Pattern Analysis on Twitter Data

Rahul Chhapgar
Department of Computer Science
University at Albany, State University of New York
Albany, New York, USA.
rchhapgar@albany.edu

***Abstract*— Twitter is a platform which is widely used by data developers and researchers to collect data for various type of analysis on textual data. Twitter provides many ways to fetch or download tweets where streaming APIs also allow to stream live data from twitter. Study mentioned here, is focusing on finding those features from tweets which are not available in tweet objects, such as age or gender of user. So, in this report I explained some of techniques which are used to identify gender of twitter user or to identify their age using patterns of user's tweet content. On the collected tweets, data cleaning techniques are applied. For gender detection, specific features such as tweet content, user name and words pattern are used. For age identification date of tweet and tweet contents are used.**

Keywords*— *Twitter API, Data collection, stop-words, Features extraction, Model selection, Vectorizer, Tf-idf, Scikit-learn, SVM, LR, DT, RC, 10-fold Cross-validation, One-vs-Rest, Multiclass classification, macro-micro average scores, ROC curves.

I. INTRODUCTION

Twitter data has number of features except twitter user's age or gender information. The raw tweet does not contain user's current age value or date of birth feature or user's gender. This master's project aims to find user's age information and gender using his or her tweeted contents. Twitter's developer API are used to collect all historical tweets for each user. Also, dataset from CrowdFlower is downloaded for gender identification program. Words which are not relevant to find outcome, are removed using stop word removal method.

For male or female identification, collection of frequent words, used by each gender is used. To identify age of user, specific set of tweets are used, where tweets contain key phrases such as 'my age', 'I am # old', etc. Using this phrases and numeric content of tweet current age can be identified with the help of current year information. Ages are classified into 7 distinct groups. At the end, using the data set for each group and their appropriate class labels, different models are trained to measure accuracy of models by plotting receiver operating characteristic curves. For training purpose dataset is divided into training and testing set using method called train-test-split from scikit-learn. After that 10-flod cross validation technique is used to find model performance accuracy.

II. RELATED WORK

Research on 'Probabilistic Inference of Twitter Users' Age based on What They Follow' [1] authors approached towards age detection of twitter user. Authors devise a language-independent methodology for determining the age of Twitter users from data that is native to the Twitter ecosystem. The key idea is to use a Bayesian framework to generalize ground-truth age information from a few Twitter users to the entire network based on what/whom they follow. Author also mentioned that their approach scales to inferring the age of 700 million Twitter accounts with high accuracy.

In an introductory paper 'An introduction to Twitter Data Analysis in Python' (2016) [3], authors explained the process of storing, preparing and analyzing twitter data. Authors described about tokenization, finding term frequency of content and how it helped them in analysis of what user frequently tweets. They examined methods and tools available in python programming language to visualize the analyzed data. They also analyzed streaming data, which allowed them to make real-time decisions on the basis of real-time data.

In research paper on 'Semantic Patterns for Sentiment Analysis of Twitter' (2014) [4] authors carried out semantic pattern analysis. They mentioned that sentiment is often implicitly expressed via latent semantic relations, patterns and dependencies among words in tweets. they propose a novel approach that automatically captures patterns of words of similar contextual semantics and sentiment in tweets. This approach does not rely on external and fixed sets of syntactical templates or patterns, nor requires deep analyses of syntactic structure of sentences in tweets. They evaluated their approach with tweet-level and entity-level sentiment analysis tasks by using the extracted semantic patterns as classification features in both tasks.

In 'Twitter conversation patterns related to research papers' [6] authors tried to deal with what academic texts and datasets are referred to and discussed on Twitter. They used document object identifiers as references to these items. They streamed tweets from Twitter API including the strings "dx" and "doi" while simultaneously streaming tweets posted by and to the authors of the tweets captured. By doing so they

were able to capture tweets referring to a digital object. Captured tweets were analyzed in diverse ways, both quantitatively and qualitatively. The conversations with at least 10 tweets were analyzed using content analysis. Their study concludes digital object identifiers were mainly referred to for self-promotion, as conversation starters or as arguments in discussions.

Social media services, such Twitter, present a convenient way to express opinions and concerns about crimes. The main objective of study on public's reaction pattern[7] is to explore people's perception of homicides, specifically, how characteristics and proximity of the event affect the public's concern about it. The analysis explores Twitter messages that refer to homicides that occurred in London in 2012. The spatial analysis revealed a strong spatial dependency between the estimated home locations of users who tweeted about homicide related news and the locations of these incidents. Analysis of crime characteristics indicates that some of them are associated with a higher frequency of tweets, where some characteristics do not have a significant impact on the posting frequency.

III. DATA COLLECTION

A. Overview of Twitter API

First of all, in order to get tweet data, one must have four unique keys provided by Twitter for development support. These four keys (alphanumeric-) include (1) consumer-key = 'your consumer key', (2) consumer-secret = 'your consumer secret', (3) access-token = 'your access token' and (4) access-secret = 'your access secret'. Now, using these keys you can access all tweet data from personal profile and store them in required format (JSON or csv or text file).

Twitter provides multiple ways to collect data from application, such as REST API's, Twitter Developer API [9], Tweepy, etc. For this project I used Developer API's standard search [10] tweet functionality. This search request returns collection of relevant tweets based on specified query. Query contains parameters like, 'name', 'max_id', 'count', etc.

