Abstract

One of the quintessential data analysis problems of our time involves predicting stock market values over time. The analysis is extremely difficult because of the boundless variables that contribute to stock price fluctuation. We propose a new approach to this classic problem in which Eastern markets of Japan and Hong Kong are used to predict U.S. markets. Since these Eastern markets close before the U.S. markets open, the same-day data can be used as predictors, greatly reducing the dimensionality of the analysis and allowing a classification-based model to be developed. We explore several classification models and discuss results that include two models which can predict stock prices with a 59.0% accuracy.

Objectives

For many years, many financial professionals have attempted to predict changes in the stock market in order to maximize profits trading. These predictions typically involve time series analysis and only focus on the stock market for which the predictions are being made.

Our objective was to use Orange Canvas in order to obtain an accurate prediction of U.S. stock market price (specifically the S&P 500) using data that included Eastern markets. Since Eastern markets close before the open of U.S. markets, we elected to use classification and regression methods instead of time series analysis. We did this, not only because we had data points that would facilitate it, but also because it offered us an opportunity to attempt a novel approach to a classic problem.

Data Set Description

** INCLUDE GENERAL INFORMATION ABOUT STOCK MARKETS AND THE INDEXES WE USED (S&P500, NIKKEI 225, HKEX) **

QUESTIONS WHICH SHOULD COVER
-why chose this measure, what it gives
-in the end of each chapter tell what will be in the next chapter
-make reference to Table, Figure, Pictures
-in conclusion tell what we have learned from this project work to help understand course
-use IEEE format

We will apply following methods on data from 2000 to 2017 of The Nikkei 225, The Standard & Poor's 500 Index (S&P 500), Hong Kong Stock stock exchanges.These data mining methods are:

1. Neural Network
2. Tree
3. Random Forest
4. SVM
5.AdaBoost
6.kNN

-**Neural Network** widget uses sklearn’s Multi-layer Perceptron algorithm that can learn non-linear models as well as linear.)
-**Tree**(**Tree** is a simple algorithm that splits the data into nodes by class purity. It is a precursor to Random Forest. Tree in Orange is designed in-house and can handle both discrete and continuous data sets.)
-**Random Forest** is an ensemble learning method used for classification, regression and other tasks. It was first proposed by Tin Kam Ho and further developed by Leo Breiman (Breiman, 2001) and Adele Cutler. **Random**

Forest builds a set of decision trees. Each tree is developed from a bootstrap sample from the training data. When developing individual trees, an arbitrary subset of attributes is drawn (hence the term “Random”), from which the best attribute for the split is selected. The final model is based on the majority vote from individually developed trees in the forest).

-**SVM**(Support vector machine) is a machine learning technique that separates the attribute space with a hyperplane, thus maximizing the margin between the instances of different classes or class values. The technique often yields supreme predictive performance results.

-**AdaBoost** widget is a machine-learning algorithm. It can be used with other learning algorithms to boost their performance. It does so by tweaking the weak learners.)
-**kNN** (The **kNN** widget uses the kNN algorithm that searches for k closest training examples in feature space and uses their average as prediction.)[1]

We used three main data sets for our analysis, all of which were all obtained from Yahoo Finance. These data sets included financial data for the S&P 500 (sp500_), Nikkei 225 (n_) , and Hong Kong Exchanges and Clearing Limited (hkex_).

The Nikkei 225 is a stock market index for the Tokyo Stock Exchange (TSE). It is a price-weighted index (the unit is yen), and the components are reviewed once a year. Currently, the Nikkei is the most widely quoted average of Japanese equities.

