Corn Price Prediction with Twitter Dataset: Sentiment Analysis and Corn Price Movement

Jia-Wei Jessie Liang

Abstract

Real-time Twitter data could be used to predict market movements of many financial instruments, and in this paper, I utilize sentiment analysis methods on each related tweets to predict future corn price movements.

Sentiment analysis that combines natural language processing, artificial intelligence, text analysis and computational linguistics allows us to use the content from each tweet on Twitter to make critical decisions in the market. In this paper, Stanford CoreNLP Sentiment is implemented to analyze each tweet's sentiment. The score, generated by sentiment analysis will be used to predict the corn price based on tweets from previous days.

By ways of supervised machine learning techniques, I outline several machine learning pipelines with the objective of identifying corn price movements. In the following sections, this paper will propose regression and classification models, compare their pros and cons, and find out the most fitted model. In addition, ensemble-learning method is utilized to see if there is any probability to build a strong model based on base models.

Lastly, for all the models I build, I calculate each of their annual return of investment (ROI) by using "all buying" or "all selling" strategy based on model's predicted signal -- with the initial fund $10,000. Result shows that multiple linear regression is the best fitted model of all, with a final annual return around 30.09%.

Data Preparation

Original Dataset

Two datasets are related to this problem: Predicting corn price movements with Twitter dataset

- ***Twitter dataset***: includes 25,695 observations (tweets) from 2008 to 2017. Text column contains tweets that are related to corn price.
- ***Corn price dataset***: includes 2,311 observations from 5/2/2008 to 6/30/2017.

The corn data dataset is clean, which can be analyzed directly. However, the Twitter dataset is noisy, and it needs additional cleansing in the text column before applying the sentiment analysis.

Data Cleaning: Tweets Text Mining

First, let's take a look at the twitter dataset (Figure 1):

date	retweets	favorites	text
6/13/2017 19:56	2	0	$ZC_F Daily chart on 6/4: $Corn has now broken above (W) at 387.2 favoring more upside #elliottwavepic.twitter.com/TzzJWR4jez
6/13/2017 18:03	0	1	Agriculture Markets Report - Tusday, June 13 $CORN $SOYB $WEAT $BAL $JO $MOO $SGG $CHOChttps://goo.gl/uCaQzk
6/13/2017 17:59	102	1	$wdrp sad to say but I am happy this is failing. $vdrm $icld $corff $snap $gyst $nseh $heb $exol $aapl $gequ $fb $corn $xiv $vois
6/13/2017 14:39	0	0	$CORN July corn settled $0.04 higher (+1.1%) at $3.81/bushel https://etfdailynews.com/etf-news/corn-july-corn-settled-0-04-higher-1-1-at-3-81bushel/ ...
6/13/2017 14:38	0	0	$CORN July corn settled $0.04 higher (+1.1%) at $3.81/bushel https://stocknews.com/news/corn-july-corn-settled-0-04-higher-1-1-at-3-81bushel/ ...
6/13/2017 14:14	0	0	In 1963, #DeKalb #XL45 was first grown commercially. #CORN #AGTECH http://briandcolwell.com/2017/04/a-giant-sized-history-of-corn/.html ... #agriculture #grains #agri #tech #agronomy $corn
6/13/2017 13:40	1	0	$CORN http://schrts.co/QxaDLf Corn buyers I think are waiting now for weather related event. Inventory has been high for years, priced in.
6/13/2017 12:12	1	1	Between 1994-2001, the price of #corn in #Mexico fell 70%. #NAFTA http://briandcolwell.com/2017/04/a-giant-sized-history-of-corn/.html ... #agriculture #agtech #grains #agri $corn #agronomy

Figure 1. tweet dataset before text mining

The texts are messy, for example, in one of the tweets:

"$wdrp sad to say but I am happy this is failing. $vdrm $icld $corff $snap $gyst $nseh $heb $exol $aapl $gequ$fb $corn $xiv $vois". All the words behind the $ sign are meaningless.

However, there are also tweets as:

"$CORN July corn settled $0.04 higher (+1.1%) at $3.81/bushel https://etfdailynews.com/etf-news/corn-july- corn-settled-0-04-higher-1-1-at-3-81bushel/ ..." The word behind the $ sign is meaningful.

In order to extract the maximum information from every tweets, I analyze all the tweets and find out patterns of useless information. Then I clean up all useless data in tweets by using R (See details in the attachment). Below is the final clean tweet data (Figure 2).

