Big Data Analytics Final Report

--Yahoo Finance Stock Analytics

Group 9

Authors Names: Pingyuan Wang & Xuanyu Chen

Columbia University

pw2435@columbia.edu; xc2363@columbia.edu

Abstract:

Today, and increasingly in the future, stock market are surrounded by masses of data and raw information. Some of this data is very relevant but much of it is not. Further, most of that data is unstructured in the form of online review, email, documents, images and different types of social media, blogs, and so on. Unstructured data is extremely difficult to access and query, it is scattered across many different locations and formats, and it requires some form of preprocessing before it can be analyzed and used. Yet, it is this unstructured type data that is primarily exploding in quantity, representing around 80 percent of the annual growth of data and doubling in quantity every two years.

Today though, big data is increasingly being used to provide deep insight and predictive analysis into everything from stock market movements to individual investment behaviors. Those that are able to make use and harness the power of this disruptive force in markets will benefit by being smarter, faster and more efficient, meaning they are more likely to seize opportunities early and thereby profit. Thus, our initial goal aims to figure out how to use online data from Stocktwits to predict next day's stock market's performance, and compare the result with just using stock price data to predict next day's stock performance from Yahoo Finance. However, due to limited time, this report would mainly focus on prediction model using stock price data from Yahoo Finance.

Key Word: Stock market prediction, Machine Learning, Sytex detection, Spark

Introduction:

What is "Big Data" and how it predicts stock market? Big Data solutions are used for data sets that are too large and complex to manipulate or analyze with traditional methods or tools. It pays more attention on the relationship instead of reason among people and things all around. The importance and necessity of this field has rapidly increased because it gives people more accurate and expeditious insight of those unstructured data, which provide a better way to help people make decisions on every fields. Our project aims to help stock investors to predict the stock market with combination of structured stock index and unstructured online experts review. According to research, Wall Street stands to benefit from Big Data analytics by using advanced algorithms to track and predict the financial markets, though many individual investors do not figure out the relationship between those data and the stock market. (Roitman, L) The stock market are 100 percent neither random nor chaotic but can be discovered a trending according to those multifarious investors with different strategies, previous stock index, dynamic industry transformation, etc. Therefore, we could use the previous stock index and online stock review to understand the market forecasts.

Problems and Challenges: Our goal of this project is to use stock index from Yahoo Finance and stock online review from Stocktwits to forecast the stock market trending. The most challenging part of our project is that the unstructured review data are really difficult to classify, process and analyze. More detail will be shown on the following sections.

Related Works:

Many experts have already use Big Data Analysis in the area of stock market predictions.(Nasseri, A., Tucker, A., & Sergio, C) Most important problem on previous related work is modeling on analysis. For example, the classification of SVM as a feature selection criterion to predict the stock trending is commonly used. In our project, we will use three different model with four classification algorithms on analysis including classification of SVM.

System Overview:

In this report, three models along with four classification algorithms is developed, which we will compare the result in small sample and determine to use the one has better performance.
Our dataset are downloaded from Yahoo Finance and Stocktwits into Mongdb. The initial data size from Stocktwits is approximately 7GB and the initial data size from Yahoo Finance is approximately 3GB.

Algorithm:

Before we start to analyze the dataset, we designed three methods with four different classification algorithms and use the first ten stocks from S&P 500 to select two models with best performance to run the overall analysis for a larger dataset. Our first method is to use first three days' stock trend (increase/decrease) as variables to predict the following day's stock trend. Our second model is to use first three days' Close-Open price as variables to predict the following day's stock trend. Our third model is to use first day's Open, Close, High price to predict the following day's stock's trend. The first ten stock's testing results are shown below.

As you can see, the number followed by the Stock index is the number of data in the test set, the first number in each entry presents the accuracy rate the second number is the number of predicted values for increase. After several tries we decide to drop the logistic regression because it performs very bad for the method 1 and it cannot be used in method 2 and 3. As a result, we decide to use Method 1 with Naive Bayes and Method 2 with SVM that they have relevant higher accuracy and lower false positive rate. Those are all done in R.