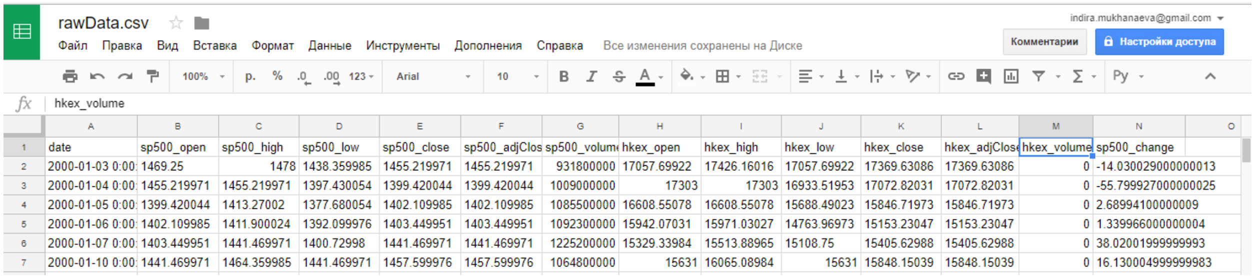
The Standard & Poor's 500 Index (S&P 500) - is an index of 500 stocks seen as a leading indicator of U.S. equities and a reflection of the performance of the large cap universe, made up of companies selected by economists. The S&P 500 is a market value weighted index and one of the common benchmarks for the U.S. stock market; other S&P indexes include small cap companies with market capitalization between $300 million and $2 billion, and an index of mid cap companies. Investment products based on the S&P 500 include index funds and exchange-traded funds are available to investors.

Hong Kong Stock Exchange is one of the world's largest securities markets by market capitalization, the Hong Kong Stock Exchange traces its origins to the founding of China's first formal securities market, the Association of Stockbrokers in Hong Kong, in 1891. A second market opened in 1921, and in 1947 the two merged to form the Hong Kong Stock Exchange[2].

Data Collection and Preparation

The dataset was stored in a table 1. There are 14 attributes. Each instance of data represented a single day and included the date, open, close, high, low, adjusted close, and volume.

Table 1.

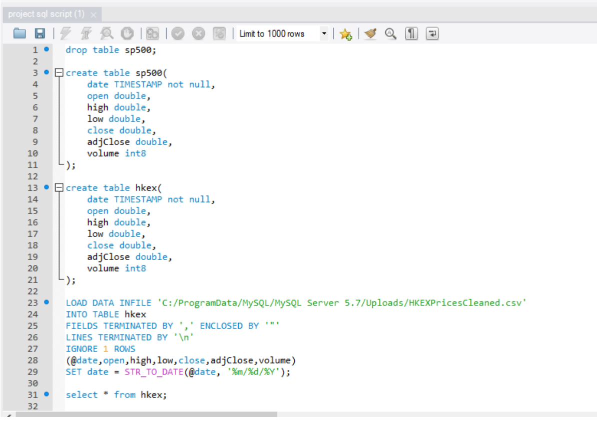


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	date	sp500_open	sp500_high	sp500_low	sp500_close	sp500_adjClos	sp500_volum	hkex_open	hkex_high	hkex_low	hkex_close	hkex_adjClos	hkex_volume	sp500_change	
2	2000-01-03 0:00:	1469.25	1478	1438.359985	1455.219971	1455.219971	931800000	17057.69922	17426.16016	17057.69922	17369.63086	17369.63086	0	-14.030029000000013	
3	2000-01-04 0:00:	1455.219971	1455.219971	1397.430054	1399.420044	1399.420044	1009000000	17303	17303	16933.51953	17072.82031	17072.82031	0	-55.799927000000025	
4	2000-01-05 0:00:	1399.420044	1413.27002	1377.680054	1402.109985	1402.109985	1085500000	16608.55078	16608.55078	15688.49023	15846.71973	15846.71973	0	2.68994100000009	
5	2000-01-06 0:00:	1402.109985	1411.900024	1392.099976	1403.449951	1403.449951	1092300000	15942.07031	15971.03027	14763.96973	15153.23047	15153.23047	0	1.339966000000004	
6	2000-01-07 0:00:	1403.449951	1441.469971	1400.72998	1441.469971	1441.469971	1225200000	15329.33984	15513.88965	15108.75	15405.62988	15405.62988	0	38.02001999999993	
7	2000-01-10 0:00:	1441.469971	1464.359985	1441.469971	1457.599976	1457.599976	1064800000	15631	16065.08984	15631	15848.15039	15848.15039	0	16.130004999999983	

Data Pre-Processing

Data shown in Figure 1 was edited in the MySQL Workbench graphical tool. The *Data Modeling* of MySQL enables to edit all aspects of your database using the comprehensive Table Editor. The Table Editor provides easy-to-use facilities for editing Tables, Columns, Indexes, Triggers, Partitioning, Options, Inserts and Privileges, Routines and Views.

