BITCOIN TREND ANALYSIS

Group No 8

PRAKYATH DAVANAM PRAVEEN
RUJUTA MIRAJKAR
YUKTHA LAKSHMI A B

IST 718: Big Data Analytics

Final Project Report

Project Overview

Historical bitcoin data contains an abundance of detailed information on which big data analytics can be used to better analyze trends over time. The primary goal is to undertake a thorough trend study of Bitcoin's price and trading activity from 2012 to 2020 utilizing historical data collected at one-day intervals on the Bitstamp exchange.

In this project, we will investigate the bitcoin price and use the spark machine learning model to predict the prices for the future 10 days using past 30 -days and 60-days bitcoin data . In addition, investigate the time series analysis of bitcoin data using the ARIMA, SARIMA, and LSTM models by comparing the Root mean square error of all models.

Transforming Unix timestamps into dates that can be read by humans improves the data's accessibility and makes it easier for a wider audience to understand. Finding major turning moments in the price movements of Bitcoin becomes increasingly important as the project develops since it offers important insights into the dynamics of bull and bear markets as well as possible contributing factors. The data for our project which spans from 2012 to 2020 corresponds with how Bitcoin and the cryptocurrency market have changed during a critical stage of development.

To sum up, the project's comprehensive methodology seeks to uncover the intricate mechanics of Bitcoin's price and trading activity on the Bitstamp platform. Through the utilization of advanced analytics and historical data, the project aims to reveal patterns and establish a basis for well-informed decision-making within the ever-changing and dynamic domain of cryptocurrency markets.

Prediction Goals

This project's main prediction objective is to estimate future Bitcoin Weighted_Price prices using past data taken from the Bitstamp exchange. With machine learning and advanced analytics, the initiative seeks to create models that can offer insights into possible future price fluctuations. Linear Regression model is used to predict the future 10 days weighted price by capturing past 30 days data and past 60 days data.

Main goals are:

1. Price Prediction
2. Weighted price Prediction
3. Price Change direction
4. Predictive feature Importance

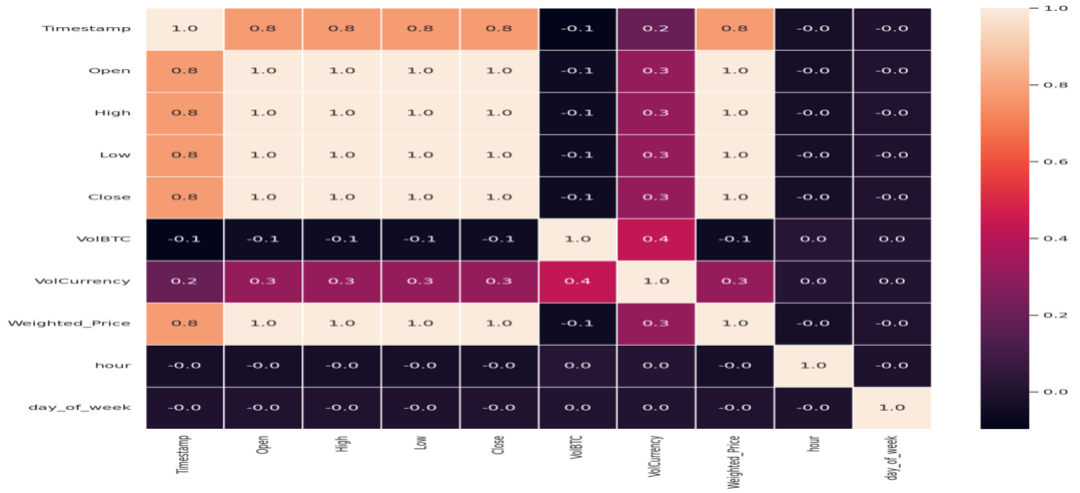
Data Exploration

We used CSV files with day-by-day updates of OHLC (Open, High, Low, Close), volume in BTC and the indicated currency, and the weighted bitcoin price are available for a selection of bitcoin exchanges from 2012 to 2021. Unix time is used for timestamps. Timestamps with no transactions or other activity have NaN values in their data fields.

