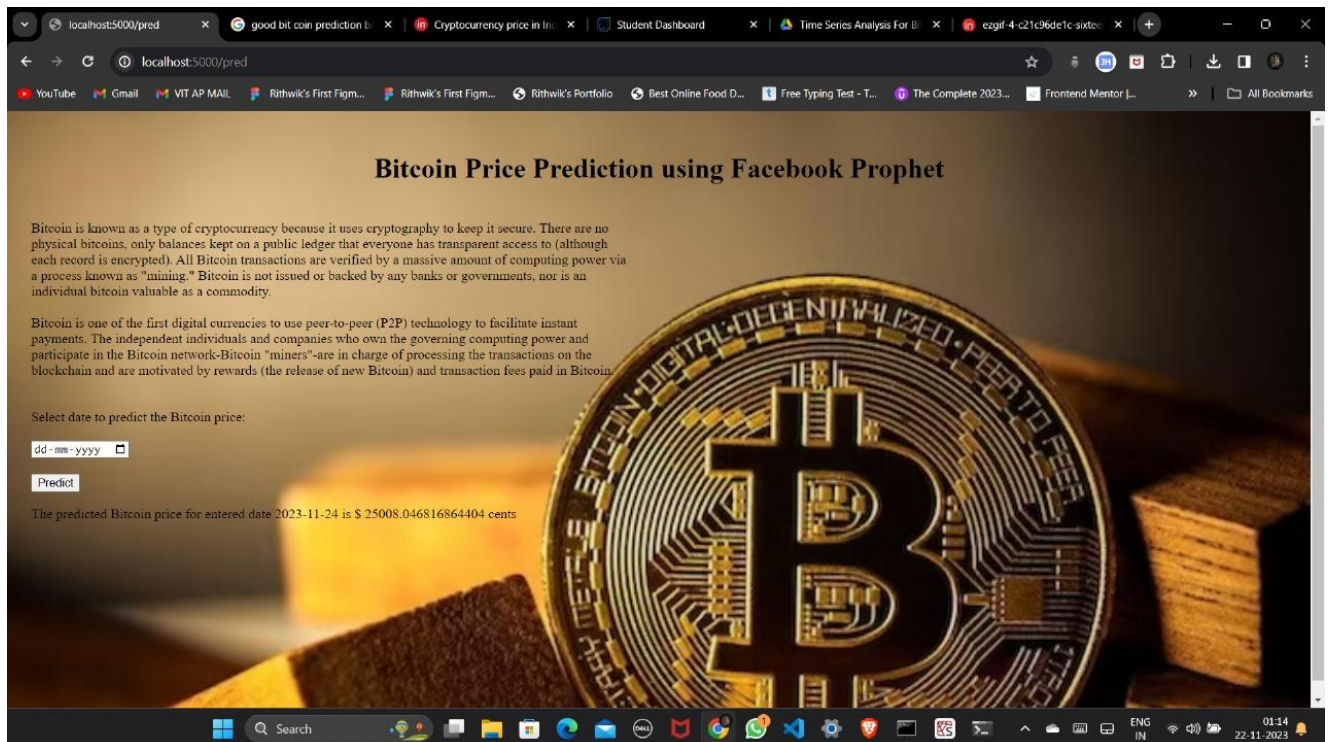
9.2. Output2

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

Ease of Use:

Prophet is designed to be user-friendly, making it accessible to both experts and 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 accurate predictions.

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 trends may 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 types of time series data.

Automatic Holiday Effects:

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

Interpretability:

The transparency of Prophet's model components allows users to interpret the contributions 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, bug fixes, 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 may not 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 strictly additive, this assumption may limit the model's accuracy.

Long-Term Predictions:

Prophet may face challenges when making long-term predictions, especially in dynamic markets like cryptocurrencies. The model's reliance on historical data might 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 a disadvantage 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 time and memory usage.

Not Necessarily the Best Fit for All Time Series:

While Prophet is effective for many time series forecasting tasks, it may not be the best fit for all scenarios. Complex or highly irregular time series data might be 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 through different phases ensures a thorough exploration of data, model selection, and proper 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 model using 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 combines the 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 generate predictions 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 handle increasing data volumes and user traffic.

Security and privacy:

The system should implement robust security measures to protect user data and 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. It will 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 to individual 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 be used 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 the data 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, and deploy the application to a cloud platform.