B. Sample tweet

Before getting started with fetched tweets dataset, it is important to know about the structure of each raw tweet. Each individual raw tweet contains mainly 5 objects; tweet object, user object, entities object, geo objects and extended entities object. Each object is in the form of key: value pair, which provide ease in accessing specific value (key in turn may also have multiple keys). Overall there are more than 70 unique keys in raw tweet. Below is the example of raw tweet [10] with some features and its values. Note: this is not full raw tweet. This tweet object is used just to give idea of content and its 'key: value' format.


```
{
  "created_at":"Sun Feb 25 18:11:01  +0000  2018",
  "id": 967824267948773377,
  "id_str":"967824267948773377",
  "text": "From pilot to astronaut, Robert H. Lawrence was…",
  "entities": {
     "hashtags":[ ],  "user_mentions":[ ],
      "urls":[ {
                "url": "https://t.co/FjPEWnh804",
                "expanded_url":"https://twitter.com/i/web",
                "display_url":"twitter.com/i/web/status/9",
              } ] },
  "metadata": { "result_type":"popular",
                 "iso_language_code":"en" },
  "source":"<a href='https://' rel="nofollow">Sprinklr</a>",
  "in_reply_to_status_id": null,
  "in_reply_to_status_id_str": null,
  "in_reply_to_user_id": null,
  "user": {
     "id": 11348282,   "id_str":"11348282",
     "name": "NASA", "screen_name":"NASA",
     "description":"Explore the universe and discover…",
     "url": "https://t.co/TcEE6NS8nD",
     "entities": {
           "url": { "urls" : [ … ] }
         },
     "followers_count": 28605561,
     "profile_background_image_url": "http://pbs.twimg.com/"
     "profile_image_url":  "http://pbs.twimg.com/..."
     "friends_count": 270,
     "created_at":"Wed Dec 19 20:20:32  +0000  2007",
     "favourites_count": 2960,
     "time_zone": "Eastern Time (US & Canada)", …
   },
  "geo": null,  "coordinates": null,  "place": null,
  "retweet_count": 988,  "retweeted": false,  "lang": "en",
  "extended_entities": {
      "media": [ {
           "expanded_url": "https://twitter.com/photo/1",
           "display_url":"pic.twitter.com/HRnGMHlW7J",
           "url": "https://t.co/HRnGMHlW7J", …
           "sizes": { … }
           "type":"photo",
           "id": 939239185755512832,
           "media_url": "http://pbs.twimg.com/media/DQ.jpg"
        } ]
      }
}
```


For implementation part, text pattern analysis and feature extraction, I used following keys: 'text' which represents the tweeted content, 'created_at' represents date of tweet creation, 'screen_name' represents user's display name on Twitter, 'name' represents twitter user's name, 'id' represents unique id number in numeric format and 'id_str' represents unique tweet id number in string format.

C. Age-range dataset

For age range classification part special dataset is formed. This dataset contains raw tweets in the same format as mentioned, but the difference is that the dataset is filtered based on tweet content. Only those tweets are being considered here, which tweet content include specific phrases: 'I am # years', 'my age is' or 'I am #'. Here # represent numeric value. Basically, dataset is collection of tweets which can describe user's age information. This filtered dataset is then used to classify all users and tweet in their appropriate age group.

D. CrowdFlower

For the gender identification part, dataset is taken from CrowdFlower's Data For Everyone Library. Data is available free of charge for the community to use and research. This data set was used to train a CrowdFlower AI gender predictor. Contributors were asked to simply view a Twitter profile and judge whether the user was a male, a female, or a brand (non-individual). The dataset contains 20,000 rows, each with a user name, a random tweet, account profile and image, location, and link and sidebar color. [11]

E. Frequent male and female words

To collect the frequent words male or female use in their day to day life while taking or texting with other, two different dataset are formed. Male dataset had more then 7000 words and female dataset had more then 6000 words. This dataset includes some common words also which are used frequently by both male and female. So, to find more accurate dataset, common words are removed from both sets and new dataset have only words which are not common. As the result I have 4728 female frequent words and 5861 male frequent words as the final dataset to use.

Some sample words of male and female dataset. Male dataset has words such as: 1080p, 720px, #eagles, #band #budget, ps4, cops, coffee, castle, devil, fight, kate, xbox, #dataprotection, program, etc. Female dataset has words such as: #babe, #babies, #autumnevenings, #style, #saycheese, ppl, pretzels, crush, omg!!!, skinny, skirt, #bestfriends, love, ignore, sea, sisters, clothes, etc.

IV. METHODS

There are some methods which are implemented by me to work with twitter dataset (raw tweets from twitter and filtered tweets for age-range program) and to extract information in such a way, which are accurate with this type of data format and provide output in optimal time. This methods include stop-words removal, male-female identification (gender prediction) and finding age of user who tweeted the tweet (age-range classification).

Before getting into detailed description of methods, let's first understand the process or flow of the program. Figure 1 (The Process) explains direction of both programs from input to output. Basically it starts it twitter dataset. Then data cleaning takes place on extraceted featureas which are useful to generate results. Here data is modified using methods such as stop words removal or tokenization and feature extraction. During this process, techniques such as findind age of user, assigning users to their suitable age group based on age value, finding male or female words frequency to identify user's gender, etc. are applied on extracted features. Data is labeled using based on their class; for age range there are seven labels 1 to 7 for each class and for gender detection there is binary classifiction.

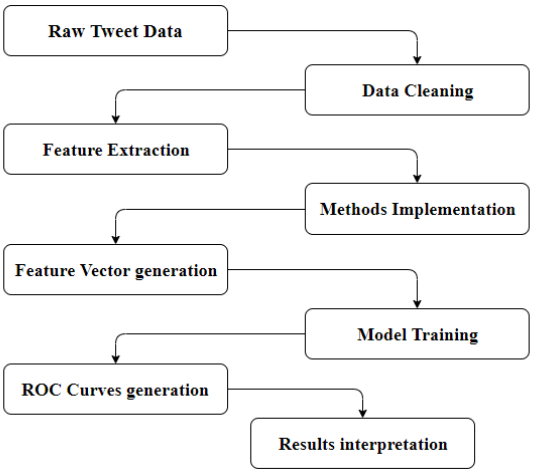

Fig.1: The Process

After that all features are converted into featuer vector or document-term matrix form which is supported by machine learning algorithms. Data is partitoined into traing and testing set using scikit-learn's module. At the end multiple models (algorithms) are trained using data to find accuracy of algorithm with given dataset and to show performance via plotting receriver operating characteristic curves.