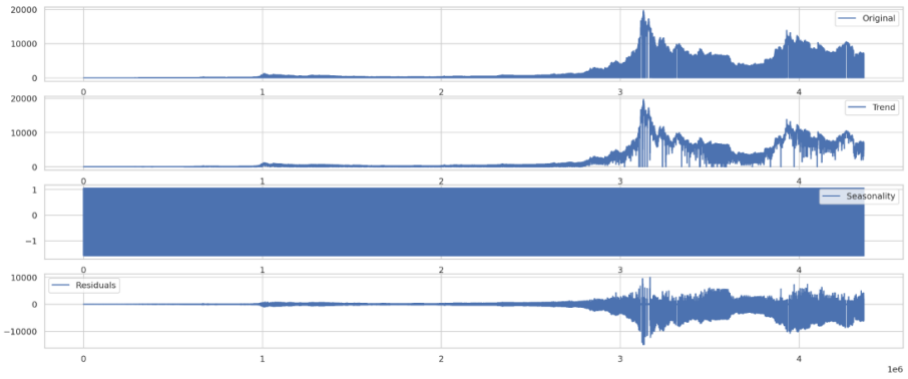
Columns in our dataset -

- Timestamp: Data collection date (in Epoch Unix format); It will thereafter be changed into a "human" date for improved comprehension; Approximately once every minute, with the time zone set to UTC
- Open: Initial currency trading value in that measurement range, in USD
- High: Highest value reached by the asset during that measurement interval, in USD
- Low: Lowest value reached by asset during that measurement interval, in USD
- Close: Value of the asset at the time of closing the measurement range, in USD
- Volume_(BTC):Volume, in BTC, traded on Bitstamp during a given measurement interval
- Volume_(Currency): Volume, in USD, traded on Bitstamp during a given measurement interval
- Weighted_Price: Average asset price in that range, in USD; Calculated based on traded volumes; It will be considered as the average price for analytical issues

In order to begin data exploration, we captured the required columns and looked at the relationships between the various kinds of data.



A visual representation of correlation from 1 to -1 is shown in the heat map below. A positive 1 indicates that the two variables are positively coordinated. Inverse coordination of the variables is indicated by a negative -1.
Data Transformation step included with converting Unix time stamp to Date and time timestamp using spark SQL, splitting the data frame with required columns for prediction.



Data Models:

Few interesting results from our analysis –

1. Linear Regression Model

The Linear Regression model employs a linear relationship between the input features 'Open,' 'High,' 'VolBTC,' 'VolCurrency,' and 'Weighted_Price' and the target variable 'Weighted_Price'.
In the following step we use VectorAssembler() where it takes all the columns to be the input and combines them into a single vector column. Later assemble.transform() is used to create a new dataframe. The Linear regression model is applied to the training (complete data set minus 10 rows) and test data set to know the predicted price and the weighted price.

Predicted_price	Weighted_Price
728.8645289776432	729.099358
728.9142216569854	729.3566123
729.5079599290792	729.5570372
730.4948238385908	730.8474942
731.9058889568385	734.054301
734.0322363456398	734.005698
733.6738538802706	732.9914991
733.9281173103851	734.0634883
731.4773547135726	731.3727073
733.0284670215345	733.3693026

We also implemented a linear regression model to evaluate the weighted price of next 10 days using the **last 60 days** weighted price. The evaluation method used is root mean square error of **0.74(74%)**. The predicted values for this model are:

Predicted_price	Weighted_Price
729.0556001675928	729.099358
729.2802955139322	729.3566123
729.5355966276295	729.5570372
730.6903654050542	730.8474942
733.7391770584784	734.054301
733.6897669546983	734.005698
732.7928796943746	732.9914991
733.7282742577	734.0634883
731.2765067779862	731.3727073
733.0850218393057	733.3693026

By taking the training data set from last 30 days to evaluate the price of next 10 days , the root mean square for this linear regression model is **0.21(21%)**. The predicted values are:

Predicted_price	Weighted_Price
729.2191084897619	729.099358
729.4413294488577	729.3566123
729.6428793298834	729.5570372
730.75110976723	730.8474942
733.3025911374841	734.054301
733.5272362970273	734.005698
732.6618568519341	732.9914991
733.5752506872676	734.0634883
731.2557969585706	731.3727073
732.9533360535556	733.3693026

2. Gradient Boost Trees(GBT)

RMSE: 54.51%

The 'Weighted_Price' is predicted using the Gradient Boosted Trees (GBT) regression model using a combination of decision trees and predetermined features. GBT, in contrast to linear regression, captures non-linear correlations between the target variable ('Weighted_Price') and the input features. Even though it is higher than in linear regression, the RMSE value of 54.51% nevertheless indicates the average prediction error.

