INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 09 ISSUE: 02 | FEB - 2025 SJIF RATING: 8.448 ISSN: 2582-3930

A Data Science Based Approach for Sentiment Classification of Social Media Data

Jyoti Bamblani[1], Prof. Pooja Hardiya[2]

***Abstract*— Sentiment classification is a crucial task in natural language processing (NLP) and data science that involves determining the sentiment or emotion conveyed in a piece of text. It has widespread applications in social media analysis, customer feedback evaluation, and financial forecasting. In this approach, the customer review dataset from Amazon reviews has been analyzed. Pre-processing of raw data has been done prior to using it to train a neural network. The regularization based Bayes Optimized Deep Neural Network has been used for sentiment classification from social media dataset. To compare the performance of the proposed system against existing research in the domain, the predication error metric has been computed. From the performance of the proposed system, it can be observed that the proposed system clearly outperforms the existing approaches in terms of prediction error and accuracy.**

Keywords—Natural Language Processing (NLP), Sentiment Analysis, Deep Neural Networks, Regulrization, Prediction Error.

I. INTRODUCTION

In the digital age, customer reviews play a crucial role in shaping consumer decisions and business strategies. Analyzing these reviews can provide valuable insights into customer satisfaction, product performance, and areas for improvement [1] However, manually analyzing thousands of reviews is time-consuming and inefficient. Machine learning (ML)-based sentiment analysis automates this process, enabling businesses to classify customer reviews as positive, negative, or neutral [2]. By leveraging ML algorithms, companies can enhance their decision-making, improve customer service, and optimize product offerings. The knowledge discovery in databases (KDD) model has been used extensively for sentiment classification which is depicted in figure 1.

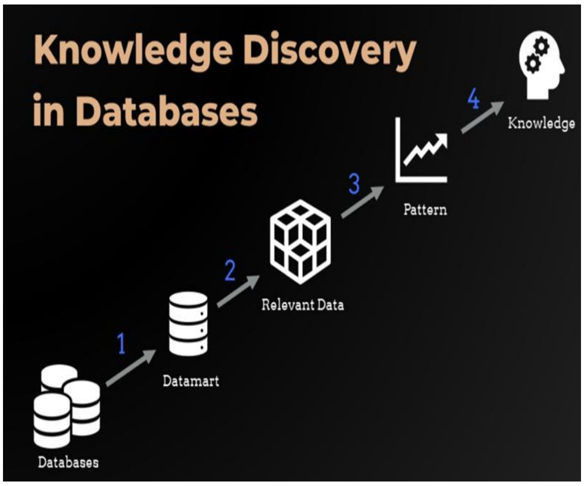

Fig.1 The knowledge discovery process

Sentiment analysis, also known as opinion mining, is a natural language processing (NLP) technique used to determine the sentiment expressed in a piece of text [3]. It involves identifying emotions, attitudes, and opinions in customer reviews and classifying them into predefined sentiment categories. Traditional sentiment analysis relied on rule-based approaches and lexicon-based methods, which involved predefined word lists and sentiment scores [4]. However, these methods lacked adaptability to complex language structures and contextual variations. Machine learning-based approaches have emerged as a more effective alternative, providing higher accuracy and better generalization [6].

Despite its advantages, sentiment analysis faces challenges such as [6]:

1. Sarcasm and Contextual Understanding: Machine learning models struggle to detect sarcasm, irony, and implicit meanings in reviews.
2. Data Imbalance: Sentiment datasets often contain an uneven distribution of positive, negative, and neutral reviews, affecting model performance.
3. Evolving Language: Slang, emojis, and evolving customer expressions require continuous updates in sentiment analysis models.

II. CONTEXTUAL ANALYSIS AND DEEP LEARNING

One of the major challenges in sentiment analysis is the contextual analysis of data. The different aspects are discussed subsequently [7].

2.1 Contextual Analysis

It is often difficult to estimate the context in which the statements are made. Words in textual data such as tweets can be used in different contexts leading to completely divergent meaning [8].

2.2 Frequency Analysis

Often words in textual data (for example tweets) are repeated such as

##I feel so so so happy today!!

In this case, the repetition of the word is used to emphasize upon the importance of the word. In other words, it increases to its weight. However, such rules are not explicit and do not follow any regular mathematical formulation because of which it is often difficult to get to the actuality of the tweet [9].

