[INST737] Digging into Data

# Milestone 3 & Final Report

December 3, 2014

Makoto Asami & Pulkit Kaushal
iSchool, University of Maryland

## Our Research Question

How people's online activities such as "tweeting" relate to their countries' various indices such as GDP per capita or unemployment rate?

## Data Collection & Cleaning

### 1. Collecting tweets from each country

We collected tweets from different countries; geo-location information (latitude and longitude) of the time of tweeting was used to determine within which county's boundary the tweet was made; also place information registered with the user's account was used to determine the country when the tweet didn't have geolocation information; tweets with none of these information was not collected. We made a Python code (tweet_per_country.py) to undertake this task for each of 214 countries and regions in the world as R turned out to be not capable of this task. The code basically collect maximum of 1000 recent English tweets from a certain country in JSON format, retrieve seven JSON attributes (such as the tweet's id, date and time of creation, and text), remove \n (change line), comma and double quotation from the text to prevent erroneous separation of each tweet in later phases, and output them as a CSV file. We only collected English tweets for ease of sentiment analysis in the later phase.

[Limitations and Considerations]

・Spans of collected tweets differ significantly between countries as some countries produce 1000 tweets (with location information, English) in a minute while some country takes 10 days to do the same.

・We couldn't collect 1000 tweets for all 214 countries and regions because our code could only get tweets up to as old as 8 days and we couldn't allocate enough time for collecting tweets for some countries to produce 1000 tweets as our project time was limited.

・It is difficult to define what it means that a tweet is attributed to a certain country. It could be tweeted by a resident of that country, a foreigner traveling in the country, a national of the country who is an expatriate now but haven't updated a location setting of his profile, or someone with no relation to the country who meaninglessly set the country as his profile location. So it would be difficult to make sense of analyses even when they show significant statistical outcome without additional researches.

・We could only get tweets of 2014 November and December, while the countries' indices data we got from World Bank is 2000 - 2012. So we couldn't compare tweets and countries' indices of the same year. We mainly used 2012 data of indices assuming that they should be the most similar to those of 2014, but it should be noted that we basically examined how countries' indices of the past relate to tweets of the present days.

## 2. Tweets' Sentiment Analysis

We used an R code (sentiment.R) after discussing with our fellow mates to conduct sentiment analysis for each tweet. The code counts how many positive words and negative words registered per tweets in CSV files and gives a cumulative score accordingly per tweet (positive.csv & negative.csv) there are, subtracts number of negative words from that of positive words, and outputs resulted number for each tweet. We calculated average of scores of all tweets in a country and used it as the sentiment score of the country to later examine relationship between it and other indices of the country.

## 3. Rewriting of CO2 Emissions data with 2010 data

We decided to compare tweets data of 2014 and countries' indices data of 2012, but noticed there were no data for CO2 emissions (EN.ATM.CO2E.PC) of 2012 and 2011 in the original dataset. We filled this column with CO2 emissions data of 2010 by adding some steps in our code (bank2012.r) used previously and running it.

**Support Vector Machines: SVM's**
**Dataset used: bank.csv**
**Rcode: svmcode.R**

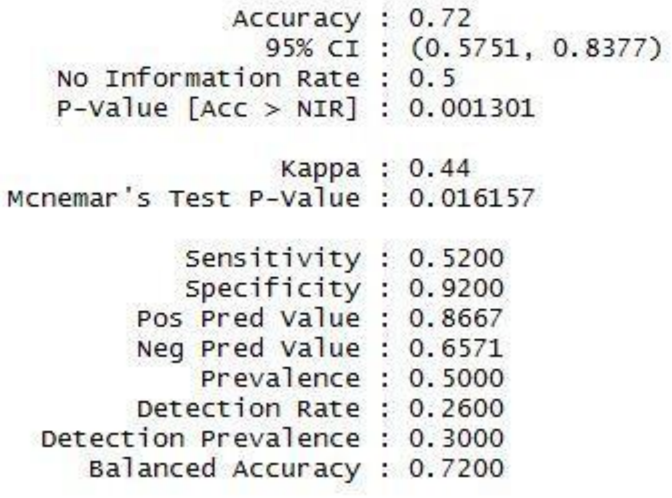
SVM's helps us in classification and regression analysis. Here, we will be conducting Linear SVM, Non-Linear SVM and Multi class SVM. Following is the explanation for each respevtively:

