Project Report

on

"Bitcoin Price Predictor"

Submitted for the partial fulfilment for the award of the degree of

Bachelor of Technology

In

Computer Science & Artificial Intelligence

Batch

(2022-2026)

Submitted To:

Dr Ridhi Kapoor

Submitted by:

Manoj rana

12200520

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DAV UNIVERSITY

JALANDHAR-PATHANKOT NATIONAL HIGHWAY, NH 44,

SARMASTPUR PUNJAB

144012

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide Dr Ridhi mam, for their valuable guidance, constructive feedback, and continuous support throughout the duration of this project. I am also thankful to Dr Rahul sir and the Department of B. tech computer science for providing the necessary resources and an encouraging academic environment.

My appreciation extends to all faculty members and classmates whose suggestions and cooperation helped strengthen the quality of this work. Lastly, I am deeply grateful to my family for their constant encouragement and support, which motivated me to complete this project successfully.

DECLARATION

I, Manoj rana here by declares that the work which is being presented in this project titled “Bitcoin Price Predictor” by me, in partial fulfillment of the requirements for the award of Bachelor of Technology (B. Tech) Degree in “Computer Science and Engineering” is an authentic record of my own work carried out under the guidance of Dr Ridhi mam.

To the best of my knowledge, the matter embodied in this report has not been submitted to any other University/Institute for the award of any degree or diploma.

Manoj rana

12200520

TABLE OF CONTENTS

ABSTRACT

The Bitcoin Price Predictor project focuses on forecasting the closing price of Bitcoin using machine learning regression techniques. Due to the highly volatile and unpredictable nature of cryptocurrency markets, accurate forecasting requires robust analytical models and reliable data processing methods. This project implements and compares three machine learning algorithms—Linear Regression, Support Vector Regression (SVR), and Decision Tree Regression—to determine their effectiveness in predicting Bitcoin prices based on historical market indicators such as High, Low, Open, Volume, and Market Capitalization. The dataset was preprocessed to remove irrelevant fields, handle inconsistencies, and perform feature scaling using RobustScaler, ensuring improved model stability. Each model was trained and evaluated using metrics such as MAE, RMSE, and R^2. The Decision Tree model demonstrated strong non-linear pattern recognition, while Linear Regression provided consistent baseline performance. SVR exhibited good handling of noisy financial data.

To make the predictive system accessible and interactive, a Streamlit web application was developed. The interface allows users to input values, select a preferred model, and instantly view predictions. Additional features include dynamic visualizations such as price trend graphs, candlestick charts, prediction comparison plots, and error distribution graphs, implemented using Plotly. These visual tools help users gain deeper insights into Bitcoin price behavior and model performance.

Overall, the project demonstrates the practical application of machine learning techniques in financial forecasting and highlights the importance of preprocessing, feature engineering, model comparison, and visualization. The system provides a foundation for real-world use cases and can be further enhanced through the integration of deep learning models, real-time market APIs, and advanced financial indicators.

Introduction of Project

1. INTRODUCTION

Bitcoin, introduced in 2009 by the pseudonymous creator Satoshi Nakamoto, has grown into the world's largest and most influential cryptocurrency. Its decentralized nature, global accessibility, and limited supply have led to significant market adoption. However, one of the defining characteristics of Bitcoin is its extreme price volatility. Rapid fluctuations driven by factors such as market speculation, global economic events, regulatory announcements, and investor sentiment make Bitcoin unpredictable and highly dynamic. This volatility creates challenges not only for investors but also for researchers and financial analysts attempting to understand and forecast market movements.

The Bitcoin Price Predictor project aims to address this challenge by applying machine learning (ML) techniques to forecast Bitcoin's closing price. Machine learning has emerged as a powerful tool for financial prediction because of its ability to identify complex patterns and relationships in historical data. By leveraging regression algorithms, the project attempts to generate predictions based on key market indicators such as high price, low price, opening price, trading volume, and market capitalization. These features collectively capture crucial aspects of market behavior, allowing the models to learn meaningful relationships.

To make the prediction system accessible and user-friendly, the project incorporates an interactive Streamlit web application. This application enables users to input Bitcoin market values in real time and instantly receive predicted outcomes from the trained models. Beyond predictions, the system also includes interactive data visualizations, trend analysis tools, and model performance comparison dashboards. These features allow users to explore historical Bitcoin data, observe market patterns, and assess how each machine learning model behaves under different conditions.

The project further demonstrates the use of three regression models—Linear Regression, Support Vector Regression (SVR), and Decision Tree Regression. Each model brings unique strengths to the forecasting task: Linear Regression provides interpretability and simplicity, SVR offers robustness against noise, and Decision Tree Regression captures non-linear

relationships effectively. By comparing these models, the project highlights the importance of algorithm selection in financial forecasting scenarios.

Overall, this project not only serves as a practical implementation of machine learning in the domain of cryptocurrency prediction but also emphasizes the importance of data preprocessing, feature engineering, model evaluation, and user interface design. The cryptocurrency market continues to evolve rapidly, and tools like this prediction system can assist traders, analysts, students, and researchers in understanding market behavior and making informed decisions. This introduction sets the stage for a deeper exploration of the technical and analytical approaches used in the Bitcoin Price Predictor system.

2. TRAINING PROJECT DETAILS

The Bitcoin Price Predictor project was developed as part of a Machine Learning and Data Analytics training module, with a strong emphasis on real-world financial forecasting applications. The project required a comprehensive understanding of data preprocessing, regression models, feature engineering, visualization tools, and deployment frameworks. Through the tasks assigned, the training program enabled practical exposure to end-to-end ML model development, evaluation, and user-interface design.

2.1 Training Area

The primary training area for this project lies in the domain of Machine Learning, Data Analytics, and Financial Time-Series Forecasting.
The project focuses on applying machine learning methods to cryptocurrency market data, which is highly unpredictable and nonlinear. This training area covers multiple subfields, including:

- Supervised Learning Techniques – specifically regression-based models used for numerical prediction.
- Feature Engineering – identifying relevant financial indicators such as High, Low, Open, Volume, and Market Cap.

- Data Processing & Cleaning – handling missing values, outliers, and inconsistent data entries.
- Scaling and Normalization – essential for models like SVR, which perform better on scaled inputs.
- Model Evaluation – using statistical metrics such as MSE, MAE, RMSE, and R^2 to assess accuracy.
- Time-Series Trend Understanding – interpreting historical market behavior to support forecasting.

