Bitcoin Price Prediction With Deep Learning

Carson Aziz 180800140

Mohamed Qadeer Assan 16027370

Introduction

- Welcome to this presentation on the use of Long Short-Term Memory (LSTM) models to predict the price of Bitcoin. In this report, we will explore the results of using LSTM models to make predictions on the price of Bitcoin. We will examine the data pre-processing requirements, feature selection, and model tuning that went into crafting our model.
- Cryptocurrency prices are known to be very volatile, and Bitcoin is no exception. The price of Bitcoin fluctuates frequently, and some of these fluctuations can be significant. Because of this volatility, it is desirable to be able to make accurate predictions on its future price.
- An LSTM model is a type of neural network, specifically a recurrent neural network (RNN), that takes sequences of data to produce a prediction. LSTM models are useful over run-of-the-mill RNNs as they can learn long-term dependencies of the sequence and use that to make predictions. They also have the ability to selectively remember or forget pieces of data. The LSTM uses both long-term and short-term data from the given sequence to make its prediction.
- LSTMs have been used in finance with positive results to make predictions on stock prices. We believe an LSTM model is an appropriate strategy for making predictions on Bitcoins price and is what we have chosen to explore in this project.

Methodology

- For model planning, we used Keras sequential models to build our neural network. Our neural network consisted of multiple LSTM layers, multiple dropout layers, and multiple dense layers. We selected LSTM layers as they are capable of capturing dependencies in sequential data, making them a good choice for time series prediction. Dropout layers were also useful in preventing overfitting by removing some input values during training, forcing the neural network to learn more generalized features. Dense layers were used to learn complex non-linear relationships between the input and output layers.
- Our model's structure processes data in stages, beginning with an LSTM layer to learn long-term dependencies. The next layer in the stage is either a dropout layer or a dense layer, with dropout layers helping to desensitize that stage from noise output by the previous stage, while dense layers help create mappings to the prediction output nodes.
- Finally, we discussed how our models differ in their feature sets, which we selected based on their minimum correlation requirements. This provided us with a diverse set of features to make predictions on our input data. We also optimized our model by tuning different parameters, such as the activation function, the number of layers and nodes, and the optimizer parameter. The performance metrics of these different configurations allowed us to select the most optimal model.