CoinPIX – A Cryptocurrency Recommender Using Data Mining

Andre Robinson
University of Oklahoma
Norman, Oklahoma, USA
andrecrob@ou.edu

Christopher Nguyen
University of Oklahoma
Norman, Oklahoma, USA
christopher.p.nguyen-1@ou.edu

Steven Wilson
University of Oklahoma
Norman, Oklahoma, USA
steven.wilson@ou.edu

ACM Reference Format:
Andre Robinson, Christopher Nguyen, and Steven Wilson. 2021. CoinPIX – A Cryptocurrency Recommender Using Data Mining. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Abstract

Cryptocurrency investment has continually increased since it's creation in 2009 with a market capitalization over 1.63 trillion dollars in 2021. Previous research into cryptocurrency pricing focused on the leading incumbent coins such as Bitcoin and Ethereum, but has not included the immature tokens generally known as alt coins.[1]

This project applies data mining to create an evaluation platform that includes the top 100 coins and tokens by market capitalization. The result is an application named CoinPIX that provides a repeatable platform for building an optimized portfolio including alt coins.

It applies clustering to group coins by risk and uses a random forest classifier to predict the current price trend for each coin. It applies a support vector regression algorithm to predict prices at one day, one week, and one year intervals. Finally, it uses a genetic algorithm to build an optimized portfolio from a selected risk cluster.

2 Introduction

The project uses data mining to recommend cryptocurrency to a user based on a selected risk tolerance. This was accomplished using the following data mining techniques.

1. **Clustering** divides the top 100 cryptocurrencies by market capitalization into **three risk clusters** (high, medium, and low). This feature enables the selection of coins based on a user's risk tolerance.
2. **Classification** determines the **current trend** (i.e. up or down) for each cryptocurrency in the chosen risk tolerance cluster.
3. **Regression** predicts the **close price** for the next day, next week, and next year for each cryptocurrency in the chosen cluster.
4. A **Genetic algorithm** optimizes a **portfolio** of coins in the chosen risk cluster.

3 Literature Review

The motivation for this project is based on the continuing growth of the cryptocurrency market and an increasing number of new coins and tokens created each year. By 2018, nearly 1500 different coins and tokens existed with a market capitalization of $830 billion that increases each year. [11] In addition, researchers have shown that coins generally fall within one of two categories, "entrenched incumbent coins" or "explosive immature tokens" [1].

Other researchers have investigated the leading incumbent coins such as Bitcoin and Ethereum, however, research has not included the immature tokens generally known as alt coins.[9][5] This has led to a highly volatile and speculative market for alt coins.

This project provides an evaluation platform for more than just the largest coins (i.e. Bitcoin and Ethereum). It investigates the top 100 coins and tokens by market capitalization and provides a repeatable platform for creating an optimized portfolio that includes alt coins.

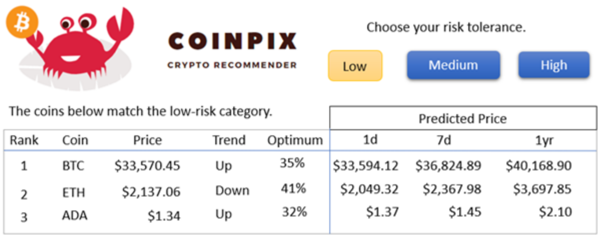
4 Results

4.1 Functionality

The application was built in Angular Javascript with data access through a flask python API. Data mining was performed using Python and PySpark in Jupyter notebooks.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn



Choose your risk tolerance.

The coins below match the low-risk category.

					Predicted Price		
Rank	Coin	Price	Trend	Optimum	1d	7d	1yr
1	BTC	$33,570.45	Up	35%	$33,594.12	$36,824.89	$40,168.90
2	ETH	$2,137.06	Down	41%	$2,049.32	$2,367.98	$3,697.85
3	ADA	$1.34	Up	32%	$1.37	$1.45	$2.10

Figure 1. Conceptual design of CoinPIX interface

Each of the four data mining techniques used in the application were custom developed with all code available in a github repository at https://github.com/nosliwes/coinpix.

The full design of the user interface is shown in the figure and each data mining objective is explained in the following paragraphs. The user can choose a risk tolerance and information is shown regarding existing coins.

