**Impact of Cyber-attack on stock price of a publicly traded company**

**Final Report**

**Team:**

**Gaurav Jain  
Kartik Kapadia  
Kaushal Shah  
Pradeep Yadav  
Vaibhav Shah**

Problem Definition 3

Description of Input Data 4

Example JSON response using API (metadata) 4

Sample Article 5

Explanation of your Big Data strategy 6

Data Flow Diagrams 7

High Level Steps 7

Data Preprocessing 7

Details of expected Result 8

Details of how your application handles bad or missing data and is your strategy robust 8

Analysis of Results 9

Multiple R: 9

Further Analysis: 10

Correlation Table: 11

Conclusion 13

Describe briefly what you learnt in this project. 13

Describe briefly how this project can be improved 14

Role of Each Team Member 14

# Problem Definition

One of the goals of this project is to analyze news articles related to a few publicly traded companies that have been the target of cyber-security attacks. In quite a few cases the cyber-attacks may have occurred for weeks or months before it was detected and counter-measures were put in place. It will be interesting to compare counts of news articles about a company related to cyber security or attacks prior to the company releasing information about a cyber-attack, and the counts of such articles after the company releases this information.

For this project we plan to analyze news articles for approximately two months; one month before the company released information about the attack and one months after. Each news article will be classified as being related to cyber security/attack or not by analyzing the content of the article. Counts of cyber security related articles for each day in the 2-month period will be tracked.

For the same 2-month period, the stock price of the company will be analyzed. It will be interesting to see if there is any correlation in the change of stock price to the number of the cyber-security related articles.

* Change, if any, to the number of articles related to cyber security for the company that was attacked over the 2-month period being considered.
* Change, if any, to the stock price of the company during the same time period.
* Conclude if there is a significant impact of the cyber-attack to the stock price of the company.

# Description of Input Data

For data collection we used:

* Google Search API
* New York Times API

The data obtained is stored in text files and then transferred to HDFS for processing.

## Example JSON response using API (metadata)

{

"web\_url":"http:\/\/www.nytimes.com\/2014\/12\/30\/opinion\/deterring-cyberattacks-from-north-korea.html",

"snippet": "The recent hacking on Sony Pictures should cause corporations and governments to redouble efforts to protect their networks.",

"lead\_paragraph": "The recent hacking on Sony Pictures should cause corporations and governments to redouble efforts to protect their networks.",

"pub\_date": "2014-12-30T00:00:00Z",

"headline":

{

"main": "Deterring Cyberattacks From North Korea ",

"kicker": "Editorial",

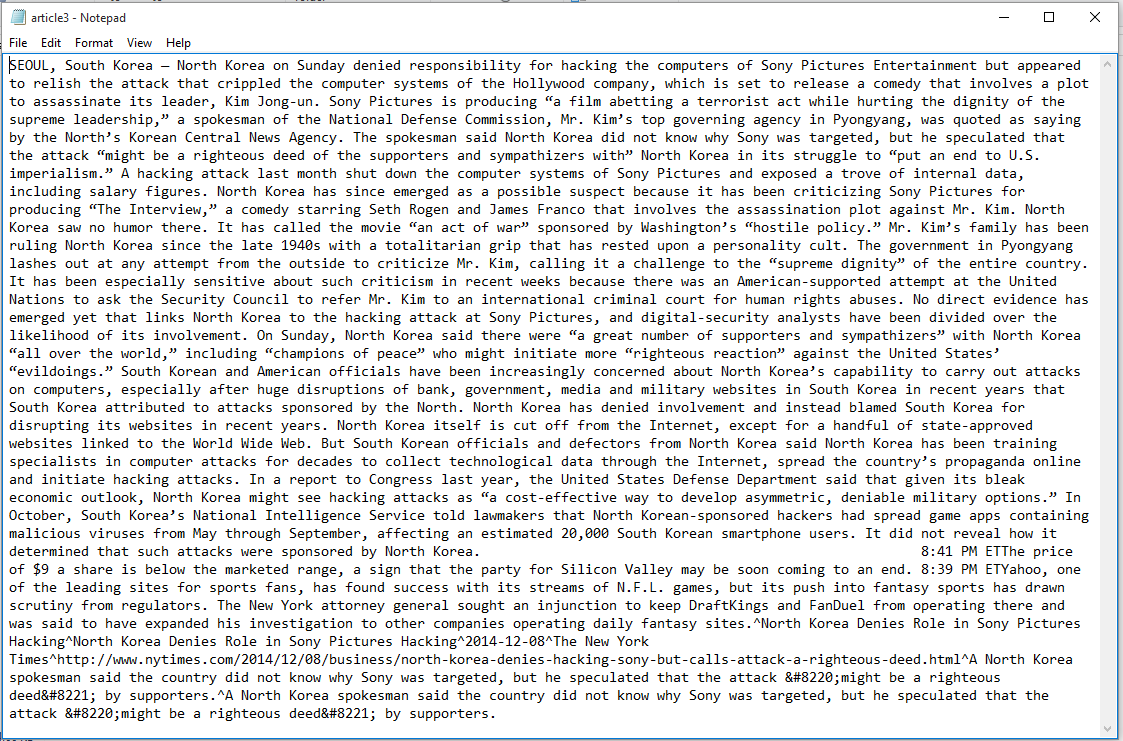
"print\_headline": "Deterring Cyberattacks From North Korea",

