# USE CASE STUDY REPORT

**Group No**.: Group 12

**Student Names**: Palash Pramod Nimbalwar and Siva Teja Reddy Seelam

## Executive Summary:

In this project, our aim is to classify the tweets posted by users as positive or negative as well predict the sentiment and score of any tweet which will enable a user in advance to avoid or read the tweet. The data i.e. tweets waere scraped completely from twitter by the scraper we made using Twitter API. We extracted the tweets and cleaned them using many techniques,gave them score using sentimentr package and then finally converted into a numerical format to feed to the data mining techniques which used to classify and predict.We used SVM-Support Vector Machine and Linear Regression for the same. The accuracy using both the techniques was pretty good and we would recommend to use SVM for sentiment analysis for prediction and classification.

## I. Background and Introduction

In this project, we did the sentiment analysis of tweets of users regarding different topics. Sentiment analysis is a part of NLP, **our aim is to classify the tweets posted by users as positive or negative as well predict the sentiment and score of any tweet**

***NLP:***

Natural Language Processing is the field of study that focuses on the interactions between human language and computers. It is at the intersection of computer science, artificial intelligence, and computational linguistics.

***Twitter:***

Twitter is an online news and social networking site where people communicate in short messages called tweets. Tweeting is posting short messages for anyone who follows you on Twitter, with the hope that your messages are useful and interesting to someone in your audience. Another description of Twitter and tweeting might be microblogging.Some people use Twitter to discover interesting people and companies online, opting to follow their tweets.

***What is Sentiment Analysis?***

Sentiment analysis is the automated process of understanding an opinion about a given subject from written or spoken language.

In a world where we generate 2.5 quintillion bytes of data every day, sentiment analysis has become a key tool for making sense of that data. This has allowed companies to get key insights and automate all kinds of processes.

***The Problem:***

The biggest problem today on social media is about the negativity spread by users for other people. This is a very serious issue and affects people very heftily. A lot of people feel depressed after reading negative posts about them, which can make them take extreme steps.

***Our Goal:***

Our aim is to understand the sentiment of users– what people are saying, how they're saying it, and what they mean. We predicted the sentiment of any tweet with the help of score. Sentiment Analysis is the domain of understanding these emotions and use that for different topics.

***The Solution:***

We used natural language processing, statistics, and text analysis to extract, and identify the sentiment of text into positive, negative, or neutral categories and understood what people were saying and in future we can build a model which can sort out the negativity from any person's social media site, which will enable them to stay away from all the negativity.

We first collected the tweets from twitter. With the collected tweets, we did sentiment analysis on data and classified tweets as positive, negative and neutral using various NLP techniques and compare those results.We trained the model using some tweets and their scores and then used different data mining techniques to predict the score of tweets. The general idea was to calculate a sentiment score for each tweet and thereby find out how positive or negative the tweet is .

We used the classical approaches in data mining like Naive Bayes,SVM and KNN, sentence subjectivity, sentiment classification and compared those results with applications of neural networks in NLP. The steps for the project are:

1) Data Preprocessing
2) Reduce the dimensions if required
3) Exploratory Data Analysis
4) Data Mining using Algorithms and interpret them
5) Pick the best model
6) Deployment

## II. Data Exploration and Visualization

After gathering the tweets, we still could not use it directly. We need to clean it properly. There were many things such as links,hashtags,@names and punctuations. This is the snap before cleaning.