Figure 1



```sql
drop table sp500;

create table sp500(
    date TIMESTAMP not null,
    open double,
    high double,
    low double,
    close double,
    adjClose double,
    volume int8
);

create table hkex(
    date TIMESTAMP not null,
    open double,
    high double,
    low double,
    close double,
    adjClose double,
    volume int8
);

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 5.7/Uploads/HKEXPricesCleaned.csv'
INTO TABLE hkex
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(@date,open,high,low,close,adjClose,volume)
SET date = STR_TO_DATE(@date, '%m/%d/%Y');

select * from hkex;

```

Then table (Figure 2.) can be read in the Orange canvas directly.

Screenshot of Table in Orange canvas

Figure 2. Preprocessing Data

In the figure 2 given attributes are in the numeric format, S&P500 and HKEH are given in the nominal format.

Each of these data sets included end-of-day information about the respective index. Each instance of data represented a single day and included the date, open, close, high, low, adjusted close, and volume. We combined these data sets into our original data set which included 15 attributes: date (continuous), sp500_open (continuous), sp500_close (continuous), hkex_open (continuous), hkex_high (continuous), hkex_low (continuous), hkex_close (continuous), hkex_adjClose (continuous), hkex_volume (continuous), n_open (continuous), n_high (continuous), n_low (continuous), n_close (continuous), n_adjClose (continuous), n_volume (continuous). In order to do this, we had to build 3 SQL tables and join them together on the date. It is shown on the Figure 1. This was necessary since there were some differences in which dates were reported (e.g. 02/11/2001 was not included in the Nikkei 225 report, but was included in the S&P 500.) After getting our raw data set, we needed to transform some of the values into more useful features. We obtained a “change” value for each of the indexes by calculating the difference between the open and close prices. The change value for the S&P 500 became our target feature and the S&P 500 close price was discarded as it would give us more information than we would expect to have when making a prediction. Our resulting data set included 17 attributes- the ones described above along with sp500_change_raw (continuous target), hkex_change (continuous), n_change (continuous).

