Classification of Sentiment Based on Movie Feedback Given By Audiences

Sumedh Shah, Alwin Anuse, Rupali Kute

***Abstract:** The need for generating automated sentiment on audience feedbacks has been the need of the hour. Manually going through the entire movie feedback becomes tedious therefore an attempt to predict the polarity of a movie based on the reviews using machine learning models is done. Usage of the IMDB movie reviews dataset has been done for training and testing. In this study we also try to depict the real-life problems of class imbalance and train-test splits, hence obtaining solutions for the same. The problem of class imbalance in today's world has affected a large amount of predictive applications such as cancer detection , fraudulent transactions in banks etc, hence this study is an attempt to perform a solution to solve the class imbalance problem. Use of the undersampling method has been done in this study to improve the accuracy of an imbalanced class. Feature extraction methods such as Bag of Words and Term Frequency Inverse document Frequency have been used to generate features from the reviews. The Logistic regression and SVM classifiers have been used in the study to measure the accuracy. Along with the accuracy the Confusion Matrix has also been calculated to showcase the class imbalance taking its effect on the accuracy.*

***Keywords:** Bag of Words(BOW),Class Imbalance, Term Frequency Inverse Document Frequency(TF-IDF),Support Vector Machine(SVM)*

I. INTRODUCTION

Sentiment Analysis or Opinion Mining is the study of extraction, quantification and identification of subjective information using natural language processing, text analysis, computational linguistics and biometrics. Sentiment analysis can be used on a variety of data and for a variety of applications, its study has been one of the top research topics in recent times. Nowadays everyone expresses their opinions on various products, movies and other items using social media platforms such as twitter, Facebook, Instagram etc. One has to go through all the reviews for a particular item which is a laborious task and requires a large amount of time. Hence the role of Machine Learning in the present age is of utmost importance as it helps humans to create intelligent systems which predict the polarity of a text in a matter of seconds. This advancement in the field of artificial intelligence has helped us solve many real like problems in a time efficient manner. In this paper, reviews from the IMDB dataset are analyzed and the machine learning models classify the reviews into positive or negative. Implementation of various tasks were done using Python programming language and the results of the same are presented in the form of accuracies for various experiments. Here a real world problem of class imbalance is also addressed. This is a scenario where the number of training data belonging to one class is significantly lower than those belonging to the other class. This complication is common in scenarios where anomaly detection is decisive like cancer detection, fraudulent transactions in banks, identification of uncommon diseases and more. In such a situation the traditional machine learning models produce flawed decisions which may be hazardous. The data we gather today is generally supervised and hence we have made use of a supervised dataset to create a real-life situation. Using the different movie reviews, we have attempted to create the problems and apply feasible solutions to the created problems in our dataset. The two problems we have recognized in this paper are:

a- Train-test split accuracy
b- Class Imbalance

By playing around with our dataset we have approached these problems using simple solutions and checked the respective accuracies.

The following figure can be used as a reference for giving a better understanding of the process:

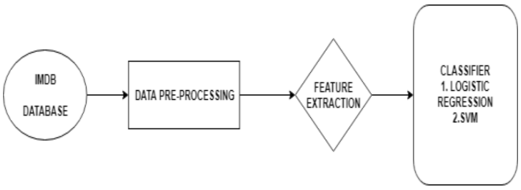


Fig1: Sentiment Analysis block diagram

II. LITERATURE SURVEY

The sentiment analysis has been an area of growing interest around the world and many published works have given a range of techniques compatible with sentiment analysis. Sentiment analysis has been applied at document level classification [1-4], it has been applied at sentence level [5]

Manuscript published on November 30, 2019.
* Correspondence Author

Sumedh Shah*, Electronics and Telecommunication, Maharashtra Institute of Technology ,Pune,India. Email: sumedhshah97@gmail.com

Alwin Anuse, Electronics and Telecommunication, Maharashtra Institute of Technology ,Pune,India. Email: alwin.anuse@mitpune.edu.in

Rupali Kute, Electronics and Telecommunication, Maharashtra Institute of Technology ,Pune,India. Email: rupali.kute@mitpune.edu.in

