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

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

- **Group No : 8**
- **Name of the group members:**
  1. Aditya Shah (1741007)
  2. Dhruvil Shah (1741024)
  3. Nilay Patel (1741038)
  4. Varun Patel (1741080)
- **Project Title: Sentimental Analysis on movie reviews**
- **Project Area: Natural Language Processing**

1 Introduction

1.1 Background

- In today's world, a bulk of information can be obtained from online documents. For a better organization of this information for users, researchers have been investigating the problem of automatic text categorization. The past few years have seen rapid growth in on-line review sites where a crucial characteristic of the posted articles is their sentiment, or overall opinion towards the subject matter for example, whether a product review is positive or negative. Labeling these articles with their sentiment would provide succinct summaries to readers; indeed, these labels are part of the appeal and value-add of such sites as www.rottentomatoes.com, which both labels movie reviews that do not contain explicit rating indicators and normalizes the different rating schemes that individual reviewers use. Sentiment classification would also be helpful in business intelligence applications, where user input and feedback could be quickly summarized; indeed, in general, free-form survey responses given in natural language format could be processed using sentiment categorization. Moreover, there are also potential applications to message filtering.
- Our work is different from [1] from the aspect that we have given polarity values 0 and 1 after classifying the reviews as positive and negative.We have got the idea of giving polarity to positive and negative reviews from [2]. The article [3] focuses on determining the review as a whole whereas in our work we intend to classify each sentence in the review and then classify review on a whole.The work [4]

focuses on classifying the texts based on their genres. However, this technique does not address our general problem of determining what the opinion actually is of the user based on the text entered.The works [5] and [6] focuses on the classification the text based on pre-determined set of seed words which are determined based on human instincts.But, however as indicated below in section 1.3 human instincts are not always reliable.Other relatable works include [7], [8], [9], [10].

- Our work is most closely related to [11] which we have chosen as our base article. [12] work on classification of reviews is also close to our work.It applies a specific unsupervised learning technique based on the mutual information between document phrases and the words "excellent" and "poor", where the mutual information is computed using statistics gathered by a search engine.

1.2 Problem Statement/ Case Study

- Instincts are different from person to person making sentimental analysis more difficult. Using ML methods on categorizing the texts may result into low performance.Whereas on the contrary, human beings find it relatively easy to differentiate positive reviews from the negative ones. Many people use certain words to show their strong sentiments which may prove to be reliable for text classification. To test the latter theory, we requested two of our peers to choose some words for positive and some for negative reviews. Their choice are believable on the first look. We first got a list of positive sentimental and negative sentimental words from the two peers. We had a data set of around 1400 reviews. Using this dataset, we calculated the accuracy of each of the peers which came out to be around 70 to 75 percent of each. We then made a list of some positive and negative words based on automated corpus-based techniques and found its accuracy.We used the same dataset for this technique as used before. The accuracy came out to be around 80 percent which is better than that of human instinct. Hence, we can conclude that it is worthy to experiment on corpus-based techniques rather than depending just on instincts.

2 Data Acquisition / Explanation of Data set

- Our dataset contains movie reviews along with their associated binary sentiment polarity labels with intention of serving as benchmark for data classification. The core dataset contains 50,000 reviews which are splitted equally into 25000 train and 25000 test sets. We have included an additional 50,000 unlabeled documents for unsupervised learning. In the train/test sets, a negative review has a score of 4 or less out of 10 whereas a positive review has a score of 7 or more out of 10 and neutral reviews are not included in the train/test sets. However, in the unsupervised set, reviews of any rating are included.

3 Machine Learning Concept Used

- Logistic Regression:

  Logistic regression is less prone to over-fitting. It not only gives a measure of how relevant a predictor is, but also it's direction of association. It is easier to implement, interpret and very efficient to train. Logistic regression performs well when the data set is linearly separable. We use logistic regression when we have a binary or a dichotomous output attribute. Also when we have explanatory input attributes that we think are related to output attribute.

  We are using logistic regression for large input data. We are using a regularization parameter and cross-validation to find the best estimator to make model and then fitting model and predicting the model to give binary output (1(pos prediction) or 0 (neg prediction)). By this model is properly predicting binary output.

