Stacking with Neural Network for Cryptocurrency Investment

Contact Information:
InnCreTech
103 Carnegie Center
Princeton, NJ 08540
Email: avinash@inncretech.com

Avinash Barnwal, Hari Pad Bharti, Aasim Ali and Vishal Singh
InnCreTech

Abstract

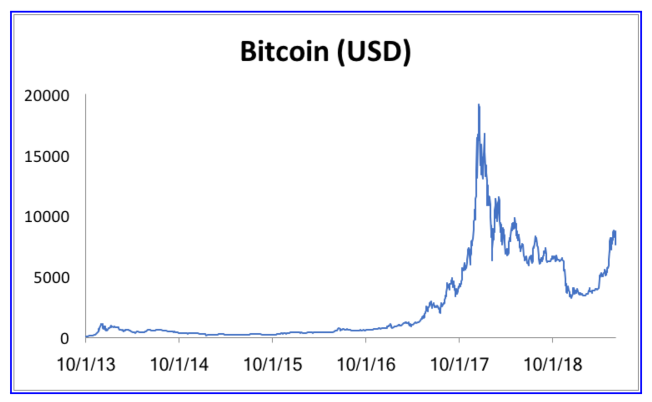
Predicting the direction of assets have been an active area of study and difficult task. Machine learning models have been used to build robust models to model the above task. Ensemble methods are one of them resulting better than single supervised method. We have used generative and discriminative classifiers to create the stack, particularly 3 generative and 6 discriminative classifiers and optimized over one-layer Neural Network to model the direction of price cryptocurrencies. Features used are technical indicators not limited to trend, momentum, volume, volatility indicators and sentiment indicators. For Cross validation, Purged Walk forward cross validation has been used. In terms of accuracy, we have done comparative analysis of the performance of Ensemble method with Stacking and individual models. We have also developed methodology for features importance for stacked model. Important indicators are identified based on feature importance.

Motivation

- Market crash in 2018.
- Non-linear factors leading the market.
- Need for developing feature importance of stacked models.

Data

- Bitcoin data from Quandl.
- Four exchanges KRAKEN, BITSTAMP, ITBITUSD and COINBASE to remove ambiguity.
- Final price is based on weighted volume of end of day prices.
- Exponential average technique for missing value treatment.
- Time Period - Aug-2017 to Jul-2018 with end of the day data.



Feature Creation

Cryptocurrency: Bitcoin

- **Volume Features** - ADI, Balance, Chaikin, Price ,Negative volume - 7 Features.
- **Volatility Features** - Range, Bollinger, Channel - 15 Features.
- **Trend Features** - MACD, VI, MI, CCI, Oscillator - 11 Features.
- **Momentum Features** - Strength, Oscillator - 5 Features.
- **Sentiment Features** - Positive, Negative and Neutral - 3 Features.

Modeling

Lets say we have feature vector $X_t \epsilon R^n$ to build the model having dependent variable y_{t+1} , here y_{t+1} is defined based on the return of the asset r_{t+1} where

$$y_{t+1} = \begin{cases} 1, & r_{t+1} > 0 \\ 0 & r_{t+1} \leq 0 \end{cases} \tag{1}$$

Modeling Activities

Models: *Discriminative, Generative*

Discriminative

- Lower Asymptotic Error
- Directly Learns $P(Y|X)$
- Used - Xgboost, LightGBM, KNN, Logistic Elastic Net, SVM and RF - 6 Models.
- Backward Inference - Inference of X not possible.
- Computation intensive.

Generative

- Higher Asymptotic Error
- Learns $P(Y|X)$ based on $P(X)$ and $P(X|Y)$.
- Used - Naive Bayes, LDA, QDA - 3 Models.
- Backward Inference - Inference of X possible
- Less Computation intensive.