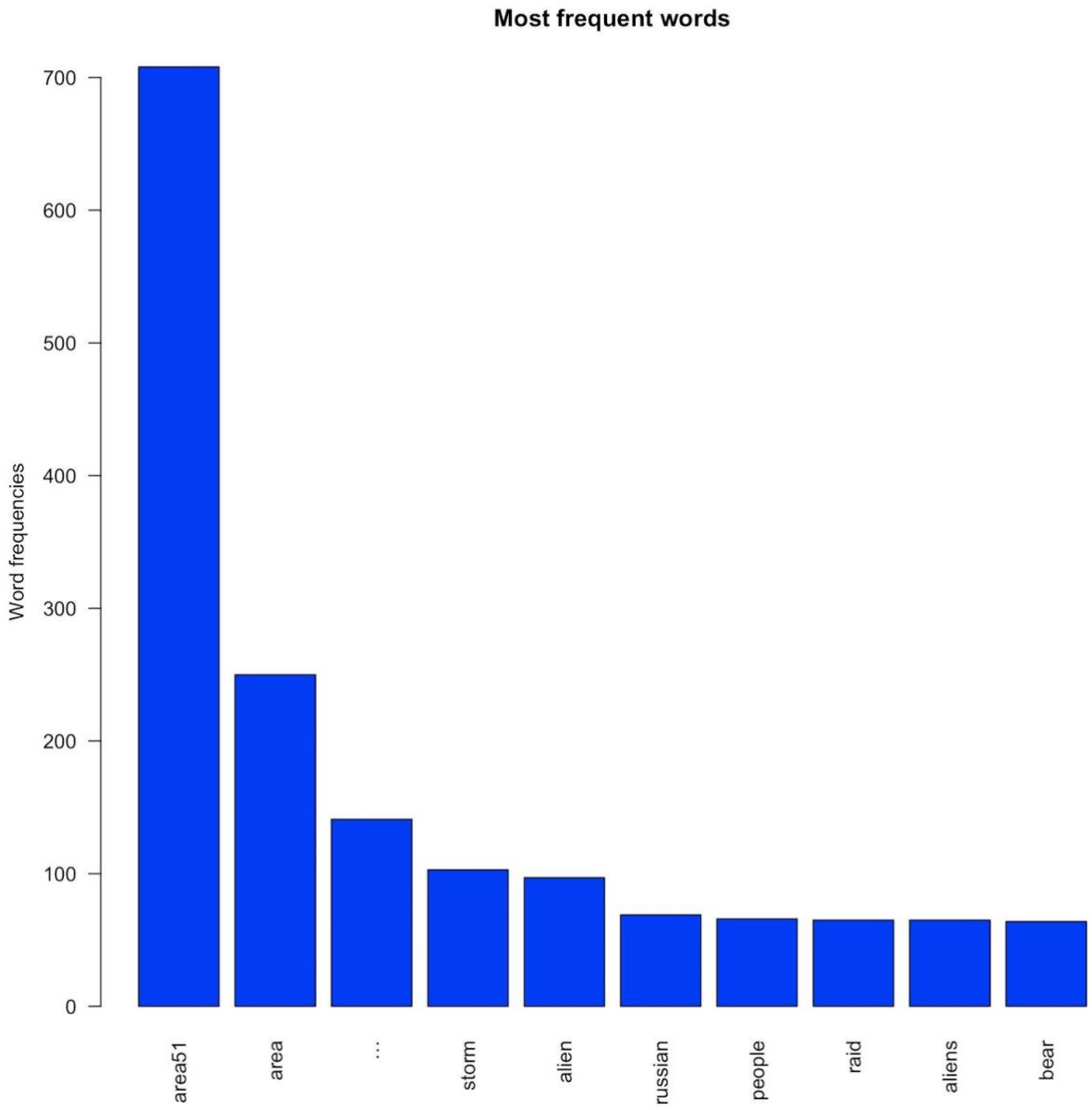
```
> head(Tweets.df$text)
[1] "RT @TrippieEd: Me listening to alien music after I get them out of Area 51 #Area51 https://t.co/0AB9TtSg7W"
[2] "RT @hirnblond: So amazed by the new design of @jccaylen 's new tixxo collection ( @twitchjccaylen #tixxo #area51 ) https://t.co/88MuDyf8Fv"
[3] "RT @GiftofFailure: FREE today. Good deals for today. https://t.co/mFOpFZ10BN #freebook #kindle #KindleUnlimited #freebooks #amreading #scie…"
[4] "Yeah for sure! Americans cannot even go everywhere in their one country! What do you think about #Area51 #Trump ? F… https://t.co/6xtwHLqRl0"
[5] "Me and my alien filming tik toks after i picked him from #Area51 https://t.co/VWr4mWplaN"
[6] "You come watch the stream, its lit in here #Youtube #Smallstreamer #Twitch #ApexLegends #Area51 live at https://t.co/yJAe8MUMbs"
```

