Contant

Introduction

Sentiment analysis is a method of determining whether a text has a positive or negative feeling. One of the most interesting NLP applications is this. By analyzing the words in a sentence, you can use NLP (Natural Language Processing) to determine whether positive or negative comments were made by the people in the given data set. And from here comes our special idea: to analyze the tweets of giant companies such as FIFA, Facebook, and Google, and analyze whether they are positive or negative.

Related Work

Many studies and experiments have been done in analyzing drug reviews using different predictive algorithms. There have been experiments using Logistic Regression (70%), Random Forest (72%), and KNN (63%).

Main Point

In this project, we are trying to build models that are able to capture review texts as features, analyze them, and predict whether they are positive or negative reviews for corporate tweets. Several predictive algorithms are used, and a comparison is made between them. We also try to take into account not only textual data, but also numerical and categorical data, such as the tweet's number, the company's name, and the tweet's content.

Experiments

1. Data Exploration :

We looked at the data. The data consists of two sets of data, namely the training dataset and the test dataset. The data for training contains 4 columns, they are [**Tweet_UniqueID , Entity , Sentiment , Tweet_content**], and 74,682 rows. Also, the data for the test contains 4 columns and 1000 rows.

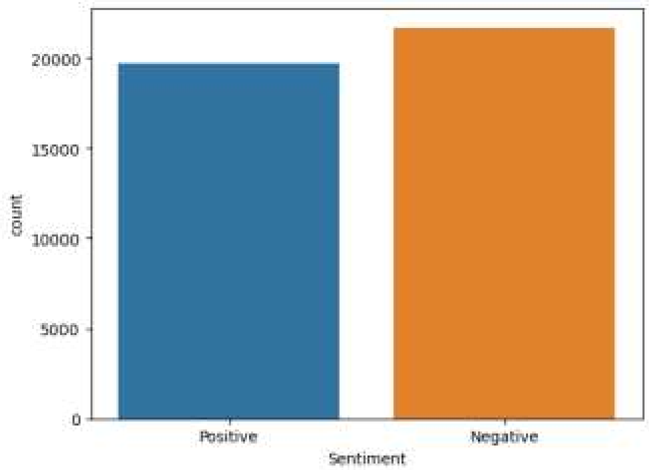
Column Names	*The Definition*
Tweet_UniqueID	The unique ID of each tweet
Entity	The names of the companies that have tweet data
Sentiment	The status of tweets if they are positive or negative
Tweet_content	Text content for each tweet

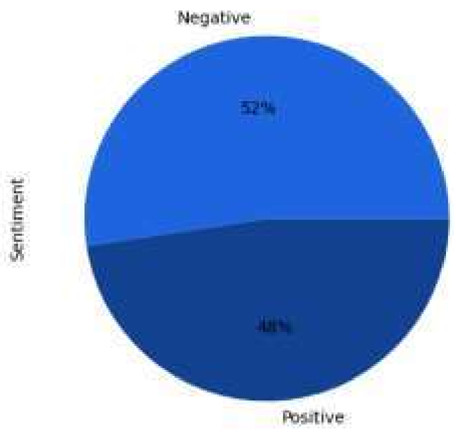
We have in the training data some of the duplicate rows and it was 2700 rows and they have been removed. In the row [Tweet_content] there were some empty cells (missing value) that were 326 and they have been removed. Neutral and Irrelevant have been removed.

.

