

## **Machine Learning Approach on Opinion Mining About Reviews of Products in Online Shopping**

### **1.M. Susmitha**

Assistant Professor, susmitha\_m@vnrvjiet.in  
Department of Information Technology,  
VNRVJIET, Bachupally, Hyderabad, Telangana

### **2. Renuka Kondabala**

Assistant Professor, renuka\_k@vnrvjiet.in  
Department of Information Technology  
VNRVJIET, , Bachupally, Hyderabad, Telangana

### **3. Y. Bhaskar Reddy**

Assistant Professor, bhaskarreddy\_y@vnrvjiet.in  
Department of Information Technology,  
VNRVJIET, Bachupally, Hyderabad, Telangana

### **4. M.Radha**

Assistant Professor, radha\_m@vnrvjiet.in  
Department of Information Technology,  
VNRVJIET, Bachupally, Hyderabad, Telangana

---

**Abstract:** Feedbacks are an important part of industrial growth but when there are hundreds and thousands of feedbacks, it is hard to get into each and every one of them. It is important to classify them as positive, negative and other adjectives for an easy understanding of what the majority of people are saying. This can be easily achieved through Sentiment Analysis. Sentiment analysis and Opinion mining is a field of study through which we can dissect individuals' feelings, opinions, from composed language. It uses natural language processing, and supervised machine learning algorithms.

**Keywords:** opinion mining, sentiment analysis, natural language processing, supervised machine learning algorithms.

---

## **1 Introduction**

In this age of modernization and improved lifestyle, people are getting habituated to shop online. The online shopping have become so feasible for any kind and for anything to be shopped. The platforms which provide goods for shopping online are called E-Commerce platforms. There are many platforms coming these days with various features, filters in-built comforting the customers. One of the main point for the customers to purchase a particular product is to check through

the reviews given by the previous customers who purchased that product. Checking all these reviews and confirming the products' state is a tedious job to do. Instead, the platforms which get updated classifying the reviews and so called the opinions of the customers into various categories helps the product features to expose out and it helps the upcoming customers to figure out the products' value.

Opinion Mining on customer reviews is so essential in the case of drawing insights of a particular product basing upon various parameters. The main concept of opinion mining is that it deals with the emotions and opinions. Here, in this case the review given by the customer for a particular product on any E-commerce website is considered to be his opinion. Dealing with these opinions can be confirmed as understanding the customers their perspectives. The dealing of the opinions given in the form of reviews can be achieved using Opinion Mining as it is a part of Machine Learning. These reviews are basically computed by using Natural Language Processing technique and are classified into positive, neutral and negative categories.

## **2 Related work**

Sentiment Classification on Twitter and Zomato Dataset Using Supervised Learning Algorithms: The authors have had explained and made clear about the polarity detection and the class labels[1].

Supervised learning based approach to aspect based sentiment analysis: The approach through which the classification algorithms can be implemented on the training dataset[2]. Sentiment analysis techniques in recent works: The establishment of class label using user-defined functions and training various models[3]. Predicting supervised machine learning performances for sentiment analysis using contextual-based approaches: Performance analysis of the classifiers and the approaches and methodologies to improve and understand the accuracy[4]. Twitter sentiment analysis using various classification algorithms: Understanding the emotions and natural language processing techniques and implementation on the text reviews [5]. Different methods of supervised are used and compared on real covid analysis using basic algorithms of Machine learning, taken the data from twitter[6]. Utilization of deep learning and semantic analysis for opinion mining in information extraction gives all ideas of opinion mining survey both in machine and deep learning approaches[7]. Detecting stress by the opinions given on social networks can be analyzed by the data engineering and statistic analysis made on data[8]. CNN, Deep learning can be used for sentiment prediction and also facial expression detection. By the images of different expressions are used and CNN is used to detect the emotion [9]. Parsing of telugu using machine learning techniques for

statistical analysis of statements, can also be extended for semantic analysis using deep learning techniques[10].