Clustering - Clustering was used to categorize all of the coins into one of three clusters corresponding to high, medium and low risk. Although clustering takes place in the application back end, the results are visible in the application when the user selects a risk tolerance.

Classification - Classification was used to determine an individual coins current trend based on daily price data. It is visible in the Trend column.

Regression - Regression was used to predict the future price in 1 day, 7 day, and 1 year intervals. It is shown in the Predicted Price columns.

Genetic Algorithm - The genetic algorithm was used to determine the percentage of each coin in an optimum portfolio. The percentages are shown in the Optimum Portfolio column.

4.2 Datasets

The dataset was obtained from yahoo finance and included Date, Open, High, Low, Close, and Volume for each day. Data was collected from 2018 to 2021 for each of the symbols shown below:

BTC, ETH, ADA, DOGE, XRP, HEX, BCH, LTC, LINK, MATIC, THETA, XLM, VET, ETC, TRX, FIL, XMR, EOS, ALGO, CRO, TFUEL, BSV, NEO, XTZ, MIOTA, LUNA1, MKR, ATOM1, KSM, BTT1, HBAR, RUNE, CHZ, WAVES, DCR, CEL, ZEC, DASH, HOT1, XEM, QNT, ZIL, ENJ, BAT, STX1, MANA, SNX, XWC, ZEN, BTG, NANO, BNT, DGB, ONE2, QTUM, ARRR, ONT, SC, ZRX, OMG, ANKR, ICX, RVN, BCD, XVG, CKB, IOST, RSR, MAID, KNC, HNC, LRC, LSK, KAVA, VTHO, RLC, GNO, BAND, STORJ, MCO, ABBC, FUN, OXT, WAXP, SNT, FET, IOTX, NKN, ANT, REP, BTS, CVC, DERO, MLN, TOMO, AVA, ARDR, XHV, ETN, BCN

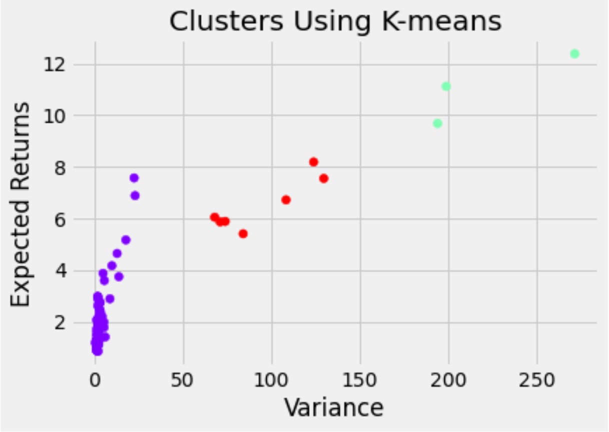


Figure 2. Risk clusters using kmeans and k=3

4.3 Data Mining Tasks

The following data mining tasks were performed to support functionality in the CoinPIX application.

Clustering - The risk of an investment can be determined by plotting expected returns versus price variance. As the price variance increases, the expected returns also increase to offset the higher risk associated with the investment. Similarly, the lower the price variance, the lower the expected returns.

The first data mining task performed was clustering to identify the high risk, medium risk and low risk coins. This was done using the K-means algorithm in the sklearn python library and k=3. The results of k-means clustering are shown in the figure below. The purple cluster corresponds to the lowest risk coins, red are the medium risk coins, and green are the highest risk coins. This information was then used by the application to filter the coins according to the risk tolerance selected by the user.

Classification - The goal of any investor is to buy at low prices and sell at high prices. In the application, classification was used to determine the current price trend of each coin. This was accomplished by adding a new feature to the dataset for the **trend**. Each daily record was then labeled as either "up" or "down" accordingly. The details of the labeling are discussed in the data pre-processing section.

After each label was created for the historical data, a random forest classifier was used to predict whether or not a coin is in an uptrend or a downtrend. The application then uses this information to determine if a coin should be included during portfolio optimization.

Regression - The length of time remaining in the current trend is another important piece of information for any investor. Investors can maximize gains by investing in the beginning of up-trends. Similarly, losses can be avoided by selling before the trend turns back down.

Regression was the third data mining task used by the application and predicts the price at various days in the future. This information is used by the application to project how many days are remaining in the current trend and becomes part of the objective function used later in portfolio optimization.