Time-Frame

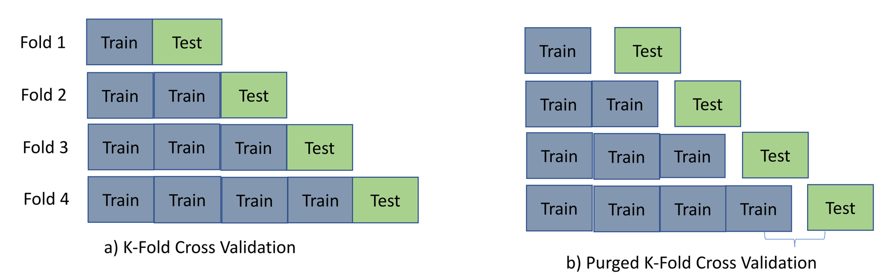
- **Training Period** - Aug-2017 to Mar-2018.
- **Testing Period** - Jul-2018.

Cross-Validation

Robust Cross-Validation is important step to find the hyper-parameters. There are different techniques to find the hyper-parameters given cross-validation folds such as random sampling and bayesian optimization. We have used random sampling. K-Fold cross-validation leads to lower error but performs worst during the live performance.

Cross-Validation Techniques

- Purged Walk Forward 5 - Fold
- To avoid look ahead bias



Fold Distribution

Fold No	Training Period	Test Period
Fold 1	Aug-Oct'17	Nov'17
Fold 2	Aug-Nov'17	Dec'17
Fold 3	Aug-Dec'17	Jan'18
Fold 4	Aug'17 - Jan'18	Feb'18
Fold 5	Aug'17 - Feb'18	Mar'18

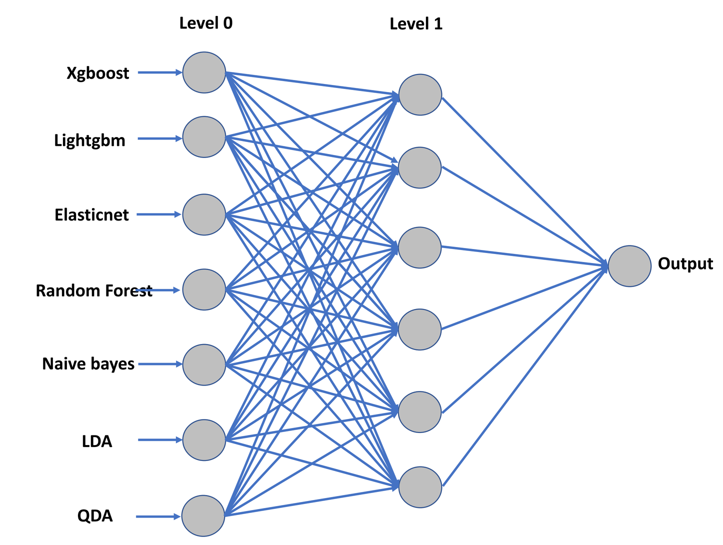
Stacked-Generalization

[2] talks about technique to reduce generalization error rate. It aims to achieve generalization accuracy by combining weak learner. It is considered to be more sophisticated than winner-takes-all strategy.

It generally creates different levels of models having one level of output being the input for next level. Primarily, it removes the biases of all models leading to generalization of all the models.

Stacked Models

- Level 0 has 7 models and Level 1 is hidden layer with 6 nodes.
- Level 0 - Training Period Aug-2017 to Mar-2018.
- Level 1 Hidden Layer - Training Period Apr-2018 to June-2018.



Feature Selection

Tree models - Feature importance is synonyms to Feature selection methodology. Following are the ways to calculate feature importance:-

- Gain - In spliting features , we calculate the decrease in gini impurity or entropy which is finally combined to create combined Gain for each feature.
- Real Cover - Similarly , when features are splitted, the split occurs over observations. We count the observations where split occured and it is finally combined leading to Real Cover.

Partial Dependence Plot-

[1] and [3] talks about visualizing the feature importance by estimating the variability in the estimated function by varying each particular variable and keeping other variables at their average.