Retrieval Number: D7237118419/2019©BEIESP
DOI:10.35940/ijrte.D7237.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

and lately feature level sentiment analysis is being carried out [6-7]. The difference levels of sentiment analysis along with their tasks have been discussed by Seema et al in their study [8]. Pang et al used movie reviews for classification using traditional machine learning algorithms such as SVM, Logistic Regression and Naïve Bayes. In this study they have concluded that these algorithms perform better than human produced baselines and that SVM and logistic regression perform better than Naïve Bayes. They use n-gram techniques for feature selection i.e.unigram, bi-gram and combination of them. Furthermore, they concluded that in their experimentation unigram technique produced better results than the others [9]. Annett et al. proposed a novel approach on Support Vector Machines by investigating variations of feature vectors involving different sizes of feature vectors, different feature representations, and different feature types. They performed the lexical approach and machine learning approach for sentiment classification and realized that the machine learning approach was more successful with better accuracy [10]. Andrew L. Maas et al. introduced the IMDB dataset of 50,000 reviews for sentiment classification, their model is a mix of unsupervised and supervised techniques to learn word vectors capturing, semantic document information as well as rich sentiment content. Their method performed better than Linear Discriminant Analysis.[11]. Justin Martineau and Tim Finin proposed a novel approach for sentiment analysis using Delta TFIDF instead of the normal TFIDF approach .In this approach heir term frequency transformation boosts the significance of words that are irregularly distributed between the positive and negative classes and by overlooking regularly occurring words. The value of irregularly occurring feature is zero.More irregular the occurrence,more important a feature should be. This method gives us a better idea of the feature’s importance in the document for sentiment classification. They used SVM with a linear kernel for classification and concluded that delta TFIDF outperformed flat term frequencies and TFIDF weights [12].Chakraborty et al used the deep learning approach on the IMDB dataset provided by Kaggle, they made the use of Google’s word2vec algorithm for application on the large dataset for classification so that the semantic information is observed. Using word2vec model word embeddings were obtained to give high accuracy [13]. A Doc2vec model can also be used to enhance this model. Sahu et al have followed a lexical approach using SentiWordNet [15] to determine the overall polarity of the movie review. They have used the movie dataset available on rotten tomatoes comprising of 8000 polar movie reviews. Usage of different classification techniques such as Random Forest, KNN, COCR, Bagging and Naïve Bayes is done and different performance parameters are found out. In their study, the Random Forest technique obtained the highest accuracy of 95%.[14]. Manek et al have proposed the use of a Gini Index [17] based feature selection method with Support Vector Machine (SVM) classifier for sentiment classification of a large movie review dataset. They obtained an impressive accuracy of 94.46% with the weight by Gini Index method on the IMDB dataset of 50,000 reviews [Maas et al]. Their proposed framework has improved accuracy than the other methods studied [16]. Parkhe et al proposed an aspect-based sentiment analysis of movie reviews [18]. An Aspect Based Text Separator was used to separate text into aspects and different aspects used were screenplay, music, acting, plot, movie and direction. The aspect specific lexicon was used to separate the reviews aspect wise and each word in the lexicon was associated with the Part of Speech of that word. An aspect classifier such as Naïve Bayes was then used to output a 1 or -1 for positive and negative reviews respectively [19].To conclude, a number of studies have been performed in this area and different methods have been proposed. This study is different than the others as it is an attempt to better the accuracy when it comes to real life problems such as class imbalance in the area of sentiment analysis.

III. DATASET DESCRIPTION

A readily available dataset of IMDB is been used in this study [11].This dataset was first used in the study by Maas et al [11] .The dataset can be accessed at [20].

We have made use of 50,000 movie reviews; it consists of equal positive and negative reviews (25,000 each). We used a csv file and loaded it into python, after which we have trained the supervised learning algorithms using different train-test splits.

Using this dataset, we have mimicked real life situations such as overfitting and class imbalance, and we have measured the accuracy of the models after applying feasible solutions for these problems.

The dataset also consists of 50,000 unlabeled reviews.

A. Dataset Visualization and Description

In this subsection, the description of the dataset used is given, the various parameters and the factors affecting the classification are found using the python environment and results are displayed. We have displayed the parameters such as count, standard deviation, different percentiles,mean, min and max.

Table1: Dataset description

Parameters	Value
Count of non-NA values	50000.000000
Mean	0.500000
Standard Deviation	0.500005
Minimum Value	0.000000
25%	0.000000
50%	0.500000
75%	1.000000
Maximum Value	1.000000

To do some visualization of our textual data usage of python wordcloudlibrary is done. The wordcloud is an informative choice for visualization, it is a visual representation of text data. It displays a list of words, the importance of each being shown with font size or colour. This format is useful for quickly perceiving the most prominent terms.

The figure2 below gives us the wordcloud of the negative terms which stand out in the negative reviews and their ranking varies with size of the word in the wordcloud.

