

GLA UNIVERSITY



A

Mini Project

Based On Machine Learning

On

Sentimental Analysis

Submitted By:-

Name :- Neelmani Mishra

University Roll No:-191500486

Submitted To:-

Name of Faculty:- Mr.Amir Khan

Technical Trainer

Gla University, Mathura

DECLARATION

I hereby declare that the project entitled –“Sentimental Analysis”, which is being submitted as Mini Project of 6th Semester to partial fulfillment of the requirement for the award of the **Bachelor of Technology** in Computer Science & Engineering and submitted to the Department of Computer Engineering & Application, **GLA UNIVERSITY, MATHURA(UP)** is an authentic record of my genuine work done under the guidance of **Mr. Amir Khan Sir**, Dept. of computer Science & Engineering, Mathura.

The Contest of this project, in full or in parts, have not been submitted to any other institute or University for the award of any degree.

Signature:

Name of the Candidate: Neelmani Mishra

University Roll No. :191500486

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Mr. Amir Khan Sir** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & GLA University, Mathura for their kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Thanking YOU

CERTIFICATE

This is to certify that Mr. Neelmani Mishra of B.Tech computer Science And Technology has successfully completed his project on topic **Sentimental Analysis** as prescribed by Mr. Amir Khan during the academic year 2021-22 as per the guidelines given by Computer Science And Application.

Signature of Mentor:

INDEX

1.Abstract	6
2.Introduction	7
3. Review of Literature	8
4. Objective of the Project	9
5. System Design	10
6. Methodology for Implementation	11
7.Implementation Details	16
8.Result and Sample Output	18
9.Conclusion	22
10.References	23

1.Abstract

Sentiment Analysis also known as Opinion Mining refers to the use of natural language processing, text analysis to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine. In this project, we aim to perform Sentiment Analysis of product based reviews. Data used in this project are online product reviews collected from “Hotel”. We expect to do review-level categorization of review data with promising outcomes.

2.Introduction

Sentiment is an attitude, thought, or judgment prompted by feeling. Sentiment analysis, which is also known as opinion mining, studies people's sentiments towards certain entities. From a user's perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. From a researcher's perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers. However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. The second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral. "It is a quite boring hotel..... but the services were good enough. " The given line is a movie review that states that "it" (the hotel) is quite boring but the services were good. Understanding such sentiments require multiple tasks. Hence, SENTIMENTAL ANALYSIS is a kind of text classification based on Sentimental Orientation (SO) of opinion they contain. Sentiment analysis of product reviews has recently become very popular in text mining and computational linguistics research.

- Firstly, evaluative terms expressing opinions must be extracted from the review.
- Secondly, the SO, or the polarity, of the opinions must be determined.
- Thirdly, the opinion strength, or the intensity, of an opinion should also be determined.
- Finally, the review is classified with respect to sentiment classes, such as Positive and Negative, based on the SO of the opinions it contains.

3.Review of Literature

The most fundamental problem in sentiment analysis is the sentiment polarity categorization, by considering a dataset containing over 5.1 million product reviews from Amazon.com with the products belonging to four categories. A max-entropy POS tagger is used in order to classify the words of the sentence, an additional python program to speed up the process. The negation words like no, not, and more are included in the adverbs whereas Negation of Adjective and Negation of Verb are specially used to identify the phrases. The following are the various classification models which are selected for categorization: Naïve Bayesian, Random Forest, Logistic Regression and Support Vector Machine. For feature selection, Pang and Lee suggested to remove objective sentences by extracting subjective ones. They proposed a text-categorization technique that is able to identify subjective content using minimum cut. Gann et al. selected 6,799 tokens based on Twitter data, where each token is assigned a sentiment score, namely TSI (Total Sentiment Index), featuring itself as a positive token or a negative token. Specifically, a TSI for a certain token is computed as:

$$TSI=(p-(tp/tn)*n)/(p+(tp/tn)*n)$$

where p is the number of times a token appears in positive tweets and n is the number of times a token appears in negative tweets is the ratio of total number of positive tweets over total number of negative tweets.

