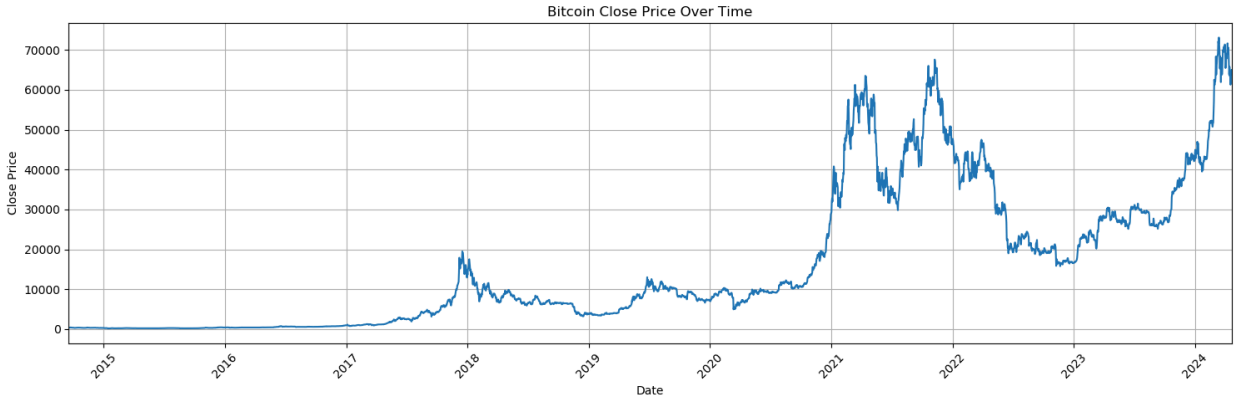
Hok Wai Chan
ICS635
5/11/2024

# Bitcoin Prediction

Bitcoin is digital money that can be bought, sold, and exchanged directly without needing to go through a bank or payment processor. It is decentralized, meaning it is not controlled by any single authority like a government. Instead, all Bitcoin transactions get recorded on a public digital ledger called the blockchain. The problem we are trying to solve is to predict the daily price movements of Bitcoin, specifically whether the closing price of Bitcoin on the following day will be higher or lower than the current day's closing price. This predictive model can be used by investors, traders, and financial institutions to make informed decisions about buying or selling Bitcoin based on the forecasted price trends. Here is an image illustrating the trend of trading using the closing price of Bitcoin:




The input to the model consists of various features related to Bitcoin's daily trading data, such as the opening price, high price, low price, closing price, and trading volume. Here is the first three rows of the dataset:

| | Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|---|
| 0 | 2014-09-17 | 465.864014 | 468.174011 | 452.421997 | 457.334015 | 457.334015 | 21056800 |
| 1 | 2014-09-18 | 456.859985 | 456.859985 | 413.104004 | 424.440002 | 424.440002 | 34483200 |
| 2 | 2014-09-19 | 424.102997 | 427.834991 | 384.532013 | 394.795990 | 394.795990 | 37919700 |

Hok Wai Chan
ICS635
5/11/2024

The output is a binary classification, where the model predicts whether the next day's closing price will be higher ("Buy" - labeled as 1) or lower ("Not Buy" - labeled as 0) than the current day's closing price.

We expect machine learning to be an appropriate approach for this problem because it can effectively capture the complex, non-linear relationships between the input features and the target variable next day's price movement. Other traditional statistical methods may struggle to model the complex patterns and dynamics present in Bitcoin's unstable pricing data. Additionally, machine learning algorithms can adapt and learn from the extensive historical trading data available, covering a decade. The dataset used for training and evaluation contains over 3505 rows of daily Bitcoin pricing data, providing a huge amount of samples for the model to learn from. Shape of the dataset:

```
Shape of the dataset: (3505, 7)
```

The dataset used in this study is a comprehensive collection of Bitcoin's daily valuation against the US dollar over the decade from 2014 to 2024. The dataset covers Bitcoin's trading history, from its basic start trading around a few hundred dollars, through multiple upswings and downturns reaching highs over $60,000. Each row of the dataset represents a trading day. It includes the following features:

- Date: Date of the trading session
- Open: The first price at which Bitcoin trades when an exchange opens on a trading day
- High: The highest price at which Bitcoin trades during a trading session

