# ECMM447: Mini Project

# Amazon Alexa Review sentimental analysis

Student id:710026485

***Abstract*—The growth of data on the Internet has increased the demand for efficient information retrieval independent of type of data. Sentimental analysis has revolutionized the way analyzing and understanding the data. Besides, Sentimental analysis provides esteemed solutions to enrich target information. There is a large amount of research available in the area of Sentimental analysis, which highlights its significance. This project aims in training Sentimental machine learning models for natural language understanding using amazon reviews data. We try to predict the sentiment of the reviews and accuracy of the prediction using machine learning models.**

***Keywords— Sentimental Analysis, Machine learning, Natural language.***

## I. INTRODUCTION

Text and speech are common modes of interactions between human beings. This interaction is called as Natural language. It is important for us to interact with our computers in this modern era of technology, where computers are an important component of our daily life. Natural language processing (NLP) aids computers in comprehending human speech. NLP is a method of using Artificial Intelligence to manipulate human speech in order for computers to understand it. It has greatly simplified human-computer interaction. NLP is a rapidly growing technology because of the enormous growth and availability of Big Data, Modified Algorithms, and Powerful devices. human language has various meanings, many words have multiple meanings, and each statement can have a variety of sounds, such as sentiment, emotive or sarcastic. Sarcasm, threat, sentiment, exclamation, and other expressions are sometimes difficult for computers to understand. However, with the help of Natural Language Understanding (NLU), a subsection of Natural Language Processing, the machine can identify various sentiments that the user may express. NLP is used in a wide range of applications, including Business Analytics, Speech Recognition, and social media, machine translations, chatbots, search engines and text analytics. Customer reactions can be studied, social media conflicts can be handled by removing unpleasant comments, and insights from a company's customer base can be obtained.

## II. CONTEXT

Sentimental analysis is a field of natural language processing and machine learning that focus on polarity the context of any text as well as the emotions associated in the sentence. This aids in the extraction of essential data from computers with human-level precision.

The following types are used in semantic analysis. They are:

- Graded Sentiment Analysis
- Emotion detection
- Aspect-based Sentiment Analysis
- Multilingual sentiment analysis

## III. OBJECTIVES

a) Introduction
b) Context
c) Data Description
d) Data exploration
e) Methods
f) Semantic Analysis
g) feature extraction
h) Model
i) Conclusion

## IV. DATA DESCRIPTION

Our dataset consists of amazon Alexa reviews for different models. It has 3149 rows and 4 columns. The following figure describes the top five rows of the data.

| | rating | date | variation | verified_reviews |
|---|---|---|---|---|
| 0 | 5 | 31-Jul-18 | Charcoal Fabric | Love my Echo! |
| 1 | 5 | 31-Jul-18 | Charcoal Fabric | Loved it! |
| 2 | 4 | 31-Jul-18 | Walnut Finish | Sometimes while playing a game, you can answer... |
| 3 | 5 | 31-Jul-18 | Charcoal Fabric | I have had a lot of fun with this thing. My 4 ... |
| 4 | 5 | 31-Jul-18 | Charcoal Fabric | Music |

## V. DATA EXPLORATION

In preprocessing I have checked if there are any null/empty values in data. As there are no null values entire data set is used for the analysis. The figure describes count of reviews with respect to rating given for the product. We can observe most of the reviews are rated 5 and a smaller number of reviews are rated 2.

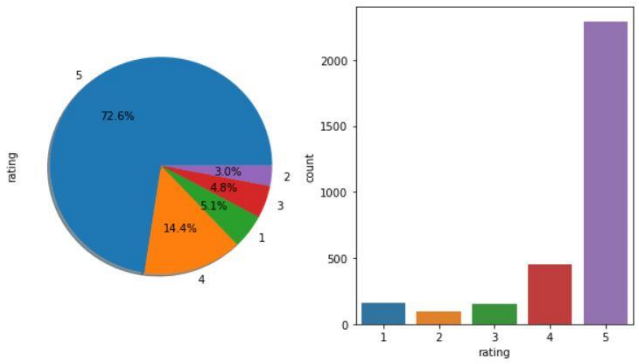
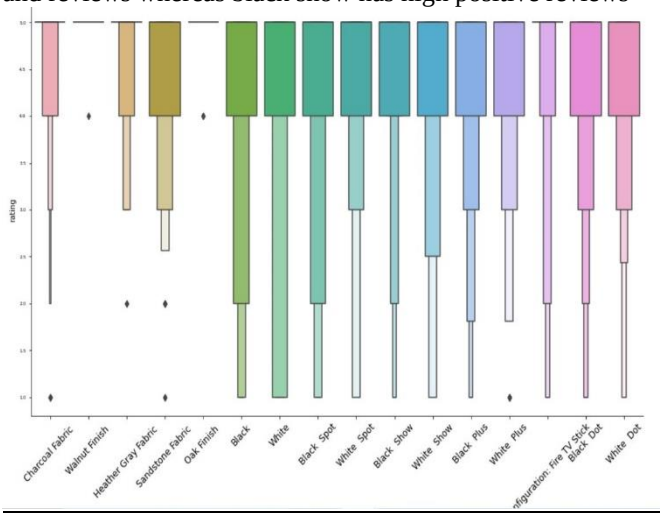