From Figure 2, all the tweets are relatively understandable compared with Figure 1.

	date	retweets	favorites	text
1	6/13/2017 19:56	2	0	Daily chart on 6/4: has now broken above W at 387.2 favoring more upside
2	6/13/2017 18:03	0	1	Agriculture Markets Report - Tusday, June 13
3	6/13/2017 17:59	102	1	sad to say but I am happy this is failing.
4	6/13/2017 14:39	0	0	July corn settled higher 1.1 at
5	6/13/2017 14:38	0	0	July corn settled higher 1.1 at
6	6/13/2017 14:14	0	0	In 1963, was first grown commercially.
7	6/13/2017 13:40	1	0	Corn buyers I think are waiting now for weather related event. Inventory has been high for years, priced in.
8	6/13/2017 12:12	1	1	Between 1994-2001, the price of in fell 70.
9	6/13/2017 11:52	0	1	WASDE Does Not Step On The Rebound In Grains Too Hard

Figure 2. tweet dataset after text mining

Methodology

Introduction on Sentiment Analysis: Use Stanford CoreNLP to do sentiment analysis.

Stanford CoreNLP is a powerful toolkit, which can apply a bunch of linguistics analysis tools to any texts. It provides available APIs for many modern programming languages such as Java, R and Python. In this paper, I utilize Stanford CoreNLP to calculate the "sentiment score" for each tweet that relates to corn price. The sentiment score shows a person's attitude on corn price. 0 and 1 means negative, 2 means neutral, 3 and 4 means positive.

Below are some pros using Stanford CoreNLP:

- It's not necessary in need to understand the complex NLP methods behind the package.
- Using API for python, one can easily get the sentiment scores for each tweet.
- Stanford CoreNLP supports 7 different languages which allow users to analyze other languages besides English.

There also exists cons:

- Some of the score may not represent a person's attitude correctly. For example, some of the tweets may be neutral while Stanford CoreNLP gave them a negative score.

Overall, Stanford CoreNLP is an efficient and user-friendly tool for NLP analysis.

Data Manipulation

1. Combine tweets and corn price datasets together

From the two datasets, it shows one corn price data but several tweets data on each individual day. Therefore, *mean value* of sentiment score will be utilized in this paper as the final score for each day. After combining the two datasets, a new dataset is created, which contains 2,310 observations.

By taking 30% data from the new dataset as *test dataset*, there will be 693 observations of test set and 1,617 observations as training set. However, tweets in 2008 and 2009 are very sparse, the reason might be that Twitter started in 2006 so in its initial years, there are not many tweets regarding corn price. It is not a correct method to fill in all the missing values with mode or mean since large numbers of "same value" will lead to huge error when doing modeling. Therefore, this paper will not take all the observations before 2010 into consideration for better accuracy.

As for tweets data after 2010, there are still some missing values in the dataset. To avoid the trouble of missing values, I use the method of filling missing values with the value of the day before. For example, if 1/2/2010 is missing, it will be filled with the value from 1/1/2010. This is reasonable because a person's sentiment today is highly dependent on their sentiment yesterday.

2. Decide the variables

As mentioned before, since the price today is relevant to the price from yesterday, I choose the price of the day before as a variable. Also, it is crucial to remember that the tweets several days ago may have impact on today's price too; so, I create another variable to calculate the average value of sentiment score for the past five days.

Since there is need to create a signal of up and down when dealing with classification problems, I also generate a categorical variable called "signal" based on the change of price. If the price increases, signal would be 1, which represents "up". Otherwise signal would be 0, which represents neutral.

The final dataset contains 6 variables and 1,633 observations. (940 as training dataset, and 693 as test dataset)

Date	The date of close corn price
Score	Sentiment score for each day based on tweets
Settle_1	The corn price of previous day (the day before)
Avg_sen5	Average sentiment score of past 5 days
Signal	Represent the price up, or down
Settle	Corn price

Data Modeling

Since I have variables representing past corn prices, the model in this paper will predict next day's corn price movements with these information. When getting the value of future corn price, the model will then includes the new value (next day's corn price) into the training set and build a new model to predict prices for the day after tomorrow.

1. Linear regression

1.1 Multiple linear regression model:

$$Settle = w_0 + w_1 settle\_1 + w_2 score + w_3 Avg\_sen5$$

In order to estimate the coefficients of linear regression, there is need to define a cost function and minimize it.