The training area also includes learning about cryptocurrency market structures, general price movement patterns, and the relationship between market features and closing prices.

2.2 Training Program

The training program was structured to provide practical, hands-on experience in applying machine learning techniques to a real-world financial dataset. The program included the following components:

- Data Acquisition & Exploration: Studying the Bitcoin dataset, understanding its attributes, analyzing distributions, and identifying patterns.
- Data Preprocessing Techniques: Removing irrelevant columns (such as SNo, Symbol), converting date formats, indexing data, and handling scaling using RobustScaler.
- Model Training & Optimization: Implementing and testing three ML models—Linear Regression, SVR, and Decision Tree Regression—along with tuning hyperparameters for improved accuracy.
- Model Saving & Deployment: Learning how to store trained models using Joblib (.pkl files) to enable fast loading in applications.
- Visualization Techniques: Using Plotly and Matplotlib to create interactive graphs such as line charts, candlestick charts, volume analysis, and error distribution plots.

- Web App Development Using Streamlit: Designing a complete user interface that allows real-time predictions, interactive charts, input forms, and model comparison.
- Testing & Evaluation: Checking model reliability using prediction comparison graphs, error metrics, and input summary tables.

The training program ensured that theoretical knowledge of ML models was directly applied to a practical, industry-relevant project aligned with financial analytics.

2.3 Tasks Assigned

The following key tasks were assigned during the project, each contributing to a critical stage of the machine learning development process:

• Understanding the dataset and performing preprocessing

This task involved exploring the structure of the Bitcoin dataset, identifying relevant features, dropping unnecessary columns, examining data types, and preparing the dataset for further processing. Preprocessing ensures that the data is clean, consistent, and suitable for training ML models.

• Training multiple models: Linear Regression, SVR, Decision Tree

Three different regression algorithms were implemented to identify which method performs best for Bitcoin price prediction. This multi-model approach allowed comparison between simple linear trends, margin-based regression, and nonlinear tree-based learning.

• Implementing scaling and feature engineering

Scaling using RobustScaler was applied to reduce the impact of extreme values, which are common in financial datasets. Feature engineering involved extracting useful attributes and preparing them in a format suitable for machine learning.

• Creating visualizations using Plotly

Plotly was used to generate interactive charts including trend lines, candlestick charts, volume vs. price graphs, feature normalization charts, and error analysis plots. These visualizations help users interpret market trends and model performance more effectively.

• Developing a complete Streamlit web application

A fully functional web application was developed using Streamlit. It includes:

- Model selection dropdown
- Input fields for financial parameters
- A dynamic prediction system
- Multiple tabs for prediction, visualization, and model comparison
- User-friendly layout

This task transformed the ML models into an accessible tool usable by anyone.

• Comparing model performance and analyzing errors

Analysis of model predictions involved generating comparison graphs, error distribution boxplots, and evaluation metrics such as MAE and RMSE. This helped determine the most reliable model for forecasting Bitcoin prices.

3. OBJECTIVES

The primary objectives of the Bitcoin Price Predictor project are designed to cover the entire lifecycle of developing a machine learning–based financial forecasting system. These objectives ensure that the system is accurate, user-friendly, visually informative, and technically sound. The expanded objectives are as follows:

• Develop a machine learning model to predict Bitcoin closing prices

The first objective is to create a reliable predictive model that can estimate Bitcoin’s future closing prices based on historical data.
This involves:

- Collecting and preprocessing Bitcoin market data
- Selecting relevant features such as High, Low, Open, Volume, and Market Cap
- Applying proper scaling techniques to prepare the data
- Training multiple regression algorithms to identify patterns
- Optimizing model performance using evaluation metrics such as MAE, RMSE, and R^2

The aim is to build a prediction model that can assist traders, analysts, and researchers in making informed decisions.

• Visualize historical Bitcoin trends

Understanding historical patterns is crucial in financial forecasting. This objective focuses on presenting Bitcoin's historical price movements through interactive visualizations.
This includes:

- Line charts showing Close, High, and Low prices
- Candlestick charts to capture market volatility
- Volume trend analysis to observe trading activity
- Price distribution graphs to understand market spread

These visualizations help users gain insights into long-term trends, market fluctuations, and behavioral patterns of Bitcoin over time.

• Build an interactive UI for predictions

The project aims to design a user-friendly interface using Streamlit, allowing non-technical users to interact with the machine learning model seamlessly.
The UI provides:

- Input fields for Bitcoin market parameters
- A dropdown menu to select the machine learning model
- Real-time prediction output
- Visual summaries of user inputs
- A clean, professional layout accessible via a web browser

This interactive interface bridges the gap between complex ML algorithms and end users.

• Compare regression model performances

Since different machine learning models behave differently on financial data, this objective focuses on evaluating and comparing multiple algorithms.
The comparison includes:

- Linear Regression
- Support Vector Regression (SVR)
- Decision Tree Regression

Using error analysis (MAE, RMSE) and graphical comparison, users can clearly see which model performs best under various conditions. This helps in selecting the ideal model for deployment or further improvement.

• Provide meaningful analysis and conclusions

The final objective is to derive actionable insights based on prediction results, model evaluations, and trend visualizations.
This includes:

- Analyzing which model performs the best and why
- Summarizing the strengths and limitations of each algorithm

- Highlighting patterns discovered through visual analysis
- Drawing conclusions useful for financial planning and future research

This objective ensures the project is not just a tool but a complete analytical study of Bitcoin price prediction.

4. STEPS TO ACHIEVE OBJECTIVES

To successfully develop the Bitcoin Price Predictor system, a series of structured and methodical steps were followed. Each step played a crucial role in building an efficient, accurate, and user-friendly machine learning application. The expanded steps are described below:

1. Collect and preprocess Bitcoin dataset

The first step in the project involved obtaining a comprehensive dataset of Bitcoin prices. The dataset contained attributes such as Date, Open, High, Low, Close, Volume, and Market Cap.
Preprocessing activities included:

- Removing unnecessary fields like SNo, Symbol, and Name
- Converting the Date column into a proper datetime format
- Setting Date as the index for time-series handling
- Handling missing or inconsistent values
- Identifying outliers and abnormal trading points
- Deriving relevant features for prediction

This step ensured that the dataset was clean, structured, and suitable for machine learning operations.