**Linear SVM**: For conducting the analysis we first prepared our dataset by adding Sentimental analysis to the grid and by some Data Manipulation. Then we created a

categorical variable based on the GDP per capita in current dollars and named it High which was going to be our Classifier. It had two values Either yes or no based on the condition that if the GDP is greater than $3500 then yes else No. Then appended High in the dataset. Used **kernlab package.**
Next step was splitting the dataset in training and testing. For training and testing we divided it into 80:20 ratio approx (76-24). Then setting seed and using KSVM applying our classifier “high” with the independent variables such as Country.Code, NY.GNP.PCAP.CD we conducted classification and prediction. And Calculated F1 Score :
2* precision * Recall / (precision+recall)

| | 1 | 2 | 3 | 4 | 5 | Best |
|---|---|---|---|---|---|---|
| Precision | 0.6111 | 0.875 | 0.8421 | 0.84 | 0.8667 | 0.875 |
| Recall | 0.6111 | 0.7 | 0.7273 | 0.875 | 0.52 | 0.875 |
| Specificity | 0.7812 | 0.85 | 0.8929 | 0.8462 | 0.92 | 0.92 |
| Accuracy | 0.72 | 0.76 | 0.82 | 0.86 | 0.72 | Av=0.776 |
| F1 Score | 0.6111 | 0.7777 | 0.7805 | 0.857143 | 0.650009 | 0.857 |

**Result:** And finally the average Accuracy came out to be 0.776
sample for the Linear SVM.



```
               Accuracy : 0.72
                 95% CI : (0.5751, 0.8377)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.001301

                  Kappa : 0.44
 Mcnemar's Test P-Value : 0.016157

            Sensitivity : 0.5200
            Specificity : 0.9200
         Pos Pred Value : 0.8667
         Neg Pred Value : 0.6571
             Prevalence : 0.5000
         Detection Rate : 0.2600
   Detection Prevalence : 0.3000
      Balanced Accuracy : 0.7200
```

**Non Linear SVM’s**

For the Non-Linear SVM we again conducted the same steps as Linear SVM until classification. While conducting KSVM changed kernel's Vanilladot to rbfdot to get Non Linear classification. Than Ran Confusion matrix the result was as following.

| | 1 | 2 | 3 | 4 | 5 | Best |
|---|---|---|---|---|---|---|
| Precision | 0.6923 | 0.6452 | 0.68 | 0.6316 | 0.7619 | 0.76 |
| Recall | 0.8571 | 0.8696 | 0.7083 | 0.6316 | 0.6957 | 0.869 |
| Specificity | 0.7241 | 0.5926 | 0.6923 | 0.7742 | 0.8148 | 0.8148 |
| Accuracy | 0.78 | 0.72 | 0.7 | 0.72 | 0.76 | Av=0.736 |
| F1 Score | 0.765936 | 0.740778 | 0.693862 | 0.6316 | 0.727297 | 0.765 |

**Multi- Class SVM's**
**Dataset used: bank.csv**
**Rcode USED: svmmulti.R**
For Multi class SVM we needed to have a variable with 3 or more categorical value. So we used "Category" from our dataset as our classifier as it depends on GDP and other factors to be determined. And used "Country.Code+ NY.GNP.PCAP.CD + Sentiment.2014tweets." as the independent variables. Used package **e1071 and caret** for this part of analysis. And used svm instead of KSVM as we are doing multi class svm here, took off the kernel value and ran the Confusion matrix twice and the result for the three variables respectively was as follows:

| | 1st | 1st | 1st | 2nd | 2nd | 2nd |
|---|---|---|---|---|---|---|
| | HD | LD | MD | HD | LD | MD |
| Precision | 0.9048 | 0.6207 | 0 | 0.7419 | 0.5263 | 0 |
| Recall | 0.8261 | 0.9 | 0 | 0.7931 | 0.7143 | 0 |
| Specificity | 0.9259 | 0.6333 | 1 | 0.619 | 0.75 | 1 |
| Accuracy | 0.74 | 0.74 | 0.74 | 0.66 | 0.66 | 0.66 |
| F1 Score | 0.863661 | 0.734701 | 0 | 0.766646 | 0.606055 | 0 |

Calculated F1 score:
2* precision * Recall / (precision+recall)

Result: Average Accuracy for Multi Class SVM was **0.70**

**Confusion Matrix and Statistics**

| | **Reference** | | |
|---|---|---|---|
| **Prediction** | **HD** | **LD** | **MD** |
| **HD** | **23** | **4** | **4** |
| **LD** | **6** | **10** | **3** |
| **MD** | **0** | **0** | **0** |