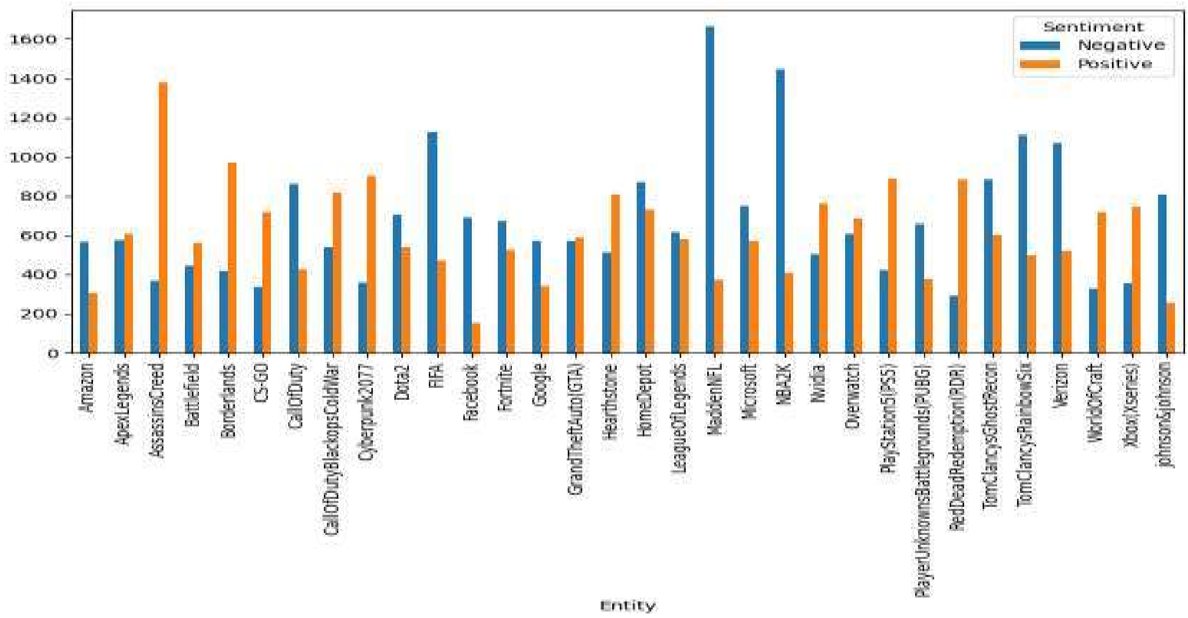
2. Data Visualization :

A plot was made showing the amount and percentage of the data in terms of positive or negative





A plot was made showing each company that issued how many tweets were positive and how many were negative



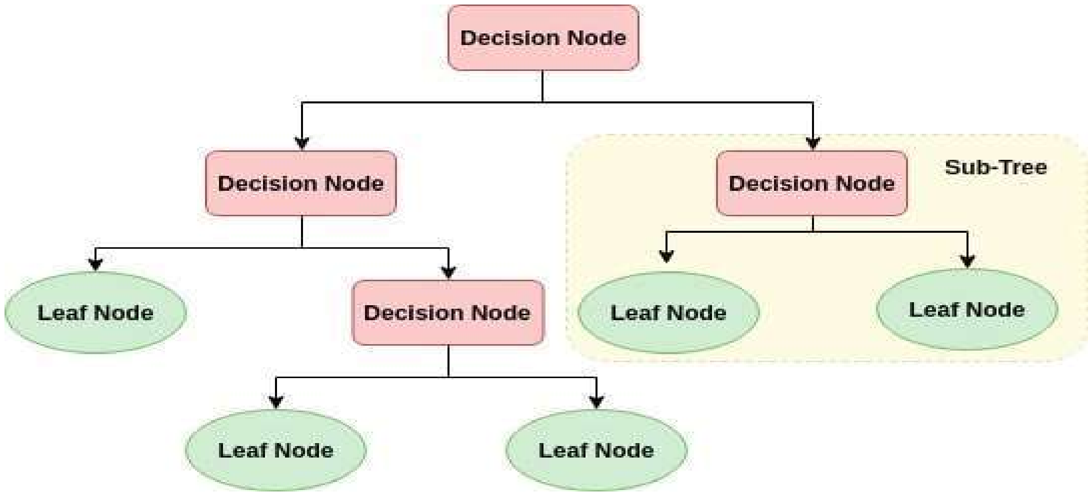
3. Data preprocessing

Preprocessing was done for the text content of the tweet in terms of making all letters lower case and any link was removed and a dictionary of abbreviations was created and also for the short word any html tags were removed and the data was cleaned from the special character and all of the ewlines, tabs, and carriage were removed returns, multiple space characters, stopwords, data stemming and tokenizer.