- PCA:

  PCA is important to reduce the problem of overfitting. Overfitting is caused when the no of features used in the training data set for training the model are very large in number. This can lead to creation of a model which is overly generalized. Therefore in many test cases it can give false results thereby declining the accuracy rate and increasing the error rate. PCA which is a technique of dimensionality reduction helps to extract out the most relevant features and removes the unnecessary or less relevant features. Thus it helps to build a model which is well generalized giving high accuracy rate.

  We have taken dataset of movie reviews. From that dataset, we have found features to find PCA. We have separate out the features and target values. Then we have found out Principal Component Analysis (PCA) through eigenvalues, eigenvectors, mean, covariance. We have used K-means clustering to plot the pairwise relationship of projected data (Seaborn pairplot).

- SVM Classifier:

  SVM is capable of doing both classification and regression. The benefit is that you can capture much more complex relationships between your datapoints without having to perform difficult transformations on your own. SVM's are mostly used for classification problems. Generally SVM's are very good when you have huge no of features. For example text classification.

  We use SVM to classify data points having large features as pos and neg by using appropriate parameter. By this we are also finding precision, accuracy and also confusion matrix.

4 Analysis/ Pseudo Code

- Logistic Reggression:

  - Analysis:

    $0 \leq R_\theta(x) \leq 1, \quad h_\theta(x) = g((\theta)^T \cdot x), \quad g(x) = \frac{1}{1+e^{-x}}$; g(x) is Logistic function

    $\therefore h_\theta(x) = \frac{1}{1+e^{-\theta^T \cdot x}}$

    $Trainingset = (x_1, y_1), ..., (x_m, y_m), \quad$ m examples x$\epsilon[x_0, x_1, ..., x_n]$

    $x_0 = 1$, y = 0 or 1

    Cost function is given by:

    $J_\theta = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2}(h_\theta x_i - y_i) \quad = \frac{1}{m} \sum_{i=1}^{m} cost(h_\theta x_i, y)$

    where,

    $cost(h_\theta(x), y) = \frac{1}{2}(h_\theta x - y)^2$

    $cost(h_\theta(x), y) = -log(h_\theta(x))$ if y=1

    $cost(h_\theta(x), y) = -log(1 - h_\theta(x))$ if y=0

    $cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1-y)log(1 - h_\theta(x)$

    If y=1, $\quad cost(h_\theta(x), y) = -log(h_\theta(x))$

    If y=0, $\quad cost(h_\theta(x), y) = -log(1 - h_\theta(x))$

    $\therefore J_\theta = -\frac{1}{m}[\sum_{i=1}^{m} y_i log(h_\theta(x_i)) + (1-y_i)log(1 - h_\theta(x_i))]$

    To minimize cost function, Apply Gradient descent Algorithm,

    Repeat until convergence ($\theta_j := \theta_j - \alpha \frac{\Delta J}{\Delta \theta_j}$)

    Simultaneously update all θ_j,

    $\frac{\Delta J}{\Delta \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta)(x_i) - y_i) x_j^{(i)}$

    $\therefore$ Repeat ($\theta_j = \theta_j - \alpha \sum_{i=1}^{m} (h_\theta x_i - y_i) x_j^{(i)}$)

    $\therefore \theta = [\theta_0, ..., \theta_n]^T$

  - Pseudo Code:

    X = $[x_0, x_1, x_2, x_3, ..., x_n]^T$ // Input Vector

    Y = $[y_0, y_1, y_2, y_3, ..., y_n]^T$ // Input Vector

    $\theta = [\theta_0, \theta_1, \theta_2, ..., \theta_n]$ // Regression Parameter Vector

    $h_\theta(x) = \frac{1}{1+e^{-\theta^T \cdot x}}$ // Predictor function

    Repeat //Gradient Descent Algorithm

    $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) x_j^{(i)}$

    $UntilConvergence$

- PCA:
  - Analysis:

    The covariance between any two features x_i and x_j o given by,

    $\text{cov}(x_i\ ,\ x_j) = \sigma_j = \frac{1}{n} \sum_{k=1}^{k=n}(x_i\text{ - }\mu_i)(x_j\text{ - }\mu_j)$

    Here, μ_i and μ_j are sample means of features i and j respectively.