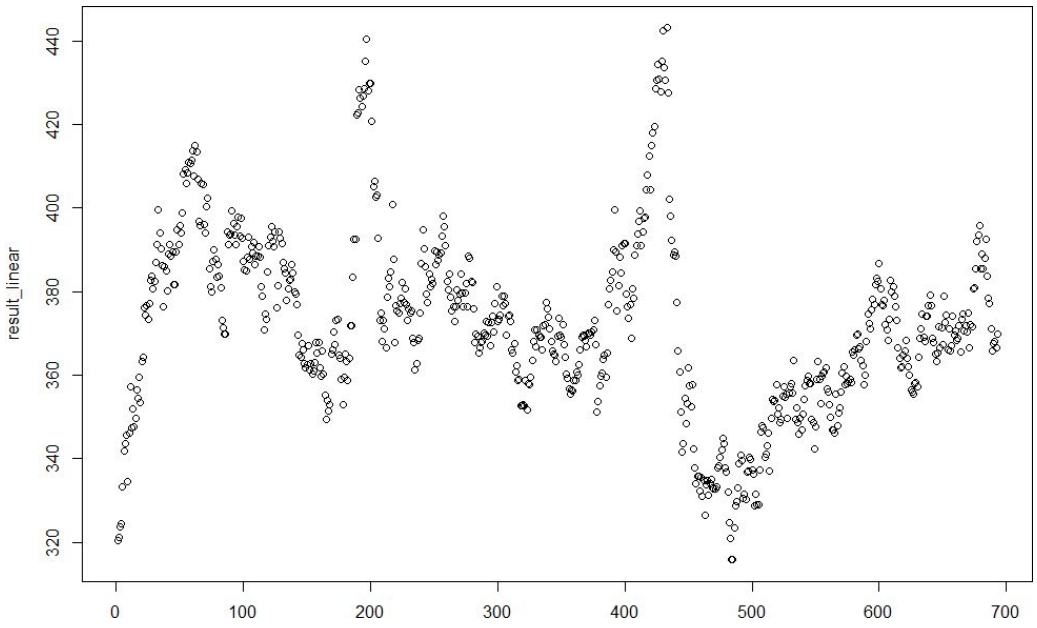
$$Min\ f(w) = \frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - H(w, x^{(i)}))^2$$