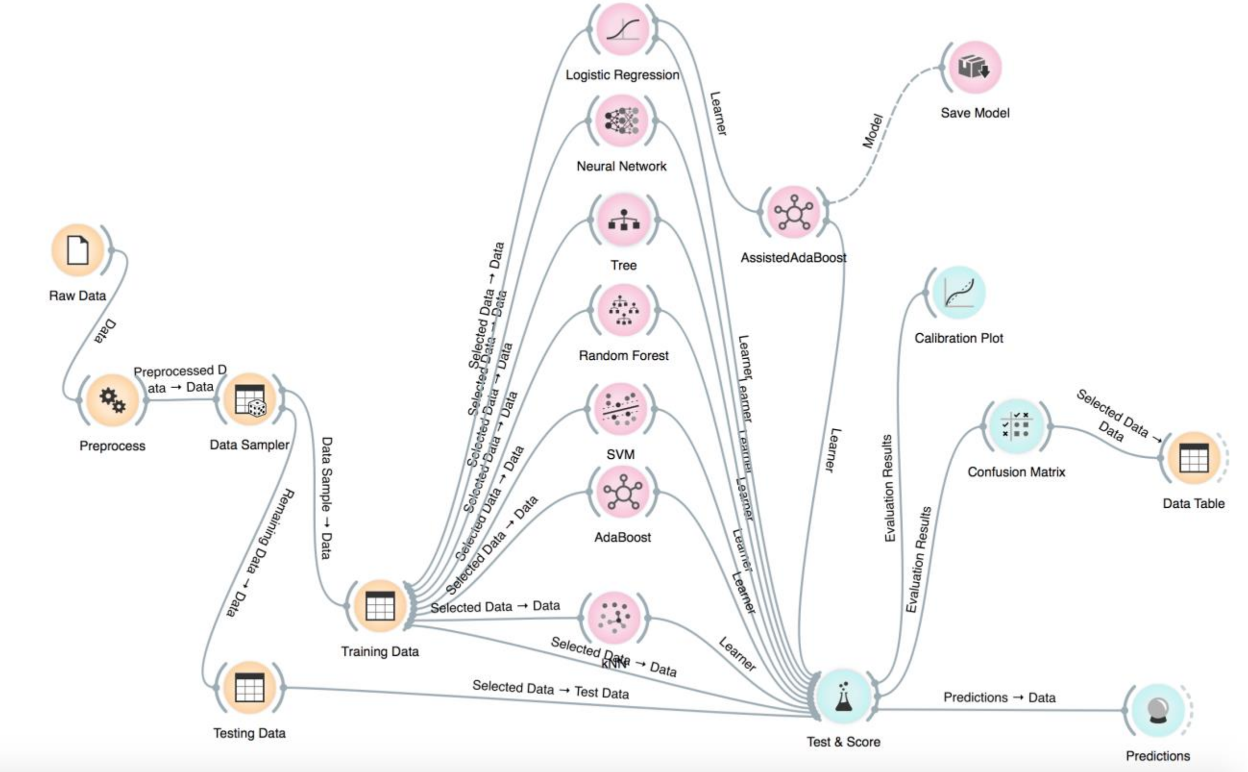
Attributes	Description
Open price	The price at which a security first trades upon the opening of an exchange on a given trading day; for example, the New York Stock Exchange opens at precisely 9:30 a.m. Eastern. The price of the first trade for any listed stock is its daily opening price. A security's opening price is an important marker for that day's trading activity, especially for those interested in measuring short-term results such as day traders.
Closing price	The final price at which a security is traded on a given trading day. The closing price represents the most up-to-date valuation of a security until trading commences again on the next trading day. Most financial instruments are traded after hours (although with markedly smaller volume and liquidity levels), so the closing price of a security may not match its after-hours price.
Adjusted Closing price	A stock's closing price on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open. The adjusted closing price is often used when examining historical returns or performing a detailed analysis on historical returns.
Volume	The number of shares or contracts traded in a security or an entire market during a given period of time. For every buyer, there is a seller, and each transaction contributes to the count of total volume. That is, when buyers and sellers agree to make a transaction at a certain price, it is considered one transaction. If only five transactions occur in a day, the volume for the day is five.
Change	The difference between the current price and the last trade of the previous day. For interest rates, change is benchmarked against a major market rate and may only be updated once a quarter.

