Name: Zhengyang Dong

School: Middlesex School

Country/State: Massachusetts, US

Instructor: Xinkai Fu

Title: Ensemble Methods of Machine Learning Models for Critical Metal Stock Trend Prediction

Ensemble Methods of Machine Learning Models for Critical Metal Stock Trend Prediction

Zhengyang Dong

Abstract

The development of modern technology is closely related to the use of metals, and metal demand has been shifting from common industrial metals to a variety of minor metals, such as cobalt or indium. The industrial importance and limited geological availability of some minor metals have led to them being considered more "critical." In this research, we develop a novel approach to investigate metal criticality, by exploring stock prices of major producers of critical metals. Specifically, we use regularized linear regression, logistic regression, support vector machine and gradient boosted trees to predict the trend of stock prices. Then, an ensemble of these base models is used for final prediction. Our ensemble methods include majority vote, logistic regression stacking, and gradient boosted trees stacking. Based on a misclassification error of 0.34 in the validation set, we further develop a stock trading strategy, which leads to a back tested return of 313%, or an excess return of 147%. Moreover, a set of significant features selected by our models suggests that markets of different metals are closely correlated, and exchange rates also play important roles in influencing stock prices.

1 Introduction

Acquirement of resources has been a continual interest in human history due to their contributions to developments of society. Among the numerous resources human utilizes, metal is an especially critical one, constituting the foundation of almost all technological innovations, from bronze weapons to steam engines. Therefore, an increasing demand for metals has been an unsurprising trend for hundreds of years. However, over the past century, the total demand of metals grew at an unprecedented rate, with the demand in the United States alone increasing 20-fold from around 160 million to 3.3 billion tons (Morse & Glover, 2000). Moreover, the demand for metals in technology has been shifting from several major metals, such as iron and copper, to numerous minor metals, such as cobalt and indium, for specialized uses. For example, the largest use of cobalt now is in common rechargeable batteries (Gunn, 2014), and modern photovoltaic cells require a wide range of minor metals such as indium and germanium (Bleiwas, 2010). Moreover, many of these minor metals do not constitute their own ores but exist in low concentrations in ores of other common metals, such as copper or aluminum. Table 1 shows an example of minor metals that are obtained from major industrial metals, sometimes referred as their "hosts."

Table 1: Minor metals obtained as byproducts of major industrial metals. The metals in the title row are the hosts and the metals below them are corresponding byproducts (Gunn, 2014).

Copper	**Zinc**	**Tin**	**Nickel**	**Platinum**	**Aluminum**	**Iron**	**Lead**
Cobalt	Indium	Niobium	Cobalt	Palladium	Gallium	REE	Antimony
Molybdenum	Germanium	Tantalum	PGM	Rhodium		Niobium	Bismuth
PGM	Cadmium	Indium	Scandium	Ruthenium		Vanadium	Thallium
Rhenium				Osmium			
Tellurium				Iridium			
Selenium							
Arsenic							

In fact, many of these increasingly demanded minor metals have high measures of metal criticality. Metal criticality is the degree of the possibility of a metal ceasing to be abundantly available in the future (Gunn, 2014). Because the minor metals required by modern technologies only exist in trace amounts and depend on their host metals, there are concerns with the stability of their supply. Therefore, metal criticality is a predictive measure that aims to alleviate future problems of shortages if the metal's potential scarcity is known in advance. While measuring metal criticality highly complicated, and multiple analytical methods exist, the method proposed by Graedel et al. is a relatively extensive one (Graedel et al., 2012), which considers three dimensions: supply risk, environmental implications, and vulnerability to supply restriction. Figure 1 is an example of the criticality measure of several metals using this method.

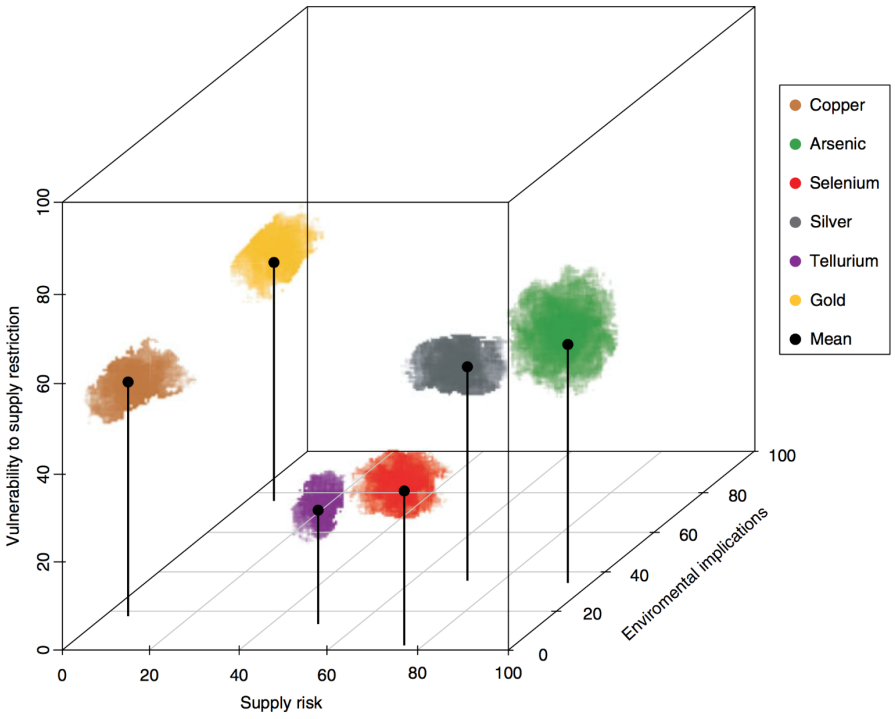

Figure 1: Example of criticality measure using method of Graedel et al. (Graedel et al., 2012). Figure from Nassar et al. (Nassar et al., 2012).

Each dimension incorporates several evaluation metrics. For example, the dimension of supply risk includes the companionality, or dependence, of a metal to its host metals as a criticality indicator. Although metal

criticality is often found to be closely related to its companionality, a clear relationship between a byproduct metal and its host is not well understood.

This research investigates the stock prices of the companies that are major producers of critical metals. Prices of a company's stock reflect the company's performance, which may then impact the production of certain critical metals if that company is a major producer. Therefore, by creating a model that predicts the trend of the stock prices of such companies, not only a valuable investment strategy can be developed, but also some indirect factors that influence critical metal production may hopefully be discovered. This research thus differs from conventional methods, i.e. criticality measures, that investigate critical metals. Criticality measures often involve weighted averages of a chosen set of criticality indicators, inevitably introducing subjective biases in indicator selection. In contrast, this research, by incorporating a large feature set and selecting the features objectively with machine learning techniques, captures important information that is influential to critical metals, which may not seem obvious by common sense.

There are many common approaches to predicting stock prices. Conventionally, financial indicators such as the dividend yield (Fama & French, 1988) or mean reversion in stock prices (Poterba & Summers, 1988) can be used to estimate stock returns. However, this research utilizes machine learning models. One of the most popular machine learning models for such time series data is the support vector machine. In a certain case, a single support vector machine model with different feature selection methods is already able to yield an average accuracy of 85.4% in stock trend prediction (Lee, 2009). Moreover, there have been a number of attempts to use neural networks for stock prediction, such as Schöneberg's research which yields the best result of 68% accuracy (Schöneburg, 1990). Furthermore, in hybrid models, genetic algorithms (Choudhry & Garg, 2008) and decision trees (Tsai, C.-F. and Wang, 2009) are also incorporated. In recent years, with progress in the field of natural language Processing, there have also been text mining approaches to stock prediction, such as mood analyses of twitter feeds (Bollen, Mao, & Zeng, 2011). However, text mining does not provide insights into valuable indicators that influence the outcome, because they are purely based on behavioral economics.