In the following method we use **GBTRegressor()** function to capture features column, prediction column, label column by defining the max dept of the tree. The model is fit using the training data and transform the training data to get the test data set.
The prediction values are:

Predicted_Price	Weighted_Price
688.9946314301155	729.099358
690.3306398832452	729.3566123
688.5221516023112	729.5570372
688.3224224763751	730.8474942
684.2271575196239	734.054301
801.9916656932893	734.005698
801.7160824640947	732.9914991
800.9814159909084	734.0634883
688.3224224763751	731.3727073
803.6520808045478	733.3693026

3. Time Series Analysis:

Time series analysis is a common approach for studying and predicting Bitcoin trends, given the temporal nature of cryptocurrency price data. Given that we are working with time series data for cryptocurrencies, we have numerous modeling methodologies to choose from, depending on our goal.
For the data exploration stage, splitting the data into train and test data set using Minmax Scalar() method.

ARIMA Model

RMSE: 375.004

Autoregressive (AR) and moving average (MA) variables are combined in the ARIMA model to generate predictions. The average prediction error is indicated by the RMSE value. In this case, lower RMSE values are better, so 375.004 might be considered relatively low.

SARIMA Model

RMSE: 395.690

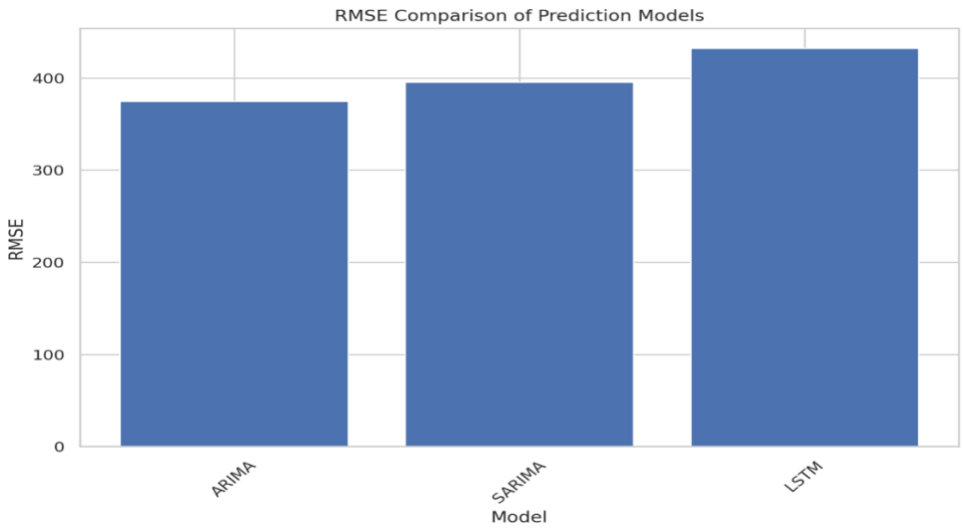
By including a seasonal component, SARIMA expands on ARIMA. The better predictive ability of the model may be inferred from the decreased

RMSE when compared to ARIMA, which takes seasonality into consideration. However, further tuning might be beneficial.

LSTM Model

RMSE: 432.15

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) suitable for sequence prediction. The higher RMSE suggests that, in its current configuration, the LSTM model might not be capturing the underlying patterns well.



Summary of Methods Used to Solve the Problem

1. Linear Regression Model:

In the linear regression analysis, the model was initially trained on the majority of the dataset, excluding the last 10 rows, which were reserved for testing. After training, predictions were generated for this test set. Further refinement was performed by training additional models on subsets of the training data, specifically the last 60 and last 30 rows. Predictions were generated for each of these models on the same test set. The final linear model exhibited a Root Mean Squared Error (RMSE) of 0.202531, indicating its predictive performance on the test data.

2. GBT Model:

In the Gradient Boosted Trees (GBT) regression analysis, we employed the GBTRegressor algorithm to predict 'Weighted_Price' based on specified features. The model was trained on the majority of the dataset, excluding the last 10 rows, which were set aside for testing. Predictions were generated for this test set using the trained GBT model. The resulting predictions were then evaluated using the Root Mean Squared Error (RMSE) metric, yielding an RMSE of 71.80.

