Weather Analysis using Twitter Data using Naive-Bayes and KNN Classifier

Aniket Patil[1] Abid Ghori[2] Akash Chaware[3] Elton Asher Jose Menezes[4]
[1,2,3,4]Agnel Institute of Technology & Design, India

Abstract— Weather is an important aspect of our human lives. Human can adapt to different climatic conditions but they also need to dress appropriately for them; so the weather conditions should be known to them. Weather Analysis using Twitter data aims to gather a series of Tweets based on a particular location and generate the weather condition/ conditions that the Tweets are making a reference to. An Unsupervised Learning Approach will be used in-order to automatically classify the Tweet/Tweets of a particular location into appropriate weather categories i.e. Sunny, Windy, Cold, Humid, Hot, Chilly, Cool. To achieve Unsupervised Learning Approach a Data Set will be given to a Classifier , like Naive Bayes/K-Nearest Neighbor in-order to determine which weather conditions the tweet or set of tweets belong to and also to help the classifier improve its accuracy in determining the correct output overtime. In order for classification, pre-processing steps will be carried out on the Tweet/Tweets obtained. The pre-processing steps consist of Removal of Stop Words, Removal of Hashtags, Removal of '@'symbol etc. The Steps will be explained in detailed throughout this document. Training of the Classifier involves having a set of Tweets that are already classified into the appropriate weather conditions. Once the Classifier has been trained; the test data is given to the classifier to determine the appropriate categories that the Tweet/Tweets fall under and what is the weather condition that the person/group of people belonging to that particular location are talking about.

Key words: Naive Bays Classifier, K-Nearest Neighbor

I. INTRODUCTION

The emergence of social media has given web users a venue for expressing and sharing their thoughts and opinions on all kinds of topics and events. The social media i.e. Twitter with millions of users and millions of messages per day has helped organizations to grow their brand by analyzing the twitter data they obtain through public tweets. Millions of Users tweet regarding a weather condition with respect to particular area. Therefore the proposed system, "Weather Analysis Using Twitter Data" where a large training data set of weather related tweets is given with corresponding class such as rainy, sunny, cloudy, windy or neutral etc. When new tweet makes an entry in document, it is unknown to which class it belongs to. So for that; The Naive Bayes classifier is used in which it classifies the text as sunny, rainy, cloudy etc. by computing the probability for each class. Using the support of this system a person comes to know about the condition of climate in particular location. If person wishes to move from one place to another place and wants to know about the weather condition in particular location, they will get the result from the collection of tweets with respect to that location using this system. Hence the person will come to know whether the climate is Sunny, Rainy, Windy, Cloudy etc.

II. RELATED WORK

Xiaoran An, Auroop R. Ganguly and Yi Fang proposed a piece work where they analysed the data hierarchically, they first applied subjectivity detection to distinguish a subjective list of tweets from the objective ones in the entire corpus set and then perform analysis of sentiments only within the subjective tweets. They represented an individual tweet with a bag-of-words representation. Since each tweet is short, they used binary word indicators as feature representation. they pre-processed their data as follows: they converted the letters to lowercase (stripping the case of all the words), tokenized the sentence (i.e. conversion of the string to a sequential list of tokens), removed stopwords (i.e. the words that occur frequently should be eliminated. One method of sentiment text classification is Naive Bayes. Feature Selection on Twitter data is also performed. The method of selection of features is vital since each tweet is quite short, wherein a message is not allowed to exceed 280 characters, causing a bag-of-word representation (which has dimensionality equal to the amount of words in a Twitter repository) for each chosen tweet to be very sparse.

III. PROPOSED CONCEPT

The process of weather prediction is a laborious task and considering that weather analysis is quite complex, dynamic and mindboggling. The task of forecasting weather is a herculean task since it depends on parameters like temperature, speed of the wind and humidity. These factors contribute to changes in calculation that vary from location to location. Weather forecast stands out among the best natural requirements in ones lives. Presently, weather forecasting is conducted using the knowledge of science and technology. It is made by gathering quantitative information sets about the present condition of the environment through climate station and deciphers by meteorologist. A diverse range of techniques used in data mining for prediction of weather forecasting include a variety of classifications like K-Nearest Neighbor, Decision Trees and Naïve Bayes..