This is the snap after cleaning:

```
[1] "Me listening to alien music after I get them out of Area"
[2] "So amazed by the new design of s new tixxo collection"
[3] "FREE today Good deals for today"
[4] "Yeah for sure Americans cannot even go everywhere in their one country What do you think about F"
[5] "Me and my alien filming tik toks after i picked him from"
[6] "You come watch the stream its lit in here live at"
```

To know the frequencies of the words in tweets, we go the frequency table and plot for it.

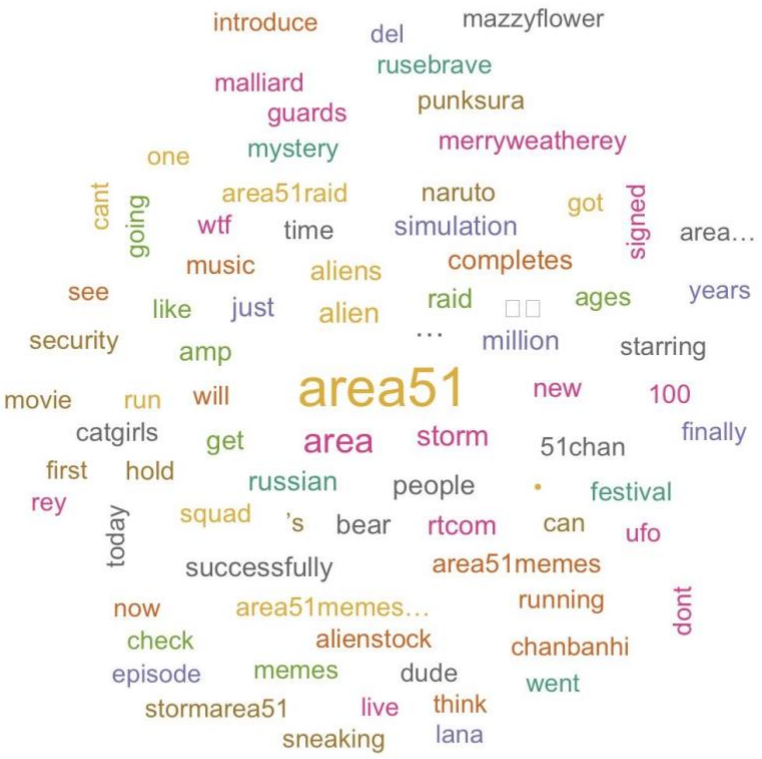
| | word | freq |
|---|---|---|
| area51 | area51 | 708 |
| area | area | 250 |
| … | … | 141 |
| storm | storm | 103 |
| alien | alien | 97 |
| russian | russian | 69 |
| people | people | 66 |
| raid | raid | 65 |
| aliens | aliens | 65 |
| bear | bear | 64 |




The order of words is completely random but the size of the words are directly proportional to the frequency of occurrence of the words in tweets.The diagram directly helps us identify the most frequently used words in tweets.

Also, we got the word cloud which is a text mining method that allowed us to highlight the most frequently used keywords in a paragraph of text.It is a visual representation showing the most

relevant words.We visualized tweets as a word cloud to find out what people are tweeting about the **area51** .




