



Cryptocurrency Trend Prediction Web App

Krzysztof Milde, Daniel Pławiak, Marcin Świerkot, Łukasz Zalewski



Introduction and goals

- cryptocurrencies has become one of the hottest topics recently
- brought thousands of new investors over the past year
- build the artificial intelligence model to predict the trend in cryptocurrency prices
- test the results of the model on such a volatile market



Backend

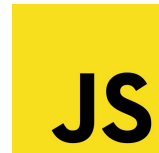
Backend service for our application is being served by Flask.

Why Flask?

- Ease of implementation
- Python based
- Smooth deployment



Frontend



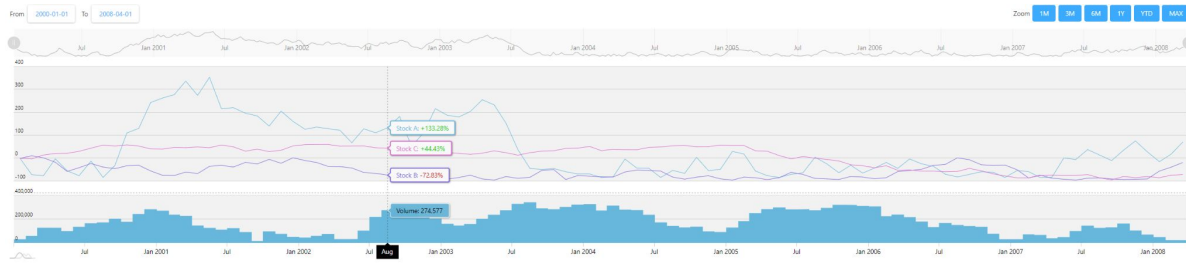
+



GUI was created with usage of vanilla JavaScript with amCharts extension.

Why amCharts?

- Detailed graphs
- Stock dedicated templates
- Interactive





Backend and frontend responsibilities

Backend	Frontend
<ul style="list-style-type: none">• Fetch and process appropriate historical data from CoinGecko• Get current price• Use implemented models to predict price trend in the future• Pass all variables to frontend	<ul style="list-style-type: none">• Create graph using data passed from the backend• Display current price• Display price change in 1, 3, 5 and 7 days from now• Display appropriate statistics below the graph (Loss and Accuracy)



Application

- provides the user with the possibility to check the current price of three cryptocurrencies
- present the trend predictions for the next 1, 3, 7 days





Screenshots

Bitcoin ▾

Date: 15/06/21 10:57

Price: 40074.48\$

Prediction:

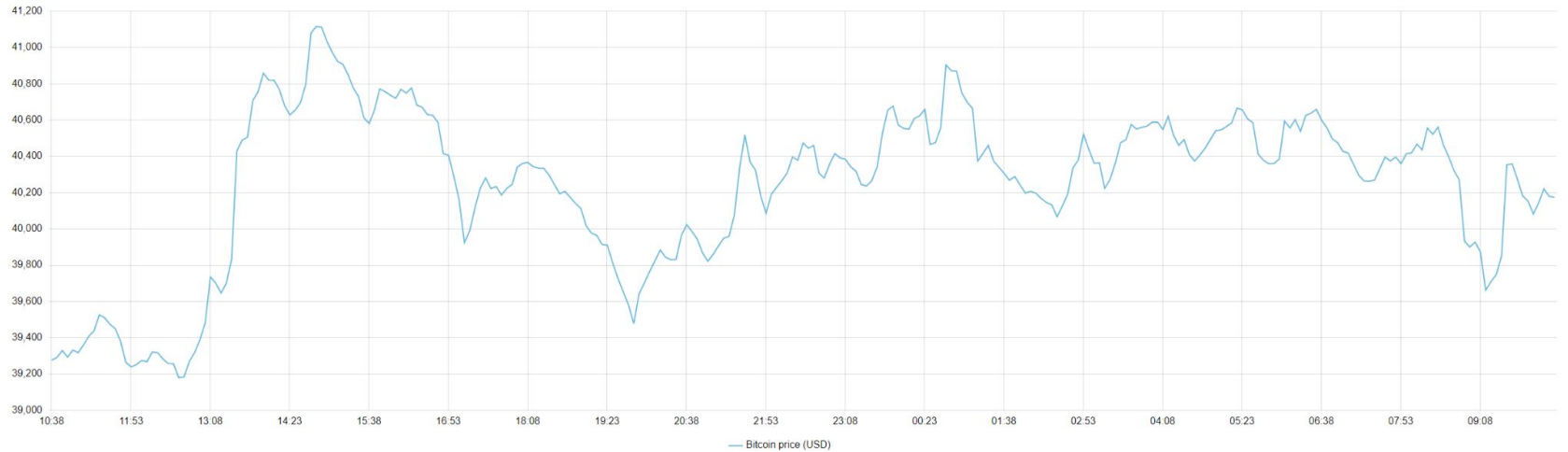
1 day: ↘

3 days: ↗

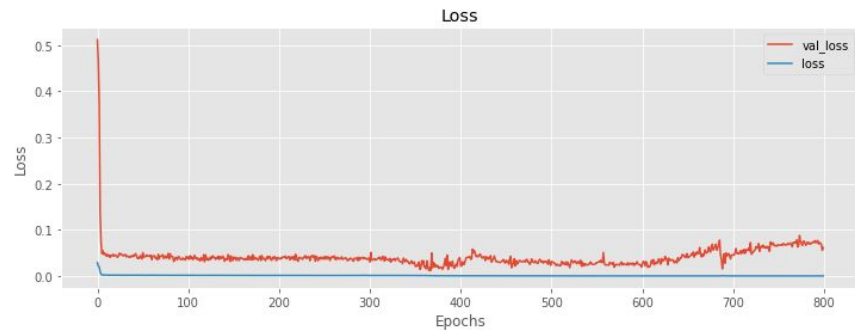
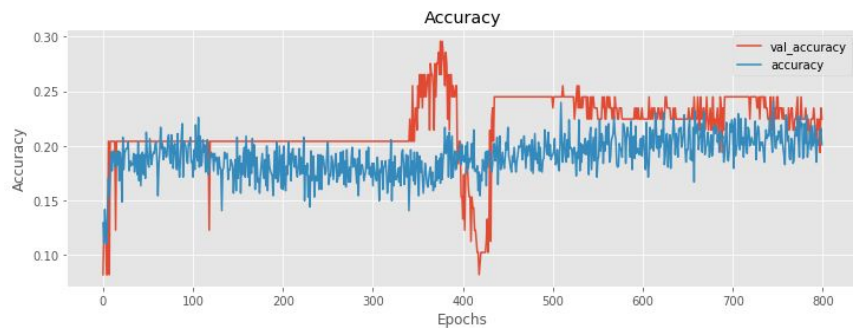
5 days: ↗

7 days: ↗

Screenshots pt.2



Screenshots pt. 3





Predicting software and data used



In order to accomplish model our project is based on, popular python library - Keras - has been used.

Why Keras?

- Open source
- Based on Google's TensorFlow
- Neural Network specific library

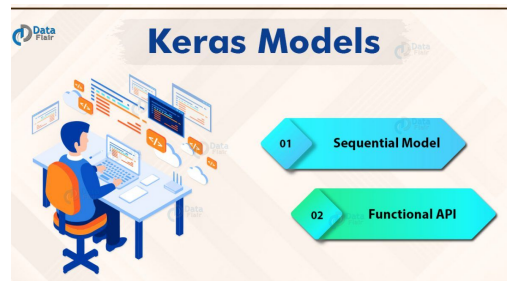
Data used to train the model has been provided by Yahoo Finance API.

Prediction model

The model was built using Keras Sequential model with Bidirectional LSTM layers

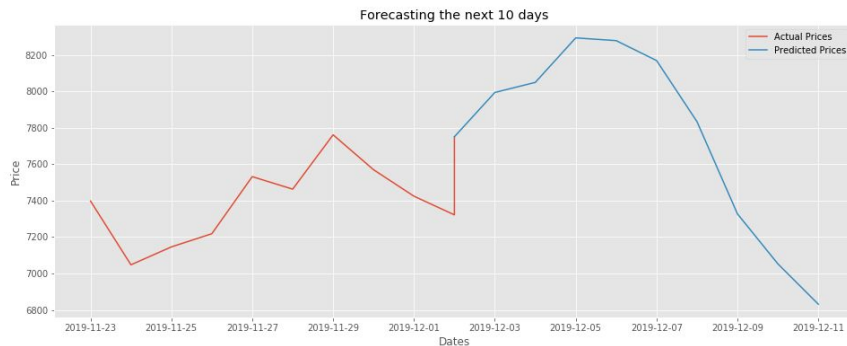
Why LSTM?

- appropriate for doing time series prediction
- based on last “n” days the model tries to predict behavior in the future



Model details

- configurable periods to look back and periods to look into future (predict),
- activation function found empirically - softsign or tanh perform the best,
- one feature - closing price,
- input layer with 30 nodes,
- 6 hidden layers, 12 nodes per layer (configurable),
- final hidden layer with 10 nodes,
- output layer with number of days to predict as number of nodes (configurable)



Exemplary prediction of the future using our model



Let's get into details

Presentation time...



Publicly available service

On the purpose of presentation, we decided to deploy our solution to Heroku.

Service can be reached under <https://project-aimini.herokuapp.com/>.