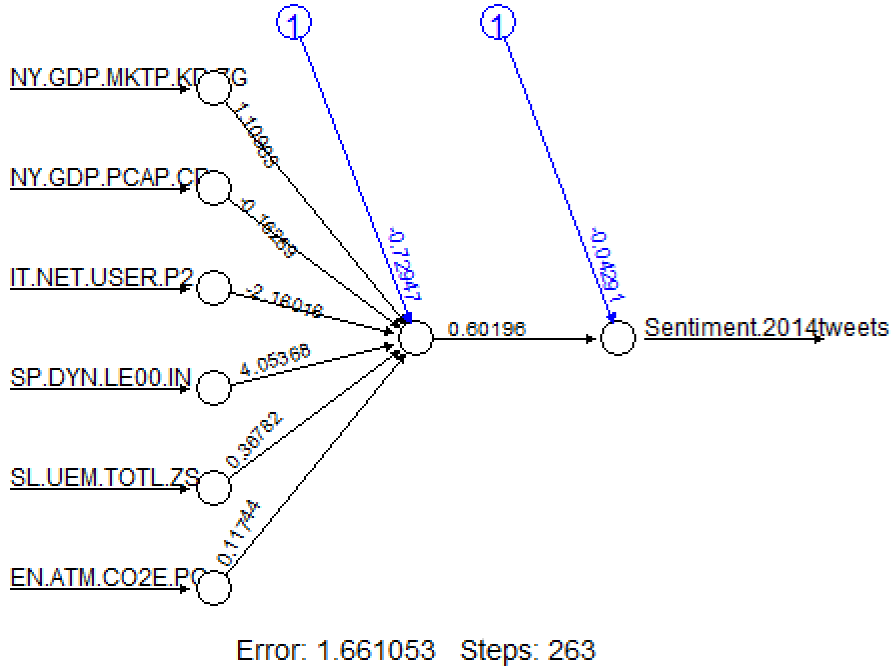
## Neural Networks

[Data Cleaning]
We were eager to use the dataset as thoroughly as possible, but the *neuralnet* function couldn't work apparently because of missing (NA) data. So we decided not to use some columns (attributes) and data (countries) which had many missing data, cut those columns and data with Excel, and prepared a new CSV file (bank2012_clean.csv) which has 155 data (reduced from 214), 9 columns (reduced from 23 columns) and no missing data.

[Analyses]
Commands were run as stated in NN.r. We separated the 155 data into Training set and Testing set in 7.5:2.5 ratio. We built a model to predict Tweets' sentiment (Sentiment.2014tweets) using normalized values of below 6 attributes:

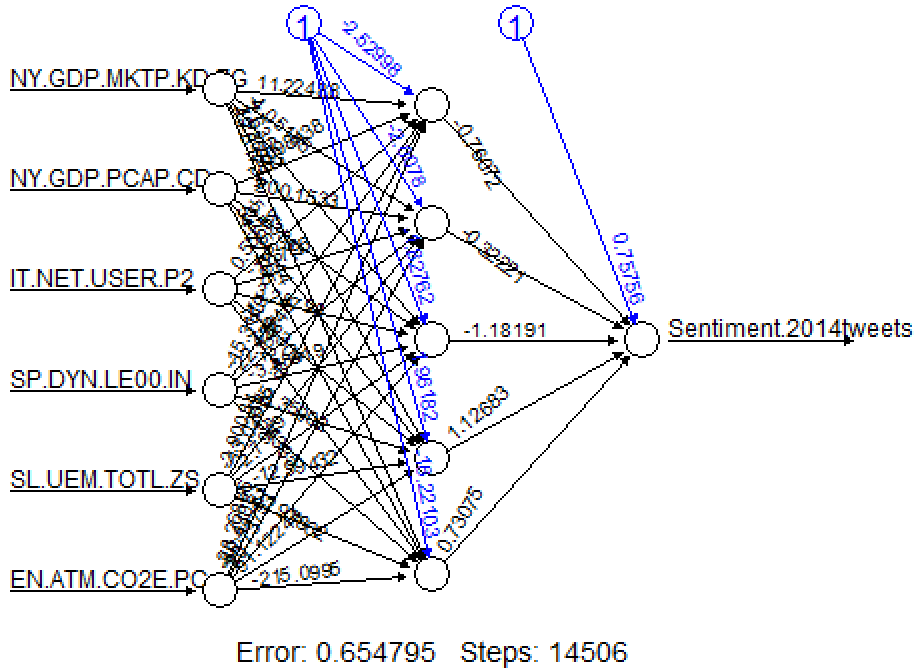
NY.GDP.MKTP.KD.ZG: GDP growth (annual %)
NY.GDP.PCAP.CD: GDP per capita (current US$)
IT.NET.USER.P2: Internet users (per 100 people)
SP.DYN.LE00.IN: Life expectancy at birth, total (years)
SL.UEM.TOTL.ZS: Unemployment, total (% of total labor force) (modeled ILO estimate)
EN.ATM.CO2E.PC: CO2 emissions (metric tons per capita)