Genetic Algorithm - The optimal amount of capital to invest in various opportunities is valuable information when building a portfolio. The final data mining task used in the application is a genetic algorithm to construct an optimal portfolio.

4.4 Data Mining Algorithms

Each of the data mining algorithms were custom developed in python and are described below.

Clustering - The most common method for finding clusters in data is the k-means algorithm because of it's speed and simplicity.[6] This algorithm works by randomly placing k centroids in the solution space. It then assigns each sample data point to the nearest cluster and calculates the total distance to the center of the assigned cluster.

After all points are assigned to clusters, the mean of each cluster is determined and compared to the centroid center. If they are the same or fall within a predefined tolerance, the process ends. If not, the new centroid center is set to the cluster mean and the process is repeated until it converges.

The number of clusters, k, was determined using the elbow method. The elbow method works by calculating the sum of the squares distance of the samples to the nearest cluster center for different values of k. The plot below shows the elbow plot for the number of clusters from 2 to 9. From the plot, either k=3 or k=4 are appropriate since the curve flattens after that point. As a result, the value of k=3 was used to match the number of risk levels desired (low, medium, high).

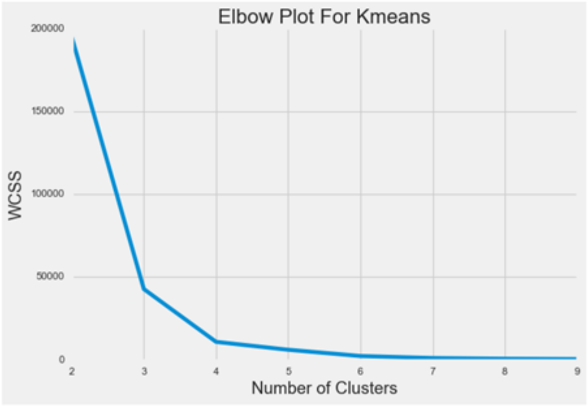


Figure 3. Elbow plot showing the selection of k=3 risk clusters

Classification - The classification algorithm used is a random forest classifier, with random trees generated by the CART method, using the Gini purity score to generate the splits. This method, like many methods, can be boiled down to tree building (or tree growth) and tree pruning.

During tree building, CART builds out to the maximal binary tree.[10] This will result in over-fitting. The algorithm addresses this later by pruning. A given node is split minimizing the children's weighted impurity score. In our case, we use the Gini impurity, but this can be substituted for any appropriate measure.

The pruning strategy for CART is cost-complexity pruning.[10] This means that the model is evaluated against a validation set, which is distinct from the training set. It will then prune the least useful leaves until the model performs well enough against the validation set. This effectively addresses the over-fitting inherent in creating a maximal decision tree during tree growth.

To create a random forest classifier from the CART trees, we generate a number of these trees, randomly selecting records and attributes to be used to train each individual tree.[4] We then use a simple majority vote to determine the class attribute (uptrend or downtrend).

Regression - Linear regression maps continuous inputs to outputs and fit a line which knows where predicted prices will fall in a certain time period. This allows users to see how closely related time is to price either positively correlated or negatively as well as predicting future values of prices based on future time periods. Based on trained data supplied from a history of coin prices over the past year, weights are found to find a straight line function to minimize cost per y = Coef0 + Coef1 * x. The cost function helps decide these values involving minimizing errors between predicted and actual values. Then the regressor is tested against actual values to check for accuracy.[7]

Support Vector Regression builds on support vector machines by creating a best fit line within a threshold. After a training set is collected with a kernel, a correlation matrix is formed from data points in the training set. An estimator is formed from from training.[8]

Genetic Algorithm - Portfolio optimization is subject to practical constraints such as trading limitations, fees, and portfolio size. Increasing constraints leads to a nonlinear mixed integer programming problem that is complex and difficult to solve. [3] For this project, a custom built genetic algorithm was used to construct the optimum portfolio.