Linear regression can be solved by using analytic solutions, but the complexity is too high. Base on my research, most people use optimization approach such as gradient descent to solve this problem. Since there are 693 values that need to be predicted, I build 693 linear regression models. Below, I choose one of these models as an example:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.704584   4.166686  -0.889    0.374
settle_1     0.997916   0.001574 633.883   <2e-16 ***
score        5.719857   2.247701   2.545    0.011 *
avg_sen5    -1.426301   4.495659  -0.317    0.751
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.45 on 1628 degrees of freedom
Multiple R-squared:  0.996,     Adjusted R-squared:  0.996
F-statistic: 1.364e+05 on 3 and 1628 DF,  p-value: < 2.2e-16
```

From the output shown above, the p-value of *settle_1* and *score* indicates that they are significant to *settle*. Also, with a 0.996 R-square value, it means data fit the model very well. The picture below shows the results of predict corn prices for the next 693 days.



The best way to test the quality of model is to calculate the accuracy of prediction. However, for the linear regression model, one cannot calculate the AUPRC[1]. One can only use RMSE[2] as a metric to measure the quality of model. In this case, RMSE for multiple linear regression is 5.436.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_t - y_t)^2}{n}}$$

[1] AUPRC (Area Under the Precision Recall Curve): Functions to compare the area under the precision recall curve (AUPRC) and the area under the F-score recall curve (AUFRC)

[2] RMSE (Root Mean Square Error) A frequently used measure of the differences between values predicted by a model or an estimator and the value observed.

Imagine in a scenario with $10,000 initial money, what will be the annual return? In this case, I will use the predication price as a reference. If prediction price is larger than today's price, then I will use all the money buying corn. Vice versa, if I find the prediction price lower than today's price, I will sell all the corn, and I will then use all the money that I got from selling the corn to buy corn in the future based on the next prediction price.

After the calculation, I will get $16,017 after 693 days. So, the annual return would be (16,017-10,000)/2/10,000=30.08%

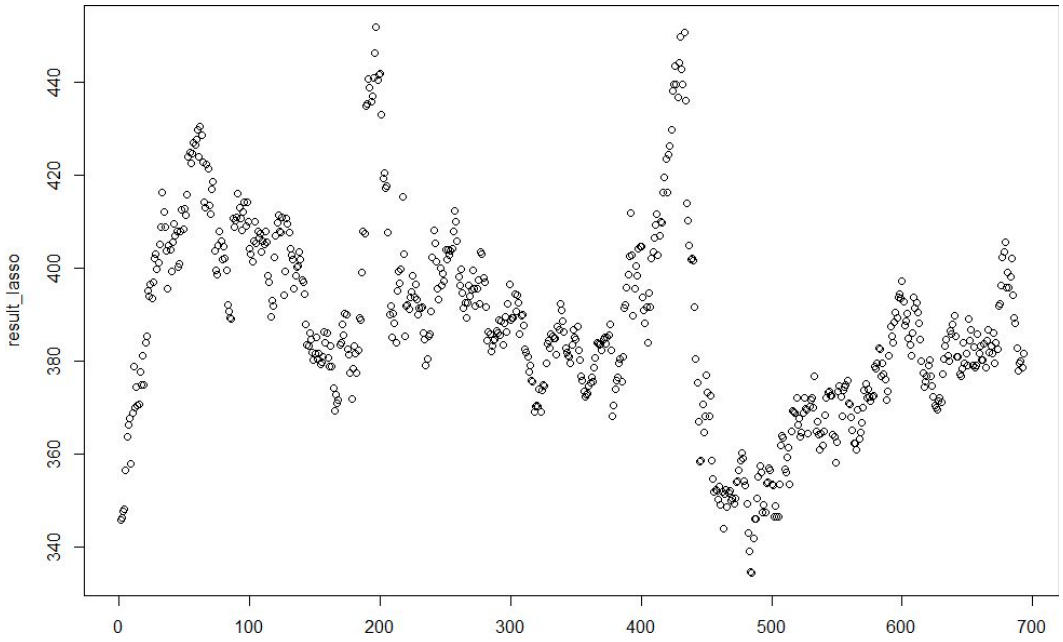
1.2 Lasso regression model:

$$Settle = w_0 + w_1 settle\_1 + w_2 score + w_3 Avg\_sen5$$

The model of lasso regression is the same as multiple linear regression, but the cost function is different:

$$Min\, f(w) = \frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - H(w, x^{(i)}))^2 + \frac{\lambda}{2}||w||_1$$

It adds regularization after the OLS cost function, which solves the problem of overfitting. The picture below shows the results of prediction on lasso regression.



