

TWITTER'ECON

LIVE SENTIMENTAL ANALYSIS

INTRODUCTION

The digital world we live in is brimmed up with data, the one who can derive information from it is the winner. Organizations are looking for ways to interpret information from the enormous data they are piling day by day from the consumer space in the internet. One such brilliant service available to us is GOOGLE TRENDS[1]. This is an analytical tool which gives in depth analytic report of the trends in of a particular keyword. Taking inspiration from it, I designed and developed TWITTER'ECON. This is a python based tool to get the trends of a keyword in real-time by exploiting the services provided by Twitter API[2].

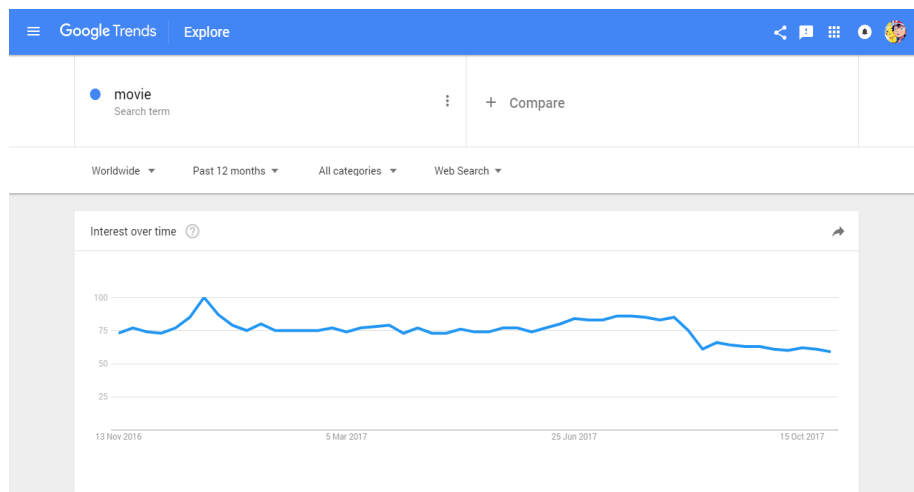


Fig.1 Google trends analytic report for keyword- 'movie'

FUNCTIONALITIES

The tool provides basic sentimental analysis report of the then retrieved live tweets. The tool takes a keyword from the user as an input and then feeds the input to a sentimental analysis module. This module makes use of a voting process by multiple pickled classifiers

and returns the sentiment along with the confidence. This is plotted as a graph against time like in fig.2.

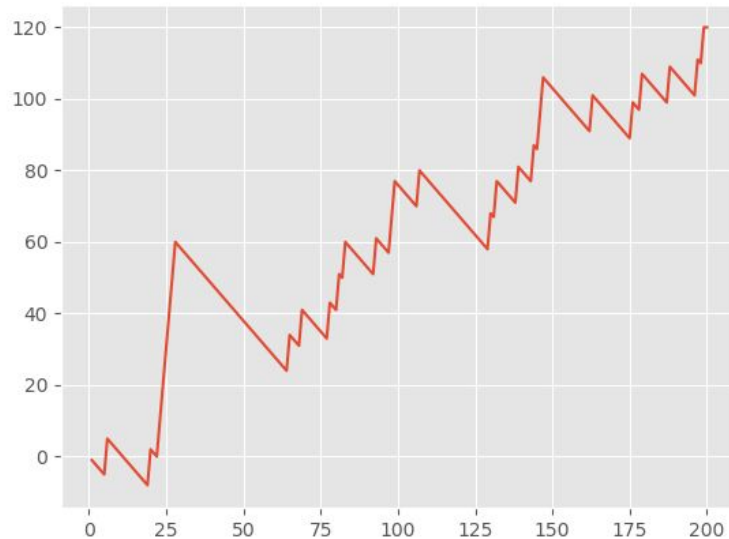


Fig.2 Twitter'econ analytic report for keyword- 'movie'

TECHNOLOGY STACK

USER INTERFACE : Tkinter [3]

PACKAGES : Scikit learn, Matplotlib, Tweepy, Statistics, Pickle

CLASSIFIERS : MultinomialNB, BernoulliNB, LogisticRegression, SGDClassifier, SVC, LinearSVC, NuSVC

CLASSIFIER PERFORMANCE

Accuracies of classifiers used over 700 features for training and 300 for testing.

