NAME OF THE PROJECT

RATINGS PREDICTION

Submitted by:

V TARAK RAM SAI

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped you and guided you in completion of the project.

INTRODUCTION

- Business Problem Framing

  This is regarding a client who has a website where people write different reviews for technical products. Now the reviewer has to rate the review out of 5 stars. For this we have to build a model which can predict the rating by seeing the review.

- Conceptual Background of the Domain Problem

  The rise in E — commerce, has brought a significant rise in the importance of customer reviews. There are hundreds of review sites online and massive amounts of reviews for every product. Customers have changed their way of shopping and according to a recent survey, 70 percent of customers say that they use rating filters to filter out low rated items in their searches.

- Review of Literature

  This story demonstrates common **Natural Language Processing** tasks performed specifically to understand the relationship between low ratings and their corresponding comments. In the last years, interest in sentiment analysis in machine learning engineering domain has risen significantly. There are studies that analyse users satisfaction and developers emotions automatically by applying sentiment analysis.

- Motivation for the Problem Undertaken

  User reviews include information that is useful for analysts and product designers, such as user requirements, bug reports, feature requests, and documents of user experiences with specific product features. Sentiment analysis is the task of assigning a quantitative value to a text with respect to the opinion, sentiment or emotion towards various entities such as products.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  Any machine learning model should follow the below steps while dealing a business problem. They are:

  i.) Business Understanding: The first step is to comprehend the research's background, the problem description, and how the proposed project will achieve the goals.

  ii.) **Data Understanding:** The second stage requires collection of data listed in the project resources. This involves in determining the data requirements and exploring key data attributes.

  iii.) **Data Preparation:** The third stage involves data cleaning and should the handle the missing values in the data.

  iv.) **Modelling:** This involves determining the modelling technique and testing the design.

  v.) **Evaluation:** Here, we should evaluate the achieved results and should determine the performance of the model with best accuracy.

  vi.) **Deployment:** The last stage is implementation of the model.

- Data Sources and their formats

  The product reviews dataset contains user rating for each product and review text for each review.

- Data Preprocessing Done

  It entails converting raw data into comprehensible format that a machine learning model can understand. The data pre-processing involves data cleaning which involves handling missing values, transformation of data.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk
```

In [2]:
```
df=pd.read_csv('C:\\Users\\DELL\\Desktop\\insss\\rate_review.csv')
```

The head of the primary table is as follows:

	Review	Rating
	Always wanted a good printer in my budget, and...	4
	👍Epson Printer is very economical and useful f...	5
	Superb Product with Superb Command Sensation &...	5
	Very good nice 👍👍👍👍👍	5
	Good quality.....Photo supper..Plastic body li...	5

The datatype of each column is as follows

```
1 df.dtypes
```

```
Review    object
Rating     int64
dtype: object
```

Converting integer Rating column into string column.

```
1 # Converting integer column into string column.
2
3 df['Rating']=df['Rating'].apply(str)
```

```
1 df.dtypes
```

```
Review    object
Rating    object
dtype: object
```

Preparing the data

Raw text data are messy to work with and pre-processing steps are needed before training the computer on them. There are many different ways and order to prepare the text data. In this example, I perform the following tasks before loading the data into a DataFrame:

- **Text Normalization** — remove punctuations using Regular Expressions (Regex can be used to find basic and complex patterns in text).

```
# Remove Special Characters,numbers and punctuations.

df['Review1']=df['Review'].str.replace("[^a-zA-Z#]"," ")
```

- **Tokenize** — tokenization is the process of splitting text into smaller pieces called tokens.

```
# Individual words considered as Tokens

tokenized_review=df['Review1'].apply(lambda x: x.split())
tokenized_review.head()
```

```
0    [Always, wanted, a, good, printer, in, my, bud...
1    [Epson, Printer, is, very, economical, and, us...
2    [Superb, Product, with, Superb, Command, Sensa...
3                                    [Very, good, nice]
4    [Good, quality, Photo, supper, Plastic, body, ...
Name: Review1, dtype: object
```

- **Remove stop words** — stop words are words that are usually filtered out before pre-processing of the text. They are generally the most common words in a language that search engines have been programmed to ignore such as personal pronouns (I, you, he, she, it, etc.).

```
# Making a Stop Words list
nltk.download('stopwords')

stopwords= nltk.corpus.stopwords.words('english')
```

```
# Removal of Stop Words

def rem_stop(tokenized_txt):
    txt_clean=[word for word in tokenized_txt if word not in stopwords]
    return txt_clean
```

```
tokenized_review=tokenized_review.apply(lambda x: rem_stop(x))
tokenized_review.head()
```

```
0    [Always, wanted, good, printer, budget, I, get...
1    [Epson, Printer, economical, useful, home, off...
2    [Superb, Product, Superb, Command, Sensation, ...
3                                    [Very, good, nice]
4    [Good, quality, Photo, supper, Plastic, body, ...
Name: Review1, dtype: object
```

- **Stemming the text** — stemming is the process of removing a part of a word, or *reducing a word to its stem* or root. This might not necessarily mean we're reducing a word to its dictionary root. We use a few algorithms to decide how to chop a word off.