## III. Data Preparation and Preprocessing

Most of the data mining techniques require the data in numerical format to work. So, we converted our data into numerical format and here is a snap of it:

| | alien | area | get | music | new | today | even | one | think | come | live | watch | last | outside | time | amp | like | link | run | alienstock | event | festiva |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This shows the word count of the words in a particular tweet and then we can use it for classification.

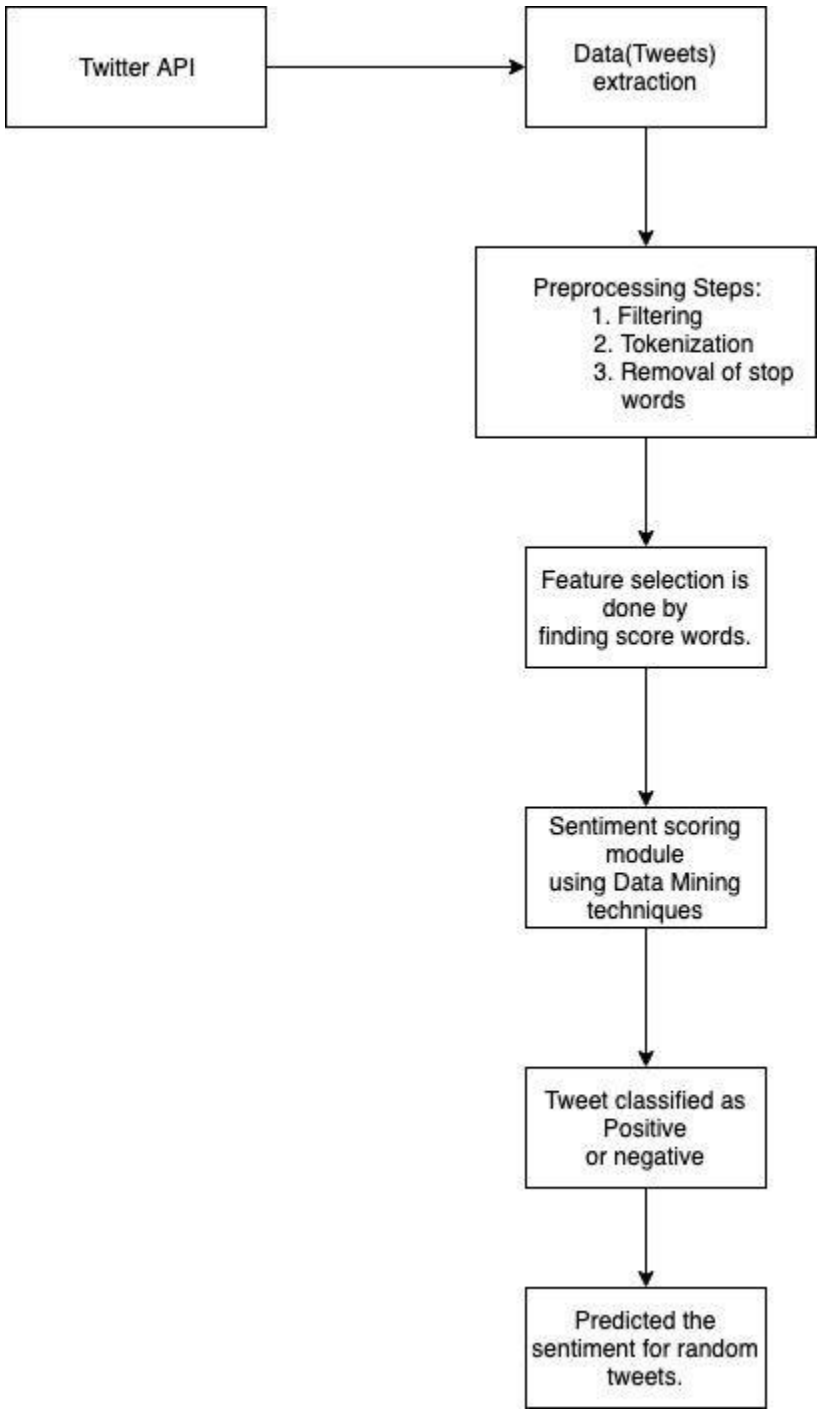
After the cleaning and processing, we divided the data into training and testing sets. Then we derived the scores for the training sets using sentimentr package based on some conditions where we classified the tweets as positive and negative . Here is the snap of the scores we got :