Model Building and Evaluation

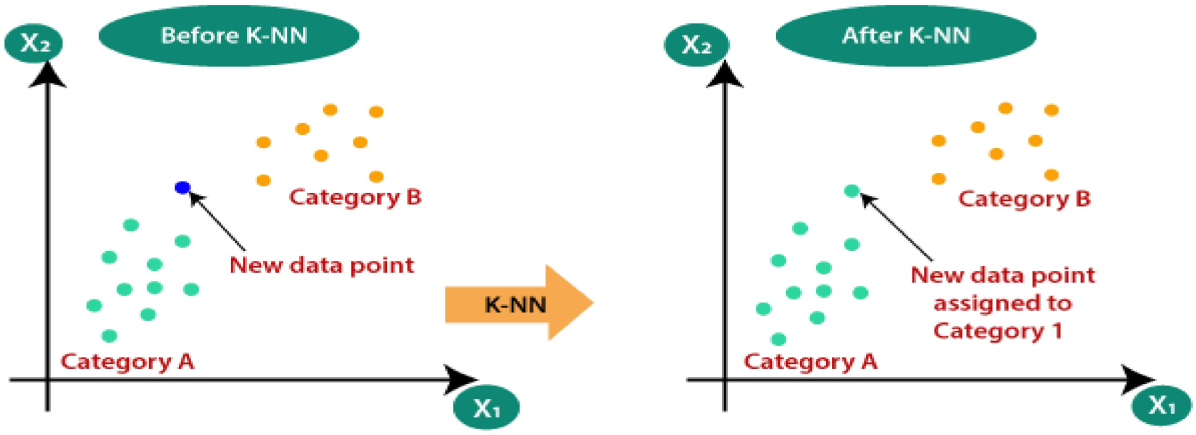
1. Decision Tree Classifier

It is a supervised learning algorithm. In this algorithm, data are continuously split into smaller parts until it reaches its class. It uses the terminologies like nodes, edges, and leaf nodes. In the Decision Tree classifier, first we compute the entropy of our database. It tells us the amount of uncertainty of our database. The smaller the uncertainty value, the better is the classification results. Each feature's information gain is calculated. This then tells us how much uncertainty reduces after spitting the database. Finally, all the information gain is calculated for all features, and now, we split the database which has high information gain. The process repeats until all nodes are cleared.



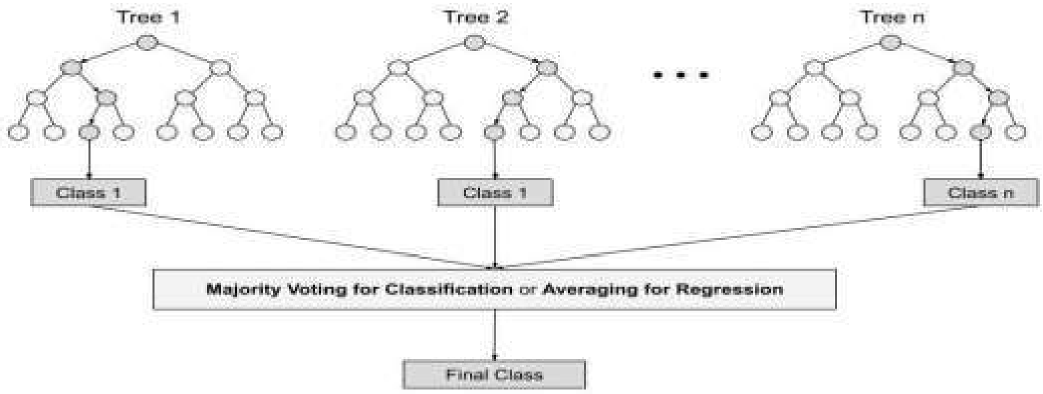
2. K-Nearest Neighbour

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.



