

AN EFFICIENT WAY TO INFERENCE LOCATION INFORMATION OF A NON-GEOTAGGED TWEETS

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

U.DINESH RATNA SAI KRISHNA (17SCS0473)
P.SANDEEP REDDY (17SCS0414)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC | 12B

status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI

SALAI, CHENNAI – 600 119

MARCH - 2021



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **UPPALA DINESH RATNA SAI KRISHNA (17SCS0473)** and **PUTTA SANDEEP REDDY (17SCS0473)** who carried out the project entitled "**AN EFFICIENT WAY TO INFERENCE LOCATION INFORMATION OF A NON-GEOTAGGED TWEETS**" under my supervision from **November 2020 to March 2021**.

Internal Guide

Mrs. R. Aishwarya, M.E.,

Head of the Department

Dr.L.Lakshmanan M.E., Ph.D

Dr. S. VIGNESHWARI, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **UPPALA DINESH RATNA SAI KRISHNA** hereby declare that the Project Report entitled “**AN EFFICIENT WAY TO INFERENCE LOCATION INFORMATION OF A NON-GEOTAGGED TWEETS**” is done by me under the guidance of **Mrs. R. Aishwarya, M.E.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D.**, Dean, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.**, and **Dr. L. Lakshmanan, M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. R. Aishwarya, M.E., Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Social media like Twitter have become globally popular in the past decade. Thanks to the high penetration of smartphones, social media users are increasingly going mobile. This trend has contributed to foster various location based services deployed on social media, the success of which heavily depends on the availability and accuracy of users' location information. However, only a very small fraction of tweets in Twitter are geo-tagged. Therefore, it is necessary to infer locations for tweets in order to attain the purpose of those location based services. In this paper, we tackle this problem by scrutinizing Twitter user timelines in a novel fashion. First of all, we split each user's tweet timeline temporally into a number of clusters, each tending to imply a distinct location. Subsequently, we adapt two machine learning models to our setting and design classifiers that classify each tweet cluster into one of the pre-defined location classes at the city level. The Bayes based model focuses on the information gain of words with location implications in the user-generated contents. The convolutional LSTM model treats user-generated contents and their associated locations as sequences and employs bidirectional LSTM and convolution operation to make location inferences. The two models are evaluated on a large set of real Twitter data. The experimental results suggest that our models are effective at inferring locations for non-geotagged tweets and the models outperform the state-of-the-art and alternative approaches significantly in terms of inference accuracy.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1.	INTRODUCTION	1
	1.1 NON-GEOTAGGED TWEETS	1
	1.2 MANAGEMENT OF LOCATION	2
2.	LITERATURE SURVEY	7
3.	SYSTEM DESIGN & METHODOLOGY	13
	3.1 EXISTING SYSTEM	13
	3.2 PROPOSED SYSTEM	13
	3.3 MACHINE LEARNING	14
	3.3.1 Introduction	14
	3.3.2 Machine Learning Tasks	15
	3.3.3 Types of Learning Algorithm	16
	3.4 SUPPORT VECTOR MACHINE	16
	3.4.1 PROS AND CONS OF SVM	24
	3.5 REQUIREMENT SPECIFICATION	25
	3.6 MODULES	25
4.	IMPLEMENTATION	24
	4.1 DESIGN AND IMPLEMENTATION CONSTRAINTS	24
	4.2 OTHER NON-FUNCTIONAL REQUIREMENTS	28
	4.3 ARCHITECTURE DIAGRAM	29
5.	RESULTS AND DISCUSSION	35

7.	CONCLUSION AND FUTURE WORK	40
	REFERENCES	41
	APPENDIX	
	A. PLAGIARISM REPORT	
	B. JOURNAL PAPER	
	C. SOURCE CODE	

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.1	ARCHITECTURE DIAGRAM	30
4.2	SEQUENCE DIAGRAM	31
4.3	USECASE DIAGRAM	32
4.4	ACTIVITY DIAGRAM	33
4.5	COLLOBARATION DIAGRAM	33
5.1	EXECUTION PAGE	35
5.2	HOME PAGE	36
5.3	LOGIN FORM	37
5.4	TWEETS SEARCH	38
5.5	OUTPUT FORM	39

CHAPTER 1

INTRODUCTION

1.1 NON-GEOTAGGED TWEETS

Social media-based life organizing destinations, for example, facebook, Twitter and so forth permit clients to share sees, photographs and recordings, posts, and to advise others about on the social media or true exercises. The accomplishment of long-range interpersonal communication administrations can be found in strength in today's society with Facebook having a gigantic 2.13 billion dynamic month to month clients and a normal of 1.4 billion day by day dynamic clients in 2017. Some informal communication administrations expect individuals to have a prior association with contact different individuals. Twitter is one of the most microblogging administrations in social media-based life organizing social media site with huge number of clients who associate to one another through this system. These clients share their thoughts, conclusions, specialized information, sees and their closely-held convictions on explicit occasions in the general public and are shared through the twitter messages. Twitter is one of the most conspicuous person to person communication applications to frame same local gatherings between specialized experts, loved ones and business people groups.

These people group individuals express their perspectives, news to different individuals in the network through twitter. Twitter messages are limited to 280 characters. Twitter organizing site permits the clients to follow their preferred researchers, specialists and other famous people. A client can make his own record in arrange transparently with no limitations, essentially giving his work force subtleties like name, staff id and address. Because of this open getting to approach into twitter arrange numerous clients abuse the system exercises. They basically misdirect the ordinary clients through retweets, url connections and hash labels. Spammers are pulled in by twitter organize as a supporting instrument to spread spam messages, commercials to certified clients. The spammers likewise send url and pernicious connects to the authentic clients.

Spam is probably the most concerning issue in social media based life destinations like facebook linkedIn Scientists shows that over 3% of tweets are spam messages. The drifting subjects are likewise assaulted by spammers.

1.2 MANAGEMENT OF LOCATION

Location inference for tweets are tested by two significant issues. Initially, Twitter constrains the length of each tweet substance to 140 characters, and along these lines a tweet just contains few words and passes on restricted information. Second, Twitter clients regularly use non-standard and short-hand terms, and tweets are frequently vague and boisterous. Consequently, discovering location hints from short, boisterous tweets is undeniably troublesome. Right now, research how to surmise the locations of non-geotagged tweets at the city level by investigating Twitter clients' courses of events utilizing a novel methodology. Our methodology joins examination on the substance of tweet short messages and that on the client courses of events with fleeting information. Along the fleeting measurement, every client course of events is part into various tweet groups; each bunch suggests a particular client location. This procedure is called worldly bunching of tweets.

Along these lines, two AI models are painstakingly adjusted to our concern setting and classifiers are intended to arrange each tweet group from a client's course of events into one of the pre-characterized location classes at the city level. The Bayes put together model concentrations with respect to the information increase of words with location suggestions in the client produced substance, while the LSTM based model treats client created substance and their related locations as arrangements and utilizes a bidirectional LSTM and convolution operation to make location inductions. Our models are prepared utilizing disconnected information, yet they can be utilized to deduce locations for chronicled tweets and social media based (approaching) tweets. The two models are tentatively assessed on an enormous genuine information set, in correlation with elective methodologies. The trial results recommend that the proposed models are powerful at gathering locations for tweets and they beat choices fundamentally as far as inference precision.

Our commitments right now abridged as follows. We structure transient bunching strategies that split a client's tweet timetable into a lot of groups every one of which contain tweets that are likely sent from a similar city. We structure a Bayes' hypothesis based model for location inference for tweet groups. The model estimates words' land scopes by processing words' information gains over all locations of intrigue. We fabricate a novel neural system that joins convolution activity and long transient memory unit when removing highlights from the substance of tweet groups. It can misuse spatially-nearby relationship while gathering locations for tweet groups. We assess the presentation of our proposed approach and models utilizing certifiable Twitter information. The outcomes show that our methodology with the models beats best in class options.

Thanks to the high penetration of smartphones, social media users are increasingly going mobile. For example, 56.5 percent Facebook users login only from mobile devices by the end of July, 2016,¹ and Twitter has approximately 257 million mobile active users monthly as per the first quarter in 2016.² This big trend has not only promoted large-scale mobile user profiling, but also fostered various location based services deployed on social media. The success of such location based services heavily depends on the availability and accuracy of users' location information that a social media platform can get access to. Knowing the locations of individual tweets of users enables a wide variety of applications, e.g., location-based summarization, location-aware recommender system friends notification influential users recommendation place advertisements and business information spreading city-scale collective attention analytics and even disaster detection. However, only a very small fraction of tweets are geotagged, i.e., being sent with GPS coordinates or place names. Cheng et al. found that only 26 percent of Twitter users in a random sample of over 1 million users reported their location in their profiles and only 0.42 percent of the tweets in the sample were geo-tagged.

Therefore, it is necessary to infer locations for Twitter users in order to attain the purpose of and to improve the quality of the location based services offered to the users. Most related works so far have focused on inferring only the home location for a Twitter user's timeline. This is apparently insufficient for location based services in general. Location inference for tweets are challenged by two major issues. First, Twitter limits the length of each tweet content to 140 characters, and

thus a tweet only contains a small number of words and conveys limited information. Second, Twitter users often use non-standard and shorthand terms, and tweets are often unclear and noisy. Consequently, finding location clues from short, noisy tweets is indubitably difficult. In this study, we investigate how to infer the locations of non-geotagged tweets at the city level by scrutinizing Twitter users' timelines using a novel approach. Our approach combines analysis on the contents of tweet short texts and that on the user timelines with temporal information. Along the temporal dimension, each user timeline is split into a number of tweet clusters; each cluster implies a distinct user location. This process is called temporal clustering of tweets. Subsequently, two machine learning models map a tweet cluster from a user's timeline into one of the pre-defined location classes at the city level. The Bayes based model focuses on the information gain of words with location implications in the user-generated contents, whereas the LSTM based model treats user-generated contents and their associated locations as sequences and employs a bidirectional LSTM and convolution operation to make location inferences.

Our models are trained using offline data, but they can be used to infer locations for historical tweets and online (incoming) tweets. The two models are experimentally evaluated on a large real dataset, in comparison with alternative approaches. The experimental results suggest that the proposed models are effective at inferring locations for tweets and they outperform alternatives significantly in terms of inference accuracy. Our contributions in this study are summarized as follows. _ We design temporal clustering methods that split a user's tweet timeline into a set of clusters each of which contains tweets that are likely sent from the same city. We design a Bayes' theorem based model for location inference for tweet clusters. The model measures words' geographical scopes by computing words' information gains across all locations of interest. We build a novel neural network that combines convolution operation and long short-term memory unit when extracting features from the contents of tweet clusters. It is able to exploit spatially-local correlation when inferring locations for tweet clusters. We evaluate the performance of our proposed approach and models using real-world Twitter data. The results show that our approach with the models outperforms state-of-the-art alternatives. The rest of this paper is organized as follows. Section

2 reviews the related work. Section 3 formulates the research problem and introduces our solution framework. Section 4 describes the temporal clustering methods for training and testing data. Sections 5 and 6 detail our two models for tweet location inference, respectively. Section 7 reports on the experimental results.

Finally, Section 8 concludes the paper and points to future work directions. Several existing techniques make use of social networks and friends' locations to infer locations for Twitter users. Davis Jr. et al. and Jurgens consider the locations of a user's friends and take the friend's location with the majority votes as the user's location. Rout et al. use an SVM classifier and features extracted from Twitter user networks to predict home locations for Twitter users. Backstrom et al. study the relationship between friendship and spatial distance for Facebook users, and exploit the relationship to compute a Facebook user's home location. McGee et al. enhance that method with social tie strength for Twitter users. Rodrigues et al. combine user-posted texts and user friendship network to infer the locations of Twitter users. Li et al. propose a unified discriminative influence model that utilizes both user contents and social network to infer user locations. Jurgens et al. [28] report on a comparative study of user location inference approaches based on social networks. As social networks are dynamic and their topology changes, predicting users' locations based on social networks may face difficulties.

Therefore, more approaches make the location inferences only using user-generated contents. Most content-based approaches aim to build a probabilistic system and determine user location by maximum likelihood. Wing et al. [29] use language models and information retrieval techniques for location prediction. Dividing the space into grids, the approach compares the distribution of words in a given user's tweets to those in each grid cell using Kullback-Leibler (KL) divergence to identify the user's most likely location. The approach has a serious data skewness problem—grid cells in rural areas tend to contain very few tweets while those in urban contain too many.