The graph looks almost the same as multiple linear regression. However, when calculating the RMSE of Lasso regression, RMSE for Lasso is 16.516 larger than multiple linear regression.

(Formula is the same as multiple linear regression)

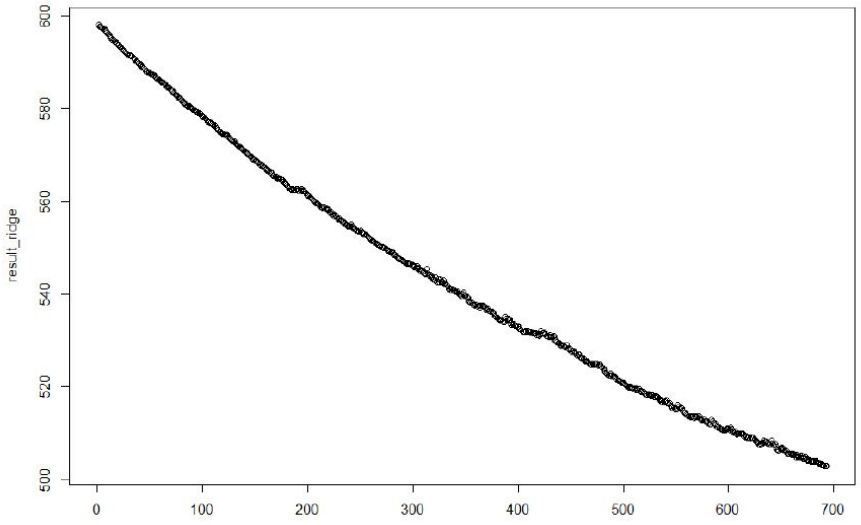
Back to the annual return problem. After 693 days using lasso regression, I will get $11,878. The annual return is 9.4%.

1.3 Ridge regression model:

$$Settle = w_0 + w_1 settle\_1 + w_2 score + w_3 Avg\_sen5$$

$$Min\ f(w) = \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - H(w, x^{(i)}))^2 + \frac{\lambda}{2}||w||_2^2$$

The only difference between ridge regression and lasso regression is also the regularization term. But in this project ridge regression may not be suitable. The picture below shows the result of prediction of ridge regression



Prediction on ridge regression indicates a negative relationship between time and price which is not reasonable, and RMSE for ridge is 173.505 which is much higher than multiple linear regression and lasso regression. Therefore, this paper will not take ridge regression model into consideration.

2. Classification

2.1 Adaline

Adaline is used to solve supervised binary classification problem. In the dataset, I generate a categorical variable called "*Signal*" based on the movements of price. If the price increases, the signal would be "1", which represents "up". Otherwise the signal would be "0", which represents "neutral". I use R to compute the model (see details in attachment) which is as follows:

$$g(w) = w_0 + w_1 settle\_1 + w_2 score + w_3 Avg\_sen5$$

The Binary classification output is determined by a Quantizer (O):

$$Signal = O(g(w)) = \begin{cases} 1, if\ g(w) > 0 \\ 0, otherwise \end{cases}$$

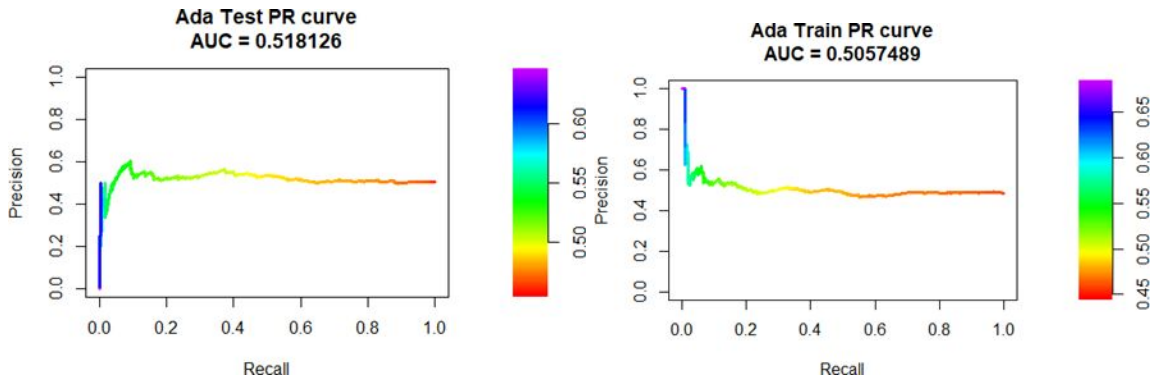
To formulate Adaline, I minimize Sum of Squared Errors (SSE):

$$Min\ f(w) = \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - w^T x^{(i)})^2$$

Adaline can also be optimized by using gradient descent. Since I want to reduce the complexity, I use stochastic gradient descent, such as mini-batch gradient descent that use a subset of samples at a time ($N_m \ll N$):

$$p(k)_j = -\frac{\partial f}{\partial w_j} = \sum_{i=1}^{N}(y^{(i)} - w^T x^{(i)})x_j^{(i)} \approx \sum_{i=1}^{N_m}(y^{(i)} - w^T x^{(i)})x_j^{(i)}$$

Pictures below show the PR curve of Adaline which we build based on the corn price dataset.



AUC represents the areas under the curve, the higher AUC, the higher quality of the classification model. The picture above shows that AUC for training and testing datasets are all around 0.5. This may be a metric for the future model selection.

2.2 Perceptron

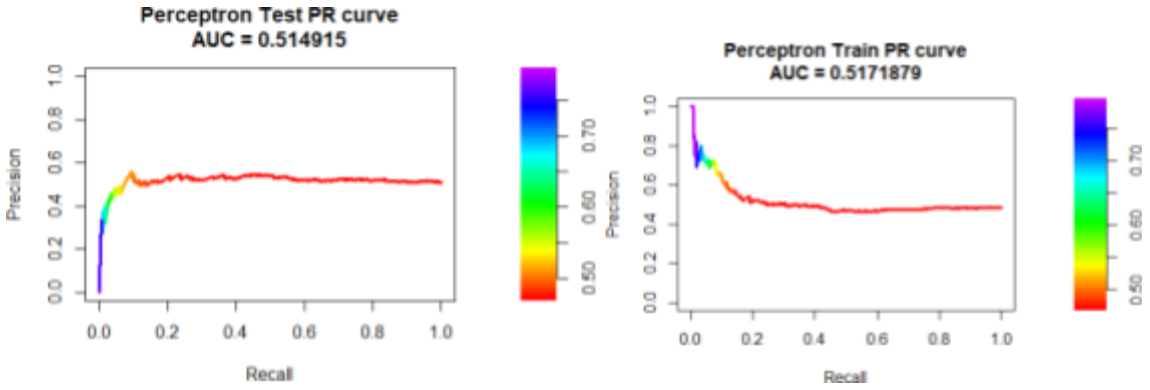