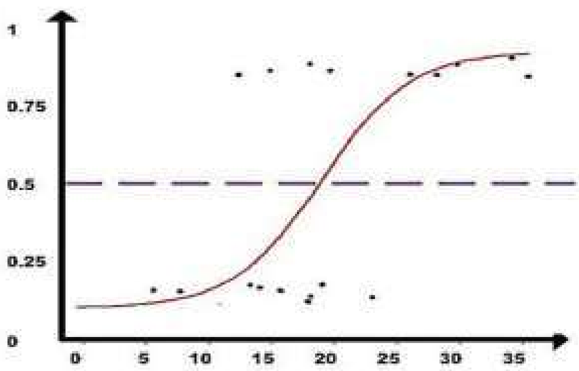
3. Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.



4. Logistic Regression

Machine learning generally involves predicting a quantitative outcome or a qualitative class. The former is commonly referred to as a regression problem. In the scenario of linear regression, the input is a continuous variable, and the prediction is a numerical value. When predicting a qualitative outcome (class), the task is considered a classification problem. Examples of classification problems include predicting what products a user will buy or if a target user will click on an online advertisement. Not all algorithms fit cleanly into this simple dichotomy, though, and logistic regression is a notable example. Logistic regression is part of the regression family as it involves predicting outcomes based on quantitative relationships between variables. However, unlike linear regression, it accepts both continuous and discrete variables as input and its output is qualitative. In addition, it predicts a discrete class such as “Yes/No” or “Customer/Non-customer”. In practice, the logistic regression algorithm analyzes relationships between variables. It assigns probabilities to discrete outcomes using the Sigmoid function, which converts numerical results into an expression of probability between 0 and 1.0. Probability is either 0 or 1, depending on whether the event happens or not. For binary predictions, you can divide the population into two groups with a cut-off of 0.5. Everything above 0.5 is considered to belong to group A, and everything below is considered to belong to group B.



5. Naïve Bayes

Let’s understand it using an example. Below I have a training data set of weather and the corresponding target variable ‘Play’ (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather conditions. Let’s follow the below steps to perform it.

- Step 1: Convert the data set into a frequency table
- Step 2: Create a Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

- Step 3: Now, use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the prediction.

6. Bernoulli Naive Bayes

Bernoulli Naive Bayes is a part of the family of Naive Bayes. It only takes binary values. The most general example is where we check if each value will be whether or not a word that appears in a document. That is a very simplified model. In cases where counting the word frequency is less important, Bernoulli may give better results. In simple words, we have to count every value binary term occurrence feature i.e. a word occurs in a document or not. These features are used rather than finding the frequency of a word in the document.

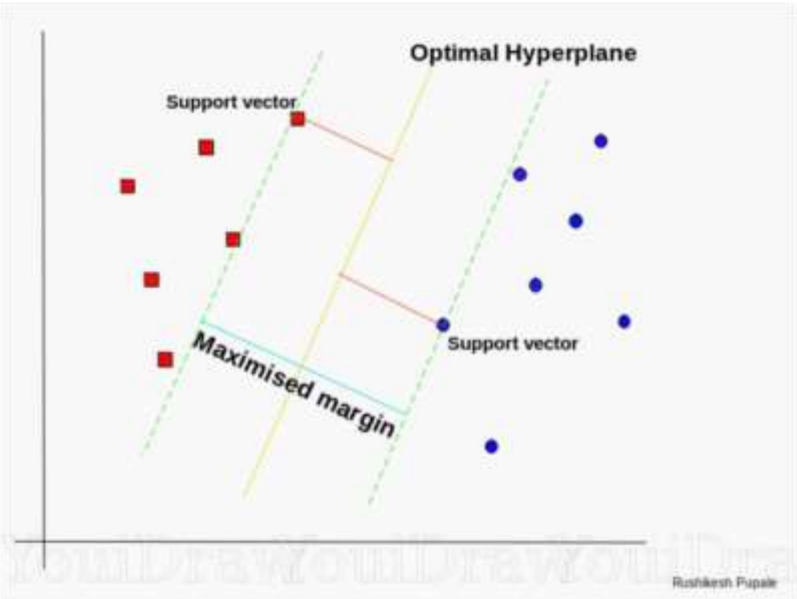
To understand it in layman’s terms, the Bernoulli distribution has two mutually exclusive outcomes: P(X=1)=p or P(X=0)=1-p.In the BernoulliNB theorem, we can have multiple features but each one is assumed to be a binary-valued variable i.e. boolean. Therefore, this class requires samples to be represented as binary-valued feature vectors. In case, any other kind of data is provided, then a BernoulliNB instance may binarize its input.