Original Naive Bayes Algo accuracy percent:', 54.998070243149364

MNB_classifier accuracy percent:', 54.824392126592045

BernoulliNB_classifier accuracy percent:', 54.727904284060216

LogisticRegression_classifier accuracy percent:', 54.95947510613662

LinearSVC_classifier accuracy percent:', 55.08490930142802

SGDClassifier accuracy percent:', 53.37707448861444

Most Informative Features

feature = True	pos : neg	=	2.4 : 1.0
screen = True	neg : pos	=	2.3 : 1.0
teen = True	neg : pos	=	2.3 : 1.0
best = True	pos : neg	=	1.9 : 1.0
more = True	neg : pos	=	1.8 : 1.0
formula = True	neg : pos	=	1.7 : 1.0
whole = True	neg : pos	=	1.7 : 1.0
inside = True	pos : neg	=	1.7 : 1.0
effective = True	pos : neg	=	1.7 : 1.0
witty = True	pos : neg	=	1.7 : 1.0
small = True	pos : neg	=	1.7 : 1.0
huge = True	pos : neg	=	1.7 : 1.0
much = True	neg : pos	=	1.7 : 1.0
right = True	pos : neg	=	1.6 : 1.0
last = True	neg : pos	=	1.6 : 1.0

Accuracies of classifiers used over 800 features for training and 200 for testing.

Original Naive Bayes Algo accuracy percent: 55.5045871559633
MNB_classifier accuracy percent:', 55.542813455657495
BernoulliNB_classifier accuracy percent:', 55.56192660550459
LogisticRegression_classifier accuracy percent:', 55.695718654434245
LinearSVC_classifier accuracy percent:', 55.877293577981646
SGDClassifier accuracy percent:', 55.026758409785934

Most Informative Features

best = True	pos : neg	=	6.6 : 1.0
love = True	pos : neg	=	3.9 : 1.0
keeps = True	neg : pos	=	3.6 : 1.0
effective = True	pos : neg	=	3.0 : 1.0
right = True	pos : neg	=	3.0 : 1.0
debut = True	neg : pos	=	3.0 : 1.0
possible = True	pos : neg	=	2.3 : 1.0
past = True	pos : neg	=	2.3 : 1.0
working = True	pos : neg	=	2.3 : 1.0
dark = True	pos : neg	=	2.3 : 1.0
happy = True	pos : neg	=	2.3 : 1.0
damned = True	neg : pos	=	2.3 : 1.0
cinematic = True	neg : pos	=	2.3 : 1.0
much = True	neg : pos	=	2.0 : 1.0
future = True	pos : neg	=	1.7 : 1.0

There is a clear but very little difference between the two ways of splitting the training set. 800 training sets helped understand the tweets better.

Many critical concepts of the project are build taking SentDex Youtube tutorial series and pythonprogramming.net as a guide [4].

REFERENCES

1. Google Trends. (2017). Google Trends. [online] Available at: <https://trends.google.co.in/trends/explore?q=movie> [Accessed 8 Nov. 2017].
2. Developer.twitter.com. (2017). Twitter Developer Platform. [online] Available at: <https://developer.twitter.com/> [Accessed 8 Nov. 2017].
3. Wiki.python.org. (2017). TkInter - Python Wiki. [online] Available at: <https://wiki.python.org/moin/TkInter> [Accessed 8 Nov. 2017].
4. Pythonprogramming.net. (2017). Python Programming Tutorials. [online] Available at: <https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/> [Accessed 8 Nov. 2017].