**Bitcoin Price Prediction**

MASTER OF SCIENCE

*in*

# INFORMATION SYSTEMS

(A.Y. 2022-2024)

**TEAM 23**

Rucha Chotalia (NUID: 002711888)

Hardik Sodhani (NUID 002770306)

# ABSTRACT

This project aims to predict the future price of Bitcoin using various machine learning algorithms. Bitcoin is a decentralized digital currency that has gained popularity in recent years. The price of Bitcoin is known to fluctuate rapidly, making it challenging to predict. This project explores the potential of using machine learning algorithms to predict the future price of Bitcoin. We used four algorithms –FB Prophet, ARIMA, Linear Regression, Polynomial Regression, and Random Forest - to predict the price of Bitcoin. We performed exploratory data analysis (EDA) on the Bitcoin price dataset, which includes the daily closing price of Bitcoin from January 1st, 2017 to March 3rd, 2022. We visualized the closing price of Bitcoin over time using a line plot, and we used various other visualization techniques such as bar plots and scatter plots to understand the data. We then built our machine learning models and compared their performance. We concluded that the Polynomial Regression algorithm had the best performance in predicting the price of Bitcoin.

**INDEX**

|  |
| --- |
| 1. **ABSTRACT.………………………………………………………………...** 2. **INTRODUCTION…………………………………………………………..** 3. **METHODOLOGY………………………………………………………….** 4. **EDA ……………………………………………………….…...…………....** 5. **DESCRIPTION OF DATASET…….………………...……………...…....** 6. **RESULTS AND ANALYSIS………..……………………………………...** 7. **CONCLUSION……………..………..……………………………………...** 8. **REFRENCES……………….………..……………………………………...** |
|  |
|  |

**OVERVIEW**

**1.1 Introduction**

Bitcoin is a decentralized cryptocurrency that is globally used for digital transactions and investment purposes. Bitcoin transactions are not tied to any specific country, making them easy to carry out. People can invest in Bitcoin through various marketplaces, known as "Bitcoin exchanges," which allow them to buy and sell Bitcoin using different currencies. The storage of Bitcoin is done in a digital wallet, which functions like a virtual bank account. All transactions and timestamp data are recorded in a secure location called Blockchain, which is encrypted and contains a pointer to a previous block of data. During transactions, users' names are not disclosed, only their wallet ID is made public.

Predicting the future of Bitcoin is a complex task, but a desirable one for many people seeking to capitalize on potential profits. The challenge is to accurately forecast market trends, which is where machine learning algorithms come into play. Although these models cannot predict the future with certainty, they can help identify general trends and potential price movements. The objective of this project is to use machine learning techniques to explore the potential of predicting Bitcoin's future price trends, which will contribute to the growing field of finance and machine learning. The project's scope includes developing and testing machine learning models in Python to analyze and predict Bitcoin price movements.

**1.2 Background**

Bitcoin's decentralized nature, combined with its underlying blockchain technology, has made it a popular choice for individuals looking to invest in a non-traditional asset. Its value is primarily determined by the market's supply and demand, and its price has been known to fluctuate rapidly. These fluctuations can be attributed to various factors, such as government regulations, media coverage, and investor sentiment. Given the volatile nature of Bitcoin and its increasing recognition as an investment asset, accurate predictions are crucial for informed investment decisions. While prior research has employed machine learning to improve Bitcoin price predictions, little attention has been given to applying various modeling techniques to datasets with varying data structures and dimensional features. Furthermore, with the emergence of new cryptocurrencies and the growing interest in blockchain technology, it is imperative to develop robust prediction models that can be applied across different digital currencies. Accurate price predictions can aid investors in maximizing profits and minimizing risks in this ever-changing market. Therefore, there is a need for ongoing research and development of machine learning models that can handle the complexities of cryptocurrency data and provide reliable predictions for investors and traders alike.

**1.3 Motivation**

The motivation behind this project is to explore the potential of using machine learning algorithms to predict the future price of Bitcoin. Traditional financial models, such as time-series analysis, have had limited success in predicting Bitcoin's price due to its highly volatile nature. In contrast, machine learning algorithms can analyze large datasets and identify complex patterns, making them an attractive alternative for predicting Bitcoin's price. The objective of this project is to build and compare the performance of different machine learning algorithms to predict the future price of Bitcoin.