    For the given dataset, $\sum$ will be,

    $$\sum = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & . & . & . & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & . & . & . & \sigma_{2d} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ \sigma_{d1} & \sigma_{d2} & . & . & . & \sigma_d^2 \end{bmatrix}$$

    Next, obtain the eigenvalues and eigenvectors for the corresponding covariance matrix.

    Suppose an eigen vector v which satisfies the following conditions:

    $\sum$ v = λv

    $\sum$ v = λIv

    $\sum$ v - λIv = 0

    ($\sum$ - λI)v = 0.....(1)

    This is similar to homogeneous system of linear equations.

    For $\vec{v} \neq \vec{0}$ eq.(1) to be true.

    $|\sum - \lambda I| = 0$

    Solving the determinant we obtain d values of λ which represent eigenvalues corresponding to d eigenvalues. Corresponding to d eigenvalues we get d eigenvectors by putting value of λ in equation (1) and solving for v.

    Now, out of d eigenvectors, select k eigenvectors corresponding to k highest eigenvalues. Now, take these k eigenvectors in decreasing order of their corresponding eigenvalues and construct a projection matrix W, where W ϵR^{dXk}

    The k columns of W represent principal components. Using W, we can transform a sample vector x into the PCA subspace obtaining $x^{'}$,

    $x^{'}$ = x W

    1Xk 1Xd dXk

    Similarly, we can transform the entire dataset onto k principal components by calculating the matrix product,

$X^{'}$ = XW

nXk nXd dXk

The dataset reduced to k features. Thus, our goal of dimensionality reduction is achieved.

- Pseudo Code:

  1) Let X be a nxd order input matrix (d dimensional data set).

  X = $[x_0, x_1, x_2, x_3, ..., x_d]^T$

  2)Standardize the given data set (Make sample means of each feature zero and variance of each feature equal to 1).

)Compute the Covariance matrix C of order (dxd) using X.

  4)Compute Eigenvalues and Eigenvectors for the obtained covariance matrix C.

  5)Select k eigenvectors corresponding to k greatest eigenvalues out of d eigenvalues obtained.

  6)These k eigenvectors are Principal components of the data set.

  7)Arrange these eigenvectors in the decreasing order of their corresponding eigenvalues to obtain a projection matrix W of the order(dxk).

  8)Now normalize the data set by computing the matrix product of X^T and W. X' = X^TW

  9) The obtained data set X' is dimensionally reduced data set as being of the order nxk.

- SVM Classifier:

  - Analysis:

    Consider a system where we want to classify data points as positive and negative.

    $\vec{w}$ = A vector of small length, $\vec{u}$ = Unknown point

    Now taking dot product of $\vec{w}$ and $\vec{u}$, we get $\vec{w} \cdot \vec{u} \geq$ c, where c is a constant

    or we can say that without the loss of generality $\vec{w} \cdot \vec{u}$ + b $\geq$ 0 -> Tront Decision Rule

    $\vec{w}$ has to be perpendicular to the decision boundary but we do not know the length nor $\vec{b}$

    c = -b

    Taking a positive sample, $\vec{w} \cdot \vec{x}_+$ + b $\geq$ 1

    Taking a negative sample, $\vec{w} \cdot \vec{x}_-$ + b $\leq$ -1

    Let's introduce a variable y_i such that y_i = +1 for positive samples and y_i = -1 for negative samples.

    For positive samples, y_i (x_i + b) $\cdot \vec{w} \geq 1$

    For negative samples, y_i ($x_i \cdot \vec{w}$ + b) $\geq 1 \implies y_i$ ($x_i \cdot \vec{w}$ + b) - 1 ≥ 0

    Now, y_i ($x_i \cdot \vec{w}$ + b) - 1 = 0

    For x_i on lines passing through the support vectors.

    width = ($\vec{x_+}$ - $\vec{x_-}$) $\cdot \frac{\vec{w}}{||\vec{w}||}$

$\vec{x_+} \cdot \vec{w} = 1$ - b, $\vec{x_-} \cdot \vec{w} = 1$ - b

width = $\frac{2}{||\vec{w}||}$

We need to maximize the width i.e. max($\frac{2}{||\vec{w}||}$) or max($\frac{1}{||\vec{w}||}$) Dropping the constant or min($||\vec{w}||$) or min($\frac{1}{2}$ $(||\vec{w}||)^2$).