2.3 Converting textual data into numerically weighted data

The biggest challenge in using an ANN based classifier is the fact that the any ANN structure with a training algorithm doesn't work upon textual data directly to find some pattern. It needs to be fed with numerical substitutes [7]. Hence it becomes mandatory to replace the textual information with numerical information so as to facilitate the learning process of the neural network [10]

the machine or artificial intelligence system requires training for the given categories [11]. Subsequently, the neural network model needs to act as an effective classifier. The major challenges here the fact that sentiment relevant data vary significantly in their parameter values due to the fact that the parameters for each building is different and hence it becomes extremely difficult for the designed neural network to find a relation among such highly fluctuating parameters. Generally, the Artificial Neural Networks model's accuracy depends on the training phase to solve new problems, since the Artificial Neural Networks is an information processing paradigm that learns from its environment to adjust its weights through an iterative process [12].

Deep learning models do have the capability to extract meaning form large and verbose datasets by finding patterns between the inputs and targets. Since neural nets directly process numeric data sets, the processing of data is done prior to training a neural network [13]. The texts are first split into training and testing data samples in the ratio of 70:30 for training and testing. Further, a data vector containing known and commonly repeated spam and ham words is prepared [4]. Text normalization is followed by removal of special characters and punctuation marks.

Subsequently the data set structuring and preparation is performed based on the feature selection. The deep learning structure is depicted in figure 2 [15].

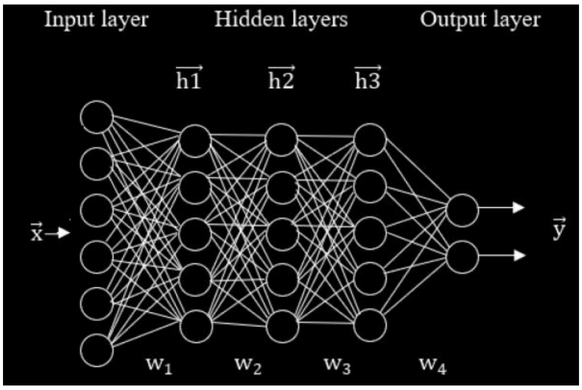

Fig.2 The deep learning structure

The deep learning structure is depicted in figure 2 and it is basically a cascade of stacked neural networks [14]. Multiple hidden layers facilitate the analysis of complex data. The cascading weight updating can be understood as [15]:

$$a^n = \varphi_n(\varphi_{n-1} \dots \dots \varphi_1\{wp + b\}) \qquad (1)$$

Here,
W is the weight
b is the bias
a is the input to the final nth layer
ϕ is the activation function

III. METHODOLOGY

The proposed approach is mathematically modelled as:

The prepared data vector for training is used for training wherein the weights are initialized randomly. A stepwise implementation is done as [16]:

1. Prepare two arrays, one is input and hidden unit and the second is output unit.

Here, a two dimensional array W_{ij} is used as the weigt updating vector and output is a one dimensional array Y_i.

3. Original weights are random values put inside the arrays after that the output [17].

$$x_j = \sum_{i=0} y_i W_{ij} \qquad (2)$$

Where,
y_i is the activity level of the j^th^ unit in the previous layer and
W_{ij} is the weightof the connection between the i^{th} and the j^{th} unit.
4. Next, activation is invoked by the sigmoid function applied to the total weighted input [18].

$$y_i = \left[\frac{e^x - e^{-x}}{e^x + e^{-x}}\right] \quad (3)$$

Summing all the output units have been determined, the network calculates the error (E).

$$E = \frac{1}{2}\sum_i (y_i - d_i)^2 \quad (4)$$

Where, y_i is the event level of the j^{th} unit in the top layer and d_i is the preferred output of the j_i unit [19].

A. Implementing Back Prop:

Calculation of error for the back propagation algorithm is as follows:
Error Derivative (EA_j) is the modification among the real and desired target:

$$EA_j = \frac{\partial E}{\partial y_j} = y_j - d_j \quad (5)$$

Here,
E represents the error
y represents the Target vector
d represents the predicted output

Error Variations is total input received by an output changed given by:

$$EI_j = \frac{\partial E}{\partial X_j} = \frac{\partial E}{\partial y_j} X \frac{dy_j}{dx_j} = EA_j y_j (1 - y_i) \quad (6)$$

Here,
E is the error vector
X is the input vector for training the neural network
In Error Fluctuations calculation connection into output unit is computed as [20]:

$$EW_{ij} = \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial X_j} = \frac{\partial X_j}{\partial W_{ij}} = EI_j y_i \quad (7)$$

Here,
W represents the weights
I represents the Identity matrix
I and j represent the two dimensional weight vector indices
Overall Influence of the error:

$$EA_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} X \frac{\partial x_j}{\partial y_i} = \sum_j EI_j W_{ij} \quad (8)$$

The partial derivative of the Error with respect to the weight represents the error swing for the system while training. The gradient is computed as [21]:

$$g = \frac{\partial e}{\partial w} \quad (9)$$

Here,
g represents the gradient
e represents the error of each iteration
w represents the weights.