4.Objective of the Project

- Scrapping product reviews on various websites featuring various products specifically kaggle.com.
- Analyze and categorize review data.
- Analyze sentiment on dataset from document level (review level).
- Categorization or classification of opinion sentiment into-
 - ✚ Positive
 - ✚ Negative

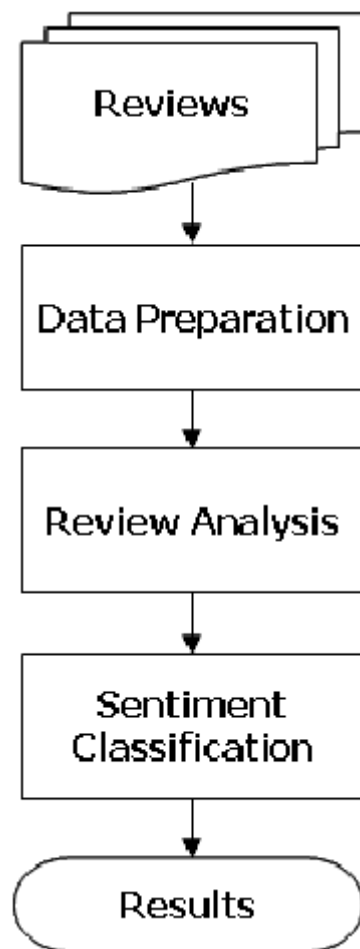


Figure1: A Typical Sentimental Analysis Model

5. System Design

Hardware Requirements:

- Core i5/i7 processor
- At least 8 GB RAM
- At least 60 GB of Usable Hard Disk Space

Software Requirements:

- Windows 10
- Python 3.7
- Python Modules

Data Information:

- The Hotel reviews dataset consists of reviews from hotel. The data span a period of 18 years, including ~35 million reviews up to March 2013. Reviews include product and user information, ratings, and a plaintext review. For more information, please refer to the following paper: J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. RecSys, 2013.
- The Hotel reviews full score dataset is constructed by Xiang Zhang (xiang.zhang@nyu.edu) from the above dataset. It is used as a text classification benchmark in the following paper: Xiang Zhang, Junbo Zhao, Yann LeCun. Character-level Convolutional Networks for Text Classification. Advances in Neural Information Processing Systems 28 (NIPS 2015).
- The Hotel reviews full score dataset is constructed by randomly taking 200,000 samples for each review score from 1 to 5. In total there are 1,000,000 samples.

6.Methodology for Implementation (Formulation/Algorithm)

DATA COLLECTION:

Data which means product reviews collected from kaggle.com from May 1996 to July 2014. Each review includes the following information: 1) reviewer ID; 2) product ID; 3) rating; 4) time of the review; 5) helpfulness; 6) review text. Every rating is based on a 5-star scale, resulting all the ratings to be ranged from 1-star to 5-star with no existence of a half-star or a quarter-star.

SENTIMENT SENTENCE EXTRACTION & POS TAGGING:

Tokenization of reviews after removal of STOP words which mean nothing related to sentiment is the basic requirement for POS tagging. After proper removal of STOP words like “am, is, are, the, but” and so on the remaining sentences are converted in tokens. These tokens take part in POS tagging In natural language processing, part-of-speech (POS) taggers have been developed to classify words based on their parts of speech. For sentiment analysis, a POS tagger is very useful because of the following two reasons: 1) Words like nouns and pronouns usually do not contain any sentiment. It is able to filter out such words with the help of a POS tagger; 2) A POS tagger can also be used to distinguish words that can be used in different parts of speech.