### **3 Methodology for Opinion mining**

Humans have a variety of emotions – unhappy or happy, fascinated or uninterested, and nice or negative. Knowing the different varieties of sentiment evaluation is essential. We may use sentiment evaluation for numerous purposes; however, we need to understand which one suits our reason the best. Different sentiment evaluation models are to be had to seize this style of emotions.

Gathering the data in which the analysis is to be drawn out is the prior step to be done and continuing to visualization of the numerical attributes present and figuring out the dependencies of various attributes using correlations. Here, as there is no strong correlation among the attributes present in the dataset, there cannot be a picturization of dependent attributes.

Continuing with classification tasks by setting a target attribute, user-defined function basing upon the ratings given to a product by the customer. Figuring out the compatibility between this ratings and review is the main method to be followed for classification. The target variable can be set on the values of rating: Positive (4, 5), neutral (3), negative (1, 2).

As, the classification algorithms can be implemented only on the numerical attributes. Here, the rating attribute is numerical but the reviews attribute is categorical. Converting this categorical attribute into numerical is achieved using BagOfWords strategy, this strategy is so essential to convert the attribute of categorical into numerical, i.e.; converting into feature vectors is the important step. Before of feature vector construction, text pre-processing must be done.

The text pre-processing is done using tokenization (breaking the whole sentence into individual words named tokens), stop words (removing all the words which have less importance in classification), lowercasing, occurrence counting (counting word occurrences). These all can be achieved using CountVectorizer () method. This is imported from sklearn. feature\_extraction.text library. The result indicates the training samples with the distinct words. As, in the huge reviews or so, there will be more occurrences of words and greater average count values which have minimal word meaning. This will neglect the small reviews that have lesser average count with same frequency and carries high meaning. So, to avoid this negligence we use Tfidf transformer. Applying this transformer, since the dataset does not have any dominations as such, the result does not change. Hence, a feature vector is

drawn out from the categorical attribute – reviews. On these two numerical attributes the below approaches are implemented as follows,

### 3.1 Naïve Bayes approach

Naïve: It is called Naïve on the grounds that it expects that the event of a specific element is autonomous of the event of different highlights. For example, if the natural product is distinguished on the foundations of shading, shape, and taste, at that point red, circular, and sweet organic product is perceived as an apple.

Henceforth each component exclusively adds to recognize that it is an apple without relying upon one another.

Bayes: It is called Bayes since it relies upon the guideline of Bayes' Theorem.

It follows the rules of Bayes theorem:

$$P(x/y) = P(x) P(y/x) / P(y)$$

In this aspect of sentiment analysis,

$$P(\text{sentiment/review}) = P(\text{sentiment}) P(\text{review/sentiment}) / P(\text{review})$$

### 3.2 Logistic regression approach

Logistic Regression is basically a supervised learning algorithm which is used to predict the probability of the class label. This is not only used for classification among the labels but also used in the case of estimation of probability, that a particular data tuple is related or is belonging to the class.

Estimating probabilities: It is an easy method, calculating the sum of input attributes including their bias term.

Vectorised form of the probabilities,

$$P = h\theta(x) = \sigma(\theta^T \cdot x),$$

Where sigmoid function yields an output of a number between 0 and 1.

Where sigmoid function,

$$\sigma(t) = 1 / (1 + \exp(-t))$$

$$P(x) = e^{(b_0 + b_1 \cdot x)} / (1 + e^{(b_0 + b_1 \cdot x)})$$

In this aspect of sentiment analysis,

$$P(\text{review}) = e^{(b_0 + b_1 \cdot \text{review})} / (1 + e^{(b_0 + b_1 \cdot \text{review})})$$

The model prediction,

$$y = 1 \text{ if } P \geq 0.5,$$

$$0 \text{ if } P < 0.5.$$

Training various models to get the highest accuracy is the main point in the project. So, in this logistic regression model, from analysis of cost function we can improve the accuracy of the training model on the

dataset, any model should yield high probabilities for  $y=1$  and low probabilities for  $y=0$ .