```
# Stem the words

from nltk.stem.porter import PorterStemmer
stemmer=PorterStemmer()

tokenized_review=tokenized_review.apply(lambda sentence:
                                        [stemmer.stem(word)
                                         for word in sentence])
tokenized_review.head()
```

```
0    [alway, want, good, printer, budget, i, get, e...
1    [epson, printer, econom, use, home, offic, cop...
2    [superb, product, superb, command, sensat, cle...
3                                  [veri, good, nice]
4    [good, qualiti, photo, supper, plastic, bodi, ...
Name: Review1, dtype: object
```

- **Vectorize the text** — in order for the computer to understand the text data, they need to be converted into a numerical representation, which is commonly described as **Word Embedding**. The most intuitive form of this process can be done through a form of one-hot encoding known as the bag-of-words and it describes the occurrence of words. In a bag-of-words, the columns represent each of the unique words in the entire corpus of documents, the rows represent each document, sentence, or topic, and the cells capture the number of times each word appears in each row.

```
from sklearn.feature_extraction.text import CountVectorizer
bow_vectorizer = CountVectorizer(max_df=0.90, min_df=2, max_features=1000)
bow = bow_vectorizer.fit_transform(df['clean_review'])
```

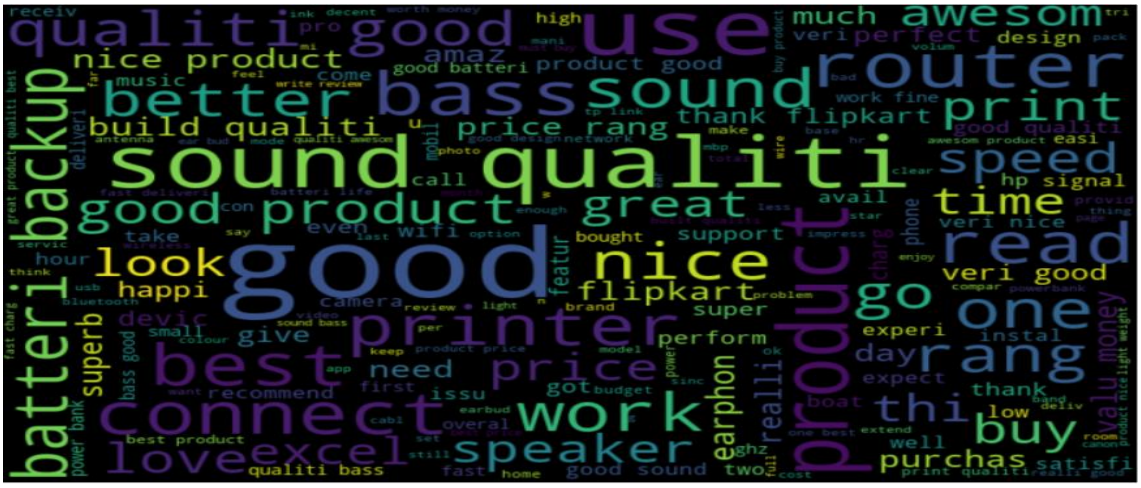
Encoding

Encoding categorical data is a process of converting categorical data into integer format so that the data with converted categorical values can be provided to the different models.

