1.5em

School of Engineering and Applied Science (SEAS), Ahmedabad University

B.Tech (ICT) Semester VI / M.tech / PhD: Machine Learning (CSE 523)

Project Submission - Final Report Submission

Submission Deadline: March 12, 2020 (11:59 PM)

- **Group No.: BNLP5**
- **Project Area: Natural Language Processing**
- **Project Title: Harnessing Twitter 'Big Data' for Automatic Emotion Identification**
- **Name of the group members :**

1. Jay Patel(AU1741018)
2. Manav Shah(AU1741042)
3. Prima Sanghvi(AU1741045)
4. Priyanshi Deliwala(AU1741047)

1 Introduction

1.1 Background

– **Introduction:**

This report describes the emotional state of a person, extracted from the popular micro-blogging service Twitter. Knowing social media tools are being used increasingly to present their day-to-day happenings, understanding the emotions behind the events helps us to interpret and perceive the behavior of millions of individuals. Focus on growing the interest behind recognizing the emotions and sentiments in text. Natural emotion rendering of text helps to learn individuals biases, inclination and personality. Thus, an application for emotional text synthesis solves two basic problems. First, what emotion or emotions most appropriately describe a certain text statement, and second, given a text statement and a specified emotional mark-up, how to render the conceit contour in order to understand the emotional content.

– **Related Work:**

In this paper, we concentrate on supervised emotion identification literature and the way to automatically collect training data rather than rule-based approaches [10], [15]. There are only some efforts on supervised methods, partly due to the labor intensive nature of the manual labeling task. Alm et al. [1] present an empirical study of applying machine learning techniques to classify fairy tale sentences into different emotions. Aman and Szpakowicz [2] combine unigrams, emotion lexicons and a thesaurus as features to classify blog sentences into six basic emotion categories.Automatically creating training data from weblogs has also been addressed. The approaches in [8], [13], [19] collect blog posts that are assigned mood labels (e.g., amused, tired) by the blog writers. Tokuhisa et al. [16] collect 1.3 million sentences in Japanese by exploiting the sentence pattern "I was ** that ...", during which "**" and "..." talk to an emotion word and also the sentence reflecting the emotion, respectively. For example, "I was disappointed that it suddenly started raining." Turning to harnessing the hashtag phenomenon on Twitter, Choudhury et al. [4] collect emotion tweets via emotion hashtags and analyze users' emotional states in social media through affective space (valence and activation). Both our work and [4] collect tweets via emotion hashtags, but identifying the writer's emotion from a tweet isn't their focus. Mohammad [9] also collects emotion tweets via emotion hashtags. However our work uses a way larger dataset and a extensive list of features. To the most effective of our knowledge, the three questions we raised earlier are largely unexplored.

– **Base Article:**

Emotion is both pervasive in and fundamental to all parts of our lives. It impacts our dynamic, influences our social connections, shapes our day by day conduct, even outlives our recollections. With the fast development of feeling rich printed content, for example, microblog posts, blog entries, and gathering conversations, there is an extraordinary need and chance to create programmed apparatuses for recognizing and breaking down individuals' emotions communicated in content.As Helen Keller states, *"The best and most beautiful things in the world cannot be seen or even touched. They must be felt with the heart",* emotions influences our decision-making, affects our social relationships, shapes our daily behavior, even outlasts our memories.Most of current emotion identification research relies on manually annotated training data. Consequently, most of existing emotion datasets are relatively small, of the order of thousands of entries, which fail to provide a comprehensive coverage of emotion-triggering events and situations. Manual annotation of data by human experts is very labor-intensive and time-consuming. Moreover, in contrast with other annotation tasks such as entity or topic detection, a human annotator's judgement of emotion in text tends to be subjective and varied, and hence, less reliable. While there is lack of sufficient labeled data for emotions we find that many of the social media services have entered into the Big Data era. Twitter is one such microblogging service which provides 500 million tweets per day. Using their API we can easily get to know the current situation of a particular area and

inner feelings of their people.

From the collected labelled dataset of emotion tweets with 7 different emotions we tried to find the effective features for emotion identification. There are variety of features such as n-gram(unigrams and bigrams), emotion lexicans, parts of speech, etc. With two machine learning algorithm Multinomial Naive Bayes and LIBLINEAR as these algorithms prove to be most efficient in large-scale dataset. The data was preprocessed by lower casing all the words, replacing user mention, anonymizing user, replacing punctuation or letters repeated more than twice, normalizing some frequently used expression and stripping of the hash symbol. The overall performance, of individual classifier is measured using accuracy.For MNB classifier best accuracy is achieved using bi-gram feature. And for, LIBLINEAR classifier the best accuracy is achieved using uni-gram feature. The classifier does not perform well on less popular emotion.

1.2 Motivation

– Emotion is essential in all aspects of life. It influences our decision-making, affects our social relationships, shapes our daily behavior, even outlasts our memories.[1] The use of social media has increased in past few years.People have started expressing their thoughts,emotions,opinions through the platform of twitter,instagram,facebook and other social networks.With this increasing trend it is necessary to know the meaning and the emotion of the sentences written by others. Twitter is a popular real time microblogging service,the content generated on it by the users is an important source for predicting people's emotions, which in turn gives a better understanding about their behaviour and action.[2]It leads to tool for accurately extracting semantic information as well as empirical studying the properties of social interaction. Due to relatively small training datasets studies on emotion identification, lack apprehensive content of emotional situations. To overcome this bottleneck, we have used a large emotional-based dataset/content.Learning techniques in order to improve emotion identification in other domains.