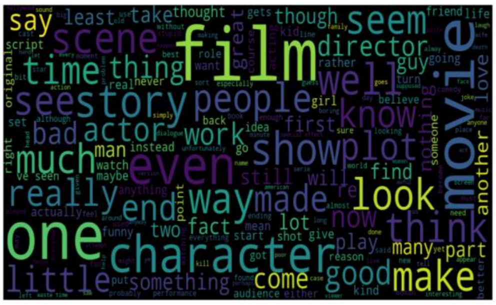


Fig2: Wordcloud of Negative Terms

Figure3gives us the wordcloud of the positive terms which stand out in the positive reviews and their ranking varies with size of the word in the worldcloud.

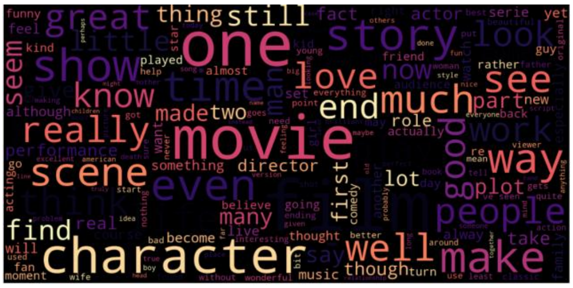


Fig3: Wordcloud of Positive

IV. PROPOSED FRAMEWORK

In this section the framework of the process is being described.

The figure below is the flowchart representation of the framework and the order of the proposed study.

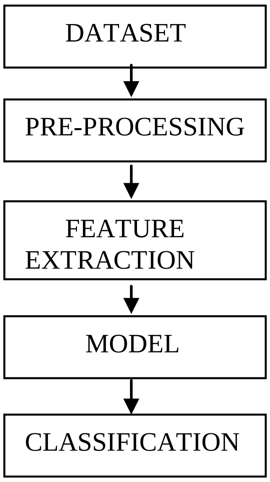


Fig4: Flowchart of framework

V. PRE PROCESSING STAGE

A preprocessing function for cleaning of the reviews from the dataset was defined.
This function removes URL, remove HTML tags, handles negation words which are split into two parts, convert the words to lower cases, remove all non-letter characters. These elements are very common and they do not provide enough semantic information for the task hence we need to perform cleaning. Preprocessing is the first step to increasing accuracy of a classifier and it need not be a very complex task. Simple cleaning will provide you with good results and this has been proved in this study.

VI. FEATURE EXTRACTION

The feature extraction section has made use of the two of the most common and effective approaches i.e. Bag of Words (BOW) and TF-IDF (Term Frequency Inverse Document Frequency). Through our results we have obtained the most effective n-gram value for our dataset and this gives us highest accuracy with both our models. In the experimentation section we will showcase these values.

A. Bag Of Words

According to Chaffar et.al. each sentence in the given dataset is represented by a feature vector composed of Boolean values for each word that occurs in a sentence. If a word is present then it's corresponding value in the matrix is set to 1 else it is set to 0.BOW takes into account words as autonomous parts hence it does not succeed in catching the semantic information from the text. However, it performs generally very well in text classification [21]. To make it simply let us consider a sentence *"It was the best of times"* and we check the frequency of words from the 10 unique words randomly selected other than the words in the sentence, for example:worst, age, wisdom and foolishness.
All these add up to 10 different words so BOW will consider each as a separate document and create a vector as given below.
"it" = 1
"was" = 1
"the" = 1
"best" = 1
"of" = 1
"times" = 1
"worst" = 0
"age" = 0
"wisdom" = 0
"foolishness" = 0
VECTOR- *"It was the best of times" = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0].*
In this method, each word or token is called a gram. A combination of two-word pairs is called a bigram model. In this way we can adjust the parameters the way we feel fit. Only disadvantage is an accuracy and computation time trade off, which means that as we increase the n-gram range the accuracy will increase but it will be computationally more expensive than a unigram or bigram model. The concept of vectorization in Machine Learning is used here where text is converted into numbers. We have used two approaches to build a BOW model using the Machine Learning libraries available in Python:

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

1. Count Vectorizer

CountVectorizer works on Terms Frequency, i.e. counting the occurrences of tokens and building a sparse matrix of documents x tokens.

The reason behind of using this approach is that keyword or important signal will occur again and again. So, if the number of occurrences represent the importance of word. More frequency means more importance.

