Project Planning Phase-II Technology Stack (Architecture & Stack)
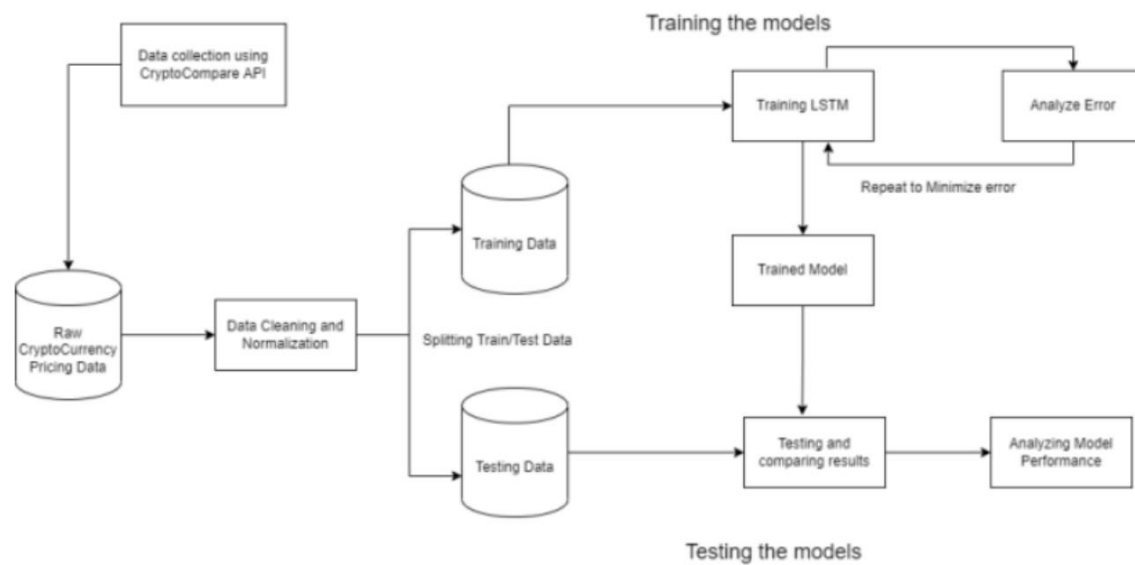
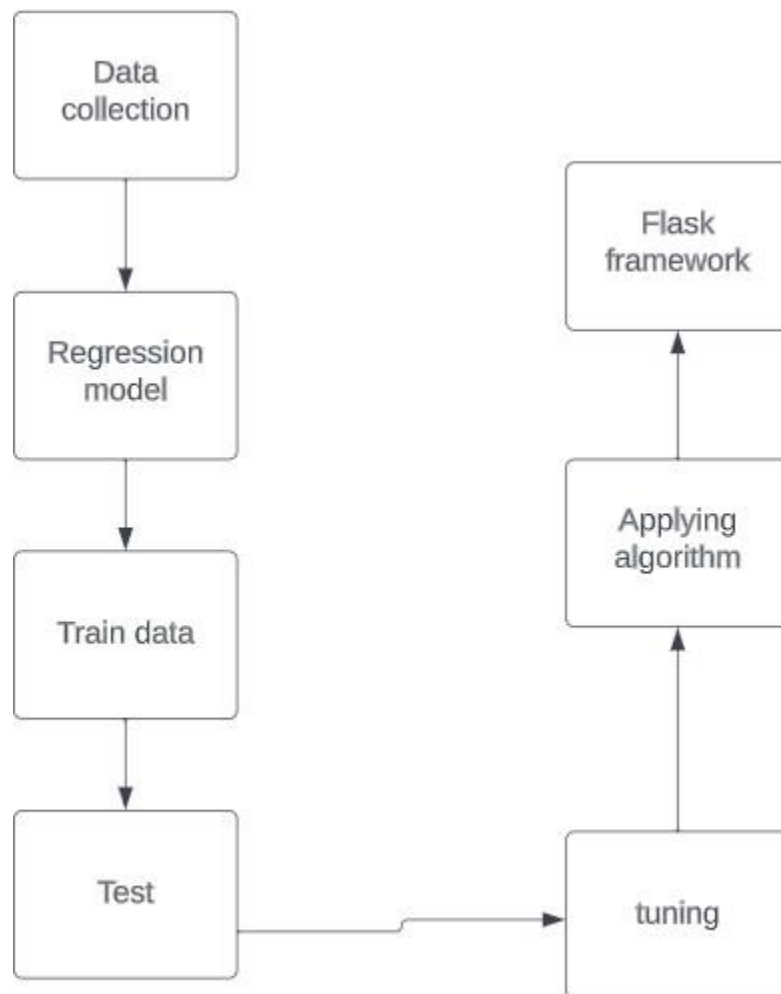| Date	13 November 2023
Team Id	592889
Project Name	Time Series Analysis For Bitcoin Price Prediction Using Prophet
Maximum Marks	4 Marks

Technical Archil'ecl'ure

• Architecture of the proposed model of bitcoin prediction using prophet.