For this research, an ensemble of some common machine learning models is used. The motivation behind ensembles is that a combination of predictions made by multiple weak classifiers can often outperform each individual (Berk, 2006). A simple ensemble method is to take the majority vote of individual classifiers, and Sun and Li offer an extensive comparison between individual performances and the majority vote performances of support vector machines (Sun & Li, 2012). Stacking, another common ensemble method, utilizes a two-level learning scheme: in the first level, some individual base models are developed; in the second level, prediction of the first-level classifiers are used as "meta features" for a new classification model for stacking. It has been found that stacking results usually outperform every individual base model (Wolpert, 1992). For this research, the methods of logistic regression and extreme gradient boosting, or XGBoost, will be used for stacking in the second level. The details will be introduced later in the article.

2 Data and Feature Engineering

The features include various metal and commodity indices, such as the London Metal Exchange index; prices of chemicals used frequently in industrial productions; prices of energy resources, including crude oil, natural gas, coal, and electricity; currency exchange rates between currencies of major metal producing countries and US dollar; and an extensive record of history prices of base metals and precious metals, as well as records of their storage in different warehouses around the world.

The output data includes stock prices of a major producer in critical metals. Specifically, stock prices of Jinchuan Group International Resources (coded 2362.HK) on Hong Kong Stock Exchange are first examined, a company with a large share in the industry of international non-ferrous mining and is a major producer in cobalt.

For the purpose of this research, historic data from the entire year of 2015 is used for training. Data of the first half of 2016 is used for testing, and data from the second half of 2016 to the end of the first half of 2017 is used for validation and final calculation of the return of the model.

2.1 Data Selection and Motivation

There are roughly 700 initial features, which would undergo a feature selection process later. Since an apparent correlation between a specific set of key features and the trend in the stock prices of major producers of critical metals has not yet been established in the field, the decision of using a large set of initial features is made behind the motivation that there might be unexpected features which would effectively reflect the stock prices. A large set of features avoids the danger of missing out less apparent key features.

The selection of these initial features is based on hypotheses of the features' possible relationship with stock prices of critical-metal-related companies, no matter how small the correlation appears to be in common sense. For example, features such as metal prices and volumes of production are apparently influential and are included. In addition, currency exchange rates, a set of less apparent features, are also included. It is known that metal prices and exchange rates of currencies for major metal producing countries may be strongly correlated in some cases. For example, it was found that Chilean Peso and copper prices are highly correlated with a correlation coefficient of 0.93 between from July 7 to September 7, 2015 (David Meyer, 2015).

For the output data, since the research focuses on critical metals, only companies that have large shares in critical metal production are examined. For example, Jinchuan Group, whose copper production is large, also has a considerable share in the production of refined cobalt (a byproduct of copper), having an annual capacity of 10,000 tons and constituting around 10% of the world production (Apodaca, 2016).

2.2 Data Cleaning and Pre-processing

All the data is cleaned at first. We only select data with daily frequencies, and features with more than 50% of the observation being NA or 0 are omitted. Afterwards, all NA values in the leftover features and outcomes are replaced with the last non-NA observation. In doing so, the difference from the last non-NA observation will be reflected on the next non-NA observation, ignoring the NA observations in between.

After the cleaning, all observations are first differenced in order to satisfy the stationarity assumptions of regression models. This is a common approach in financial and economic studies involving the use of time series data (Priestley, 1983). Moreover, since the final goal is to predict the trend in stock

prices, changes in the time series data are more important than its actual values. Afterwards, the first five lags of all features are used instead of the lead terms, since we must only use past data for future prediction. A total of five lags also quintuples the number of features and can roughly capture information in the period of a week prior to the date of prediction.

Furthermore, highly correlated features within the feature set are omitted in order to remove redundancies. Specifically, a correlation matrix is generated and one feature in the pair of features whose correlation is larger than 0.95 is removed. Figure 2 shows the correlation matrix for a subset of features, and we can see that some features, such as different commodity indices, exhibit strong pairwise correlation.

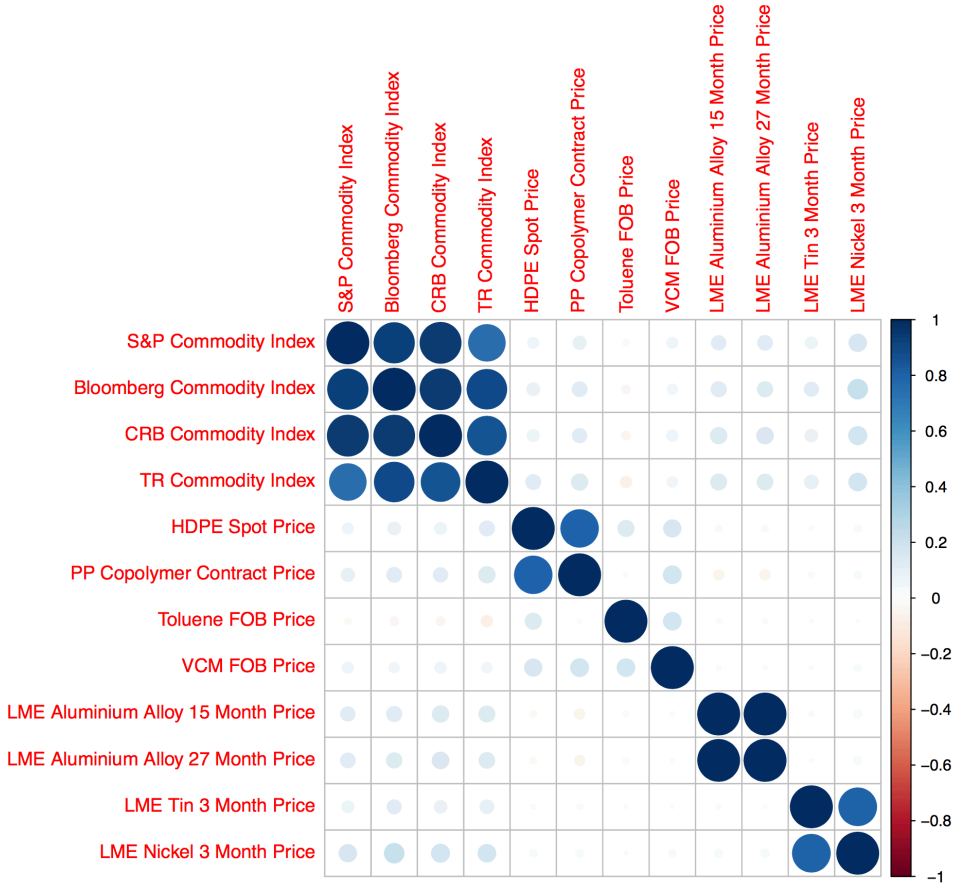

Figure 2: Correlation matrix of a sample set of features. Darker colors represent higher correlations, corresponding to the legend on the right.

Lastly, before feature selection, the outcome is converted from continuous, first-differenced values to binary values. This is because the final goal is to classify the trend of stock price into rise or fall. Thus, values greater than or equal to 0 are converted to 1 to signify a rise in price, and negative values are converted to 0 to signify a fall in price.

2.3 Feature Selection

Since we start with several features, feature selection is necessary for both reducing the computational complexity and avoiding overfitting. A combination of three methods are used for feature selection, namely, univariate regression, stepwise regression and L1 regularization.

2.3.1 Concepts of Linear Regression

All the feature selection methods are heavily based on concepts in linear regression. Therefore, a brief introduction to the core concepts is worthwhile. Linear regression is a basic model which assumes that the outcome is a linear combination of the features. The form of linear regression is:

$$f(\boldsymbol{X}) = \beta_0 + \sum_{j=1}^{p} \boldsymbol{X}_j \beta_j$$

or, in matrix form:

$$f(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{\beta}$$

The residual sum of squares is used as the loss function because of its convenience for optimization.

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j \right)^2$$

By differentiation with respect to $\boldsymbol{\beta}$, the unique solution for $\boldsymbol{\beta}$ is:

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

2.3.2 Univariate Regression

We first perform univariate regression on all features individually. In the evaluation of regression models, a common metric is the coefficient of determination, or the R^2, which reflects the performance of

the model. R^2 is defined to be the proportion of the variance of output observations that can be predicted from input observations. Therefore, the higher the R^2 is, the better the model performs.