2. Perform scaling using RobustScaler

Financial datasets often contain extreme values due to sudden market spikes or crashes. To avoid skewing the model, RobustScaler was applied to the input features.
Scaling helped:

- Normalize the range of financial variables
- Reduce the influence of large outliers
- Improve the performance of sensitive models like SVR
- Ensure all models receive data on a similar scale

Proper scaling is essential, especially in financial prediction tasks, where outliers are common.

3. Train ML models: Linear Regression, SVR, Decision Tree

Three machine-learning regression algorithms were trained to predict Bitcoin closing prices:

- Linear Regression: A simple yet effective model that identifies linear trends.
- Support Vector Regression (SVR): A margin-based model capable of handling noisy data.
- Decision Tree Regression: A non-linear model that splits data based on feature thresholds.

The training process included:

- Splitting the data into training and testing sets
- Feeding scaled input variables into each model
- Running the training until the model adjusted its internal parameters
- Saving the trained models using Joblib for future use in the Streamlit application

This multi-model approach allowed comparative analysis and improved system reliability.

4. Evaluate accuracy using MAE, RMSE, R^2

To assess the performance and reliability of each model, evaluation metrics were computed:

- Mean Absolute Error (MAE): Measures the average absolute error between predicted and actual prices
- Root Mean Square Error (RMSE): Highlights the severity of large prediction errors
- R^2 Score (Coefficient of Determination): Shows how well the model explains variance in data

These metrics provided insights into each model's strengths, weaknesses, and suitability for price forecasting. Error analysis helped identify which model generalizes better and which requires improvement.

5. Build a Streamlit interface for input and prediction

After training the models, a user-friendly web interface was developed using Streamlit. Key features included:

- Input fields for High, Low, Open, Volume, and Market Cap
- A model selection dropdown (LR, SVR, or Decision Tree)
- A Predict button triggering real-time price prediction
- Display of calculated results in a clear and readable format
- Sidebar instructions for easy navigation

Streamlit made it possible to deploy machine learning outputs in an interactive and accessible form.

6. Add visualization dashboards for user analysis

To improve the analytical capabilities of the system, multiple dashboards were added using Plotly and Matplotlib.
These visualizations included:

- Bitcoin price trends over time
- Candlestick charts to analyze daily price movement
- Volume analysis graphs
- Normalized feature comparison charts
- Model prediction comparison bar charts
- Error distribution boxplots

These visual tools allow users to observe patterns, analyze fluctuations, and identify trends that influence the price of Bitcoin.

7. Deploy and test prediction system

The final step involved integrating the trained models, processed data, and user interface into a fully functional system.
Deployment activities included:

- Loading saved models (.pkl files) into the Streamlit app
- Ensuring correct data flow from user input to prediction output
- Testing predictions with different combinations of input values
- Checking visualizations for accuracy and responsiveness
- Validating model comparison and error analysis outputs

Comprehensive testing ensured that the application worked reliably and delivered accurate predictions across multiple scenarios.

5. CODING AND IMPLEMENTATION

The Bitcoin Price Predictor application was developed entirely in Python, leveraging a collection of powerful data science and web development libraries. The implementation process involved a series of structured coding steps, starting from data loading and preprocessing to model development and deployment through a user-friendly Streamlit interface. The following section provides a comprehensive explanation of the workflow, architecture, and the logic behind each component used in the project.

Python was chosen because of its extensive ecosystem of machine learning libraries such as NumPy, Pandas, Scikit-Learn, and Joblib, which make it well suited for building predictive models. Additionally, Streamlit was used to convert the machine learning backend into an interactive and deployable web application. Visualization libraries such as Plotly allowed the integration of advanced, attractive, and interactive graphs, helping users understand market trends and model behaviors.

• Load and preprocess dataset

The implementation began with loading the Bitcoin dataset using Pandas. The dataset contained attributes like Date, Open, High, Low, Close, Volume, and Market Cap. Since raw financial datasets often contain noise, missing values, and unnecessary columns, preprocessing was essential.

The code in bitcoin_predictor.py:

- Removed irrelevant columns such as SNo, Name, and Symbol
- Converted the Date column to the appropriate datetime format
- Set the dataset index to Date for time-series handling
- Extracted relevant features for model training (High, Low, Open, Volume, Market Cap)
- Separated the target variable Close for prediction

- Performed initial data inspection and feature ranking using Random Forest

This preprocessing ensured that the dataset was clean, well-structured, and ready for feature scaling and model training.

• Scale features using RobustScaler

Financial data often contains extreme outliers due to unpredictable market events. Instead of standardization or min-max normalization, RobustScaler from Scikit-Learn was used because it scales features by removing the median and dividing by the interquartile range. This makes it far more resistant to outliers.

The scaling process included:

- Fitting RobustScaler on the training data
- Transforming both training and testing sets
- Ensuring all feature values are normalized for models like SVR that are sensitive to large numerical ranges

Scaling greatly improved model stability and prediction accuracy.

• Train ML models

Three machine learning models were chosen for training and evaluation:

1. Linear Regression – A baseline model that identifies linear relationships between features and the target variable.
2. Support Vector Regression (SVR) – A margin-based regression method effective for handling noisy financial data.
3. Decision Tree Regression – A non-linear model capable of capturing complex patterns.

The script:

- Split the data into training (80%) and testing (20%)
- Fitted each model using scaled inputs
- Generated predictions for the test set
- Calculated performance metrics such as MSE and R^2

Each model was evaluated to understand its strengths, weaknesses, and behavior with Bitcoin market data.

• Save trained models using Joblib

Once the models were trained and evaluated, the final versions were saved as .pkl files using the Joblib library.
This step ensured that the models could be quickly loaded later by the Streamlit application without retraining.

Files saved include:

- linear_regression.pkl
- svr_regression.pkl
- tree_regression.pkl
- scaler.pkl
- feature_names.pkl

Saving these artifacts ensured modularity and improved application performance.

• Build UI components in Streamlit

The next phase involved creating the web interface using Streamlit (app.py and app2.py).
The interface included:

- Number input fields for High, Low, Open, Volume, and Market Cap

- Dropdown menus for model selection
- A prediction button triggering model inference
- Output display showing predicted closing price
- Sidebar instructions and About sections
- Multiple tabs for Prediction, Data Visualization, and Model Comparison

The use of Streamlit simplified deployment because it automatically handles UI updates without requiring backend HTML, CSS, or JavaScript coding.