1.3 Problem Statement/ Case Study

– Emotions are universal and extend beyond boundaries of language, literature, religion, age, etc.Communicating with people is not just about transmitting information or a message but also expressing your emotions. Through our project we aim to make machines detect emotions from any form of text, which constitutes about 70% of information available to us. Given any form of text, the machine will identify the specific emotions of happiness, sadness, anger,surprise, fear or love that the text expresses. In a world full of blog posts, tweets and emails, the implications for businesses to be able to identify emotions contained in written communications are clear: a greater understanding of their users' behaviours and their desires. Understanding emotions exposes us to an array of possibilities: personalized information generation, like advertisements, search results

and so on, development of powerful human- computer interaction machines and evolution of more intuitive and emotionally characterized text to speech systems.

- The identification of the emotions expressed in text is challenging for somewhat these reasons:
  1. Emotions expressed in text can be deprived of explicit emotion-bearing words. Hence it become difficult to predict emotions purely by keywords.
  2. The other available labelled datasets for emotion are relatively small hence it fails in comprehensive content of emotion handling.

  The dataset used here is labelled with 6 different emotions and is huge as 2.5 million samples. The size of dataset plays an important role in training the model. Big the data less would be the overfitting and would yeild apprehensive coverage of emotions expressed.
- We use different machine leaning concepts/algorithm for sentiment analysis using extracted features. For feature extraction we use TF-IDF(Term Frequency Inverse Document Frequency) and Count Vectorizer. We have implemented Logistic Regression, Principal Component Analysis and Classification(reproduced article) for all with got different accuracies and have shown the results for all.

2 Data Acquisition / Explanation of Data set / Preprocessing

- For a domain like natural language processing it is righteous saying that *"The more the amount of data, the more accurate the results"*. Our dataset has 0.5 million tweets labelled with their respective emotions. Using the 7 sets of emotion words[11] for 7 different emotions from existing psychology literature, and then utilized the Twitter API to collect the tweets that have an emotion word in the form of hashtag[10]. On lemmatizing these hashtags we mapped the emotion to its respective one of the seven classes with the help of set of emotion words. This hashtag was then removed and the tweet was then sent for further preprocessing.

- Totally 0.50 million tweets were collected. Before using these tweets as training examples, it is necessary to verify their quality, i.e., whether the emotion hashtags truly indicate the authors' emotional states. A tweet was labeled as relevant if the emotion hashtag in the tweet reflects the writer's emotion[12].

- A set of filtering heuristics was developed on the aforemen- tioned set of 400 tweets (development set).
  1. We kept only the tweets with the emotion hashtags at the end. Based on our observation and corroborated, if the emotion hashtag is not at the end of a tweet, it is less likely that the hashtag indicates the author's emotional state.

2. We discarded tweets which have less than five words, since they may not provide sufficient context to infer emotions.
3. We removed the tweets which contain URLs or quotations. A large amount of tweets with URLs are information-oriented, which do not convey emotions. Furthermore, we removed all the retweets, non-English tweets and tweets having more than 3 hashtags. The precision on the dataset were 95.08%

– Thus, our filtering heuristics were effective in removing irrelevant tweets. After applying the heuristics on all the collected tweets, we finally obtained a collection of 488,982 tweets.

– The preprocessing part contained many parts[13]:

1. **Tokenization:**
   We tokenized every word for better understading the individual interpretation of words.
2. **POS filtering:**
   We used POS tags to filter the dataset by removing conjuctions, punctutaions, quotes and special symbols.
3. **User Mentions:**
   We replaced user mentions to @ladygaga to @user.
4. **Normalized informal expressions:**
   Converted informal expression to normal english words(e.g $ll \rightarrow will$, $dnt \rightarrow donot$, $coooool \rightarrow cool, etc.$).
5. **Lemmatization:**
   We lemmatized the data to reduce the redundant features and get the root meaning of the words.

3 Machine Learning Concept Used

– First, we can understand how Machine Learning work with Natural Language Processing. Text information requires a special way to deal with Machine Learning. This is on the grounds that text information can have a huge number of words and expressions, however will in general be extremely inadequate. For instance, the English language shares around 100,000 words for all intents and purpose use. In any case, some random tweet just contains two or three dozen of them. This contrasts from something like video content where you have extremely high dimensionality, however you have tons of information to work with, along these lines, it's not exactly as sparse. In our respective implementation of Harnessing Twitter 'Big Data' for Automatic Emotion Identification. We have applied the basic concepts of Machine Learning and annotated the supervised

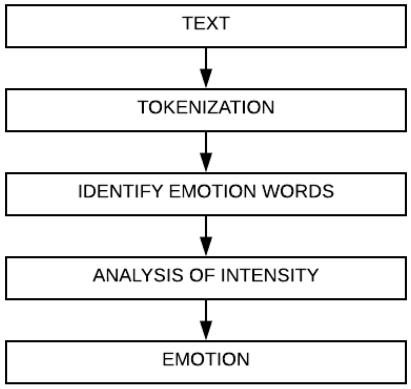


Figure 1: Technique for Text Processing

learning in emotion identification.

Logistic Regression:

Logistic regression is an instance of supervised classification in which the correct label y (either 0 or 1) is known for each observation x. There are two components of logistic regression: first is the metric of the distance between the current label $\hat{y}$ is to the original label y. This distance is loss function or cost function, for logistic regression it is commonly called cross-entropy function. Second is the optimization algorithm for updating the weights to minimize the loss function. The algorithm used for this is gradient descent.