Raw Data

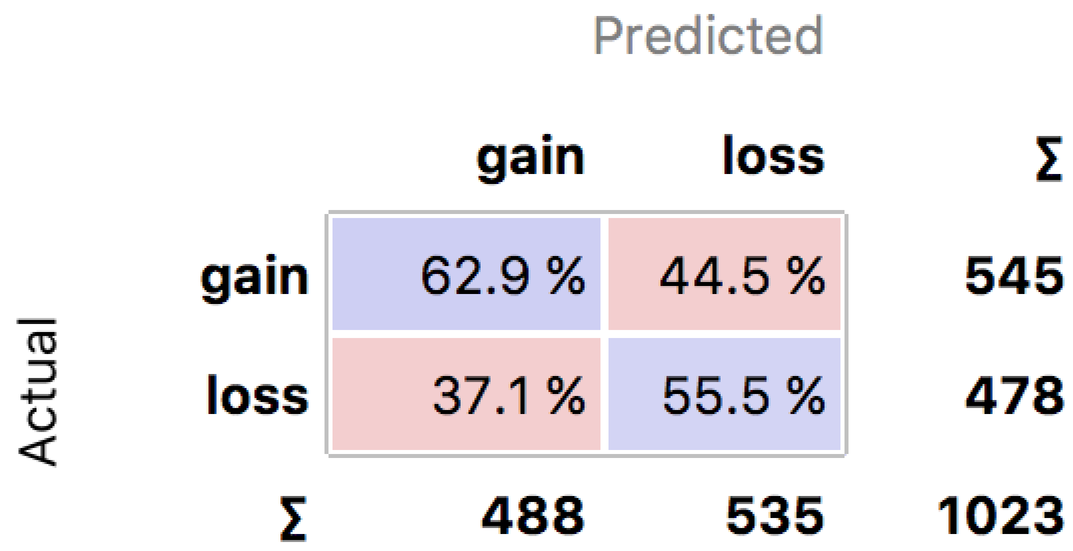
** INCLUDE SQL TO SHOW CHANGES MADE**

Data Mining Process

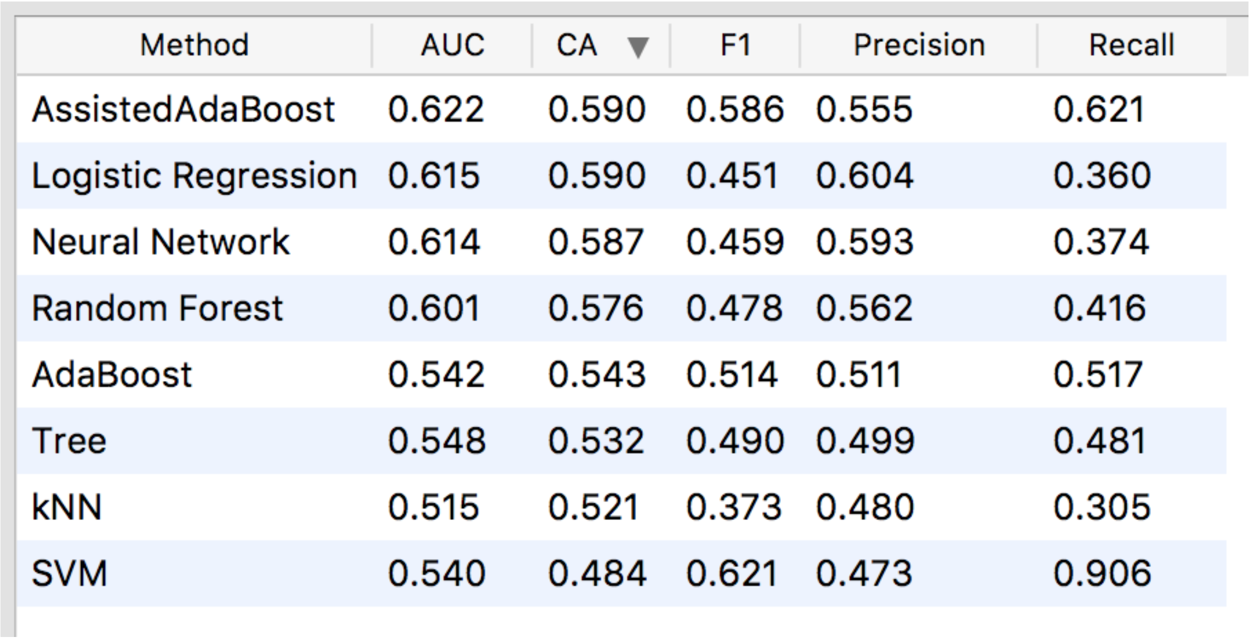
Before we could begin investigating which algorithms would build the best model, we had to clean our data. There were certain dates, especially in 2000-2001 where volume was not reported for one or more of the indexes. We decided to remove rows with incomplete data by incorporating a preprocessor in our Orange Canvas workflow. After cleaning our data, we split our data set into a training set and a testing set using 70% and 30% respectively of our total data. While we had originally attempted to build a regression model, we quickly decided that a continuous dependent variable was not necessary. Since the ultimate extension of this project would be to build a model that determines when to buy into the S&P 500, it was decided that a binary classification model would be more appropriate, and likely more accurate than the more complex regression model. The first model we attempted to build utilized a kNN algorithm. The best accuracy was achieved when using 6 neighbors in a uniformly-calculated Euclidean distance space. This still only resulted in a 52.1% accuracy and a precision of 48.0% and a recall of 30.5%. Upon analysis of the kNN false results (false positives and false negatives), we can conjecture that the ineffectiveness of this model is due to the high variance in independent variable values. The challenge of high variance was met again when attempting a decision tree model. A binary decision tree with a minimum of 3 instances per leaf and a minimum subset split of 5 instances was used to produce the tree with highest accuracy. A maximal tree depth of 100 was used, although this was believed to be irrelevant due to the limited number of features. The classification was set to stop when the majority reached 95%. This model offered a slight improvement over the kNN model with an accuracy of 53.2%, precision of 49.9%, and a recall of 48.1%. We still found these results to be subpar, especially considering the sub-50% precision and recall; again we believed this was due to the high variance in our data. It was becoming clear that any model that relied on the use of a distance space (a binary decision tree could be seen as acting on a Euclidean distance space in which branches form orthogonal set boundaries) would likely be ineffectual. Still, we attempted to use a Random Forest as a means of pushing the utility of the decision tree further. Using 50 trees with the same settings as the previously used decision tree model, we were able to obtain a 57.6% accuracy with 56.2% precision and 41.6% recall. While this model had given us the best results to that point, we still decided to move on to other models. In an attempt to move away from more restrictive and traditional distance-based classification models, we decided to embrace the blackbox model behind a neural network. Using 100 neurons per layer, an identity activation function, a stochastic gradient descent solver, and a regularization term of 0.00001 through 300 iterations, we were able to achieve 58.2% accuracy, 59.5% precision, and 32.2% recall. This brought us our most accurate model yet, but with a recall of less than one third, we were not content. We attempted a logistic regression using a Lasso (least absolute shrinkage and selection operator) regularization with an L1-norm max value of 1. This led to an accuracy of 59.0%, precision of 60.4%, and a recall of 36.0%. While these results were similar to what we achieved using a neural network, they did offer us one key advantage. Because logistic regressions support weighted learners, unlike neural networks, we were able to use our logistic regression model as an input to an AdaBoost algorithm. Our AdaBoost model used 50 estimators, a learning rate of 1 (discarding old data in the learner), a SAMME classification algorithm, and a linear regression loss function. This gave us an accuracy of 59.0%, 55.5% precision, and 62.1% recall. While there was a slight decrease in precision, this was the first model to provide a greater than 50% performance in all categories, including an F1 score of 62.1%.