Fig: The plot above describes count of ratings of the product

The figure below describes rating for different variations of the product. We can observe that walnut has smaller rating and reviews whereas black show has high positive reviews




## VI. METHODS

### 1. TEXT PREPROCESSING

Text preprocessing is a crucial phase in the NLP workflow; without it, we wouldn't be able to analyze text data. The three major phases in text preprocessing are listed below.


### 2. Noise Removal

Noise is any text that is unrelated to the context of the data or the task that we are attempting to complete. HTML tags, white spaces,punctuations, and URLs are the most prevalent noise in text data. We remove the noise from the text in this step by using NLTK library. In our data set we used NLTK library to remove noises also we further processed data into lower case for good prediction results.

### 3. Text Normalization

Higher level normalisation is used to minimise the dimensionality of features so that machine learning models can handle the data more efficiently. Text data has various representations of the same word. For example, the words "play," "played," "player," "playing" are all versions of "play." These differences are useful for speech analysis but not so much for text analysis. We turn all of a word's discrepancies into their normalised form during text normalisation. We can perform Tokenization, lemmatization as part of text normalization.

- **Tokenization**:

In any NLP pipeline, tokenization is one of the first steps. Tokenization is the process of breaking down a large piece of text into small parts of words or sentences known as tokens. It is required because the words that make up a sentence determine its meaning. We may readily interpret the meaning of the text by studying the words in the text. We can utilise statistical tools and methodologies to gain deeper insights into the text once we have a list of words. In our analysis we used Textblob to tokenize the data of verified reviews column.

```
import nltk
nltk.download('punkt')
df["verified_reviews"].apply(lambda x: TextBlob(x).words).head()
```

- **Lemmatization:**

Lemmatization eliminates the word's inflected ending and returns it to its base/root/dictionary form. The word lemma refers to the root form of the word. In our analysis we used TextBlob library to lemmatize the data which is tokenized.

```
nltk.download('wordnet')
df['verified_reviews'] = df['verified_reviews'].apply
                        (lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))
```

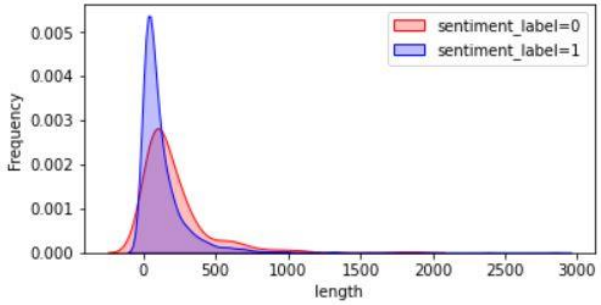
## VII. SENTIMENT ANALYSIS

Sentiment Analysis is a technique for analyzing the text's emotion. In other words, it is the process of determining if a text contains a positive or negative emotion. Because customers are free and more direct in expressing their opinions about the products or services they use these days, sentiment analysis has become an important tool for businesses to better understand their customers. Sentiment Analysis allows businesses to determine what type of emotion or sentiment their customers have for them. This can be quite beneficial because companies can enhance their products/services based on client feedback.

```
sia = SentimentIntensityAnalyzer()
nltk.download('wordnet')
df['verified_reviews'] = df['verified_reviews'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))
```

In our analysis we used SentimentalIntensityAnalyzer of NTLK library on reviews column to get polarity of the text. Here SentimentalIntensityAnalyzer is an object and polarity_scores method of that object is used to get the sentiment of the text. We have included polarity above zero as positive sentiment remaining as negative sentiment.

Fig: this graph below shows the frequency of length of reviews based on its sentiment




As part of analysis, I have created word cloud of negative and positive sentiment.


Fig: figure above shows Positive sentimental reviews


Fig: figure above shows negative sentimental reviews.

We can observe words in positive sentiment contain love, great, good, music which shows positive emotion of users whereas in negative reviews contain disappointed, problem, trouble, repair, annoying which indicates negative emotion.

## VIII. FEATURE EXTRATCION

**Text Vectorization**:

It is an NLP technique for the conversion of text into vector of real numbers. This can be used to find the word predictions and word similarities. Most used text vectorization methods are Tf-Idf, bag of words or Count vectorization.