| | element_id | word_count | sd | ave_sentiment |
|---|---|---|---|---|
| **1** | 1 | 12 | *NA* | 0.000000e+00 |
| **2** | 2 | 11 | *NA* | 6.331738e-01 |
| **3** | 3 | 6 | *NA* | 5.103104e-01 |
| **4** | 4 | 18 | *NA* | 0.000000e+00 |
| **5** | 5 | 12 | *NA* | -1.732051e-01 |
| **6** | 6 | 11 | *NA* | 0.000000e+00 |
| **7** | 7 | 11 | *NA* | 0.000000e+00 |
| **8** | 8 | 14 | *NA* | -2.886421e-01 |
| **9** | 9 | 10 | *NA* | 1.581139e-01 |
| **10** | 10 | 16 | *NA* | -1.250000e-01 |
| **11** | 11 | 5 | *NA* | -2.236068e-01 |
| **12** | 12 | 24 | *NA* | -2.245366e-01 |
| **13** | 13 | 11 | *NA* | -5.879471e-01 |
| **14** | 14 | 12 | *NA* | 1.587713e-01 |
| **15** | 15 | 21 | *NA* | -1.636634e-01 |
| **16** | 16 | 10 | *NA* | 0.000000e+00 |
| **17** | 17 | 4 | *NA* | 0.000000e+00 |
| **18** | 18 | 4 | *NA* | 0.000000e+00 |
| **19** | 19 | 4 | *NA* | 4.000000e-01 |

## IV. Data Mining Techniques and Implementation

***We did classification as well as prediction using SVM and Linear Regression.***

This is the flow of our project:




**Step 1: Twitter API**

First step was to perform create a twitter application. This application allowed us to perform a connection from our R console to the twitter using the Twitter API.

**Step 2: Data Extraction**

The second step was to get the data i.e the tweets. This was the most important part because we had to build our own scraper which extracted data from Twitter. Instead of using data sets, we scraped tweets from twitter api using our own keys and authorization.

**Step 3: Preprocessing Steps**

We already explained the preprocessing steps above.

**Step 4: Feature selection**

We explained this step also in the earlier topic.

**Step 5: Data Mining Techniques**

We used ***SVM and Linear regression*** for this problem. Our aim was first just to classify the tweets as positive and negative but then after the project progressed we thought of predicting the sentiment for the tweets and we successfully did that using the algorithms. In short, We ***classified as well as predicted*** the sentiments for the tweets.

***Linear Regression:***

LR allocates weight parameter, theta for each of the training features. The predicted output(h(θ)) will be a linear function of features and θ coefficients.

$$h_\theta = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots$$

During the start of training, each theta is randomly initialized. But during the training, we correct the theta corresponding to each feature such that, the loss (metric of the deviation between expected and predicted output) is minimized.

We used linear regression to predict scores, 30% data cross validation, 20% test data and 50% training data was partitioned.

Prediction of sentiment score using Linear regression:

| | original_tweets | sentiment | predicted_sentiment_score |
|---|---|---|---|
| 1 | The saga continues for the raid concert The residents now s... | -0.4944333 | -0.004793174 |
| 2 | These munchkins are ready to storm Now all they need are ... | 0.1333946 | 0.133394594 |
| 3 | Lilly seriously i never noticed that before | 0.0000000 | 0.020824185 |
| 4 | the internet is outtapocket in i have fallen victim to all thr... | -0.3274231 | 0.020824185 |
| 5 | So anyone packed survival gear and loaded up for the Area ... | 0.1279204 | 0.029505178 |
| 6 | A patron just destroyed Earth to their child as the planet we... | -0.1632993 | -0.138991122 |
| 7 | A patron just destroyed Earth to their child as the planet we... | -0.1146829 | -0.138991122 |
| 8 | the internet is outtapocket in i have fallen victim to all thr... | -0.3485685 | 0.020824185 |
| 9 | These munchkins are ready to storm Now all they need are ... | 0.1333946 | 0.133394594 |
| 10 | For real | 0.0000000 | 0.035908416 |
| 11 | Cant we just talk | -0.2000000 | -0.200000000 |