2. TfidfVectorizer

This is the same vectorization that takes place in the TFIDF approach but here we specify the parameters as we did in CountVectorizer which we do not use in the next feature extraction method. It basically gives us the normalized count occurrence that is if you think that extremely high frequency may dominate the result and causing model bias. Normalization can be applied to pipeline easily.

B. Term Frequency Inverse Document Frequency(TF-IDF)

According to Li Ping et al [22] the values of the vector elements Wi for a document d are calculated as a combination of the statistics $Tf(t,d)$ and $Df(t)$. The term frequency $Tf(t,d)$TF (t, d is the number of times word t occurs in document d. The document frequency$Df(t)$ is the number of documents in which the word t occurs at least once. The inverse document frequency $IDF(t)$IDF(t) can be calculated from the document frequency. $|D|$ is the total number of documents. The inverse document frequency of a word is low if it occurs in many documents and is highest if the word occurs in only one. The value W of features T for document d is then calculated as the product$\boldsymbol{W} = \boldsymbol{TF}(\boldsymbol{t},\boldsymbol{d}) \cdot \boldsymbol{IDF}(\boldsymbol{t})$

VII. CLASSIFICATION MODELS

A. Logistic Regression

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by evaluating probabilities using a sigmoid activation function.

In this study binary logistic regression is used for classification of the reviews, being a very strong model, it provides us with good accuracy with minimum training time. With the help of the Scikit-learn package available in python the implementation of logistic regression is done.

B. Support Vector Machine(SVM)

SVM's are universal learners and can outperform Naïve Bayes in text categorization, they have proved to be highly effective for classification applications [23]. For implementing linear SVM too we have made the use of python's Scikit-learn package. Both logistic regression and SVM gave us higher accuracy than other models and the results will be displayed in the experimentation.

VIII. EXPERIMENTATION WITH TABLES AND FIGURES

The dataset consists of 50,000 reviews which are equally divided into negative and positive (25,000 each), the preprocessing of the reviews was done in order to make training easier and increase testing accuracy. After pre-processing the clean data was used for feature extraction in order to train the classifiers, the common methods such as Bag of Words and the TF-IDF vectorizer was used. Classification accuracy was higher than expected for these traditional methods and that was because of the systematic cleaning of data along with the change in parameters for feature extraction. In the proposed framework we will discuss the various parameters used and how the accuracy increased on varying them.

For addressing the imbalance situation usage of the confusion matrix has been done to compare the accuracies of different cases related to class imbalance. A class imbalance was created by reducing the number of positive reviews for training and keeping the negative reviews the same, the training of our models was done using this imbalance of reviews and the testing was done on a set with equal polarity of negative and positive reviews. It was found that the model was inaccurately classifying the negative reviews and with this we created an imbalanced situation. In today's world there are a lot of situations related to class imbalance and one of the main areas is for cancer detection using AI. We find a majority of negative cases and comparatively fewer positive ones so the model is trained on imbalanced data and this leads to inaccurate results which is deleterious for the outcome of the patients in case they are wrongly diagnosed.

Our experimentation section is divided into two parts:

A. Train-Test combinations

The main aim of this paper is to identify various real-life situations and form a solution for the same. In Machine Learning problems the splitting of train and test is of utmost importance, accuracy and other performance parameters vary according to the split taken. Nowadays more and more data are coming in which needs to be tested but the training is at times stagnant, in this approach we have calculated accuracy for different cases with more testing and less training examples.

Changing of the parameters can be done to get different accuracies on the test set. So far, we have implemented:

1. Min= 0, max= 2, ngram range= 2,2
2. Min=0, max= 5, ngram range= 1,4(long computation time)
3. **Min =0, max= 4, ngram range= 1,4 (long computation time)**

In the tables below we have listed the maximum accuracy obtained using the highlighted parameters.

Case1

Total Number of Reviews – 50,000
Train –Test split: 50%-50%

Table2: Accuracy for 50-50 split

Model	TF-IDF	BOW TF-IDF	BOW COUNT
1.Logistic Regression	89.15%	83.35%	85.18%
2.SVM	89.45%	85.75%	84.38%

Figure5 gives us the visual representation of the accuracy using columnar chart for 50-50 train-test split.