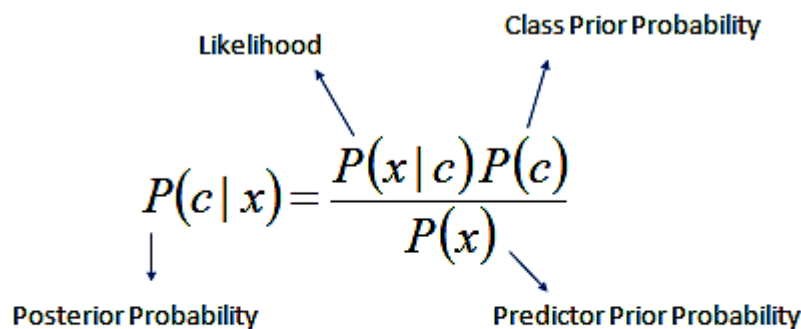
NEGATIVE PHRASE IDENTIFICATION:

Words such as adjectives and verbs are able to convey opposite sentiment with the help of negative prefixes. For instance, consider the following sentence that was found in an hotel parking review: “The Parking area of the hotel was not very revolutionary.” The word, “revolutionary” is a positive word according to the list in. However, the phrase “nothing revolutionary” gives more or less negative feelings. Therefore, it is crucial to identify such phrases. In this work, there are two types of phrases have been identified, namely negation-of-adjective (NOA) and negation-of-verb (NOV).

SENTIMENTAL CLASSIFICATION ALGORITHM:

Naive Bayesian classifier:

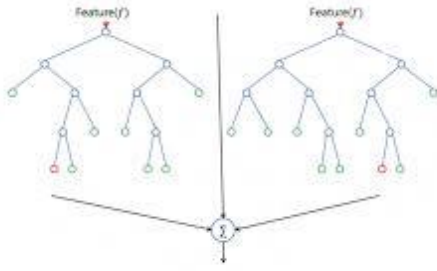
The Naive Bayesian classifier works as follows: Suppose that there exist a set of training data, D , in which each tuple is represented by an n -dimensional feature vector, $X = x_1, x_2, \dots, x_n$, indicating n measurements made on the tuple from n attributes or features. Assume that there are m classes, C_1, C_2, \dots, C_m . Given a tuple X , the classifier will predict that X belongs to C_i if and only if: $P(C_i | X) > P(C_j | X)$, where $i, j \in [1, m]$ and $i \neq j$. $P(C_i | X)$ is computed as

$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$


$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Random forest:

The random forest classifier was chosen due to its 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 samples of D , with each of the samples denoting as D_i . A D_i has the same number of 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. The 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, X , each tree returns its class prediction counting as one vote. The final decision of X ’s class is assigned to the one that has the most votes. The decision tree algorithm implemented in scikit-learn is CART (Classification and Regression Trees). CART uses Gini index for its tree induction. For D , the Gini index is computed as:

$$\begin{aligned} \text{Gini Index} &= 1 - \sum_{i=1}^n (P_i)^2 \\ &= 1 - [(P_+)^2 + (P_-)^2] \end{aligned}$$

Where p_i is the probability that a tuple in D belongs to class C_i . The Gini index measures the impurity of D . The lower the index value is, the better D was partitioned.

Support vector machine:

Support vector machine (SVM) is a method for the classification of both linear and nonlinear data. If the data is linearly separable, the SVM searches for the linear optimal separating hyperplane (the linear kernel), which is a decision boundary that separates data of one class from another. Mathematically, a separating hyper plane can be written as: $W \cdot X + b = 0$, where W is a weight vector and $W = w_1, w_2, \dots, w_n$. X is a training tuple. b is a scalar. In order to optimize the hyperplane, the problem essentially transforms to the minimization of $\|W\|$, which is eventually computed as:

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \phi(x_i) \cdot \phi(x_j)$$

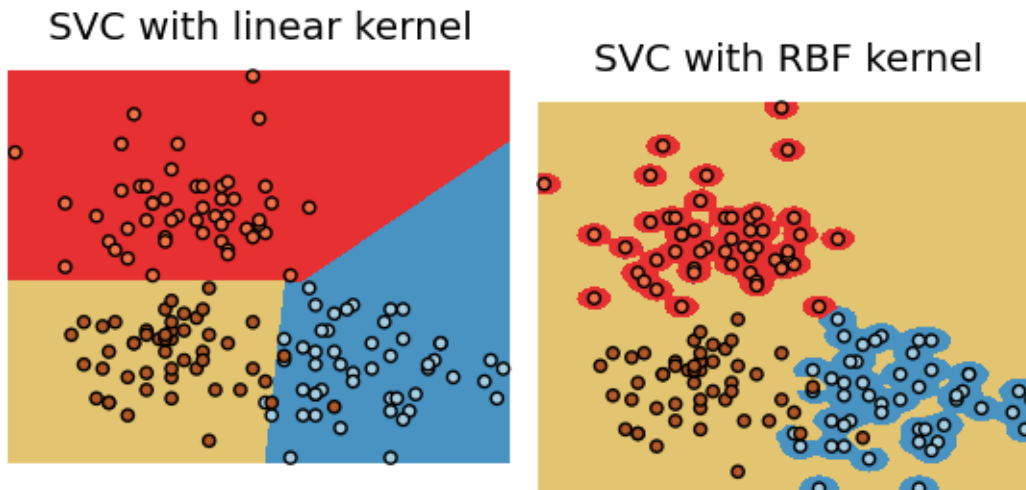
$$\text{where } 0 \leq \alpha_i \leq C \forall i$$

where α_i are numeric parameters, and y_i are labels based on support vectors, X_i .

If the data is linearly inseparable, the SVM uses nonlinear mapping to transform the data into a higher dimension. It then solve the problem by finding a linear hyperplane. Functions to perform such transformations are called kernel functions. The kernel function selected for our experiment is the Gaussian Radial Basis Function (RBF):

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

where X_i are support vectors, X_j are testing tuples, and γ is a free parameter that uses the default value from scikit-learn in our experiment. Figure shows a classification example of SVM based on the linear kernel and the RBF kernel –



Logistic Regression :

Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical). A linear regression is not appropriate for predicting the value of a binary variable for two reasons:

- A linear regression will predict values outside the acceptable range (e.g. predicting probabilities outside the range 0 to 1)
- Since the dichotomous experiments can only have one of two possible values for each experiment, the residuals will not be normally distributed about the predicted line.

On the other hand, a logistic regression produces a logistic curve, which is limited to values between 0 and 1. Logistic regression is similar to a linear regression, but the curve is constructed using the natural logarithm of the “odds” of the target variable, rather than the probability. Moreover, the predictors do not have to be normally distributed or have equal variance in each group.

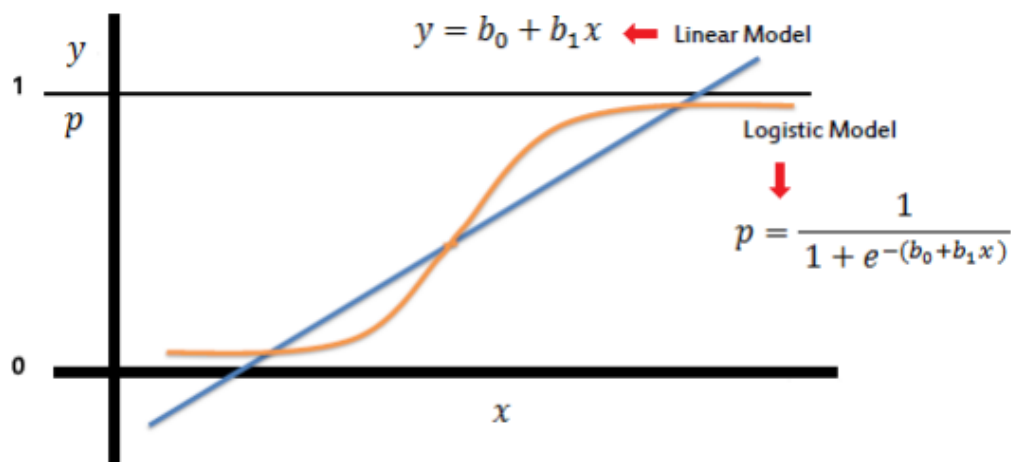
Logistic regression uses maximum likelihood estimation (MLE) to obtain the model coefficients that relate predictors to the target. After this initial function is estimated, the process is repeated until LL (Log Likelihood) does not change significantly.