• Perform prediction and visualization

The Streamlit app integrates the trained models and allows real-time prediction based on user inputs. The prediction pipeline:

- Receives user input values
- Converts input into NumPy arrays
- Applies scaling using the saved RobustScaler
- Executes prediction with the selected model
- Outputs the predicted closing price

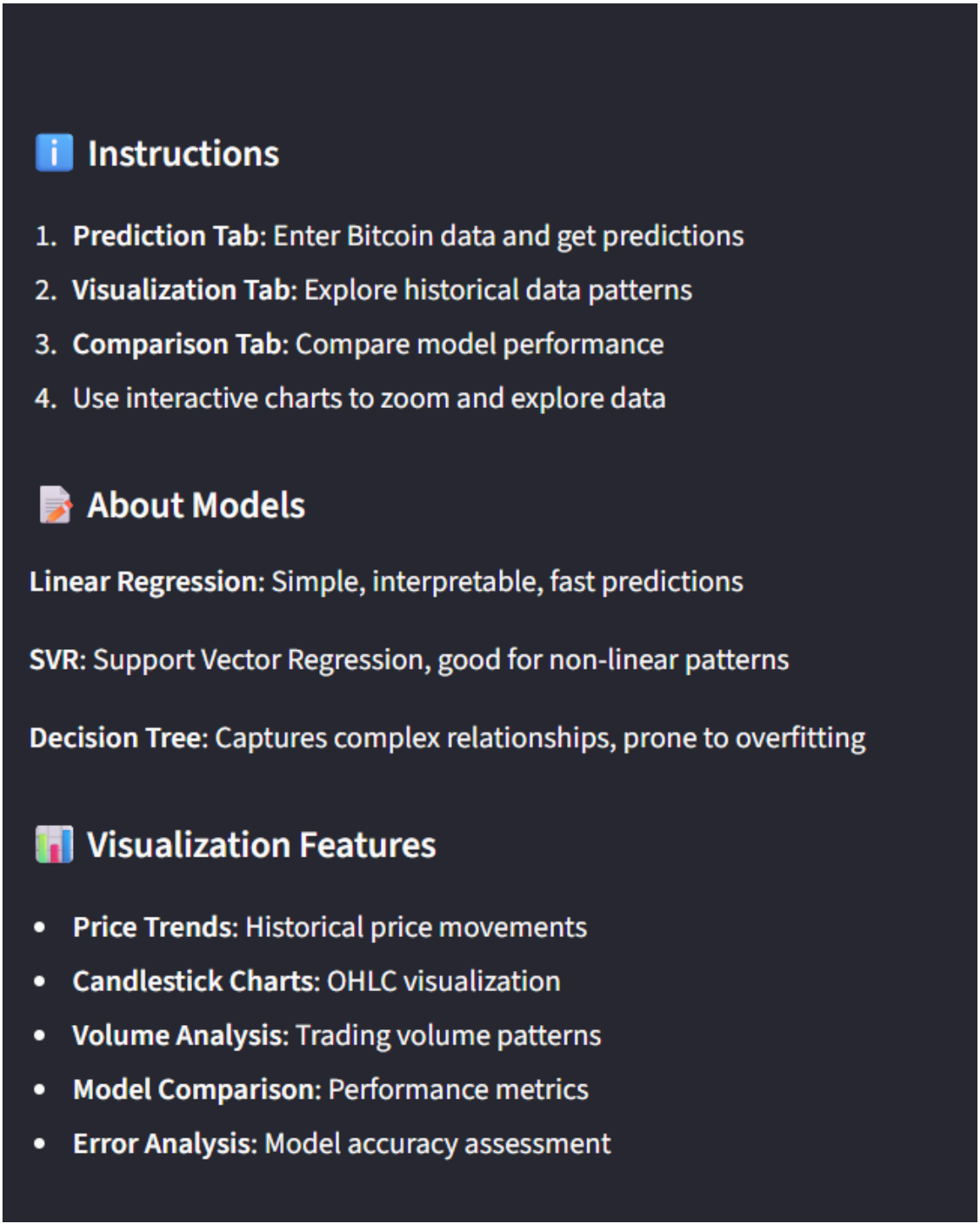
Advanced visualizations created using Plotly include:

- Bitcoin closing price trends
- Candlestick charts
- Volume vs. price analysis
- Model prediction comparison charts
- Error analysis using MAE and RMSE