Partial dependence function for regression is defined below:-

$$\hat{f}_{x_t}(x_t) = E_{x_{-t}}[\hat{f}(x_t, x_{-t})] = \int \hat{f}(x_t, x_{-t}) dPx_{-t} \tag{2}$$

- **Monte Carlo** - To estimate the partial function.
- **Uncorrelated** - Features are uncorrelated.

$$\hat{f}_{x_t}(x_t) = \frac{\sum_{i=1}^{n} \hat{f}_{x_t}(x_t, x_{-t}^{(i)})}{n} \tag{3}$$

For variable x_t

$$imp(x_t) = \begin{cases} \sqrt{\frac{\sum_{i=1}^{n}[\hat{f}(x_t i) - \bar{\hat{f}}(x_{ti})]^2}{n-1}} & \text{if } x_t \text{ is continuous} \\ \frac{max_i(\hat{f}(x_{ti})) - min_i(\hat{f}(x_{ti}))}{4} & \text{if } x_t \text{ is categorical} \end{cases}$$

Steps

- Analyze level-0 to calculate variable importance for each generalizer.
- Analyze variable importance for each generalizer.
- Calculate variable importance weighted over importance of each generalizer.

We have K-generalizer then $imp(x_t)$ is calculated as below:-

- $imp(T_k)$ for given model T_k based on feature importance model.
- $w_k = \frac{imp(T_k)}{\sum_{k=1}^{K} imp(T_k)}$
- $imp_k(x_t)$ is calculated based on above formula for each T_k
- $imp(x_t) = \sum_{k=1}^{K} w_k * imp_k(x_t)$

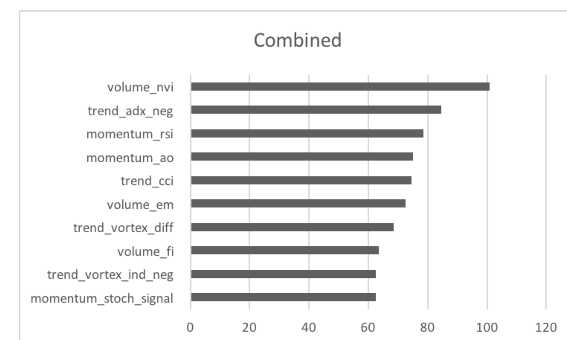
This is model independent variable importance for each feature.

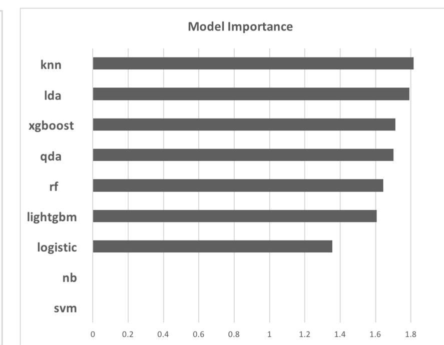
Result

To measure the performance of machine learning models AUC, Accuracy, Precision, Recall and F1 metrics are used.

SG

Parameter	Apr-May 2018	June - July 2018
AUC	0.61	0.50
Accuracy	0.52	0.54
Precision	0.61	0.52
Recall	0.59	0.59
F1	0.60	0.55





Conclusion

- Best Standalone Model Quadratic Discriminate Analysis with 0.52 accuracy, 0.56 precision and 0.55 AUC.
- Worst performing models - SVM and KNN.
- Stacked Generalization Improved accuracy from 0.52 to 0.54.

References

[1] Ghaith Abdulsattar A. Jabbar Alkubaisi, Siti Sakira Kamaruddin, and Husniza Husni. Stock market classification model using sentiment analysis on twitter based on hybrid naive bayes classifiers. *Computer and Information Science*, 11:52–64, 2018.

[2] H. Jang and J. Lee. An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. *IEEE Access*, 6:5427–5437, 2018.

[3] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 841–848, Cambridge, MA, USA, 2001. MIT Press.