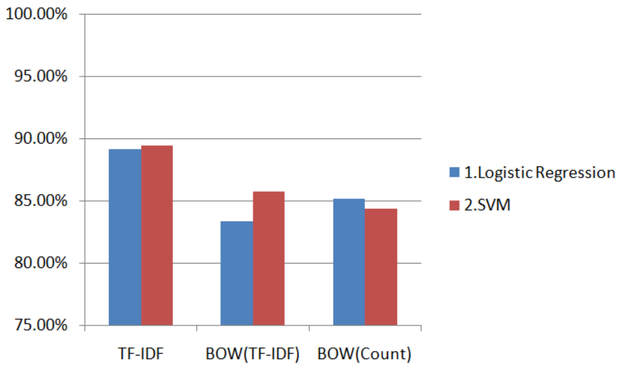


Fig5: 50-50 split

Case2

Total Number of Reviews- 50,000
Train-Test split: 40%-60%

Table3: Accuracy for 40-60 split

Model	TF-IDF	BOW TF-IDF	BOW COUNT
1.Logistic Regression	88.82%	85.41%	84.66%
2.SVM	89.24%	85.45%	83.80%

Figure6 gives us the visual representation of the accuracy using columnar chart for 40-60 train-test split.

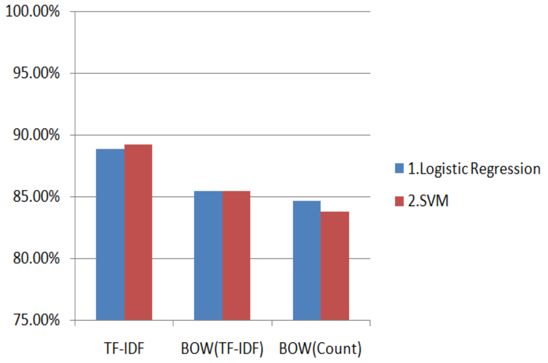


Fig6: 40-60 split

Case3

Total Number of Reviews- 50,000
Train-Test split: 20%-80%

Table4: Accuracy for 20-80 split

Model	TF-IDF	BOW TF-IDF	BOW COUNT
1.Logistic Regression	87.42%	80.72%	83.58%
2.SVM	87.98%	83.69%	82.80%

Figure7 gives us the visual representation of the accuracy using columnar chart for 20-80 train-test split.

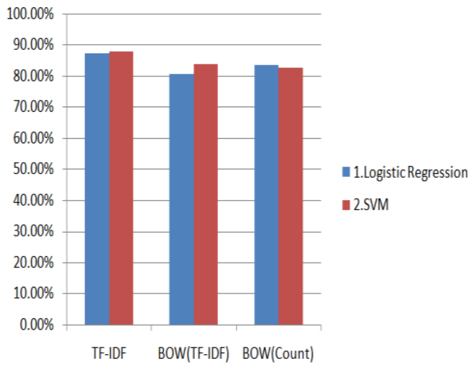


Fig7: 20-80 split

Case4

Total Number of Reviews- 50,000
Train-Test split: 10%-90%

Table5: Accuracy for 10-90 split

Model	TF-IDF	BOW TF-IDF	BOW COUNT
1.Logistic Regression	85.76%	75.68%	81.49%
2.SVM	86.74%	81.38%	80.61%

Figure8 gives us the visual representation of the accuracy using columnar chart for 10-90 train-test split.

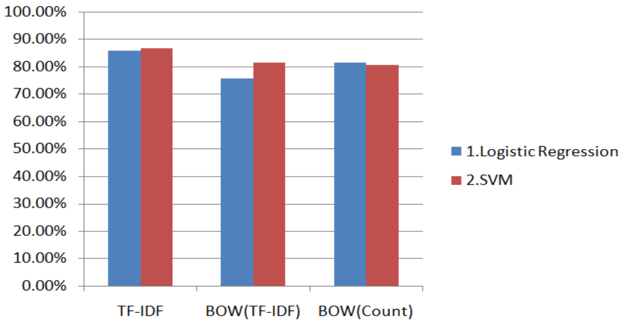


Fig8: 10-90 split

Observations – The 50%-50% and 40%-60% train-test split provided us with the best accuracy going as high as 89.45%. BOW Parameters- Min_df=0, Max_df=4, Ngram range = 1,4 gave us best accuracy for bag of words hence it was presented in the tables.

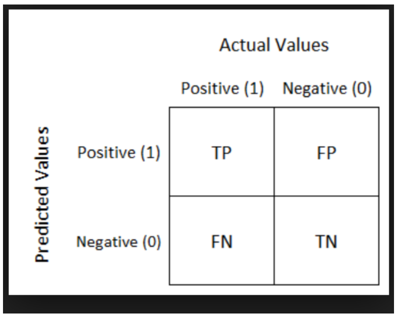
Only accuracy of SVM with countvectorizer decreases rest everything increases. The n-gram range can be increased but this increases number of features and hence the computation time. So, a tradeoff is present between accuracy and computational cost.

