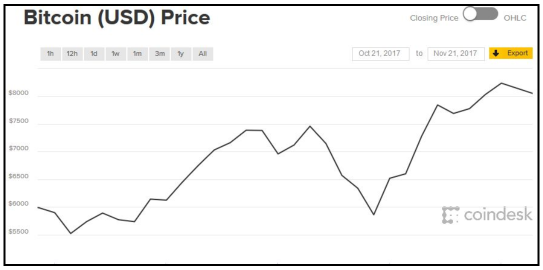
Detecting Anomalies in Bitcoin Stock Data

Thomson Reuters - REUT-2

Team Members: Rohan Johar, Tommy Suen, Eric Chang, Ivan Wong
{rjohar, tsuen, echanglc, wongi}@bu.edu



Project Task: The task is to determine whether a new bitcoin data point is an anomaly and if it is an anomaly, create a basic headline and report extra information that could have influenced the price. Some difficulties with the project include determining how many months of data to use and how different the actual prices and the predicted prices need to be for the point to be an anomaly.

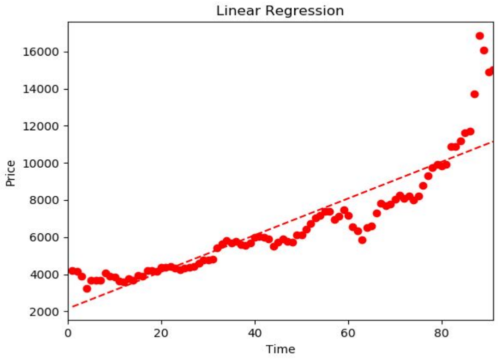
Related Work: This type of task usually falls under predicting stock prices. One research paper that we looked at when deciding our approach was "Predicting Stock Prices Using Polynomial Classifiers: The Case of the Dubai Market [1]". This compares neural networks and polynomial classifiers for predicting stock data.

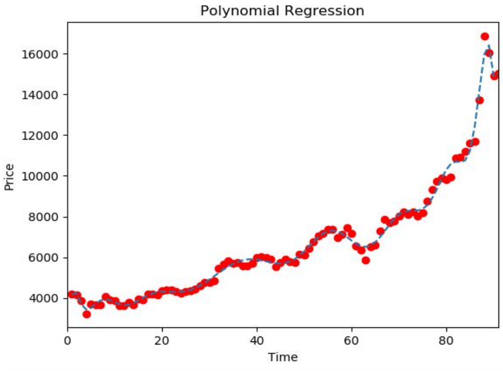
A second research paper, "A hybrid ARIMA and support vector machines in stock model forecasting" attempted to use ARIMA for linear data and SVM's for nonlinear data [2]. We read this paper and decided to use multiple models because the data could be linear or nonlinear.

Approach: We decided to use two regression models, linear and polynomial, to predict the price based off of previous data. Bounds are created using either a low, medium, or high percentage (10%, 15%, 25% respectively), and if the actual price is outside the bounds, the point is labeled an anomaly.

The linear regression model is a simple implementation of a line of best fit given the appropriate stock data. The equations follow The equation for linear regression is $y = x_i^T \beta + \varepsilon_i$ for i = 1, 2, … n and the equation for polynomial regression is: $y = \beta_0 + \beta_1 x_i + \ldots + \beta_m x_i^m + \varepsilon_i$ for i = 1, 2, …, n. For the degree of the polynomial regression, we follow 2 to 25 to prevent a secondary linear regression line, underfitting and overfitting. We use grid search with cross validation from scikit-learn to figure out which polynomial degree to use.

The reason for using two models is because the data could be linear or nonlinear. An example of our regression lines for 12/11/2017, where the polynomial regression is better than the linear regression, is:





To assess the type of headline, we utilized the prediction functions to detect whether an anomaly was positive or negative change. Since we have both linear and polynomial regression curves, we selected the predicted value closest to the anomaly for better accuracy. The headlines are static but the information is dynamically changing. The headline is in the format “BTC Hits Unusual High / BTC Hit New Lows - Breaking predicted values by (Percentage Difference).”

We also included extra information that might have influenced the bitcoin price by scraping through web pages using Beautifulsoup API. For the output, we generated an excel file using openpyxl, containing the price, whether it is an anomaly or not, the headline, percentage change from the day before, and the extra information.

Dataset and Metric: The dataset the program uses comes from Coindesk’s API. We chose to either use one, three, or six months of data in the format of (date, price).

The evaluation metrics we are using are precision, recall, F1, and accuracy. The metric for success is the F1 score, as there isn’t any preference on precision or recall. The baseline metric is having an F1 score of 0.50. The metric for success is having an F1 score above the baseline metric, and we should have an F1 score of 0.75.

Evaluation: The formulas for the evaluation metrics used in this project are:

Precision $= \frac{TP}{TP+FP}$

Recall $= \frac{TP}{TP+FN}$

F1 $= 2(\frac{Precision * Recall}{Precision + Recall})$

Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$

Where TP is true positives, TN is true negatives, FP is false positives, and FN is false negatives.