The archil'ecl'ure of Bil'coin price predicl'ion

Data collection using CryptoCompare API

Raw CryptoCurrency Pricing Data

Data Cleaning and Normalization

Splitting Train/Test Data

Training Data

Testing Data

Training the models

Training LSTM

Analyze Error

Repeat to Minimize error

Trained Model

Testing and comparing results

Analyzing Model Performance

Testing the models
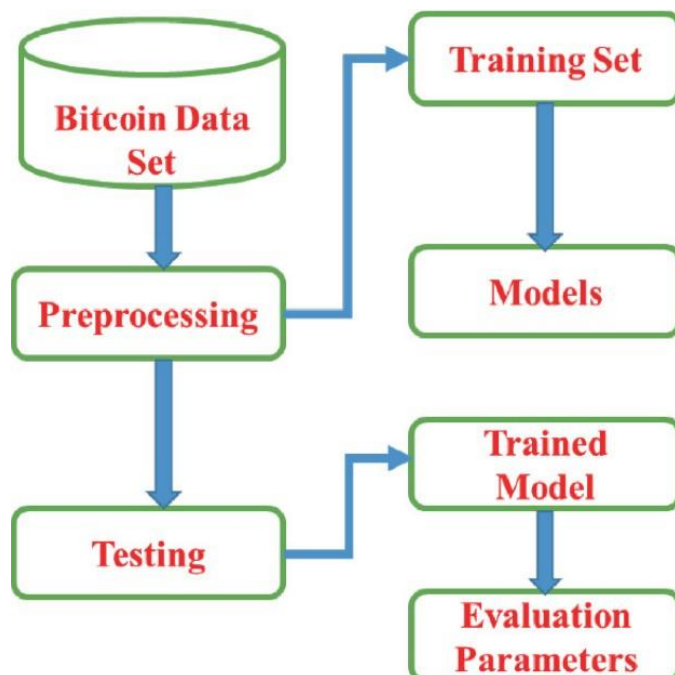
• Acl'ivil'y Diagram

• Flow Charl'
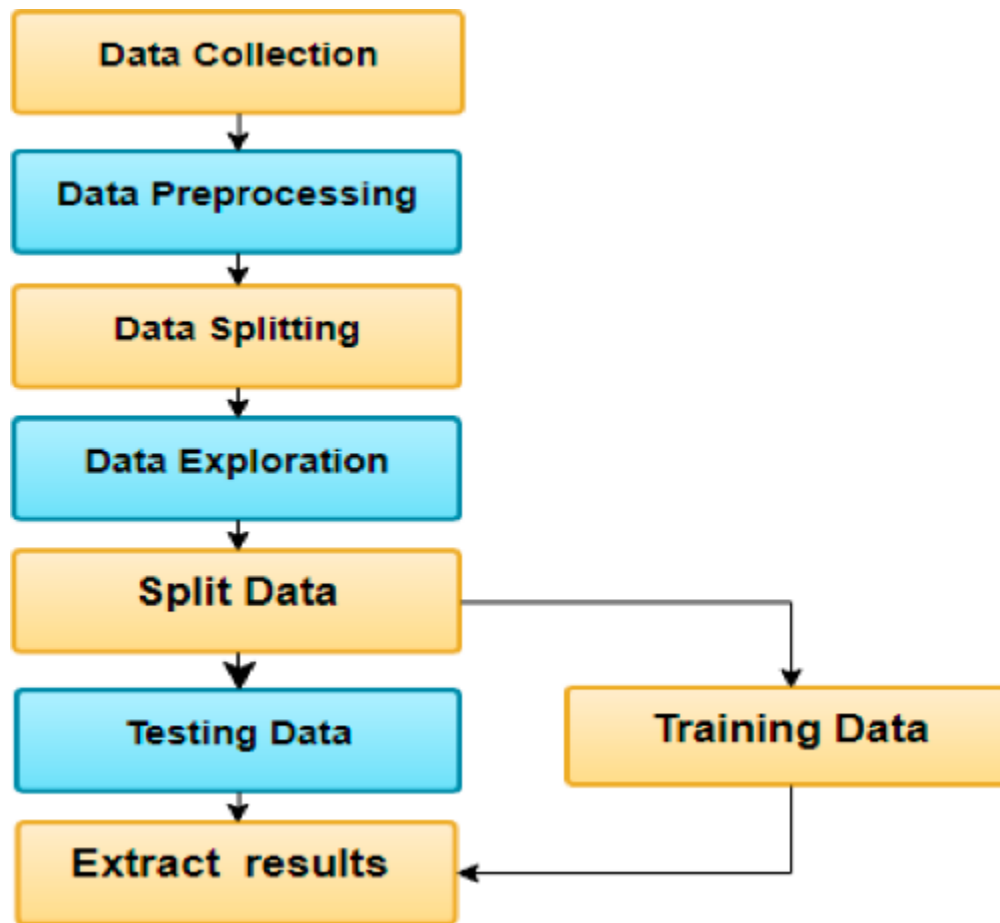
• Archil'ecl'ure oul'line of l'he Time Series Analysis For Bil'coin Price Predicl'ion Using Prophel'