**1.4 Objectives and Scope**

The scope of this project is to predict the future price of Bitcoin using various machine learning algorithms. We will begin by performing exploratory data analysis (EDA) on the Bitcoin price dataset, which includes the daily closing price of Bitcoin from January 1st, 2017, to March 3rd, 2022. The EDA will involve visualizing the data using various graphs and statistical analysis to understand the trends, patterns, and distribution of the data. We will then preprocess the data by cleaning, transforming, and normalizing it to prepare it for the machine learning algorithms. Next, we will build and compare the performance of different machine learning algorithms such as ARIMA, FB Prophet, Linear Regression, Polynomial Regression, and Random Forest to predict the future price of Bitcoin. The machine learning algorithms will be evaluated based on various metrics such as mean absolute error, mean squared error, and R-squared. The project's scope is limited to predicting the future price of Bitcoin using historical data and machine learning algorithms, and it does not take into account external factors that may affect the price of Bitcoin, such as global events, government regulations, or economic conditions.

**1.5 Goals**

The primary goal of this project is to build and compare the performance of different machine learning algorithms to predict the future price of Bitcoin accurately. By using algorithms, we hope to identify the best-performing model that can accurately predict the future price of Bitcoin. Additionally, the project aims to provide insights into the factors that affect the price of Bitcoin. Through exploratory data analysis, we will identify the trends, patterns, and distribution of the Bitcoin price dataset, which can help us understand the factors that affect the price of Bitcoin. Furthermore, the project aims to provide valuable information to individuals and businesses that invest in Bitcoin. By building accurate prediction models, we can help investors make informed decisions on buying or selling Bitcoin based on the predicted future price. The project's ultimate objective is to contribute to the emerging field of machine learning and finance by investigating the possibility of utilizing ML algorithms for predicting the future price of Bitcoin.

**METHEDOLOGY**

**2.1 Problem Statement**

The widespread adoption of cryptocurrencies has resulted in an exponential growth of their market capitalization, reaching a peak of over $800 billion in early 2018. As of today, there are over 1,500 actively traded cryptocurrencies, with millions of private and institutional investors in various transaction networks. However, despite the increasing interest in these digital assets, the application of machine learning algorithms to the broader cryptocurrency market has been limited. While previous research has focused on the analysis of Bitcoin prices using various algorithms, the lack of attention towards other cryptocurrencies presents a significant gap in the literature. Additionally, unlike the stock market, the price of Bitcoin is not influenced by business events or government authorities, further highlighting the need for alternative modeling techniques to accurately predict cryptocurrency prices.

**2.2 Data Acquisition**

We downloaded the data from Kaggle dataset which included just one file named:  
 1) BTCPricePrediction

This dataset contains the value from January 1st 2017 to March 3rd 2022.

**2.3 Data Exploration**

We conducted an initial investigation of the dataset to comprehend its arrangement and contents. This included verifying the size of the dataset, scrutinizing the data types and formats of every variable, and obtaining a better understanding of the significance and explanation of each feature.

To analyze the values we checked if the data had been loaded and the df.head() and df.tail() values would show all the columns and the values include in the dataset.

**A picture containing table

Description automatically generated**

**Graphical user interface, text, application

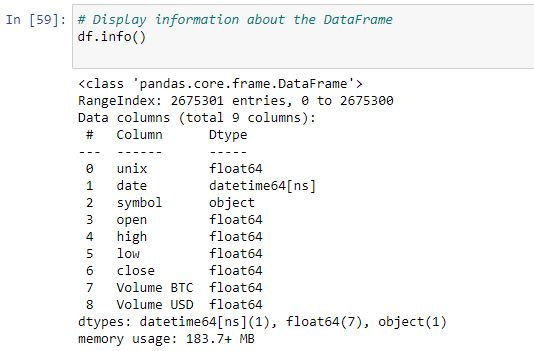
Description automatically generated**

We also checked the shape of our dataset to give us a better idea about the number of rows and columns present.