"content\_kicker": "Editorial"

}

}

# Sample Article



# Explanation of your Big Data strategy

We need to generate a dictionary of unique words from all the articles obtained for a particular company. For this we use Map Reduce.

The Map Reduce takes input all the text files i.e. the articles of a company and outputs all the unique words.

**Mapper**

Input key: Byte offset of each line of an article

Input value: Content of the line

Output key: Word

Output value: Count of word in the key

**Reducer**

Input key: Word

Input value: List of counts of word in key

Output key: Word

Output value: Count of word in key

We use this dictionary to generate a TF-IDF vector using java code.

For classification we use Apache Spark MLlib which is a machine learning library containing implementations of popular classification algorithms.

We use two algorithms for classification:

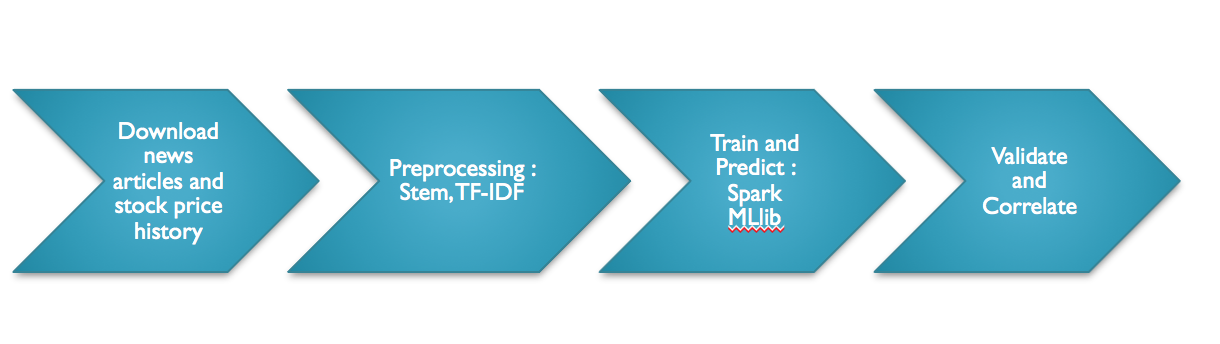
* Naïve Bayes
* Support Vector Machines

We use Scala to build and test our model on Sparl MLlib.

We compare the accuracy of both these algorithms on a small dataset that is manually labeled.

# Data Flow Diagrams

## High Level Steps



## Data Preprocessing

Word1, 1

Word2, 1

Word3, 1

.

.

.

Wordn, 1

Articles

Mapper

Word1, <1,1…>

Word2, <1,1…>

Word3, <1,1…>

.

.

.

Wordn, <1,1…>

Word1, 34

Word2, 67

Word3, 45

.

.

.

Wordn, 111

Reducer

Dictionary

# Details of expected Result

The output will be in a CSV format contain:

* Date
* Count of cyber security articles published on that date
* Stock price of the company on that date.

Various visualization techniques can be used to create a graph or charts from this data. This graph will depict the co-relation between online publishing of articles about a company and its stock price if any.

# Details of how your application handles bad or missing data and is your strategy robust

Initially the news article downloaded by the crawler had a lot of extra data with it, for example the ads on the page, links to other articles and comments. After some research we found that these can be ignored by getting everything in “ptag”.

For the 2-month period we were interested in, we found news articles for every day. Sometimes when there is too less data in an article our web crawler simply ignores that article.