Logistic Regression
Neural Network
Tree
Random Forest
SVM
AdaBoost
kNN
AssistedAdaBoost
Save Model
Model
Learner
Raw Data
Data
Preprocess
Preprocessed Data → Data
Data Sampler
Data Sample → Data
Remaining Data → Data
Training Data
Testing Data
Selected Data → Data
Selected Data → Test Data
Test & Score
Calibration Plot
Confusion Matrix
Evaluation Results
Selected Data → Data
Data Table
Predictions → Data
Predictions

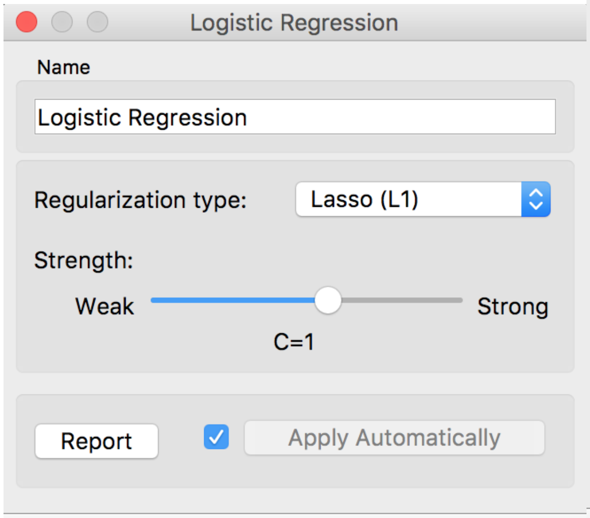
Results



		Predicted		
		gain	loss	Σ
Actual	gain	62.9 %	44.5 %	545
	loss	37.1 %	55.5 %	478
	Σ	488	535	1023