The evaluation of R^2 is utilized for feature selection. The method is to first fit a linear model between every single feature and the outcome. Then all the features are ranked by the R^2 values of the univariate regressions. Although these R^2 values are relatively small because only a single feature is used each time, they can still reflect the individual correlation each feature has with the outcome on a relative scale. Top 150 features with the highest R^2 values are selected. However, as an additional testing, all the 150 selected features must have p-values (t-test) smaller than 0.2, and those that fail this test are still removed. The R^2 method is used as the first step because controlling the number of the resulting selection is easy so it can serve as a rough filter for the subsequent steps.

2.3.3 Method of Stepwise Regression

Stepwise regression is a technique that fits regression models by automatically deciding which features to include. It involves a chosen model selection criterion. For this research, the Bayesian Information Criterion (BIC) is used. It can be expressed as follows:

$$BIC = \ln(n)\, k - 2\ln(\hat{L})$$

where n is the number of observations, k is the number of regressors in the current model and $\hat{L}$ is the maximized value for likelihood function. The likelihood function is:

$$\prod_{i=1}^{n} p(y_i|x_i;\boldsymbol{\beta}, s^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(y_i-\boldsymbol{\beta}x_i)^2}{2s^2}}$$

Compared to maximum likelihood estimation, BIC avoids overfitting by adding a penalty term, $\ln(n)\, k$, that is proportional to the number of features in the current model.

Stepwise regression seeks to minimize the BIC of the regression model in a step-by-step manner. There are three possible directions for stepwise regression: forward, backward, and bidirectional. The forward selection begins merely with a constant in the model and calculates a new BIC for the single feature it adds at each step. The feature that results in the lowest BIC is selected and the step is repeated until no addition of new features lowers the BIC score. The backward elimination begins with all the features and

eliminates one feature which lowers the BIC score the most at each step. The bidirectional method starts with a given number of features, and chooses to either add or eliminate a feature, whichever way that lowers the BIC the most. Thus, the stepwise regression essentially evaluates the efficiency of the features because the contribution to the maximized likelihood value must overcome the penalty on the number of features. For this research, all three directions of stepwise regression are attempted for feature selection.

2.3.4 Method of L1 Regularization

Regularization is a common machine learning technique to avoid overfitting. In the final step, L1 regularization is used because it can set coefficients of features to zero (compared to L2 regularization), thus achieving the goal of feature selection. Compared to linear regression using least squares, the objective function for L1-regularized regression adds an L1 penalty term:

$$min_{\boldsymbol{\beta}}\{\|\boldsymbol{y}-\boldsymbol{X\beta}\|^2+\lambda\|\boldsymbol{\beta}\|_1\}$$

where $\|\boldsymbol{\beta}\|_1$ is the L1 norm of the vector, i.e. $\|\boldsymbol{\beta}\|_1=\sum_{i=1}^{n}|\beta_i|$. λ is the regularization parameter.

L1 regularization can set coefficients of the features to zero. Take a two-dimensional situation for instance, the regularization constraint is in a diamond shape, so when the contour of the loss function touches the constraint, it might be on either one of the axes, making one estimated coefficient to be zero. Figure 3 is a visualization of L1 regularization.

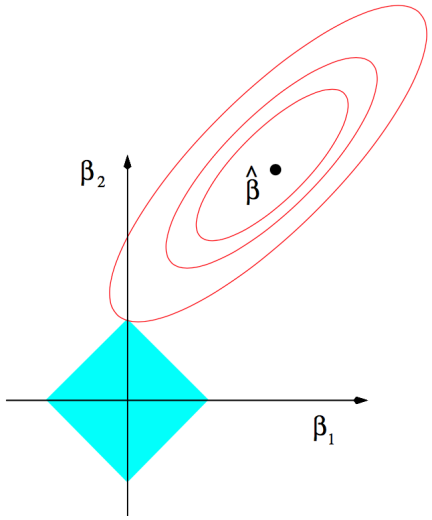

Figure 3: Visualization of L1 regularization when only two features are present. The blue diamond is the regularization constraint, and the red contour is the value of the loss function, i.e. least squares for generalized linear regression. Figure from Hastie et al. (Hastie et al., 2008).

We perform a 10-fold cross-validation to select the optimal lambda value that minimizes the cross-validation error. Figure 4 shows the cross-validated mean absolute error as a function of lambda.

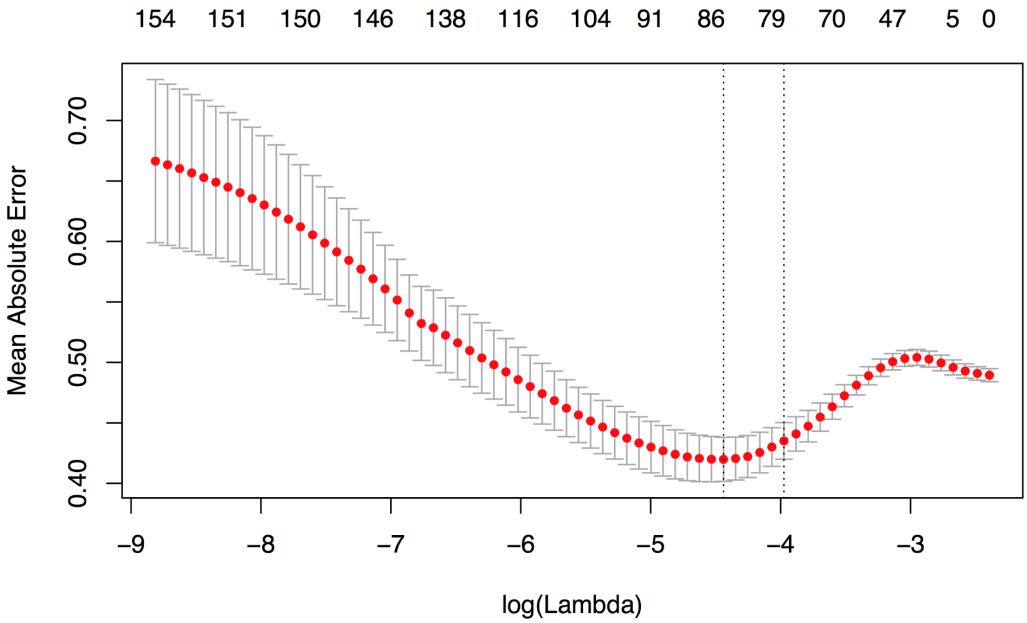

Figure 4: Cross-validation for selection of regularization parameter. The top x-axis is the number of features. The first dashed vertical line on the left shows the position of the lambda leading to the best MSE, while the second dashed line on the right shows the lambda that is one standard deviation away. The error bars on each point show their standard deviations.

Therefore, 85 features are selected as the final feature set for training models. The real-life significances of these features are discussed later.

3 Models and Methodology

The four base models used for the ensemble are introduced first. Then we discuss the ensemble and stacking methods. The formula and proof of the models in this section are based on Hastie, Tibshirani, and Friedman's textbook, *Elements of Statistical Learning* (Hastie, Tibshirani, & Friedman, 2008). Figure 5 is an illustration of the entire approach, where we start from data cleaning and feature engineering to the final ensemble of base models.