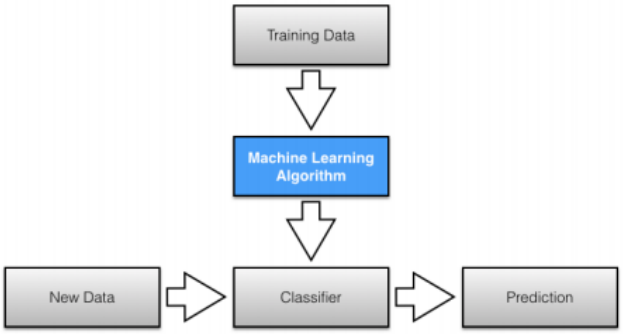
Principal Component Analysis:

Principal component analysis is fundamentally a dimensionality reduction technique that transforms the columns of a dataset into a new set features. It does this by finding a new set of projection (like X and Y axes) that explain the maximum variability in the data information[16]. Dimensionality reduction could be a method which helps us reduce our computation time by compressing the data-set without losing any useful information. this system is very used when there are over hundreds or thousands of measurements for every data sample which frequently results in failure of statistical models likewise. Original Data often contains many redundant, and unimportant features that when removed improves the speed and efficiency of processing and likewise as allow us to figure with more compact representation of information without losing any vital information. PCA (Principal Component analysis) is one such dimensionality reduction technique. Its main aim is to cut back the dimensionality of a dataset consisting of multiple variables which are correlated with one another and at the identical time, retaining maximum variability within the dataset. it's widely utilized in applications like biometric authentication,

computer vision and compression regarding the sector of finance, psychology and bioinformatics. this system transforms the variables/features within the dataset to a group of orthogonal principal components which also are called the eigenvectors of the covariance matrix. they're ordered specified the variable that contributes more variance is listed before the variable that contributes less variance comparatively. Hence, the primary eigenvector or the primary principal component contributes maximum variance. The output generated from PCA are such principal components whose numbers are either lesser or adequate the first variables.

Classification:

Supervised pattern classification is that the task of training a labeled training data which then is accustomed to assign a pre-defined class label to new objects. One example that we'll explore is via multinomial naive Bayes classifiers and linear support vector machine so as to predict whether a brand new text message is categorized in which class(Sad, fear, anger, etc.).



* **Multinomial Naive Bayes:**

  An alternative approach to characterize text documents instead of binary values — is that the term frequency (tf(t, d)). The term frequency is usually defined because the number of times a given term t (i.e., word or token) appears in an exceedingly document d (this approach is typically also called raw frequency). In practice, the term frequency is usually normalized by dividing the raw term frequency by the document length[17].

$$normalizedtermfrequency = \frac{tf(t,d)}{n_d}$$

  where,

  • tf(t, d): Raw term frequency (the count of term t in document d).

  • n_d: The total number of terms in document d.

  The term frequency - inverse document frequency (Tf-idf) is another alternative for characterizing text documents. It will be understood as a weighted term frequency, which is

very useful if stop words haven't been faraway from the text corpus. The Tf-idf approach assumes that the importance of a word is inversely proportional to how often it occurs across all documents. Although Tf-idf is most typically accustomed rank documents by relevance in numerous text mining tasks, like page ranking by search engines, it may be applied to text classification via naive Bayes.

Performance of Multinomial Naive Bayes:

In contrast to the multi-variate Bernoulli event model,the multinomial model captures word frequency information in documents. within the multinomial model, a document is an ordered sequence of word events, drawn from the identical vocabulary.Another point to think about is that the multinomial event model should be a more accurate classifier for data sets that have an outsized variance in document length.The multinomial event model naturally handles documents of varying length by incorporating the evidence of every appearing word[18].However, within the multinomial model more care must be taken. The non-text features shouldn't be added to the vocabulary because then the event spaces for the various features would compete for the identical probability mass although they're not mutually exclusive.

* **LIBLINEAR:**

  LIBLINEAR is an open source library for large-scale linear classification. It supports logistic regression and linear support vector machines. we offer easy-to-use command-line tools and library requires users and developers.

  LIBLINEAR can train large-scale problems very efficiently. It supports linear support vector machine. Also, LIBLINEAR is competitive with or even faster than state of the linear classifiers. As LIBLINEAR is written in an exceedingly modular way, a replacement solver can be easily plugged in. This makes LIBLINEAR not only a machine learning tool but also an experimental platform. Making extensions of LIBLINEAR to languages aside from C/C++ is straightforward.LIBLINEAR is still being improved by new research results and suggestions from users[18]. The ultimate goal is to make easy learning with huge data possible.

4 Pseudo Code/ Algorithm

4.1 Principal Component Analysis

4.1.1 Algorithms:

Algorithm 1 : Normalize

- **procedure** NORMALIZE(X)

1: mu ⟵ mean(x,axis := 0)

2: std ⟵ std(x, axis :=0)

3: a ⟵ X - mu

4: $if(std > 0)$ then new_std := std

5: else $new\_std := 1$

6: $Xbar = a/new_std$

-

Algorithm 2 : PCA

- **procedure** PCA($X, numcomponents$)

1: sum ⟵ 0

2: $\bar{X}$ ⟵ normalize(X)

3: covariance ⟵ $\bar{X}\ \bar{X}^\intercal$

4: S ⟵ covariance

5: eigenvecs, eigenvalues ⟵ eig(S)

6: for i ⟵ 1 to num_components do

B ⟵ concatenate(B, eigenvecs(i)) sum ⟵ sum + eigenvalues(i)

7: P ⟵ $B.B^\intercal$

8: $X\_reconstruct$ ⟵ $P.X^\intercal$

9: $X\_reconstruct$ ⟵ $X\_reconstruct^\intercal$

-

Algorithm 3 : Loss and Variance Calculations

1: $\bar{X}$, mu, std ⟵ normalize(X)

2: for i ⟵ 1 to 5 do

reconst, sum ⟵ PCA($\bar{X}$,i)

error = mse(reconst, $\bar{X}$)

reconst ⟵ reconst*std + mu

reconstructions.append(reconst)

loss.append(i,error)

variance_values.append(i,sum)

4.1.2 Mathematical Representation

(a) Let's consider dataset $X = \{x_1, x_2, ..., x_n\}$, where $x_n \in R^D$ with 0 mean and Covariance matrix given as:

(b)

$$S = \frac{1}{N}\sum_{n=1}^{N}(X_n - \bar{X})(X_n - \bar{X})^\intercal$$

Where $\bar{X}$ is the mean row vector which is zero. Hence:

$$S = \frac{1}{N}\sum_{n=1}^{N} x_n * x_n^T$$

Covariance indicates the direction of linear relationships between variables and the matrix is a symmetric matrix.

(c) S is the covariance matrix, λ is the eigen value and $\vec{v}$ is the eigen vector. There for principal components:

$$S\vec{v} = \lambda\vec{v}$$

$$S\vec{v} - \lambda\vec{v} = 0$$

$$\vec{v}(S - \lambda I) = 0$$

(S-λ I) has to be non-invertible

$$\det((S - \lambda I)) = 0$$

Solving the above equation gives us the set of eigen vectors and corresponding eigen-values.

(d) The eigen values are sorted in decreasing order to form the projection matrix:

(e)

$$B = [\vec{v_1}....\vec{v_m}] \in \mathbb{R}^{nXm}$$

for m $\leq$ d

(f) Projection matrix thus formed is:

$$P = BB^\intercal$$

(g) New feature subspace Z:

$$Z = BB^\intercal X$$

Z= P*X

4.2 Logistic Regression

4.2.1 Algorithms:

Algorithm 4 : Sigmoid Function

- **procedure** SIGMOID(x)

1. $sigmd = 1/(1 + e^{-x})$

2. $sigmd\_grad = x * (1 - x)$

-

4.2.2 Mathematical Representation:

1. **Cross Entropy Loss Function:**

   $L(\hat{y}, y)$ = how much $\hat{y}$ differs from true value of y

   Loss function prefers the correct class labels of the training examples to be more likely. It is called conditional maximum likelihood estimation: the parameters are w,b that maximize the log probability of the true y labels in the training data given the observations x. The resulting loss function is the negative log likelihood loss, generally called the cross-entropy loss.

   Let us consider,

   (a) $\hat{y} = P(y = 1/x)$

   (b) $\hat{y}$ is the probability that y=1, given x.

   (c) $1 - \hat{y} = P(y = 0/x)$

   (d) $p(y/x) = \hat{y}^y(1 - \hat{y})^{1-y}$

   (e) If $y = 1 \implies P(y/x) = \hat{y}$

   (f) Taking log likelihood ; $log(\hat{y}^y * (1 - \hat{y})^{(1-y)})$

   (g) $ylog\hat{y} + (1 - y)log(1 - \hat{y})$

   (h) $-L(\hat{y}, y)$

   (i) $logP(y/x) = -L(\hat{y}, y)$ The negative log likelihood is to maximize the probability by minimizing loss function.

2. **Maximum Likelihood Function :**

   Since we now are using more than two classes the log of the maximum likelihood function becomes:

$$L(\beta_i^T, X) = \sum_{i=1}^{N} log\big(p_i(x_i, \beta_i^T)\big) = \sum_{i=1}^{N} log\Big(\frac{e^{\beta_i^T x_i}}{1 + e^{\beta_i^T x_i}}\Big)$$

3. **The Gradient :**

The derivation of the gradient of the maximum likelihood function below:

$$\begin{aligned}\frac{\partial}{\partial\beta}L(\beta,x_i) &= \sum_{i=1}^{N}\frac{\partial}{\partial\beta}y_i\beta^T x_i - \frac{\partial}{\partial\beta}ln(1+\mathrm{e}^{\beta^T x_i}) \\ &= \sum_{i=1}^{N} y_i x_i - \frac{x_i \mathrm{e}^{\beta^T x_i}}{1+\mathrm{e}^{\beta^T x_i}} \\ &= \sum_{i=1}^{N} x_i(y_i - p_i)\end{aligned}$$

Turning this into a matrix equation is more complicated than in the two-class example — we need to form a N(K1)(p+1)(K1) block-diagonal matrix with copies of X in each diagonal block matrix

4.3 Classification (Reproduced Article)

4.3.1 Mathematical Representation:

1. **Linear SVM**

   - **SVM Hypothesis:-**

     Hypothesis, w.r.t. a machine learning model is the model itself, which is nothing but our classifier (which, is a function).

     $$h_{w,b}(x) = g(w^T x + b)$$

   - **Class Labels :-**

     Class labels are denoted as from 0 to 6 for different emotions.This is a convex optimization problem, with a convex optimization objective function and a set of constraints that define a convex set as the feasible region. Convex functions look like a bowl placed right-side-up. Convex set is a set of points in which a line joining any two points lies entirely within the set. I would have loved to talk on these in more detail, but it would be more convenient to just google the terms in italics. Before delving into the actual part, we should be familiar with two terms- Functional margin and Geometric margin. Optimization problem that the SVM algorithm solves :-

$$\phi(w) = \frac{1}{2} w^T w \; - \; minimize$$

$$Subject\ to\ d_i(w^T x + b) \geq 1\ \forall\ i$$

– **Functional margin and Geometric margin :-**

Following is how we are going to notate the hyperplane that separates the different emotions examples throughout this article:

Equation of separating hyperplane; w is the normal to the hyperplane

$$\pi\colon\ \mathrm{w}^T X^{(i)} + b = 0$$

Each training example is denoted as x, and superscript (i) denotes ith training example. In the following section y superscripted with (i) represents label corresponding to the ith training example.