Neural net model with a single hidden node built with Training set generated below topology:



We verified the accuracy of this model by using Testing set, and the predicted values and actual values showed correlation of **-0.002768198211**.

We tried to improve the model by adding five hidden nodes to it, and the below topology was generated:



This time, the correlation between predicted values and actual values in the Testing set was **0.01986744919** which showed some improvement from the previous model.
But since it's still a very weak correlation, it is implied that predicting tweet's sentiments by those indices is not quite realistic.

### Clustering

We tried to classify countries with tweets' sentiment and other indices used in Neural Network with k-means clustering. Here are result of when we tried to classify them in four clusters:

※ values are normalized

| | Cluster 1<br>(N = 48) | Cluster 2<br>(N = 12) | Cluster 3<br>(N = 44) | Cluster 4<br>(N = 51) |
|---|---|---|---|---|
| Tweets' sentiment | 0.4028757711 | 0.4420320250 | 0.4447136186 | 0.4945318040 |
| GDP Growth | 0.6253131536 | 0.5192513182 | 0.4330676490 | 0.5379635145 |
| GDP per capita | 0.01673819311 | 0.62533767142 | 0.26137975406 | 0.05230515851 |
| Internet users | 0.09445283519 | 0.79400091804 | 0.75413982406 | 0.37617806739 |
| Life expectancy | 0.3575154573 | 0.8775479066 | 0.8787444936 | 0.7266415768 |
| Unemployment | 0.2473958333 | 0.1186951744 | 0.3234898350 | 0.2518704847 |
| CO2 emission | 0.02445209119 | 0.49940150166 | 0.20019497070 | 0.07423105472 |
| Characteristics | Developing<br>Low internet<br>Low life expectancy<br>Low CO2 emission | Developed<br>Enough Internet<br>Long life<br>Low unemployment<br>High CO2 emission | | Under developed<br>Moderate internet<br>Low CO2 Emission |

We couldn't find difference in tweets' sentiment by classifying with other indices.

### <u>Comparative Analysis</u>

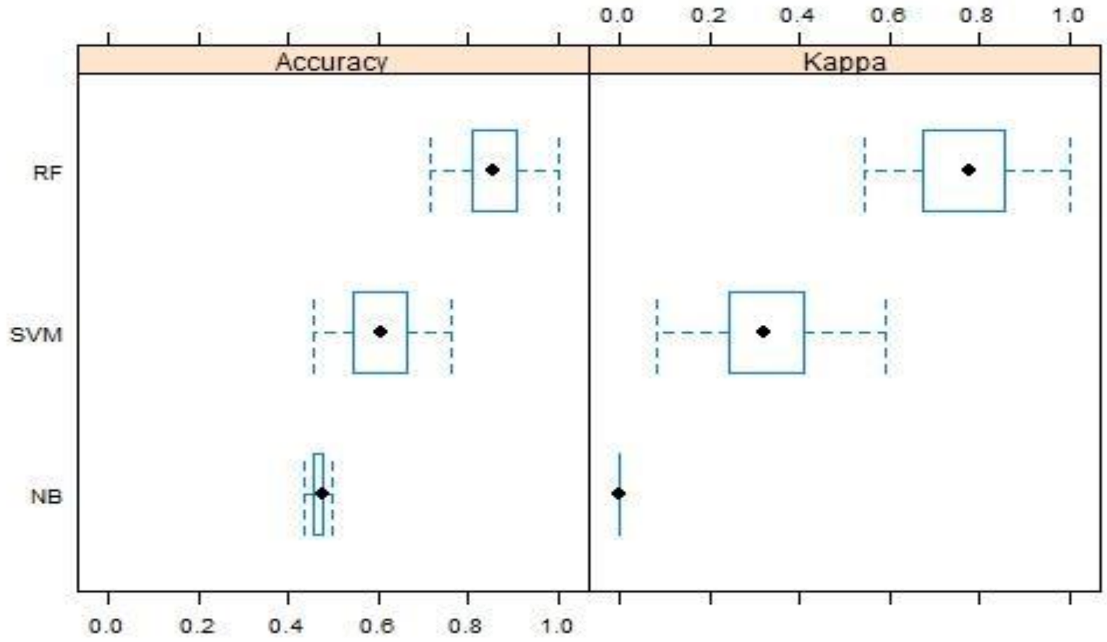
**Dataset Used: bank.csv**
**Rcode used: compana.R**
For Conducting Comparative analysis we used Caret and klaR packages. And by using these packages we compared our SVM, Naive Bayes and random Forest classifiers.