To alleviate the problem, Roller et al. use a k-d-tree-based adaptive grid in which cells contain approximately the same amount of data. Topic models are also often

used to infer locations based on contents. Eisenstein et al. build geographic topic models to infer the home locations of Twitter users in terms of regions and states. The follow-up works use sparse additive models to combine region-specific, userspecific and non-informative topics. Chen et al. build a topic model to mine user interest from short texts and then establish a mapping between locations and user interests. Some probabilistic approaches use the idea of naïve Bayes to estimate the probabilities that users are located in given locations.

CHAPTER 2

LITERATURE SURVEY

Project Title: A Probabilistic Framework for Location Inference from Social Media

Author Name: Yujie Qian, Jie Tang, Zhilin Yang, Binxuan Huang, Wei Wei, Kathleen M. Carley

Year of Publishing: 2017

Abstract:

We study the extent to which we can infer users' geographical locations from social media. Location inference from social media can benefit many applications, such as disaster management, targeted advertising, and news content tailoring. In recent years, a number of algorithms have been proposed for identifying user locations on social media platforms such as Twitter and Facebook from message contents, friend networks, and interactions between users. In this paper, we propose a novel probabilistic model based on factor graphs for location inference that has several unique advantages for this task. First, the model generalizes previous methods by incorporating content, network, and deep features learned from social context. The model is also flexible enough to support both supervised learning and semi-supervised learning. Second, we explore several learning algorithms for the proposed model, and present a Two-chain Metropolis-Hastings (MH+) algorithm, which improves the inference accuracy. Third, we validate the proposed model on three different genres of data – Twitter, Weibo, and Facebook – and demonstrate that the proposed model can substantially improve the inference accuracy (+3.3-18.5% by F1-score) over that of several state-of-the-art methods.

Project Title: Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Author Name: Nitish Srivastava, Geoffrey E. Hinton, Alex S. Ilya, Ruslan Salakhutdinov

Year of Publishing: 2014

Abstract:

Deep neural networks with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by

combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different "thinned" networks. At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights. This significantly reduces over fitting and gives major improvements over other regularization methods. We show that dropout improves the performance of neural networks on supervised learning tasks in vision, speech recognition, document classification and computational biology, obtaining state-of-the-art results on many benchmark data sets.

Project Title: Home Location Identification of Twitter Users

Author Name: JALAL MAHMUD, JEFFREY NICHOLS, CLEMENS DREWS

Year of Publishing: 2012

Abstract:

We present a new algorithm for inferring the home location of Twitter users at different granularities, including city, state, time zone or geographic region, using the content of users' tweets and their tweeting behavior. Unlike existing approaches, our algorithm uses an ensemble of statistical and heuristic classifiers to predict locations and makes use of a geographic gazetteer dictionary to identify place-name entities. We find that a hierarchical classification approach, where time zone, state or geographic region is predicted first and city is predicted next, can improve prediction accuracy. We have also analyzed movement variations of Twitter users, built a classifier to predict whether a user was travelling in a certain period of time and use that to further improve the location detection accuracy. Experimental evidence suggests that our algorithm works well in practice and outperforms the best existing algorithms for predicting the home location of Twitter users.

Project Title: ImageNet Classification with Deep Convolutional Neural Networks

Author Name: Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton

Abstract:

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the Image Net LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way soft max. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce over fitting in the fully-connected layers we employed a recently-developed regularization method called “dropout” that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

Project Title: Location Prediction in Social Media Based on Tie Strength

Author Name: Jeffrey McGee, James Caverlee, Zhiyuan Cheng

Year of Publishing: 2013

Abstract:

We propose a novel network-based approach for location estimation in social media that integrates evidence of the social tie strength between users for improved location estimation. Concretely, we propose a location estimator { Friendly Location { that leverages the relationship between the strength of the tie between a pair of users, and the distance between the pair. Based on an examination of over 100 million geo- encoded tweets and 73 million Twitter user profiles, we identify several factors such as the number of followers and how the users interact that can strongly reveal the distance between a pair of users. We use these factors to train a decision tree to distinguish between pairs of users who are likely to live nearby and pairs of users who are likely to live in different areas. We use the results of this decision tree as the input to a maximum likelihood estimator to predict a user's location. We find that this proposed method significantly improves the results of location estimation relative to a state- of-the-art technique. Our system reduces the average error distance for 80% of Twitter users from 40 miles to 21 miles using only information from the user's friends and friends-of-friends, which has great significance for augmenting traditional social media and

enriching location-based services with more refined and accurate location estimates.

Project Title: On the importance of initialization and momentum in deep learning

Author Name: Ilya Sutskever, James Martens, George Dahl, Georey Hinton

Year of Publishing: 2013

Abstract:

Deep and recurrent neural networks (DNNs and RNNs respectively) are powerful models that were considered to be almost impossible to train using stochastic gradient descent with momentum. In this paper, we show that when stochastic gradient descent with momentum uses a well-designed random initialization and a particular type of slowly increasing schedule for the momentum parameter, it can train both DNNs and RNNs (on datasets with long-term dependencies) to levels of performance that were previously achievable only with Hessian-Free optimization. We find that both the initialization and the momentum are crucial since poorly initialized networks cannot be trained with momentum and well-initialized networks perform markedly worse when the momentum is absent or poorly tuned. Our success training these models suggests that previous attempts to train deep and recurrent neural networks from random initializations have likely failed due to poor initialization schemes. Furthermore, carefully tuned momentum methods suffice for dealing with the curvature issues in deep and recurrent network training objectives without the need for sophisticated second-order methods.

Project Title: Text-Based Twitter User Geo location Prediction

Author Name: Bo Han B, Paul Cook, Timothy Baldwin

Year of Publishing: 2014

Abstract:

Geographical location is vital to geospatial applications like local search and event detection. In this paper, we investigate and improve on the task of text-based geo location prediction of Twitter users. Previous studies on this topic have typically assumed that geographical references (e.g., gazetteer terms, dialectal words) in a text are indicative of its author's location. However, these references are often buried in informal, ungrammatical, and multilingual data, and are therefore non-

trivial to identify and exploit. We present an integrated geo location prediction framework and investigate what factors impact on prediction accuracy. First, we evaluate a range of feature selection methods to obtain “location indicative words”. We then evaluate the impact of non geo tagged tweets, language, and user-declared metadata on geo location prediction. In addition, we evaluate the impact of temporal variance on model generalization, and discuss how users differ in terms of their geo locatability. We achieve state-of-the-art results for the text-based Twitter user geo location task, and also provide the most extensive exploration of the task to date. Our findings provide valuable insights into the design of robust, practical text-based geo location prediction systems.

Project Title: Towards Real-Time, Country-Level Location Classification of Worldwide Tweets

Author Name: Arkaitz Zubiaga, Alex Voss, Rob Procter, Maria Liakata, Bo Wang¹, Adam Tsakalidis

Year of Publishing: 2017

Abstract:

The increase of interest in using social media as a source for research has motivated tackling the challenge of automatically geo locating tweets, given the lack of explicit location information in the majority of tweets. In contrast to much previous work that has focused on location classification of tweets restricted to a specific country, here we undertake the task in a broader context by classifying global tweets at the country level, which is so far unexplored in a real-time scenario. We analyze the extent to which a tweet’s country of origin can be determined by making use of eight tweet-inherent features for classification. Furthermore, we use two datasets, collected a year apart from each other, to analyze the extent to which a model trained from historical tweets can still be leveraged for classification of new tweets. With classification experiments on all 217 countries in our datasets, as well as on the top 25 countries, we offer some insights into the best use of tweet-inherent features for an accurate country-level classification of tweets. We find that the use of a single feature, such as the use of tweet content alone – the most widely used feature in previous work leaves much to be desired. Choosing an appropriate combination of both tweet content and metadata can actually lead to substantial improvements of between 20% and 50%.

We observe that tweet content, the user's self-reported location and the user's real name, all of which are inherent in a tweet and available in a real-time scenario, are particularly useful to determine the country of origin. We also experiment on the applicability of a model trained on historical tweets to classify new tweets, finding that the choice of a particular combination of features whose utility does not fade over time can actually lead to comparable performance, avoiding the need to retrain. However, the difficulty of achieving accurate classification increases slightly for countries with multiple commonalities, especially for English and Spanish speaking countries.

CHAPTER 3

SYSTEM DESIGN & METHODOLOGY

3.1 EXISTING SYSTEM

As informal organizations are not static & their topology changes, foreseeing clients' locations dependent on interpersonal organizations may confront troubles. Thusly, more methodologies make the location inductions just utilizing client produced substance. Most substance based methodologies expect to fabricate a probabilistic framework and decide client location by greatest probability. Existing strategies for construing locations in informal organizations fall in two classes: First those deducing a Twitter client's location for example their location inferences are totally founded on tweet substance. A few existing methods utilize interpersonal organizations and companions' locations to construe locations for Twitter clients and Second those inducing locations for tweets. In any case, a solitary tweet might be too short to even consider containing adequate highlights like different tweets that are even posted in a similar location. What's more, individuals frequently post comparable tweets at various occasions and locations.

Business can obtain the geo location of their user with the geo-tagged content like images or tweets. But not all tweets were geo-tagged. The existing system depends on the provision of the content to get the user location. The user provided content is not always contains geo-location.

3.2 PROPOSED SYSTEM

In proposed framework about non geotagged tweets, locations at the city level by investigating twitter clients' courses of events utilizing a novel methodology. Our methodology joins investigation on the substance of tweet short messages and that on the client courses of events with transient information. Along the fleeting measurement, every client timetable is part into various tweet bunches; each group suggests an unmistakable client location. This method is called transient bundling of tweets. In this manner, two AI models are meticulously changed in accordance with our anxiety setting & classifiers have planned to portray each tweet bunch from a customer's course of occasions.

Our framework utilizes the content substance and different properties of the tweet like time zone, language and so forth to distinguish the tweet nation. A SVM model is prepared with given information set. This model is utilized to anticipate the nation of beginning on the tweet. Our models are prepared utilizing disconnected information, however they can be utilized to derive locations for verifiable tweets and social media based (approaching) tweets. The two models are tentatively assessed on an enormous genuine information set, in examination with elective methodologies.

To overcome the fallback in the existing system we propose a machine learning based system to programmatically identify the user tweet country. Our system uses the text content and other properties of the tweet like time zone, language etc. to identify the tweet country. An SVM model is trained with given dataset. This model is used to predict the country of origin on the tweet.

3.3 MACHINE LEARNING

3.3.1 INTRODUCTION

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through learning. In its application across business problems, machine learning is also referred to as predictive analytics.

3.3.2 MACHINE LEARNING TASKS

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

Classification algorithms and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and false. Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object.

In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of "features", or inputs, in a set of data. Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of

of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning to play a game against a human opponent. Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

3.3.3 TYPES OF LEARNING ALGORITHM

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

- SUPERVISED LEARNING
- UNSUPERVISED LEARNING
- SEMI-SUPERVISED LEARNING

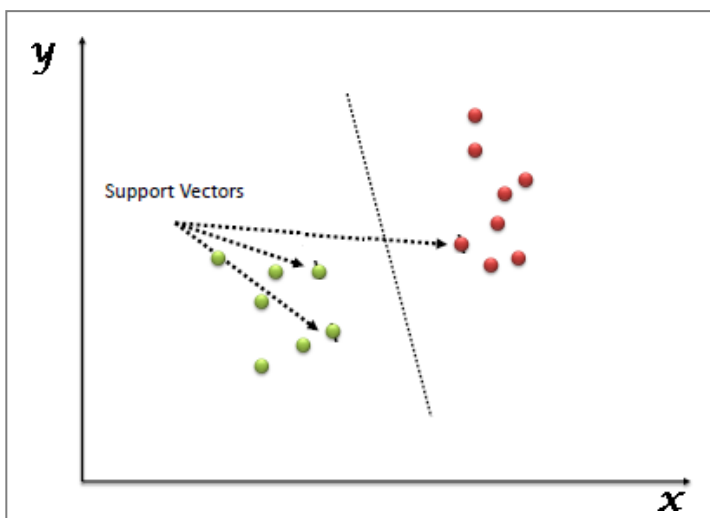
3.4 SUPPORT VECTOR MACHINE(SVM):

Mastering machine learning algorithms isn't a myth at all. Most of the beginners start by learning regression. It is simple to learn and use, but does that solve our purpose? Of course not! Because, you can do so much more than just Regression!

Think of machine learning algorithms as an armory packed with axes, sword, blades, bow, dagger etc. You have various tools, but you ought to learn to use them at the right time. As an analogy, think of 'Regression' as a sword capable of slicing and dicing data efficiently, but incapable of dealing with highly complex data. On the contrary, 'Support Vector Machines' is like a sharp knife – it works on

smaller datasets, but on them, it can be much stronger and powerful in building models.