- Low: The lowest price at which Bitcoin trades during a trading session
- Close: Price at which Bitcoin trades at the end of the trading session
- Adj Close: Closing price after adjustments for all applicable splits and dividend distributions
- Volume: Total number of Bitcoin coins traded

To tackle this problem, we have employed a variety of machine learning models, each with its own strengths and inductive biases, to explore their effectiveness in predicting the daily price movements of Bitcoin. The chosen models are Logistic Regression, Gradient Boosting, K-Nearest Neighbors (KNN), Random Forest, and XGBoost. All models will be trained from scratch:

- Logistic Regression: Picked for its simplicity and because it has built-in ways to avoid overfitting (modeling the noise too closely).
- Gradient Boosting: Chosen because it can effectively capture complicated, non-linear patterns in the data, which may be useful for predicting Bitcoin prices.
- K-Nearest Neighbors (KNN): Selected for its simplicity and ability to identify local patterns in the data, which could help predict price movements.
- Random Forest: Included because it is robust to noise in the data and can handle datasets with many features, making it a good fit for this problem.
- XGBoost: Chosen because it has advanced techniques to prevent overfitting, can process data efficiently in parallel, and could be effective for tackling complex data problems.

Hok Wai Chan
ICS635
5/11/2024

By employing a diverse set of models, we aimed to leverage their collective strengths and explore different approaches to capture the complicated patterns in Bitcoin's pricing data. Additionally, we developed an ensemble model that combines the predictions of these individual models using a simple averaging approach.

The data went through several preprocessing steps:

- Handling missing data: Missing data were checked in the dataset, and there were no missing values as showed.

```
Missing values in the dataset:
Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

- Feature removal: The 'Adj Close' column was dropped since it contained the same information as the 'Close' column.
- Feature split: The 'Date' column was split into separate 'Year', 'Month', and 'Day' features to use for better understanding outliers.
- Outlier detection and handling: Outliers (data points that significantly differ from other observations in a dataset) in the dataset were identified using the boxplot method, which revealed significant outliers in the years 2021 and 2024. Row removal was attempted initially, which improved performance. However, consider that could lead to data loss.

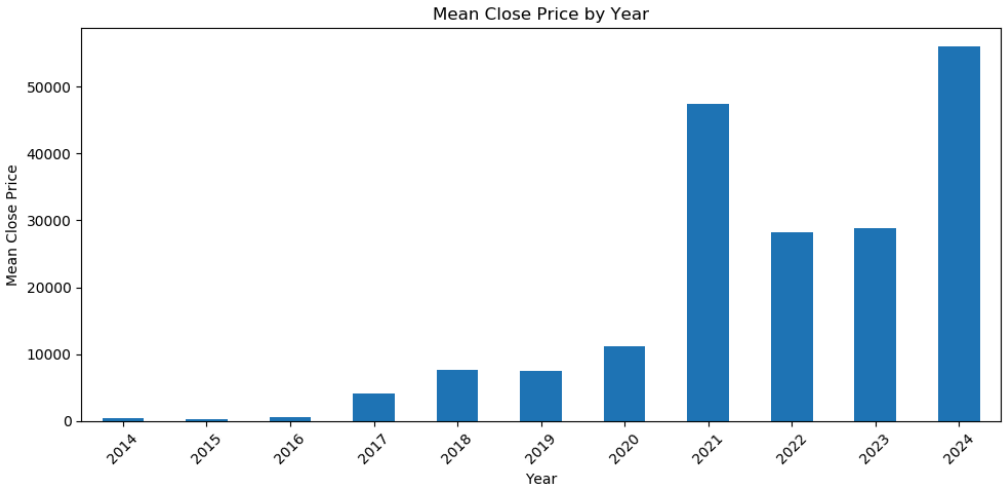
Hok Wai Chan
ICS635
5/11/2024

Instead, re-assignment with thresholds using the Interquartile Range (IQR) method was applied. Values below the lower threshold were replaced with the lower threshold value, and values above the upper threshold were replaced with the upper threshold value.

Calculate outliers:

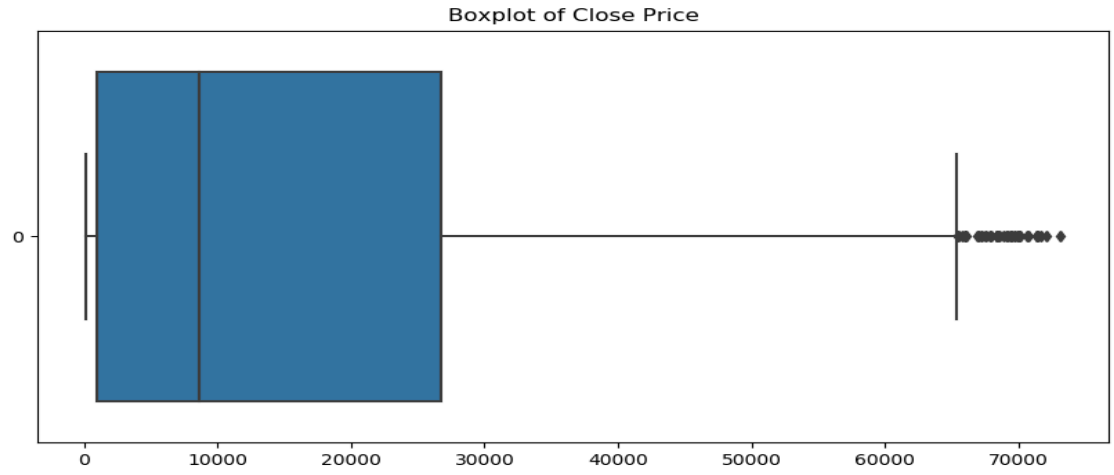
To identify outliers in a dataset, the interquartile range (IQR) method is commonly used. This method involves several steps. First, we find the first quartile (Q1) and the third quartile (Q3) of the dataset. Then, the IQR is calculated by subtracting Q1 from Q3. With the IQR, lower and upper limits for outliers are established. The lower limit is set at Q1 minus 1.5 times the IQR, and the upper limit is set at Q3 plus 1.5 times the IQR. Any data point that falls below the lower limit or above the upper limit is flagged as an outlier.

Image showing 2021 and 2024 the price of Bitcoin is higher compared with other years:



Hok Wai Chan
ICS635
5/11/2024

Boxplot method to show outliers:




- Additional column creation: A new column 'Is_Outlier' is created to indicate whether a data point was an outlier or not (0 for non-outlier, 1 for outlier). Before the re-assignment, we can see there are outliers remaining in the dataset.