The pipeline of the project is organized in the following way.

Firstly, obtaining the sets of data (both the training and the testing) has to be done. After preprocessing a set of texts, the resulting sentences (i.e. the cleaned tweets) become the instance for a subsequent new training set. Subsequently, the data set is used for training a classifier and the corresponding Test set is classified. Finally, the accuracies of the classifiers obtained from different pre-processing modules are compared with each other, for the purpose of verifying the efficiency and effectiveness of each technique. [3]

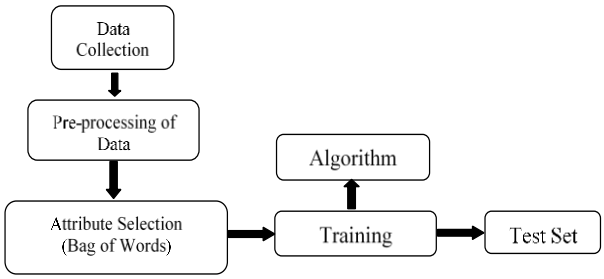


Fig. 1: Steps for training a classifier

A. Proposed Architecture

The initial module performs basic cleaning operations, consisting of removing unimportant or disturbing elements for the subsequent phases of analysis. A clean tweet should not contain any of the following URLs, hashtags (i.e.#happy) or mentions (i.e. @BarackObama). The following task is to remove the vowels repeated in a sequence that occur at least three times, by doing so the words are normalized: for example, two words that mean the same but imply the same meaning (i.e. cooooool and cool) will be treated as equals. The final step is to transfigure many types of emoticons into tags that express their sentiment (i.e. :)! smile happy). Finally, all the text is converted to lower case, and extra blank spaces are removed. [3]

The project uses different well-known classifiers like Naive Bayes and K-Nearest Neighbor classifiers and compare its accuracy. This approach will help in automatically identifying weather patterns in a tweet without any manual intervention.

After performing pre-processing, the Collected raw information is then given it to a Classifier namely Naive Bayes(NB).

Naive Bayes does the job of Separating out the tweets of one Sentiment from another. It compares each pre-processed tweet against a range of Words and finds the occurrence of them (each word of the document) to the given Word. From here it is possible deduce the probability that a new tweet belongs to a particular class and hence obtaining the desired output.

This model works with the BOWs feature extraction that ignores the position of the word in the document.

$$P(label \ / \ features \ ) = \frac{P(label \ ) * P(\ features \ / label \ )}{P(\ features \ )} \quad (1.1)$$

BOWs – Bag of words

P(label) indicates the prior probability of a label or the likelihood that a random feature set the label.

P(features|label) indicates the prior probability that a given feature set is being classified as a label.

P(features) indicates the prior probability that a given feature set is occurred [3]

After Pre-Processing and working with a Classifier next comes the third stage. Here the Data Set is obtained which has around 66,000 entries. Another Classifier called K-Nearest Neighbour (KNN) is also taken into account. In KNN a set of tweets has been taken and has been classified. K-Nearest Neighbour classifier states that if the samples are similar, they generally lies in close vicinity. K-Nearest Neighbour is an instance based classifier. Instance based classifiers are also referred to as lazy learners as they can store the training samples and they do not build a classifier until a new sample that is unlabelled needs to be classified. A case is arranged by a greater part vote of its neighbors, with the case being doled out to the class most basic among its K closest neighbors estimated by a separation work. On the off chance that K= 1, at that point the case is just allotted to the class of its closest neighbor.

B. Proposed Algorithms

1) Naive Bayes Classification Algorithm Steps

1) Consider a Training dataset with corresponding class to each document.
2) From this training dataset identify the unique words from all the documents.
3) Convert the document into feature sets, where the attributes are possible words and values are the number of times a word occurring in the given document d.
4) For each class compute:

$$P(V_j) = \frac{\text{No of Documents belonging to a class}}{\text{Total number of Documents}} \quad (1.2)$$

5) Calculate probability of each word in the document as

$$P(Wk/V_j) = \frac{nk+1}{n + |\text{vocabulary}|} \quad (1.3)$$

nk – Number of times word k occurs in class

n - Number of words in particular class

6) Consider a new document (Pre-processed)

Compute VNB for all the classes,

$$VNB = \text{argMax } P(Vj) \prod_{W \in \text{words}} P(W|Vj) \quad (1.4)$$

$P(v_j)$ – probability of class

w-word

7) Now check if which class is having maximum value.
8) The class with maximum value, the document belongs to that class.