In the below cost function it is so clear that the positive probability i.e.;  $-\log(P)$  grows very large when  $P$  approaches to 0, concluding the model estimating a low probability which is approaching 0. Similarly, in the negative probability case i.e.;  $-\log(1-P)$  the cost function grows high whenever  $P$  approaches 1.

Cost function in logistic regression is as shown,

$$C(\theta) = -\log(P) \text{ if } y=1,$$

$$-\log(1-P) \text{ if } y=0$$

The above cost function is only for a single instance, the cost function for the whole model predicting on the training dataset is the average of the cost functions of individual data tuple. The average is named as log loss,

$$\text{Log loss} = -1/m \left( \sum_{i=1}^m (y(i) \log(p(i)) + ((1 - y(i)) \log(1 - p(i)))) \right)$$

This average cost function or log loss is a convex function. Therefore, Gradient Descent is so guaranteed to estimate and figure out the global minimum with a perfect learning rate. In our case by training this model, if a particular data is given basing upon the classification the model predicts the probabilities and a threshold value is set to judge the final class label.

### 3.3 Support vector machine approach

Support vector machine is a classification algorithm which can be used in both the cases i.e.; classification and regression.

In SVM, the plot is pointed by using data items in the  $n$ -featured space. Then the classification is performed by using the hyperplanes that are used to differentiate the classes into categories. Here, classifying the reviews (decisions) using decision boundaries. Here, this decision boundary is so fixed with the parallel dashed lines called large margin classifiers. These large margin classifiers are so close to the classes. Basically every class has an instance representing the whole troop and is heading towards the decision boundary of course is placed on the margin. These type of instances which are on the margin are called support vectors.

Here, to avoid the problem of outliers i.e.; instance which belong to one class is visualized based upon its value in the other troop and if the instances belonging to the off- street are more in number the model is of course not predicting the related number of accuracy and is called hard margin, implementation of the soft margin can be used in this case.

To decrease the off-street width, using C hyper parameters, i.e.; a small C value yields a big width of off-street and vice versa. Here, with a small off-street it yields to a huge case of margin violations, that is the inclusion of irrelevant class instances into other classes. But, with higher C value there is a decrease in off-street width but results to a smaller margin. By the instance division, a particular instance can be judged basing upon its placement, and is finalized by the troop in which it is involved.

This SVM classifier can be implemented using SGDClassifier, where SGD is stochastic gradient descent classifier in which random classification is done, by setting loss function to hinge and alpha to  $(1/(m*C))$ . This method though it is not fast as LinearSVC but is used under training datasets with huge number and to compete the online models for classification.

Therefore, SVM is capable to classify and visualize the planes with both linearly defined boundaries and non-linearly defined boundaries. The classification using hyper planes is set to divide among the positive, negative and neutral reviews.

### 3.4 Decision trees approach

As, it is mentioned in the name itself, the classification is done in the form of tree by splitting the data values and it helps us in analysis various statistical parameters. The whole tree can be built using two parameters: decision nodes and leaves. Where decision nodes are the nodes on which decision the data tuples are classified and the leaves are the data tuples. The way it works is, at first the root is said to be the training set which is used and depending on the attribute values, the nodes are recursively distributed. The form of a decision tree is also called as Disjunctive Normal form. The main challenge here is attribute selection at each level. This can be done by using Information Gain and Gini Index.

Let us take an Instances Set 'X', attributes as 'T', a subset of instances set with attributes as 'Xv' and 'Val(T)' as the set of all the values possible for 'T'. Then,

$$\text{Gain}(X, T) = \text{Entropy}(X) - \sum \text{Val}(T) \left[ \frac{|X_v|}{|X|} * \text{Entropy}(X_v) \right]$$

And the Gini Index is calculated as

$$\text{Gini} = 1 - \sum_{i=1}^n (p_i)^2$$

The decision nodes in this project is the attribute in which is defined basing upon the ratings attribute.