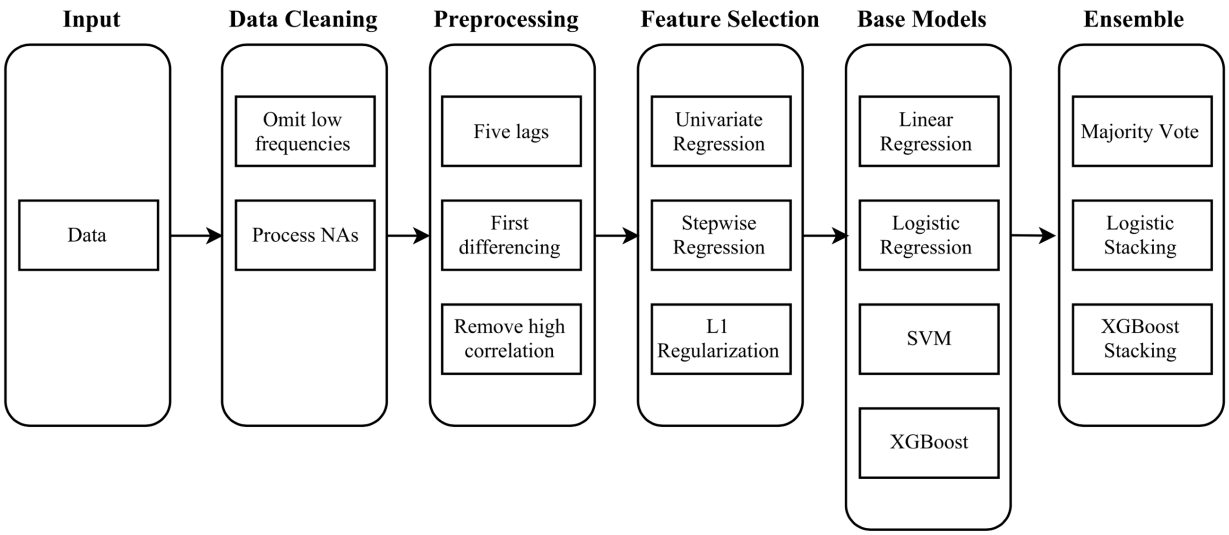

Figure 5: Flowchart of the entire process, from data cleaning to ensemble.

3.1 Linear Regression

Linear regression, introduced before as the foundation for our feature selection methods, is also used as the first base model with elastic net regularization. Elastic net regularization is a weighted combination of L1 regularization and L2 regularization. The objective function is thus:

$$min_{\boldsymbol{\beta}}\{\|\boldsymbol{y}-\boldsymbol{X\beta}\|^2+\lambda(\alpha\|\boldsymbol{\beta}\|^2+(1-\alpha)\|\boldsymbol{\beta}\|_1)\}$$

where λ, the regularization parameter, is greater than 0, and α, the relative strength of L2 regularization compared to L1, is between 0 and 1.

The motivation behind using linear regression is that as a simple model, interpretation of the trained model is intuitive and provides valuable insights into the relationship between the features and the outcome. For example, after normalization, the coefficients of the features essentially represent the relative importance of each feature to the prediction of the outcome.

For this research, the optimal regularization parameter, λ, is obtained from testing a ten-fold cross validation. The optimal relative strength of L2 regularization, α, is found through an exhaustive grid search. The plot of the selection process for α is shown in Figure 6.

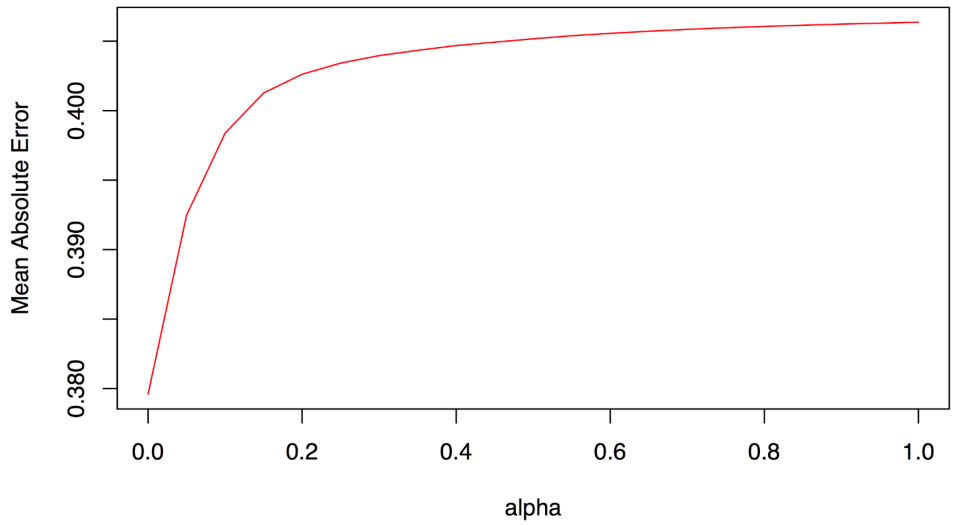

Figure 6: MSE corresponding to α values in the regularization, while λ is fixed.

Therefore, the relative strength of L2 regularization is 0, indicating that an L1 regularization optimizes the model. However, the mean absolute error does not change much, suggesting that the relative strength of regularization does not influence the performance of the model significantly.

3.2 Logistic Regression

Logistic regression is used as the second model. In the general case, when provided with multiple classes, the logistic regression model gives the probability of resulting in each class, and the probability sums to one. For this research, logistic regression with two classes is considered. Essentially, a logistic function is used to fit the outcome. A logistic function can be expressed in the following form:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

where L is the limit, or the maximum value of the logistic function. x_0 is the x-value of the curve's midpoint, and k is the steepness of the curve. Therefore, a binary logistic regression model can be simply created by modifying a logistic function to represent a linear function. Specifically, the limit is set to 1 in order to signify binary classification with classes 0 and 1. The steepness and midpoint are also analogous to the slope and intercept of a linear function, respectively. Thus, in a univariate case, the logistic function can be modified as:

$$F(x) = \frac{1}{1 + e^{-(\beta_0+\beta_1 x)}}$$

or, in matrix form of which the linear function is parameterized by $\boldsymbol{\theta}$:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{x}^T\boldsymbol{\theta}}}$$

Therefore, the model gives the probability of resulting in class 1 or 0 when a set of features and parameters are given:

$$Pr(y|\boldsymbol{x};\boldsymbol{\theta}) = h_{\boldsymbol{\theta}}(\boldsymbol{x})^y\big(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})\big)^{1-y}$$

The parameters are estimated through maximum likelihood, and the likelihood function is expressed as follows:

$$L(\boldsymbol{\theta}|\boldsymbol{X}) = Pr(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\theta}) = \prod_i Pr(y_i|x_i;\boldsymbol{\theta})$$

Similar to linear regression, elastic net regularization is performed, resulting in the final objective function:

$$max_{\boldsymbol{\theta}}\left\{\left(\sum_i logPr(y_i|x_i;\boldsymbol{\theta})\right) - \lambda(\alpha\|\boldsymbol{\theta}\|^2 + (1-\alpha)\|\boldsymbol{\theta}\|_1)\right\}$$

Since a logistic regression model can be transformed into a linear regression model, logistic regression seems to differ little from linear regression. However, logistic regression undergoes a logistic transformation, and the optimization is done through maximum likelihood rather than least squares. As a result, the expectation is that logistic regression would capture some non-linear relationships between the features and the outcome. The optimal regularization parameter and relative regularization strength are found in the same way as linear regression.

3.3 Support Vector Machine

Support vector machine (SVM) is the third base model, which is based on support vector classifiers (SVC), a method which finds an optimal separating hyperplane in the feature space. We briefly introduce support vector classifiers (SVC) here, and then we generalize it to the nonlinear case.

