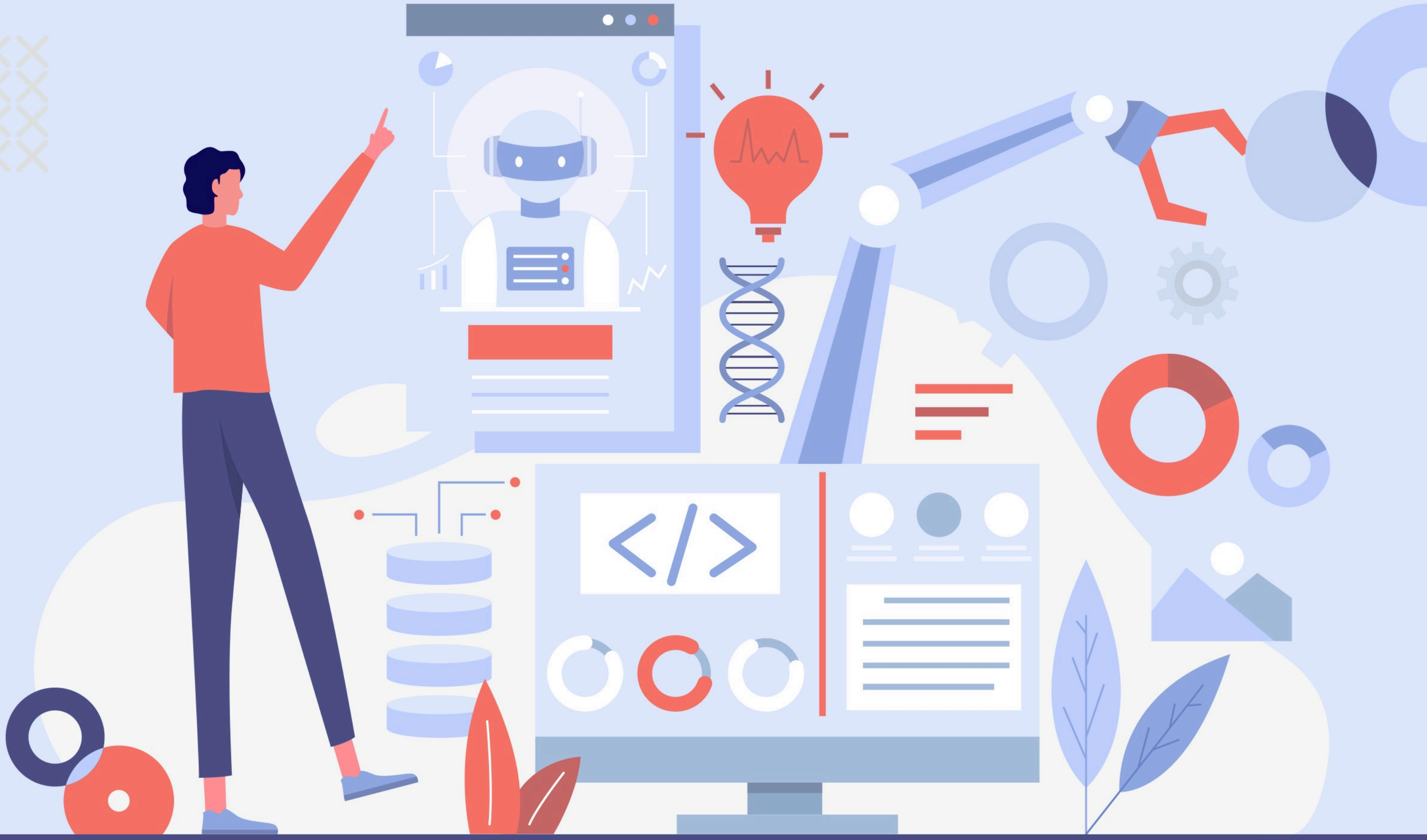
Machine Learning Project

Cryptocurrency Liquidity Prediction for Market Stability



PW SKILLS

Cryptocurrency Liquidity Prediction for Market Stability

Problem Statement

Cryptocurrency markets are highly volatile, and liquidity plays a crucial role in market stability. Liquidity refers to the ease with which assets can be bought or sold without significantly impacting the price. A lack of liquidity can lead to increased price fluctuations and market instability.

In this project, you are required to build a **machine learning model** to predict **cryptocurrency liquidity levels** based on various market factors such as **trading volume, transaction patterns, exchange listings, and social media activity.** The objective is to **detect liquidity crises early** to help traders and exchange platforms manage risks effectively.

Your final model should provide insights into **market stability** by forecasting liquidity variations, allowing traders and financial institutions to make informed decisions.

Dataset Information

You will use a dataset that includes historical cryptocurrency price and trading volume data from below link.
The dataset consists of records from **2016 and 2017.**
Dataset : - *https://drive.google.com/drive/folders/10BRgPip2Zj_56is3DilJCowjfyT6E9AM*

Data Preprocessing Required:

- Handle missing values and ensure data consistency.
- Normalize and scale numerical features.
- Engineer new features related to market liquidity trends.

Project Development Steps

1. **Data Collection:** Gather historical cryptocurrency price, volume, and liquidity-related data.
2. **Data Preprocessing:** Handle missing values, clean data, and normalize numerical features.
3. **Exploratory Data Analysis (EDA):** Analyze data patterns, trends, and correlations.
4. **Feature Engineering:** Create relevant liquidity-related features such as moving averages, volatility, and liquidity ratios.
5. **Model Selection:** Choose appropriate machine learning models such as time-series forecasting, regression, or deep learning approaches.
6. **Model Training:** Train the selected model using the processed dataset.
7. **Model Evaluation:** Assess model performance using metrics such as RMSE, MAE, and R^2 score.

8. **Hyperparameter Tuning:** Optimize model parameters for better accuracy.
9. **Model Testing & Validation:** Test the model on unseen data and analyze predictions.
10. **Local Deployment:** Deploy the trained model locally using Flask or Streamlit for testing.

Expected Deliverables

1. Machine Learning Model

- A trained model that predicts cryptocurrency liquidity.
- Evaluation metrics showing how well the model performs.

2. Data Processing & Feature Engineering

- Cleaned and prepared dataset.
- A brief explanation of new features added.

3. Exploratory Data Analysis (EDA) Report

- Summary of dataset statistics.
- Basic visualizations (trends, correlations, distributions).

4. Project Documentation

- High-Level Design (HLD) Document: Overview of system and architecture.
- Low-Level Design (LLD) Document: Breakdown of how each component is implemented.
- Pipeline Architecture: Explanation of data flow from preprocessing to prediction.
- Final Report: A simple summary of findings, model performance, and key insights.

Guidelines & Submission Requirements

- **Code Documentation:** Ensure all scripts are well-commented and easy to follow.
- **Report Structure:** The report must be structured and should clearly explain the methodology followed.
- **Diagrams & Visuals:** Use appropriate diagrams and plots to explain data processing, model selection, and performance evaluation.
- **Deployment:** If possible, deploy the model using a simple interface (e.g., Streamlit or Flask API) for testing predictions.

Submission Format

- The project must be submitted as a **GitHub repository or a zipped folder** containing:
  - **Source Code**
  - **EDA Report**
  - **HLD & LLD Documents**
  - **Pipeline Architecture and Document**
  - **Final Report**