***SVM:***

A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

We used SVM to predict sentiments of the tweets. Our data was partitioned into 30% cross validation, 20% test data and 50% training data.

Prediction of sentiment using SVM:

| | original_tweets | label | predicted_sentiment |
|---|---|---|---|
| 350 | The saga continues for the raid concert The residents now s... | negative | positive |
| 351 | These munchkins are ready to storm Now all they need are ... | positive | negative |
| 352 | Lilly seriously i never noticed that before | positive | positive |
| 353 | the internet is outtapocket in i have fallen victim to all thr... | negative | positive |
| 354 | So anyone packed survival gear and loaded up for the Area ... | positive | positive |
| 355 | A patron just destroyed Earth to their child as the planet we... | negative | positive |
| 356 | A patron just destroyed Earth to their child as the planet we... | negative | positive |
| 357 | the internet is outtapocket in i have fallen victim to all thr... | negative | positive |
| 358 | These munchkins are ready to storm Now all they need are ... | positive | negative |
| 359 | For real | positive | positive |
| 360 | Cant we just talk | negative | positive |

**Step 6: Classified and Predicted**

In the final step, using algorithms, we classified the tweets as positive and negative and predicted the sentiments of random tweets .

## V. Performance Evaluation

**SVM :**

Below is the confusion matrix for cross validation:

| | Actual | |
|---|---|---|
| Predictions | negative | positive |
| negative | 14 | 6 |
| positive | 21 | 90 |

Below is the confusion matrix for test:

| | Actual | |
|---|---|---|
| Predictions | negative | positive |
| negative | 22 | 13 |
| positive | 40 | 122 |

The **accuracy** we get using SVM is : **0.7309645**

**Linear Regression:**

linear regression training **data residual error**:- **0.2212** linear regression validation **data residual error**:- **0.2229** linear regression test **data residual error**:- **0.2082**

The **accuracy** we got for Linear Regression is : **0.56437**

## VI. Discussion and Recommendation

This project was implemented using SVM and Linear Regression, we learned a lot about both the techniques.

Both the techniques were good but gave results for positive sentiment properly. The data set was small, if the data set would have been large , the results could have been better. We achieved the almost 73% accuracy for prediction which is pretty good. Extracting and cleaning were also very important steps . The scraper we built can be made for real time tweets and classification of sentiments. Also, we could try different NLP,Data Mining techniques for classification and prediction.

This project can be carried forward and can be made a real time tweet classification project which could help a lot to many people to avoid negativity on the internet.

## VII. Summary

We were successful in classifying the tweets posted by users as positive or negative as well predict the sentiment of any tweet using SVM (73% accuracy) and Linear Regression(56% accuracy). Tweets were extracted by twitter and cleaned them using many techniques,we gave them score using sentimentr package and then finally converted into a numerical format to feed to the data mining techniques which used to classify and predict.We used SVM-Support Vector Machine and Linear Regression for the same. The accuracy using both the techniques was pretty good and we would recommend to use SVM for sentiment analysis for prediction and classification.

The flowchart shown depicts the flow of the project. This project helped us learn a lot about classification and prediction using data mining techniques. Different performance measures were tested for both techniques and found that SVM was better.

We predicted sentiment as well as score for the tweets, which could give insight about tweets before even reading them and can be turned into a major project.