A. Stop-word removal:

The stop-words removal method identified english stop words such as 'a', 'an', 'the', 'of', 'on', etc. and removed them from the original text content. Using this technique, after removing such words, length of text content is reduced. In this method I splitted text into list of single word, each word will be checked with stop-words and at the end all non-stop-words were concatenated into string again. This will reduce the size of feature vector also, as there will not be extra words in content at time of vectorization. This method was used in gender prediction program and age-range classification also.

Example: let say we have sentence like 'This is a sample sentence and showing off the words filtration', then output of this sentence after aplying stop-word removal method is 'this sample sentence showing words filtration'.

This method also change upper case letters to lower case. This prevents duplication of same words in different cases.

B. Male-female identification:

The male-female identification worked in following way. Using two datasets (explained in section D of Data collection), by counting the occurrence of male and female words in their tweet content, method can make decision (based on high probability) that the tweet is of male or female twitter user. This method was used in gender indentification and prediction program.

Example: let say we have tweet, such as 'hey bro, let's play counter-strike game tonight. Now we have play-station and controllers at place… #cs #psp', this tweet is identified as a male tweet by identification method, as it has more words that are used by male compare to female.

C. Age identification:

The next method is finding age of twitter user, using his/her tweet content. As mentioned earlier, filtered collection of tweets are used for this program. Now, this method takes raw tweet as input and finds one or two digit number followed by phrases and stores number as one of age value. Method also fetches the year of tweet posted online. It adds the year difference between post date and current data, and add this difference to age value to find current age of user. This method proved accurate in finding age group with filtered dataset with such key-phrases.

Example: we have tweet from database: 'but seriously I'm 20 years old and I've never had a limo for my birthday!!! #spoiledkids #smh'. In this case, method will identify and return age as 25, because this tweet was posted in year 2013 and current year is 2018.

V. VECTORIZER

This section covers description of vectorizer, what it represents different methods of vectorizer, which are used in implementation. Vectorizer can simply be defined as a method to generate feature extractor. Machine learning algorithms do not support datasets of format text or image. This format must be converted to vector format which machine learning algorithms support.

The 'feature_extraction' module of scikit-learn provides functionality to create feature vector from text or image dataset. Feature extraction is very different from Feature selection, extraction consists of transforming arbitrary data (text or images) into numerical features usable for machine learning. Feature selection is a machine learning technique applied on these extracted features. Feature-extraction has classes such as CountVectorizer, Tf-idf, etc. which are very useful in feature vector generation. [12]

A. CountVectorizer:

It converts collection of text document to a matrix of token counts. This method produces a sparse representation of the counts using's matrix. Each row in sparse matrix have either 0 or 1, where 0 means a feature (token) is not present in particular document and 1 means token is part of document.

Example: let say we have three documents, 'I hate dogs and knitting', 'I love dogs' and 'Knitting is my hobby and my passion'. Then output of count vector using vocabulary of these three sentences will look like following image (figure 2). [13]

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

Fig.2: Count Vectorizer example

B. Tf-idf:

Tf-idf vectorizer converts a collection of raw documents to a matrix of TF-IDF features. It is equivalent to Count-Vectorizer followed by Tf-idf Transformer. TF-IDF stands for Term Frequency – Inverse Document Frequency. Terms frequency summarizes how often a given word or token appears within a document. The transform method is used to transform documents to document-term matrix. The resulting vector from above methods is used by learning algorithms such as Support Vector Machine, Random Forest classification, Logistic Regression, etc. [14]

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	0.18	**0.48**	0.18							
Doc 2	0.18		0.18	**0.48**	0.18	0.18				
Doc 3					0.18	0.18	**0.48**	**0.95**	**0.48**	**0.48**

Fig.3: TF-IDF example

Example: Here we use same three documents which were used in previous example. Term frequency basically determines how important a word is by looking at how frequently it appears in the document. Now, for a word to be considered as an important word of a document, it shouldn't appear that often in the other documents. Thus, an important word's document frequency must be low, which means its inverse document frequency must be high. Figure 3 shows the value for each word based on their importance. [15]

VI. ALGORITHMS

A. Detailed explainition of algorithm

This section covers theoretical fundamentals of machine learning algorithms which are used to measure accuracy of model with given dataset, to find sensitivity and to

plot ROC curves using multiclass settings for classifiers. These algorithms include support vector machine, decision tree, random forest, logistic regression, ridge classification (ridge regression), multiclass classification and one-vs-rest classifier method.

1) Support Vector Machine (SVM):

SVM is a supervised machine learning algorithm which can be used for classification and regression challenges. However, it is mostly used in classification problems. It can classify both linear and nonlinear data.

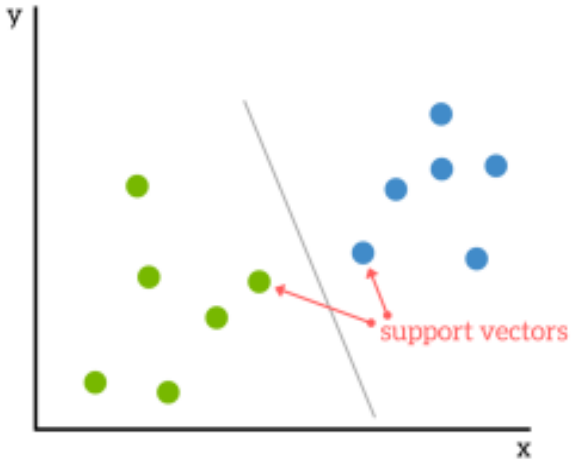

Fig.4: Hyperplane classifying two classes

If the data is linearly separable, the SVM searches for the linear optimal separating hyperplane, which is a decision boundary that separates data of one class from another class. Mathematically, a separating hyperplane can be written as: $W \cdot X+b=0$, where W is a weight vector and $W = w_1, w_2, \ldots, w_n$, X is a training tuple and 'b' is a scalar value. If the data is linearly inseparable, the SVM uses nonlinear mapping to transform the data into a higher dimension. It then solves the classification problem by finding a linear hyperplane. [2] [16]