– **Functional margin of a hyperplane w.r.t. ith taining example is defined as:-**

$$\hat{\gamma}^{(i)} = \ y^{(i)}(w^T x^{(i)} + b)$$

– **Geometric margin of a hyperplane w.r.t. ith training example is defined as functional margin normalized by norm(w):-**

$$\gamma^{(i)} \ = \ y^{(i)}(\frac{w^T x^{(i)} + b)}{||w||}$$

– **This makes our optimization problem as :-**

$$(w^*, b^*) = \mathrm{argmax}_{w,b} \frac{2}{||w||}$$

$$\text{such that,}$$

$$y^{(i)}(w^T x^{(i)} + b) \geq 1$$

2. **Multinomial Naive Bayes**

The dependent feature vector is $(x_1, x_2, ..., x_n)$ and the class is C_k

$$P(C_k \mid x_1, \ldots, x_n) = \frac{P(C_k)P(x_1, \ldots, x_n \mid C_k)}{P(x_1, \ldots, x_n)}$$

Now, for each given class C_k each feature vector x_i is conditionally independent of other feature vectors x_j for i $\neq j$

$$P(x_i \mid C_k, x_1, \ldots, x_n) = P(x_i \mid C_k)$$

Thus it can simplified as:

$$P(C_k \mid x_1, \ldots, x_n) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)}{P(x_1, \ldots, x_n)}$$

Since $P(x, \ldots, xn)$ is constant, if the values of the feature variables are known, the following classification rule can be used:

$$P(C_k \mid x_1, \ldots, x_n) \propto P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)$$
$$\Downarrow$$
$$\hat{y} = \arg\max_k P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)$$

Log probabilities can be used to avoid underflows:

$$\hat{y} = \arg\max_k \left(\ln P(C_k) + \sum_{i=1}^{n} \ln P(x_i \mid C_k)\right)$$

The multinomial distribution is parametrized by vector $\theta_k = (\theta_k 1, \ldots, \theta_k n)$ for each class C_k,where n is

the number of features and $\theta_k i$ is the probability $P(x_i|C_k)$ of feature i

The parameters θ_k is estimated by maximum likelihood,

$$\hat{\theta}_{ki} = \frac{N_{ki} + \alpha}{N_k + \alpha n}$$

where $N_k i$ is the number of times feature i appears in a sample of class k in the training set T, and N_y is the total count of all features for class C_k. The smoothing priors

$\alpha \neq 0 accounts for features not present$

in the learning samples and prevents zero probabilities in further computations. Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$is called Lidstone smoothing.

Thus, it can be represented as:

$$\hat{y} = \arg\max_k (\ln P(C_k) + \sum_{i=1}^{n} \ln \frac{N_{ki} + \alpha}{N_k + \alpha n})$$

5 Coding and Simulation

5.1 Simulation Framework

PCA:

- Num_components: It is the number of features which are to be reduced.
- S = The covaraiance to find the eigen value and eigen vector.
- p = projection matrix for pca
- x_reconstruction = Reconstructed feature matrix.
- num_datapoints = Number of rows/samples of the dataset.
- B = Reconstructed eigen vector.

Logistic Regression:

- L_data = Converting data into list.
- data_dict = Converting data to dictionary.
- Tf-idf = Converting word to vector.
- x = input for the sigmoid function.

Classification(reproduced article):

- lbl_enc = Randomly assign labels.
- γ = Functional margin to hyperplane.
- $\phi(w)$ = Optimization problem

5.2 Results:

The codes of Logistic Regression, PCA and reproduced article(Classification) are attached below.

URL links: To see Code, Results and datasets :- CLICK ME

Logistic Regression Results:

Results Of Logistic Regression :

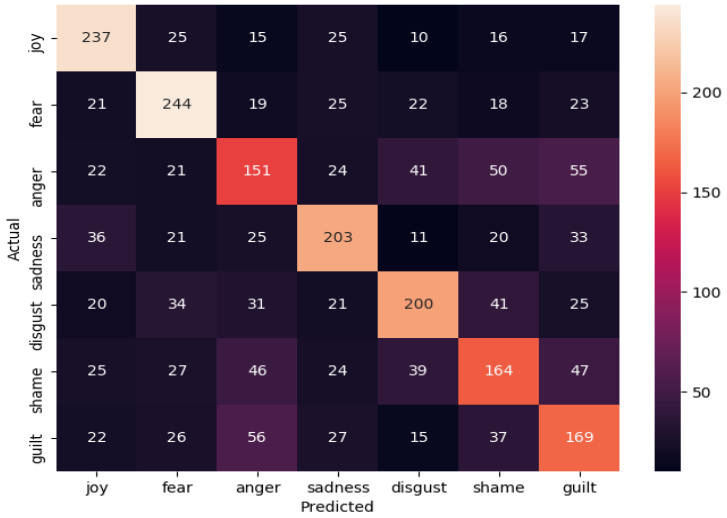


Figure 2: Confusion Matrix

Inference :

The confusion matrix is a table test which describes the performance of the classification model on the test data for which the true values are already known, in order to confusion to evaluate the model. For example: In our model you can see the shaded which represents the actual true positive predictive value of joy with joy and the region with the shade of black represents the false negative- where the predicted value was fear but the actual value was joy.

```
RangeIndex: 7652 entries, 0 to 7651
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   emotions  7652 non-null   object
 1   text      7652 non-null   object
dtypes: object(2)
memory usage: 119.7+ KB
              precision    recall  f1-score   support

         joy       0.62      0.69      0.65       345
        fear       0.61      0.66      0.63       372
       anger       0.44      0.41      0.43       364
     sadness       0.58      0.58      0.58       349
     disgust       0.59      0.54      0.56       372
       shame       0.47      0.44      0.46       372
       guilt       0.46      0.48      0.47       352

    accuracy                           0.54      2526
   macro avg       0.54      0.54      0.54      2526
weighted avg       0.54      0.54      0.54      2526
```