## References:


1) **Twitter: https://twitter.com**
2) **R: https://www.r-project.org/**
3) **Lifewire: https://www.lifewire.com/what-exactly-is-twitter-2483331**
4) **Github: https://github.com/**
5) **Kaggle: https://www.kaggle.com/**

6) **MonkeyLearn:** **https://monkeylearn.com/sentiment-analysis/** 7) **Towards Data Science:** **https://towardsdatascience.com/comparative-study-on-classic-machine-learning-alg orithms-24f9ff6ab222**
8) **IE7275 Course Material**

## Creative Contribution:

The contribution of both the members was equal and we learned a lot during this project.

## Appendix: R Code for use case study

```
library("RColorBrewer")
library("wordcloud")
library("NLP") library(tm)
library(twitteR)
library(ROAuth)
library(plyr)
library(stringr)
library(base64enc)
library(e1071)
library(SnowballC)
library(caret)

consumerKey<-"Z8lh3fQ0WmaXPajr4vN3wzsaF"
consumerSecret<-"Q1sjgjpzC9EOWFc9KIxp51vHR9zTU3ublItoECM8v2tyUIyCKa"
accessToken<-"1101711618755444736-ZJQYY3iEiT6ma22FQOkEX9xoyExxFT"
accessTokenSecret<-"JHBgc0eTA96xex8Hr6qj0kTw35GtVWZylfCSRd3ZcF4HB"
setup_twitter_oauth(consumerKey,consumerSecret,accessToken,accessTokenSecret)

Tweets=searchTwitter("#Area51",n=1000,lang="en")
Tweets.df<-twListToDF(Tweets)
head(Tweets.df$text) tweet_text=Tweets.df$text tweet_text =
gsub("(RT|via)((?:\\b\\W*@\\w+)+)", " ", tweet_text) tweet_text=
gsub("(f|ht)(tp)(s?)(://)(.*)[.|/](.*)", " ", tweet_text)
tweet_text = gsub("#\\w+", " ", tweet_text) tweet_text =
gsub("@\\w+", " ", tweet_text) tweet_text =
gsub("[[:punct:]]", " ", tweet_text) tweet_text =
gsub("[[:digit:]]", " ", tweet_text) tweet_text = gsub("[
\t]{2,}", " ", tweet_text) tweet_text = gsub("^\\s+|\\s+$",
"", tweet_text) tweet_text = gsub("\n","",tweet_text)
tweet_text = iconv(tweet_text,"UTF-8","ASCII",sub="")
head(tweet_text)
tweet_text = tweet_text[!is.na(tweet_text)]
names(tweet_text)=NULL tweet_text=unique(tweet_text)
library(sentimentr) tweet_text<-get_sentences(tweet_text)
score<-sentiment(tweet_text)
```

```
tweet_text_Corpus<- Corpus(VectorSource(tweet_text)) tweet_text_Corpus<-
tm_map(tweet_text_Corpus, removePunctuation) tweet_text_Corpus<-
tm_map(tweet_text_Corpus, content_transformer(tolower)) tweet_text_Corpus<-
tm_map(tweet_text_Corpus, removeWords, stopwords("english")) tweet_text_Corpus<-
tm_map(tweet_text_Corpus, stripWhitespace) tweet_text_Dtm<-
DocumentTermMatrix(tweet_text_Corpus,control = list(weighting= function(x)
weightBin(x))) tweet_text_Dtm<-removeSparseTerms(tweet_text_Dtm,.99) scr<-
score[,"sentiment"] m<-median(scr[["sentiment"]]) for (i in
1:length(scr[["sentiment"]])){   if (scr[i,"sentiment"]>=0.0){     scr[i,"label"]="positive"
 }else{
   scr[i,"label"]="negative"
 } }
length(scr[["label"]]) dtm.train <-
tweet_text_Dtm[1:220,] dtm.cross.val <-
tweet_text_Dtm[220:350,] dtm.test <-
tweet_text_Dtm[350:546,] y.train<-
scr[1:220,"label"]
y.cross.val<-scr[220:350,"label"]
y.test<-scr[350:546,"label"]