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for either classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

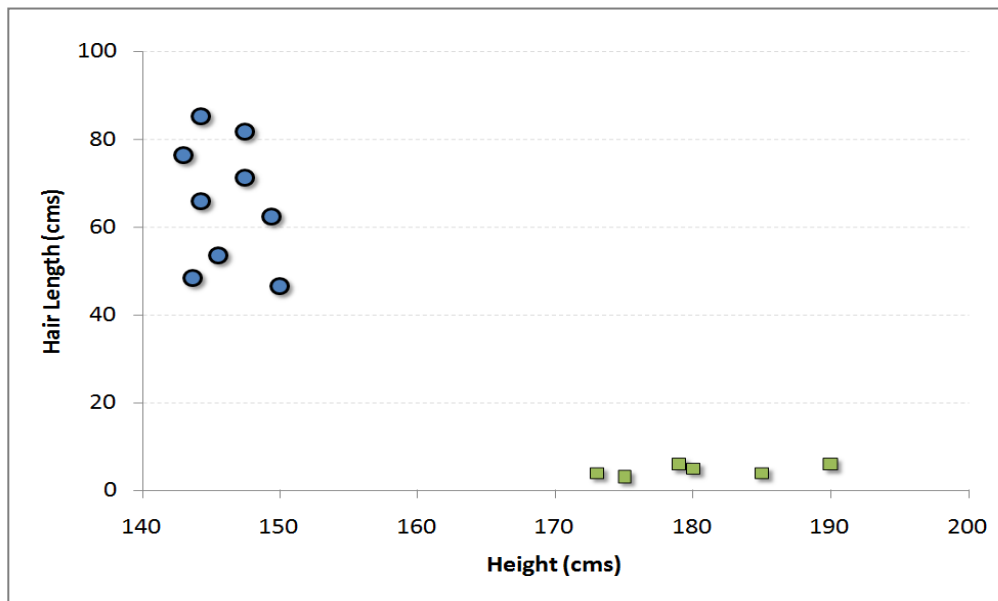
Support Vector Machine – Simplified

Let's talk about how support vector machine works. This is suitable for readers who do not know much about this algorithm and have a curiosity to learn a new technique. In following explore the technique in detail and analyze cases where such techniques are stronger than other techniques.

Classification analysis

Let's consider an example to understand these concepts. We have a population composed of 50%-50% Males and Females. Using a sample of this

population, you want to create some set of rules which will guide us the gender class for rest of the population. Using this algorithm, we intend to build a robot which can identify whether a person is a Male or a Female. This is a sample problem of classification analysis. Using some set of rules, we will try to classify the population into two possible segments. For simplicity, let's assume that the two differentiating factors identified are: Height of the individual and Hair Length. Following is a scatter plot of the sample.



The blue circles in the plot represent females and green squares represents male. A few expected insights from the graph are:

1. Males in our population have a higher average height.
2. Females in our population have longer scalp hairs.

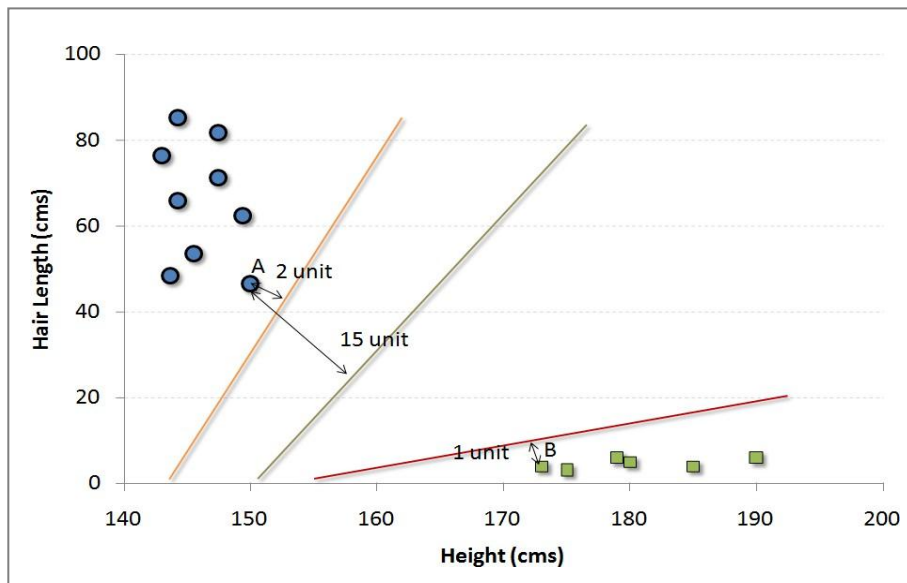
If we were to see an individual with height 180 cm's and hair length 4 cm's, our best guess will be to classify this individual as a male. This is how we do a classification analysis.

Support Vector and SVM

Support Vectors are simply the co-ordinates of individual observation. For instance, (45,150) is a support vector which corresponds to a female. Support Vector Machine is a frontier which best segregates the Male from the Females. In

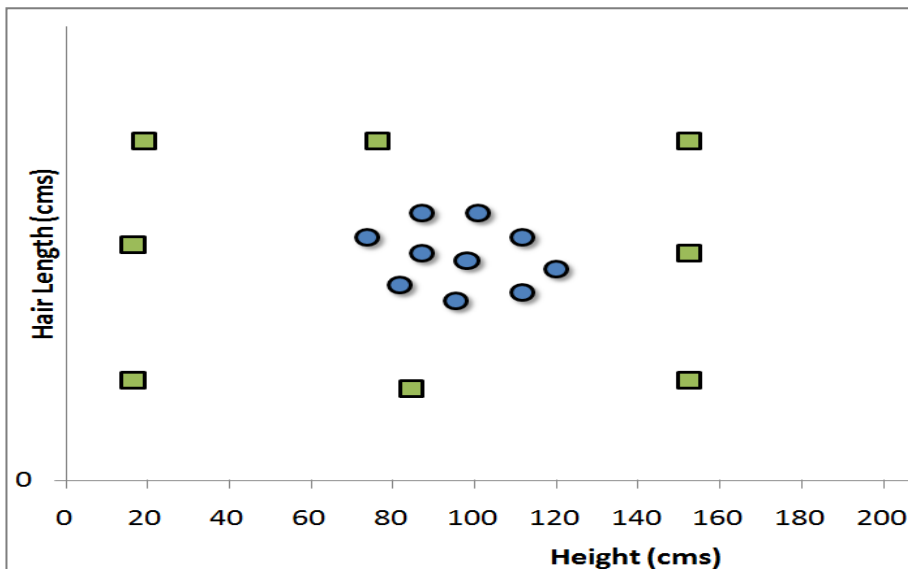
this case, the two classes are well separated from each other; hence it is easier to find a SVM.

There are many possible frontiers which can classify the problem in hand. Following are the three possible frontiers.

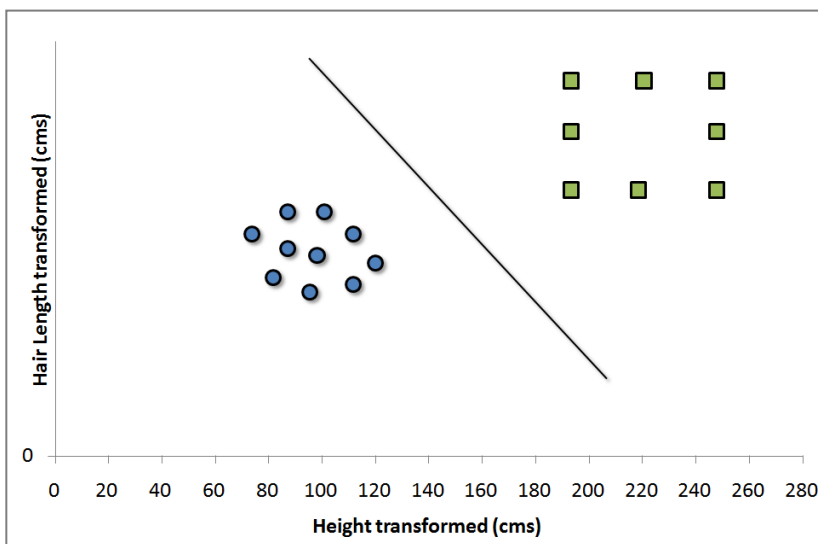


The easiest way to interpret the objective function in a SVM is to find the minimum distance of the frontier from closest support vector (this can belong to any class). For instance, orange frontier is closest to blue circles. And the closest blue circle is 2 units away from the frontier. Once we have these distances for all the frontiers, we simply choose the frontier with the maximum distance (from the closest support vector). Out of the three shown frontiers, we see the black frontier is farthest from nearest support vector (i.e. 15 units).

Our job was relatively easier finding the SVM in this business case.



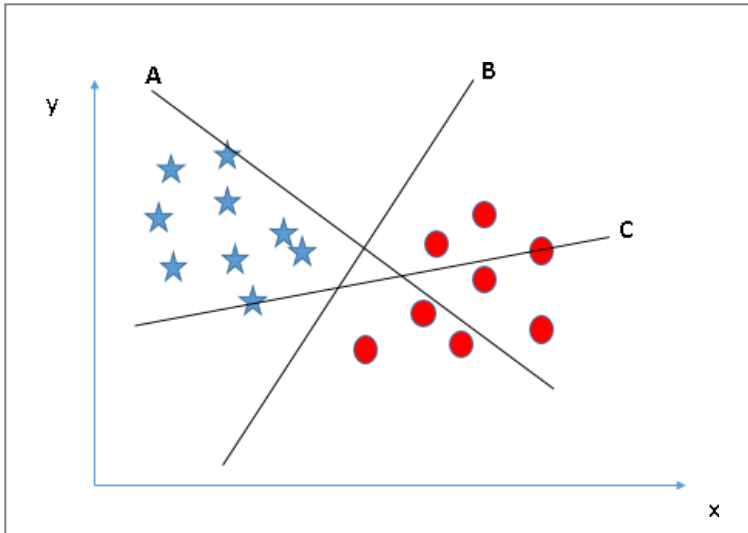
In such cases, we do not see a straight line frontier directly in current plane which can serve as the SVM. In such cases, we need to map these vectors to a higher dimension plane so that they get segregated from each other. Such cases will be covered once we start with the formulation of SVM. For now, you can visualize that such transformation will result into following type of SVM.



Each of the green squares in original distribution is mapped on a transformed scale. And transformed scale has clearly segregated classes. Many algorithms have been proposed to make these transformations.

Above, we got accustomed to the process of segregating the two classes with a hyper-plane. Now the burning question is “How can we identify the right hyper-plane?” Don’t worry; it’s not as hard as you think!

- Let’s understand: Identify the right hyper-plane (Scenario-1): Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.

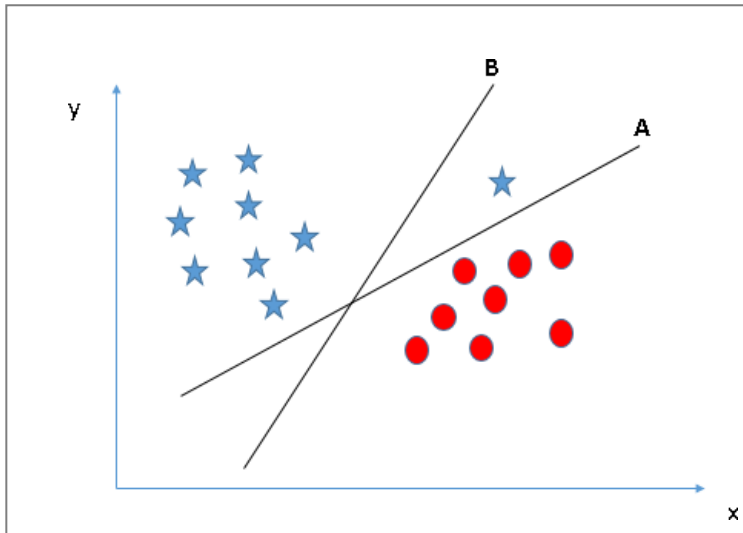


You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

Identify the right hyper-plane: Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, how can we identify the right hyper-plane. Here, maximizing the distances between nearest data points (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin.

Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

Identify the right hyper-plane: Hint: Use the rules as discussed in previous section to identify the right hyper-plane



Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A**. But, here is the catch; SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A**.

