**Price Estimation and Prediction Research**

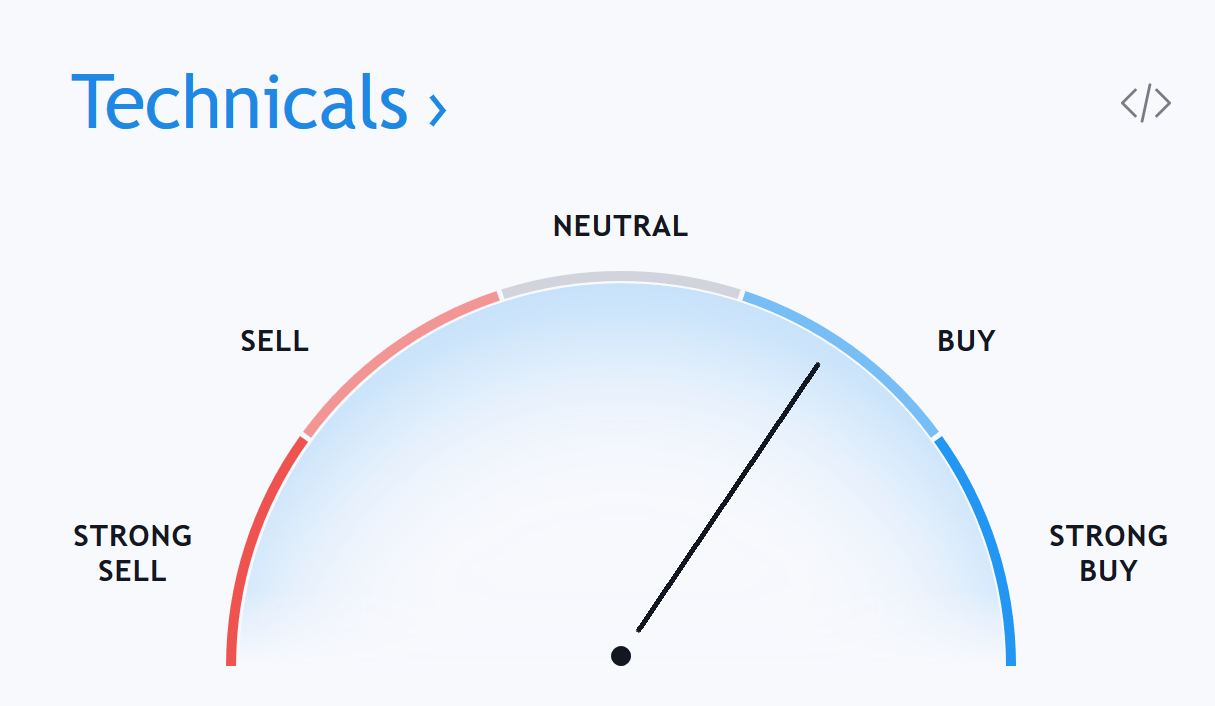
The purpose of this pet project is to investigate and implement various machine learning/rule based methods for estimating and predicting the future value of a given crypto currency. These user stories will cover the users interaction with these predictions.

The outcome of the pet project will be to provide a page on the website with an implementation of my learnings.

There are different variations on a prediction:

Trading view has a diagram that tells you if the current market is in buy or sell:

<https://uk.tradingview.com>



<https://medium.com/datadriveninvestor/predicting-cryptocurrency-prices-with-machine-learning-1b5a711d3937>

Rule based machine learning:

<https://en.wikipedia.org/wiki/Rule-based_machine_learning>

Python machine learning techniques:

<https://towardsdatascience.com/4-machine-learning-techniques-with-python-ceee451b0085>

Plan

To start off with I will focus on regression techniques to predict the prices. Regression techniques include linear and polynomial. Linear will create a straight line through the given data points. This will allow for a very basic prediction based on the current trend. Linear regression will work well when the value of the cryptocurrency is stable, but will start to fall off as the price becomes more volatile. The second regression technique I will use will be polynomial, this will be similar to linear, however it will roughly follow the trend of the data set and therefore should provide a slightly more accurate prediction when trends are consistent. This method will also fall off with volatility, when the price fluctuates with no obvious trend.

I will also use a technique called radial-basis function (aka RBF) as this allows a similar form of prediction to polynomial, however from practice it appears to provide an even more accurate result, usually having a ~4-5% deviation from the actual price on average.

Following the implementation of all three techniques, I will create a machine learning algorithm to combine the results of all three and calculate the most accurate combination (The weighting of each prediction). This should in theory provide a prediction with an accuracy that is greater than that of RBF alone.

Results

After implementing the three techniques and creating the combination technique I am able to get around a 97% accuracy. This means that on average the prediction differs from the actual by roughly 3%.

This prediction prompted me to start looking at Keras which is a deep learning tool that can be used in Python.

Deep learning with Keras

<https://keras.io/>