Hyperplane can be defined using simple example. Let say we have classification task with only two features (Figure 4), we can think of hyperplane as a line that linearly separates and classifies a set of data. Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplane as possible, while still being on the correct side of it. [17]

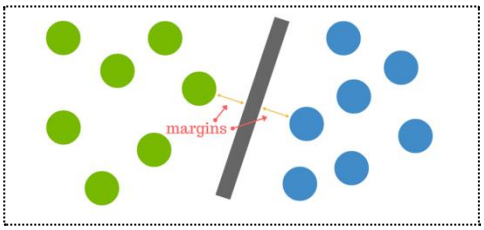

Fig.5: SVM Margin

The distance between the hyperplane and the nearest data point from either set is known as the margin. Above figure shows margin and data points with separating hyperplane. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly.

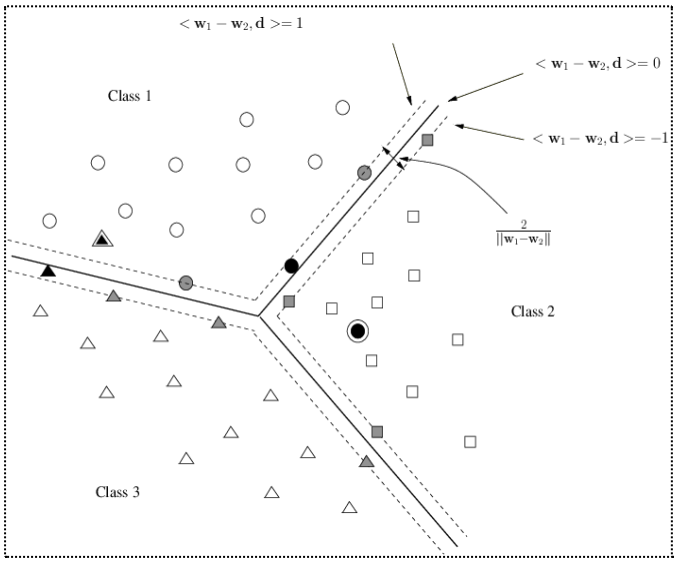

Fig.6: Hyperplane classifying multiple classes [18]

Figure 6 shows Multiclass classification solved by multiclass SVMs. Solid lines represent separating hyperplanes, while dotted lines are hyperplanes with confidence margin equal to one. The multiclass margin is the minimal distance between two dotted lines. Dark points are support vectors. Black points are also constraints violations and extra borders indicate violations which are also training errors. It shows how multiple hyperplane best classifies data into three perfect regions.

2) Decision Tree (DT):

A decision tree is a Machine Learning algorithm capable of fitting complex datasets and performing both classification and regression tasks. The idea behind a tree is to search for a pair of variable-value within the training set and split it in such a way that will generate the 'best' two child subsets. The goal is to create branches and leaves based on an optimal splitting criterion, a process called tree growing. Specifically, at every branch or node, a conditional statement classifies the data point based on a fixed threshold in a specific variable, therefore splitting the data. To make predictions, every new instance starts in the root node and moves along the branches until it reaches a leaf node.

The algorithm used to train a tree is called Classification And Regression Tree (CART). The algorithm seeks the best feature–value pair to create nodes and branches. After each split, this task is performed recursively until the

maximum depth of the tree is reached or an optimal tree is found. Depending on the task, the algorithm may use a different metric to measure the quality of the split. It is important to mention that due to the greedy nature of the CART algorithm, finding an optimal tree is not guaranteed and usually, a reasonably good estimation will suffice.

Predictors				Target
Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Fig.7: dataset for decision tree

These two figures (Fig.7 and Fig.8) are example of how sample decision tree can be constructed using given dataset. [19]

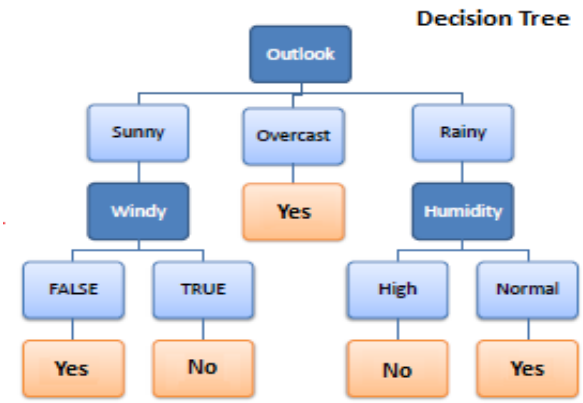

Fig.8: Decision Tree

3) Logistic Regression (LR):

Logistic regression (LR) classification implements regularized logistic regression using the 'liblinear' library [20], 'newton-cg', 'sag'[5] and 'lbfgs' solvers. It can handle both dense and sparse input. In the case of multiclass classification, the training algorithm uses the one-vs-rest (OvR) scheme by setting the 'multi_class' option to 'ovr'.

Unlike actual regression, logistic regression does not try to predict the value of a numeric variable given a set of inputs. Instead, the output is a probability that the given input point belongs to a certain class. For simplicity, let's assume that we have only two classes, and the probability P+ is the probability that a certain data point belongs to the '+' class, the probability P- is the probability that a certain data point belongs to the '-'class. In other words, P- is equals to 1 - P+. Thus, the output of logistic regression always lies in [0, 1].

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent variables. The name logistic regression is mostly used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique class. In our age range classification, we have more than two values for dependent variables.

4) Random Forest (RF):

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods such as decision tree, random forest, etc. empower predictive models with high accuracy. Random forest classifier has superior performance over a single decision tree with respect to accuracy. It is essentially an ensemble method based on bagging.