7. LinearSVC

The most applicable machine learning algorithm for our problem is Linear SVC. Before hopping into Linear SVC with our data, we're going to show a very simple example that should help solidify your understanding of working with Linear SVC. The objective of a Linear SVC (Support Vector Classifier) is to fit the data you provide, returning a "best fit" hyperplane that divides or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is. This makes this specific algorithm rather suitable for our uses, though you can use this for many situations. Let's get started.

8. SVC

At first, an approximation on what SVMs do is to find a separating line(or hyperplane) between the data of two classes. SVM is an algorithm that takes the data as an input and outputs a line that separates those classes if possible. Let us begin with a problem. Suppose you have a dataset as shown below and you need to classify the red rectangles from the blue ellipses(let's say positives from the negatives). So your task is to find an ideal line that separates this dataset into two classes (say red and blue).



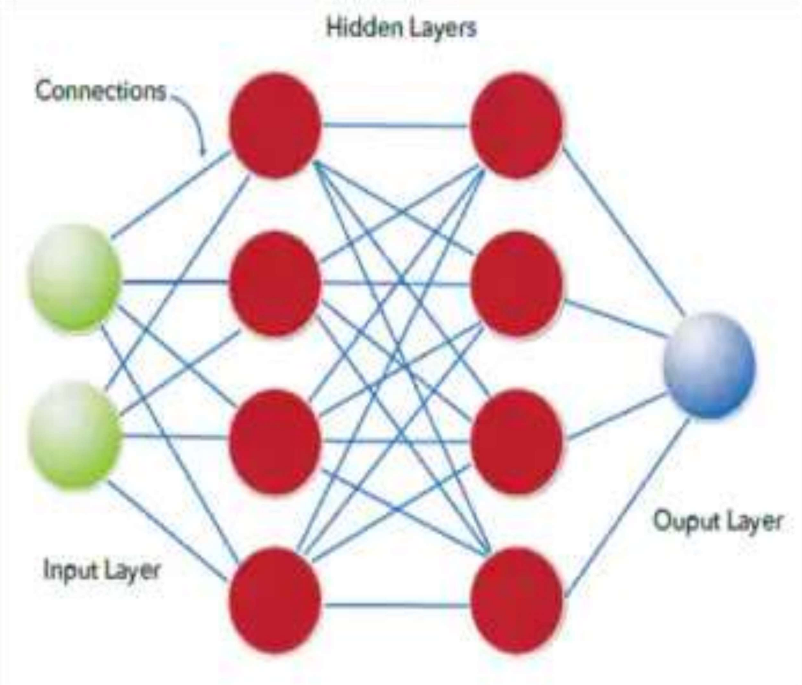
9. Why are neural networks important?

Neural networks are also ideally suited to help people solve complex problems in real-life situations. They can learn and model the relationships between inputs and outputs that are nonlinear and complex; make generalizations and inferences; reveal hidden relationships, patterns and predictions; and model highly volatile data (such as financial time series data) and variances needed to predict rare events .As a result, neural networks can improve decision processes.

Types of Neural Networks: There are different kinds of deep neural networks – and each has advantages and disadvantages, depending upon the use. Examples include:
Convolutional neural networks (CNNs) contain five types of layers: input, convolution, pooling, fully connected and output. Each layer has a specific purpose, like summarizing, connecting or activating. Convolutional neural networks have popularized image classification and object detection. However, CNNs have also been applied to other areas, such as natural language processing and forecasting.
Recurrent neural networks (RNNs) use sequential information such as time-stamped data from a sensor device or a spoken sentence, composed of a sequence of terms. Unlike traditional neural networks, all inputs to a recurrent neural network are not independent of each other, and the output for each element depends on the computations of its preceding elements. RNNs are used in forecasting and time series applications, sentiment analysis and other text applications. Feedforward neural networks, in which each perceptron in one layer is connected to every perceptron from the next layer. Information is fed forward from one layer to the next in the forward direction only. There are no feedback loops.

