Capstone Project

Sai Indukuri
April 15, 2018

I. Definition

Project Overview

Cryptocurrency is a digital currency in which encryption techniques are used to regulate the generation of units of currency and verify the transfer of funds, operating independently of a central bank. There is lot of buzz about cryptocurrency and everyday in the news we hear volatility in the prices of cryptocurrencies such as Bitcoin.

Bitcoin is actual implementation of decentralization issued under the consent of participants not the central bank. Hence variants like purchasing power or interest rate parity does not impact Bitcoin. Bitcoin price is largely impacted by demand and supply.16.5 million Bitcoins are mined since it was created and the Bitcoins are capped at 21 millions part of the reason for scarcity and rush. Hence to predict Bitcoin price we need to predict Demand and Supply.

Blockchain is the technology used to create Bitcoin."Block Chain information" along with "Bitcoin" trading information includes features that can determine demand and supply and hence the price. Therefore, the project uses historic blockchain information and Bitcoin trading information to predict future Bitcoin prices. This project uses data from https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory which is publicly available. The data consist of "Bitcoin_Dataset" which consist of historic Blockchain information and Bitcoin prices for last 4 years.

Bitcoin dataset consist of several important blockchain information that might help us understand rise and fall continuum , most importantly, Total bitcoins, Trade volume, block size, avg block size, Orphaned blocks, transactions, hashrate, Mining difficulty and Miners revenue. We have a balanced dataset with around 47% of records with rising prices and 43% of records with falling prices and 10% of records with no price change.

Problem Statement

The extremely nonlinear nature of the crypto market data makes it very difficult to design a system that can predict the future direction of the crypto prices in Bitcoin with sufficient accuracy. Goal of the project is to predict price movements(that's either up or down) of cryptocurrencies such as Bitcoin. As problem involves predicting and classifying Bitcoin prices to rise , fall or remain the same this is a multi class classification problem.
The main goal of speculating a cryptocurrency price is to make profitable trades. To make profitable trades one doesn't need to know accurate value of the cryptocurrency price, but one merely needs to predict whether prices rise or fall or remain the same. In these lines we will train our models to classify subsequent day's closing price to be likely higher, lower than the last or the same as last, based on past days price data.
The solution involves

1. Preprocessing the data
2. Selecting features that can most help find the bitcoin trend
3. Testing the data on selected features on varying time series data using the models SVM, Random Forest and Logistic regression
4. Select the best model based on the models performance measured by Accuracy, Precision and FBeta score.
5. Fine tune the model to improve accuracy.

The final solution will be able to predict the bitcoin price trend(Up, Down, Constant) with accuracy better than that of Naive prediction.

Metrics

With Bitcoin predictions it is very important to Accurately predict price trends. Hence using accuracy as a metric for evaluating a particular model's performance would be appropriate. Additionally, Identifying price rise when price actually drops is detrimental as the investors will lose money. Therefore, a model's ability to precisely predict price rise is more important than the model's ability to recall . We can use F-beta score as a metric that considers both precision and recall:

In particular, when β=0.5β=0.5, more emphasis is placed on precision. This is called the F0.50.5 score (or F-score for simplicity).

In addition to FBeta score, we will also use Precision to measure the performance.

II. Analysis

Data Exploration

Bitcoin is actual implementation of decentralization issued under the consent of participants not the central bank. Hence variants like purchasing power or interest rate parity does not impact Bitcoin. Bitcoin price is largely impacted by demand and supply.16.5 million Bitcoins are mined since it was created and the Bitcoins are capped at 21 millions part of the reason for scarcity and rush.
Hence to predict Bitcoin price we need to predict Demand and Supply. Blockchain is the technology used to create Bitcoin."Block Chain information" along with "Bitcoin" trading information includes features that can determine demand and supply and hence the price. Therefore, the project uses historic blockchain information and Bitcoin trading information to predict future Bitcoin prices. This project uses data from https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory which is publicly available. The data consist of "Bitcoin_Dataset" which consist of historic Blockchain information and Bitcoin prices for 8 years from 2009 to 2017.