2) K-Nearest Neighbor Classification Algorithm Steps[5]

1) Determination of weight matrix

– Think about matrix with dimensions NxM, where N dimension is characterized by a number of unique words in a sample of document and M represents the number of documents to be classified.
– Each matrix element A[i,j] represents weight value of word i in the document j.

2) Determination of weight value A[i,j]

– Term frequency

Computation of number of repetitions of a word(term) in the document j.

$a_{ij} = f_{ij}$ = frequency of term i in document j

– Term frequency-inverse document frequency(TF-IDF)

- TF-IDF strategy decides the relative recurrence of words in a particular Document through a reverse extent of the word over the whole document corpus.
- In determining the value, the method uses two elements: TF - term frequency of term i in document j and IDF - inverse document frequency of term i.
- Two types to find weight value

– Used when documents are of same length

$$a_{ij} = tf_{ij} idf_i = tf_i \times \log_2 \left(\frac{N}{df_i} \right) \quad (2.1)$$

tf_{ij}-term frequency of term i in document j

idf_i-inverse document frequency of term i
N-number of documents in the collection/number of unique words
df_i – document frequency of term i in the collection

– Used when documents are of different length

$$a_{ij} = tf_{ij} idf_i = \frac{f_{ij}}{\sqrt{\sum_{s=1}^{N} (tfidf\ (a_{sj}))^2}} \log_2 \left(\frac{N}{df_i} \right) \quad (2.2)$$

– KNN classification process

- Select the document that will be carried out for classification.
- Include this document in the matrix (if new it contains new word then add it to the row).
- For the chosen document determine its weight value using the TF-IDF method, as well as for all other documents.
- Determine the K value.
- K estimation of the KNN calculation is a factor which shows a required number of documents from the gathering which is nearest to the chosen document. The Classification procedure determines the vectors distance between the documents by using the following equation

$$d(x, y) = \sqrt{\sum_{r=1}^{N} (a_{rx} - a_{ry})^2} \quad (2.3)$$

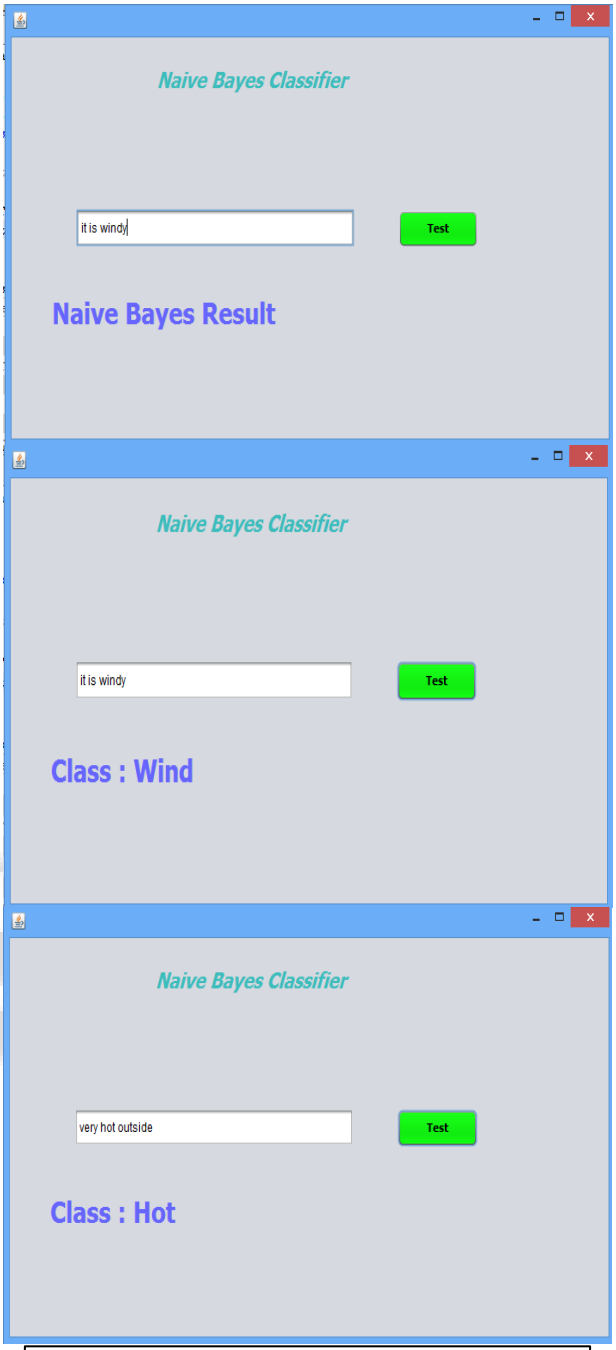
d(x,y)-distance between two documents
a_{rx} - weight of term r in document x
a_{ry} – weight of term r in document y