The classifier works as follows: Given D, the classifier firstly creates k bootstrap instance of D, with each of the instances denoting as D_i. A D_i has the same number of the tuples as D that are sampled with replacement from D. By sampling with replacement, it means that some of the original tuples of D may not be included in D_i, whereas others may occur more than once. Classifier then constructs a decision tree based on each D_i. As a result, a 'forest' that consists of k decision trees is formed. To classify an unknown tuple, each tree returns its class prediction counting as vote. The final decision of tuple's class is assigned to the one that has the most votes. [2]

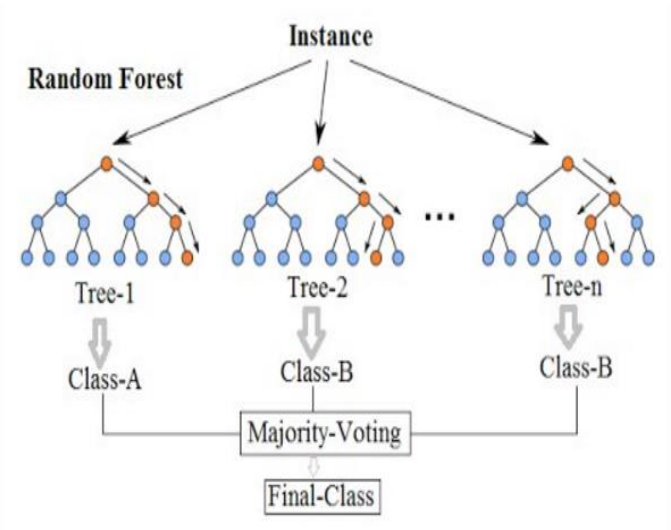

Fig.9: Random Forest process flow

5) Ridge Classifier (RC):

Ridge regression is a classical data modeling method to solve multicollinearity problem of covariates in samples. Here, multicollinearity refers to a situation in which more than one predictor variables in a multiple regression model are highly correlated. If multicollinearity is perfect, the regression coefficients are indeterminate, and their standard errors are infinite. If it is less than perfect, the regression coefficients although determinate but possess large standard errors, which means that the coefficients cannot be estimated with great accuracy.

Using a fundamental concept that samples from a specific class lie on a linear subspace, a new test sample from any class can be represented as a linear combination of class-specific training samples. This assumption can be formulated as a linear model in terms of ridge regression. For multiclass classification, n class classifiers are trained in one-vs-rest approach. [21] [23]

6) Multiclass classification:

It refers to the classification task with more than two classes: for example, classify a set of text documents which may be a report, letter or resume. Multiclass classification makes the assumption that each sample is assigned to one and only one label, that is text document can be either a report or a letter or resume but not all three at the same time. [24]

Multiclass classification problems can be categorized into One-vs-Rest and One-vs-One. The techniques developed based on reducing the multi-class problem into multiple binary problems. It can also be called as 'problem transformation techniques'.

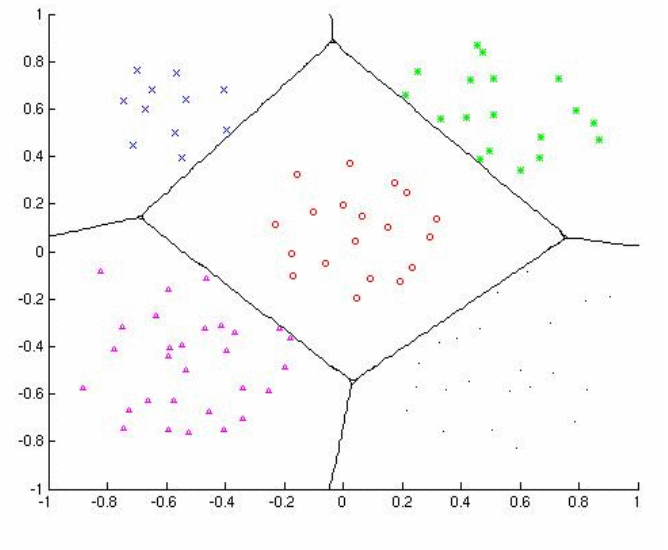

Fig.10: Multiclass classification [22]

The One-vs-Rest (OvR) or one-vs-all (OvA) or one-against-all (OAA) is strategy that involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. This strategy requires the base classifiers to produce a real-valued confidence score for its decision, rather than just a class label; discrete class labels alone can lead to ambiguities, where multiple classes are predicted for a single sample.

In One-vs-One (OvO) reduction, one trains K*(K−1)/2 binary classifiers for a K-way multiclass problem; each receives the samples of a pair of classes from the original training set and must learn to distinguish these two classes. At prediction time, a voting scheme is applied: all K*(K−1)/2 classifiers are applied to an unseen sample and the class that got the highest number of "+1" predictions gets predicted by the combined classifier.

B. Training Process

For age-range classification I had total seven distinct dataset, one for each age group. Each one contains tweet text, date-created and user-name of all user, who belongs to that group. Here all tweets are filtered with specific key-phrases as mentioned in data collection section.

For age group 13-18 there are 16455 tweets and info, for group 19-24 there are 74035 tweets, group 25-34 had 37885 tweets, group 35-44 had 7613 tweets, group 45-54 had 3717 tweets, group 55-64 had 2951 tweets and group 64 plus had 5988 tweets. All dataset has their unique labels (y). All content was divided into training and testing set using scikit-learn's model selection method called 'train_test_split', where 60% data assigned for training and remaining for testing. Size of x_train = (420, 2717), x_test = (280, 2717), y_train = (420,) and y_test = (280,). Here 'x' is feature vector of all tweet content and 'y' is data class value. This x and y are then used for model training.

Models like SVM, DT, RF, etc. are trained with 10-fold cross validation technique. Folding can be explained as following. This method splits dataset into 10 consecutive folds (without shuffling). Each fold is then used once as a validation while the remaining 9 folds form the training set. Now the model is trained using 9 of the folds as training data and the resulting model is validated using remaining part of the data. The performance measure of model is reported by 10-fold cross-validation by computing average of all values.