Bitcoin dataset consist of several important blockchain information that might help us understand rise and fall continuum , most importantly, Total bitcoins, Trade volume, block size, avg block size, Orphaned blocks, transactions, hashrate, Mining difficulty and Miners revenue.

We have an Unbalanced dataset with around 47% of records with rising prices and 43% of records with falling prices and 10% of records with no price change. Also the dataset contained few blank values and needs to be cleaned.

Algorithms and Techniques

I choose to train the data with Support Vector Machines(SVM) and Random Forest algorithms. Below is the brief definition, strengths and weaknesses of these algorithms,reason I picked the algorithm and techniques applied on the algorithm..

Support Vector Machines(SVM):

Support vector machine uses a kernel mechanism that maximizes distance between closest member of separate classes.

Strengths: Unlike Logistic regression, SVM can model non linear decision boundary and have many kernels to choose from; SVMs are fairly robust against overfitting especially in high dimension space.

Weaknesses:Isn't suited to large dataset as training time might be high; Less effective with noisier dataset with overlapping classes.

Support vector machines may be a good fit for the problem since the dataset is small(~2000)

Ensemble Methods:Random Forest :

Real World application: Most real world applications use some sort of ensemble learning as ensemble learning is a way of combining the output of multiple models.

Strengths: The major strength of Random Forest is that it is very difficult for the model to overfit, because the model only improves as you increase the complexity of the model

Weakness:The major weaknesses to do with Random Forest are related to increased storage and computation. Since RandomForest require multiple classifiers, there is a need for more storage for the classifiers which can be costly. In a similar vein, all of the classifiers need to be processed, rather than just one, which means that the run time of the algorithm is higher

RandomForest may be a good fit since we need to perform binary classification and has to test multiple hypothesis.

Benchmark

Looking at the distribution of classes (price rise(46.58%), drop(42.53%) and no change(10.89%)) there are very few records for 'No change' and a higher percentage of records with price drop, This can greatly affect **accuracy**, since we could simply guess the prices will go down all the time.Making such a statement would be called **naive**, since we have not considered any information to substantiate the claim. We will use a naive algorithm that would always predict price go down as Benchmark.

Following are the benchmark results for the Naive algorithm.

Metrics	Accuracy	Precision	F Score
Benchmark Results	0.4253	0.4253	0.4806

III. Methodology

Data Preprocessing

Before data can be used as input for machine learning algorithms, it often must be cleaned, formatted, and restructured — this is typically known as **preprocessing**. Below are the preprocessing steps performed :

Cleaning:

As with any other realtime dataset bitcoin dataset came with its share of defects.

- There are few blank values within btc_trade_volume
  There are 21 instances where btc_trade_volume is blank. Btc_trade_volume cannot be blank especially when other features have values. Trading volume information is not available during the dates and hence they were blanks. Missing values are handled by replacing them with mean of the btc_trade_volume.
- The records are not sorted on dates.
  It is very important to order the records by dates with the time series problems like bitcoin estimation.The records in the input dataset are not sorted and needs to be sorted.

Transforming:

A dataset may sometimes contain at least one continuous feature whose values tend to lie near a single number, but will also have a non-trivial number of vastly larger or smaller values than that single number. Algorithms can be sensitive to such distributions of values and can underperform if the range is not properly normalized. Bitcoin dataset has around 16 features that are continuous and that need to be transformed. Log transformation is applied on all these features.

Normalizing:

In addition to performing transformations on features that are highly skewed, it is often good practice to perform some type of scaling on numerical features. Applying a scaling to the data does not change the shape of each feature's distribution however, normalization ensures that each feature is treated equally when applying supervised learners. Note that once scaling is applied, observing the data in its raw form will no longer have the same original meaning, as exampled below. Bitcoin dataset has 22 numerical features that are scaled and normalized.