The gradient is considered as the objective function to be reduced in each iteration. A probabilistic classification using the Bayes theorem of conditional probability is given by [22]:

$$P\left(\frac{H}{X}\right) = \frac{P\left(\frac{X}{H}\right)P(H)}{P(X)} \quad (10)$$

Here,
Posterior Probability [P (H/X)] is the probability of occurrence of event H when X has already occurred
Prior Probability [P (H)] is the individual probability of event H
X is termed as the tuple and H is is termed as the hypothesis.
Here, [P (H/X)] denotes the probability of occurrence of event X when H has already occurred. The proposed algorithm for the approach is presented next:

Proposed Algorithm:

As the customer review texts may have overlapping tags or tokens, hence a probabilistic Bayes Classifier has been proposed. As sentiments do not possess a particular decision boundary (fixed), hence a probabilistic approach happens to be more effective which can be done employing the Deep Bayes Net whose classification depends on the following relation [23]:

$$P\left(\frac{X}{X_i, k_1, k_2, M}\right) = \frac{P\left(\frac{X_i}{X, k_2, M}\right) P\left(\frac{X_i}{k_1, M}\right)}{P\left(\frac{X}{k_1, k_2, M}\right)} \quad (12)$$

Here,
P represents probability.
X_i represents weights and bias vectors (combined).
X represents the data to be used for the purpose of training.
M represents data units (neurons) in network.
k_1 *and* $k_2$represents the term responsible for penalty based regularization [24].
$\rho = \frac{k_1}{k_2}$ is often considered the regularization factor which is acted upon the objective function (J) to me optimized based on the training dataset, and renders the regularized cost function [25]:

$$F(w) = \mu w^T w + v[\frac{1}{n}\sum_{i=1}^{n}(p_i - a_i)^2] \quad (13)$$

If $(\pi \ll v)$: errors in training are typically rendered low.

else if $(\pi \geq v)$: errors are typically rendered high needing a weight reduction or Penalty. The proposed algorithm is presented next:

Start
{
Step.1 *Obtain annotated dataset.*

Step.2 *Divide the data into a ratio of 70:30 as training and testing data samples.*

Step.3 *Define match token data length (n) and*
$for\ i = 1{:}n$
$Search\ (token == match\ text)$
end

Step.4 *Design a neural network with multiple hidden layers.*

Step.5 *Initialize training with random weights.*

Step.6 *Train models with training data and updated weights based on the back propagation rule as:*

$$w_{k+1} = w_k - \left[J_k J_k^T + \mu I\right]^{-1} J_k^T e_k \quad (14)$$

Step.7 *if (Cost Function J stabilizes over multiple iterations)*
Truncate
else if (iterations==max. iterations defined)
Truncate
else
{
Apply data and update (w, b)
Feedback (e)
}

Step.8 *Calculate error% and Classification Accuracy*
Stop
}

The performance parameters used for evaluation of the algorithm is the accuracy % which is computed as:

$$Accuracy\% = 100 - error\% \quad (15)$$

IV. RESULTS

The experiment has been run on MATLAB with the deep learning library (toolbox). The Amazon customer review dataset has been obtained from Kaggle.