VII. Receiver Operating Characteristic

To evaluate classifier's output quality, Receiver Operating Characteristic (ROC) curve is used. ROC curve features true positive rate on Y axis and false positive rate on X axis. This means that the top left corner of the plot is the 'ideal' point with false positive rate of zero, and a true positive

rate of one. This is not very realistic, but it does mean that a larger area under the curve (AUC) is usually better.

ROC curves are typically used in binary classification to study the output of a classifier. To extend ROC curve and ROC area to multi-class or multi-label classification, it is necessary to binarize the output. To binarize the output, scikit-learn provides function called label_binarize. This function binarizes the labels in one-vs-all (one-vs-rest) fashion. [24]

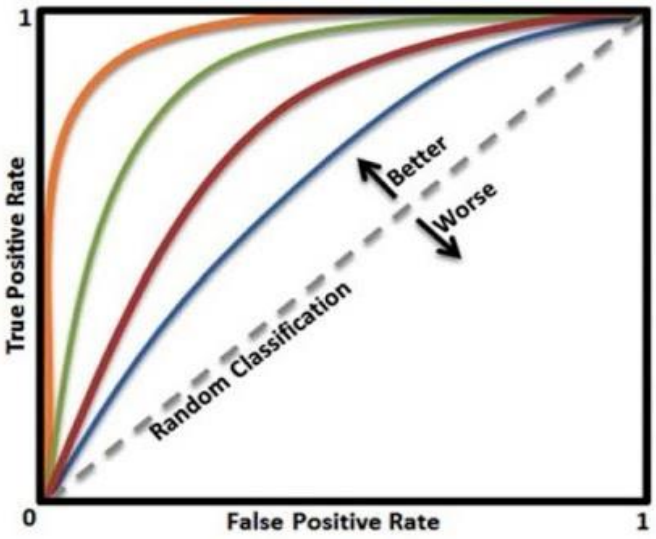

Fig.11 ROC basics

Micro and macro average scores: The micro average method sums up the individual true positives, false positives, and false negatives of the system for different sets and apply them to obtain statistics. The macro average is straight forward, it is the average of the precision and recall of the system on different sets. And the final macro or micro average f-score is the harmonic mean of precision and recall. [25]

VIII. RESULTS

Following are the resulting ROC curves for multiclass classification using various machine learning algorithms. These curves represent micro and macro average score for each classifier and the value of area under the curve (AUC) for each class of dataset. For age-range classification, total 7 age groups were formed starting from age 13 to 18, age 19 to 24, 25 to 34, 35 to 44, 45 to 54, age 55 to 64 and last age group is of 64 plus age users.

Figure 12 shows the ROC curve for multiclass support vector machine algorithm. This classifier achieved micro average of 0.68 and macro average of 0.67 and age group (class) 19 to 24 had maximum value for area under the curve, that is 0.73 out of 1. Age group 64 plus had lowest score, 0.58.

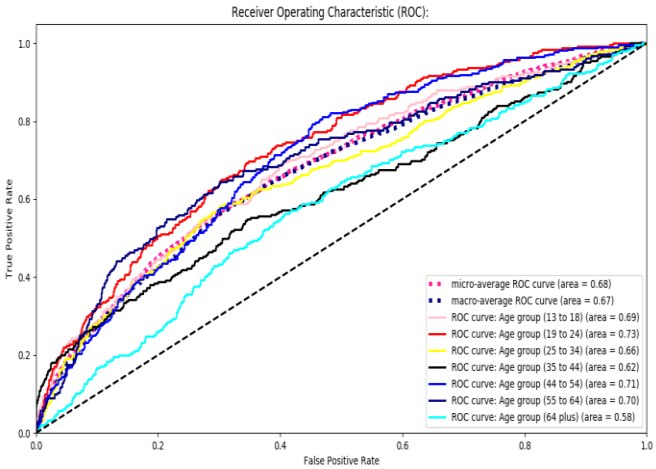

Fig.12 ROC curves for Support Vector Machine

Figure 13 shows the ROC curve for random forest algorithm. This classifier achieved micro average of 0.59 and macro average of 0.58 which is comparatively low then SVM's results; and class of age 19 to 24 had maximum value 0.65 for area under the curve. Here also the age group 64 plus had lowest value that is 0.51 as AUC.

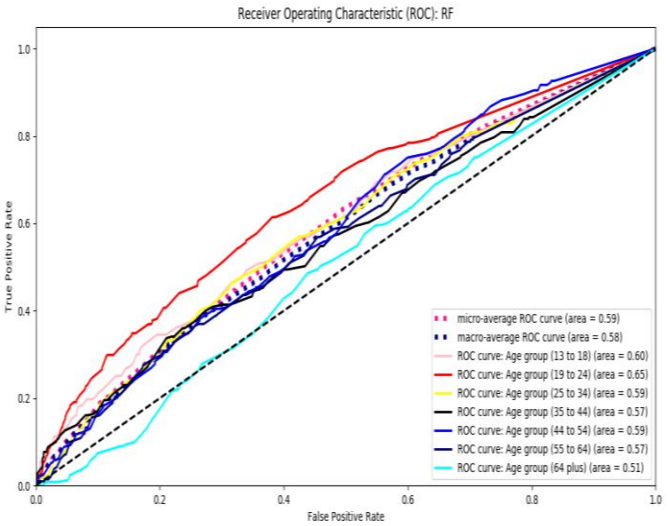

Fig.13 ROC curves for Random Forest

Figure 14 shows the ROC curve for logistic regression with multiclass setting. This classifier achieved micro average of 0.64 and macro average of 0.66 out of 1 (comparatively this classifier's results are close to SVM's results) and age group 44 to 54 had maximum value 0.72 for area under the curve. In this model, again the age group 64 plus came up with lowest value for area under the curve, i.e. 0.57.

