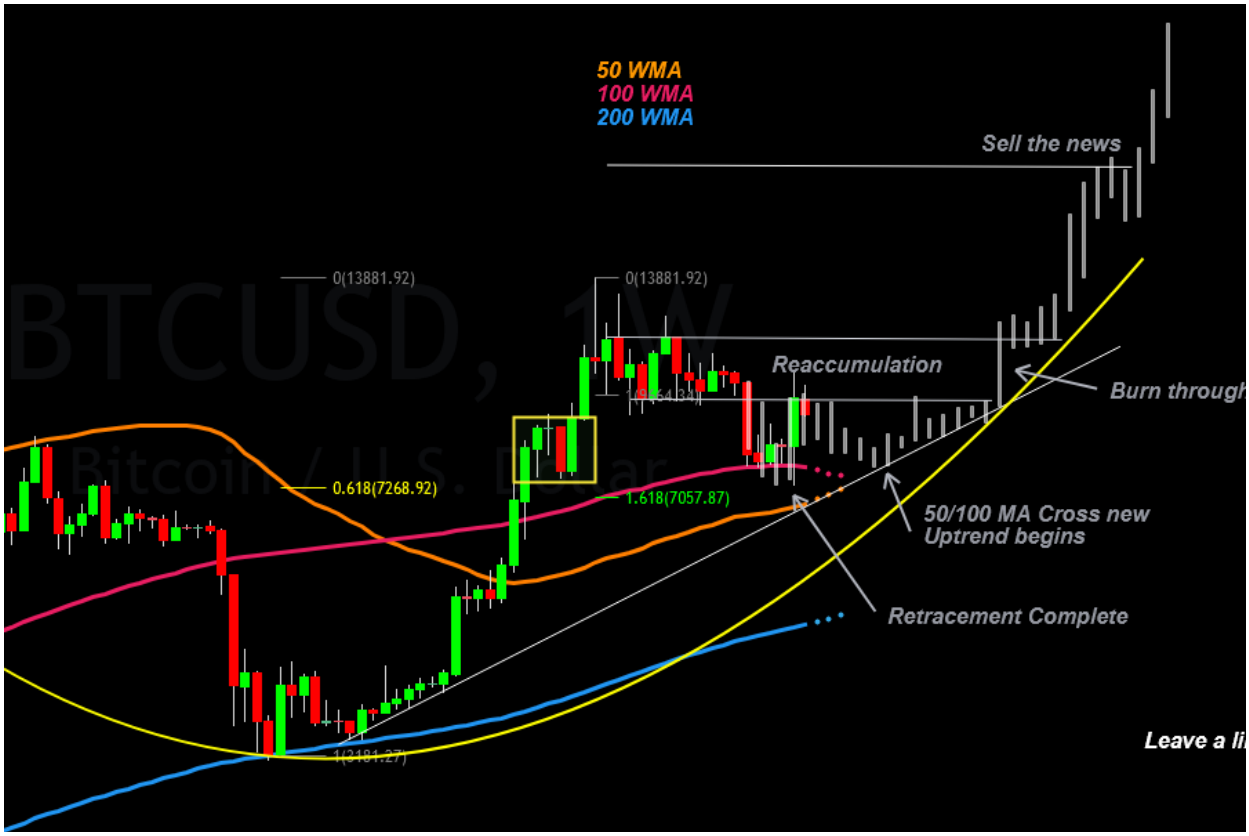

Bitcoin Price Forecasting

08.10.2021

—

Florian Mitterbauer

Domain Background

Future price predictions based upon past data can be achieved with the field of **time series forecasting**. With its help, you can predict future prices of a stock, given historic data. Such a prediction would be a regression problem since we want the output to be an array of continuous predicted numbers (e.g. prices). One good forecasting model would be the ETS (explicit model of error, trend and seasonality) which is an exponential smoothing technique. Other models are the ARIMA model (autoregressive integrated moving average) or the LTSM which is a type of an RNN.

I chose this domain because I was fascinated by the case study example about price predictions and wanted to do a project on it in regards of predicting Bitcoin prices, since the crypto currency world is one of my many fields of interest and I have the chance within this project to use AWS cloud resources, I wanted to dive deeper into the AWS deepAR service and what it has to offer.

Problem Statement

The project is revolving around future Bitcoin price prediction. Given current pricing data of the last week/month/year, to predict future price movements of different time frames (weeks, months up to a year). The input and output data contains a time series of prices for Bitcoin. The task should be solvable with a time series forecasting model/solution like the AWS deepAR algorithm. The price prediction should be shown on a simple webpage displaying the given input (current and past prices) and its future predictions (output of the predictor) as a graph.

Datasets and Inputs

The input/training data for this project was downloaded from kaggle (https://www.kaggle.com/mczielinski/bitcoin-historical-data). It contains the historic Bitcoin prices from 2012 to March 2021 in one minute intervals.

It contains columns about opening, high, low and closing price of a given timeframe (open, high, low, close), the volume and the weighted price. Each record also has a timestamp which is in unix time format.

Most relevant, for this use case, will be the timestamp and the weighted price column. Since the data has a couple missing values for the prices and timestamps, it needs to be cleaned of those first. After that, it might make sense to lower the resolution of the time-series from one minute to either one hour or one day to reduce training time and compensate for missing values. Such a high resolution of one minute is anyway not needed, since we want to predict prices with the resolution of one price value per day for a time frame of at least a week and at most a year.

Solution Statement

For creating a time series forecast on Bitcoin prices, the AWS deepAR algorithm will be used. After cleaning and preprocessing the data, it will be trained on deepAR to create a model. When the model was sufficiently evaluated to be good enough, it will be deployed as an endpoint. Via an AWS Lambda function and the API Gateway the model will be exposed as an API to be used by a webpage showing the future Bitcoin price prediction results.

Benchmark Model

For benchmarking, I found this paper:

'Statistical and Machine Learning forecasting methods: Concerns and ways forward' from Spyros Makridakis, Evangelos Spiliotis, Vassilios Assimakopoulos (https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0194889)

Which compares the performance of various machine learning time series forecasting algorithms to classic statistical forecasting methods. The results on sMAPE (Symmetric mean absolute percentage error) of ARIMA and MLP will be used as benchmark values to compare against.

Evaluation Metrics

Measuring the performance of the price predictor will be done by visually comparing the predictions to existing (past) prices and check if the predictions are mostly within a given confidence interval or not. To compare it with other price prediction approaches/algorithms the sMAPE will also be evaluated, for comparing with benchmark results.

Project Design

The project will be split in four different parts:

I. Data Cleaning and Preprocessing

When the data is loaded into the notebook, it needs to be cleaned from missing values and the resolution should be lowered to hours or days. Finally the data must be split into training and test data.

II. Creating and Training the deepAR Model

The model needs to be created, configured and trained on the training data using Amazon SageMaker.

III. Evaluate the Model

A predictor should be created to evaluate the performance of the model and compare it to the benchmarks. This should also include a visual presentation of the forecast compared to the real values and the sMAPE values.

IV. Deployment and Web App

An AWS Lambda function should be prepared to use the endpoint for the bitcoin price prediction. The lambda will be connected to a new API endpoint created in the AWS API Gateway, which will be used by the web page showing the Bitcoin price predictions. The web page should show a graph of past (input) and predicted (output)Bitcoin prices.