The proposed system utilizes the textual data in the form of tweets to be analyzed based on positive, negative and neutral tokens to be represented by -1, 0 and 1 respectively. Subsequently, the number of tokens with polarity is also fed to the neural network as a training parameter. The customer reviews are also ranked from 1 to 5 depending upon the review. Each of the processes is presented next:

Fig.3 Sentiment Data

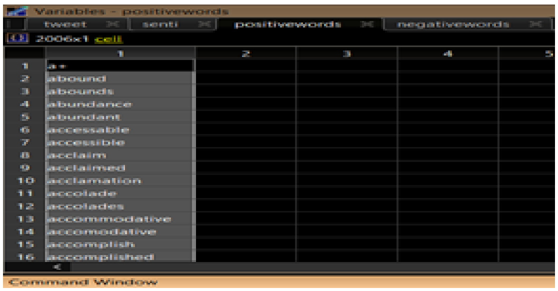

Fig.4 Positive Tokens

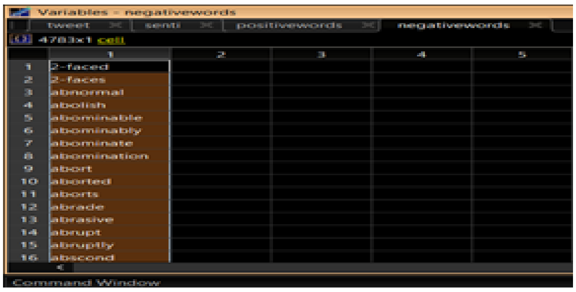

Fig.5 Negative Tokens

Figure 4 and 5 depict the positive and negative tokens to train the Bayesian Model presented next.

Fig.6 Design of the Deep Neural Network

Fig.7 GUI for classification (happy)

Fig.8 GUI for classification (sad)

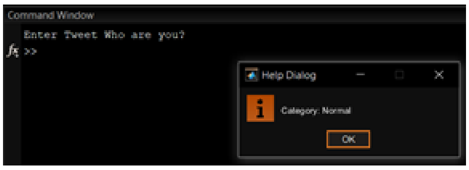

Fig.9 GUI for classification (neutral/normal)

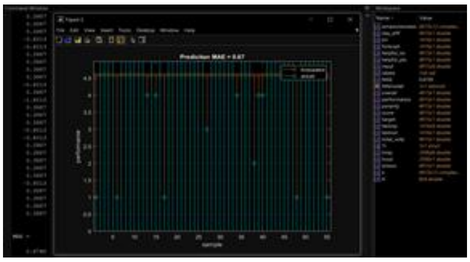

Fig.10 Obtained MAE for Model

The proposed system parameters can be summarized in table 1.

Table 1. Summary of Results

Parameter	Value
ML category	Bayesian Net
No. of hidden layers	5
MAE	0.67
Accuracy (Proposed Work)	99.3% (APPROX)
Accuracy (Previous Work, [1])	93.5%

CONCLUSION

It can be concluded from previous discussions that sentiment classification is a very important application of NLP and data science. This paper presents a probabilistic Deep Bayes Net with regularization for sentiment classification, analyzing social medi data for product reviews. A Bayesian Network (BayesNet) is a directed acyclic graph (DAG) where nodes represent variables (such as words, phrases, or sentiment labels), and edges define probabilistic dependencies between them. In sentiment analysis, a BayesNet can model the probabilistic relationships between words in a tweet, post, or comment and their associated sentiment. Unlike traditional classifiers, BayesNet does not solely rely on word frequencies; instead, it captures contextual dependencies and incorporates prior knowledge into the classification process. Bayesian Networks with regularization provide a robust framework for sentiment classification of social media data by capturing probabilistic relationships between words and sentiments while mitigating overfitting. Regularization techniques such as Laplace smoothing, L1/L2 penalties, and Bayesian priors enhance the model's generalization ability, making it well-suited for noisy, informal social media text. The prediction results clearly indicate the improved performance of the proposed approach in comparison with existing research in the domain.

REFERENCES

[1] Y Zhao, M Mamat, A Aysa, K Ubul, "Multimodal sentiment system and method based on CRNN-SVM", Neural Computing and Applications, Springer, 2023, pp.1-13.

[2] M Dhyani, GS Kushwaha, S Kumar, "A novel intuitionistic fuzzy inference system for sentiment analysis", International Journal of Information Technology, Springer 2022, vol.14., pp. 3193–3200.

[3] A Vohra, R Garg, "Deep learning based sentiment analysis of public perception of working from home through tweets", Journal of Intelligent Information Systems, Springer 2022, vol.60, pp. 255–274.

[4] H. T. Phan, N. T. Nguyen and D. Hwang, "Aspect-Level Sentiment Analysis Using CNN Over BERT-GCN," in IEEE Access, 2022, vol. 10, pp. 110402-110409.