MMM	503		
	Model 1	Model 2	Model 3
Naïve Bayes	52.88/503	49.7/421	52.88/503
Logistic Regr	51.09/268	NA	NA
SVM	52.88/503	52.29/498	52.88/503
Random For	52.88/503	48.5/275	53.68/455
ABT	503		
	49.9/443	50.50/410	49.1/9
	NA	NA	NA
	51.49/503	51.49/503	48.5/0
	42.48/446	49.30/296	48.5/244
ABBV	151		
	51.65/151	51.65/113	51.65/78
	51.65/151	54.30/127	51.65/151
	51.65/151	52.32/94	52.31/90
A-C-N	503		
	54.87/503	54.68/446	45.12/0
	54.87/503	55.67/485	54.87/503
	54.87/503	49.3/313	54.87/471
ATVI	503		
	49.7/199	49.1/224	51.09/0
	48.7/258	49.1/70	52.88/67
	49.3/137	51.09/244	50.89/9
First five with grid search for SVM's gamma and cost			
AYI	503		
	50.30/199	45.52/291	51.09/0
	50.7/395	46.32/291	46.72/343
	53.08/503	47.12/295	47.91/7
ADBE	503		
	56.46/503	51.49/358	43.53/0
	56.46/503	53.28/355	48.90/393
	56.46/503	51.88/264	44.53/7
AAP	503		
	47.71/254	49.9/205	50.7/27
	49.3/304	52.09/260	49.9/359
	50.3/189	50.89/250	50.5/0
AES	503		
	53.68/201	52.49/139	51.49/60
	52.29/258	53.88/146	49.9/0
	53.88/138	52.08/245	50.3/188

After the model determination, we decide to use 10 years S&P 500 data, the steps are shown below:

1. Download 10 years S&P 500 stock price data from Yahoo Finance
2. Data cleaning in Jupyter Notebook
3. Convert data libsvm format in Python
4. Run SVM in Scala, 80% training and 20% test
5. Use spark to run Naive Bayes

Software Package Description:

R Package

Quantmod, e1071, randomForest, ggplot2, misc3d, rgl, MASS, pRoc, httr, XML, sparklyr, sparkR

Quantmod, misc3d, rgl, MASS, httr, XML are used to download the Yahoo Finance data. E1071 and randomForest are used to run classification algorithms in R. Sparklyr and sparkR is used to run spark on R. We used spark.naiveBayes to run naive bayes on R spark. Ggplot2, misc3d, pRoc are used to visualize SVM's result.

Python Package

Panda, sys, cvs

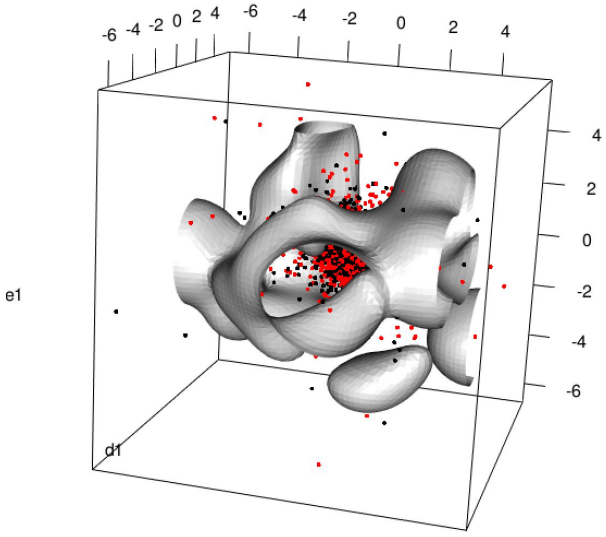
We used panda to convert the data to the structure we want because its faster speed. Sys and csv is used to convert the original data to libsvm data structure that scala required to run svm.

MLlib

MLlib is a spark machine learning library that allows us to implement machine learning algorithm on spark with specified input data structure. We used this library to run SVM.

Experiment Result:

SVM VISUAL



As you can see, the dots are stock's trend, red is increasing and black is decreasing. The curve shows that the SVM algorithm split the data to two categories.