SVC aims to maximize the margin that the two classes are from the separating hyperplane. Consider a feature space $\boldsymbol{x}_i \in \mathbb{R}^P$, in which all the points belong to either of the two classes, $y_i \in \{-1, 1\}$. Then a hyperplane can be defined as:

$$f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\beta} + \beta_0 = 0$$

$\boldsymbol{\beta}$ is a unit vector orthogonal to the hyperplane. Therefore, a classification rule based on this separating hyperplane can be defined as:

$$G(\boldsymbol{x}) = sign(\boldsymbol{x}^T\boldsymbol{\beta} + \beta_0)$$

where $\boldsymbol{x}^T\boldsymbol{\beta} + \beta_0$ is the signed distance from any point $\boldsymbol{x}$ to the hyperplane. However, in cases where the points are not perfectly separable,

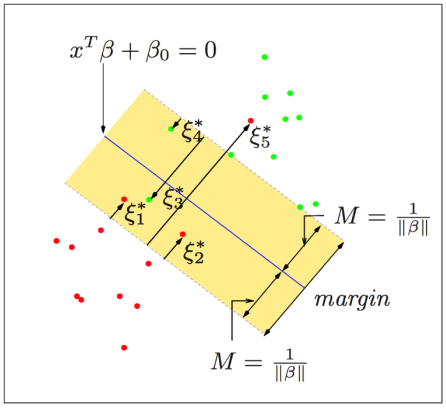

Figure 7: Optimal separating hyperplane in a linearly inseparable case. Figure from Hastie et al. (Hastie et al., 2008).

we could define slack variables, $\xi = (\xi_1, \dots, \xi_N)$, to tolerate misclassifications of some points. This can be seen in Figure 7. ξ_i is defined as the relative distance point $\boldsymbol{x}_i$ is from the margin, depending on the width of the margin. $\xi_i = 0$ for points that lie outside the margin. ξ_i is between 0 and 1 when the point is inside the margin but still correctly classified, and is bigger than 1 when misclassified. We denote the margin to be M, and the optimization problem can be expressed as follows:

$$max_{\boldsymbol{\beta},\beta_0,\|\boldsymbol{\beta}\|=1} \quad M$$

$$s.t.\ y_i({\boldsymbol{x}_i}^T\boldsymbol{\beta} + \beta_0) \geq M(1 - \xi_i), i = 1, \dots, N$$

A constraint on the degree of error tolerance is also introduced:

$$\sum_{i=1}^{N} \xi_i \leq C$$

Solving this optimization problem involves the use of Lagrange multiplier. The main idea is that the Lagrangian can be expressed as a function of the objective function and constraints, and the parameters that lead to an optimal solution are be found by taking partial derivatives of the Lagrangian with respect to the parameters. While the Lagrangian for SVC will not be introduced in detail, one point of interest is that the solution only depends on the dot product of pairs of points ($\boldsymbol{x}_i$ and $\boldsymbol{x}_j$), instead of each individual point, which lowers the computational complexity.

SVM expands this idea by replacing the dot product of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with the inner product $\langle h(\boldsymbol{x}_i), h(\boldsymbol{x}_j)\rangle$ in the Lagrangian, where $h(\boldsymbol{x}_i)$ and $h(\boldsymbol{x}_j)$ are usually non-linear transformations of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, expanding the feature space into higher dimensions. Using the kernel method (Hofmann, Schölkopf, & Smola, n.d.), SVM does the operations in an implicit feature space, without having to express the explicit form of $h(\boldsymbol{x}_i)$ and $h(\boldsymbol{x}_j)$. This method is often computationally cheaper than explicit computation. Common kernels include linear, polynomial, and radial basis.

The reason to include support vector machine is its appealing ability to expand the feature space. By projecting the features into higher dimensions, we hope that the support vector machine may capture more non-linear patterns of the relationship that the linear and logistic regressions ignore. For this research, the kernel for support vector machine is chosen to be radial basis, which is effective when the data generating process follows Gaussian distribution. An important fact to notice is that support vector machines can often leads to perfect classifications of training data and overfitting, if the feature space is expanded to extremely high dimensions. The constraint, C, on ξ solves this problem by allowing some errors. Therefore, tuning C is especially important. For this research, because of the relatively small size of features and observations, C is found through grid search as well.

3.4 Extreme Gradient Boosting (XGBoost)

XGBoost is a gradient boosting method based on classification and regression trees (CART). We introduce the basic concepts of CART and XGBoost here. The basic idea of CART is to partition the feature space into different regions based on split points. For example, in the case of two features, the feature space may be partitioned recursively from R_1 to R_5, as is shown in Figure 8.

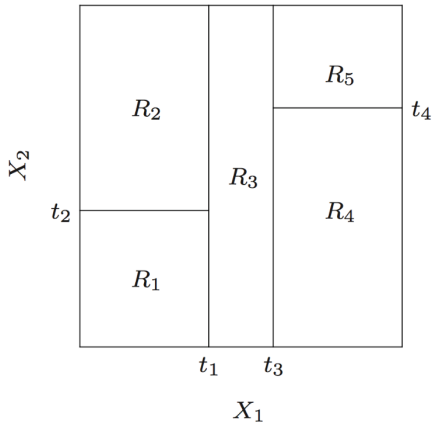

Figure 8: Illustration of regions split in CART. Figure from Hastie et al. (Hastie et al., 2008).

In this case, the feature space is first split at point t_1. The left region is further partitioned into R_1 and R_2, and the right region into R_3, R_4, and R_5. Then each region is assigned with a constant, which will be the prediction value for any observations that fall into that region. Therefore, a CART model can be defined as:

$$\hat{f}(\boldsymbol{x}) = \sum_{m=1}^{M} c_m I(\boldsymbol{x} \in R_m)$$

where c_m is the constant assigned to region R_m, and I is the identity function. The determination of c_m depends on the loss function for each leaf node. For example, for regression trees that uses mean-squared error, $\hat{c}_m = ave(y_i|\boldsymbol{x}_i \in R_m)$, the average of the outcomes of all the observations in that region. Classification trees use other evaluation metrics such as misclassification error instead. The algorithm of CART starts from an empty tree and makes one split in the feature space each time. A split can be defined by the splitting feature j and a split point s, and a pair of regions created by a split can be defined as:

$$R_1(j,s) = \{\boldsymbol{x}|x_j \leq s\}, \qquad R_2(j,s) = \{\boldsymbol{x}|x_j > s\}$$

For regression trees, find the optimal split by finding the j and s that

$$min_{j,s}[min_{c_1} \sum_{x_i \in \mathrm{R}_1(j,s)} (y_i - c_1)^2 + min_{c_2} \sum_{x_i \in \mathrm{R}_2(j,s)} (y_i - c_2)^2]$$

The step is repeated multiple times, and a new node is created after each iteration.

The "best" tree that prevents overfitting is achieved by tree-pruning. The idea of tree pruning is to first grow a tree with many nodes and prune the tree by collapsing certain nodes together. Like the elastic net, some regularization parameter will determine the optimal size of the tree. Regularization is performed with the cost-complexity criterion:

$$C_\alpha(T) = \sum_{m=1}^{|T|} \left(\sum_{x_i \in \mathrm{R}_m} (y_i - \hat{c}_m)^2 \right) + \alpha |T|$$

where $|T|$ is the tree size (node of trees) and $\alpha \geq 0$ is the regularization parameter to be tuned. An optimal value can be found through cross validation, for example.

We now introduce gradient boosting, which is based on an ensemble of CARTs. Gradient bosting is an additive training method that can be illustrated as a series of steps:

$$\hat{y}_i{}^{(0)} = 0$$

$$\hat{y}_i{}^{(1)} = f_1(x_i) = \hat{y}_i{}^{(0)} + f_1(x_i)$$

$$\hat{y}_i{}^{(2)} = f_2(x_i) = \hat{y}_i{}^{(1)} + f_2(x_i)$$

...

$$\hat{y}_i{}^{(t)} = \sum_{m=1}^{t} f_m(x_i) = \hat{y}_i{}^{(t-1)} + f_t(x_i)$$

At each step, a new CART model $f_m(x)$ is added in the ensemble to improve the current model performance. Gradient boosting "greedily" reduces its error by setting the negative gradient of the ensemble's error to be the target of the next model being added. In that way, the ensemble always grows in the direction that reduces its error most effectively. The target of the m-th model, $f_m(x_i)$ is thus:

$$r_{im} = -\left[\frac{\partial L\big(y_i, f_{m-1}(x_i)\big)}{\partial f(x_i)} \right]$$

where L is the loss function. An additional regularization term in be included to prevent overfitting.