The call summary was as follows:
Call:

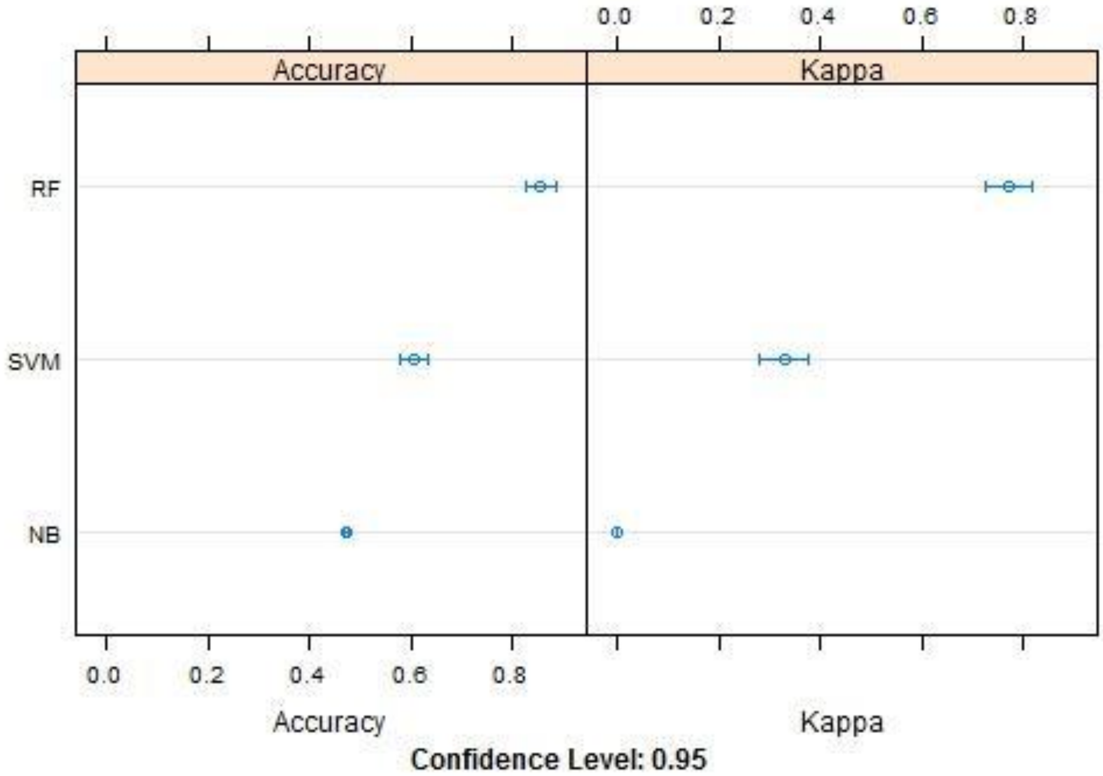
```
summary.resamples(object = result)

Models: NB, RF, SVM
Number of resamples: 30

Accuracy
      Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
NB  0.4348  0.4600 0.4762 0.4721  0.4762 0.5000    0
RF  0.7143  0.8095 0.8571 0.8567  0.9080 1.0000    0
SVM 0.4545  0.5519 0.6050 0.6061  0.6591 0.7619    0

Kappa
       Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
NB  0.00000  0.0000 0.0000 0.0000  0.0000 0.0000    0
RF  0.54350  0.6805 0.7778 0.7722  0.8532 1.0000    0
SVM 0.08333  0.2459 0.3220 0.3281  0.4055 0.5882    0
```

And as a result we have a grid for kapp and accuracy values for RF, NB and SVM models.

The Boxplot is as follows:



The dotplot output is as follows:



**Result: From both the outputs its evident that random forest model is leading the way. Hence it’ll be the best fit or our model. And the Kappa value is also highest for Random forest being approximately 0.8 which is a good agreement.**