### 3.5 Random forest approach

Random forest is a combination of various decision trees which are merged together to get a good predictions and to improve the accuracy. Basically, it consists of huge number of decision trees and are merged together. Here, each tree gives a particular class prediction to give an accurate model prediction. This algorithm has almost same hyper parameters as decision tree but injecting additional randomness to the model while merging the trees, the accuracy increases. This is done because outcome of merged sources will outperform that of individual sources. Models that are uncorrelated will produce a group of predictions that are more accurate when compared to single individual ones. This effect is caused because the merged trees protect every singular one from their errors. Identifying signals in the features to build models using them which are better than random guessing and to make sure low correlations are there among the predictions made by singular trees. It combines the concepts of random selection and bagging by generating a set of regression trees and merging them efficiently.

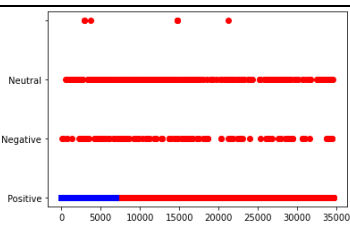
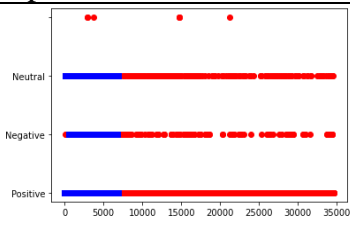
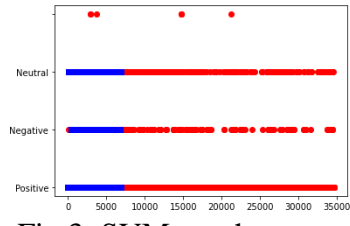
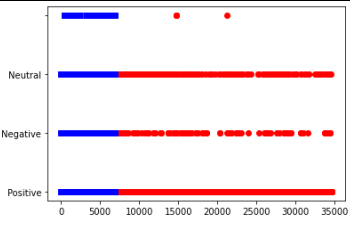
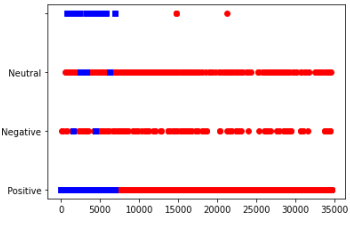
To increase prediction capabilities of random forest, hyper-parameters are used. For incrementing the predicting power, N\_Estimators hyper-parameter: number of trees built before taking means of predictions Max\_Features hyper-parameter : max features the random forest has split into a node Min\_Sample\_Leaf hyper-parameter: min leaves for splitting internal nodes. For incrementing the model speed, N\_Jobs hyper-parameter : number of processors to use Random\_State hyper-parameter : definitive value for training data OOB\_score hyper-parameter : out of bag cross validation method.

## 4 Comparison of Approaches

Comparing the above mentioned approaches:

The below comparison is shown by the graph representations of the classification performed on various models. The red color depicts the data points of the dataset which is to be categorized into positive, neutral and negative. The blue color depicts the prediction done by the model on the test dataset.

The models with their accuracies as 93.4%, 93.7%, 93.8%, 89.99%, 93.5% shows the performance on the test data.

Classifier Name	Graph	Accuracy
Naïve Bayes Approach	 <p>Fig 1: Naïve Bayes graph</p>	93.4%
Logistic Regression Approach	 <p>Fig 2: Logistic Regression graph</p>	93.7%
Support Vector Machine Approach	 <p>Fig 3: SVM graph</p>	93.8%
Decision Tree Approach	 <p>Fig 4: Decision tree graph</p>	89.99%
Random Forest Approach	 <p>Fig 5: Random forest graph</p>	93.5%



It is clear that SVM has the high prediction, continuing with fine tuning the model using GridSearchCV, the performance analysis can be depicted. This indicates the best score, best estimators and best parameters for improving accuracy.

Giving best score as 0.9372585915169948 with best estimators: pipeline (steps= [('vect', CountVectorizer (ngram\_range= (1, 2))), ('tfidf', TfidfTransformer (use\_idf=False)),('SVC', LinearSVC())]), best parameters: {'tfidf\_\_use\_idf': True, 'vect\_\_ngram\_range': (1, 2)}

## 5 Classification Report:

**Precision:** Decides the number of items chosen were right. (Percent of predictions are correct).

**Recall:** Discloses the number of the items that ought to have been chosen were really chosen. (Percent of positive cases chosen).

**F1-score:** estimates the loads of review and exactness (1 methods precision and recall are similarly significant, 0 in any case). (Percent of positive predictions are chosen).

	precision	recall	f1-score	support
	0.00	0.00	0.00	7
Negative	0.71	0.29	0.41	161
Neutral	0.39	0.07	0.11	286
Positive	0.95	1.00	0.97	6472
accuracy			0.94	6926
macro avg	0.51	0.34	0.37	6926
weighted avg	0.92	0.94	0.92	6926

Accuracy: 0.9395033208200981

Fig 6: Classification report

Fine tuning helps the model to improve its accuracy, here increment is from 93.8% to 93.9%.

## 6 Conclusion

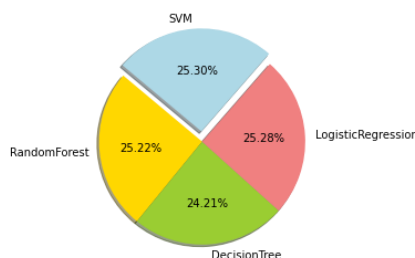


Fig 7: Comparison of various models

As, it shows the accuracy when various models are trained Support Vector Machine has the highest among all. So, the sentiment analysis using the supervised learning algorithm Support Vector Machine predicts based upon the ratings and reviews attributes with high accuracy.

## **7 Future Scope**

Extending the model to different languages, so that we can classify opinions in languages other than English. Sentiment analysis can be carried out using un-supervised learning algorithms. By sentiment analysis on reviews of customers, it can be developed for improving customer experience and developing the market.

## **8 References**

- [1]. Sentiment Classification on Twitter and Zomato Dataset Using Supervised Learning Algorithms: Rajkumar S. Jagdale; Sonal S. Deshmukh, INSPEC Accession Number: 20307476
- [2]. Supervised Learning Based Approach to Aspect Based Sentiment Analysis: Nipuna Upeka Pannala; Chamira Priyamanthi Nawarathna; J. T. K. Jayakody; Lakmal Rupasinghe; Kesavan Krishnadeva. , INSPEC Accession Number: 16726214
- [3]. Sentiment analysis techniques in recent works: Zohreh Madhoushi; Abdul Razak Hamdan; Suhaila Zainudin. INSPEC Accession Number: 15419967
- [4]. Predicting Supervise Machine Learning Performances for Sentiment Analysis Using Contextual- based approach: Azwa Abdul Aziz; Andrew Starkey. INSPEC Accession Number: 19347353
- [5]. Twitter sentiment analysis using various classification algorithms: Ajay Deshwal; Sudhir Kumar Sharma. INSPEC Accession Number: 16544168
- [6]. M. Susmitha and R. L. Pranitha, "Performance assessment using supervised machine learning algorithms of opinion mining on social media dataset," in Proceedings of Second International Conference on Advances in Computer Engineering and Communication Systems ICACECS, 2022, pp. 419–427, doi: 10.1007/978-981-16-7389-4\_41
- [7]. Mekala Susmitha, Shaik Razia  
"Utilization of deep learning and semantic analysis for opinion mining in information extraction: a review" in Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), p-ISSN: 2502-4752, e-ISSN: 2502-4760 , April 2023 Vol 30, No 1

- [8]. H. Lin et al., "Detecting stress based on social interactions in social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1820–1833, Sep. 2017, doi: 10.1109/TKDE.2017.2686382
- [9]. Deepthi, Dara, and B. V. SeshuKumari. "Identification and Recognition of Facial expression using CNN Algorithm." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* Vol.12No.6, 5<sup>th</sup> April 2021 pp:, 5496-5504, ISSN: 13094653
- [10]. Venkata Seshu Kumari, B., A. Giri Prasaad, M. Susmitha, and Roheet Bhatnagar. "Exploring Different Approaches for Parsing Telugu." In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019)* 4, pp. 546-555. Springer International Publishing, 2020.