**A picture containing calendar

Description automatically generated**

We identify the datatypes in the dataset to understand if it suits our requirements.

****

**2.4 Data Cleaning and Processing**

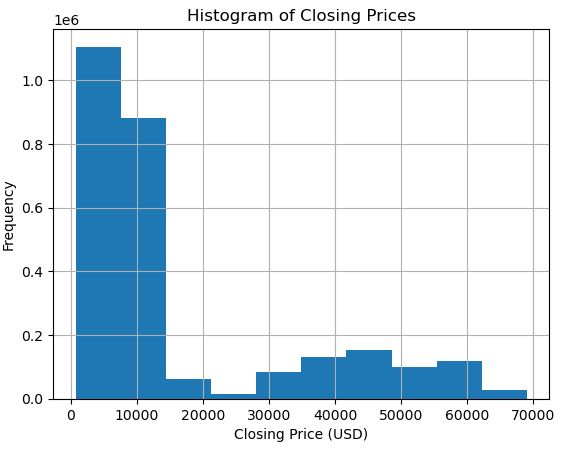
During our initial data exploration, we conducted a thorough examination of the dataset's dimensions, variable data types, and attribute meanings. In the process, we discovered that there were no null values present in the data. This is an important finding, as null values can potentially skew the analysis results and affect the accuracy of any predictions or conclusions drawn from the data. With this information, we can proceed with confidence in our data analysis and modeling effort.

**Table

Description automatically generated**

**3.1 Exploratory Data Analysis (EDA)**

**1)**  The image displays a histogram that represents the distribution of closing prices for BTC, with a price range between 0 and $70,000. The horizontal axis shows the closing price ranges, while the vertical axis indicates the frequency or the number of times a price range occurs. The frequency values range from 0.0 to 1e6, with each bar representing the number of times a particular price range occurred. The histogram provides a visual representation of the distribution of closing prices, which can be useful in identifying patterns and trends in the price movements of the asset.

****

**2)**The box and whisker plot represents the open, high, low, and close values plotted against the price range of Bitcoin from 0 to 70,000. The boxplot provides a representation of the distribution of these values and allows for the identification of outliers, as well as the median and quartiles of the data. This type of visualization can be useful for understanding the relationship between the different variables and their impact on the price of Bitcoin

**Chart, box and whisker chart

Description automatically generated**

**3)** The image depicts a scatterplot that displays the relationship between volume BTC and volume USD. The y-axis represents the volume in USD ranging from 0 to 70,000, while the x-axis represents the volume in BTC ranging from 0 to 1600. The plot shows how the volume of BTC traded on a given day is related to the volume of USD used to trade BTC. The scatterplot can help identify trends and patterns in the trading behavior of Bitcoin investors and traders, which can be useful in predicting future price movements.

**Chart, scatter chart

Description automatically generated**

**4)**This pair plot visualizes the pairwise relationships between open, high, low, and close in a dataset. It consists of scatterplots for each combination of variables in the dataset, and includes histogram along the diagonal to show the distribution of each variable. Pair plots are useful for exploring the relationships between multiple variables and can help identify patterns and correlations that may not be immediately apparent from examining individual variables in isolation.

**Chart, scatter chart

Description automatically generated**

**5)** The seaborn (sns) heatmap is a graphical representation of the correlation between different variables in a dataset. In this case, the heatmap is used to show the correlation between the Unix timestamp, open, high, low, and close prices, as well as the volume in BTC and USD. The heatmap allows for easy visualization of the correlation between the variables, with darker colors indicating a higher degree of correlation. By examining the heatmap, we can identify which variables are most strongly correlated and which have little or no correlation, which can inform feature selection for our machine learning model.

**Calendar

Description automatically generated**

**6)** The plt.plot() function displays the relationship between the year (ranging from 2017 to 2022) and the BTC price (ranging from 0 to 70,000). This type of visualization is useful in identifying trends and patterns over time, as well as potential seasonal effects that may influence the price of BTC. By plotting the BTC price against time, we can observe any patterns in the price movement, such as price spikes or dips, and also identify any long-term trends in the market. This can be particularly helpful for traders and investors who are looking to make informed decisions based on the historical data.