In order to maximize the width,

L = $\frac{1}{2}$ $(||\vec{w}||)^2$ - $\sum$ α_i [y_i ($\vec{w} \cdot x_i$ + b) - 1]

Taking partial derivative w.r.t. $\vec{w}$ and setting it to 0,

$\frac{\partial L}{\partial \vec{w}}$ = $\vec{w}$ - $\sum$ α_i y_i x_i = 0

$\therefore \vec{w} = \sum_i \alpha_i\ y_i\ x_i$

Taking partial derivative w.r.t. b,

$\frac{\partial L}{\partial b}$ = - $\sum$ α_i y_i = 0

$\sum$ α_i y_i = 0

$\therefore$ L = $\frac{1}{2}$ ($\sum$ α_i y_i $\vec{x_i}$) ($\sum$ α_j y_j $\vec{x_j}$) - $\sum$ α_i y_i $x_i \cdot$ ($\sum$ α_j y_j $\vec{x_j}$) - $\sum$ α_i y_i b + $\sum$ α_i

$\therefore$L = $\sum$ α_i - $\frac{1}{2}$ $\sum_i \sum_j$ α_i α_j y_i y_j x_i x_j

We need to maximize L,

If $\sum$ α_i y_i $\vec{x_i} \cdot \vec{u}$ + b $\geq$ 0 then, positive samples else negative samples.

Here, $\vec{u}$ is the unknown point.

– Pseudo Code: Define number of features+1 as F and SVs+1 as SV

FOR each SV

FOR each feature of the SV

Read streamed data

Convert it to float

Store into array_SVs [SV][F]

END FOR

END FOR

Read Streamed data

Convert it to float

Store into array_ay [0] (b value)

FOR each SV

Read Streamed data

Convert it to float

Store into array_ay [SV]

END FOR

FOR each feature

```
    Read Streamed data
    Convert it to float
    Store into array_test [F]
  END FOR
  FOR each feature
    Clear_array_AC[F]
  END FOR
  FOR each SV
    FOR each feature of the SV
      array_AC[F] += array_ay[SV]*array_SVs[SV][F]
    END FOR
  END FOR
  FOR each feature
    Distance_value += array_AC[F] * array_test[F]
  END FOR
  Distance_value -=b
  IF (Distance_value≥th)THEN
    RETURN 1
  ELSE
    RETURN -1
  END IF
```

5 Coding and Simulation

5.1 Simulation Framework

- Logistic regression:
  C = [0.001,0.01,0.1,1,10]
  cv(cross validation) = 5
- PCA:
  n_components=4
  n_clusters = 4
- SVM:
  SGD classifiier loss =hinge
  random_state = 42