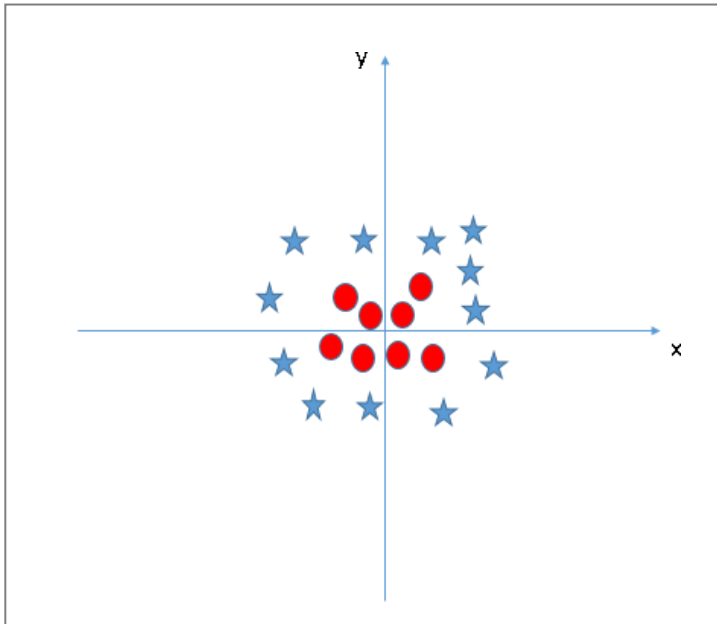
Can we classify two classes : Below, I am unable to segregate the two classes using a straight line, as one of star lies in the territory of other(circle) class as an outlier.

As I have already mentioned, one star at other end is like an outlier for star class. SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to outliers.

Find the hyper-plane to segregate two classes: In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.

SVM can solve this problem. Easily! It solves this problem by

introducing additional feature. Here, we will add a new feature $z=x^2+y^2$. Now, let's plot the data points on axis x and z:



In above plot, points to consider are:

- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z .

In SVM, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, SVM has a technique called the kernel trick. These are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then find out the process to separate the data based on the labels or outputs you've defined.

3.4.1 Pros and Cons associated with SVM

Pros:

- It works really well with clear margin of separation
- It is effective in high dimensional spaces.
- It is effective in cases where number of dimensions is greater than the number of samples.
- It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Cons:

- It doesn't perform well, when we have large data set because the required training time is higher
- It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping
- SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library

3.5 REQUIREMENT SPECIFICATION

This proposed software runs effectively on a computing system that has the minimum requirements. Undertaking all the equipment necessities are not satisfied but rather exist in their systems administration between the customer's machines already. So the main need is to introduce appropriate equipment for the product. The requirements are split into two categories, namely:

Software Requirements

The basic software requirements to run the program are:

Operating System: Windows 7 and above (64-bit).

Python: 3.6

Hardware Requirements

The basic hardware required to run the program are:

Hard disk: 500 GB and above.

Processor: i3 and above.

am: 4GB and above

TECHNOLOGIES USED:

- Programming Language: Python.
- OAuth
- SVM

3.6 MODULES

The location inference for non-geo tagged tweets model consists of mainly three actors. They are tweet gathering svm model generation and prediction. The primary task of the tweet gathering is to collect live tweets from the twitter. The user uses the credentials to access the twitter so that they can collect tweets. we propose a machine learning based system to automatically identify the user tweet country. An SVM model is used and cultivated with data. The representation is employed to predict the country of origin on the tweet.

- Tweet gathering
- SVM model generation
- Prediction, result presentation

Tweet Gathering

We like to use Python language because it has vast collection of machine learning libraries. First to read live tweet from twitter, we need to use twitter API for python. To access the Twitter server we will be using OAuth for authentication. Once we generate keys from Twitter server, we can use these keys to authenticate of program to access twitter server. We use search API to search tweets based on contextual text. A Python API is used to access Twitter server for searching tweets. Generating Access Key and consumer key from Twitter website to access tweets.

Initiating OAuth (Open Authorization) with the access key to login to twitter server for tweet search. OAuth allows the python to access the twitter server without

exposing the username and password. We like to utilize Python language because it consists of various AI libraries. First to read live tweet from twitter, need to capitalize twitter API for python. To enter into the Twitter server need to capitalize on the OAuth for validation.

While we bring out keys from Twitter server, use these keys for entering into to server by verifying program. Then using search API for the tweets which considers a unique content. The majority of the tweets usually can be divided into 3 parts, not specifically in the same order.

SVM Model Generation

We use Support Vector Machine (SVM) algorithm to train our system. Support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. SVM training algorithm builds a model that assigns new examples to one category or the other. We train our system with dataset and create the model for future prediction.

Support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. SVM training algorithm builds a model that assigns new examples to one category or the other. We train our system with dataset and create the model for future prediction.

We use Support Vector Machine (SVM) calculation to prepare our framework. Bolster vector machines are regulated learning models with related learning calculations that investigate information utilized for grouping and relapse examination. SVM preparing calculation assembles a model that appoints new guides to one class or the other. We train our framework with information set and make the model for future expectation.

The first part contains the people whom the tweet is intended to or who is it referencing. This is usually denoted by "@". The second part contains the actual message. This is what the user actually wants to convey. The third part contains the hash tag denoted by "#". This is usually to categorize the tweets. This can also

be used by others to find tweets related to the particular content. The tweets are pre-processed to filter the first part and third part since they do not hold very less to zero significance in sentiment analysis, later, the remaining message part of the tweet is pre-processed further for obtaining useful keywords which accounts much significance in identifying the location.

It doesn't perform well, when we have large data set because the required training time is higher. It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping. SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library.

Prediction and Result

SVM training takes time directly proportional to size of the data. Instead of training our model each and every time we save the trained model and use it for prediction. This saves us time and increases the efficiency. Once the country was identified, we present the tweet with the country of origin. SVM preparing requires some serious energy legitimately relative to estimate of the information. Rather than preparing our model every single time we spare the prepared model and use it for expectation. This spares us time and expands the productivity. When the nation was recognized, we present the tweet with the nation of beginning.

It works really well with clear margin of separation. It is effective in high dimensional spaces. It is effective in cases where number of dimensions is greater than the number of samples. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

CHAPTER 4

IMPLEMENTATION & METHODOLOGY

4.1 Design and Implementation Constraints

4.1.1 Constraints in Analysis

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure

4.1.2 Constraints in Design

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses
- Global actions and Constraint Realization

4.1.3 Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

4.2 Other Nonfunctional Requirements

4.2.1 Performance Requirements

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service.
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

4.2.2 Safety Requirements

1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
6. Systems with different requirements for safety levels must be separated.
7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

4.2 ARCHITECTURE DIAGRAM

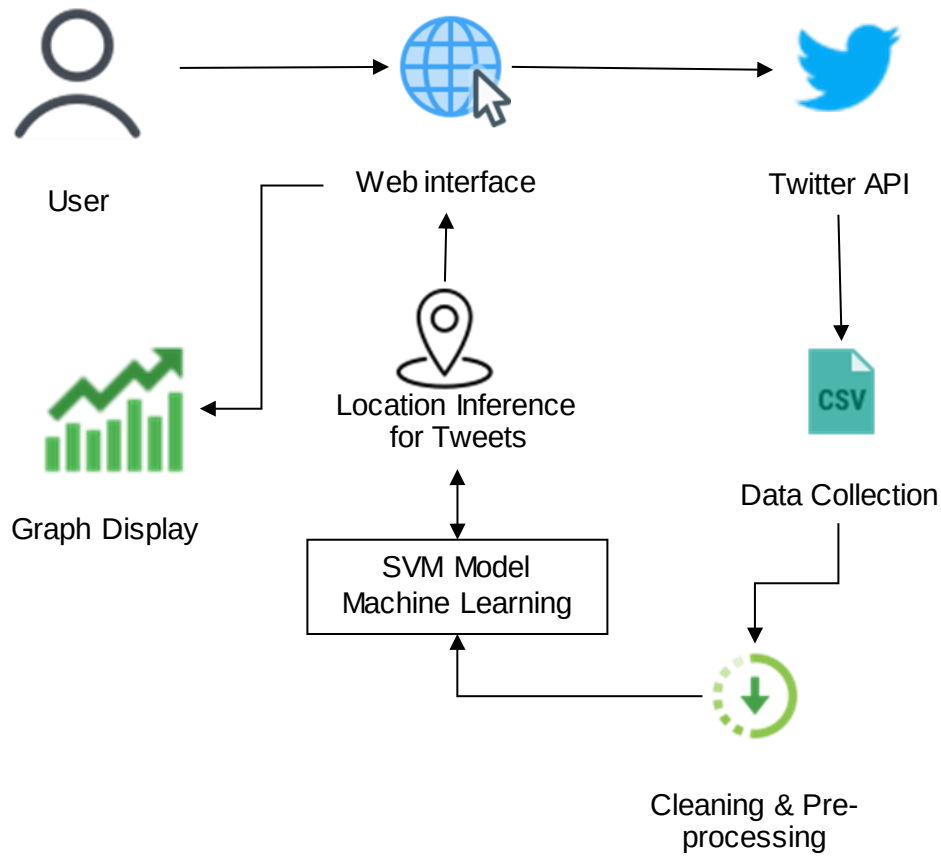


Fig 5.1: Architecture Diagram

First, user goes through web interface which allows user to interconnect with the content in web browser. Then, search for twitter API in web browser and login to account. By using OAuth authentication and streaming API the user can collect the data from twitter with python library. The data which is collected from twitter will be in cleaning and pre-processing. after cleaning and pre-processing and the data will run through an algorithm called SVM (Support Vector Machine) this algorithm is used for both classification and regression. From this algorithm the location for particular tweets will be found. It will also find from which countries most of those tweets are tweeted and finally the graph displays the country names and the amount of tweets from a country is displayed by a graph.

4.2.1 Sequence Diagram

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

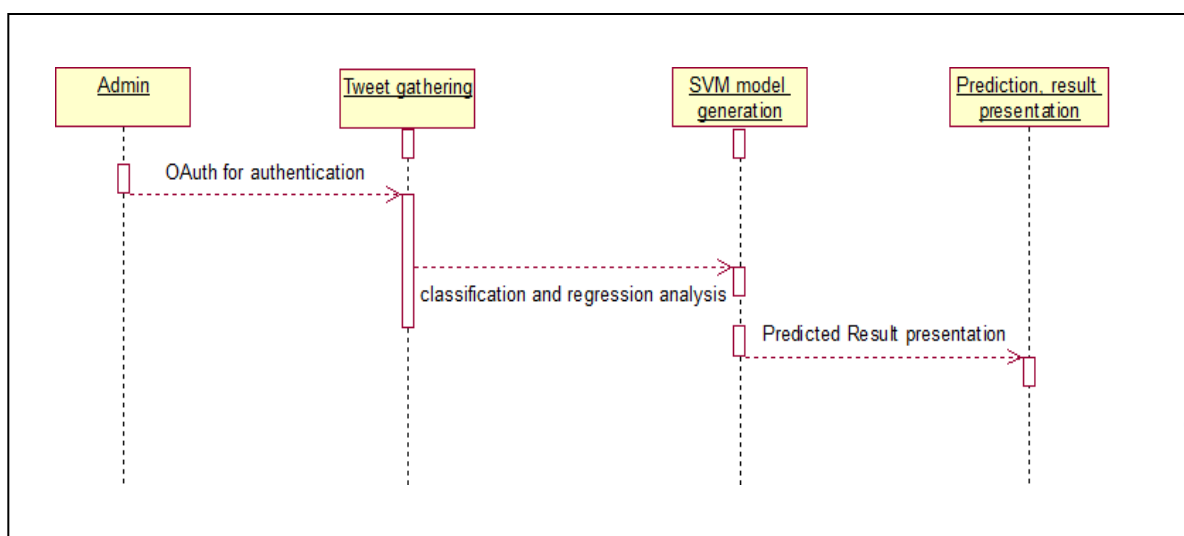


Fig 5.2:

4.2.2 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

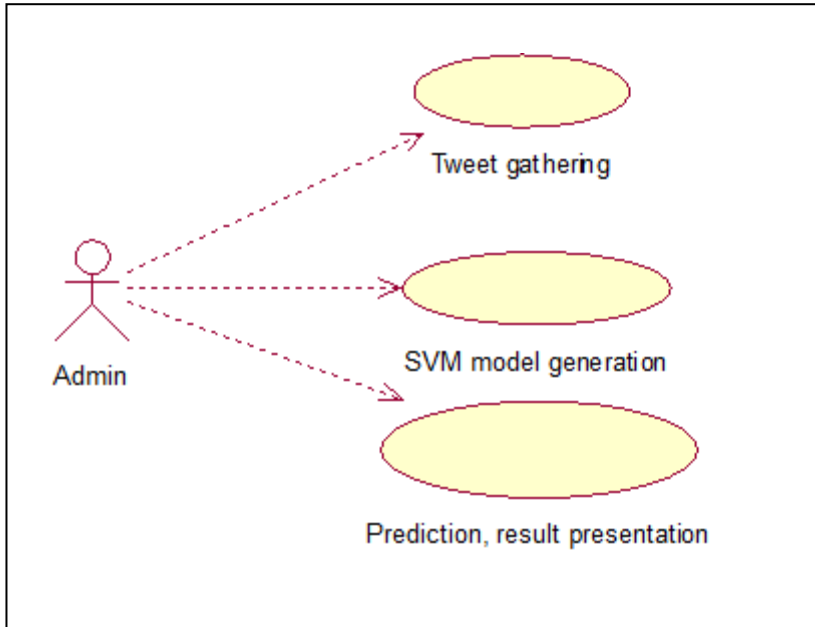


Fig 5.3:

4.2.3 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

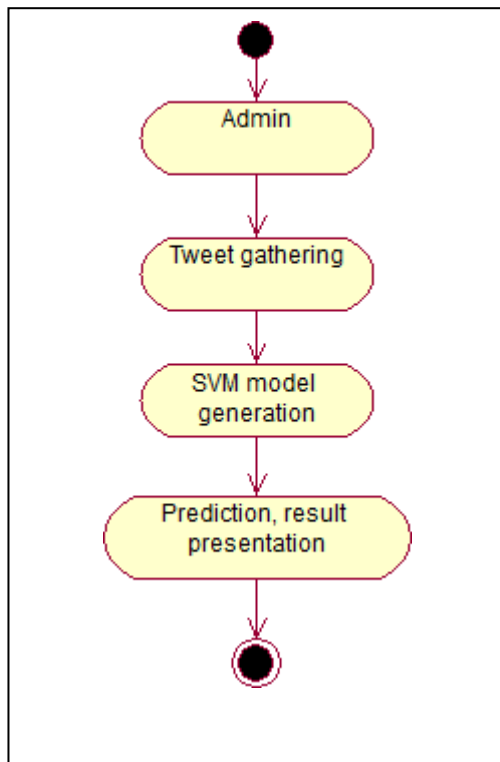


Fig 5.4

4.2.4 Collaboration Diagram:

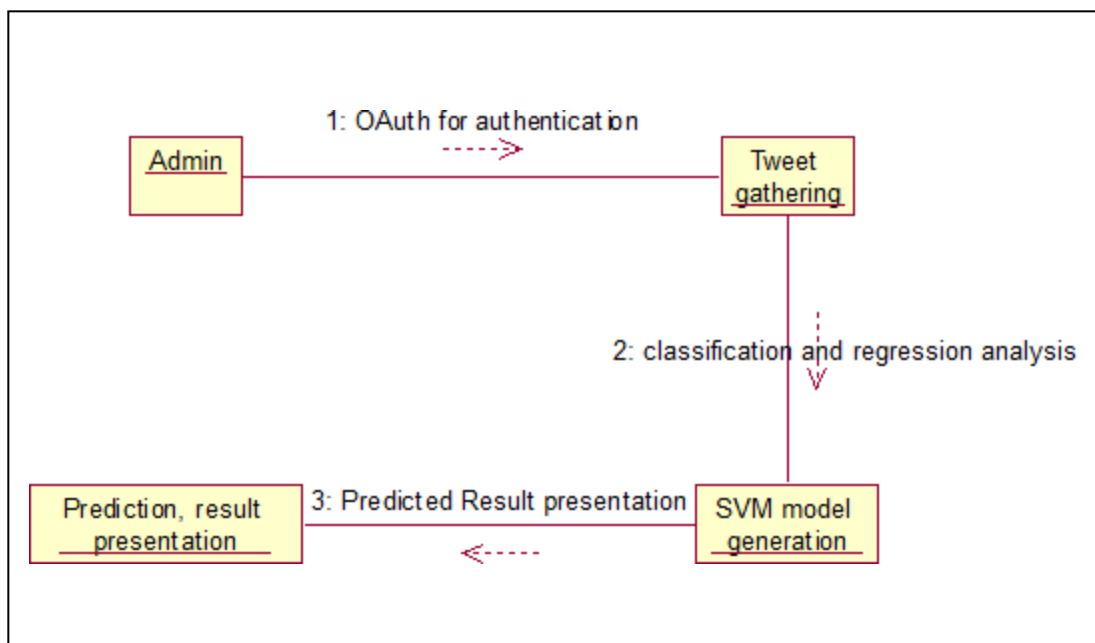


Fig 5.5

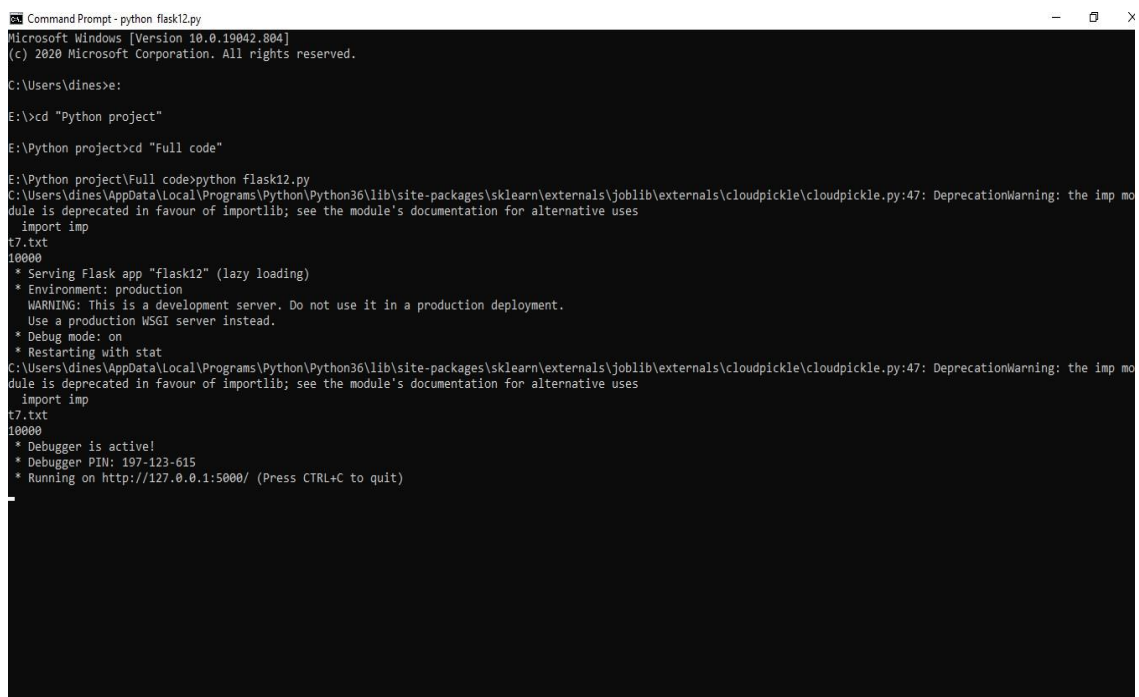
UML Collaboration Diagrams illustrate the relationship and interaction between software objects.

