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

**Final Report**

**Team:**

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

**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 four-month period will be tracked.

For the same four-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 four-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.

**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 Apache MLlib.

We compare the accuracy of both these algorithms and suggest which one should be used in a real world application.

**Data Flow Diagram:**

Diagram for big data strategy:

## Data Preprocessing







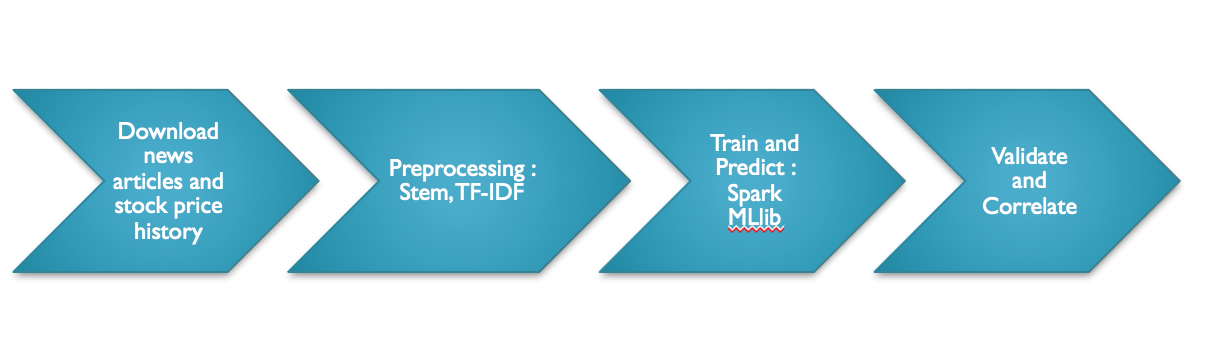








Data flow diagram for machine learning:



**Details of expected Result:**

The output will be in a CSV format contain:

* Date
* Number of articles related to cyber security 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**:

Since we are downloading articles from the internet, there is no missing data. 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

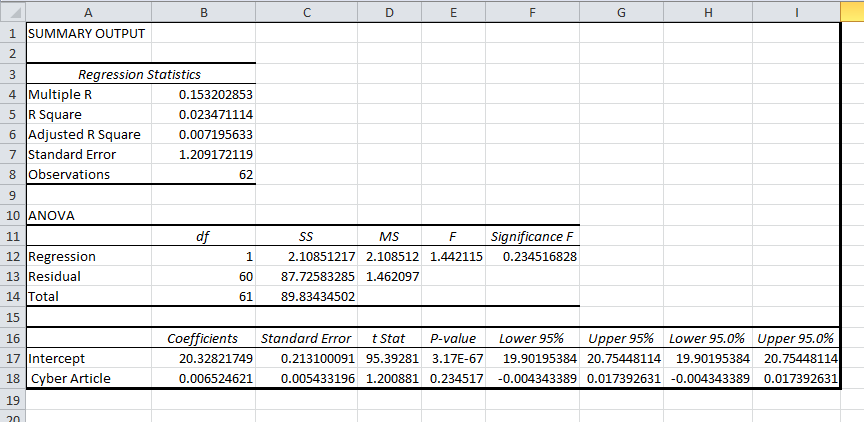
In 96% of the cases where the classification did not match, the article was classified as not Cyber Attack related by SVM while Naive Bayes classified it as Cyber Attack related.

**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 above, 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.

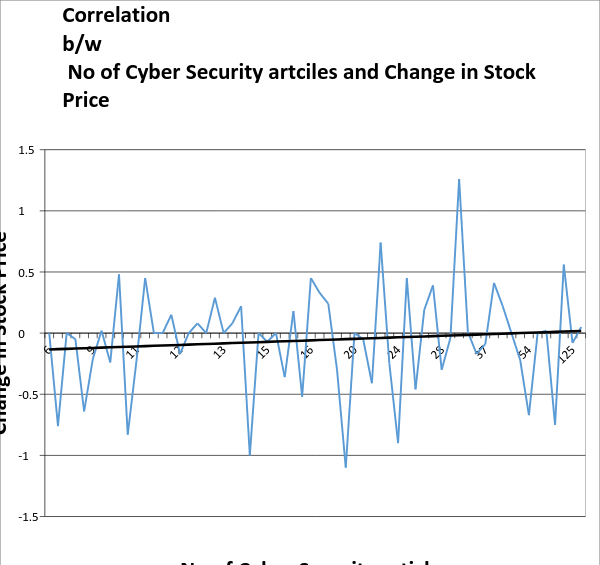
**Analysis of Results:**

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 do not determine the stock price of a company.



Sorted on No.of Articles:

The table below is plotted with the change in the stock value of sony corporation (SNE)



**Conclusion:**

**Explain how using Big Data helped you with this project? What are the advantages and disadvantages of using Big Data for this analysis?**

Generating a dictionary from a large number of articles was not an easy task with normal programming but with big data it was a matter of seconds.

Classification algorithms on Apache MLlib are also very fast giving the desired output in very less time.

Advantages:

* Takes less time
* 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

**Describe briefly what you learnt in this project.**

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.

**Role of Each Team Member:**

Data Collection from NYTimes API: Kartik and Pradeep

Data Collection from Google Search and Boss Yahoo API: Vaibhav

Pre-Processing and TF-IDF: Vaibhav and Kaushal

Training and testing: Gaurav and Kartik

Validation of results: Gaurav and Kaushal