$$\beta^1 = \beta^0 + [X^T W X]^{-1} \cdot X^T (y - \mu)$$

β is a vector of the logistic regression coefficients.

W is a square matrix of order N with elements $n_i \pi_i (1 - \pi_i)$ on the diagonal and zeros everywhere else.

μ is a vector of length N with elements $\mu_i = n_i \pi_i$.



7. Implementation Details

The training of dataset consists of the following steps:

✚ **Unpacking of data:** The huge dataset of reviews obtained from kaggle.com comes in a .json file format. A small python code has been implemented in order to read the dataset from those files and dump them in to a pickle file for easier and fast access and object serialization. Hence initial fetching of data is done in this section using Python File Handlers.

✚ **Preparing Data for Sentiment Analysis:**

- 1) The pickle file is hence loaded in this step and the data besides the one used for sentiment analysis is removed. As shown in our sample dataset in Page 11, there are a lot of columns in the data out of which only rating and text review is what we require. So, the column, “review Summary” is dropped from the data file.
- 2) After that, the review ratings which are 3 out of 5 are removed as they signify neutral review, and all we are concerned of is positive and negative reviews.
- 3) The entire task of preprocessing the review data is handled by this utility class-“NltkPreprocessor”.

✚ **Preprocessing Data:** This is a vital part of training the dataset. Here Words present in the file are accessed both as a solo word and also as pair of words. Because, for example the word “bad” means negative but when someone writes “not bad” it refers to as positive. In such cases considering single word for training data will work otherwise. So words in pairs are checked to find the occurrence to modifiers before any adjective which if present which might provide a different meaning to the outlook.

✚ **Training Data/ Evaluation:** The main chunk of code that does the whole evaluation of sentimental analysis based on the preprocessed data is a part of this. The following are the steps followed:

- 1) The Accuracy, Precision, Recall, and Evaluation time is calculated and displayed.
- 2) Navie Bayes, Logistic Regression, Linear SVM and Random forest classifiers are applied on the dataset for evaluation of sentiments.
- 3) Prediction of test data is done and Confusion Matrix of prediction is displayed.

- 4) Total positive and negative reviews are counted.
- 5) A review like sentence is taken as input on the console and if positive the console gives 1 as output and 0 for negative input.

8. Results and Sample Output

The ultimate outcome of this Training of Public reviews dataset is that, the machine is capable of judging whether an entered sentence bears positive response or negative response.

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while **Recall** (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Both precision and recall are therefore based on an understanding and measure of relevance.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

F1 score (also **F-score** or **F-measure**) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

In statistics, a receiver operating characteristic curve, i.e. ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The Total Operating Characteristic (TOC) expands on the idea of ROC by showing the total information in the two-by-two contingency table for each threshold. ROC gives only two bits of relative information for each threshold, thus the TOC gives strictly more information than the ROC.

When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative'). This can be seen as follows: the area under the curve is given by (the integral boundaries are reversed as large T has a lower value on the x-axis).

$$A = \int_{-\infty}^{\infty} \text{TPR}(T) \text{FPR}'(T) dT = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(T' > T) f_1(T') f_0(T) dT' dT = P(X_1 > X_0)$$

The machine evaluates the accuracy of training the data along with precision Recall and F1

The Confusion matrix of evaluation is calculated.

It is thus capable of judging an externally written review as positive or negative.