- Smaller the Euclidean distance between the documents indicates their higher compaability. Distance 0 implies that the documents are completely equal.

IV. RESULTS

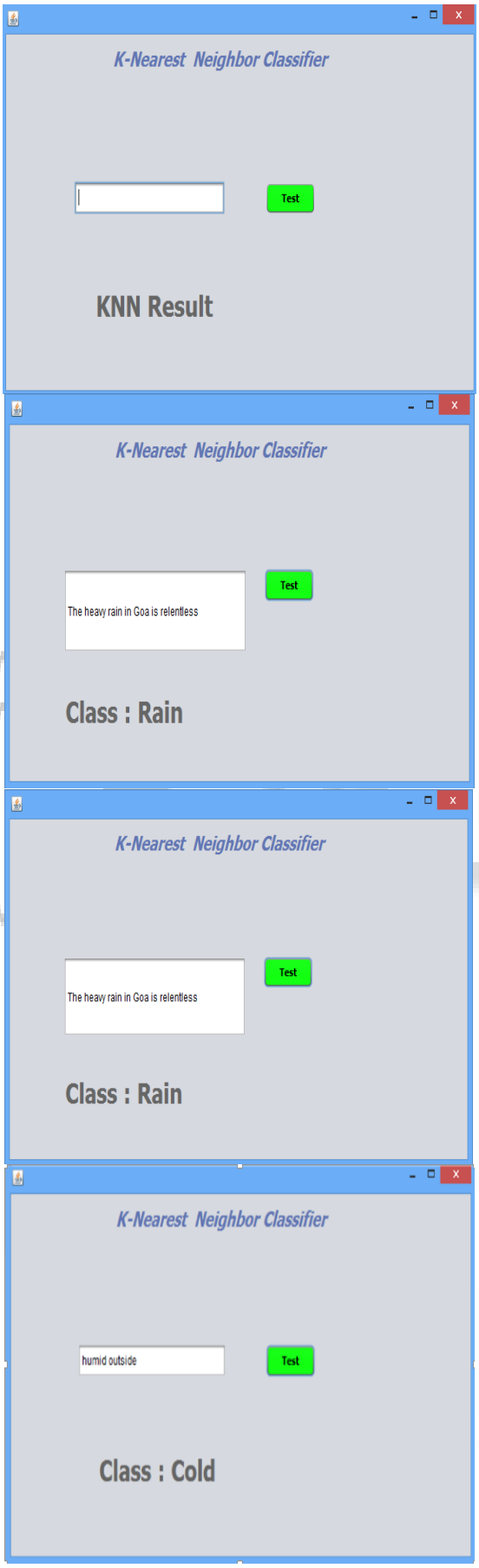
A. *Naive Bayes*

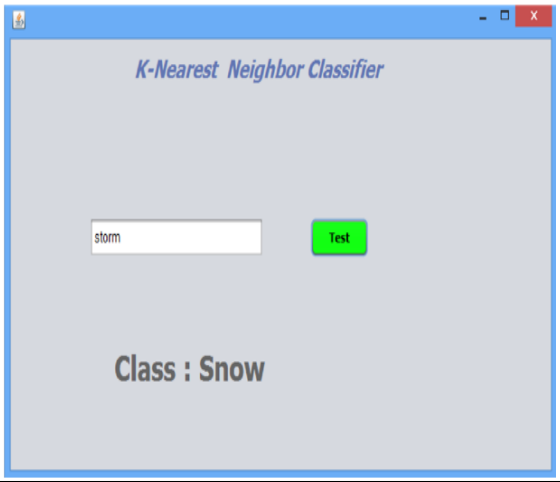
No	Tweet	Class
1	forecast today mostly cloudy shower thunderstorm l...	Clouds
2	partly cloudy tonight low c sunn thursday high nea...	Clouds
3	cloud layer ksc expect scatter endeavour s et laun...	Clouds
4	boston weather forecast cloud shower prevail week ...	Clouds
5	st look weather endeavour s st launch go concern l...	Clouds
6	san antonio texa weather partly cloudy san antonio...	Clouds
7	dear cloud stop don t want rain graduate weekend	Clouds
8	boston weather forecast sun cloud new boston	Clouds
9	cloud end perfect photo weather least end first wa...	Clouds
10	ugh cant believe ll rain cloudy chilly next week m...	Clouds
11	current condit partly cloudy fforecast sat clear h...	Clouds
12	accid wb ramp hard pl pillow st merritt ave weathe...	Clouds
13	current wx st mary s county time temp feel cond pa...	Clouds
14	nice day today albuquerque cloudy windy nice tempe...	Clouds
15	mostly cloudy butler county automat weather observ...	Clouds
16	wish house freez night sleep nice cold	Clouds
17	s degree cold tomorrow high s short weather right	Clouds
18	partly cloudy tonight low c sunn monday high near ...	Clouds



No	Tweet	Class
1076	lol crazy weather	I cant tell
1077	come enjoy amaz weather meet fabul dog o reilly s ...	I cant tell
1079	cold outside	Cold
1080	hate rain season	Hurricane
1081	hot outside	Hot
1082	humid weather	Humid
1083	heavy rain	Rain
1084	cloudy outside	Clouds
1085	sound thunderstorm	Storm
1086	windy	Wind
1087	hot outside	Hot

B. KNN Classifier





No	Tweets	Class
1076	lol crazy weather	I cant tell
1077	come enjoy amaz weather meet fabul dog o reilly s ...	I cant tell
1078	s hot humid cold windy dry sunn	I cant tell
1079	heavy rain outside	Hot
1080	windy	Snow
1081	humid outside	Cold
1082	storm	Snow

V. CONCLUSION