This research uses the extreme gradient boosting (XGBoost), which is an algorithm of the gradient boosting family that is optimized to compute extremely fast (Chen & Guestrin, 2016). XGBoost is used as

one of the base models in this research. It has been widely recognized as one of the top solutions for many machine learning competitions. There are many parameters to be tuned for XGBoost, such as the regularization parameters, learning rate, maximum depth of each node, early stopping rounds, percentage of column sample. For example, percentage of column sample allows one to randomize each tree by assigning slightly different subset of features to each tree. This is one of the reasons XGBoost is incorporated as a base model, because the randomizing feature is especially appealing when the feature set is rather large to avoid overfitting. Although XGBoost has many parameters to be tuned, the tuning is still done through a series a grid searches, where each search only investigates a few of the parameters so that the process is not overly time-consuming. However, while the test error is indeed reduced after tuning, no significant improvements in performance are seen so that an exhaustive parameter tuning in this case is not necessarily crucial.

3.5 Ensemble and Stacking

After all the base models produce their predictions, we explore several methods in ensemble learning to obtain better prediction performance. The idea of ensemble learning is that certain combinations of individual machine learning models might outperform the result of each one. Conceptually, ensemble of models will allow weak classifiers to be compensated by stronger ones, at the cost of performing some extra computation.

The simplest ensemble method is a majority vote. As a simple illustration, suppose there are three weak classifiers each with a classification accuracy of 60% on the test set. If a new observation is presented, the probability that at least two classifiers are correct is 64.8%, which is higher than the individual performances. In this research, predictions of all the base models are combined and for each observation, the prediction with most votes becomes the prediction for that observation.

Stacking is used as the second approach, a special ensemble method that is more complex than simple majority vote. Stacking usually involves putting the predictions of the base models as new features into another model. Specifically, the training data is first partitioned into five random folds. In order to make a prediction for each fold, the training data of the other four folds are used to train the base models.

Therefore, after five iterations, all the outcomes of the training period are predicted. In other words, instead of simply using the training set predictions of the base models, we produce a cross-validated training set prediction where the folds are identical for each base model. This is the meta feature for the stacking models. For this research, logistic regression and XGBoost are used as the two stacking models, and they are trained on these meta features. The "meta parameters," or the tuning parameters for stacking models, are optimized to achieve the best test set performance. Ideally, stacking would discover more complex relationships between individual predictions of the base models and the outcome, because an additional model is used to look for that relationship.

4 Results

4.1 Classification Performance

Jinchuan Group International Resources was primarily investigated. Feature selection is performed in three steps: firstly, the features are ranked by univariate R^2 values and the top 150 features are considered; then only the features with regression p-values smaller than 0.2 are selected; at last, these selected features are put into a L1 regularized linear regression for further selection. We also performed stepwise regression for Jinchuan's data, but it always gave a very small number of selected features and decreased classification accuracy. It is therefore not used in the feature selection. 85 features are selected as the final feature set. Moreover, we discover that the stacking models yield poor cross-validation results if only the meta features are used. We hypothesize the reason to be that the stacking models do not have enough features, since there are only four base models. Therefore, five lags of the stock prices themselves are included as autoregressive terms in addition to the meta features for stacking models, and the cross-validation performance is improved. Table 2 shows the performance of the different models.

Table 2: Performances of different models for Jinchuan

	Linear	Logistic	SVM	XGBoost	Vote	Logistic Stacking	XGBoost Stacking
Train	0.125	0.129	0.035	0.000	NA	NA	NA
Test	0.377	0.338	0.354	0.346	0.331	0.323	0.323
Validation	0.344	0.364	0.352	0.375	0.340	0.340	0.340

All the base models perform relatively well in the training set, with XGBoost achieving 0 error rate. However, the same level of performance is not continued in the test set, although the error rate is still decent at around 0.3. The validation performance for the base models is similar to their test performance, which not only indicates that the base models already have sufficient predictive powers, but also suggests that the selected features have long-term influence on Jinchuan's stock prices, because they still show effective signals after a year. One point of interest is the stability of the four base models. We find that linear regression performs best in the validation set, but its validation performance differs relatively greatly from its relatively poor test result, suggesting it to be an unstable model. In contrast, SVM is concluded as the most stable model with favorable and consistent performances in both test and validation.

One majority vote ensemble and two stacking models, logistic regression and XGBoost, are further developed. All three methods slightly improve the performance on the validation set to 0.340. One possible cause for this slight improvement is that the base models all make similar mistakes, therefore they do not correct each other effectively enough in ensemble models. Nevertheless, these performances are still better than any of the base models, effectively proving the power of ensembles.

4.2 Trading Strategy Performance

The performance of the trading strategy on validation set directly reflects the practical value of the models. Because this research uses model ensemble as the final models for trading, only the strategies of majority vote, logistic stacking, and XGBoost stacking are produced and compared. Jinchuan's stock yields a relative return of 166%. For the strategies produced by majority vote, the relative return is 287%. For strategies produced by logistic stacking, the relative return is 287%. For strategies produced by XGBoost stacking,

the relative return is 313%. All three ensembles produce similar strategies and returns, and they outperform Jinchuan's the base return of 166%. Figure 9 is the strategy performance of XGBoost as an illustration.

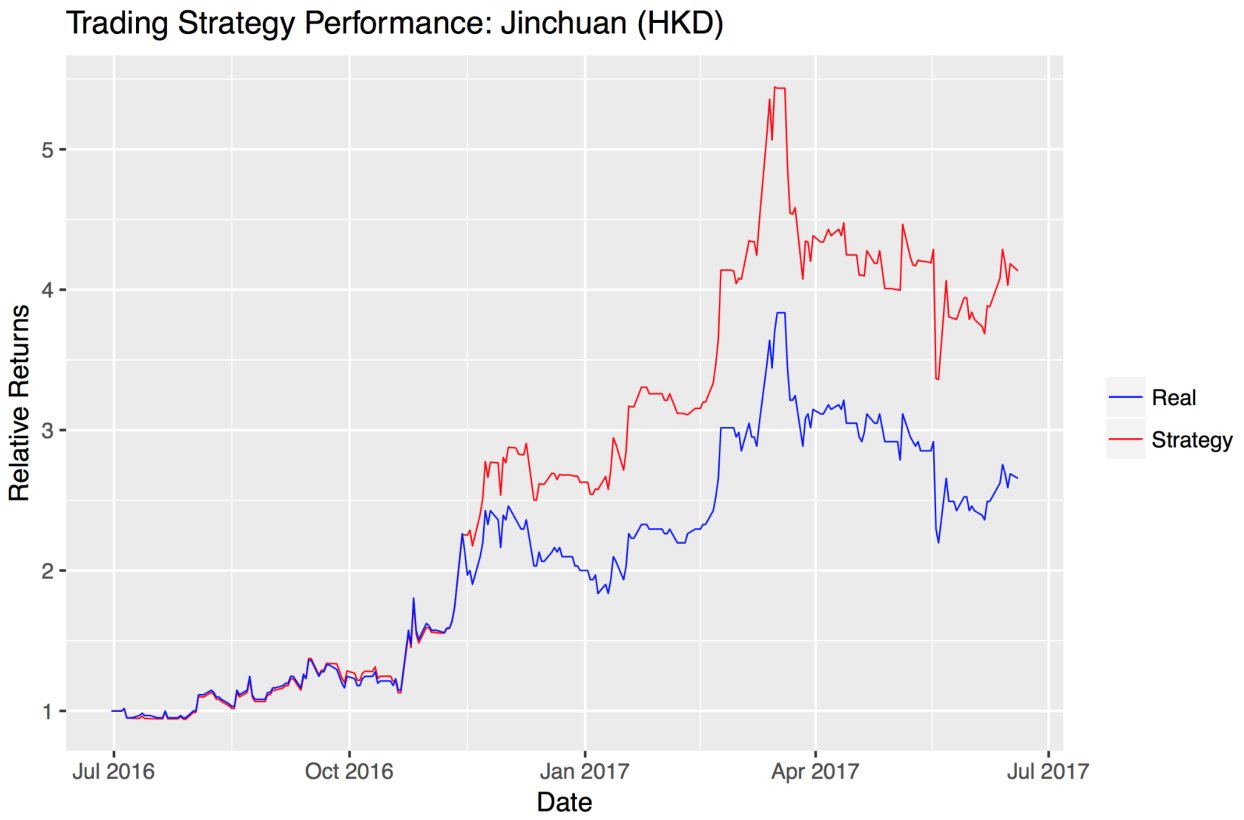

Figure 9: Relative returns of Jinchuan's stock compared to relative returns of the trading strategy produced by XGBoost.