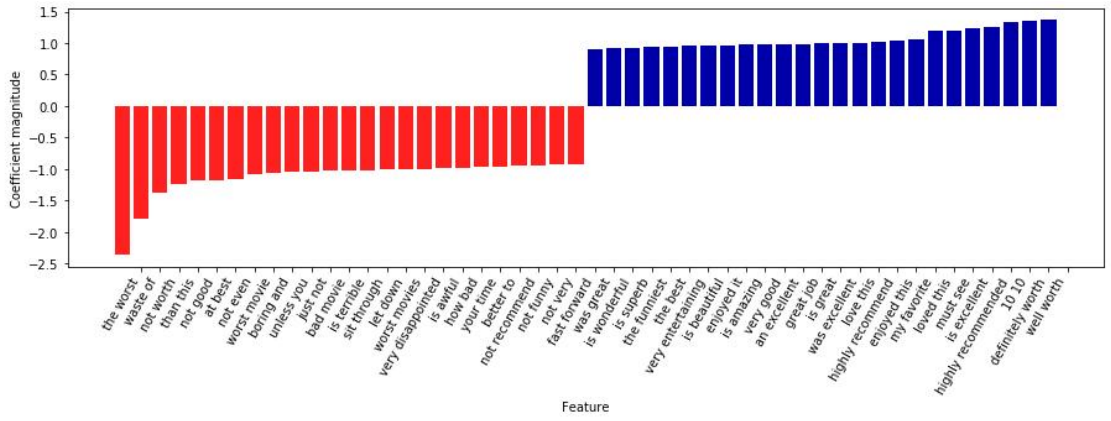
5.2 Results

- Used Tool - Python
- Figure-1: Result of Base Article

<bound method NDFrame.describe of

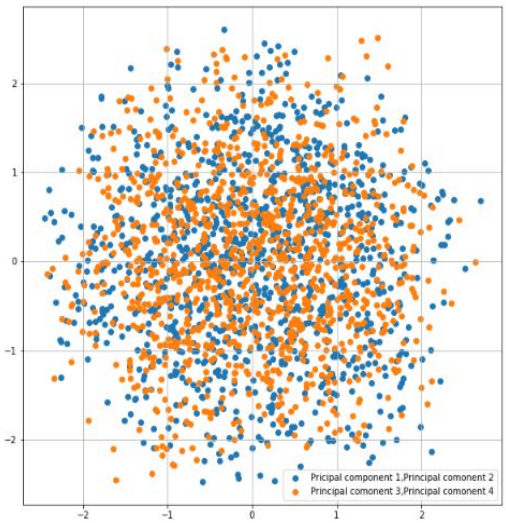
	Features:	MaxEnt	NB	SVM
0	unigrams(freq.)	NaN	0.856081	0.85652
1	unigrams(pres.)	0.87168	0.858080	0.85304
2	unigrams and bigrams(pres.)	0.88716	0.882520	0.88308
3	bigrams(pres.)	0.84916	0.870280	0.84856
4	bigrams(pres.)	0.84916	0.856960	0.84856
5	bigrams(pres.)	0.84916	0.856960	0.84856
6	unigrams+POS	0.87416	0.856960	0.86072
7	adjectives	0.80412	0.819720	0.78644>

- Figure-2: Logistic regression



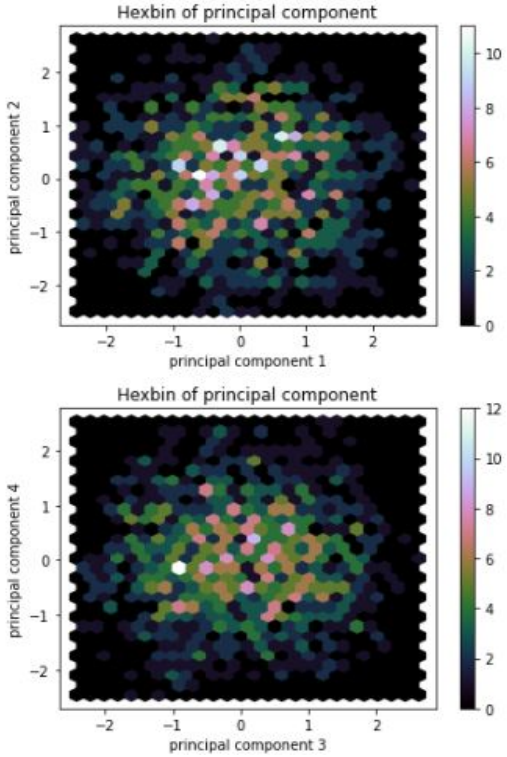
The figure shows us the top 25 best and worst features. Bar shows us the size of each coefficient using grid best estimator. Negative coefficients on left are indicative of negative reviews. While positive coefficients are indicative of positive reviews.

- Figure-3: Scattering of PCA



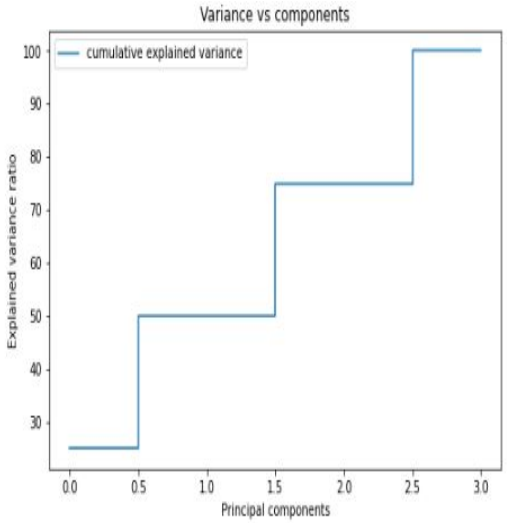
This graph shows the scattering of collection of points of 2 columns taken in pair present in dataframe.