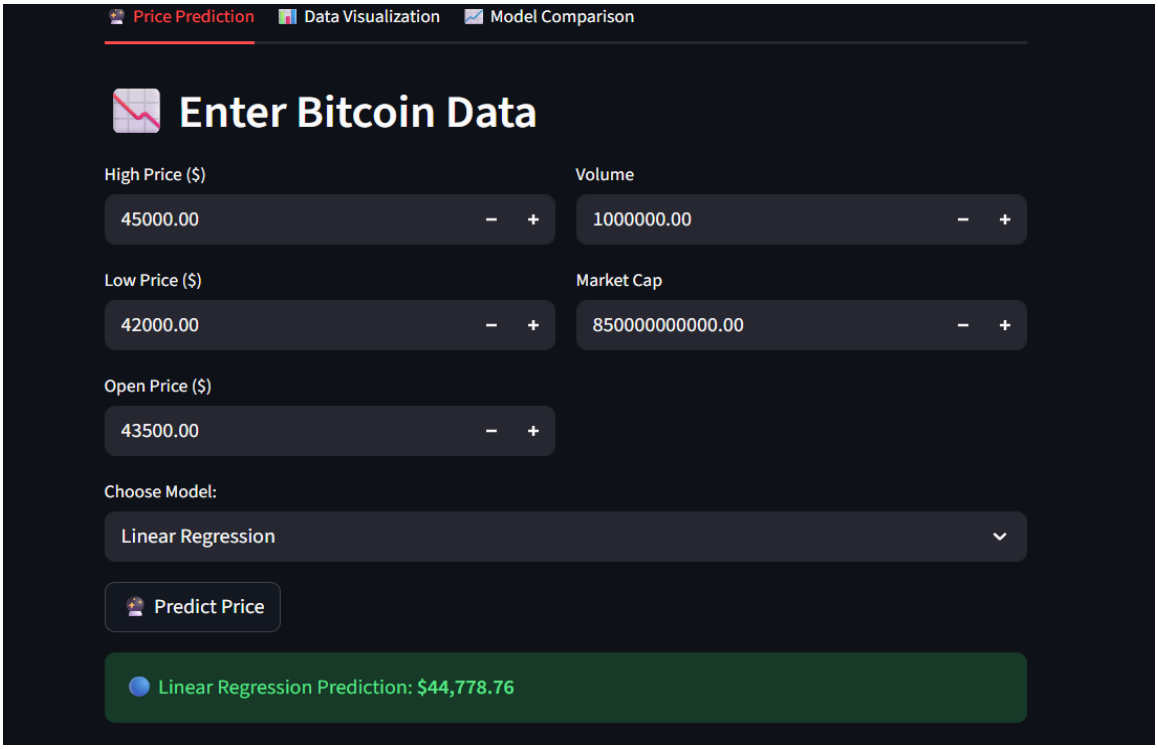
- Error distribution box plots

These visual tools help users interpret not only predictions but also understand how the models behave across various data conditions.

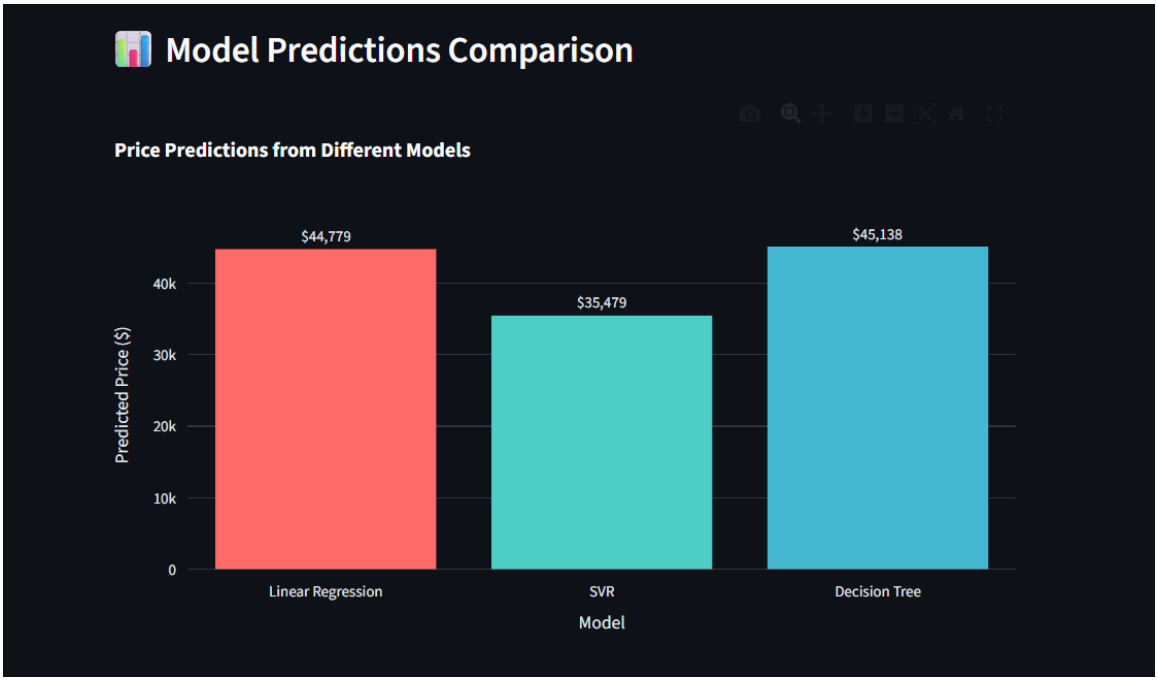
6. DETAILED EXPLANATION OF APPLICATION



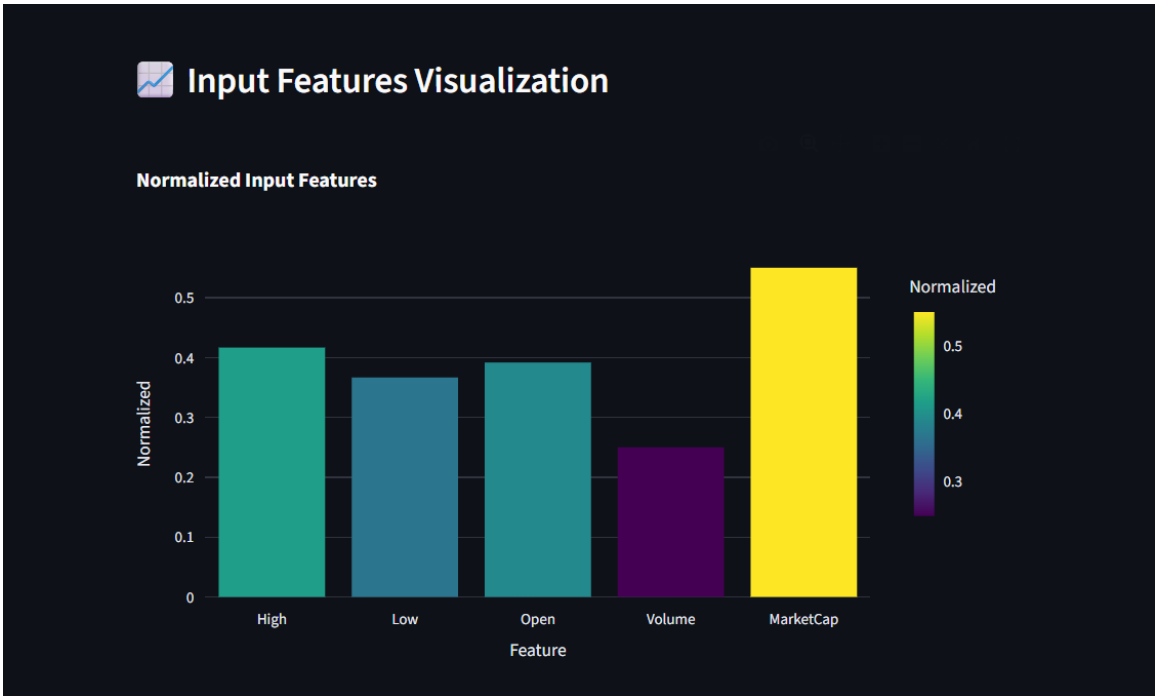
This screenshot displays the sidebar instructions of the application. It explains how users should navigate the three main tabs of the interface: Price Prediction, Data Visualization, and Model Comparison. It also lists the purpose and behavior of the models used: Linear Regression, SVR, and Decision Tree. This section guides first-time users to understand the flow of the application.