They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

CHAPTER 5

RESULTS AND DISCUSSION

As a result of this system getting introduced in our tweets system, it will act like a bridge to cover the gap of communication between the user and twitter. We implement machine learning algorithm SVM and inferred the country of the tweet to enhance the location based services quality and the flexibility to the business.



```
Command Prompt - python flask12.py
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\dines>
E:\>cd "Python project"
E:\Python project>cd "Full code"
E:\Python project\Full code>python flask12.py
C:\Users\dines\AppData\Local\Programs\Python\Python36\lib\site-packages\sklearn\externals\joblib\externals\cloudpickle\cloudpickle.py:47: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  import imp
t7.txt
10000
* Serving Flask app "flask12" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
C:\Users\dines\AppData\Local\Programs\Python\Python36\lib\site-packages\sklearn\externals\joblib\externals\cloudpickle\cloudpickle.py:47: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  import imp
t7.txt
10000
* Debugger is active!
* Debugger PIN: 197-123-615
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig 5.1: Execution Page

The Fig 5.1 is the execution part of project by opening the cmd and the part of code will be executed. The backend of code will be running by selecting the location of code. It also contains the part from where we can get various types or different types of information. It will collect the live tweets from twitter and takes it to the unstructured data to structured data.

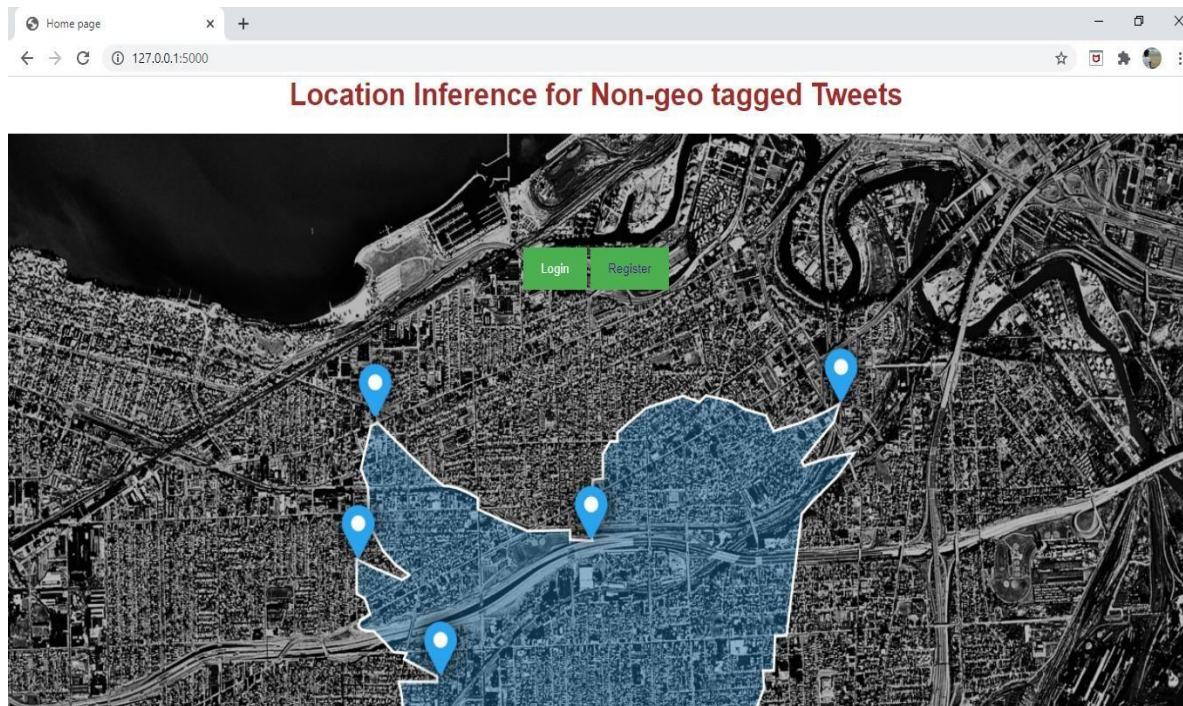


Fig 5.2: Home Page

The Fig 5.2 is the news-feed of the project. This page will appear once the user or the admin make a log-in into the portal. The news feed keeps scrolling as the user starts to work on it. Here the user the gets the multiple options to execute or access their task as per need. In this system the user interface build is very attractive and professional. It is the frontend of the project or it can also be termed as user interface. Here the user the gets the multiple options to login or register as per need. The system has access to the login of the page which is created for the frontend of page. This page will appear once the user or the admin make a log-in into the portal. It is the frontend of the project or it can also be termed as user interface. The admin of the system can only control, make changes and maintain the databases. The admin provides the required security to the users so that the data does not get deleted or leaked. This section checks the authentication and if the log in credentials is correct then the respective user are able to access the page.

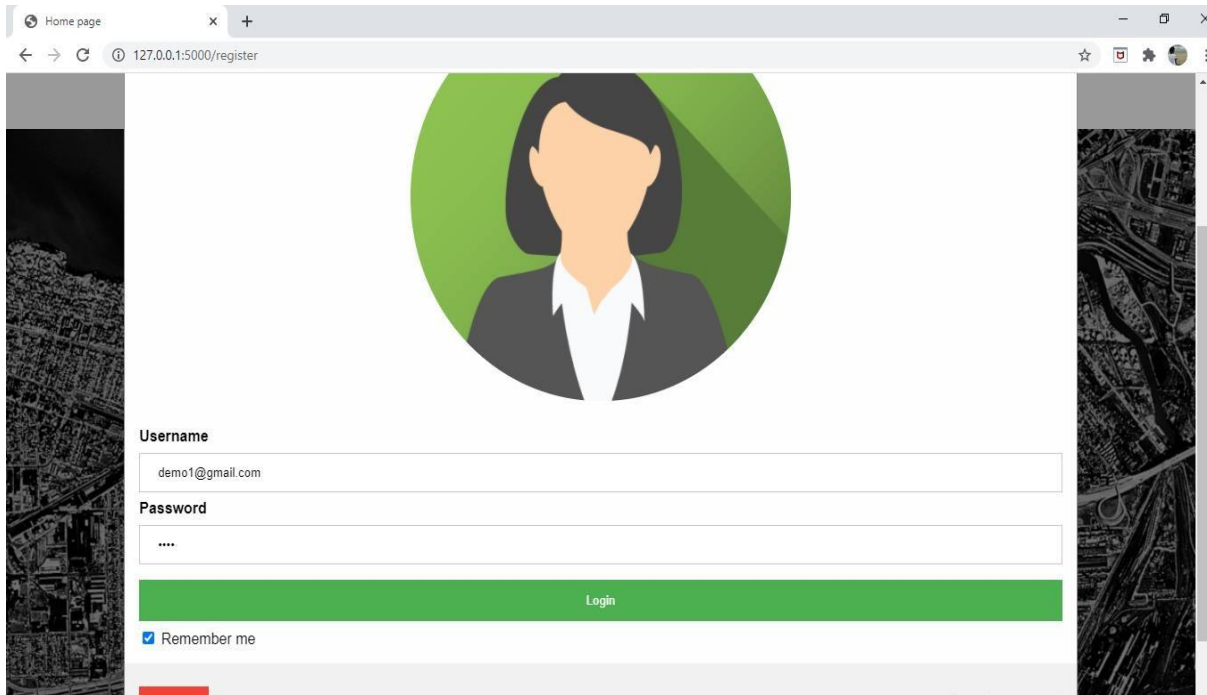


Fig 5.3: Log-In Form

The Fig 5.3 is the log-in page which generally is the entry or authentication of every secured portal. This portal can be logged in by both user and the person have access. The admin can log-in to check the working of the portal and to maintain the database. The user log-in to access the portal and get required information. The page provides the required security to the users so that the data does not get deleted or leaked. This section checks the authentication and if the log in credentials is correct then the user can log in and know the information. It is the frontend of the project or it can also be termed as user interface. Here the user the gets the multiple options to execute or access their task as per need.. This page will appear once the user or the admin make a log-in into the portal. It is the frontend of the project or it can also be termed as user interface. This section checks the authentication and if the log in credentials is correct then the user can log in and report crimes. This module keeps the whole system secure and it doesn't allow any unauthorized users to access the system.

127.0.0.1:5000/login

Location Inference for Non-geo tagged Tweets

Inferring Location for Tweets

Please Search for a tweets using hashtag

tweet barchat

Fig 5.4: TWEETS SEARCH

The Fig 5.4 is the search form and it can be called the heart of the project. This is the cause for which we build the project. Here in the report form the tweets can collect location by giving the hashtags or the tweets information. The page shows the search of the tweets for the location of non-geotagged tweets. It uses the svm training takes time directly proportional to size of the data. Instead of training our model each and every time we save the trained model and use it for prediction. This saves us time and increases the efficiency. Once the country was identified, we present the tweet with the country of origin The user interface contains the log in or registers form. Here by giving the tweets and the hashtags we can search know the location of non-geotagged tweets .

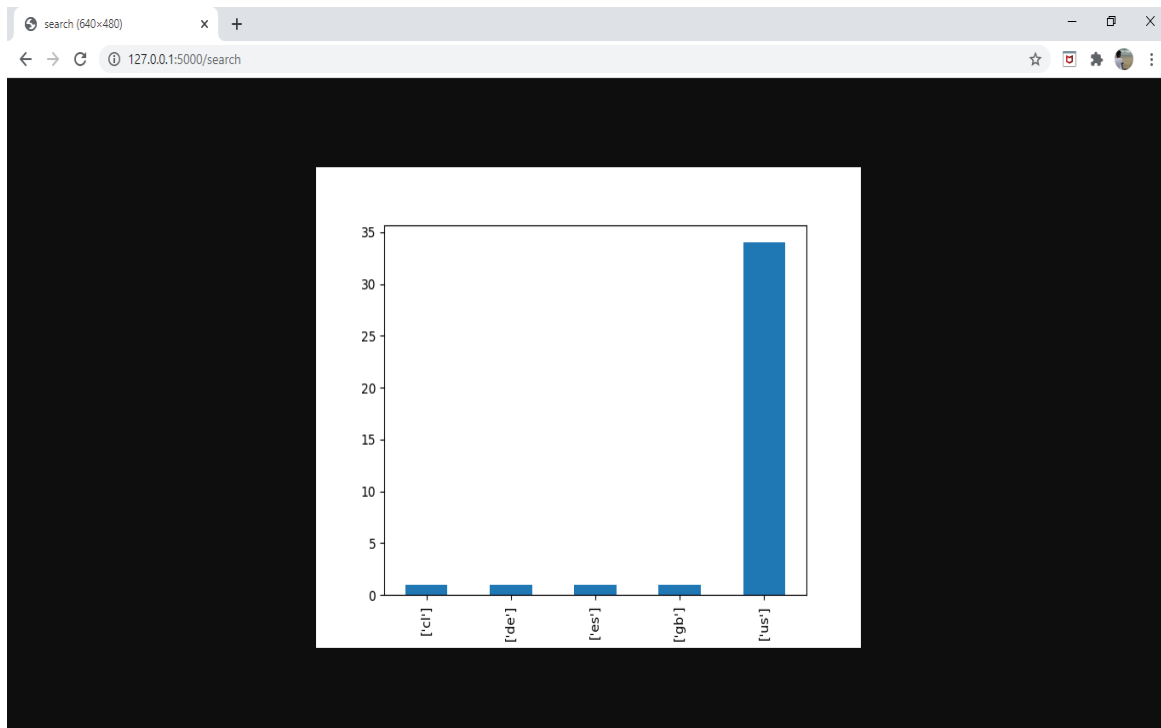


Fig 5.5: Output Form

The fig 5.5 is the final output which shows the countries of the tweets or the hashtags in which the output or the final verdict gives the non-geotagged tweets. The final output shows in the bar graph format so that the trending and the happenings across the world in a particular country. This makes the tweet topic to be considered as the trending in that particular country. We can test by giving the other Caribbean or the some of countries which will be popular in the particular country. So that the location inference of tweets displayed in the bar graph format which will be executed by an active network to make sure these are from live twitter.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

The proposed models right now tweet substance as it were. It is interesting to think about other data, for instance, social relationship among customers and normal models shared by customers. Right when gotten together with tweet substance, such data may be utilized to improve shockingly area deductions. Furthermore it is possible to use the couple of geo-labeled tweets in a customer's course of occasions through spatio transient goals, in the longing for improving or encouraging area induction for non-geotagged tweets. We execute Machine learning calculation SVM and incited the country of the tweet to redesign the area based organizations quality and the flexibility to the business.

The system consists of three functional modules: a knowledge capture module, search module, and control module, and a knowledge utilization module. The system maintains an database of tweets for conclusion of future work.

6.2 FUTURE WORK

The proposed models in this use tweet contents only. For future work, it is interesting to consider other information such as social relationship among users and frequent patterns shared by users. We conduct extensive experiments using large real datasets collected from Twitter. The experimental results demonstrate that IG-Bayes and BiLSTM-C achieve high location inference accuracy in multiple settings and clearly outperform the state-of-the-art and alternative approaches. When combined with tweet contents, such information may be utilized to make even better location inferences. Also, it is possible to make explicit use of the few geo-tagged tweets in a user's timeline, e.g., through spatio-temporal constraints, in the hope of improving or easing location inference for non-geotagged tweets.

REFERENCES

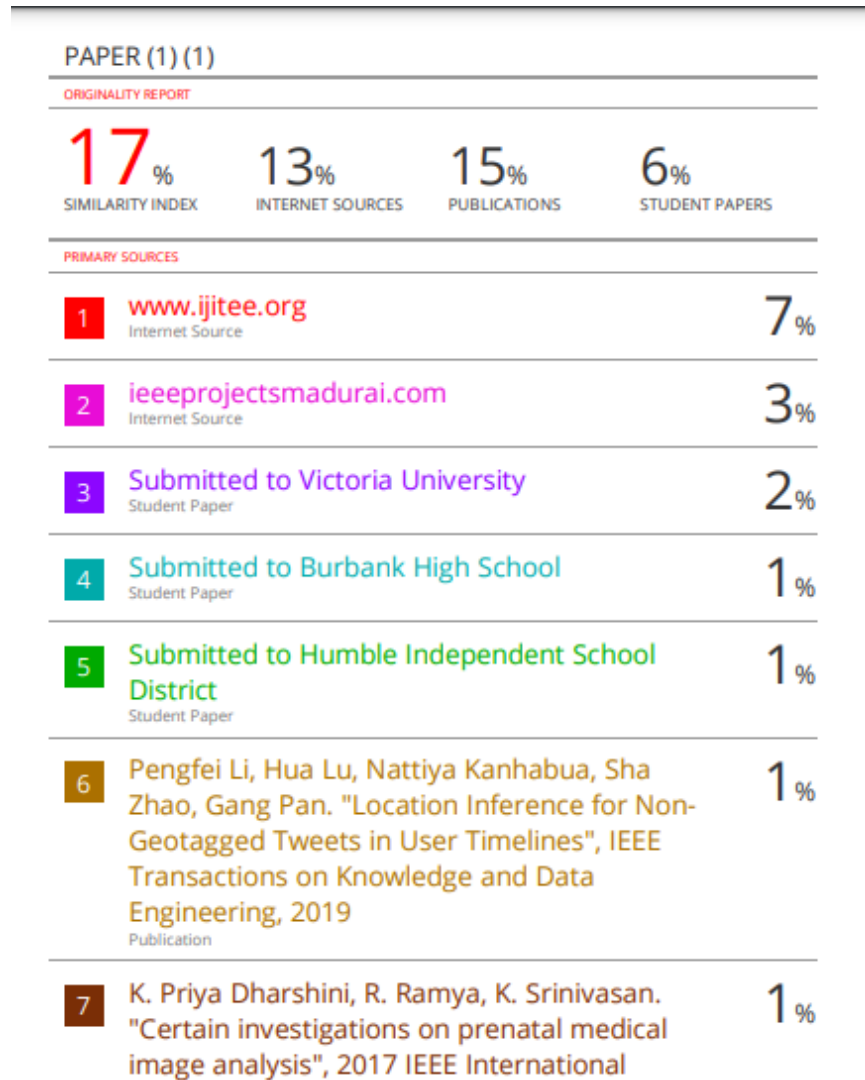
- [1] De Wang, Danesh Irani and Calton Pu. A social spam detection framework. In proceedings of the Eight annual collaboration, Electronic messaging, AntiAbuse and Spam Conference (CEAS 2011), 2011.
- [2] Benjamin Markines, Ciro Cattuto and Filippo Menczer. Social Spam Detection. ACM-2009.
- [3] Enhua Tan, Lei Guo, SongQing, Xiaodong Zhang and Yihong Zhan. UNIK: Unsupervised Social Network Spam Detection. ACM-2013.
- [4] Xueying Zhang, Xianghan Zheng. A Novel Method for Spammer Detection in Social Networks. IEEE-2015.
- [5] Faraz Ahmed, Muhammad Abulaish, An MCL based approach for spam profile detection in online social networks, 11th international conference on trust, security and privacy in computing and communications, IEEE2012.
- [6] cheng cao and James Caverlee, Detecting Spam URLs in Social Media via Behavioral Analysis. In springer 2015
- [7] Hailu Xu, Weiqing Sun, Ahmad Javaid, Efficient Spam Detection across online social networks, IEEE-2015
- [8] Xianghan Zheng, Zhipeng Zeng, Zheyi chen, Yuanlong Yu, Chunming Rong, Detecting spammers on social networks, Elsevier-2015
- [9] koJ. Li, T. Sellis, J. S. Culpepper, Z. He, C. Liu, and J. Wang, "Geo social influence spanning maximization," IEEE Trans. Knowl. Information Eng., vol. 29, no. 8, pp. 1653–1666, Aug. 2017.
- [10] S. Roller, M. Speriosu, S. Rallapalli, B. Wing, and J. Baldrige, "Supervised text-based geolocation using language models on an adaptive grid," in Proc. 2012 Joint Conf. Empirical Methods in Natural LanguageProcessing and Computational Natural

Language Learning, 2012, pp. 1500–1510. 1334

- [11] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing, "A latent variable model for geographic lexical variation," in Proc. Conf. Empirical Methods Natural Lang. Process., 2010, pp. 1277–1287.
- [12] Eisenstein, A. Ahmed, and E. P. Xing, "Sparse additive generative models of text," in Proc. 28th Int. Conf. Int. Conf. Mach. Learn., 2011, pp. 1041–1048.
- [13] Urus Parry (2007-07-19). "How to improve your security when using a public terminal (part 3 of 4)". Windows secrets. Retrieved 2010-06-15.
- [14] Usha Richards,b(2015, June 8) EyeWitness app lets smartphones report war crimes. <http://www.bbc.com/news/technology-33029464> Date of extraction: June 12, 2015.
- [15] Vian Brown, "The regional crime analysis program (RECAP): A Frame work for mining data to catch criminals," in Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Vol. 3, pp. 2848-2853,1998.
- [16] Vladimir Shane September-October 2013. Smartphone: Android Vs IOS <http://www.thesij.com/papers/CSEA/2013/September-October/CSEA-01.pdf> Date of extraction: June 19, 2015.
- [17] ApacheTomcat<http://tomcat.apache.org>.
- [18] Boba, R. (2005). Crime Analysis and Crime Mapping.
- [19] MySQL<http://www.mysql.com>.
- [20] UMLofficialwebsite<http://www.uml.org/>.

APPENDIX

A. PLAGIARISM REPORT



B. Journal Paper

AN EFFECTIVE WAY TO INFERENCE LOCATION INFORMATION A NON-GEOTAGGED TWEET

DINESH RATNA U [1], SANDEEP REDDY P [2], AISHWARYA R [3]

[1][2] UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

[3] Associate professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

Abstract: Social media-based life sway on society is developing step by step. With all the information drifted on social media empowers the business to offer best types of assistance to client. One such help is location-based substance or future conveyance. Not all Tweets were geo-tagged. To induce the client location with the assistance of the tweet to guarantee the clients were served well. To expand fame of the person-to-person communication administration, Twitter, has made it progressively associated with everyday interchanges, fortifying social connections and information spread. This pattern has added to encourage different location put together administrations sent with respect to social media-based social networking, the accomplishment of which intensely relies upon the accessibility and exactness of clients' location information. Thusly, it is essential to induce areas for tweets to accomplish the explanation behind the area-formed organizations. At the present time, handle this issue by looking at Twitter customer timetables in a novel way. As an issue of first significance, we split each customer's tweet course of occasions temporarily into different gatherings, each tending to recommend a specific area. Thusly, we modify two Machine learning styles to our design and structure category that describe individual tweet bundle into particular pre-portrayed area groups in burghal. Our framework utilizes the clients content substance and different properties of the tweet like time zone, language and so forth to recognize the tweet nation. A SVM model is prepared with given information set. This model is utilized to foresee the nation of cause on the tweet.

KEYWORDS: Twitter, Social Media, SVM, Location Inference.

I INTRODUCTION

Social media-based life organizing destinations, for example, facebook, Twitter and so forth permit clients to share sees, photographs and recordings, posts, and to advise others about on the social media or true exercises. The accomplishment of long-range interpersonal communication administrations can be found in strength in today's society with Facebook having a gigantic 2.13 billion dynamic month to month clients and a normal of 1.4 billion day by day dynamic clients in 2017. Some informal communication administrations expect individuals to have a

most microblogging administrations in social media-based life organizing social media site with huge number of clients who associate to one another through this system. These clients share their thoughts, conclusions, specialized information, sees and their closely-held convictions on explicit occasions in the general public and are shared through the twitter messages. Twitter is one of the most conspicuous person to person communication applications to frame same local gatherings between specialized experts, loved ones and business people gr

individuals express their perspectives, news to different individuals in the network through twitter. Twitter messages are limited to 280 characters. Twitter organizing site permits the clients to follow their preferred researchers, specialists and other famous people. A client can make his own record in arrange transparently with no limitations, essentially giving his work force subtleties like name, staff id and address. Because of this open getting to approach into twitter arrange numerous clients abuse the system exercises. They basically misdirect the ordinary clients through retweets, url connections and hash labels. Clients of the Twitter arrange have various degrees of Knowledge as for security dangers covered up in online life systems. Spammers are pulled in by twitter organize as a supporting instrument to spread spam messages, commercials to certified clients. The spammers likewise send urls and pernicious connects to the authentic clients. Spam is probably the most concerning issue in social media based life destinations like faebook, linkedIn. Scientists shows that over 3% of tweets are spam messages. The drifting subjects are likewise assaulted by spammers.