A positive review will be marked as [1], and a negative review will be hence marked as [0].

Results obtained using Hold-out Strategy(Train-Test split) [values rounded upto 2 decimal places].

Name of classifier	F ₁	Accuracy	Precision	Recall	ROC AUC
Multinomial NB	85.25%	85.31%	85.56%	84.95%	85.31%
Logistic Regression	88.12%	88.05%	87.54%	88.72%	88.05%
Linear SVC	88.12%	88.11%	87.59%	88.80%	88.11%
Random Forest	82.43%	81.82%	79.74%	85.30%	81.83%

The Confusion Matrix Format is as follows:



The Confusion Matrix of Each Classifier are as follows:

68556	11470
12032	67942

Classifier 1: Multinomial NB

69963	10063
8955	17019

Classifier 3: Liner SVC

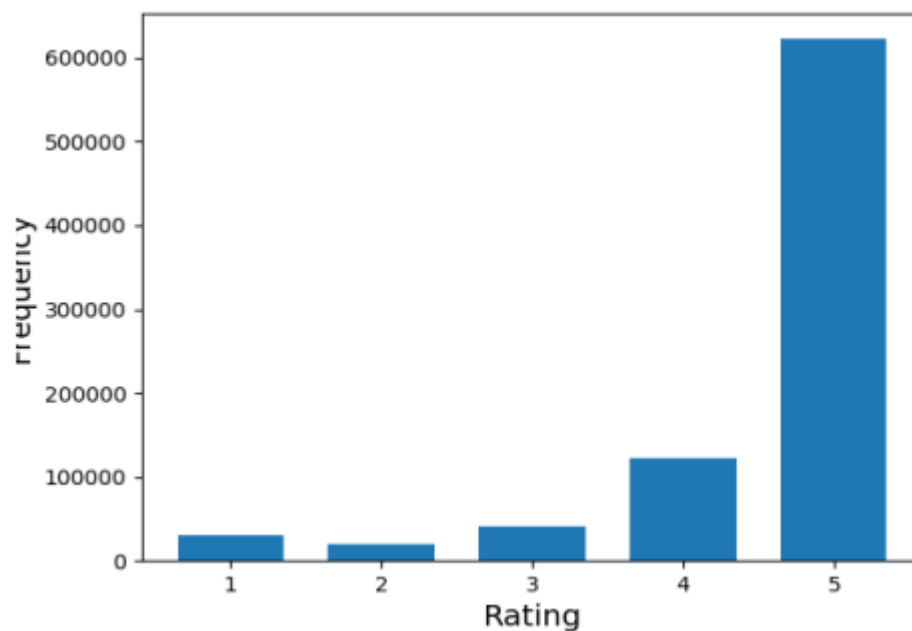
69928	10098
9023	70951

Classifier 2: Logistic Regression

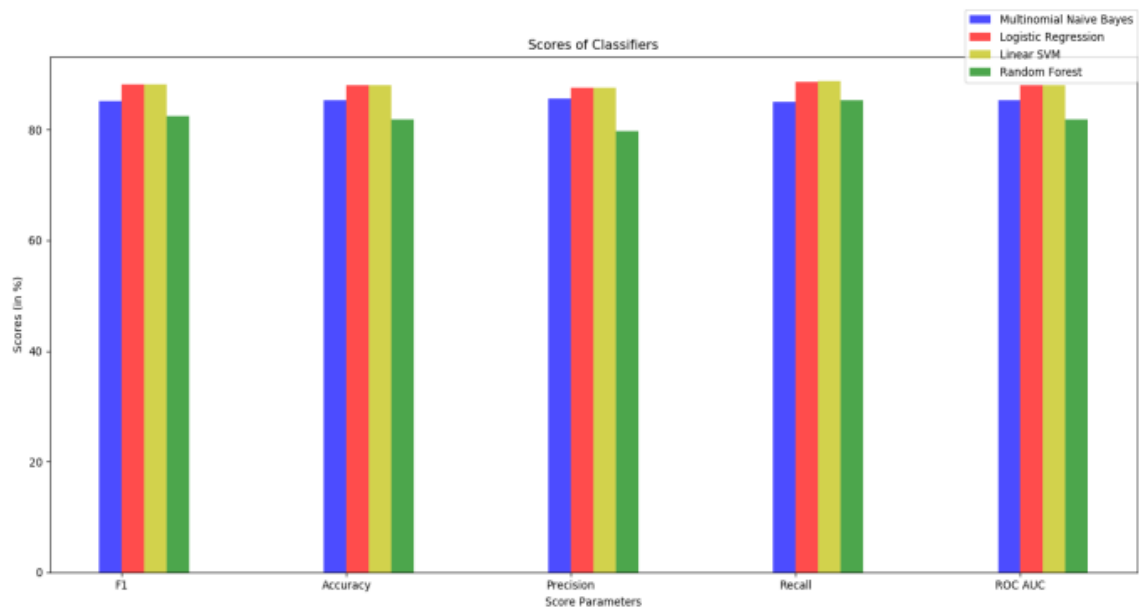
62695	17331
11749	68225

Classifier 4: Random Forest

The Bar Graph showing the Frequency of Ratings in the dataset



This Bar graph shows the score of each classifier after successful training. The parameters be: F1 Score, Accuracy, Precision, Recall and Roc-Auc.



9. Conclusion

Sentiment analysis deals with the classification of texts based on the sentiments they contain. This article focuses on a typical sentiment analysis model consisting of three core steps, namely data preparation, review analysis and sentiment classification, and describes representative techniques involved in those steps.

Sentiment analysis is an emerging research area in text mining and computational linguistics, and has attracted considerable research attention in the past few years. Future research shall explore sophisticated methods for opinion and product feature extraction, as well as new classification models that can address the ordered labels property in rating inference. Applications that utilize results from sentiment analysis is also expected to emerge in the near future.