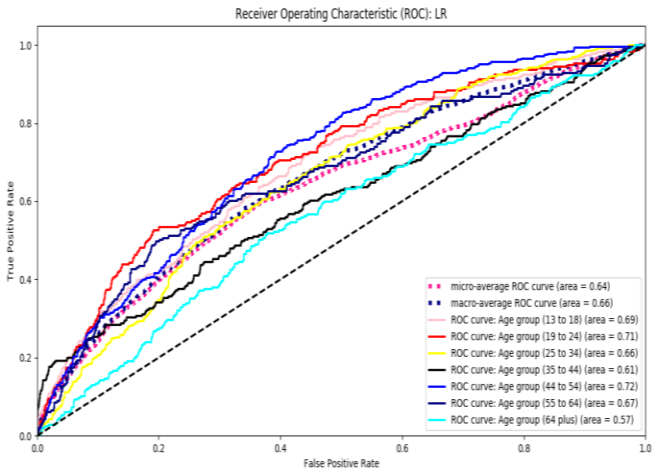

Fig.14 ROC curves for Logistic Regression

Figure 15 shows the ROC curve for ridge classifier. This classifier achieved micro average of 0.68 and macro average of 0.67 which is exactly same as SVM's results; and class of age 19 to 24 had maximum value 0.74 for area under the curve. These results are very much identical to SVM. Again, age group 64 plus had lowest value, 0.58 for AUC.

Figure 16 shows the ROC curve for Decision Tree classification. In this case micro and macro average both are 0.54 which is worst compare to other model's results. Age group 35-44 had maximum value 0.56 for area under the curve and age group 64 plus had minimum value 0.50 for AUC. This classifier performed almost similar with all age-group data

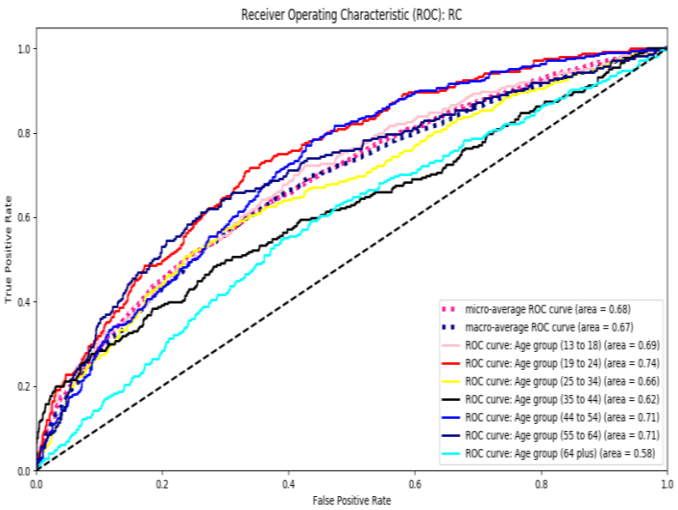

Fig.15 ROC curves for Ridge Classifier

Figure 17 shows the ROC curve for One-vs-Rest SVC classification. In this case micro average is 0.67 and macro is 0.66 which is better than random forest's results. Age group 55-64 had maximum value 0.7 for area under the curve and age group 64 plus had minimum value 0.58 for AUC.