Location inference for tweets are tested by two significant issues. Initially, Twitter constrains the length of each tweet substance to 140 characters, and along these lines a tweet just contains few words and passes on restricted information. Second, Twitter clients regularly use non-standard and short-hand terms, and tweets are frequently vague and boisterous. Consequently, discovering location hints from short, boisterous tweets is undeniably troublesome. Right now, research how to surmise the locations of non-geotagged tweets at the city level by investigating Twitter clients' courses of events utilizing a novel methodology. Our methodology joins examination on the substance of tweet short messages and that on the client courses of events with

fleeting information. Along the fleeting measurement, every client course of events is part into various tweet groups; each bunch suggests a particular client location. This procedure is called worldly bunching of tweets.

Along these lines, two AI models are painstakingly adjusted to our concern setting and classifiers are intended to arrange each tweet group from a client's course of events into one of the pre-characterized location classes at the city level. The Bayes put together model concentrations with respect to the information increase of words with location suggestions in the client produced substance, while the LSTM based model treats client created substance and their related locations as arrangements and utilizes a bidirectional LSTM [17] and convolution operation to make location inductions. Our models are prepared utilizing disconnected information, yet they can be utilized to deduce locations for chronicled tweets and social media based (approaching) tweets. The two models are tentatively assessed on an enormous genuine informationset, in correlation with elective methodologies. The trial results recommend that the proposed models are powerful at gathering locations for tweets and they beat choices fundamentally as far as inference precision. Our commitments right now abridged as follows. We structure transient bunching strategies that split a client's tweet timetable into a lot of groups every one of which contains tweets that are likely sent from a similar city. We structure a Bayes' hypothesis based model for location inference for tweet groups. The model estimates words' land scopes by processing words' information gains over all locations of intrigue. We fabricate a novel neural system that joins convolution activity and long transient memory unit when removing highlights from the substance of tweet groups. It can misuse spatially-nearby relationship [18], [19], [20] while gathering locations for tweet

groups. We assess the presentation of our proposed approach and models utilizing certifiable Twitter information. The outcomes show that our methodology with the models beats best in class options.

II. RELATED WORK

Twitter is one of the ordinary long range casual correspondence goals like facebook, Myspace and linkedIn. Twitter arranging site is one of the short illuminating organization where customers can send short messages to their friends. A customer of the twitter orchestrate is recognized by username and besides perceived by authentic name. A customer can follow another customer, consequently that customer gets messages in solitude page. One customer can follow another customer who needs. Messages can be accumulated using hashtags beginning with a "#" extraordinary personality. Hashtags grant customers to glance through tweets reliant on interest. Right when a customer likes someone customers tweet, he can in like manner retweet that message. In this manner that sent message is appeared to all of his supporters. A web based life based life mastermind customer can shield his profile from various customers. The nature of online life frameworks rely upon organize trust. To look at spam disclosure in casual networks various investigators proposed various systems. De *Wang et al [1]* proposed a spam area diagram work. At the present time they portray models specifically profile model, message model and page model. They inspect cross social corpora gathering and familiar game plan. In Benjamin *Markines et al[2]*, makers recognize and dismember six particular features to perceive spam. They use these features to perceive spammers from genuine customers. They used direct features like TagSpam, TagBlur, DomFp, NumAds, Plagiarism and ValidLinks. They used SVM and AdaBoost guided learning estimations

to create social spam identifiers subject to these features. *Enhua Tan et al [3]* proposed an independent relational association spam ID theory to recognize spammers. The proposed UNIK plan can recognize spammers with sham positive pace of 0.6% and fake negative pace of 3.7%. UNIK plot has a comparative level of revelation execution as SD2.

Hailu et al [7] talk about a spam recognizable proof methodology utilizing facebook and twitter information sets. They utilized about normal highlights between two information sets to identify spam. Right now join twitter spam information with facebook information for preparing and consolidate facebook spam information with twitter information for preparing. Random woodland classifier shows the best execution contrast with other conventional classifiers. In *Xianghan Zheng et al[8]* creators talk about one of the managed machines learning way to deal with recognize spam in social media based life systems. Right now content based highlights and client based highlights to identify spammers. To execute this methodology they utilized SVM classifier. This approach has given most elevated accuracy, review and F measure.

As casual associations are dynamic and their topology changes, foreseeing customers' areas subject to relational associations may stand up to inconveniences. Right now, approaches make the area enlistments simply using customer created substance. Most substance based systems intend to make a probabilistic structure and choose customer area by most prominent likelihood. *H.Po et al [20]* use language models and data recuperation frameworks for area desire. Confining the space into cross sections, the approach contemplates the transport of words in a given customer's tweets to those in each system cell using

Kullback-Leibler (KL) uniqueness to recognize the customer's most likely area. **A. Ahmed et al. [11]** develop geographic subject models to find the home areas of Twitter customers to the extent regions and states. The ensuing works use sparse added substance models to join area express, customer unequivocal and non-instructive focuses. **Ryoo et al [15]** apply a comparative plan to Korean Twitter clients. **Hecht et al. [16]** break down the client location field in client profiles and utilize multinomial credulous Bayes to assess client locations at nation and state levels. As of late, **Qian et al. [17]** propose a probabilistic model dependent on factor diagrams for Twitter client's location deduction at the nation level.

III EXISTING SYSTEM

As informal organizations are not static & their topology changes, foreseeing clients' locations dependent on interpersonal organizations may confront troubles. Thusly, more methodologies make the location inductions just utilizing client produced substance. Most substance based methodologies expect to fabricate a probabilistic framework and decide client location by greatest probability. Existing strategies for construing locations in informal organizations fall in two classes: First those deducing a Twitter client's location for example their location inferences are totally founded on tweet substance. A few existing methods utilize interpersonal organizations and companions' locations to construe locations for Twitter clients and Second those inducing locations for tweets. In any case, a solitary tweet might be too short to even consider containing adequate highlights like different tweets that are even posted in a similar location. What's more, individuals frequently post comparable tweets at various occasions and locations.

3.1 DISADVANTAGES OF THE EXISTING SYSTEM

- Cannot find accurate locations

IV PROPOSED SYSTEM

In proposed framework about non geotagged tweets, locations at the city level by investigating twitter clients' courses of events utilizing a novel methodology. Our methodology joins investigation on the substance of tweet short messages and that on the client courses of events with transient information. Along the fleeting measurement, every client timetable is part into various tweet bunches; each group suggests an unmistakable client location. This method is called transient bundling of tweets. In this manner, two AI models are meticulously changed in accordance with our anxiety setting & classifiers have planned to portray each tweet bunch from a customer's course of occasions into one of the pre-described area classes at the city level. To crush the fallback in the present structure we propose an AI based system to automatically recognize the client tweet nation.

Our framework utilizes the content substance and different properties of the tweet like time zone, language and so forth to distinguish the tweet nation. A SVM model is prepared with given information set. This model is utilized to anticipate the nation of beginning on the tweet. Our models are prepared utilizing disconnected information, however they can be utilized to derive locations for verifiable tweets and social media based (approaching) tweets. The two models are tentatively assessed on an enormous genuine information set, in examination with elective methodologies.

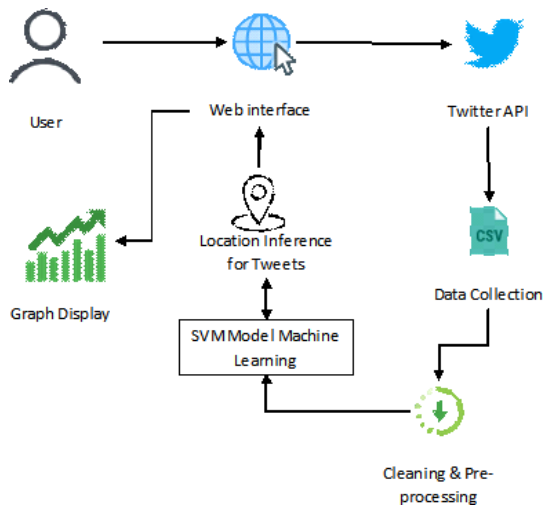


FIG 1 OVERVIEW OF THE PROPOSED SYSTEM

4.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Able to move beyond location changes
- Location of any client can be found with their companion's location

V MODULE DESCRIPTION

5.1 TWEET GATHERING

We like to utilize Python language because it consists of various AI libraries. First to read live tweet from twitter, need to capitalize twitter API for python. To enter into the Twitter server need to capitalize on the OAuth for validation. While we bring out keys from Twitter server, use these keys for entering into to server by verifying program. Then using search API for the tweets which considers a unique content.

5.2 SVM MODEL GENERATION

We use Support Vector Machine (SVM)

calculation to prepare our framework. Bolster vector machines are regulated learning models

with related learning calculations that investigate information utilized for grouping and relapse examination. SVM preparing calculation assembles a model that appoints new guides to one class or the other. We train our framework with information set and make the model for future expectation.

5.3 PREDICTION, RESULT PRESENTATION

SVM preparing requires some serious energy legitimately relative to estimate of the information. Rather than preparing our model every single time we spare the prepared model and use it for expectation. This spares us time and expands the productivity. When the nation was recognized, we present the tweet with the nation of beginning.

VI CONCLUSION

The proposed models right now tweet substance as it were. It is interesting to think about other data, for instance, social relationship among customers and normal models shared by customers. Right when gotten together with tweet substance, such data may be utilized to improve shockingly area deductions. Furthermore it is possible to use the couple of geo-labeled tweets in a customer's course of occasions through spatio transient goals, in the longing for improving or encouraging area induction for non-geotagged tweets. We execute Machine learning calculation SVM and incited the country of the tweet to redesign the area based organizations quality and the flexibility to the business.

REFERENCES

- [1]. De Wang, Danesh Irani and Calton Pu. A social spam detection framework. In proceedings of the Eight annual collaboration

Electronic messaging, AntiAbuse and Spam Conference (CEAS 2011), 2011.

[2]. Benjamin Markines, Ciro Cattuto and Filippo Menczer. Social Spam Detection. ACM-2009.

[3]. Enhua Tan, Lei Guo, SongQing, Xiaodong Zhang and Yihong Zhan. UNIK: Unsupervised Social Network Spam Detection. ACM-2013.

[4]. Xueying Zhang, Xianghan Zheng. A Novel Method for Spammer Detection in Social Networks. IEEE-2015.

[5]. Faraz Ahmed, Muhammad Abulaish, An MCL based approach for spam profile detection in online social networks, 11th international conference on trust, security and privacy in computing and communications, IEEE2012.

[6]. cheng cao and James Caverlee, Detecting Spam URLs in Social Media via Behavioral Analysis. In springer 2015

[7]. Hailu Xu, Weiqing Sun, Ahmad Javaid, Efficient Spam Detection across online social networks, IEEE-2015

[8]. Xianghan Zheng, Zhipeng Zeng, Zheyi chen, Yuanlong Yu, Chunming Rong, Detecting spammers on social networks, Elsevier-2015

[9] J. Li, T. Sellis, J. S. Culpepper, Z. He, C. Liu, and J. Wang, "Geo social influence spanning maximization," IEEE Trans. Knowl. Information Eng., vol. 29, no. 8, pp. 1653–1666, Aug. 2017.

[10] S. Roller, M. Speriosu, S. Rallapalli, B. Wing, and J. Baldridge, "Supervised text-based geolocation using language models on an adaptive grid," in Proc. 2012 Joint Conf. Empirical Methods in Natural Language

Processing and Computational Natural Language Learning, 2012, pp. 1500–1510. 1334

[11] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing, "A latent variable model for geographic lexical variation," in Proc. Conf. Empirical Methods Natural Lang. Process., 2010, pp. 1277–1287.

[12] J. Eisenstein, A. Ahmed, and E. P. Xing, "Sparse additive generative models of text," in Proc. 28th Int. Conf. Mach. Learn., 2011, pp. 1041–1048.

[13] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsoulouklis, "Discovering geographical topics in the twitter stream," in Proc. 21st Int. Conf. World Wide Web, 2012, pp. 769–778.

[14] Y. Chen, J. Zhao, X. Hu, X. Zhang, Z. Li, and T.-S. Chua, "From interest to function: Location estimation in social media," in Proc. 27th AAAI Conf. Artif. Intell., 2013, pp. 180–186.

[15] K. Ryoo and S. Moon, "Inferring twitter user locations with 10 km accuracy," in Proc. 23rd Int. Conf. World Wide Web, 2014, pp. 643–648

[16] B. Hecht, L. Hong, B. Suh, and E. H. Chi, "Tweets from Justin Bieber's heart: The dynamics of the location field in user profiles," in Proc. SIGCHI Conf. Human Factors Comput. Syst., 1352 2011, pp. 237–246.

[17] Y. Qian, J. Tang, Z. Yang, B. Huang, W. Wei, and K. M. Carley, "A probabilistic framework for location inference from social media," CoRR, vol. abs/1702.07281, 2017. [Online]. Available: <http://arxiv.org/abs/1702.07281>

- [18] H.-w. Chang, D. Lee, M. Eltaher, and J. Lee, “@phillies tweeting 1358 from Philly? predicting twitter user locations with spatial word usage,” in Proc. Int. Conf. Adv. Soc. Netw. Anal. Mining, 2012, pp. 111–118.
- [19] Z. Liu and Y. Huang, “Where are you tweeting?: A context and user movement based approach,” in Proc. 25th ACM Int. Conf. Inf. Knowl. Manage., 2016, pp. 1949–1952.
- [20] H. Bo, P. Cook, and T. Baldwin, “Geolocation prediction in social media data by finding location indicative words,” in Proc. COL1366 ING, 2012, pp. 1045–1062.

Signed Bonafide Page

C.Source Code