B. Class Imbalance

To create a class imbalance in our training set we decreased the number of positive samples and kept the negative samples constant at 20,000 reviews. Four different cases were implemented and for each case calculation of the accuracy and confusion matrix was done which gives us an indication of the imbalanced state. At the end a solution for the imbalanced situation was implemented using the under-sampling method. The solution provided gives us a substantial increase in accuracy for all the feature extraction methods. This solution can also be implemented in real life applications where class imbalance occurs. The testing set consists of 10,000 reviews with equal negative and positive samples. The different cases with their accuracies have been shown below

Confusion Matrix:

Figure 9 below is the basic structure of a confusion matrix. This make it easier to analyze the performance of the model on the dataset.



	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Fig9: Confusion Matrix

Case1

Negative reviews- 20,000
Positive Reviews- 10,000

Table6: Case1 accuracy for Class Imbalance

Model	TF-IDF	BOW TF-IDF	BOW COUNT
1.Logistic Regression	83.87%	50.01%	50.36%
2.SVM	82.80%	50.41%	50.53%

Confusion Matrix tables for case 1:

1. TF-IDF method

Table7: Confusion Matrix values for TF-IDF (case1)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	4784	3603	1397	216
2.SVM	4695	3585	1415	305

2. BOW with TF-IDF vectorizer

Table8: Confusion Matrix values for BOW with TF-IDF vectorizer(case1)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	5000	1	4999	0
2.SVM	4998	43	4957	2

3. BOW with Count vectorizer

Table9: Confusion Matrix values for BOW with Count vectorizer(case1)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	4999	37	4963	1
2.SVM	4996	57	4943	4

Case2

Negative reviews- 20,000
Positive Reviews- 7500

Table10: Case2 accuracy for Class Imbalance

Model	TF-IDF	BOW TF-IDF	BOW COUNT
1.Logistic Regression	80.34%	50.00%	50.25%
2.SVM	81.01%	50.27%	50.28%

Confusion Matrix tables for case 2:

1. TF-IDF method

Table11: Confusion Matrix values for TF-IDF (case2)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	4876	3158	1842	124
2.SVM	4754	3347	1653	246

2. BOW with TF-IDF vectorizer

Table12: Confusion Matrix values for BOW with TF-IDF vectorizer(case2)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	5000	0	5000	0
2.SVM	4999	28	4972	1

3. BOW with Count vectorizer

Table13: Confusion Matrix values for BOW with Count vectorizer(case2)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	4999	26	4974	1
2.SVM	4999	29	4971	1

Case3

Negative reviews- 20,000
Positive Reviews- 5000

Table14: Case3 accuracy for Class Imbalance

Model	TF-IDF	BOW TF-IDF	BOW COUNT
1.Logistic Regression	73.68%	50.00%	50.17%
2.SVM	77.95%	50.20%	50.18%

Confusion Matrix tables for case 3:

1. TF-IDF method

Table15: Confusion Matrix values for TF-IDF (case3)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	4926	2442	2558	74
2.SVM	4823	2972	2028	177

2. BOW with TF-IDF vectorizer

Table16: Confusion Matrix values for BOW with TF-IDF vectorizer(case3)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	5000	0	5000	0
2.SVM	5000	20	4980	0

3. BOW with Count vectorizer

Table17: *Confusion Matrix values for BOW with Count vectorizer(case3)*

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	4999	18	4982	1
2.SVM	4999	19	4981	1

Case4

Negative reviews- 20,000
Positive Reviews- 1000

Table18: Case4 accuracy for Class Imbalance

Model	TF-IDF	BOW TF-IDF	BOW COUNT
1.Logistic Regression	50.75%	50.00%	50.02%
2.SVM	59.45%	50.02%	50.01%

Confusion Matrix tables for case 4:

1. TF-IDF method

Table19: Confusion Matrix values for TF-IDF (case4)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	4999	76	4924	1
2.SVM	4992	953	4047	8

2. BOW with TF-IDF vectorizer

Table20: Confusion Matrix values for BOW with TF-IDF vectorizer(case4)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	5000	0	5000	0
2.SVM	5000	2	4998	0

3. BOW with Count Vectorizer

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table21: Confusion Matrix values for BOW with Count vectorizer(case3)

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	5000	2	4998	0
2.SVM	4999	2	4998	1

Observations- The true negative value of all the class imbalance cases decreases as shown in the confusion matrix result tables. This takes place due to the imbalance of positive and negative reviews.