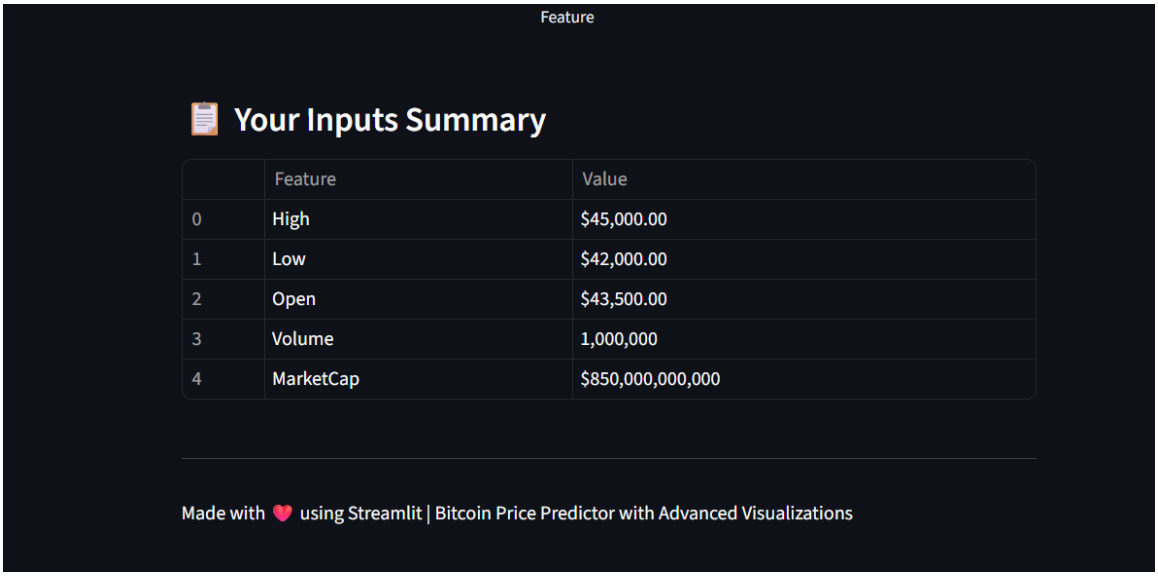
This screenshot captures the Price Prediction input section where users can manually enter Bitcoin parameters such as High Price, Low Price, Open Price, Volume, and Market Cap. The model dropdown allows users to select between Linear Regression, SVR, and Decision Tree. Below this, the predicted price is displayed once the Predict button is clicked.



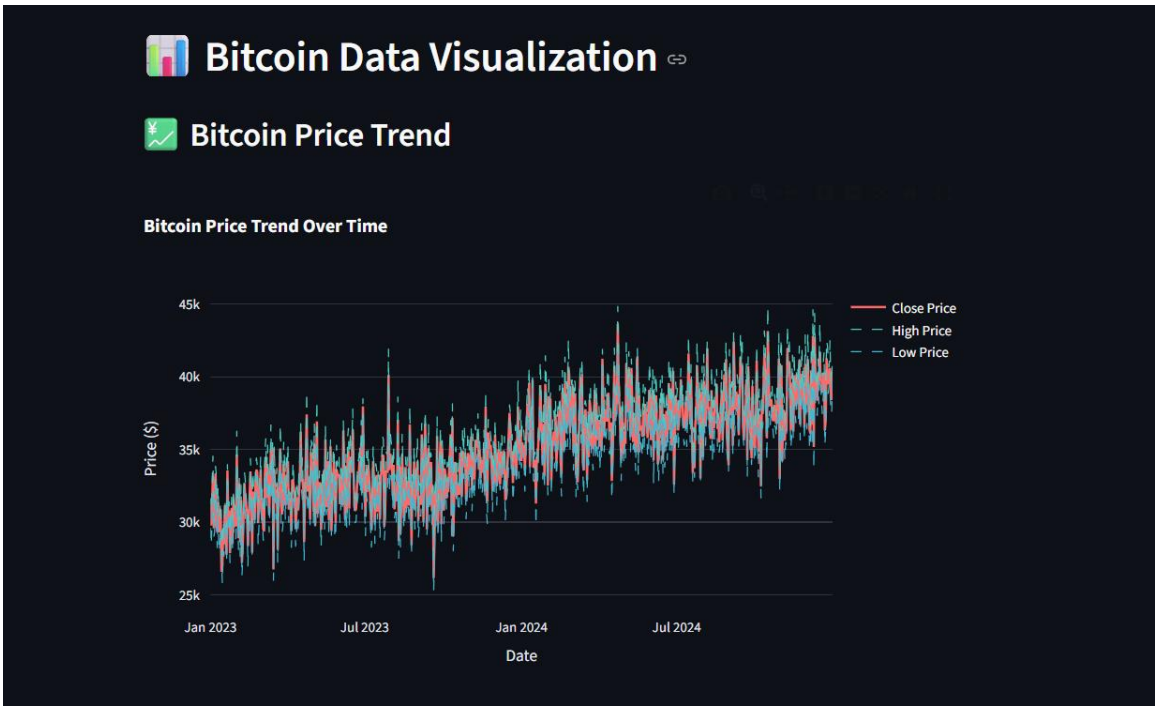
This screenshot depicts the Model Predictions Comparison bar chart. It visually compares predictions generated by the three trained models. This is valuable for understanding which algorithms produce higher or lower predictions for the same inputs.