In the count vectorization technique, a document term matrix is created, with each column containing the count of reviews showing the number of times a word appears in a document, also known as term frequency. The document term matrix is a collection of dummy variables that indicate whether a given word appears in the document. Each word in the corpus has its own column. if a particular word appears many times in positive or negative sentiment reviews, it has a strong predictive potential for detecting whether the review is positive or negative.

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
X_count = vectorizer.fit_transform(corpus)
vectorizer.get_feature_names()
X_count.toarray()
```

For feature extraction I have used the preprocessed reviews as corpus.

The TF-IDF method generates a document term matrix, with each column representing a single unique word, similar to the count vectorization method. The TF-IDF approach differs in that each cell does not represent the term frequency, but rather a weighting that emphasizes the relevance of that particular word to the document.

TF-IDF formula:

$$W_{x,y} = tf_{x,y^*} \log\left(\frac{N}{df_x}\right)$$

$W_{x,y}$= Word x within document y

$tf_{x,y}$= frequency of x in y

$df_x$= number of documents containing x

N=total number of documents

```
from sklearn.feature_extraction.text import TfidfVectorizer
tf_idf_word_vectorizer = TfidfVectorizer()
X_tf_idf_word = tf_idf_word_vectorizer.fit_transform(corpus)
```

In the above code I have used Tf-Idf vectorizer to extract features from the corpus of verified reviews.

## IX. MODEL

**Logistic Regression:**

To implement this model, I have taken the feature extracted corpus of both counter vectorizer and Tf-Idf. I obtained best results from Tf-Idf extracted features. I have used fit() method of LogisticRegression() to fit data and cross_validate method of sklearn to train and test data with 5 folds.

```
log_model = LogisticRegression().fit(X_tf_idf_word, y)
from sklearn.metrics import make_scorer, accuracy_score, precision_score, recall_score, f1_score

scoring = {'accuracy' : make_scorer(accuracy_score),
           'precision' : make_scorer(precision_score),
           'recall' : make_scorer(recall_score),
           'f1_score' : make_scorer(f1_score)}

results=sklearn.model_selection.cross_validate(log_model,
                X_tf_idf_word,
                y, scoring=scoring,
                cv=5)
```

The following are results obtained:

```
accuracy:  0.8605905064728594
f1 score:  0.9227488066269662
precision:  0.8591479534617357
recall:  0.9965779467680609
```

**Random Forest Classifier:**

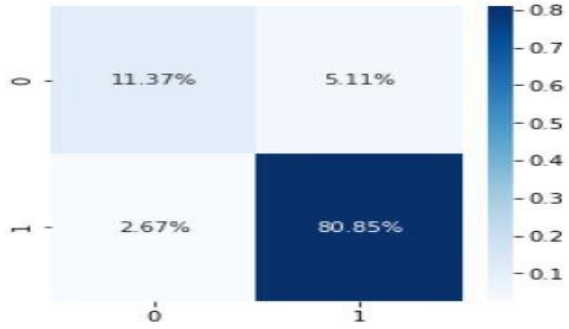
To implement this model, I have taken the feature extracted corpus of both counter vectorizer and Tf-Idf. I have used fit() method of RandomForestClassifier() to fit data and cross_validate method of sklearn to train and test data with 5 folds.

```
rf_model = RandomForestClassifier().fit(X_count, y)
results=sklearn.model_selection.cross_validate(rf_model, X_count, y,scoring=scoring, cv=5, n_jobs=-1)
```

```
accuracy: 0.9320422943952356
f1 score:  0.9599111941447399
precision:  0.9469797001169132
recall:  0.973384030418251
```

We are able to predict sentiment of the data with higher accuracy obtained from random forest classifier with accuracy of **0.932**.

I have plotted the confusion matrix after applying the random forest classifier.




## X. CONCLUSION

We have obtained a good F1 score and accuracy which shows that the model is a good fit. To further analyze the data, we can perform emotional analysis We understood Sentimental analysis using machine learning models is effective in understanding the natural language of humans. This can help in analyzing and using data in faster way. It is critical for any customer-centric enterprise to learn about its consumers and acquire insights from customer feedback in order to develop.

## XI. REFERENCES


1. https://arxiv.org/pdf/1301.3781.pdf
2. https://www.machinelearningplus.com/nlp/cosine-similarity/
3. https://medium.com/@adriensieg/text-similarities-da019229c894
4. https://towardsdatascience.com/nlp-in-python-vectorizing-a2b4fc1a339e
5. http://ceur-ws.org/Vol-2823/Paper13.pdf
6. https://www.analyticssteps.com/blogs/top-10-applications-natural-language-processing-nlp
7. https://arxiv.org/abs/1806.02847
8. https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation
9. https://www.analyticsvidhya.com/blog/2021/06/part-5-step-by-step-guide-to-master-nlp-text-vectorization-approaches/
10. https://monkeylearn.com/blog/beginners-guide-text-vectorization/
11. https://monkeylearn.com/sentiment-analysis/