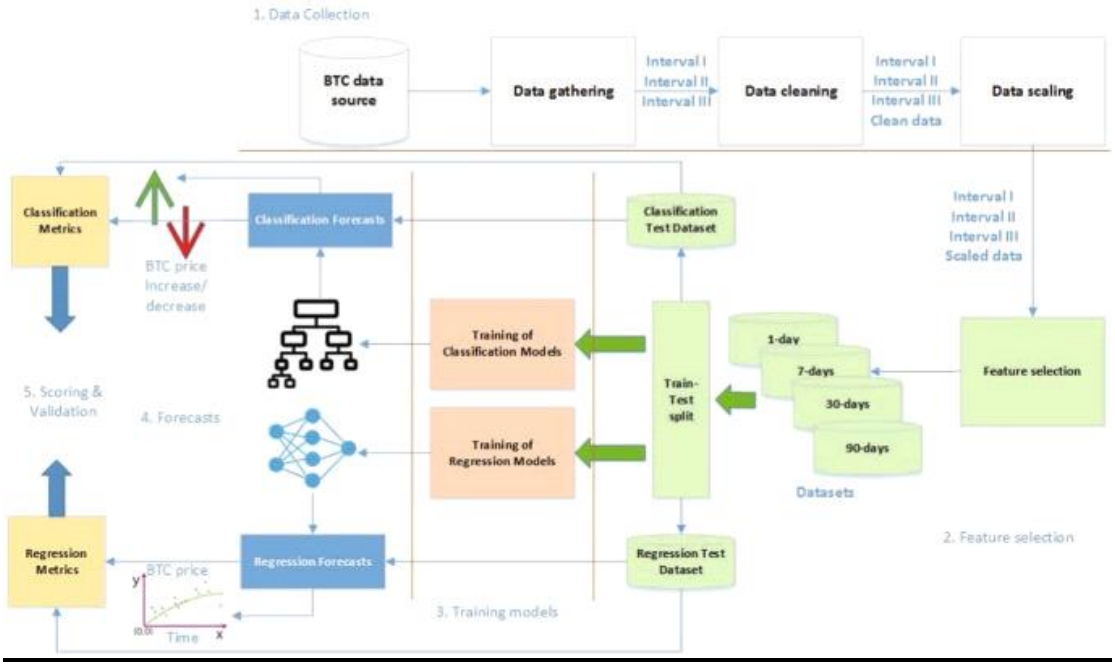
Project Design Phase
Solution Architecture

Date	06 NOv 2023
Team members	Lohith Bheemisetty Bulle Likhith Raj Anvesh J Surya Sathvik C Rohan Siddarth
Project Name	Time Series Analysis For Bitcoin Price Prediction Using Prophet
Maximum Marks	**5 Marks**
Team id	**592390**

Solution Architecture



1.**Data collection and preparation:**

Data collection involves gathering relevant data from various sources such as meteorological stations, weather satellites, or public weather APIs. The data should include features like temperature, humidity, wind speed, atmospheric pressure, cloud cover, and historical rainfall measurements.

Data preprocessing includes cleaning the collected data by handling missing values, removing outliers, and ensuring consistency in formatting. It may also involve feature engineering, where additional meaningful features are derived from the raw data, such as aggregating data over specific time intervals or incorporating domain knowledge.

2.**Model selection and training:**

Model selection depends on the specific problem and the characteristics of the dataset. Common machine learning algorithms for rainfall prediction include linear regression, decision trees, random forests, gradient boosting, and neural networks. The choice of algorithm may consider factors such as interpretability, scalability, and the ability to handle complex relationships in the data.
Once the algorithm is selected, the model is instantiated and trained on the prepared dataset. The training process involves feeding the input features and corresponding rainfall measurements to the model and adjusting its internal parameters to minimize the prediction error.

3.**Model evaluation:**

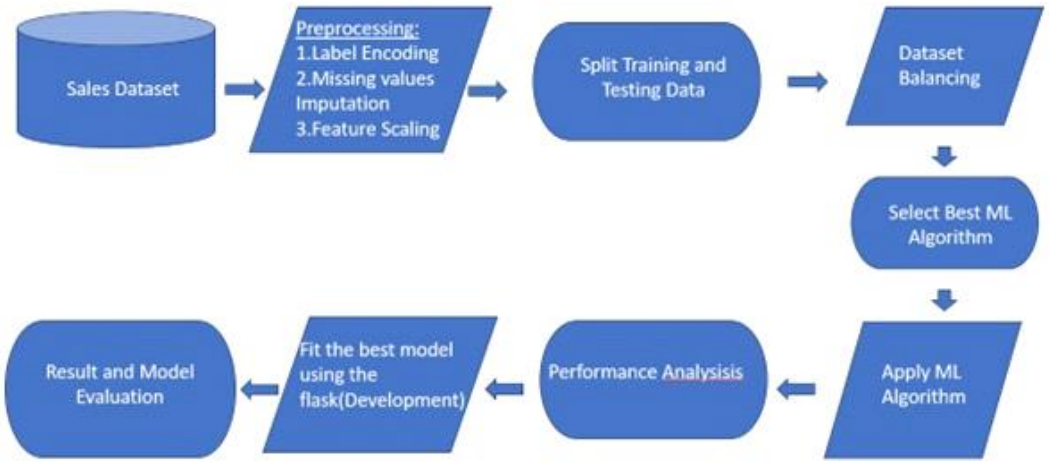
Model evaluation is crucial to assess the performance and generalization capability of the trained model. Evaluation metrics for rainfall prediction can include mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), or correlation coefficient. The model is evaluated on a held-out test set, which contains data that the model has not seen during training, to provide an unbiased estimate of its performance.

4.**Model deployment:**

Model deployment involves creating an application or system that integrates the trained model to provide real-time rainfall predictions. This can be done by developing a user interface (UI) where users can input relevant weather features, and the model generates the corresponding rainfall prediction.
The application can be deployed on a local server or a cloud computing platform to ensure accessibility and scalability. It should be designed to handle user requests efficiently and provide fast

predictions based on the trained model. Regular updates and maintenance may be required to incorporate new data and improve the model's performance over time.
Solution Architecture Diagram:

5.**The solution architecture diagram**



visually represents the flow of data and the components involved in the system. It provides a high-level overview of how data is collected, processed, and used to train the model. It also illustrates the deployment of the trained model into an application or system for real-time rainfall prediction.

Reference: https://link.springer.com/article/10.1007/s00521-020-05129-6