# Project Report: Bitcoin Price Prediction using Machine Learning

## 1. Introduction and Objectives for the Project

Building a machine learning model to forecast the price of Bitcoin using many variables, including open, high, low, and volume, is the goal of this project. Predicting Bitcoin prices is an important task for investors, financial experts, and tech enthusiasts because of the market's high level of volatility and interest. This research involves using machine learning methods, namely Random Forest and Linear Regression models, to predict Bitcoin prices using historical data with modern day cloud platform system such as AWS.

## 2. Reasoning for Choosing this Dataset.

The dataset for this project comes from Yahoo Finance, which offers historical market data for Bitcoin (BTC) with daily price information. It includes columns such as 'Open', 'High', 'Low', 'Close', 'Volume', and 'Date'. Covering a period from 2017 to the present, this dataset contains over a million rows, making it a great fit for machine learning applications.

- **Relevance**: Bitcoin is one of the most traded cryptocurrencies, and studying its past values can reveal important information about market volatility and patterns.
- **Size**: With more than a million rows, the dataset is sufficiently large to be appropriate for machine learning methods that need a lot of data for training.
- **Availability**: Yahoo Finance makes the data publicly available, which facilitates processing and access.

## 3. Data Processing and Transformation Steps

The dataset required several preprocessing steps before being used in machine learning models:

- **Preparing Dataset:**
    - The dataset was initially extracted from the Yahoo API and stored as raw data in an S3 bucket. Using AWS Athena, the dataset was cleaned, and the refined version was then uploaded back to the S3 bucket as the cleaned dataset, which was subsequently used for building the machine learning model.

- **Data Cleaning**:
    - Missing values were handled by dropping rows with missing data, ensuring that all columns contained valid entries for model training and then it was stored in s3 bucket.
- **Feature Engineering**:
    - Feature columns such as 'Open', 'High', 'Low', and 'Volume' were selected to predict the 'Close' price (target variable).
    - In some cases, additional features could be engineered, like lag features or moving averages, but for simplicity, the primary features were used.

- **Normalization**:
    - The feature data was standardized using StandardScaler from the scikit-learn library to ensure that all features contributed equally to the model. This is especially important for algorithms like linear regression.
- **Splitting the Data**:
    - The dataset was split into training and test sets using a train-test split (80% training and 20% testing). This ensures that the model is tested on unseen data to evaluate its performance.
- **Feature Vectorization**:
    - The features were combined into a single vector column using ***VectorAssembler*** in PySpark, allowing the model to handle multiple features simultaneously.

## 4. Machine Learning Model Development

The machine learning models developed for this project include:

- **Linear Regression Model**:
    - A **Linear Regression** model was built using Sagemaker AI and trained using features like 'Open', 'High', 'Low', and 'Volume' to predict the 'Close' price of Bitcoin.
    - **Model training** was performed using ***scikit-learn*** in JupyterLab (on SageMaker AI), and the model performance was evaluated using ***Root Mean Squared Error (RMSE)***.
- **Random Forest Model**:
    - A **Random Forest** model was also implemented using Pyspark to predict the Bitcoin price. Random Forest models are generally more powerful than Linear Regression for non-linear relationships between variables but since in our case we use on 10 numtree we get the very high RMSE if we change that to 100 to 500 it will outperform linear regression model the only drawback is that it will take much more time to process the data.

## 5. Evaluation and Results

The models were evaluated based on their ability to predict the 'Close' price of Bitcoin using several metrics:

- **Linear Regression**:
    - The model achieved a ***Root Mean Squared Error (RMSE*)** of 20.53 on the test set, indicating the average deviation of the predicted values from the actual values.
- **Random Forest**:
    - The Random Forest model did not show better ***Root Mean Squared Error (RMSE*)** of 620.27 compared to Linear Regression, because we are saving computation, but it will outperform in future if we change the numtree from 10 to 100 or in between 100-500.

Both models showed promise in predicting Bitcoin prices, but the Linear Regression model performed better in capturing complex relationships between features.

*Roughly, linear regression model's predictions are off by about **20.53 USD** from the actual Bitcoin prices.*

*Roughly Random Forest model's predictions are off by about **620.27 USD** from the actual Bitcoin prices.*

*If the Bitcoin price is 40,000 USD, for example, my model's predicted price would typically be within a **± 20.53 USD** range of the actual price and same goes with the RMSE of random forest model, but the difference is huge.*

***Lower RMSE values** indicate better performance, while higher RMSE values suggest that the model's predictions are not very accurate, but this can be change with different techniques.*

## 6. Challenges Faced and How They Were Addressed

- **Data Preprocessing**:
  - The dataset contained missing values that needed to be handled before training. This was solved by using simple techniques like dropping missing rows.
- **Model Overfitting**:
  - Linear Regression tended to overfit due to the simplicity of the model. To address this, a Random Forest model was used, which is less prone to overfitting.
- **Model Selection**:
  - Deciding between Linear Regression and Random Forest models was a challenge due to the non-linear relationships present in the data (overall for the data).
- **Distributed Computing Benefits**:
  - PySpark was used to train the Random Forest model, allowing the data to be processed in parallel across multiple nodes, improving computational efficiency and scalability.

### 8. Conclusion and Future Work

This project demonstrates the potential of machine learning in predicting Bitcoin prices. The Linear Regression outperformed Random Forest model in this task, showcasing the importance of using.

Future work can focus on:

- ***Hyperparameter tuning*** for better model performance.
- ***Time series analysis*** to account for sequential dependencies in Bitcoin prices.
- ***Incorporating external features***, such as market sentiment data, to improve prediction accuracy.

## **ARCHITECTURE DIAGRAM** *(using LucidChart):*