The SVM result provides an approximate 50.4% accuracy:

```
scala>

scala> val scoreAndLabels = test.map { point =>
     |   val score = model.predict(point.features)
     |   (score, point.label)
     | }
scoreAndLabels: org.apache.spark.rdd.RDD[(Double, Double)] = MapPartitionsRDD[42
] at map at <console>:39

scala> val metrics = new BinaryClassificationMetrics(scoreAndLabels)
metrics: org.apache.spark.mllib.evaluation.BinaryClassificationMetrics = org.apa
che.spark.mllib.evaluation.BinaryClassificationMetrics@4cf5cf47

scala> val auROC = metrics.areaUnderROC()
auROC: Double = 0.5040873185079047

scala>

scala> println("Area under ROC = " + auROC)
Area under ROC = 0.5040873185079047

scala>

scala>
```

Then, we run Naive Bayes on RSpark:

```
import org.apache.spark.ml.classification.NaiveBayes
import org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator

// Load the data stored in LIBSVM format as a DataFrame.
val data = spark.read.format("libsvm").load("/Users/Van/Desktop/sample.csv")

// Split the data into training and test sets (30% held out for testing)
val Array(trainingData, testData) = data.randomSplit(Array(0.7, 0.3), seed = 1234L)

// Train a NaiveBayes model.
val model = new NaiveBayes().fit(trainingData)

// Select example rows to display.
val predictions = model.transform(testData)
predictions.show()

// Select (prediction, true label) and compute test error
val evaluator = new MulticlassClassificationEvaluator().setLabelCol("label").setPredictionCol("prediction").setMetricName("accuracy")
val accuracy = evaluator.evaluate(predictions)
println("Accuracy: " + accuracy)
```

The Naive Bayes provides us an approximate 53% accuracy. Since Rspark does not allow a large input dataset. We divide the data to 20 small samples and run the analysis, the overall performance is roughly 53% of accuracy, which is very similar to the result in our test sample. Before the analysis, we believe the accuracy should be higher than the result we get; however, without utilization of the data from Stocktwits, the overall performance can hardly be significant high. Since we only use previous stock index, all the other factors, such as those multifarious investors with different strategies, dynamic industry transformation and other news, are not fully considered.

Conclusion:

In this project, we learn the basic knowledge background of Data analysis and have done a stock index prediction with scientific method. Our implemented models are intuitive and visualized, the most important thing we learned in doing this project is that never stop trying, as we all did so. Although the final accuracy is not tremendously improved, it still indicates that previous stock data can be used to predict stock trend. Moreover, with the knowledge we learned in this

course, we can use Big Data Analysis to solve problems or figure out anything we want to know in the future.

Contribution

Pingyuan Wang (60%)

Implement model, run analysis, presentation

Xuanyu Chen (40%)

Literature review, final report, presentation slides

Future Work

As we encounter lots of problems when we try to do sentiment analysis on Stocktwits data, we would mainly focus on solving those problems as future work.

Problems:

1.Customers reviews on stocktwits are not independent. First one write review could be assumed as independent but the following cannot because they may see the previous posts.

2. Lots of reviews do not show strong positive or negative predictions with sentiment analysis, we tried a small sample on text-processing but the result is not good. In addition, just using positive and negative as baseline to predict, it's hard to decide what positive/negative rate could be regarded as a prediction to increase/decrease.

3. Number of reviews for each day and for each stock varies a lot without sufficient number of reviews, the result is not reliable.

4. People write comments on some specific days, some stocks, difficult to have a standard to give weights for each individual

Acknowledgement

The authors would like to thank Dr.Lin, Eric, and all TAs all help during this semester. Since we two are all pretty new to this field, and really try our best on this final project. At the beginning of the project, we proposed to predict the stock trending with the combination of previous stock index and stock online reviews; however, due to limited time, we only have the Yahoo Finance analytics done. And we got all the stocktwits reviews data from an AI researcher Junyan Wu. We would also to thank him even though we did not really use that data too much.

Reference:

1. Django. http://text-processing.com/demo/sentiment/
2. Nasseri, A., Tucker, A., & Sergio, C. (2014). Big Data Analysis of StockTwits to Predict Sentiments in the Stock Market. Discovery Science,8777, lecture notes in computer science, 13-24. Retrieved December 21, 2016.
3. Roitman, L., Dr. (2016, August 25). The "Big Data" Solution For Wall Street. Retrieved December 21, 2016, from https://iknowfirst.com/the-big-data-solution-for-wall-street
4. Spark MLlib http://spark.apache.org/docs/latest/mllib-linear-methods.html#linear-support-vector-machines-svms http://spark.apache.org/docs/latest/sparkr.html#naive-bayes-model
5. UCLA. http://web.cs.ucla.edu/~mtgarip/linear.html