**Project 5: Bitcoin Price Prediction Using Machine Learning**

**Project Overview**

In this project, you will build a machine learning model to predict Bitcoin prices based on historical data. The primary goal is to understand how time series forecasting techniques and regression models can be applied to predict cryptocurrency prices. You will work with a dataset that contains historical Bitcoin prices, including features like open, high, low, close prices, trading volume, and more. By the end of this project, you will learn data preprocessing, regression modeling, and time series forecasting for Bitcoin price prediction.

**Project Steps**

1. **Dataset Overview and Loading**
   * Load the dataset containing historical Bitcoin data into your Python environment.
   * Explore the dataset to understand the features available, such as **date**, **open**, **high**, **low**, **close**, and **volume**.
   * Perform initial exploratory data analysis (EDA) to identify any trends, seasonality, or patterns in the data.
2. **Data Preprocessing**
   * **Handle Missing Values**: Check for any missing values and handle them appropriately, either by **imputation** or by removing incomplete records.
   * **Feature Engineering**: Create new features such as moving averages, daily returns, and other time-based features to improve the model's performance.
   * **Data Splitting**: Split the dataset into:
     + **Training Data (80%)**: For training the machine learning models.
     + **Testing Data (20%)**: For evaluating the performance of the model.
3. **Model Selection and Training**
   * Choose appropriate regression algorithms for predicting Bitcoin prices. Suggested models include:
     + **Linear Regression**
     + **Decision Tree Regressor**
     + **Random Forest Regressor**
     + **XGBoost Regressor**
   * Train each model on the training dataset and perform **hyperparameter tuning** to optimize their performance.
4. **Time Series Forecasting (Optional)**
   * Apply time series forecasting techniques such as **ARIMA** or **LSTM** (Recurrent Neural Network) for advanced prediction of Bitcoin prices.
   * Compare the performance of traditional regression models with time series models to understand their effectiveness in predicting cryptocurrency prices.
5. **Model Evaluation**
   * Evaluate the performance of each model using metrics like **Mean Absolute Error (MAE)**, **Mean Squared Error (MSE)**, and **R-squared**.
   * Plot **predicted vs. actual prices** to visually assess how well the model predicts Bitcoin price movements.
   * Compare different models and select the best one based on evaluation metrics.
6. **Output and Analysis**
   * Generate predictions for the test dataset and plot them against the actual prices.
   * Provide a detailed analysis of each model's performance, including strengths and weaknesses.
   * Discuss the impact of various features, including newly engineered ones, on the model's predictive capability.

**Expected Output**

* A Jupyter notebook containing:
  + Steps for data preprocessing, feature engineering, and time series analysis.
  + Model training and evaluation for linear regression, decision tree, random forest, and XGBoost.
  + Visualizations of Bitcoin price trends, feature importance, and predicted vs. actual prices.

**Techniques to Learn**

* **Data Preprocessing**: Handling missing values, feature engineering, and dataset splitting.
* **Time Series Analysis**: Understanding trends, seasonality, and time-based feature creation.
* **Regression Models**: Training linear regression, decision tree, random forest, and XGBoost models.
* **Model Evaluation**: Calculating metrics like MAE, MSE, and R-squared to evaluate model performance.
* **Time Series Forecasting**: Applying ARIMA or LSTM for advanced forecasting.

**Deliverables**

1. **Jupyter Notebook**: A well-documented notebook showing the entire process, including code, visualizations, and explanations.
2. **Project Report**: A PDF report summarizing your approach, techniques used, findings, model evaluation, and visualizations.