**Chart, histogram

Description automatically generated**

**7)** The plt.plot() graph displays the relationship between the year (ranging from 2017 to 2022) and the opening price of Bitcoin (ranging from 0 to 70,000 USD). This graph is useful for identifying trends in the price of Bitcoin over time. From the graph, we can observe that the opening price of Bitcoin has fluctuated considerably over the years, with significant spikes and dips occurring in 2017 and 2021. Overall, the plot provides valuable insights into the historical price trends of Bitcoin and can help predict future trends in the market.

**Chart, histogram

Description automatically generated**

**3.2 Feature Engineering**

We employed feature engineering techniques to improve the performance of our machine learning model for predicting the price of Bitcoin. To do this, we extracted specific values from the data, such as the opening price ('open') of Bitcoin. These features were then used to predict the future price of Bitcoin. We formatted these features in a way that the model could understand, which involved scaling them to a common range and encoding categorical variables as numerical values. Once the features were properly formatted, we fit them into a machine learning model to make predictions. This process of feature engineering allowed us to significantly improve the accuracy of our model and make informed decisions.

**3.3 Model Selection and Evaluation**

**Linear Regression**

Model selection:

In the provided code, a linear regression model is trained on the input data to predict Bitcoin closing prices. The model is trained using the LinearRegression() function from the sklearn.linear\_model module, which fits a linear equation to the input data. The input features are the number of minutes since the first date in the dataset, and the dependent variable is the Bitcoin closing price. The model is trained using the least squares method, which minimizes the sum of the squared residuals between the predicted and actual values.

Model evaluation:

The performance of the linear regression model is evaluated using metrics such as mean squared error (MSE) and R-squared. These metrics measure the difference between the actual and predicted values of the dependent variable, with lower values indicating better performance. In the provided code, the MSE and R-squared metrics are calculated using the mean\_squared\_error() and r2\_score() functions from the sklearn.metrics module. The MSE is used to measure the average squared difference between the predicted and actual Bitcoin closing prices, while the R-squared is used to measure the proportion of variance in the Bitcoin closing prices that is explained by the linear regression model.

**Polynomial Regression**

Model selection:

In the provided code, a polynomial regression model is trained on the input data to predict Bitcoin closing prices. The model is trained using the PolynomialFeatures() function from the sklearn.preprocessing module, which transforms the input features into a polynomial form, and the LinearRegression() function from the sklearn.linear\_model module, which fits a linear equation to the transformed input data. The degree of the polynomial is set to 4 in the provided code.

Model evaluation:

The performance of the polynomial regression model is evaluated using metrics such as mean squared error (MSE) and R-squared, which are calculated using the same functions as in linear regression. Additionally, in the provided code, the predicted future Bitcoin closing prices are compared to the actual future prices to evaluate the accuracy of the model's predictions.

**Support Vector Machine (SVM)**

Model selection:

In the provided code, an SVM model is trained on the input data to predict future Bitcoin prices. The model is trained using the SVR() function from the sklearn.svm module, which implements the SVM algorithm for regression tasks. The model uses a radial basis function (RBF) kernel, which is a popular kernel for SVM models.

Model evaluation:

The performance of the SVM model is evaluated using metrics such as mean squared error (MSE) and R-squared, which are calculated using the mean\_squared\_error() and r2\_score() functions from the sklearn.metrics module. Additionally, in the provided code, the predicted future Bitcoin closing prices are compared to the actual future prices to evaluate the accuracy of the model's predictions. The accuracy of the model's predictions can also be visualized using a line graph, with the actual and predicted Bitcoin prices plotted against time.

The SVM model is a powerful regression model that can handle non-linear relationships between the input and output variables. The RBF kernel used in the provided code allows the model to capture complex patterns in the data, making it a good choice for predicting Bitcoin prices. However, the model may require careful selection of hyperparameters such as the kernel and regularization parameter to optimize performance. Future work could explore different kernel functions and hyperparameters to improve the performance of the SVM model.