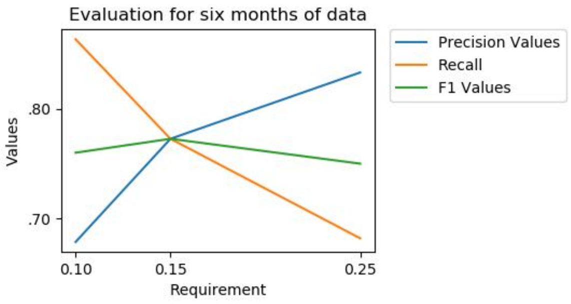
Then in order to use these equations, it is necessary to annotate data and classify data points into either anomalies or non-anomalies. We manually annotated our data by looking at the graph for Bitcoin prices over time and assigning dates as anomalies based on if it was significantly different from the trends beforehand. 22 anomalies and 22 non-anomalies were used for the metrics.

The table below is for evaluation metric results when using different percentages for the bounds and different amounts of data for the construction of the regression lines.

	Low (10%)	Med (15%)	High (25%)
1 Month of Data	Precision: 0.75 Recall: 0.54 F1: 0.63 Accuracy: 0.68	Precision: 0.88 Recall: 0.37 F1: 0.51 Accuracy: 0.66	Precision: 1.00 Recall: 0.14 F1: 0.24 Accuracy: 0.57
3 Months of Data	Precision: 0.64 Recall: 0.64 F1: 0.64 Accuracy: 0.64	Precision: 0.81 Recall: 0.59 F1: 0.68 Accuracy: 0.73	Precision: 1.00 Recall: 0.50 F1: 0.67 Accuracy: 0.75
6 months of Data	Precision: 0.69 Recall: 0.86 F1: 0.76 Accuracy: 0.73	Precision: 0.77 Recall: 0.77 F1: 0.77 Accuracy: 0.77	Precision: 0.83 Recall: 0.68 F1: 0.75 Accuracy: 0.77

In this table, as the bounds increase, the precision increases and the recall decreases. The reason behind this is that when the requirement increases for an actual price to be considered an anomaly, the number of false positives is reduced, as the system concludes that fewer data points are anomalies. But, as the requirement increases, false negatives increase, as less points are chosen to be anomalies, and this leads to a decrease in recall.

This graph shows what this relationship looks like when using six months of data:



For the amount of data the program uses, almost all of the evaluation metrics increase as the program uses more data. A possible reason for this is that the regression lines become more accurate when using more data. Comparing the evaluation metrics to the baseline metrics defined in the previous section, we are significantly above the baseline, when using six months of data. All three F1 scores are above 0.75 and they satisfy the metric for success.

One limitation of using six months of data is that there is a performance drop when the data changes rapidly. In November and December 2017, the prices dramatically increased, and by taking in data from previous months, the predictions generated are too low compared to actual prices. Two examples of this limitation can be shown by the R-Squared Value, the Standard Error Values, and the percentage difference between the actual price and the closest predicted price.

	December 7th 2017	December 8th 2017
1 Month of Data	R-Squared: 0.87 Linear SE: $663 Poly SE: $439 Difference: 34%	R-Squared: 0.81 Linear SE: $1017 Poly SE: $738 Difference: 14%
6 Months of Data	R-Squared: 0.77 Linear SE: $1067 Poly SE: $472 Difference: 42%	R-Squared: 0.74 Linear SE: $1237 Poly SE: $873 Difference: 56%

Looking at these metrics, the one month of data regression lines are more accurate, as the data in November is completely different from the data beforehand. On December 8th, the program wouldn't classify the price as an anomaly when only looking at one month of data.

Conclusion: We've come to the conclusion that the approach we've taken for this project has properly satisfied the success metrics for a dataset over 6 months. However, we recognize that for time periods when there are significant anomalies in the data, using fewer data points for training is more beneficial. Due to Bitcoin's stock-like nature, this program can be taken and applied to stocks or other cryptocurrencies, given data points.

Roles:

Task (Lines of Code)	Lead
Data Retrieval (25 Lines)	Ivan Wong
Linear Regression (20 Lines)	Tommy Suen
Polynomial Regression (70 Lines)	Tommy Suen Rohan Johar
Regression Graphs (35 Lines)	Eric Chang
Anomaly Detection (40 Lines)	Rohan Johar
Headline Creation (40 Lines)	Tommy Suen
Extra Information (30 Lines)	Ivan Wong
Annotating Data (60 Lines)	Ivan Wong
Evaluation (100 Lines)	Rohan Johar
Evaluation Graphs (20 Lines)	Eric Chang
Excel Output (50 Lines)	Rohan Johar
Reports + Poster	ALL

Link to Github Repo:
https://github.com/ivanwong225/MLThomsonReuters

References:

1) Assaleh, Khaled, et al. "Predicting Stock Prices Using Polynomial Classifiers: The Case of Dubai Financial Market." *Journal of Intelligent Learning Systems and Applications*, vol. 03, no. 02, 2011, pp. 82–89., doi:10.4236/jilsa.2011.32010. https://file.scirp.org/pdf/JILSA20110200001_85718640.pdf

2) Pai, Ping-Feng, and Chih-Sheng Lin. "A Hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting." *Omega*, vol. 33, no. 6, 2005, pp. 497–505., doi:10.1016/j.omega.2004.07.024.