3. ARIMA Model:

In the ARIMA (Autoregressive Integrated Moving Average) model, we utilized past 'Weighted_Price' values to forecast the time series. The order specification (10, 1, 7) denotes the configuration of the ARIMA model, where:

Autoregressive (p): To capture the temporal dependencies in the data, we took into account 10 autoregressive terms and included the influence of the previous 10 'Weighted_Price' observations.

Integrated (d): We used first-order differencing (d=1) to achieve stationarity, which involved calculating the difference between successive observations to turn the time series into a stationary one.

Moving Average (q): In order to smooth out variations in the time series, we took into consideration seven moving average terms, each of which includes the weighted total of previous forecast errors.

In order to accurately forecast, the resulting ARIMA model with the given order (10, 1, 7) sought to capture both short- and long-term patterns in the 'Weighted_Price' time series data.

4. SARIMA Model:

In the SARIMA (Seasonal Autoregressive Integrated Moving Average) model, we considered the 'Weighted_Price' time series data. The model was configured with an order of (1, 1, 1) for the non-seasonal components and a seasonal order of (0, 1, 1, 12), indicating a seasonal pattern with a yearly cycle (12 months). The breakdown is as follows:

Non-seasonal Order (p, d, q): The non-seasonal components, which consist of moving average, differencing, and autoregressive terms, were assigned a (1, 1, 1) value.

Seasonal Order (P, D, Q, S): We used (0, 1, 1, 12) for the seasonal components, where 'P' stands for the seasonal autoregressive terms, 'D' for seasonal differencing, 'Q' for seasonal moving average terms, and 'S' for the length of the season (12 months).

The objective of the SARIMA model was to represent the seasonality and temporal dynamics present in the 'Weighted_Price' time series.

5. LSTM Model

For the LSTM (Long Short-Term Memory) model, we used a neural network architecture, specifically designed for sequence prediction tasks. The 'Weighted_Price' time series data was scaled using Min-Max scaling before training the model. The architecture included:

The input shape of the model was set to (1, 1), which denotes a single input feature (univariate time series) with a single time step.

Layers: The temporal dependencies were recorded using a single 128-unit LSTM layer, while the output was recorded using a Dense layer with a single unit.

Loss Function and Optimizer: The model was compiled with the mean squared error loss function and the Adam optimizer.

Results Summary

Our time series model is predicting Bitcoin closing prices for the 10-day period following April 13, 2020. The model utilized the last 15 days of available historical data to generate these forecasts.
Implemented sequential Analysis by taking past 15 days data to predict the price of future next 10 days.

Problems Encountered

Completeness of Data:

The dataset needed to be handled carefully because it had missing values in a number of columns, including "Open," "High," "Low," and "Close."
forward filling for OHLC data and mean imputation for other columns were used to address missing values.

Selecting a Time Series Model:

Time series forecasting may not be the best use for linear regression, even though it works well for other predictive applications.
We evaluated the effectiveness of the LSTM, SARIMA, and ARIMA models in comparison to the linear regression model.

Summary of achieved Prediction and Inference Goals

We tried to successfully apply models to meet our prediction objectives and deliver precise estimates for Bitcoin values. Using linear regression, we forecast Bitcoin values based on characteristics such as 'Open, High, VolBTC, VolCurrency, and Weighted_Price. We created a neural network with Long Short-Term Memory (LSTM) to capture complex temporal relationships in Bitcoin values.
Using ten training epochs, a deep learning model was trained, offering insights into the difficulties and complexities of temporal pattern modeling. By deriving significant insights into the seasonal and temporal patterns of bitcoin values, the inference aims were achieved. A thorough grasp of the dynamics of Bitcoin prices was made possible by the combination of our sophisticated deep learning techniques and conventional time series models.

Citations

Bitcoin Historical Data
https://www.kaggle.com/datasets/mczielinski/bitcoin-historical-data

David Andres (June 15, 2023) Step-by-Step Guide to Time Series Forecasting with SARIMA Models
https://mlpills.dev/time-series/how-to-train-a-sarima-model-step-by-step/

Chaitanya Gawande (2023, Oct 17) Time Series Analysis and Prediction of Cryptocurrency Prices: A Step-by-Step Guide
https://medium.com/@cgawande12/time-series-analysis-and-prediction-of-cryptocurrency-prices-a-step-by-step-guide-9e87d219eb16