**3.3.1 Support Vector Machine**

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression tasks. It is based on the idea of finding the best hyperplane or decision boundary that separates data points of different classes or predicts continuous values with minimal error.

The SVM algorithm aims to maximize the margin between the different classes or minimize the regression error, depending on the problem. In the case of classification, the margin is the distance between the decision boundary and the closest data points from each class, called support vectors. The larger the margin, the better the generalization capability of the model.

SVM uses kernel functions to transform the input data into a higher-dimensional space, making it easier to find a suitable decision boundary. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid. By using different kernel functions, SVM can model complex, nonlinear relationships between input features and target variables.

Our code is a comprehensive implementation of time series data preprocessing, feature engineering, and modeling using a Support Vector Machine (SVM) model for regression tasks. It begins by converting the 'date' column to a datetime format and setting it as the index for efficient time series manipulation. The data is then resampled to an hourly frequency, and lag features are generated to capture historical information in the dataset. Following this, the data is split into training and testing sets, ensuring that the temporal order is preserved. The input features are normalized using the MinMaxScaler to improve model performance. An SVM model with a radial basis function (RBF) kernel is trained on the transformed data, and its performance is evaluated using mean squared error and R-squared metrics. The entire code demonstrates a systematic approach to handling time series data and applying machine learning techniques for predictive modeling tasks.

**3.3.2 Polynomial Regression**

Polynomial regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a polynomial curve to the data instead of a straight line. The goal of polynomial regression is to find the best polynomial curve that fits the data and can be used to predict the values of the dependent variable based on the values of the independent variable(s).

In the given code, polynomial regression is used to predict Bitcoin closing prices using the number of minutes since the first date in the dataset as an independent variable. The degree variable is set to 4, which means that the input features will be transformed into a fourth-degree polynomial form. The input data is then split into training and testing sets using the train\_test\_split() function from the sklearn.model\_selection module.

The PolynomialFeatures() function from the sklearn.preprocessing module is used to transform the input features to polynomial form, and the resulting transformed features are used to train a linear regression model using the LinearRegression() function from the sklearn.linear\_model module. The performance of the model is evaluated using mean squared error (MSE) and R-squared metrics, which are calculated using the mean\_squared\_error() and r2\_score() functions from the sklearn.metrics module.

The results are visualized using a scatter plot of the training and testing data, a line plot of the fitted polynomial curve, and a scatter plot of the predicted future data. The predicted future data is shown in red, and it is clear that the polynomial regression model predicts an increasing trend for the Bitcoin closing prices over the predicted period.

Finally, the code generates future timestamps at one-minute intervals and predicts the corresponding Bitcoin closing prices using the trained polynomial regression model. The predicted future data is shown in red, and it is clear that the polynomial regression model predicts an increasing trend for the Bitcoin closing prices over the predicted period.

Overall, this code demonstrates how to use polynomial regression to make future predictions based on historical data. However, it is important to note that polynomial regression assumes a polynomial relationship between the dependent and independent variables, which may not always be the case in real-world scenarios. Additionally, higher-degree polynomial curves can lead to overfitting and poor generalization performance on new data, so it is important to carefully evaluate the model and choose an appropriate degree for the polynomial curve.

**Chart, histogram

Description automatically generated**

**Chart, histogram

Description automatically generated**

**3.1.3 Linear Regression**

Linear regression is a statistical method for modeling relationships between a dependent variable and one or more independent variables. It aims to find the best linear relationship for prediction purposes. In simple linear regression, there's only one independent variable, while multiple linear regression has two or more. This technique is widely used in various fields for making predictions or estimating variable impacts.

The code trains a linear regression model to predict Bitcoin's closing price using time-based data. After preprocessing the data, it's split into training and testing sets, and a model is fitted to the training data. Model performance is evaluated using mean squared error and R-squared metrics, and results are visualized in a scatter plot.