It is important to point out that performance of the trading strategy does not necessarily indicate the effectiveness of a model. Since all the models only classify the states of the stock price into rise or fall, they do not predict the degree of the price change. A model with higher misclassification error may still result in a better return if it happens to correctly classify days which the stock prices rise most significantly.

5 Discussion

5.1 Sensitivity Analysis

To examine the sensitivity of the models to changes in features, we re-run the models under several different scenarios. First, we find that the stock prices stay the same for certain days within our observation window. In the data cleaning process, we labeled such days as rises. However, if we try to remove the days in which the stock prices remain constant and instead only train the model on days which the stock prices change, there are notable changes in the selected features and performances become poor. This is because by removing these days, the consistency of the lag structure is disturbed. The significantly different results suggest that the lag structure is crucial to the models and is quite sensitive to disturbances.

In addition, we discover that whether the outcomes were converted into binary values from continuous values before feature selection yields noticeable changes in the selected features. It is observed that more potent features would be selected if the outcomes are binary values. This makes an intuitive sense because for a classification task, whether an observation is classified correctly is more important than how close the predicted value is to the classification threshold. For example, suppose the real value of an outcome is 0.2, which is greater than 0 and symbolizes a rise in stock price. In this case, the prediction of 0.6 is preferred over -0.1 because 0.6 predicts the rise in price correctly, even if -0.1 is closer to the real value. However, while such minor changes often do not alter results significantly, our models' degree of sensitivity in this scenario is noticeable enough to show that a consistency in data between feature selection and model-fitting is especially important in this research. One possible explanation is that because both the feature selection and some of the base models utilize linear regression, inconsistencies in data render the selected features ineffective for linear-regression-based models.

Moreover, a more interesting observation arises when the number of observations used in feature selection is changed. In order to maintain the objective of testing the predictive abilities of the models, only the training set is used in feature selection. However, just for the sake of investigation, we tried to incorporate data of the test and validation sets in addition to the training set. When given data of the future, producing a different feature set is certainly expected, nor is an increase in the classification accuracy surprising. However, this new feature set differs from the original feature set greatly enough that a point of interest arises. For Jinchuan Group, only 45 out of the 66 features in the original set are also present in the

new set (different lags are considered as one feature in this case). One explanation for this fluctuation in selected features is that the feature selection method is not potent enough to capture stable features. A more likely hypothesis is that part of the features influencing Jinchuan's stock price are themselves highly dynamic and expire to be valid in a few months. Therefore, when new data of the next year (i.e. test and validation sets) is incorporated into feature selection, features that expire to be effective in the next year are replaced, leaving only 45 features that seem to be continuously effective in both years. This dynamic property of the feature selection is certainly worth future investigations and confirmations.

5.2 Proposed methods of future improvement

The first proposed method is based on the idea of on-line updating. If important features in the history do not continue to be effective in the future, adding the model's ability to constantly update the feature set may yield better validation performance. Since the trading frequency is relatively low in this research, the models can be updated to re-select the feature set and re-train themselves automatically on a daily basis. Therefore, after predicting the results of the n-th day, the models would incorporate the new stock price data of the n-th day to its training set, perform feature selection on the new set, train the models again with the updated data, and tune the model parameters automatically.

Moreover, voting and stacking did not seem to improve the performance to a great extent. One possible explanation is that some of these model predictions are too similar so they cannot correct each other's mistakes enough in an ensemble. Indeed, an ideal ensemble would contain base models that are as uncorrelated as possible. Therefore, a proposal to improve model performance is to diversify each model to generate less correlated predictions. When selected features were reviewed, it was discovered that the feature set only contained a few specific categories of data, such as data about metal commodities or data about energy resources. Therefore, if the feature set is divided into subsets according to their different categories, and each feature subset would be trained into a different sub-model, diversity in base models would in theory be achieved. For example, if three feature subsets are created, there will be three different models using linear regression, three using logistic regression, three using SVM, and three using XGBoost. This method also makes practical sense, as each diversified sub-model only investigates connections

between that specific category of data and the outcome. Therefore, ideally, these sub-models would be less correlated. In addition, Sun and Li also suggest diversification through using different kernels in SVM (Sun & Li, 2012). Specifically, separate SVM models can be created using different kernels to further increase the number of base models. Since different kernels expand the dimension of the feature space in varying fashions, it is reasonable to assume that their predictions would be different.

5.3 Investigation of feature stability

We further investigate whether some features have continuing influence on the outcome In order to test if the feature selection procedure is stable with respect to included dates, three feature sets were selected independently to investigate feature stability: one using just the training data, one using both training and test data, and one using all training, test, and validation data. It was discovered while all three sets were all quite different in certain parts, a set of 35 features remain present in all three sets. These features were considered to have continuing influence on the stock price and are informally identified as the "stable features." The stability of the features is compared to their statistical significances. Specifically, the selected features are ranked according to their p-values. Table 3 is a chart of the top 10 features with lowest p-values.

Table 3: Top 10 features with lowest p-values. Italicized features are not stable features.

Feature	p-value
Palladium Price	0.0067
Fijian Dollar	0.0113
Russian Ruble	0.0178
Fuel Oil Spot Price	0.0190
Electricity Base Rate PJM	0.0195
Gasoil FOB Price	0.0218
Gold Coin Price	0.0240
Ethylene FD Price	0.0250
Crude Oil FOB Price	0.0261
Gasoline FOB Price	0.0266

While most of the top ten selected features are also stable features, some of them are not and are italicized in the table. For example, the Gasoil Free On Board Price ranked number 6 in statistical significance by p-value out of the entire feature set, but is not even included in the other feature sets. This suggests that even features that are highly correlated with the outcome in the training set do not guarantee a continuing influence on the outcome, and can still quickly cease to be effective in a few months.

5.4 Investigation of significant features

For the investigation of significant features, all the discussed features in this section are already identified as "stable features" in the previous section. A general look at the makeup of the entire feature set provides a holistic insight. We classify these features into different categories, and Table 4 is a summary of these categories.

Table 4: Summary of categories of selected features.

Chemicals	2
Crude Oil and Products	8
Electricity	8
Exchange Rates	20
Commodity Index	3
Metals	16
Precious Metals	28

Consistent with common sense, data related with metals and precious metals makes up the majority of the feature set, suggesting that it has significant influence to Jinchuan's stock prices in various different ways. For example, the price of palladium has the highest statistical significance by p-value, ranking top in table 3. The fact that palladium is commonly produced as by-products of copper and cobalt, both major products of the Jinchuan Group, might explain this feature's high correlation with Jinchuan's stock price. Moreover, prices and indices of many base metals are also significant features (though not shown in Table 4), such as copper, lead, and zinc, many of which are host metals for critical metals. Precious metals such

as gold also play a part in the feature set. This observation suggests that a strong inter-correlation among many different metal cycles may exist, and is a point of interest for further research.

Moreover, crude oil and other energy-related features constitute a large portion of the feature set. The metal industry is extremely dependent on energy to power massive mining machines and conduct large-scale productions. Therefore, the performance and profit of mining companies may be correlated with trends in the energy sector. While a further investigation into the relationship and causation between metals and energy is valuable, this correlation may be general for any industrial metals, not just specific to critical metals.