Perceptron is quite similar as Adaline, which is a supervised binary classification problem. The only difference is objective Sum of Squared Errors (SSE), which is captured after activation.

$$f(w) = \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - O(w^T x^{(i)}))^2$$

This model is essentially optimized via stochastic gradient descent, and uses the same method to update parameters:

$$w(k+1) = w(k) + \alpha(k) \times p(k)$$

The pros of Perceptron is that Perceptron uses O(wx) in the cost function which is more reasonable than Adaline.



The PR curve shows that AUC in Training dataset is better than Adaline, but the performance in test dataset is not good. As a trader, I may care more about the performance in test dataset.

2.3 Logistic Regression

Logistic regression can be treated as Adaline with a new activation function and new objective function. The new activation function is:

$$z = w_0 + w_1 settle\_1 + w_2 score + w_3 Avg\_sen5$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$Signal = O(g(w)) = \begin{cases} 1, if\ g(w) > 0 \\ 0, otherwise \end{cases}$$

The new objection function is :

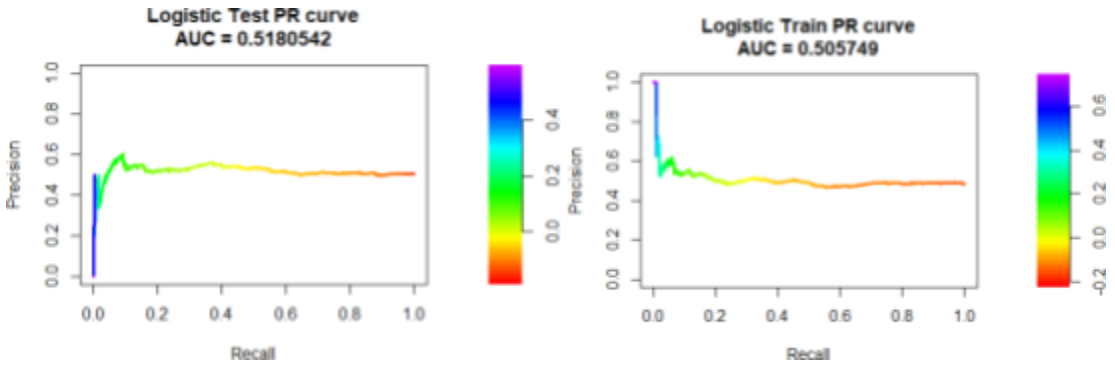
$$Max\ L(w) = \prod_{i=1}^{N} (g(w^T x^{(i)}))^{y^{(i)}} (1 - g(w^T x^{(i)}))^{1-y^{(i)}}$$

Which can be converted into a minimization problem:

$$Max\ f(w) = \sum_{i=1}^{N} [-y^{(i)} \log (g(w^T x^{(i)})) - (1-y^{(i)}) \log (1 - g(w^T x^{(i)}))]$$

I can use the same Adaline algorithm to solve logistic regression except changing the activation function.

Logistic regression is the most widely used methods to do the binary classification. However, it seems that Logistic regression doesn't fit well with the corn price dataset from the below picture.



2.4 SVM

SVM is a linear classifier that outputs an optimal hyperplane for classification for a linearly separable dataset. Because since the dataset is not linear separable, so the SVM needs to be extended to use the "soft-margin". The hyperplane is defined as:

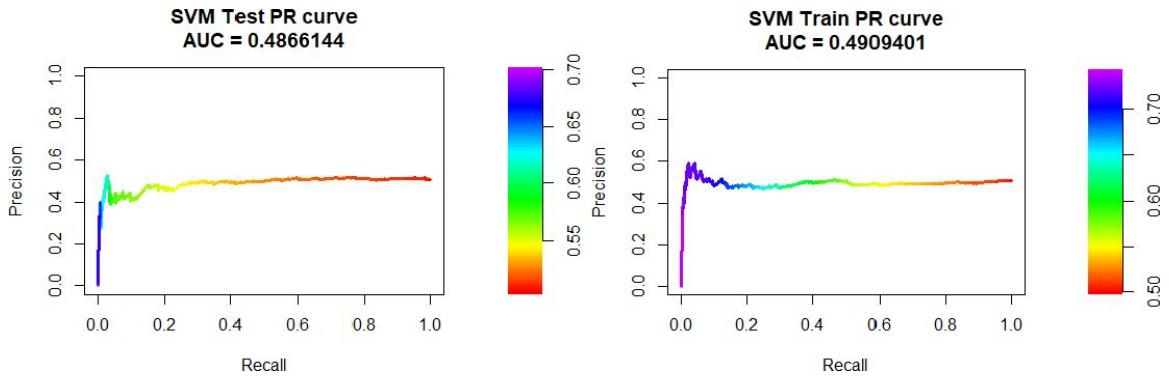
$$w^T x^{(i)} + b$$

In this paper, I would introduce a positive slack variable for each data to model how "wrong" we allow the prediction to be:

$$min_{w,b,\xi}: f(w) = \frac{1}{2}\|w\|_2^2 + C\sum_i \xi_i$$

(for all data sample $x(i)$, $\xi i \geq 0$)

It is convex quadratic programming (QP) problem, and there are many techniques have been developed to perfectly solve QP with global minimum. In order to compare SVM with other classification methods, I also plot the PR curve to calculate the AUC.



Although SVM is considered to be a powerful tool to solve classification problems, it does not fit well in the corn price dataset.

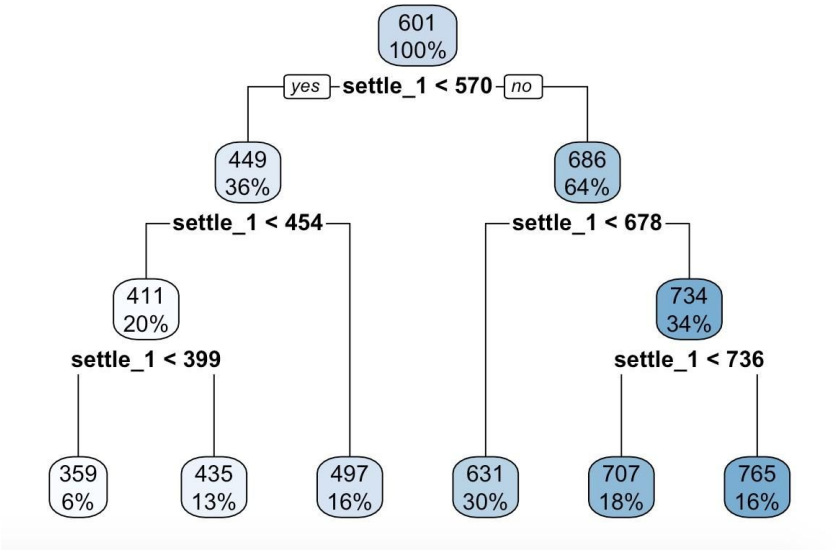
2.5 Decision Tree

