Capstone Project

Nasser Alnassar

March 2024

1 Project: Predicting Bitcoin Weighted Price Using Time Series Forecasting

1.1 Project Overview

This report details the development and evaluation of a time series forecasting model for predicting the future weighted price of Bitcoin. The project explores the effectiveness of various forecasting techniques in analyzing historical price data and identifying patterns to anticipate future trends. The model's performance will be assessed based on its ability to minimize prediction errors.

1.2 Problem Statement

Bitcoin's price is notoriously volatile, making accurate price predictions a significant challenge. This project aims to address this by creating a model that can learn from historical data to forecast future weighted prices.

1.3 Strategy

Our strategy to solve the problem of predicting Bitcoin's weighted price involves the following steps:

1. Data Acquisition and Preprocessing: Acquire historical Bitcoin price data and clean it by handling missing values, scaling features, and potentially transforming the timestamp format for more granular analysis.

2. Data Exploration and Visualization: Explore the data to understand its characteristics, identify trends and seasonality, and visualize relationships between features using techniques like time series plots, scatter plots, and heatmaps.

3. Model Implementation and Evaluation: Implement and compare different time series forecasting models, such as ARIMA and LSTM. Fine-tune hyperparameters of each model and evaluate their performance using the defined metrics (MSE and MAE) on a separate testing set to avoid overfitting.

4. Model Selection and Justification: Select the model that achieves the lowest MSE on the testing set. Justify this choice by explaining its strengths and limitations compared to other models.

5. Result Interpretation and Conclusion: Draw conclusions based on the model's performance, considering its limitations and potential areas for improvement.

1.4 Metrics

The model's performance will be evaluated using two key metrics:

* Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values. Lower MSE indicates better performance.

*Visualizing the results

1.5 Data Description

This section will describe the specific dataset used for the project. Include details such as:

- Source of the data: Kaggle - Bitcoin historical Price Dataset
- Timeframe covered by the data: 2012 - 2021
- Features included in the dataset: Timestamp, Open, High Low Close, VolumeBTC VolumeCurrency, WeightedPrice

2 Analysis

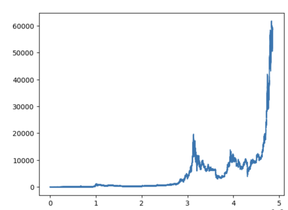

Figure 1: "Showcasing Bitcoin market data"

2.1 Data Exploration

I will explore historical Bitcoin price data, focusing on factors like opening price, closing price, trading volume, but mainly on the Weighted Price since it captures all the aspects i need, and it being a good indicator of the market value, while only using timestamps to predict.

2.1.1 Abnormalities Found

- We have found that timestamp wasn't a good representative of time in terms of time series forecasting.
- I have found that 25 percent of the dataset is null values,
- We had to use feature scaling since we will only use the timestamp to predict the weighted price

2.2 Data Visualization

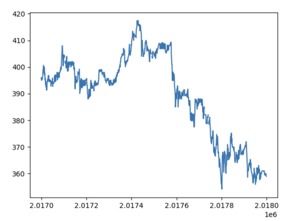

Figure 2: Data from 2017-2018

In figure 1: We can see that Bitcoin market pricing hasn't been stable at all, which is one of the main reason i am building a time series forecaster.

In figure 2: We can see the amount of deviation given a smaller time interval.

3 Methodology

3.1 Different Types of models used

NumofParameters	numLayers	BatchSize
350	2	64
570	4	128
1540	8	128
1570	8	1024

3.2 Data Preprocessing

The data preprocessing steps will include:

* Handling Missing Values: Filling missing data points using techniques like interpolation. * Feature Scaling: Standardizing features to a common scale to ensure all features contribute equally to the model. * Time Stamp Conversion: Transforming the timestamp column from a basic timestamp format to a datetime format for more granular analysis.

3.3 Implementation

We will implement and compare different time series forecasting models, such as:

* Long Short-Term Memory (LSTM) * Multi Layer Perceptron (MLP)

We will be using The Timestamp as the input data and the Weighted price as the target

3.4 Refinement

We will fine-tune hyperparameters of the model to optimize the performance and compare the accuracy using the defined metrics (MSE).

4 Results

4.1 Model Evaluation and Validation

After Trying different models and libraries including pytorch scikit learn and tensorflow I have found that tensorflow performed faster even the model is larger,

NumofParameters	numLayers	BatchSize	MSE
350	2	64	14000000
570	4	128	785000
1540	8	128	460000
1570	8	1024	244745

At first i was starting with a smaller model with 1 hidden layer with a total number of parameters: 350

Secondly i tried to make the model larger up to 570 with noticeable improvement,

Thirdly i went for a different activations and found that Relu was an amazing choice for the hidden layers

And lastly i made the model larger close to 1600 parameters and found that it resulted the best given the limited computing resources.

4.2 Justification

I have found that i the Neural Network i built needed more parameters to learn more about that data therefore increasing the size of the network helped, but it required better split date for the train and test data and required more epochs to be fitted, but overall LSTM neural network worked the best

5 Conclusion

5.1 Reflection

This project will provide insights into the effectiveness of time series forecasting for predicting Bitcoin prices, Being this unpredictable can make it difficult for models to learn certain aspects, especially since this field is difficult to begin with.

5.2 Improvement

By getting access to better resources and more up to date data, there will noticeable improvement to the accuracy and effectiveness of the model.

Also, research has found the Autoregressive integrated moving average - ARIMA which is a statistical model has been effective at predicting stock prices and could work well in this scenario