C. Class Imbalance Solution to increase accuracy

As shown above the accuracy significantly decreases if the polarity of reviews is imbalanced. Hence in order to gain some accuracy we under sampled the negative reviews to bring them close to the positive ones. We took the first case of 10,000 positive reviews, under sampled the negative reviews to 10,000 and tested it on the same test set.

Positive reviews- 10,000
Negative reviews- 10,000

Table22: Class Imbalance solution accuracy

Model	TF-IDF	BOW TF-IDF	BOW COUNT
1.Logistic Regression	86.93%	83.04%	79.29%
2.SVM	87.02%	82.24%	69.11%

Confusion Matrix tables:

1. TF-IDF method

Table23: Confusion Matrix for Class Imbalance solution with TF-IDF

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	4214	4479	521	786
2.SVM	4301	4401	599	699

2. BOW with TF-IDF vectorizer

Table24:Confusion Matrix for Class Imbalance solution with BOW using TF-IDF vectorizer

Model	True Positiv e	True Negative	False Negative	False Positive
1.Logistic Regression	4101	4203	797	899
2.SVM	3998	4226	774	1002

3. BOW with Count vectorizer

Table25: Confusion Matrix for Class Imbalance solution with BOW using Count vectorizer

Model	True Positive	True Negative	False Negative	False Positive
1.Logistic Regression	3784	4145	855	1216
2.SVM	2023	4888	112	2977

Observations – The solution of undersampling the negative reviews is a simple but effective one as it increases the accuracy by a substantial margin. This solution can be applied in certain real life instances too where class imbalance dominates.

IX. CONCLUSION AND FUTURE SCOPE

This study makes an attempt to classify the IMDB movie reviews dataset using logistic regression and SVM. Usage of different feature extraction methods have been done; it is observed that usage of TF-IDF approach generally gives a higher accuracy. The accuracy for Bag of Words approach also increases with increase in the n-gram range but the computation time for extracting features increases drastically .In the train-test split scenario we obtained highest average accuracy for the 50-50 split. In the class imbalance situation, the under sampling of negative reviews proved to be quite useful in increasing accuracy. Hence proving to be a simple but effective solution to tackle the Class Imbalance problem.
This study on sentiment analysis also has its limitations:

1. It is observed that in many reviews usage of symbols makes a big difference in obtaining the sentiment (☺,☹). However, our models are not yet trained to make use of these.
2. Reviews may not always be in the correct grammatical form e.g. Fine may be spelled as fineeee , great maybe spelled as greattt however the weightage of these words are not taken in their exact account.
3. Computation time for extracting features makes the analysis a slow process, this happens with most feature extracting methods.

Further work that can be examined in this area of sentiment analysis of movie reviews. In our class imbalance case if we can provide extra weight to the features extracted from the positive reviews than those extracted from the negative ones, we can hope to improve accuracy further. The ratio of weights given to the positive and negative features should be the same as the ratio of the positive-negative review split for the imbalanced case. Hybrid techniques for feature extraction must also be studied to further improve the accuracy.

Overall this study has taken into account different real life constraints faced for a large amount of predictive applications and an attempt to tackle those using effective solutions has been done. Along with these constraints this study has also showcased the importance of analyzing audience feedbacks in today's day and age. It has umpteen advantages and if used correctly it can save everyone a lot of trouble and time.