Decision tree is a decision support tool that uses a tree-like graph to represent the decision rules (if-then rules) and their possible consequences. There are three key elements of decision tree. First, a metric to measure the improvement of split quality. The metric we used is Entropy as below.

$$\Delta I(P) = \frac{1}{n_P}\left(n_P I(P) - \sum_{j=1}^{m} n_{C_j} I(C_j)\right) = I(P) - \sum_{j=1}^{m} \frac{n_{C_j}}{n_P} I(C_j)$$

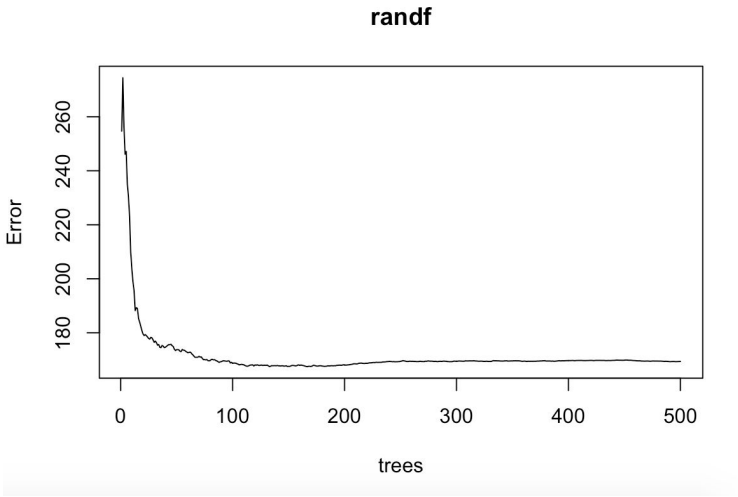
$$I(P) = -\sum_{j=1}^{t} p_j log_2 p_j$$

Second, a stopping criterion. Last but not least, ways to prevent overfitting. The tree we got has 3 nodes. It is not a very good model because sentiment score is even not included, The split is only based on the yesterday's corn price. The RMSE is 20.37 and the annual profit is 12140.78.



2.6 Random Forest

Random Forest is a bagging algorithm, which its model is based on decision tree. It randomly draws sunset features for training each base model decision tree. Since each base model is a decision tree, it is a high-level algorithm. The model we got has 500 trees. The more trees we have, the lower the error is. The RMSE of random forest is 6.52, and the profit we would get is $14,062.03. Therefore, it has much better performance than decision tree model.



2.7 XGBoost

XGBoost is short for "Extreme Gradient Boosting". It is used for supervised learning problems, where I use the training data to predict a target variable. I use XGBoost r package to get this model. However, the performance is not very good. The RMSE is 182.26 and the profit is 11878.41. So, this model should not be taken into consideration.

Conclusion

After exploring all the models listed above, there is need to find out the best model. Imagine if I am a trader, what I care about most is the performance for test dataset and the annual return. From the analysis above, it shows that multiple linear regression has the highest annual return. Random forest and Adaline also have high annual returns; however, when looking at the AUC of PR curve for classification models, both models have the value around 0.51, which indicates bad performances. Therefore, I will not choose Adaline as the best model. Additionally, multiple linear regression has a RMSE of 5.44, which is smaller than random forest.

However, the most accuracy model may not be the best model, we still need to consider the complexity of its explanation. Multiple linear regression is easy to explain because it is easy to get the formula for each model; while random forest is difficult to explain. All in all, based on these considerations, multiple linear regression is the best model for predicting future corn price. The final annual return will be around 30.09% with a $10,000 investment.

Appendix

- Use R to remove all patterns that might lead to meaningless tweets:

28 lines (26 sloc) | 1.22 KB