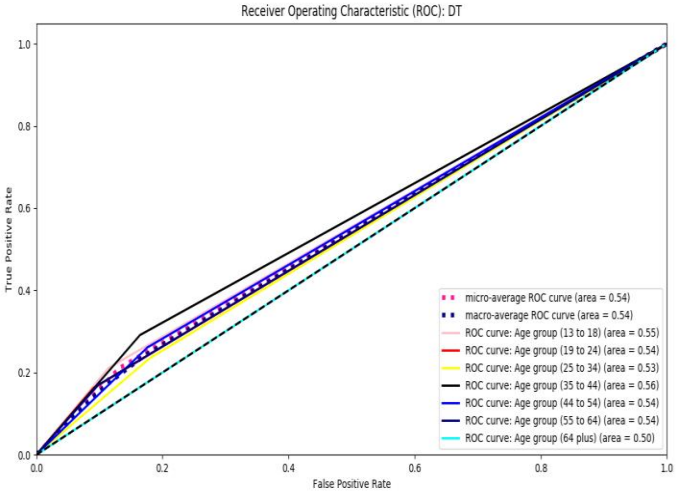

Fig.16 ROC curves for Decision Tree

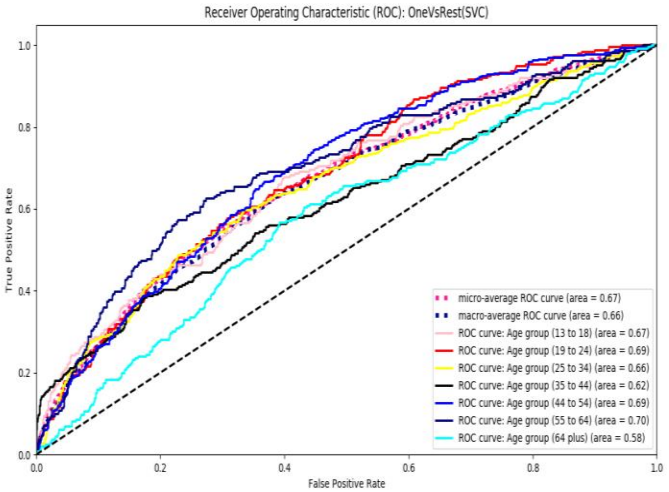

Fig.17 ROC curves for One-vs-Rest SVC

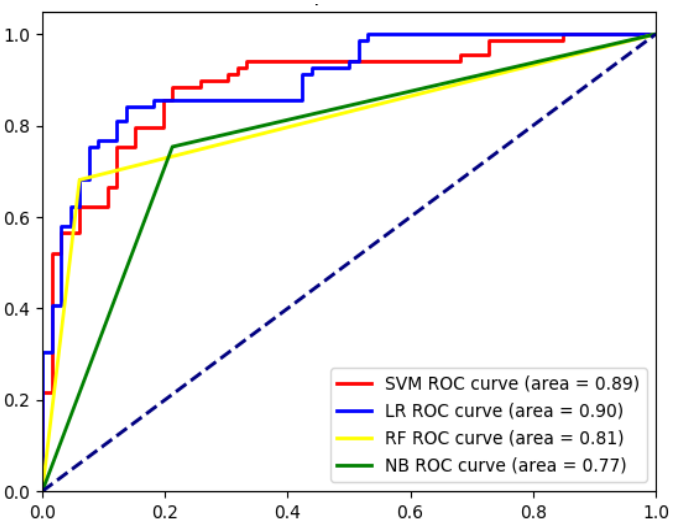

Fig.18 ROC curves for gender identification data

Next is the ROC curves (Figure 18) generated using dataset used for gender identification program. On this dataset four different machine learning algorithms are applied to find which algorithm best fits the data and able to classify them more accurately compare to other algorithms. As results support vector machine and logistic regression algorithm performed almost identical with value of 0.89 and 0.90 respectively for area under the ROC curve. Random Forest achieved 0.81 and Naïve Bayes achieved 0.77 as AUC value.

IX. CONCLUSION

This project started with raw tweet collection and one dataset downloaded from CrowdFlower. After applying mentioned methods, techniques and algorithms, program can make cluster or set of users belong to same age group and stores user's tweet content with user name and ID. Using this stored information, it is easy to find text patterns, similarities and differences between different age groups. Moreover, results of age value have good accuracy as some outputs are cross checked with user's profile to check weather the identified age group for him/her is correct or not.

After analyzing all results of receiver operating characteristic curves, micro and macro average scores and AUC (area under the curve) for various machine learning algorithm; I can conclude that for multiclass classification, support vector machine (SVM) and ridge classifier (RC) results outperform other algorithms. During the comparison of micro score between different classifiers, ridge classifier won with highest value as shown in below figure.

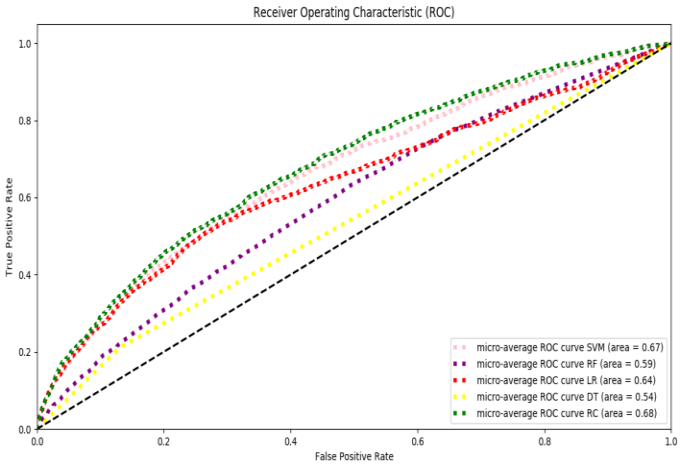

Fig.19 micro-average comparison

Moreover, data for age group 19 to 24 is classified more nicely through all algorithms and achieved best value for AUC in all cases. Results of gender identification dataset concludes that, data can be classified more accurately using SVM or logistic regression (LR) with almost 90% AUC value and achieved best performance among all.

REFERENCES

1] Benjamin Paul Chamberlain, Clive Humby, Marc Peter Deisenroth (2017), 'Probabilistic Inference of Twitter Users' Age based on What They Follow', arXiv: 1601.04621v2.

2] Xing Fang and Justin Zhan (2015), 'Sentiment analysis using product review data', Fang and Zhan Journal of Big Data, 2:5 DOI10.1186/s40537-015-0015-2.

3] Wisdom, Vivek & Gupta, Rajat. (2016). 'An introduction to Twitter Data Analysis in Python'. 10.13140/RG.2.2.12803. 30243.

4] Saif H., He Y., Fernandez M., Alani H. (2014) Semantic Patterns for Sentiment Analysis of Twitter. In: Mika P. et al. (eds) The Semantic Web – ISWC 2014. Lecture Notes in Computer Science, vol 8797. Springer, Cham.

5] Mark Schmidt, Nicolas Le Roux, Francis Bach (2017) 'Minimizing Finite Sums with the Stochastic Average Gradient.' Mathematical Programming B, Springer, 2017, 162 (1-2), pp.83-112.

6] Gustaf Nelhans and David Gunnarsson Lorentzen (2016), 'Twitter conversation patterns related to research papers', Information Research Vol. 21 no. 2, June 2016.

7] Kounadi O, Lampoltshammer TJ, Groff E, Sitko I, Leitner M (2015) Exploring Twitter to Analyze the Public's Reaction Patterns to Recently Reported Homicides in London. PLoS ONE 10(3): e0121848.

8] Cheng, J. Caverlee and K. Lee. You are where you tweet: a content-based approach to geo-locating Twitter users. In CIKM, 2010.

9] https://developer.twitter.com/en/docs

10] developer.twitter.com/en/docs/tweets/search/api-reference

11] https://www.kaggle.com/crowdflower/twitter-user-gender-classification

12] scikit-learn.org/stable/modules/feature_extraction.html

13] https://towardsdatascience.com/another-twitter-sentiment-analysis-with-python-part-4-count-vectorizer-b3f4944e51b5

14] https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

15] http://datameetsmedia.com/bag-of-words-tf-idf-explained/

16] www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

17] https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html

18] https://www.researchgate.net/figure/Multiclass-classification-problem-solved-by-multiclass-support-vector-machines-Solid_fig1_225541675%2023

19] http://www.saedsayad.com/decision_tree.htm

20] https://www.csie.ntu.edu.tw/~cjlin/liblinear/

21] https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf

22] cmp.felk.cvut.cz/cmp/software/stprtool/examples.html

23] https://www.researchgate.net/publication/285955730_Kernel_ridge_regression_classification

24] http://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html

25] http://rushdishams.blogspot.com/2011/08/micro-and-macro-average-of-precision.html