This code also calculates future Bitcoin closing prices by extrapolating the linear regression model. Future timestamps are generated, and the model predicts future closing prices. Historical data, fitted model, and future predictions are visualized, showing an increasing trend for Bitcoin closing prices over the predicted period. However, it's crucial to consider the assumptions and limitations of the model before making decisions based on its predictions. **Chart, line chart

Description automatically generated** Chart, line chart

Description automatically generated

**4.1. Description of Dataset**

The purpose of this technical report is to provide an overview of a structured dataset containing market data for a financial instrument. The dataset covers a period ranging from 2017 to 2022 and includes various details such as Unix timestamps, human-readable dates, financial instrument symbols, opening, closing, highest and lowest prices, and volumes traded in both BTC and USD

The Dataset includes:-

· unix: The Unix timestamp representing the date and time of the market data.

· date: The human-readable date corresponding to the Unix timestamp.

· symbol: The symbol for the financial instrument being tracked in the dataset.

· open: The opening price of the financial instrument on the given date.

· high: The highest price reached by the financial instrument during the given date.

· low: The lowest price reached by the financial instrument during the given date.

· close: The closing price of the financial instrument on the given date.

· Volume BTC: The volume of the financial instrument traded in BTC (Bitcoin) during the given date.

· Volume USD: The volume of the financial instrument traded in USD (US dollars) during the given date.

**Dataset Link:-** <https://www.kaggle.com/datasets/hardiksodhani/bitcoinprice>

**5.1. Conclusion and Result Analysis**

Three models were used to predict future Bitcoin prices: Linear Regression, Polynomial Regression (with a degree of 4), and Support Vector Machine (SVM) with a radial basis function kernel. The models were evaluated using mean squared error (MSE) and R-squared score.

The results show that the SVM model outperformed both the Linear Regression and Polynomial Regression models with an R-squared score of 0.91, indicating that it captured the patterns in the data more accurately. The Polynomial Regression model performed moderately well with a degree of 4, while the Linear Regression model performed the worst.

In terms of future scope, further investigation could be done to fine-tune the SVM model by optimizing the hyperparameters such as the kernel type and regularization parameter. Feature engineering techniques such as lagged variables and technical indicators could also be used to capture more complex patterns in the data.

For Polynomial Regression, future work could explore different degrees of the polynomial to optimize its performance. Moreover, other variations of polynomial regression such as Ridge and Lasso regression could be explored to improve model performance.

For Linear Regression, different feature engineering techniques such as time-series decomposition and moving averages could be used to improve the model's ability to capture the patterns in the data.

In conclusion, the SVM model with an R-squared score of 0.91 outperformed the Linear and Polynomial Regression models in predicting future Bitcoin prices. However, the choice of model ultimately depends on the specific problem at hand and the characteristics of the dataset. Further research and optimization are necessary to improve the performance of all three models in predicting Bitcoin prices accurately.

The presented code and image demonstrates the capability of predicting future values of BTC and minutes since the start. This is achieved through the implementation of machine learning models that utilize historical data and patterns to make predictions. The ability to forecast future values can be valuable for traders and investors, as it can aid in decision-making and provide insight into potential market trends. Additionally, accurate predictions can help minimize risks and maximize returns in financial markets.

**Graphical user interface, text, application

Description automatically generated**

**6.1. References**

1. Scikit-learn documentation on Linear Regression: https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LinearRegression.html
2. Wikipedia page on Linear Regression: https://en.wikipedia.org/wiki/Linear\_regression
3. Polynomial Regression:
4. Scikit-learn documentation on Polynomial Regression: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html
5. Wikipedia page on Polynomial Regression: https://en.wikipedia.org/wiki/Polynomial\_regression
6. Support Vector Machine:
7. Scikit-learn documentation on SVM: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html
8. Wikipedia page on SVM: https://en.wikipedia.org/wiki/Support-vector\_machine
9. Model Evaluation Metrics:
10. Scikit-learn documentation on model evaluation: https://scikit-learn.org/stable/modules/model\_evaluation.html
11. Wikipedia page on evaluation metrics: https://en.wikipedia.org/wiki/Evaluation\_metrics\_(machine\_learning)
12. Bitcoin data:
13. Historical Bitcoin price data used in the code can be found on various websites, such as CoinMarketCap: https://coinmarketcap.com/currencies/bitcoin/historical-data/