Interestingly, we find that many currency exchange rates are also selected. While a recent finding, mentioned previously in this article, show that Chilean Peso has a high correlation with copper prices (David Meyer, 2015), the observation in this research suggests that many other currencies are correlated with metals as well. One possible hypothesis is that since currency exchange rates are a reflection of the country's GDP, if the country's economy relies heavily on the export of certain types of metals, their productions may impact the exchange rates. For example, currency exchange rates between Russian rouble and US dollar have high statistical significance by p-value. Mining is a major industry in Russia. Export of metals contributes a large proportion to Russia's GDP, and in 2013, Russia's total value of output from mining constitutes 14.6% of the GDP (Safirova, 2015). Since Russia is the fourth largest cobalt producing country in 2016 (USGS, 2017), and Jinchuan is also a large cobalt producer, the correlation between Russian ruble and Jinchuan's stock prices is therefore reasonable. In addition, the presence of exchange rates and some commodity indices in the feature set also reveals that factors in macroeconomics play an important role.

The investigation of selected features in this research only serves to provide a rough sense of potentially influential factors as well as a hint and intuition to potential points of interest worthy of future research. Because machine learning methods used in this research avoid human biases, some previously unknown factors that are suggested here may hopefully be validated by future researches. However, until clear understandings of the relationships and detailed analyses of the causations are established, one cannot

know whether these factors are simply correlated by chance. This research does not focus on investigating the detailed implications of such relationships but only aims to offer a possibility and an intuition to future researchers.

5.5 Investigations of other companies

Although Jinchuan was selected as the primary subject of the research, three additional companies were briefly investigated at the last phase of the research in order to confirm that findings of this research are not particular to Jinchuan, but can be generalized to fit other critical metal companies. Specifically, Yunnan Lincang Xinyuan Germanium Industrial Co. Ltd., Huludao Zinc Industry Co., Ltd., and Teck Resources Ltd. were investigated. The features selected for these three companies do not differ significantly from Jinchuan's feature set. While Lincang Xinyuan and Huludao yield performances worse than Jinchuan, the cause may be that they are only local companies in China instead of large international corporations such as Jinchuan. Therefore, they might be less responsive to signals in other industrial sectors and other countries. However, Teck yields especially poor performances and deserves attentions. Table 6 are the results for Teck:

Table 6: Performance of different models for Teck.

	Linear	Logistic	SVM	XGBoost
Train	0.062	0.078	0.012	0
Test	0.531	0.562	0.508	0.508
Validation	0.581	0.597	0.545	0.534

There may be multiple reasons for this bad performance. The extremely low error rate for the training set suggests that there might be some unseen overfitting in all the models. However, since the feature selection method works well for the other three companies, it is puzzling why it would over-fit the fourth. Another possibility is that the time series data of Teck may underwent some sort of structural change during 2015 and 2017, in which a large amount of signals becomes invalid in the test and validation periods.

Nevertheless, the poor results of Teck suggest that investigations on more international mining companies in the future is be valuable.

6 Conclusion

Overall, by achieving a final classification error rate of 0.34 and generating a trading strategy that yields a 147% excess return compared to the stock itself, this research proves that model ensembles indeed improve the prediction performance. Moreover, as proposed above, methods of diversifying the base models can in theory improve model performance even further. In addition, modifying the model to be able to self-update itself on a daily basis may also boost the final return rate.

Besides findings about the models themselves, insights into factors that influence stock prices of a critical metal producer, thus the production of critical metals, are also gained. Interestingly enough, we find that the feature selection is only partially stable. Moreover, some unexpected factors appear to have high correlation with the stock prices, though their clear relationship is yet unknown.

For future works, we believe that improvements of the models can certainly be studied and implemented, focusing on understanding and solving the problem of Teck's poor performance. Moreover, additional investigations into the cause and points of interest behind the fluctuating feature space may also be conducted to better understand the complications of critical metals and their producers.

7 Acknowledgements

The author would like to thank the tremendous support of the research instructor, Xinkai Fu, on this research, who provided valuable suggestions and insights to the author when challenges were encountered. Moreover, the author is grateful for the opportunity Dongrun-Yau Science Awards give to have this research paper be viewed in front of a panel of professional judges.

References

Apodaca, L. E. (2016). 2014 Minerals Yearbook: Sulfur, (May). Retrieved from goo.gl/0RvR5P

Berk, R. A. (2006). An Introduction to Ensemble Methods for Data Analysis. *Sociological Methods & Research*, *34*(3), 263–295. https://doi.org/10.1177/0049124105283119

Bleiwas, D. I. (2010). Byproduct Mineral Commodities Used for the Production of Photovoltaic Cells. *Usgs*, *1365*, 18. Retrieved from http://pubs.usgs.gov/circ/1365/Circ1365.pdf

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*(1), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007

Chen, T., & Guestrin, C. (2016). XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16* (pp. 785–794). New York, New York, USA: ACM Press. https://doi.org/10.1145/2939672.2939785

Choudhry, R., & Garg, K. (2008). A Hybrid Machine Learning System for Stock Market Forecasting. *World Academy of Science, Engineering and Technology*, *2*(15), 315–318.

David Meyer. (2015). The High Correlation between the Chilean Peso and Copper Prices - Market Realist. Retrieved August 28, 2017, from http://marketrealist.com/2015/09/high-correlation-chilean-peso-copper-prices/

Fama, E. F., & French, K. R. (1988). Dividend Yields and Expected Stock Returns. *Journal of Financial Economics*, *22*, 3–25.

Graedel, T. E., Barr, R., Chandler, C., Chase, T., Choi, J., Christoffersen, L., … Zhu, C. (2012). Methodology of metal criticality determination. *Environmental Science and Technology*, *46*(2), 1063–1070. https://doi.org/10.1021/es203534z

Gunn, G. (Ed.). (2014). *Critical Metals Handbook*. American Geophysical Union; John Wiley & Sons, Ltd. https://doi.org/https://doi.org/10.1016/S0065-3233(08)60135-7

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The Elements of Statistical Learning* (2nd ed.).

Hofmann, T., Schölkopf, B., & Smola, A. J. (n.d.). Kernel Methods in Machine Learning. *The Annals of Statistics*, *36*, 1171–1220. https://doi.org/10.2307/25464664

Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, *36*(8), 10896–10904. https://doi.org/10.1016/j.eswa.2009.02.038

Morse, D. E., & Glover, A. N. (2000). *Minerals and Materials in the 20th Century—A Review*.

Nassar, N. T., Barr, R., Browning, M., Diao, Z., Friedlander, E., Harper, E. M., … Graedel, T. E. (2012). Criticality of the Geological Copper Family. *Environmental Science & Technology*, *46*(2), 1071–1078. https://doi.org/10.1021/es203535w

Poterba, J. M., & Summers, L. H. (1988). Mean reversion in stock prices. *Journal of Financial Economics*, *22*, 27–59. https://doi.org/10.1016/0304-405X(88)90021-9

Priestley, M. B. (1983). *Spectral Analysis and Time Series*. *Academic Press*.

Safirova, E. (2015). *2013 Minerals Yearbook: Russia*. *USGS*.

Schöneburg, E. (1990). Stock price prediction using neural networks: A project report. *Neurocomputing*, *2*(1), 17–27.

Sun, J., & Li, H. (2012). Financial distress prediction using support vector machines: Ensemble vs. individual. *Applied Soft Computing Journal*, *12*(8), 2254–2265. https://doi.org/10.1016/j.asoc.2012.03.028

Tsai, C.-F. and Wang, S.-P. (2009). Stock Price Forecasting by Hybrid Machine Learning Techniques. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, *1*. Retrieved from http://www.iaeng.org/publication/IMECS2009/IMECS2009_pp755-760.pdf

USGS. (2017). *Mineral Commodity Summaries - Cobalt*. *Mineral Commodity Summaries*. https://doi.org/http://dx.doi.org/10.3133/70140094.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

本参赛团队声明所提交的论文是在指导老师指导下进行的研究工作和取得的研究成果。尽本团队所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果。若有不实之处，本人愿意承担一切相关责任。

参赛队员： 董正阳　　指导老师： 傅心恺

2017 年 09　月 13　日