- Figure-4: Hexbin plot of PCA



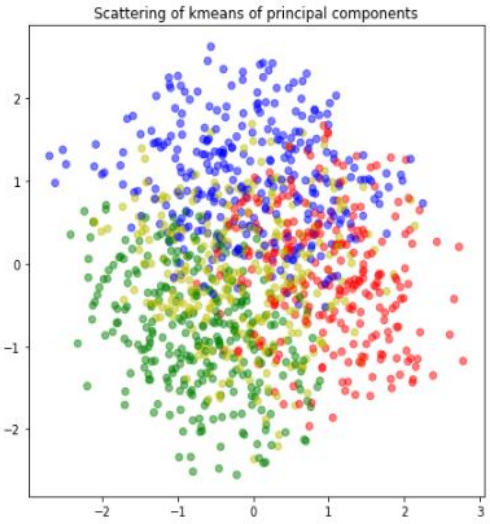
Hexbin plot represent the relationship of 2 numerical variables when you have a lot of data point. Instead of overlapping, the plotting window is split into several hexbins, and the number of points per hexbin is counted. The colour denotes this number of points.

- Figure-5: Cumulative explained variance



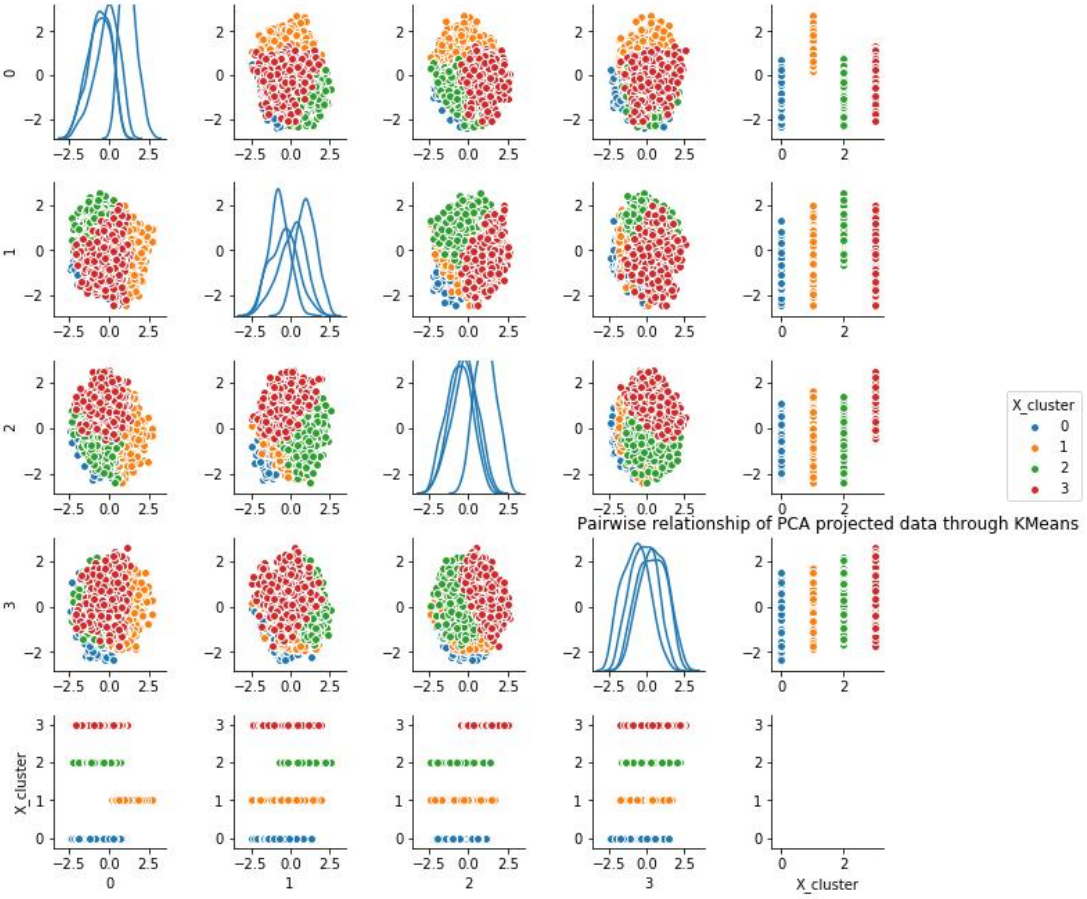
It shows cumulative variance. It is found from individual variance and latter is found from sum of eigenvalues and eigenvalues.

- Figure-6: Scattering of Kmeans of PCA



Shows cluster indices and cluster centre of our PCA data. It also shows the normalized value of all the components.

- Figure-7: Pairwise relationship of PCA projected data through KMeans



Showing pairwise relationship to visualize our Kmeans clustering on PCA projected data(Seaborn pairplot)

- Figure-8: SVM Classifier

	precision	recall	f1-score	support
Positive	0.90	0.22	0.36	5028
Negative	0.55	0.98	0.71	4972
avg / total	0.73	0.60	0.53	10000

	precision	recall	f1-score	support
Positive	0.96	0.01	0.02	5028
Negative	0.50	1.00	0.67	4972
avg / total	0.73	0.50	0.34	10000

The figure shows the classification report of both bag of words(count vectorizer) and tfidf features. In the classification report, it shows precision, support, f1 score, recall of pos and neg.

6 Conclusions

- We can conclude from our base article results that how well different model or classifiers can be applied on the same dataset by calculating accuracy on different features. We also came to know by applying logistic regression on how to train model, fit model with best grid estimator, using cross validation to find accuracy score of model and also predict binary output (1 or 0).

  By applying PCA we learnt how to dimensionally reduce the data from high dimensional data to low dimensional data to make system faster. We also learnt to calculate eigenvalue and eigenvector, mean, variance.

  By applying SVM we learnt how to classify data points or features into positive and negative on basis of polarity 1 and 0 and also on basis of positive and negative tagging. We also learnt to speed up accuracy of system by increasing regularization parameters and selecting particular classifier, random state.

7 Contribution of team members

7.1 Technical contribution of all team members

Tasks	Team member 1	Team member 2	Team member 3
Analysis and Theoretical Support	Aditya	Nilay	
Pseudo Code	Nilay		
Coding and Simulation	Aditya	Dhruvil	Varun
Errors and Queries	Dhruvil	Varun	

7.2 Non-Technical contribution of all team members

Tasks	Team member 1	Team member 2	Team member 3
Documentation	Aditya	Varun	
Integration of modules	Dhruvil	Nilay	

References

[1] M. Hu and B. Liu, "Mining and summarizing customer reviews," *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, 2004.

[2] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp. 347–354, 2005.

[3] K. Dave, S. Lawrence, and D. Pennock, "Mining the peanut gallery: opinion extraction and semantic classification of product reviews," *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pp. 519–528, 2003.

[4] V. Hatzivassiloglou and J. Wiebe, "Effects of adjective orientation and gradability on sentence subjectivity," *COLING '00: Proceedings of the 18th conference on Computational linguistics*, vol. 1, pp. 299–305, 2000.

[5] V. Hatzivassiloglou and K. McKeown, "Predicting the semantic orientation of adjectives," *ACL '98/EACL '98: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 174–181, 1997.

[6] P. Turney and M. Littman, "Unsupervised learning of semantic orientation from a hundred-billion-word corpus," 2002.

[7] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[8] B. Pang and L. Lee, "Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales," *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 115–124, 2005.

[9] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," *Natural Language Processing and Text Mining*, pp. 9–28, 2007.

[10] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences," *EMNLP '03: Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 129–136, 2003.

[11] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, vol. 10, pp. 79–86, 2002.

[12] P. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 417–424, 2002.