REFERENCES

1. B. Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity analysis using subjectivity summarization based on minimum cuts. ACL
2. Peter D. Turney, (2002), Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp. 417-424
3. Yan Zhao, Suyu Dong and Leixiao Li, (2014), Sentiment Analysis on News Comments Based on Supervised Learning Method, International Journal of Multimedia and Ubiquitous Engineering, Vol.9, No.7 pp.333-346
4. Richa Sharma, Shweta Nigam and Rekha Jain, (2014), Opinion Mining of Movie Reviews at Document Level, International Journal on Information Theory (IJIT), Vol.3, No.3
5. S M Kim and E Hovy. 2004. Determining the sentiment of opinions. Coling.
6. [6]Padmapani P. Tribhuvan, S.G. Bhirud, Amrapali P. Tribhuvan, (2014), A Peer Review of Feature Based Opinion Mining and Summarization, International Journal of Computer Science and Information Technologies, Vol. 5 (1), ISSN: 0975-9646
7. S. ChandraKala and C. Sindhu, (2012), Opinion Mining and Sentiment Classification: A Survey, ICTACT Journal on Soft Computing, Vol- 03, ISSUE: 01, ISSN: 2229-6956
8. Seema Kolkur, Gayatri Dantal and Reena Mahe (2015), Study of Different Levels for Sentiment Analysis, International Journal of CurrentEngineeringand Technology, Vol.5, No.2, E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
9. Pang, B., Lee, L., &Vaithyanathan, S. (2002). Thumbs up? sentiment classificationusing machine learning techniques. In Proceedings of the ACL-02conference on Empirical methods in natural language processing-Volume 10 (pp. 79–86). Association for Computational Linguistics.
10. Annett, M., &Kondrak, G. (n.d.). A Comparison of Sentiment Analysis Techniques: Polarizing Movie Blogs. Lecture Notes in Computer Science
11. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts, Stanford University (Stanford, CA 94305) HLT'11 proceedings of the 49th Annual Meeting of the association for Computational Linguistics: Human Language Technologies- Volume 1.
12. Justin Martineau, and Tim Finin, Delta TFIDF: An Improved Feature Space for Sentiment Analysis, Proceedings of the Third International ICWSM Conference (2009).
13. Chakraborty, K., Bhattacharyya, S., Bag, R., &Hassanien, A. E. (2018). Comparative Sentiment Analysison a Set of Movie Reviews Using Deep Learning Approach. Advances in Intelligent Systems and Computing, 311–318.
14. Sahu, T. P., & Ahuja, S. (2016). Sentiment analysis of movie reviews: A study on feature selection & classification algorithms. 2016 International Conference on Microelectronics, Computing and Communications (MicroCom).
15. Baccianella, Stefano, Andrea Esuli, and FabrizioSebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." LREC. Vol. 10. 2010.
16. Manek, A. S., Shenoy, P. D., Mohan, M. C., & R, V. K. (2016). Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier.
17. Raileanu, L. E., & Stoffel, K. (2004). Theoretical Comparison between the Gini Index and Information Gain Criteria. Annals of Mathematics and Artificial Intelligence
18. Parkhe V, Biswas B (2014) Aspect based sentiment analysis of movie reviews: finding the polarity directing aspects. In: Proceedings of international conference on soft computing and machine intelligence 2014
19. Parkhe, V., & Biswas, B. (2015). Sentiment analysis of movie reviews: finding most important movie aspects using driving factors. Soft Computing, 20(9), 3373–3379.
20. http://ai.Stanford.edu/amaas/data/sentiment/index.html
21. Chaffar, S., &Inkpen, D. (2011). Using aHeterogeneous Dataset for Emotion Analysis in Text. Lecture Notes in Computer Science, *62–67*.doi:10.1007/978-3-642-21043-3_8.
22. Li-Ping Jing, Hou-Kuan Huang, & Hong-Bo Shi. (n.d.). Improved feature selection approach TFIDF in text mining. Proceedings. International Conference on Machine Learning and Cybernetics. doi:10.1109/icmlc.2002.1174522
23. Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the ECML'98, 1998, 137–142.

AUTHORS PROFILE

Sumedh Shah has completed his B.E in Electronics and Telecommunication from Maharashtra Institute of Technology,Pune in affiliation with Savitribai Phule Pune University.He has obtained a distinction in his engineering and has also represented the college team for Badminton. Currently working as a Data Engineer at Modak Analytics in Hyderabad. His research areas are Machine Learning,Database Management,Optimization Models , Image Recognition etc. His next aim is to complete a Master's and expand his knowledge in the field of Analytics .

AlwinAnuse received the B.E , M.E and Ph. D degrees from Pune University. He also did Post graduate Diploma in Indian Film Studies from SPPU . He is currently working as Associate Professor in DrVishwanathKarad MIT-World Peace University . He has got his PH.D degree from College of Engineering Pune. His research topic was ,"A Framework for Classification under Geometric Invariance in Images'.He also has research publications in Springer and Elsevier journals.Research areas also include signal processing,artificial intelligence and film studies.

RupaliKute received the Ph.D. degree from the Savitribai Phule Pune University of India at College of Engineering Pune in 2019. Her research interests include face recognition, fingerprint recognition, biometric association, and machine learning. She is a recipient of the Gold Medal Award at Savitribai Phule Pune University,2011 in M.E.(Electronics-Digital Systems). She has done her research in person identification using biometric association under heterogeneous environment. In biometric association different parts of the face are associated with the full face using transfer learning technique. Her research interests include face recognition, fingerprint recognition, biometric association, and machine learning.