Table-1 : Components & Technologies:

Sr.No	Component	Description	Technology
1	User Interface	Interface allowing user interaction and display of results. For this analysis, it might involve a web-based dashboard or visualization tool for presenting predictions	HTML/CSS/JavaScript for web interfaces, React, Vue.js, or Angular for frontend development,D3.js, or Bokeh for interactive data visualization.
2	Application logic	Core logic handling data flow, processing, and interaction between components	Python (Flask, Django, FastAPI) for APIs, alongside Node.js, manages backend logic, using Celery to handle background processing.
3	Data Base	Persistent storage for structured data. It might store user data, model configurations, or historical Bitcoin price data.	SQL (PostgreSQL, MySQL) and NoSQL (MongoDB, Cassandra) databases manage structured storage, while SQLAlchemy in Python handles.
4	File/Data Storage	Temporary or long-term storage for various data types, including historical Bitcoin price datasets and model parameters.	Cloud-based storage (Amazon S3, Google Cloud Storage) for scalability and object storage services for storing large volumes of historical data.
5	Framework	A framework providing the structure and tools for developing, deploying, and managing the application.	For backend: Flask or Django. TensorFlow/Keras for deep learning, Prophet for time series
6	Deep-Learning Model	In this context, LSTM and CNN are employed for intricate time series prediction tasks, enhancing complex pattern recognition.	TensorFlow for building deep learning modelsLSTM or CNN architectures for time series prediction tasks,
7	Infrastructure (Server/Cloud):	The foundational framework sustaining the entire system encompasses hardware, software components, such as servers, cloud services, and networking elements.	Cloud services (AWS, Azure, GCP) for scalable infrastructure,Kubernetes for container orchestration, virtual machines (VMs) for computation

Table-2: Application Characteristics:

Sr. No.	Component	Description	Technology
1	Open-Source Frameworks	Open-source frameworks offer flexibility, community support, and cost-effectiveness, using available tools for app development and deployment.	Apache Kafka for real-time data, Flask/Django for backend, Prophet for time series, Jupyter for development, Git for version control, and Prometheus/Grafana for monitoring.
2	Security Implementations	Strong security is crucial to protect sensitive data, ensuring privacy and preventing unauthorized access or breaches in the app and its data repositories.	SSL/TLS for secure communication, OAuth/JWT for authentication, RBAC for access control, encryption for data, secure coding, audits, firewalls, IDS, and monitoring tools.
3	Scalable Architecture	A scalable architecture efficiently manages increased loads or data volumes without performance drawbacks, dynamically expanding based on demand.	Cloud services (AWS, Azure, GCP) for scalability, Kubernetes for container orchestration, load balancing, auto-scaling, and microservices for modularity.
4	Availability	High availability means designing a system to reduce downtime, keeping the app accessible even during failures or disruptions for continuous user access.	Component redundancy, multi-region deployment, automated failover, disaster recovery planning, HA clusters, monitoring for issues, and round-the-clock support.
5	Performance	Improving performance means boosting system responsiveness, cutting latency, and using resources efficiently for accurate, timely Bitcoin price predictions.	Caching for frequent data, indexing for speed, database optimization, CDN for content, performance monitoring, profiling, and benchmarking tools.