Figure 3: Accuracy

Principal Component Analysis Results:

- **Sorted Eigenvalues in decreasing order:**

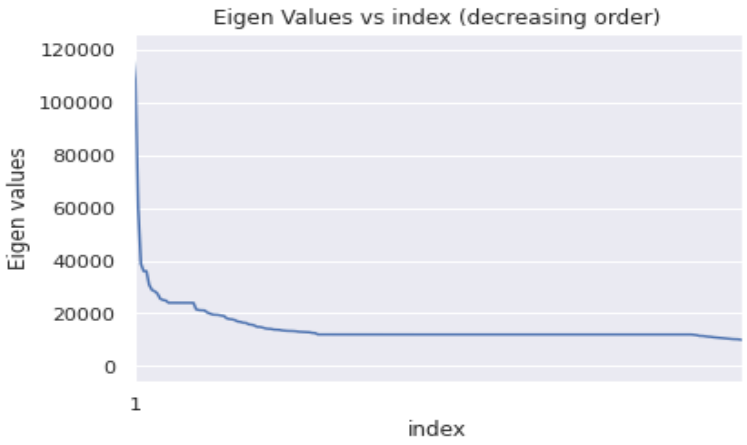


Figure 4: eigen values vs index

- **Inference:**

  The above graph shows the sorted eigenvalues in decreasing order. We have 300 features, and corresponding to that we have 300 eigenvalues and 300 corresponding eigenvectors found from the covariance matrix S. We sort the eigenvalues in decreasing order and eigenvectors correspondingly.Hence, higher the eigenvalue; higher will be the variance presented by the eigenvector.

- **Maximum Captured Variance:**

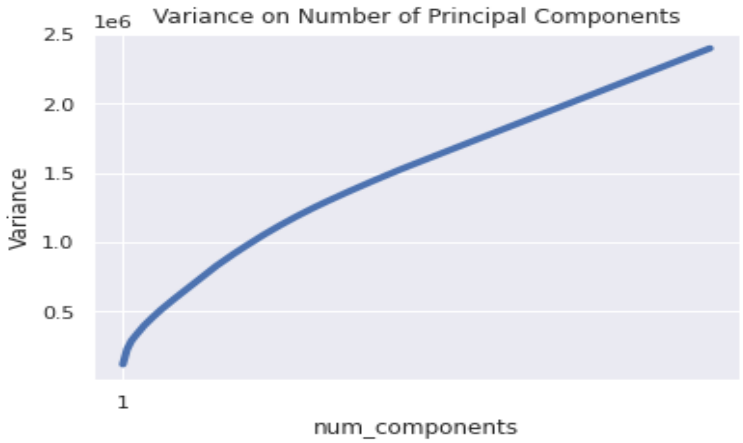


Figure 5: Variance

- **Inference:**

  The above graph shows the maximum amount of variance with respect to principal components. Maximum captured variance is characterized by the aggregate of the eigenvalues of the principal components, it means as we go to the right, number of principal components increases and hence, change would likewise increment as the quantity of variance would also increase as the number of eigenvalues in the addition is increasing. But since, eigenvalues are sorted in decreasing order, at first there is a rapid increase in the captured variance due to high eigenvalues, whereas as we go to the right, the eigenvalues that are adding have less value, henceforth it is then expanding yet at a more slow rate.

- **MSE vs Number of PCA(Principal component Analysis) for sorted eigenvalues:**

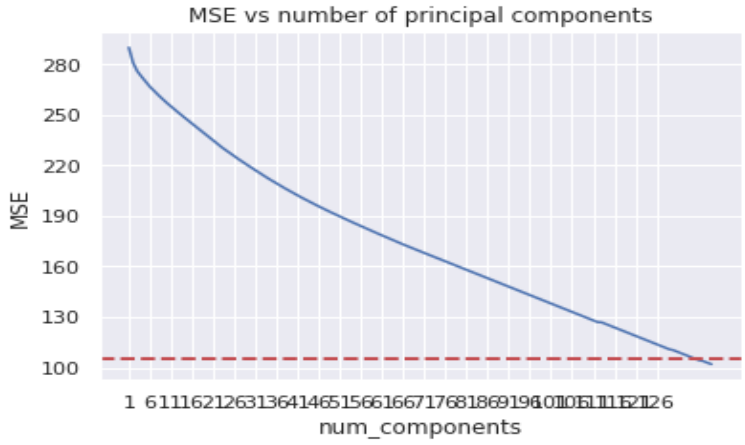


Figure 6: MSE vs Principal Components(sorted)

- **Inference:**

The above graph shows that the mean sqaured error with respect to the number of principal component. The explanation is that as we take increasingly more principal components, there is less loss in the dimensions and hence, there is less misfortune in the data. Therefore, its natural that error will decrease as we take more principal components.

- **MSE vs Number of PCA(Principal Component Analysis) with unsorted eigenvalues:**

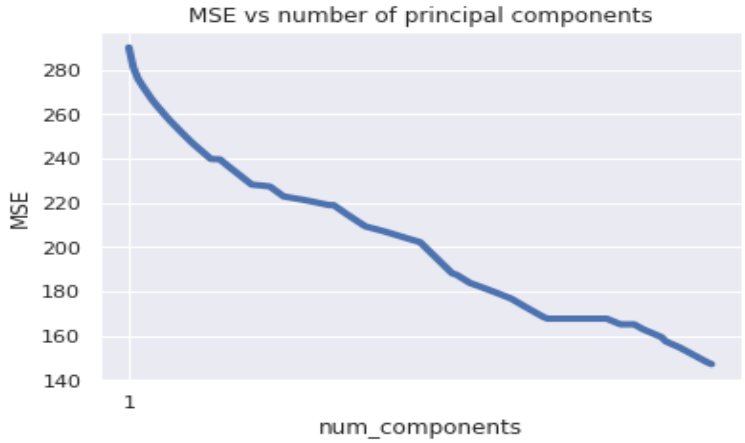


Figure 7: MSE vs Principal Components(unsorted)

Inference:

he above graph shows the mean squared error with respect to the number of principal components. The MSE error is computed between projected data matrix and the original data. The eigen values are non-ordered and not sorted before applying PCA to compute the projected data matrix and computing the MSE error. Since, there are 300 features the features are significantly large and the eigenvalues is unordered, the MSE error first decreases then for some values becomes constant and then again decreases.