The purpose of the genetic algorithm is to perform a stochastic search in a solution space and converge without iterating the entire solution space. The genetic algorithm begins by initializing a portfolio of coins with random values. For the fitness function, it uses the Sharpe ratio, which is a measure of the expected return versus the coin risk.

$$S_a = \frac{E[R_a - R_b]}{\sigma_a}$$

where:

$$S_a = Sharpe\ Ratio$$
$$E = Expected\ Value$$
$$R_a = Coin\ Return$$
$$R_b = Risk\ Free\ Return$$
$$\sigma_a = Coin\ Standard\ Deviation$$

Parents are selected next for crossover, mutation and breeding. The children are then checked for fitness and the process is repeated for successive generations until the stopping criteria is met. The final result is an optimized portfolio describing the percentage of each coin to hold.

4.5 Data Preprocessing

Missing Data - There was no missing data in the dataset since the crypotcurrency market is continuously open and does not close. However, there were coins in the top 100 that were created after 2018. As a result, we removed all coins that did not have 3 years of historical data.

Outliers - The returns and variance showed significant skew and outliers as shown in the boxplots below.

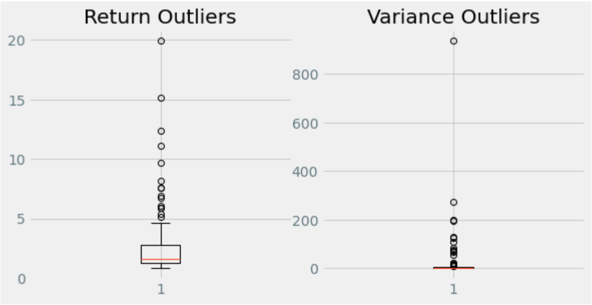


Figure 4. Boxplot showing return and variance outliers

However, the top two variance outliers were also the highest return outliers. As a result, we removed the top two outliers and the remaining dataset showed good separation when clustering.

Calculated Data - The raw data is a collection of OHLCV data (open, high, low, close, volume) from a single day. In order to capture the direction of a trend, it was necessary to include historical data from previous days. As a result, each row was augmented with the simple moving averages (SMA), relative strength index (RSI), and the MACD (Moving Average Convergence Divergence). Adding these values in addition to the OHLCV data allowed the classifier to predict trends from a single day of OHLCV data.

Labeling - In order to train the classification model, the training data needed to be labeled for each row in the training set. This was accomplished by using the moving averages to find crossover points. Then the minimum or maximum was found between each crossover point. Each day where a minimum was found was considered a bottom and each maximum was considered a top. Finally, each day between the bottom and top days were labeled as either up or down trends depending on the direction.

An example is shown in the figure below for Bitcoin. The red and green arrows indicate cross over days where the trend changes. A red arrow is when the trend is starting down and a green arrow is when the trend is starting up. From these points, then the blue arrows were found for the top and bottom of the trends.

4.6 Implementation Details

The main technical components of the CoinPIX application were an Angular Javascript Framework user interface with a Flask application containing a python API. The Flask application contained each of the four

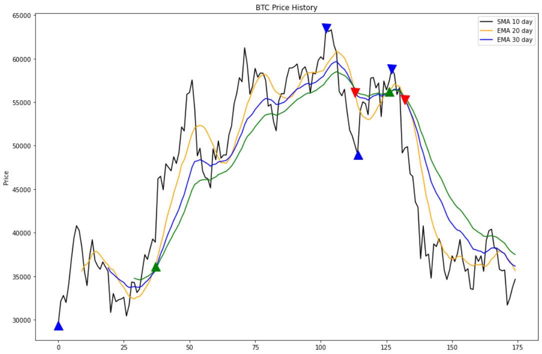


Figure 5. Plot showing labeling approach for trend classification

custom data mining applications used in the project. Data mining activities including data ingestion, cleaning and modeling were perfromed in Jupyter Notebooks using Python and PySpark. All source code is stored in a github repository https://github.com/nosliwes/coinpix.

4.7 Performance Evaluation

Clustering - For k-means clustering, the silhouette coefficient was initially considered to evaluate the separation of the clusters. The silhouette score ranges from -1 to 1 and can be interpreted using the following.[2]

- 1: Clear separation between clusters
- 0: No separation between clusters
- -1: Clusters are assigned the wrong way

The elbow method, however, showed a clear slope change at k=3 and therefore was sufficient to determine the best number of clusters.

Classification - The classification model was evaluated by generating a confusion matrix and by calculating the F1 score of the whole binary forest. The reason the F1 score was included was because, in general, the market for cryptocurrency during the time period we are studying was moving up. This means that there were more uptrends than down-trends affecting the precision-recall for the classification.