```r
library(stringr)

data <- read.csv("tweets_data.csv")
tweets <- data[,4]
tweet <- str_replace_all(tweets, "http(s)?://[^[:blank:]]+", "")
tweetnew <- str_replace_all(tweet, "#[^[:blank:]]+", "")
tweetnew <- str_replace_all(tweetnew, "[$][^[:blank:]]+", "")
tweetnew <- str_replace_all(tweetnew, "[^[:blank:]]+\\.com[^[:blank:]]+", "")
tweetnew <- str_replace_all(tweetnew, ">+", "")
tweetnew <- str_replace_all(tweetnew, "for.+[Hh]elp", "")
tweetnew <- str_replace_all(tweetnew, "for.+&", "")
tweetnew <- str_replace_all(tweetnew, "for$", "")
tweetnew <- str_replace_all(tweetnew, "^[:blank:]+", "")
tweetnew <- str_replace_all(tweetnew, "…", "")
tweetnew <- str_replace_all(tweetnew, "\\(", "")
tweetnew <- str_replace_all(tweetnew, "\\)", "")
tweetnew <- str_replace_all(tweetnew, "@", "")
tweetnew <- str_replace_all(tweetnew, "%", "")
tweetnew <- str_replace_all(tweetnew, "\\+", "")
tweetnew <- str_replace_all(tweetnew, "-", "")
tweetnew <- str_replace_all(tweetnew, "´", "")
tweetnew <- str_replace_all(tweetnew, "\"", "")
tweetnew <- str_replace_all(tweetnew, "—", "")
tweetnew <- str_replace_all(tweetnew, "\\*", "")
tweetnew <- str_replace_all(tweetnew, "[:blank:][:blank:]+", " ")
data[,4] <- tweetnew
write.csv(data, "newtweet.csv")
```

Reference

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60

https://stackoverflow.com/questions/32879532/stanford-nlp-for-python

https://en.wikipedia.org/wiki/Logistic_regression http://www.statmethods.net/advstats/glm.html

https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf

https://www.quora.com/What-is-Precision-Recall-PR-curve