[5] R. Obiedat R. Qaddoura, A. Al-Zoubi, L. Al-Qaisi, O. Harfoushi, M. Alrefai, H. Faris., "Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary

SVM-Based Approach in an Imbalanced Data Distribution," in IEEE Access, vol. 10, pp. 22260-22273, 2022.

[6] S Vashishtha, S Susan, "Neuro-fuzzy network incorporating multiple lexicons for social sentiment analysis", Applications in computing, Springer 2022, vol.26, pp. 487–4507.

[7] A. Saha, A. A. Marouf and R. Hossain, "Sentiment Analysis from Depression-Related User-Generated Contents from Social Media," 2021 8th Intern, "ational Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia, 2021, pp. 259-264

[8] MLB Estrada, RZ Cabada, RO Bustillos, "Opinion mining and emotion recognition applied to learning environments", Journal of Expert Systems, Elsevier 2020, vol. 150., 113265

[9] A. M. Rahat, A. Kahir and A. K. M. Masum, "Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset," 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), Moradabad, India, 2019, pp. 266-270.

[10] H. Hasanli and S. Rustamov, "Sentiment Analysis of Azerbaijani twits Using Logistic Regression, Naive Bayes and SVM," 2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 2019, pp. 1-7.

[11] R. B. Shamantha, S. M. Shetty and P. Rai, "Sentiment Analysis Using Machine Learning Classifiers: Evaluation of Performance," 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 2019, pp. 21-25.

[12] M. Yasen and S. Tedmori, "Movies Reviews Sentiment Analysis and Classification," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 2019, pp. 860-865.

[13] P. Karthika, R. Murugeswari and R. Manoranjithem, "Sentiment Analysis of Social Media Network Using Random Forest Algorithm," 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 2019, pp. 1-5.

[14] L Zheng, H Wang, S Gao," Sentimental feature selection for sentiment analysis of Chinese online reviews", International journal of machine learning and cybernetics, Springer, 2018, vol.9, pp. 75–84.

[15] R. D. Desai, "Sentiment Analysis of Twitter Data," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 114-117.

[16] A. Bayhaqy, S. Sfenrianto, K. Nainggolan and E. R. Kaburuan, "Sentiment Analysis about E-Commerce from Tweets Using Decision Tree, K-Nearest Neighbor, and Naïve Bayes," 2018 International Conference on Orange Technologies (ICOT), Nusa Dua, Bali, Indonesia, 2018, pp. 1-6.

[17] L. Wang, M. Han, X. Li, N. Zhang and H. Cheng, "Review of Classification Methods on Unbalanced Data Sets," in IEEE Access, vol. 9, pp. 64606-64628, 2021.

[18] G. Karatas, O. Demir and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," in IEEE Access, 2020, vol. 8, pp. 32150-32162.

[19] D. Dablain, B. Krawczyk and N. V. Chawla, "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data," in IEEE Transactions on Neural Networks and Learning Systems, 2023, vol. 34, no. 9, pp. 6390-6404.

[20] M. I. Zul, F. Yulia and D. Nurmalasari, "Social Media Sentiment Analysis Using K-Means and Naïve Bayes Algorithm," 2018 2nd International Conference on Electrical Engineering and Informatics (ICon EEI), Batam, Indonesia, 2018, pp. 24-29.

[21] M. I. Zul, F. Yulia and D. Nurmalasari, "Social Media Sentiment Analysis Using K-Means and Naïve Bayes Algorithm," 2018 2nd International Conference on Electrical Engineering and Informatics (ICon EEI), Batam, Indonesia, 2018, pp. 24-29.

[22] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine and M. Bennamoun, "Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users," in IEEE Computational Intelligence Magazine, 2022, vol. 17, no. 2, pp. 29-48.

[23] A. A. Abdullah, M. M. Hassan and Y. T. Mustafa, "A Review on Bayesian Deep Learning in Healthcare: Applications and Challenges," in IEEE Access 2022,, vol. 10, pp. 36538-36562

[24] C. Xia, D. H. K. Tsang and V. K. N. Lau, "Structured Bayesian Compression for Deep Neural Networks Based on the Turbo-VBI Approach," in IEEE Transactions on Signal Processing, vol. 71, pp. 670-685, 2023.

[25] R. J. J. Boussi Fila, S. H. Attri and V. Sharma, "Mitigating Overfitting in Deep Learning: Insights from Bayesian Regularization," 2024 IEEE Region 10 Symposium (TENSYMP), New Delhi, India, 2024, pp. 1-6.