- **Variance Ratio:**

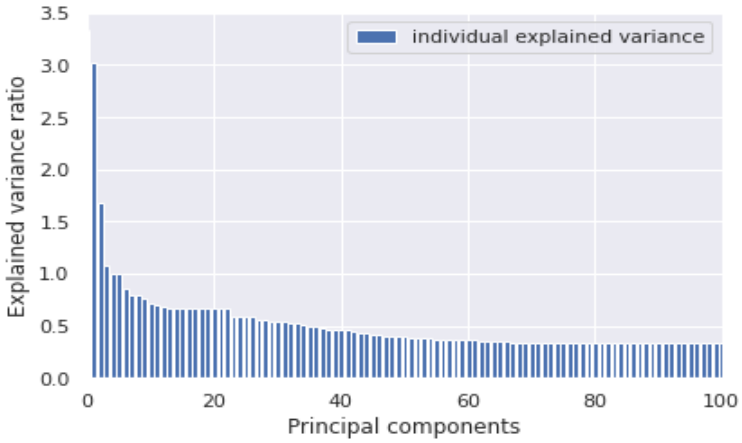


Figure 8: Explained variance ratio

Inference:

After sorting the eigen pairs, the above plotted graph is the result of variance to each of the principal components. It can be inferred from the graph that the first two principal component contribute to maximum variance. Hence we choose first two components as low dimensional space for further analysis.

Classification(Reproduced Base Article)Results:

- **No. of Samples Vs Accuracy :**

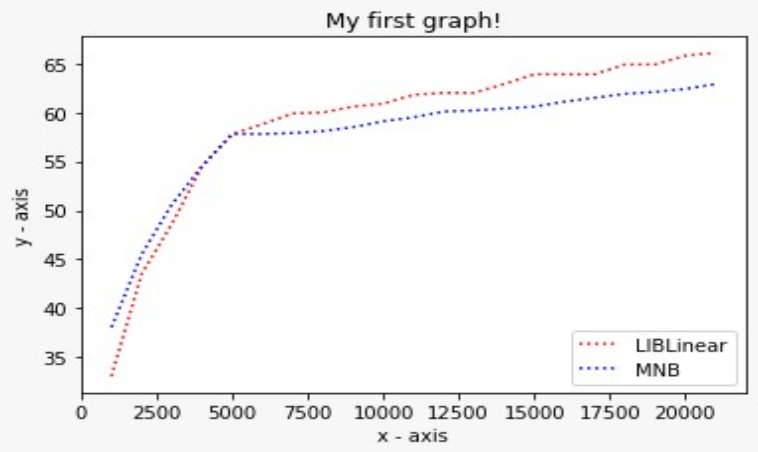


Figure 9: Accuracy

- **Inference:**

  It can infered from the figure, that increasing the dataset from 1000 samples upto 20,000 samples results in gain of accuracy upto 65% with LIB-Linear classifier. Whereas with MNB classifiers we obtain an

accuracy gain upto 62%. For around 5200 samples the accuracy for LIb-Linear and MNB classifiers is around 57%.

- **Models for computation Vs Accuracy:**

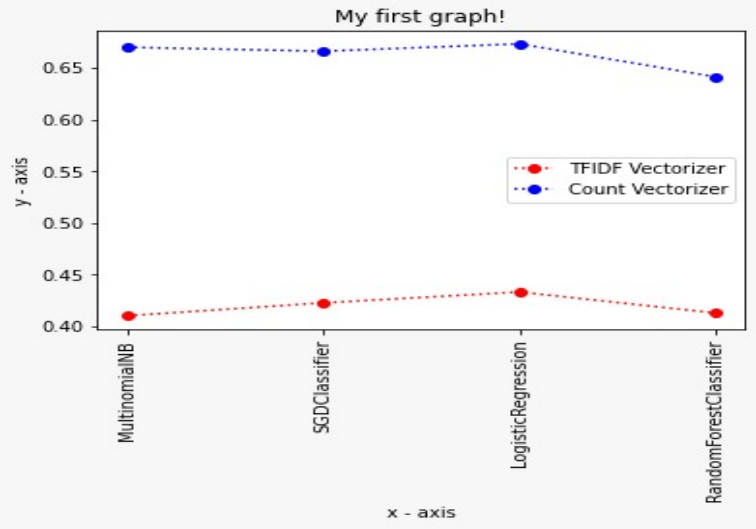


Figure 10: computational models against accuracy

- **Inference:**

  There are two methods for extracting the features and for different machine learning algorithm they give different accuracies.The highest accuracy is obtained for Logistic Regression with Count Vectorizer.Also the count vectorizer gives better accurate results as compared to TFIDF vectorizer.

Students are advised to share the new derivations with results in correlation with the reproduce results. Write clear inference for the new results. You are also advised to add new analysis along with the codes.

6 Conclusions

- We used 0.5 million emotion tweets with 7 different emotions for automatic emotion identification. Before training the dataset, data preprocessing was done to make it suitable to feed into the model. We implemented four machine learning concepts namely: Logistic regression, PCA, Classification with LIBLINEAR and Multinomial Naive Bayesian classifiers.