The paper aims to achieve an automated system for the prediction of what individuals are talking about in their Tweets on Twitter hence the name of the Project "Weather Analysis using Twitter Data". There are different pre-processing strategies to expel undesirable artifacts in the tweet which are not required for the classification procedure. These pre-processing techniques help with the ultimate classification of the Tweet into weather various categories. This project implements two types of classifiers Naive-Bayes Classifier and K-Nearest Neighbor Classifier. The Training data is given to each of the classifiers so that they can use the trained data and help classify a new tweet by an individual.

The results of classification have proved that Naive-Bayes Classifier provides a better method of classification that is faster as well as more accurate compared to the K-Nearest Neighbor Classifier which takes a much longer time and seldom provides accurate results.

The techniques applied in this paper help us Classify Tweets into weather categories. There is a possibility to improve this classification with other pre-processing techniques.

REFERENCES

[1] Daniel Jurafsky & James H. Martin," Naive Bayes and Sentiment Classification", (Speech and Language Processing. Copyright c 2016. All rights reserved. Draft of August 7, 2017.)
[2] Mr. Mane Mayur R. , Mr. Kalambate Akshay R., Mr. Rane Zilu Ramkrishna, Prof. Gamare P. S," Machine

Learning Algorithm For Sentimental Analysis of Twitter Feeds".

[3] Giulio Angiani, Laura Ferrari, Tomaso Fontanini, Paolo Fornacciari, Eleonora Iotti, Federico Magliani, and Stefano, "A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter",(Manicardi Dipartimento di Ingegneria dell'Informazione Universita` degli Studi di Parma Parco Area delle Scienze 181/A, 43124 Parma, Italy).

[4] Kiruthika .M Department of Computer Engineering." Sentiment Analysis of Twitter Data"

[5] BrunoTrstenjak ,Sasa Mikac, Dzenana Donko,"KNN with TF-IDF Based Framework for Text Categorization"