x.train <- as.matrix(dtm.train)
x.val<-as.matrix(dtm.cross.val)
x.test<-as.matrix(dtm.test) training_data <- as.data.frame(cbind(y.train,x.train))
cross_val_data <- as.data.frame(x.val) test_data <- as.data.frame(x.test) sv <-
svm(label~., training_data, type="C-classification", kernel="linear", cost=1)
prediction1<-predict(sv,test_data)
c.f<-table("Predictions"= prediction1,  "Actual" = y.test[["label"]] )
TP = c.f[1,1] + c.f[2,2];  # true predictions
total = nrow(test_data);  # total predictions acc = TP / total
ypred.train<-scr[1:220,"sentiment"] ypred.cross.val<-
scr[220:350,"sentiment"] ypred.test<-
scr[350:546,"sentiment"] training_pred_data<-
as.data.frame(cbind(ypred.train,x.train))
linear<-lm(sentiment~., training_pred_data)   summary(linear) cross_val_pred_data<-
as.data.frame(cbind(ypred.cross.val,x.val)) linear_val<-lm(sentiment~., cross_val_pred_data)
summary(linear_val) test_pred_data<- as.data.frame(cbind(ypred.test,x.test)) linear_test<-
lm(sentiment~., test_pred_data) summary(linear_test)
twt=c(tweet_text[[350]],tweet_text[[351]],tweet_text[[352]],tweet_text[[353]],
tweet_text[[354]],tweet_text[[355]],tweet_text[[356]],tweet_text[[357]],
tweet_text[[358]],tweet_text[[359]],tweet_text[[360]]) label=scr[350:360,"label"]
prd=prediction1[1:11]
output=data.frame(original_tweets=twt,original_sentiment=label,predicted_sentiment=prd,string
sAsFactors = FALSE) score1=scr[350:360,"sentiment"]
```

```
pred_scr=linear_test[["fitted.values"]][1:11]
output_reg=data.frame(original_tweets=twt,original_sentiment_score=score1,predicted_sentime
nt_score=pred_scr,stringsAsFactors = FALSE) summary(linear_test)
error_per=abs((linear_test[["residuals"]]/scr[350:546,"sentiment"]))*100 mean(error_per)
#for the word cloud and frequency table:
wordll.packages("SnowballC") library(wordcloud) library(SnowballC)
library(tm) word<- searchTwitter('#area51', n=1000, lang='en') word word_text
<- sapply(word, function(x) x$getText())       word_text_corpus <-
Corpus(VectorSource(word_text)) word_text_corpus <-
tm_map(word_text_corpus, removePunctuation) word_text_corpus <-
tm_map(word_text_corpus, content_transformer(tolower)) word_text_corpus <-
tm_map(word_text_corpus, function(x)removeWords(x,stopwords()))
word_text_corpus <- tm_map(word_text_corpus, removeWords, c('RT',
'are','that')) removeURL <- function(x) gsub('http[[:alnum:]]*','', x)
word_text_corpus <- tm_map(word_text_corpus,
content_transformer(removeURL)) word_2 <-
TermDocumentMatrix(word_text_corpus) word_2 <- as.matrix(word_2)
word_2 <- sort(rowSums(word_2),decreasing=TRUE) word_2 <-
data.frame(word = names(word_2),freq=word_2) head(word_2, 10)
barplot(word_2[1:10,]$freq, las = 2, names.arg = word_2[1:10,]$word,
        col ='blue', main ='Most frequent words',
ylab = 'Word frequencies')
set.seed(1234)

wordcloud(word_text_corpus,min.freq=1,max.words=80,scale=c(2.2,1),
colors=brewer.pal(8,"Dark2"),random.color=T,random.order=F)
```