Autoencoder neural networks are used to create abstractions called encoders, created from a given set of inputs. Although similar to more traditional neural networks, autoencoders seek to model the inputs themselves, and therefore the method is considered unsupervised. The premise of autoencoders is to desensitize the irrelevant and sensitize the relevant. As layers are added, further abstractions are formulated at higher layers (layers closest to the point at which a decoder layer is introduced). These abstractions can then be used by linear or nonlinear classifiers.

How Neural Networks Work: A simple neural network includes an input layer, an output (or target) layer and, in, between, a hidden layer. The layers are connected via nodes, and these connections form a "network" – the neural network – of interconnected nodes.



A node is patterned after a neuron in a human brain. Similar in behavior to neurons, nodes are activated when there is sufficient stimuli or input. This activation spreads throughout the network, creating a response to the stimuli (output). The connections between these artificial neurons act as simple synapses, enabling signals to be transmitted from one to another. Signals across layers as they travel from the first input to the last output layer – and get processed along the way.When posed with a request or problem to solve, the neurons run mathematical calculations to figure out if there's enough information to pass on the information to the next neuron. Put more simply, they read all the data and figure out where the strongest relationships exist. In the simplest type of network, data inputs received are added up, and if the sum is more than a certain threshold value, the neuron "fires" and activates the neurons it's connected to As the number of hidden layers within a neural network increases, deep neural networks are formed. Deep learning architectures take simple neural networks to the next level. Using these layers, data scientists can build their own deep learning networks that enable Machinelearning , which can train a computer to accurately emulate human tasks, such as recognizing speech, identifying images or making predictions. Equally important, the computer can learn on its own by recognizing patterns in many layers of processing. So let's put this definition into action. Data is fed into a neural network through the input layer, which communicates to hidden layers. Processing takes place in the hidden layers through a system of weighted connections. Nodes in the hidden layer then combine data from the input layer with a set of coefficients and assigns appropriate weights to inputs. These input-weight products are then summed up. The sum is passed through a

node's activation function, which determines the extent that a signal must progress further through the network to affect the final output. Finally, the hidden layers link to the output layer – where the outputs are retrieved.
A simple neural network
How Neural Networks Work. A simple neural network includes an input layer, an output (or target) layer and, in between, a hidden layer. The layers are connected via nodes, and these connections form a "network" – the neural network – of interconnected nodes. A node is patterned after a neuron in a human brain.
Convolutional Neural Network
A convolutional neural network (CNN or ConvNet) is a network architecture for deep learning that learns directly from data. CNNs are particularly useful for finding patterns in images to recognize objects, classes, and categories. They can also be quite effective for classifying audio, time-series, and signal data.

Recurrent Neural Network (LSTM): Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning.

Evaluation The Model

Model	Accuracy	Recall	Precision	F1
Decision Tree	0.8931860036832413	0.8938139572758612	0.8944885529392572	0.8931682496607871
Naive Bayes	0.5377532228360957	0.5336445807659944	0.5403076322721212	0.5156489322762117
KNN	0.7642725598526704	0.7654447490567574	0.7682921810699588	0.7638488719760805
Logistics Regression	0.5340699815837937	0.53645395076138	0.5385476463834673	0.5286927286138223
Random Forest	0.8766114180478821	0.8723772427458538	0.8769819460855706	0.8708127141225732
SVC	0.5340699815837937	0.5328709861295838	0.5333351684650958	0.5317768954175968
BernoulliNB	0.5046040515653776	0.5040647648000869	0.5040747190552638	0.5039310712551239
LinearSVC	0.5230202578268877	0.5166662142721424	0.5275077281483804	0.4670276989309251