- In classification using many of known approaches of our base article we discovered that Multinomial naive bayes and Liblinear are most efficient for the classification of data. The results show that maximum accuracy is achieved by using parts-of-speech,lemmatization/stemming. The accuracy achieved was 65% using 20,000 samples. The accuracy can be increased by increasing the training dataset.

- In logistic regression the redundancy of data was removed by using different natural language processing concepts in which lemmatization of the text played a vital role in minimizing our feature space and provided us with relevant meaning bearing features. Using these features we compared TF-IDF vectorization and constructed feature matrix. We then employed four different machine learning concepts and discovered that Multinomial naive bayes and Liblinear are effective for twitter data set. From these results we got accuracy of 55% and then after, produced the confusion matrix which helped us to analyse the performance of our respective model.
- In Principal Components Analysis the main goal was to reduce the dimensionality of the features. For which we formed a feature matrix containing the most frequently words as feature. For that the concept used was TF-IDF vectorizer and Count Vectorizer. We used PCA tool to reduce the complexity and size of data set with minimum information. It was also observed that taking into account more number of features gives better output for mse graph i.e. the error between actual data set and when projected onto reduced subspace was minimum.
- For future works we can develop an automatic identification machine to classify the emotion of a neutral tweet without any emotion/hashtags attached to text.Also using the current tweeter API, news API's and employing emotion detection we can derive a summary generator which can predict the perspective and tone of the data/text information to indicate the situation of real time.

7 Contribution of team members

7.1 Technical contribution of all team members

Tasks	Jay Patel	Manav Shah	Prima Sanghvi	Priyanshi Deliwala
Logistic Regression	Integration	Implementation	Mathematical Analysis	Preprocessing
PCA	Feature matrix extraction Coding	Data Handling	Coding	Mathematical analysis
Reproduce Article	Coding	Data Acquisition	Mathematical analysis	Coding

7.2 Non-Technical contribution of all team members

Enlist the non-technical contribution of members in the table. Redefine the tasks (e.g Task-1 as report writing etc.)

Tasks	Jay Patel	Manav Shah	Prima Sanghvi	Priyanshi Deliwala
Research	Base article	Base article	Dataset	Dataset
Abstract	Introduction	Background	Contribution	Motivation
LR-report	Mathematical Analysis	Introduction	Mathematical Analysis Code	Results Inference
PCA-report	Figure Allocation, References	Case-2	Inference, Background	Case -1
Final Report	Mathematical Analysis,background	Section-2 and Section-6	Conclusions Inferences Algorithms	Mathematical Analysis Section 3 and Section 1

References

[1] W. Wang, L. Chen, K. Thirunarayan and A. P. Sheth *.W. Wang, L. Chen, K. Thirunarayan and A. P. Sheth, "Harnessing Twitter "Big Data" for Automatic Emotion Identification," 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, Amsterdam, 2012, pp. 587-592.*

[2] G. Mishne, *"Experiments with mood classification in blog posts," in Proceedings of ACM SIGIR 2005 Workshop on Stylistic Analysis of Text for Information Access*

[3] M. D. Choudhury, S. Counts, and M. Gamon, *"Not all moods are created equal! exploring human emotional states in social media," in Proceedings of ICWSM, 2012.*

[4] S. Mohammad, *"emotional tweets," in Proceedings of the Sixth Inter- national Workshop on Semantic Evaluation. ACL, 7-8 June 2012, pp.246–255.*

[5] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, *"Affect analysis model: Novel rule-based approach to affect sensing from text," Natural Language Engineering, vol. 17, no. 1, pp. 95–135, 2011.*

[6] C. Strapparava and R. Mihalcea, *"Learning to identify emotions in text," in Proceedings of the 2008 ACM symposium on Applied computing. ACM, 2008, pp. 1556–1560.*

[7] C. Strapparava and R. Mihalcea, *" "Semeval-2007 task 14: affective text," in Proceedings of the 4th International Workshop on Semantic Evaluations, ser. SemEval '07, 2007, pp. 70–74.*

[8] R. Tokuhisa, K. Inui, and Y. Matsumoto, *"Emotion classification using massive examples extracted from the web," in Proceedings of COLING. ACL, 2008, pp. 881–888.*

[9] C. Yang, K. Lin, and H. Chen, *"Emotion classification using web blog corpora," in IEEE/WIC/ACM International Conference on Web Intelligence. IEEE, 2007, pp. 275–278.*

[10] *ISEAR - DATASET*

[11] *Mining Twitter*

[12] *Text_Emotion*

[13] Weng, Jiahao. "NLP Text Preprocessing: A Practical Guide And Template". Medium, 2016, *https://towardsdatascience.com/nlp-text-preprocessing-a-practical-guide-and-template-d80874676e79.*

[14] Kim, Ricky. *"Another Twitter Sentiment Analysis With Python — Part 1". Medium, 2017, https://towardsdatascience.com/another-twitter-sentiment-analysis-bb5b01ebad90.*

[15] Shaikh, Javed. *"Machine Learning, NLP: Text Classification Using Scikit-Learn, Python And NLTK.". Medium, 2017, https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a.*

[16] Prabhakaran, Selva. *"Principal Component Analysis (PCA) - Better Explained — ML+". Machine Learning Plus, 2019, https://www.machinelearningplus.com/machine-learning/principal-components-analysis-pca-better-explained/.*

[17] Sergey Smetanin *"Sentiment Analysis Of Tweets Using Multinomial Naive Bayes". Medium, 2018, https://towardsdatascience.com/sentiment-analysis-of-tweets-using-multinomial-naive-bayes-1009ed24276b.*

[18] Krishna Kumar Mahto. *""Demystifying Maths Of SVM". Medium, 2019, https://towardsdatascience.com/demystifying-maths-of-svm-13ccfe00091e.*