# Analysis of Results

1. Comparison of classification between SVM and NB

A set of 100 articles were manually labeled as Cyber Attack related or not. The labels and term frequency vectors from this set of 100 labeled articles was used to determine the accuracy of the classification algorithms. In each test, 90% of the data was used for training and 10% for tests. The tests were repeated 5 times with training and test data selected randomly. For SVM, the accuracy was calculated as the Area under the ROC curve.

Average Accuracy for Naive Bayes : 81.34%

Average Area under ROC Curve for SVM : 90.4%

Then the 100 article labeled dataset was used as training data for predicting the classification of the remaining ~2500 articles. These articles were from a month prior to and a month after the details of the cyber attack were released by the company, so they contained articles that were related to the Cyber Attack and those that were not. For 84% of the unlabeled articles, the classification predicted by both the algorithms matched. Here are some details.

Total Articles : 2485

Classification Matched : 2090

Classification did not match : 395

Match % : 84.1

In 96% of the cases where the classification did not match, the article was classified as not Cyber Attack related by SVM while Naive Bayes classified it as Cyber Attack related.

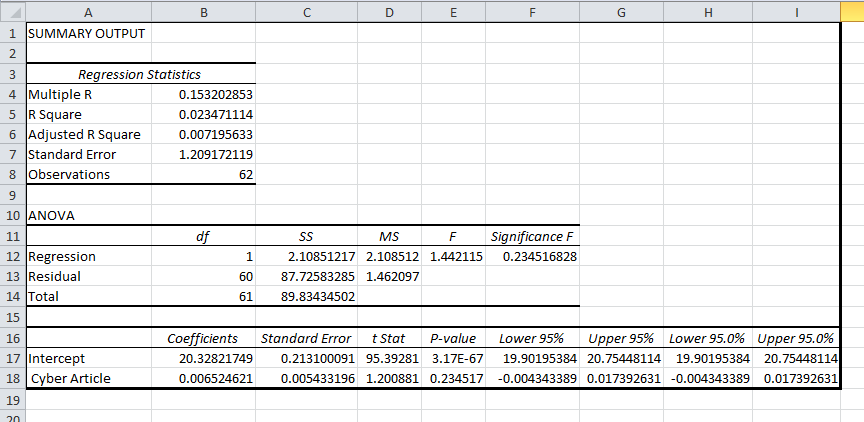
## Multiple R:

Square root of R-squared. It represents the correlation between the closing-price for each day within the selected time-frame. The value of 1 represents perfect linear relationship and 0 represents no correlation between the variables being tested.

As we can see below, the Multiple-R value is 0.1532 which is very low and it is evident that the number of articles related to cyber security about Sony Pictures had no significant impact on the stock price of the company.

## Further Analysis:

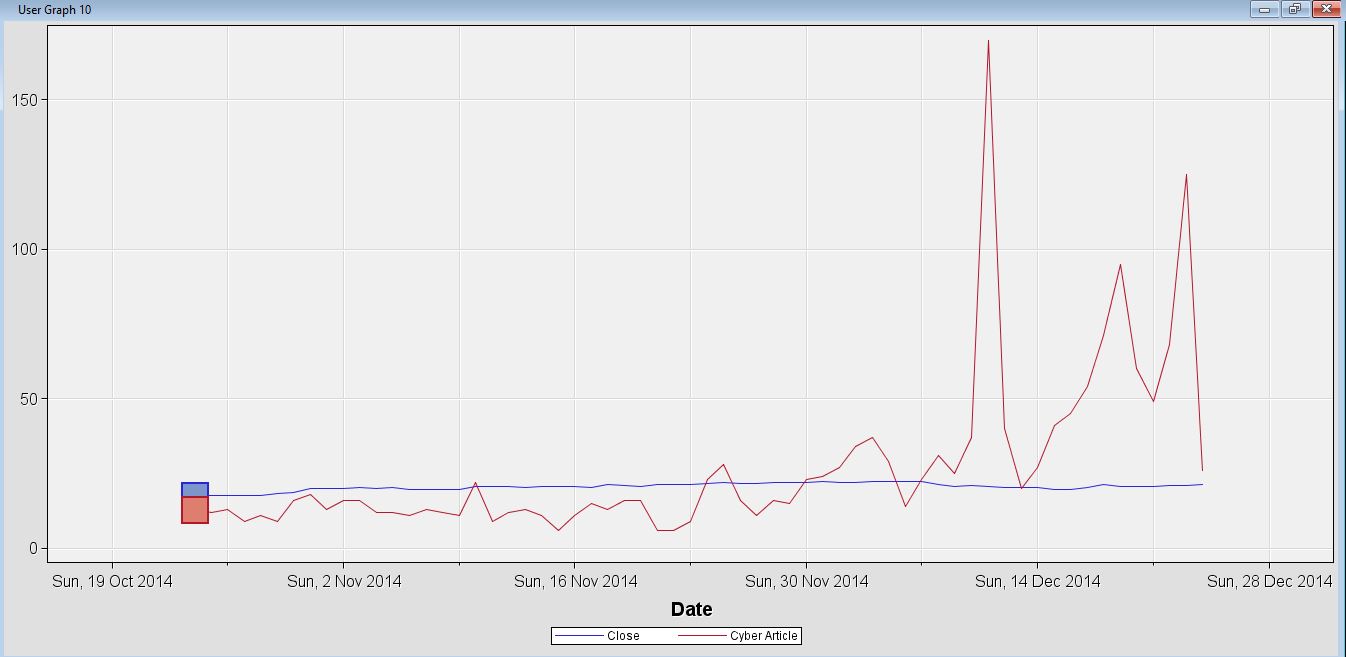
This is the coefficient of determination, which essentially tells us how many points fall on the regression line. i.e. What percentage of data fits the model. As we can see, only 1.12% of the values fit the model. This makes it clear that number of cyber security related articles are not correlated with the change in stock price of the company.