References

- S. ChandraKala¹ and C. Sindhu², "OPINION MINING AND SENTIMENT CLASSIFICATION: A SURVEY," Vol .3(1), Oct 2012, 420-427.
- G.Angulakshmi , Dr.R.ManickaChezian , "An Analysis on Opinion Mining: Techniques and Tools". Vol 3(7), 2014 www.iarcce.com.
- Callen Rain, "Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning" Swarthmore College, Department of Computer Science.
- Padmani P .Tribhuvan, S.G. Bhirud, Amrapali P. Tribhuvan, " A Peer Review of Feature Based Opinion Mining and Summarization"(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1), 2014, 247-250 www.ijcsit.com.
- Carenini, G., Ng, R. and Zwart, E. Extracting Knowledge from Evaluative Text. Proceedings of the Third International Conference on Knowledge Capture (K-CAP'05), 2005.
- Dave, D., Lawrence, A., and Pennock, D. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. Proceedings of International World Wide Web Conference (WWW'03), 2003.
- Zhu, Jingbo, et al. "Aspect-based opinion polling from customer reviews." IEEE Transactions on Affective Computing, Volume 2.1, pp.37-49, 2011.
- Na, Jin-Cheon, Haiyang Sui, Christopher Khoo, Syin Chan, and Yunyun Zhou. "Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews." Advances in Knowledge Organization Volume 9, pp. 49-54, 2004.
- Nasukawa, Tetsuya, and Jeonghee Yi. "Sentiment analysis: Capturing favorability using natural language processing." In Proceedings of the 2nd international conference on Knowledge capture, ACM, pp. 70-77, 2003.
- Li, Shoushan, Zhongqing Wang, Sophia Yat Mei Lee, and Chu-Ren Huang. "Sentiment Classification with Polarity Shifting Detection." In Asian Language Processing (IALP), 2013 International Conference on, pp. 129-132. IEEE, 2013.