Regression - For evaluating regression models, metrics include Mean Absolute Error, Mean Squared Error, and R2 Score. MAE sums absolute values of distances from the fitted line to gauge accuracy for prices. MSE sums squares of distances and focuses on larger prediction errors. R2 subtracts the quotient of sum of squares of residuals by total sum of errors from 1 to find out the regression predictions approximate the actual points where 1 is perfect fit.[5] This project used the MSE sum of squares to evaluate the regression models.

Genetic Algorithm - Genetic algorithms are heuristic in nature and attempt to minimize a fitness function for each successive generation. As a result, genetic algorithms converge on the "best" solution when a stopping criteria is reached. However, there is no guarantee that a genetic algorithm has converged on the global minimum or has found a local minimum instead. For this project, the genetic algorithm was run ten times varying the hyper-parameters each iteration protecting against finding local minimums.

5 Conclusions and Future Work

As cryptocurrency continues to rise, new tools are being crafted to analyze and discuss trends among coins. CoinPIX adds to this body of knowledge by predicting price trends and providing an interface for users to choose coins based on a systematic approach. Through the use of data mining algorithms, the application applies classification, regression, and genetic algorithms to choose coins based on a chosen risk tolerance.

CoinPIX captures the time series nature of the data by aggregating moving averages and locating cross over points. This works reasonably well for trend prediction and classification, however, it does not perform well when forecasting future prices. Additional work could evaluate neural networks instead of support vector regression to better capture the time series nature of the data.

6 References

References

[1] Jethin Abraham et al. "Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis". In: *SMU Data Science Review. SMU Scholar.* 2018.

[2] Ashutosh Bhardwaj. "Silhouette Coefficient". In: 2020. URL: https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c.

[3] Tun-Jen Chang, Sang-Chin Yang, and Kuang-Jung Chang. "Portfolio optimization problems in different risk measures using genetic algorithm". In: *Expert Syst. Appl.* 36.7 (2009), pp. 10529–10537. DOI: 10.1016/j.eswa.2009.02.062.

[4] Misha Denil, David Matheson, and Nando De Freitas. "Narrowing the Gap: Random Forests In Theory and In Practice". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 665–673. URL: http://proceedings.mlr.press/v32/denil14.html.

[5] M. K. Dobbali. "Metrics to Understand Regression Models in Plain English". In: 2020. URL: https://towardsdatascience.com/metrics-to-understand-regression-models-in-plain-english-part-1-c902b2f4156f.

[6] Charles Elkan. "Using the Triangle Inequality to Accelerate k-Means". In: *Machine Learning, Proceedings of the Twentieth*

International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA. Ed. by Tom Fawcett and Nina Mishra. AAAI Press, 2003, pp. 147–153. URL: http://www.aaai.org/Library/ICML/2003/icml03-022.php.

[7] Anas Al-Masri. “How Does Linear Regression Actually Work”. In: (2020). URL: https://towardsdatascience.com/how-does-linear-regression-actually-work-3297021970dd.

[8] Ashwin Raj. “Unlocking the True Power of Support Vector Regression”. In: 2020. URL: https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0.

[9] Akhter Mohiuddin Rather, Arun Agarwal, and V.N. Sastry. “Recurrent neural network and a hybrid model for prediction of stock returns”. In: *Expert Systems with Applications* 42.6 (2015), pp. 3234–3241. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2014.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0957417414007684.

[10] Priyanka Gupta Sonia Singh. “COMPARATIVE STUDY ID3, CART AND C4.5 DECISION TREE ALGORITHM: A SURVEY”. In: *International Journal of Advanced Information Science and Technology* 27.27 (2014). ISSN: 2319:2682. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.4929&rep=rep1&type=pdf.

[11] Ke Wu, Spencer Wheatley, and Didier Sornette. “Classification of cryptocurrency coins and tokens by the dynamics of their market capitalizations”. In: *Royal Society Open Science* 5.9, 180381 (Sept. 2018), p. 180381. DOI: 10.1098/rsos.180381. arXiv: 1803.03088 `[physics.soc-ph]`. URL: https://ui.adsabs.harvard.edu/abs/2018RSOS....580381W.