Method	AUC	CA ▼	F1	Precision	Recall
AssistedAdaBoost	0.622	0.590	0.586	0.555	0.621
Logistic Regression	0.615	0.590	0.451	0.604	0.360
Neural Network	0.614	0.587	0.459	0.593	0.374
Random Forest	0.601	0.576	0.478	0.562	0.416
AdaBoost	0.542	0.543	0.514	0.511	0.517
Tree	0.548	0.532	0.490	0.499	0.481
kNN	0.515	0.521	0.373	0.480	0.305
SVM	0.540	0.484	0.621	0.473	0.906


Logistic Regression

Name

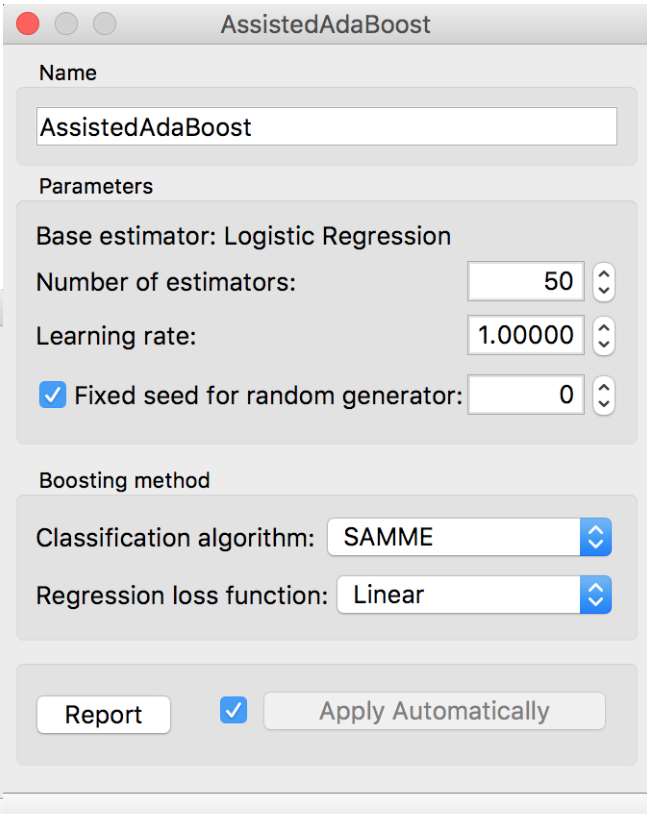
Logistic Regression

Regularization type: Lasso (L1)

Strength:

Weak Strong

C=1


AssistedAdaBoost

Name

AssistedAdaBoost

Parameters

Base estimator: Logistic Regression

Number of estimators: 50

Learning rate: 1.00000

☑ Fixed seed for random generator: 0

Boosting method

Classification algorithm: SAMME

Regression loss function: Linear

Conclusion

When attempting to predict the stock market, any accuracy greater than 50% generally translates to profitability. We were able to show, using a novel classification-based approach, that same-day performance of Eastern stock markets, coupled with past performance of U.S. markets can be used to predict U.S. markets with unambiguous precision. Through our analysis, we gained a deeper understanding of both regression and classification models and their strengths and weaknesses. These include the shortfalls of distance-based metrics when dealing with a high number of dimensions. Further, we were able to see a similar effect to this "curse of dimensionality" when the range of values for a particular variable was excessive compared to the average value. Ultimately, we were successful in overcoming realistic obstacles and applying the course material to a quintessential data analysis problem and achieving results that surpassed our own expectations.

Reference

1.Documentation https://orange.biolab.si
2.http://www.investopedia.com/terms/l/leadingindicator.asp