```
Date,Open,High,Low,Close,Volume,Buy_or_Not,Year,Month,Day,Is_Outlier
2024-03-17,65316.34375,68845.71875,64545.31640625,68390.625,44716864318,0.0,2024,3,17,1
2024-03-18,68371.3046875,68897.1328125,66594.2265625,67548.59375,49261579492,0.0,2024,3,18,1
2024-03-19,67556.1328125,68106.9296875,61536.1796875,61912.7734375,74215844794,1.0,2024,3,19,0
2024-03-20,61930.15625,68115.2578125,60807.78515625,67913.671875,66792634382,0.0,2024,3,20,1
```

- Target variable creation: A new column 'Buy_or_Not' was added to represent the target variable for the prediction task. This column was calculated by comparing the closing price of the current day with the closing price of the next day. If the next day's closing price was higher, the value was set to 1 (indicating a "Buy" signal), and if the next day's closing price was lower, the value was set to 0 (indicating a "Not Buy" signal).

After processing the features used for training the models are (Open, High, Low, Close, and Volume), and the feature used for prediction is Buy_or_Not.

Hok Wai Chan
ICS635
5/11/2024

Cross-validation was applied, and the dataset was divided into three subsets (the total dataset contained 3505 examples):

- Training set: 50% of the data
- Validation set: 20% of the data
- Test set: 30% of the data

The models were trained on the training set, while the validation set was used for hyperparameter optimization and model selection. The test set was held out and used for the final evaluation of the selected models to obtain an unbiased estimate of their performance on unseen data. This approach ensures that the test set is truly held out and never used for any decision-making or model tuning, mitigating the risk of overfitting and providing a reliable estimate of the model's generalization performance.

For the Logistic Regression model, we performed a grid search over the following hyperparameters:

- C: Inverse of regularization strength (explored values: [0.001, 0.01, 0.1, 1, 10, 100])
- Penalty: Regularization type (explored values: ['l1', 'l2'])
- Solver: Algorithm to use for optimization (explored values: ['liblinear', 'saga'])

For the other models (Gradient Boosting, KNN, Random Forest, and XGBoost), we attempted manual hyperparameter tuning by trying different combinations of hyperparameters. However,

the results indicated that these models were still overfitting and do not have a significant improvement, as evidenced by the big difference between the AUROC scores on the training and validation sets. Due to the overfitting issues observed in the other models and the relatively simpler nature of the Logistic Regression model, we focused primarily on the Logistic Regression model for further analysis and evaluation.

Gradient Boosting:

- AUROC on validation set: 0.4958032972900814
- AUROC on training set: 0.8058714299368042

KNN:

- AUROC on validation set: 0.5036877586437057
- AUROC on training set: 0.7591955417337539

Random Forest:

- AUROC on validation set: 0.4839891202776665
- AUROC on training set: 0.9962735750468166

XGB:

- AUROC on validation set: 0.49727172607128556
- AUROC on training set: 0.9878938452021228

Hok Wai Chan
ICS635
5/11/2024

Ensemble:

- AUROC on validation set: 0.4889700974502737

Logistic Regression:

- AUROC on validation set: 0.5193899345881725
- AUROC on training set: 0.5123120114187407

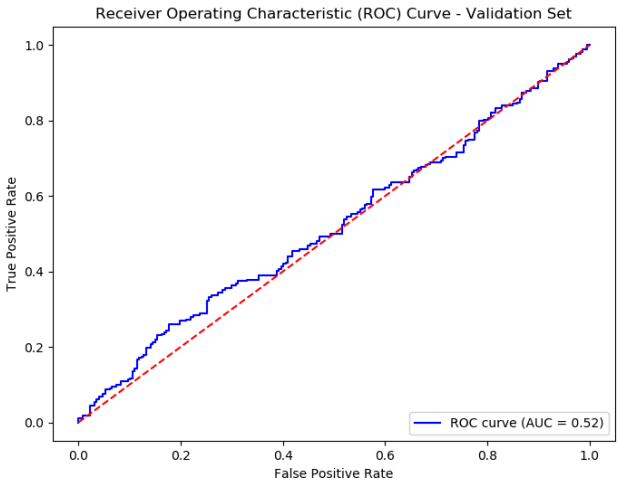
For the Logistic Regression model, we employed scikit-learn's GridSearchCV to perform a grid search over the specified hyperparameter space. The model with the highest AUROC score on the validation set was selected as the final model.

Hyperparameters:

LogisticRegression(C=1, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='warn', n_jobs=None, penalty='l2', random_state=20, solver='liblinear', tol=0.0001, verbose=0, warm_start=False)

This graph displays the true positive rate and false positive rate of the logistic regression model:

Hok Wai Chan
ICS635
5/11/2024




These AUROC scores on the held-out test set provide an unbiased estimate of the model's performance on unseen data, as the test set was completely isolated from any decision-making or model-tuning processes.

The performance of the models on the test set is as follows:

- Logistic Regression: AUROC on test set = 0.5219303308850056
- Ensemble Model: AUROC on test set = 0.506233856188401

The dataset used in this study covers a decade and encompasses a wide range of market conditions, including multiple boom and bust cycles, and regulatory changes. While this extensive time period and diversity of data may help the model capture various market dynamics, there is still a risk of distribution shift when applying the model to data from a different time period or market conditions. Factors such as changes in cryptocurrency regulations, adoption rates, and global economic conditions can potentially alter the distribution of the data, leading to

a degradation in the model's performance. Additionally, the inherent volatility and unpredictability of cryptocurrency markets pose challenges in ensuring the model's long-term stability and generalization capabilities. To mitigate the risk of distribution shift, incorporating relevant external factors, such as news sentiment, regulatory changes, and macroeconomic indicators, could potentially improve the model's ability to adapt to changing market conditions.

Hok Wai Chan
ICS635
5/11/2024

# References

Dataset:

https://www.kaggle.com/datasets/saswattulo/bitcoin-price-history-btc-usd/data

Outliers solution:

https://medium.com/@sedefftaskin92/feature-engineering-data-pre-processing-outliers-e072f7bdcc63