## Correlation Table:

The table below is plotted with the change in the stock value of sony corporation (SNE) from a period of 2 months (1 month before and 1 month after the cyber-attack) versus the number of articles corresponding to that day. X-axis values are sorted in ascending order.

This shows that the change in stock price cannot be predicted by just looking at the number of cyber security related articles on the day.

****

# Conclusion

Generating a dictionary from a large number of articles was not an easy task with normal programming but using MapReduce and HDFS it was a matter of seconds. Also, one does not need to be concerned about how the work will be distributed across various nodes in the cluster and how results will be aggregated. Hadoop takes care of that.

Classifying ~2500 news articles as Cyber Security related or not would not be an easy task if done manually. The machine learning algorithms related to text classification are the key here. Using the implementation in Spark MLlib, the classification task was quick and automated. This process can be scaled to much larger datasets fairly easily whereas manual tagging can get very expensive and time consuming.

Advantages:

* Takes less time compared to manual classification.
* Process can be automated to scale with the dataset.
* Hardware resources are managed by Hadoop platform.
* Do not have to worry about internal implementations of classification algorithms

Disadvantages:

* Have to transfer data to HDFS for even a trivial task.
* Debugging map reduce jobs is hard.

## Describe briefly what you learnt in this project.

One of the first things we learnt was using News and Search APIs to get information about news articles. This however did not provide the full content of the article. So we had to learn to use a web crawler to download the complete content of the articles.

This project helped us to learn basics of Scala programming and RDDs. We learnt about the sparse and dense vector formats, and how sparse format can be so efficient when the data is sparse. We learnt how to transform RDDs using various functions. The number of news articles related to one company for a two month period can be very large and not easy to classify manually. So we studied the various machine learning algorithms related to text classification. We learnt that two commonly used algorithms are Naive Bayes and Support Vector Machines. Spark MLlib has implementations for both these, so we learnt how to use the MLlib implementations of these.

Though initially we thought otherwise, the analysis shows that Cyber Attack does not necessarily have a big impact on the stock price of the company.

## Describe briefly how this project can be improved

* We have targeted a few companies from the long list of the companies that suffered cyber attack. More data (from other companies) need to be collected to better understand the impact of cyber attack on the stock price.
* In some way the tagged data used to train classification models could be increased for better efficiency.
* Better techniques for cleaning and preprocessing the data could be used which would give more relevant data.
* Currently, we have compared results of two classification algorithms: Naive Bayes and SVM. More text classification algorithms can be explored and the one with best accuracy can be chosen for optimal classification.

# Role of Each Team Member

Data Collection from NYTimes API: Kartik and Pradeep

Data Collection from Google Search API: Vaibhav

Pre-Processing and TF-IDF: Kaushal

Training and testing: Gaurav and Kartik

Validation of results: Gaurav, Kaushal, Pradeep