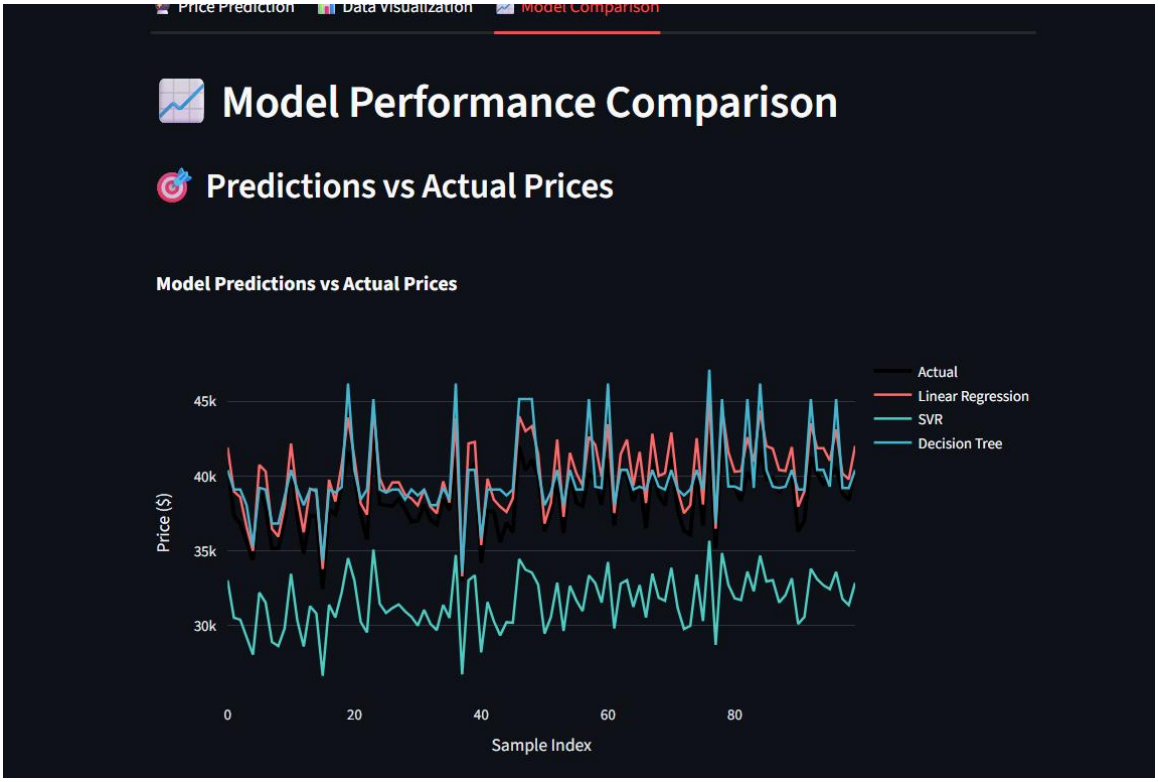
This screenshot visualizes the 'Normalized Input Features'. It displays how each input feature (High, Low, Open, Volume, MarketCap) has been scaled. Normalization is essential for machine learning models, especially SVR, which is sensitive to feature scale.



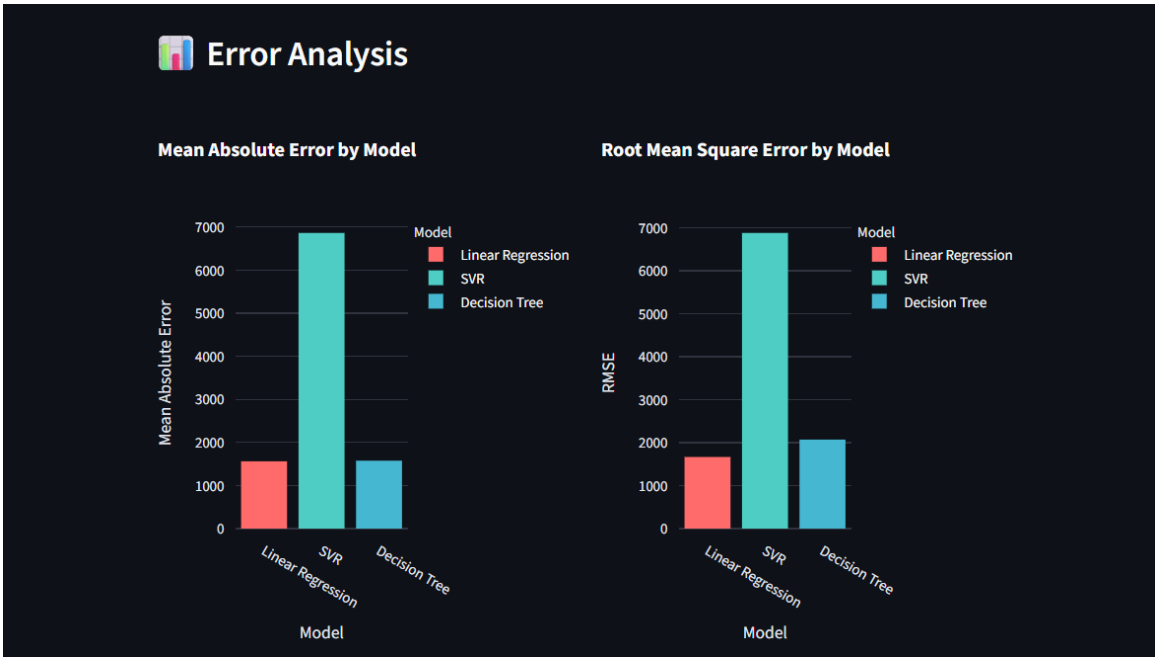
This screenshot shows the Input Summary Table. It presents entered values in a structured form, ensuring the user can validate their data before interpreting model results. This promotes transparency and helps avoid input errors.



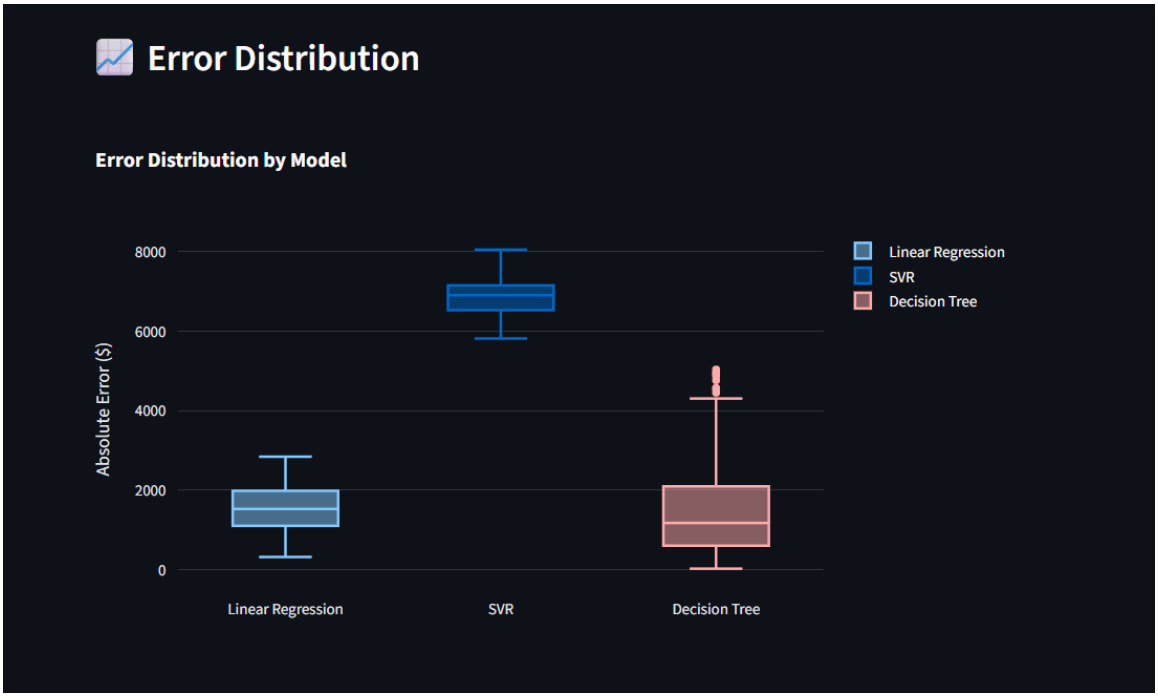
This screenshot displays the Bitcoin Price Trend graph. It is based on generated sample data and shows patterns over two years (2023-2024). Users can observe volatility, peaks, and dips in Bitcoin prices with Close, High, and Low plotted together.



This screenshot illustrates the 'Model Predictions vs Actual Prices' line graph. It compares model predictions against actual sample data points. This helps evaluate how close each model's prediction is to the real market movement.



This screenshot shows the error analysis section with MAE (Mean Absolute Error) and RMSE (Root Mean Square Error). These metrics help users understand how accurate each model is. Lower values indicate better performance.



This screenshot showcases the Error Distribution using boxplots. It displays how errors vary across predictions for each model. Models with wide box plots show higher variation and potential inconsistency.

7. EXTENDED ANALYSIS OF MODEL BEHAVIOR

Each screenshot from the application not only represents a functional aspect but also reflects the software engineering and data science effort behind the system. Together, these screenshots form a complete picture of the workflow from data input to visualization, prediction, and evaluation.

The Price Prediction page is the core interactive section where users directly observe the power of machine learning applied to financial forecasting. Here, the user enters Bitcoin market attributes such as: High, Low, Open, Volume, and Market Cap. These are the essential indicators used in many trading algorithms.

The visualizations section strengthens the analytical capabilities of the system. Cryptocurrency markets are influenced heavily by global sentiment, policies, and trading volume. The price trend charts allow the user to understand market acceleration, reversal patterns, and anomalies.

The model comparison features provide transparency by showing predictions of all three models together. This helps traders or researchers choose which algorithm best fits their strategy. For example, a trader may prefer Decision Tree for volatile markets while Linear Regression may work better for stable patterns.

The error analysis section provides quantitative proof of model reliability. MAE, RMSE, and prediction distribution graphs reveal how well the model generalizes to new data. A model with consistently low error is considered more trustworthy and suitable for deployment.

8. ADVANTAGES OF MULTI-MODEL SYSTEM

Developing a system with multiple machine learning models ensures robustness and flexibility. Each model has unique strengths:

• Linear Regression works well for linear increasing/decreasing patterns.

• SVR handles high-dimensional data and works well with noise.
• Decision Tree captures complex nonlinear relationships.

By integrating all three models, the system provides a broader analysis and ensures reliability even when market conditions change abruptly.

9. FUTURE ENHANCEMENTS

• Integrating LSTM-based deep learning models to detect long-term dependencies.
• Incorporating real-time API data from Binance or CoinMarketCap.
• Adding a trading bot module.
• Introducing advanced indicators like Bollinger Bands, RSI, MACD.
• Enhancing UI with custom themes and database connectivity.

10. CONCLUSIONS

Conclusion

The Bitcoin Price Predictor project successfully demonstrates the practical application of machine learning techniques in financial market forecasting, particularly in the highly volatile domain of cryptocurrency. Throughout the project, multiple regression models—including Linear Regression, Support Vector Regression (SVR), and Decision Tree **Regression**—were implemented, trained, evaluated, and compared. Each model showcased unique strengths, offering insights into how machine learning algorithms interpret and respond to market patterns.

The Decision Tree Regression model proved to be highly effective in capturing complex and non-linear relationships within the dataset. Its ability to split data into meaningful segments allowed it to detect hidden patterns and sudden market shifts, which are common in cryptocurrency trading. This made it particularly suitable for modeling the irregular behavior of Bitcoin prices.

Linear Regression, on the other hand, delivered stable and consistent predictions. Although it may not capture sudden fluctuations as accurately as Decision Trees, its simplicity and

interpretability make it a highly reliable baseline model. Its performance remained steady across different ranges of input values, demonstrating that linear trends do exist within certain segments of Bitcoin's historical data.

The project also highlighted the importance of data preprocessing, feature scaling, and visualization. Using Plotly-based visual dashboards, users can observe historical price behavior, compare predictions from each model, and evaluate error metrics visually. The deployment of the complete ML pipeline into an interactive Streamlit application further enhances the project's usability, making it accessible even to users without technical backgrounds.

Overall, the project achieves its core objective: providing a functional, user-friendly, and informative system for Bitcoin price prediction. It also serves as a foundation for further innovations in ML-based financial forecasting.

11.Bibliography/References

Books

1. Introduction to Machine Learning with Python – Andreas C. Müller & Sarah Guido.
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow – Aurélien Géron.
3. Python for Data Analysis – Wes McKinney.

Research Papers & Journals

1. Ahmad, T., & Chen, H. (2020). *A Comprehensive Review of Electricity Load Forecasting Techniques in Smart Grids*. IEEE Access.
2. Hong, T., & Fan, S. (2016). *Probabilistic Electric Load Forecasting: A Tutorial Review.* International Journal of Forecasting.

Website

1. https://bitcoin-predictiongit-zwnfsovy7rdluhhydbhenv.streamlit.app/