Implementation

Implementation can be divided in to 3 steps

Feature selection:

We can determine relevance of each of the input feature by training a supervised random forest regression learner on the data. I have used ‘Mean decrease impurity’ within random forest regressor to identify the feature importance and feature importance is determined based on the features that has most information gain/entropy. Features are then ordered based on Feature importance and features are selected that would provide 80% of information Gain. Below are the 13 features shortlisted based of feature importance out of total 24 features.

btc_trade_volume
btc_transaction_fees
Btc_estimated_transaction_volume_usd
Btc_estimated_transaction_volume
Btc_n_transactions_per_block
Btc_median_confirmation_time
Btc_n_unique_addresses
Btc_output_volume
Btc_avg_block_size
Btc_miners_revenue
Btc_n_transactions
Btc_n_transactions_excluding_chains_longer_than_100
btc_market_cap

Creating a training and predicting pipeline:

To properly evaluate the performance of each model , it's important that we create a training and predicting pipeline that allows us to quickly and effectively train models using various sizes of training data and perform predictions on the testing data. I have used TimeSeriesSplit algorithm to split the data samples across fixed time lines and to create a training and predicting pipeline across the timelines.
I have created a 3 fold split where the algorithm splits the data and creates training sets as supersets of those that come before them. So we have 3 sets of training and prediction pipeline each with 730 testing samples and 730,1460 and 2190 training samples.

Training and Testing the Model :

SVC and Random tree algorithms are time series trained and tested with the training data that is split as discussed in the previous section. Algorithms are trained with varying sample size 730,1460 and 2190 and tested on 730 samples each time.

Refinement

I used grid search (`GridSearchCV`) to refine the Random forest algorithm.

Below are the parameters and scoring used to refine the algorithm.

Parameters

- Varying size of estimaters(100,200,300) are used to allow random forest to test on varying number of trees.
- Max_features of ‘sqrt’ will allow us to select sqrt of all features while searching for a best split. This will limit number of features considered for split and thus avoid overfitting.
- Min samples leaf with minimum values of (1,5,10,50,100,200,500) would help us try GridSearchCV with varying samples. Since the Bitcoin dataset is imbalanced it makes more sense to *use low*er values to avoid overfitting.

Scoring

When the dataset is imbalanced Accuracy is misleading. Hence It makes more sense to score the algorithm on Precision due to the the imbalanced bitcoin dataset. Accordingly, Algorithm is tuned and scored based on precision score.

IV. Results

Justification

Below are the results of the optimized model compared with unoptimized and Benchmark.

	Benchmark	Unoptimized	Optimized
Accuracy	0.4253	0.56	0.9274
Precision	0.4253	0.54	0.9271
Fscore	0.4806	0.53	0.9289

Clearly unoptimized model performed better than benchmark and model performance has significantly improved after optimizing in all the metrics the model is evaluated.

The optimized solution is satisfactory and has performed to the expectations. The optimized model is built by tuning the parameters as discussed in previous section and is significant enough to solve the problem.

V. Conclusion

Reflection

Predicting Bitcoin price is not only interesting but also challenging. Challenging because the problem statement needs lot of research to understand the variables in the bitcoin price trend. As any real time problems Bitcoin dataset is not clean and the dataset is cleaned by replacing empty data with mean of the features and by formatting/sorting the dataset on the dates.

After Data cleansing the data needed to be transformed as most of the features are continuous and are highly skewed. In addition there are varying units among the feature which needed us to scale all the features.

Once data is cleansed, transformed and normalized we have selected the features that matter most by training the data on random forest algorithm and choosing the features that provided 80% of the information gain. The features are later trained on SVC and Random forest ensemble algorithms on a time series split.

Unoptimized model performed comparatively better than benchmark model and the model has been tuned well with accuracy and precision over 0.9. Final model and solution has fit well to my expectations and can be used in general setting to solve any cryptocurrency trend estimation problems.

Improvement

I see further scope for improvement in the solution in the following areas.

1. Although bitcoin prices are largely affected by demand and supply continuum, prices can also be predicted from the sentiment analysis. The algorithm can further be improved by applying sentiment analysis to predict price movement.
2. Since Bitcoin is relatively new the data available is limited and very small. To have a larger dataset algorithm can further be improved to predict on lower frequency time series (5 mins, 10 mins, 30 mins) instead of a day.

I believe improving the algorithm in above two areas might further improve the performance.