```
from __future__ import print_function
import numpy as np
import json
from scipy.sparse import csc_matrix
from sklearn import metrics
import sys, os
import pandas as pd
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.externals import joblib
import json
import operator
from pprint import pprint
import gensim
import re
import codecs
from tweepy import OAuthHandler
import tweepy
from samplegeo import findCountry
import matplotlib.pyplot as plt

def text_clean(text):

    remove_space = '\s+'
    find_url = ('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|'
        '[!*\(\)\,]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
    find_mention = '@[w\.-]+'
    process_text = re.sub(remove_space, ' ', text)
    process_text = re.sub(find_url, "", process_text)
    process_text = re.sub(find_mention, "", process_text)
    process_text = process_text.encode('ascii','ignore')
    return process_text

con_key = 'PG2atN7eFZlfjqPugmrQ0DTHN'
con_secret = 'BcKjeAsjePUjw5SqOj7DqQy7QgTvNo6fbNXoT6PHoV68CwqWSX'
acc_token = '93137600-fD6r6VCcw2f4ahKrqTllmxhFicNYcpy17HkL8Ne4'
acc_token_secret = 'lYUwaYol6j5nHk7pijChJUxzDuY99osg64rs9vjhOejWU'

try:
    auth = OAuthHandler(con_key, con_secret)
    auth.set_access_token(acc_token, acc_token_secret)
    api = tweepy.API(auth)

except:

    print("Error: Twitter Not Connected")
```

```

def getTweetLocation(searchTxt):
    query = "#" + searchTxt
    count = 40
    tweets = []
    clean_tweets = []
    try:
        fetched_tweets = api.search(q = query, count = count)
        cnt = 0;

        for tweet in fetched_tweets:
            parsed_tweet = {}

            parsed_tweet['tweet'] = tweet._json
            parsed_tweet['country'] = findCountry((tweet._json))
            clean_tweets.append(parsed_tweet)
            print("\r Processing...", cnt, "\r")
            cnt = cnt + 1

        df = pd.DataFrame(clean_tweets)
        df.to_csv("clean_tweets.csv")

        # a simple line plot
        df = pd.read_csv("clean_tweets.csv")
        df.groupby('country')['tweet'].nunique().plot(kind='bar')
        plt.savefig("tweet.png")
    #
        plt.show()

except tweepy.TweepError as e:
    print("Error : " + str(e))

print ("*****")
print ("Total tweets read from twitter server ")
print (len(tweets))

```

```

from __future__ import print_function
import numpy as np
from scipy.sparse import csc_matrix
from sklearn import metrics
import sys, os
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.externals import joblib
import json
import operator
from pprint import pprint
import gensim
import re
import codecs
#import io

def get_top_labels_and_feats(top, topinc, filename=""):
    featids = {'content': [], 'name': [], 'tz': [], 'ulang': [], 'uloc': []}
    topcountries = {}
    if os.path.exists('feats/topcountries' + str(top)) and
os.path.exists('feats/content' + str(topinc)) and os.path.exists('feats/name' +
str(topinc)) and os.path.exists('feats/tz' + str(topinc)) and
os.path.exists('feats/ulang' + str(topinc)) and os.path.exists('feats/uloc' +
str(topinc)):
        with open('feats/topcountries' + str(top), 'r') as fh:
            for line in fh:
                data = line.strip().split('\t')
                topcountries[data[0]] = int(data[1])
        with open('feats/content' + str(topinc), 'r') as fh:
            featids['content'] = fh.read().splitlines()
        with open('feats/name' + str(topinc), 'r') as fh:
            featids['name'] = fh.read().splitlines()
        with open('feats/tz' + str(topinc), 'r') as fh:
            featids['tz'] = fh.read().splitlines()
        with open('feats/ulang' + str(topinc), 'r') as fh:
            featids['ulang'] = fh.read().splitlines()
        with open('feats/uloc' + str(topinc), 'r') as fh:

```

```

    featids['uloc'] = fh.read().splitlines()
    return topcountries, featids

```

```

def build_matrix(tweet, featureids, featurelengths, toplabels, istest = 0):

```

```

    colcount = featurelengths['content'] + featurelengths['name'] +
    featurelengths['tz'] + featurelengths['ulang'] + featurelengths['uloc']

```

```

    linecount = 0

```

```

    rows = []

```

```

    cols = []

```

```

    gt = []

```

```

    values = []

```

```

    for a in range(1):

```

```

        if istest == 1 or tweet['location']['address']['country_code'] in toplabels:

```

```

            for ctoken in list(set(re.sub(r'([^\s\w]_)+', ' ', tweet['text'].lower()).split('
            '))):

```

```

                if ctoken in featureids['content']:

```

```

                    rows.append(linecount)

```

```

                    cols.append(featureids['content'][ctoken])

```

```

                    values.append(1)

```

```

                for nametoken in list(set(re.sub(r'([^\s\w]_)+', ' ',
                tweet['user']['name'].lower()).split(' '))):

```

```

                    if nametoken in featureids['name']:

```

```

                        rows.append(linecount)

```

```

                        cols.append(featureids['name'][nametoken] +
                        featurelengths['content'])

```

```

                        values.append(1)

```

```

                if str(tweet['user']['time_zone']) in featureids['tz']:

```

```

                    rows.append(linecount)

```

```

                    cols.append(featureids['tz'][str(tweet['user']['time_zone'])] +
                    featurelengths['content'] + featurelengths['name'])

```

```

                    values.append(1)

```

```

                if tweet['user']['lang'] in featureids['ulang']:

```

```

                    rows.append(linecount)

```

```

                    cols.append(featureids['ulang'][tweet['user']['lang']] +
                    featurelengths['content'] + featurelengths['name'] + featurelengths['tz'])

```

```

        values.append(1)

        for loctoken in list(set(re.sub(r'([^\s\w]_)+', ' ',
tweet['user']['location'].lower()).split(' '))):
            if loctoken in featureids['uloc']:
                rows.append(linecount)
                cols.append(featureids['uloc'][loctoken] +
featurelengths['content'] + featurelengths['name'] + featurelengths['tz'] +
featurelengths['ulang'])
                values.append(1)

            if istest == 0:

gt.append(toplabels[tweet['coordinates']['location']['address']['country_code']]
else:
    gt.append(0)

        linecount += 1
        sys.stdout.flush()

# TODO: tlang

row = np.asarray(rows)
col = np.asarray(cols)
data = np.asarray(values)

return (csc_matrix((data, (row, col)), shape=(linecount, colcount)), gt)

testdata = "t7.txt"
print(testdata)
classifier = "maxent"
output = 'how.txt'
top = int(217)

topinc = int(10000)
print(topinc)

def findCountry(tweets):
    toplabels, featids = get_top_labels_and_feats(top, topinc)
    featureids = {'content': {}, 'name': {}, 'tz': {}, 'ulang': {}, 'uloc': {}}
    featurelengths = {'content': 0, 'name': 0, 'tz': 0, 'ulang': 0, 'uloc': 0}

```

```

for ftype in ['content', 'name', 'tz', 'ulang', 'uloc']:
    fid = 0
    for feat in featids[ftype]:
        featureids[ftype][feat] = fid
        fid = len(featureids[ftype])
    featurelengths[ftype] = len(featureids[ftype])

cmodel = joblib.load('models/maxent-217-10000.pkl')

test_data, test_gt = build_matrix(tweets, featureids, featurelengths,
toplabels, 1)

predicted = cmodel.predict(test_data)
rest=[]
for conn in predicted:
    rest.append(list(toplabels.keys())[list(toplabels.values()).index(conn)])

return rest

```