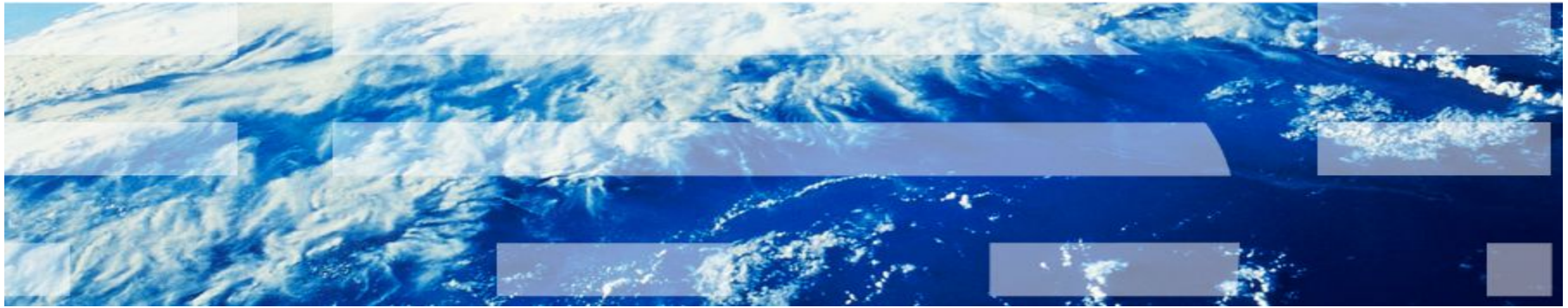
E6893 Big Data Analytics:

Yahoo Finance Stock Analytics
Group 9

Team Members:
Pingyuan Wang
Xuanyu Chen



November 17, 2016

Business Value

1. Stock reviewing websites are surrounded by masses of unstructured data.(social media, blogs,etc)
2. Big data analysis is increasingly being used to provide deep insight and predictive analysis into stock market movements and 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.

Implementation and Algorithm

Data

- Yahoo Finance

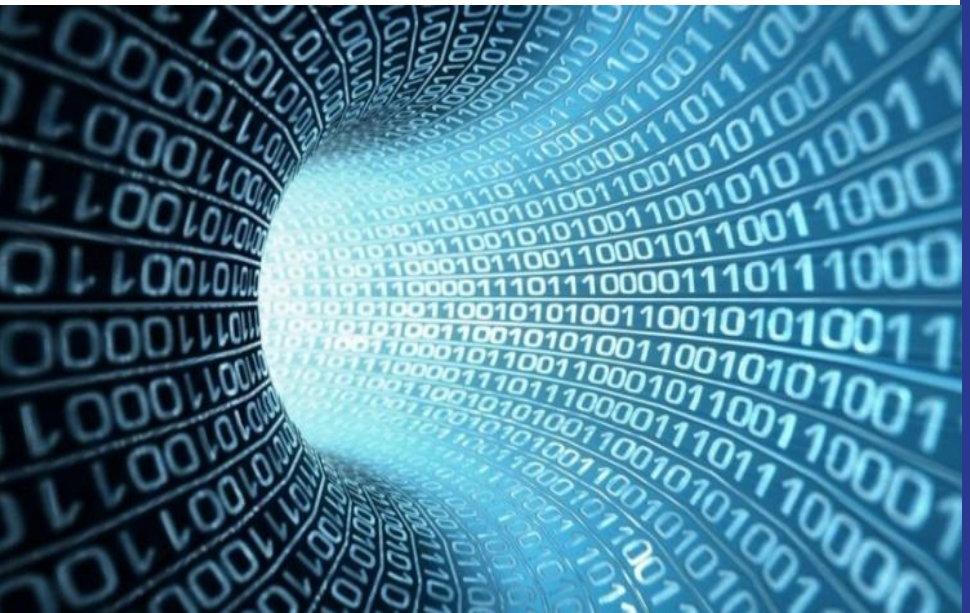
Algorithm

- Classification Method in Yahoo Finance
- Naive Bayes
- Logistic Regression
- Random Forest
- Support Vector Machine

Software Used

- R
- Python
- Spark

Process



1.Download 10 years S&P 500 stock data form Yahoo Fomamce

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 sparklyr and SparkR to run Naive Bayes in RStudio

Dataset Structure

From Yahoo Finance

	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
2007-01-03	86.29	86.58	81.90	83.80	309579900	10.90416
2007-01-04	84.05	85.95	83.82	85.66	211815100	11.14619
2007-01-05	85.77	86.20	84.40	85.05	208685400	11.06681
2007-01-08	85.96	86.53	85.28	85.47	199276700	11.12147
2007-01-09	86.45	92.98	85.15	92.57	837324600	12.04533
2007-01-10	94.75	97.80	93.45	97.00	738220000	12.62176
2007-01-11	95.94	96.78	95.10	95.80	360063200	12.46562
2007-01-12	94.59	95.06	93.23	94.62	328172600	12.31207
2007-01-16	95.68	97.25	95.45	97.10	311019100	12.63477
2007-01-17	97.56	97.60	94.82	94.95	411565000	12.35501
2007-01-18	92.10	92.11	89.05	89.07	591151400	11.58990
2007-01-19	88.63	89.65	88.12	88.50	341118400	11.51573
2007-01-22	89.14	89.16	85.65	86.79	363506500	11.29322

Challenges and Finding

Challenges

- Data is too messy
- ML algorithm did not work well on Yahoo Finance Data

Finding

1. Use first three days performance to predict the first day's performance
2. Use first three days' performance as continuous variables to predict forth day's performance
3. Use first day's stock price to predict second day's predict second day's performance

Conclusion

```
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>
```

```
1  import org.apache.spark.ml.classification.NaiveBayes
2  import org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator
3
4  // Load the data stored in LIBSVM format as a DataFrame.
5  val data = spark.read.format("libsvm").load("/Users/Van/Desktop/sample.csv")
6
7  // Split the data into training and test sets (30% held out for testing)
8  val Array(trainingData, testData) = data.randomSplit(Array(0.7, 0.3), seed = 1234L)
9
10 // Train a NaiveBayes model.
11 val model = new NaiveBayes().fit(trainingData)
12
13 // Select example rows to display.
14 val predictions = model.transform(testData)
15 predictions.show()
16
17 // Select (prediction, true label) and compute test error
18 val evaluator = new MulticlassClassificationEvaluator().setLabelCol("label").setPredictionCol("prediction").setMetricName("accuracy")
19 val accuracy = evaluator.evaluate(predictions)
20 println("Accuracy: " + accuracy)
```

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.


thank you
рахмат
danke
謝謝
ngiyabonga
teşekkür ederim
Баярлалаа
спасибо
faafetai lava
mersi
kia ora
barka
welalin
spas
tack
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
gracias
kiitos
dankie
nanni
nandri
dhanyavad
bedankt
bayarlalaa
gracje
hvala
mauruuru
köszönöm
dankon
akun
aciu
хвала
asante
manana
obrigada
tenki
enkosi
djiere dieuf
tau
mochchakkeram
murakoze
chokrane
дякую
mamnun
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
go raibh maith agat
sobodi
dėkuji
sağolun
sukriya
kop khun krap
taiku
arigatô
takk
dakujem
trugarez
obrigado
mèsi
didi madloba
najis tuke
ありがとう
terima kasih
tanemirt
rahmet
grazie
ευχαριστώ
diolch
dhanyavadagalu
shukriya
mercé
мерси
kam sah hamnida
rahmat
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci