**International Journal of Engineering Technology and Applied Science**
**(ISSN: 2395 3853), Vol. 7 Issue 4 June 2021**

# AN ENHANCED SENTIMENT CLASSIFICATION OF AMAZON MOBILE PRODUCT REVIEWS USING DECISION TREE ALGORITHM

Deepika Gaur[1], Chetan Gupta[2]
deepikajuly.gaur17@gmail.com[1], chetangupta.gupta1@gmail.com[2]
M.Tech Scholar, PG Dept. of CSE[1], Asst. Prof., PG Dept. of CSE[2,3]
SIRTS, Bhopal[1, 2, 3]

***Abstract*— for the past 20 years, internet shopping has been on the rise, and several e-commerce companies, such as Amazon, have sprung up to satisfy the demand. As a result, a single product can be purchased on multiple websites, with pricing varying. Customers normally want the best quality at the lowest price, but they can't check it out for themselves. Therefore, sentiment analysis has proven essential to understand a product's popularity among the buyers all over the world. Sentiment analysis is a classification process whereby machine learning techniques are applied on text-driven datasets in order to analyze its sentiment. In these, we proposed Decision Tree algorithm with Count Vectorizer feature extraction techniques on Amazon mobile product review dataset.**

***Keywords*— Sentiment Analysis, Amazon Reviews, Machine Learning, Decision Tree, Count Vectorizer.***

## I. INTRODUCTION

People nowadays are more prone to seek assistance from the internet and other technologies. People are more reliant on current technologies than on traditional methods of doing things. Sentiment analysis is one of the most rapidly growing research areas in computer science, making it difficult to keep up with all of the activity. We present consumer feedback reviews on products, in which we use opinion mining, text mining, and sentiments to change the surrounding world's perspective on a given product This holds true even in terms of their decision-making. People frequently turn to the internet or other services to help them make sense of their bewilderment. Large amounts of misunderstanding develop when purchasing a product, and customers

frequently check for product reviews before making a purchase. [6]. Sentiment polarity categorization is a key challenge in sentiment analysis that is addressed in this study. The data for this study was gathered from Amazon.com online product reviews. A sentiment polarity categorization and POS procedure has been developed, as well as full descriptions of each stage. Pre-processing, pre-filtering, biasing, data accuracy and other processes are included in this procedure.

Our main target is to build a system which generates a rating of any product. This rating is solely dependent on public opinion of a product. Depending on public opinion, a products rating would be given. "What others think? It is usually important information in a decision-making process. Everyday people discuss about various products on blogging websites and social websites.

Sentiment analysis on product reviews has now becoming a very fashionable in text mining and Natural Language Processing research [4]. Here, we are reviewing the correlation between Amazon product reviews and therefore the rating of the products given by the purchasers [7].

### Decision Tree

The most powerful and widely used tool for classification and prediction is the decision tree. A decision tree is a flowchart-like tree structure in which each internal node represents an attribute test, each branch represents the test's outcome, and each leaf node (terminal node) holds a class label. Both

categorical and continuous data can be used with decision tree techniques. The data is divided into two or more homogenous sets in this method. This is frequently done in order to generate as many unique groups as feasible based on the most important attributes/independent variables.

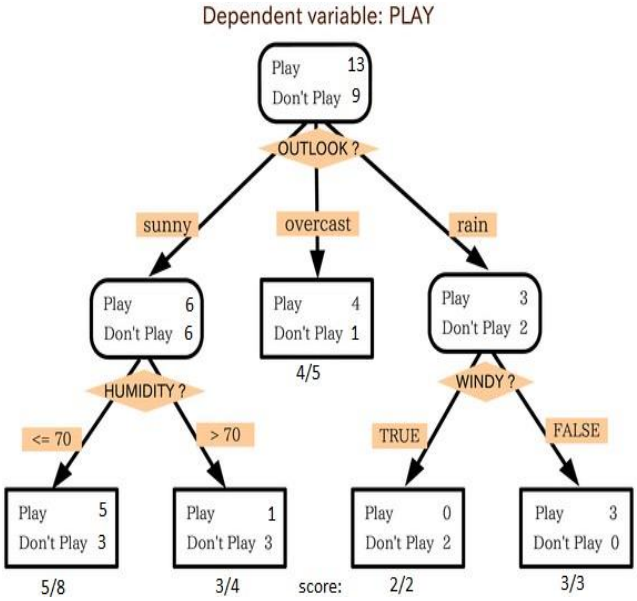


**Figure .1: Decision Tree Induction**

### Feature Extraction

There are a variety of approaches for converting text into vector features. Because machine learning algorithms cannot work directly with raw text data, we must convert it to numeric form (kind of vectors of numbers). There are various feature extraction techniques in which BOG (Bags of Words) technique works on frequency of works which is present into training dataset.[11]

## II. LITERATURE REVIEW

Several attempts have been made in the past to create algorithms that automatically identify movie reviews as Thumbs Up or Thumbs Down. [11].

A system is provided for categorising reviews of a variety of things, including movies. The algorithm finds two word sequences comprising specific combinations of nouns, adjectives, and adverbs and estimates their semantic orientation by looking for the phrase's co-occurrence frequency with the phrases excellent and terrible in WWW-based texts [2].

Situations during which the reviewer deliberately contrasts her overall opinions with evidence that opposes this opinion, as within the sentence [3].

This study proposes an universal method for extracting opinion features from online reviews by leveraging the differences in opinion feature statistics between two collections, one domain-specific and one domain-independent [13].

In this paper they present user feedback reviews on products, in which they use opinion mining, text mining, and feelings to change people's minds about a product. Online product reviews from Amazon.com were used in this study as data. A sentiment analysis of the obtained reviews was carried out [15].

## III. PROBLEM DEFINITION

To extract complete, accurate, and actionable information from a piece of text, it's crucial to not only recognise each of those five parts separately, but also to comprehend how they interact to provide the full context and sentiment. Machine learning and data processing are used in natural language processing to provide a more full picture, but the inherent complexity of language makes it challenging to ensure that computers appropriately analyse tone and context.

## IV. PROPOSED WORK

The transformation of texts into features can be done in a variety of ways. For the study of Amazon mobile product reviews, we used the Decision Tree method. Decision trees are applied math data processing technique that categorical freelance attributes and a dependent attributes logically AND during a tree formed structure. Call tree sometimes separates the complicated downside into several straightforward ones and resolves the sub issues through repeatedly

mistreatment .Decision trees square measure prognosticative call support tools that make mapping from observations to doable consequences.

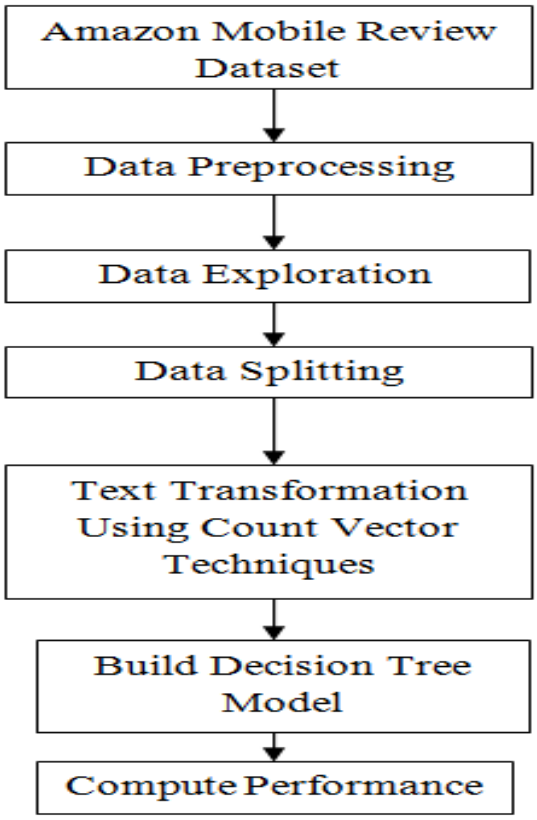


**Figure 2. Proposed Block Diagram**

### Algorithm Steps

**Step-1.** Loading Dataset: First we can collect the amazon mobile product review dataset from web and stored it for processing.
**Step-2.** Data Pre-processing: In these we can remove the missing fields records from the dataset and also we remove the neutral reviews, Not the dataset is only consist positive and negative review.
**Step-3.** After pre-processing we can analyse the data by exploration.
**Step-4.** Feature Extraction: After pro-processing we apply feature extraction techniques for training a model.
**Step-5.** Building a Model: We can build three Decision Tree Model based on Count Vector feature extraction techniques.
**Step-6**. Compute the Model performance.

### Decision Tree

**Given**:

Partition Data (D), - which is an set to tuplues of training lebel class

List_of_Attribute;

Method_of_Selecting_Attribute,which is used to find bestcondition of splitting.

1. Finding a starting node N for tree.
2. ApplyMethod_of_Selecting_Attribute(D, attribute_list) ;

```
# 2) Import Data

amazon_data = pd.read_csv("Amazon_Unlocked_Mobile.csv")

amazon_data = amazon_data.dropna(axis = 0)

amazon_data.shape

(334335, 6)

amazon_data.head()
```

| | Product Name | Brand Name | Price | Rating | Reviews | Review Votes |
|---|---|---|---|---|---|---|
| 0 | "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 5 | I feel so LUCKY to have found this used (phone... | 1.0 |
| 1 | "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 4 | nice phone, nice up grade from my pantach revu... | 0.0 |
| 2 | "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 5 | Very pleased | 0.0 |
| 3 | "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 4 | It works good but it goes slow sometimes but i... | 0.0 |
| 4 | "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7... | Samsung | 199.99 | 4 | Great phone to replace my lost phone. The only... | 0.0 |

3. Label N with splitting condition;
4. Updating theattribute_list

## V. EXPERIMENTAL & RESULT ANALYSIS

All of the tests and their results are analysed in a Jupyter notebook (IDE of python). The amazon mobile review datasets (shown in figure 3) can be loaded.

**Figure 3. Loading Amazon Mobile Product Review Dataset**

After loading the dataset we can explore the data and then we start processing the data, we can remove the missing records from the dataset and also the dataset contains review rating between 1 and 5. We can convert the records into positive reviews who's rating is 4 or 5, Negative review who's rating is 1 or 2, And remove the neutral reviews from the datasets who has review rating is 3, the pre-processing steps are shown in figure 4.

```
amazon_data=amazon_data[["Reviews","Rating"]]
```

```
amazon_data.head()
```

| | Reviews | Rating |
|---|---|---|
| 0 | I feel so LUCKY to have found this used (phone... | 5 |
| 1 | nice phone, nice up grade from my pantach revu... | 4 |
| 2 | Very pleased | 5 |
| 3 | It works good but it goes slow sometimes but i... | 4 |
| 4 | Great phone to replace my lost phone. The only... | 4 |

```
amazon_data_pos=amazon_data[amazon_data["Rating"].isin([4,5])]
```

```
amazon_data_pos.head()
```

| | Reviews | Rating |
|---|---|---|
| 0 | I feel so LUCKY to have found this used (phone... | 5 |
| 1 | nice phone, nice up grade from my pantach revu... | 4 |
| 2 | Very pleased | 5 |
| 3 | It works good but it goes slow sometimes but i... | 4 |
| 4 | Great phone to replace my lost phone. The only... | 4 |

**Figure 4. Preprocessing of Data**

After pre-processing we can split the dataset into training and testing dataset shown in figure 5, and on training data we can apply count vector feature extraction techniques on which we can trained our predictive Decision Tree model.

3)Spilt Train And Test data

```
from sklearn.model_selection import train_test_split
```

```
ta,x_test_data,Y_train_data,y_test_data=train_test_split(amazon_data_filtered["Reviews"],amazon_data_filtered["r"],test_size=0.3)
```

```
x_test_data.shape
```

```
(12000,)
```

```
Y_train_data.head()
```

**Figure 5. Splitting the Dataset**

We can split the data into 70:30 ratio means from total we can split the 12000 records for testing our model.Then we transform text into vector for which we use count vectorizer techniques allows us to use bag of words approach by converting collection of text documents in to a matrix of token counts shown in figure 6.

4) Text Transformation

```
#Count vectorization Technique
```

```
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
```

```
count_vector=CountVectorizer(stop_words="english")
```

```
count_vector.fit(X_train_data)
```

```
CountVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
                lowercase=True, max_df=1.0, max_features=None, min_df=1,
                ngram_range=(1, 1), preprocessor=None, stop_words='english',
                strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, vocabulary=None)
```

```
print(count_vector.get_feature_names()[0:20])
print(count_vector.get_feature_names()[-20:])
```

```
['00', '000', '00emotional', '00now', '00pm', '00so', '00time', '01', '013435003182980', '014', '018633051660f', '02', '03', '0
4', '04th', '06', '07', '08', '0804245', '09']
['zenphone', 'zenphone2', 'zentalk', 'zenui', 'zero', 'zf2', 'zillion', 'zip', 'ziploc', 'zippy', 'zmax', 'zone', 'zones', 'zoo
m', 'zooming', 'zte', 'item', 'óptico', 'ýn', 'šo']
```

**Figure 6. Text Transformation**

We can build a model on the features and the prediction results are shown in figure 7.

```
# Decision Tree

from sklearn import tree

tree_model = tree.DecisionTreeClassifier()

tree_model.fit(X_train_data_new,Y_train_data)

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=N
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=Nc
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')

predictions["DecisionTree"] = tree_model.predict(x_test_data_new)

accuracy_score(y_test_data,predictions["DecisionTree"])

0.9378333333333333
```

**Figure 7. Prediction result of Decision Tree model**

The validity of the model can be observed using error or accuracy of the model along with “false positive” and “false negatives”.

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of data points}}$$

| Algorithm | Testing Samples | Accuracy |
|---|---|---|
| Random Forest (Existing) | 12000 | 92.00% |
| Decision Tree (Proposed) | 12000 | 93.78% |

**Table 1. Performance Comparison**

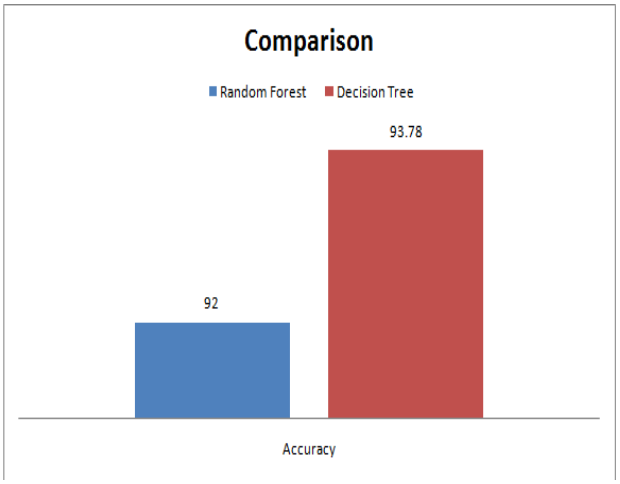


**Figure 8. Comparison of Accuracy**

## VI. CONCLUSION

Customers normally want the best quality at the lowest price, but they can't check it out for themselves. Sentiment research has proven to be critical in determining a product's popularity among purchasers around the world. Sentiment analysis is a classification procedure that uses machine learning techniques to analyze the sentiment of text-driven datasets. In these we can build Decision Tree model based on count vectorizer feature extraction techniques such as BOW (Bag of Words, from experimental result we can say that Decision Tree model provides better accuracy as compared Random Forest.

## REFERENCES

[1] AbdullahilKafi, Sayeed Bin Hossain, Hossain Arif,“Feature-Based Mobile Phone Rating Using Sentiment Analysis and Machine Learning Approaches” inIEEE, May 2019.

[2] Pankaj, Prashant Pandey, Muskan, NitashaSoni,“Sentiment Analysis on Customer Feedback Data: Amazon Product Reviews”inIEEE, Feb 2019.

.

[3] Robert Dzisevič ;DmitrijŠešok,"Text Classification using Different Feature Extraction Approaches" inIEEE, April 2019.

[4] Xin Chen, Mihaela Vorvoreanu, and Krishna Madhavan "Mining Social Media Data for Understanding Students' Experiences",in IEEE2014.

[5] Junlan Feng "Robust Sentiment Detection on Twitter from Biased and Noisy Data"in , China, COLING 2010.

[6] Vimalkumar, M. Jadav "Analysis of Various Sentiment Classification Techniques" in IJCA 2016

[7] Yu Chen ;Genshe Chen "Scalable sentiment classification for Big Data analysis using Naïve Bayes", in IEEE, 2013.

[8] Liu, B., "Sentiment analysis and opinion mining", in "Synthesis lectures on human language technologies", in2012

[9] O. Etzioni, "Extracting Product Features and Opinions from Reviews" in, 2005.

[10] Gobinda G. Chowdhury (2006). "Natural language processing".

[11] Bing Liu list of words, http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar

[12] Bill McDonald list of words, http://www3.nd.edu/~mcdonald/Word_Lists.html.

[13] Muhammaad Zubair, Aurangzeb Khan, Shakeel Ahmad, Fazal Masud Kundi and Asghar, "A Review of Feature Extraction in Sentiment Analysis", ISSN 2090-4304 Journal of Basic and Applied Scientific Research 2014.

[14] Pankaj, Prashant Pandey, Muskan, Nitasha Soni, "Sentiment Analysis on Customer Feedback Data: Amazon Product Reviews", International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), India, 14th -16th Feb 2019