```
1 from sklearn.preprocessing import LabelEncoder
2 lab=LabelEncoder()
3 df['Rating']=lab.fit_transform(df['Rating'].values.reshape(-1,1))
```

- Data Inputs- Logic- Output Relationships

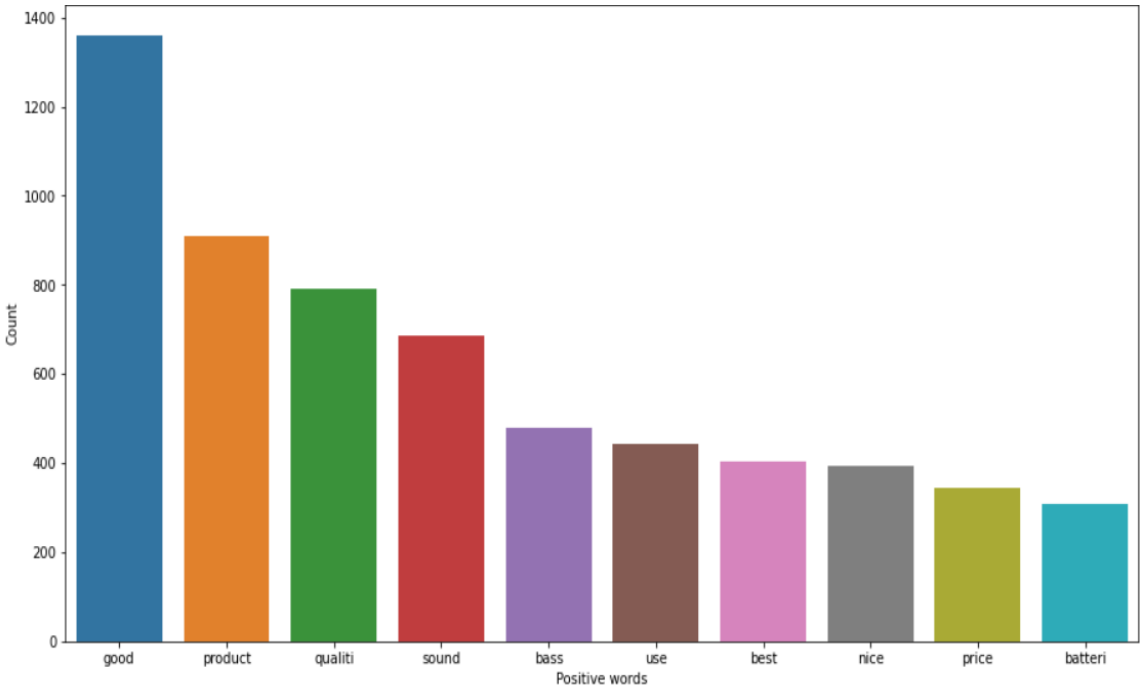
The Word Cloud for the reviews is as follows-



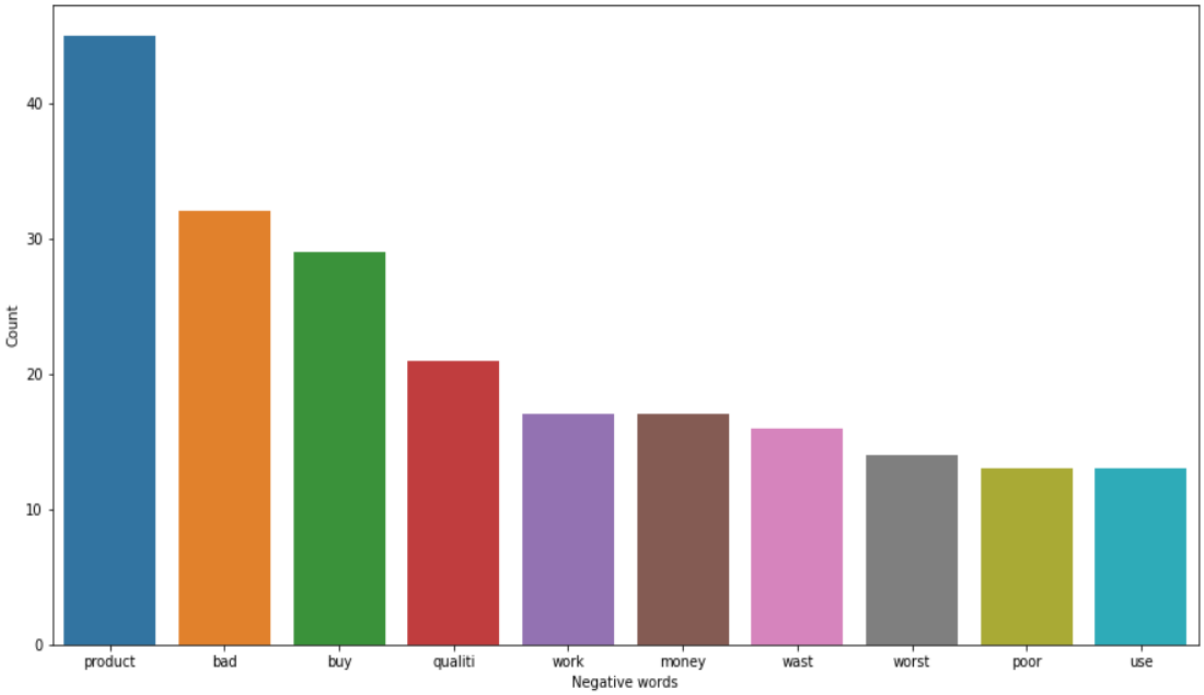
The Word Cloud for the positive reviews is as follows-

The Word Cloud for the negative reviews is as follows-

Top 10 positive words for the reviews are as follows-



Top 10 negative words for the reviews are as follows-



- Hardware and Software Requirements and Tools Used

The hardware requirements for the project includes a laptop with at least 4GB RAM. This project uses a Jupyter Notebook as a code editor. The Machine Learning models are implemented using python version 3.7 with libraries like numpy, pandas, matplotlib, seaborn and sklearn.

Model/s Development and Evaluation

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(bow, df['Rating'],
                                                    random_state=42, test_size=0.2)
```

- Multinomial Naïve Bayes

Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.
Naive Bayes is a powerful algorithm that is used for text data analysis and with problems with multiple classes. To understand Naive Bayes theorem's working, it is important to understand the Bayes theorem concept first as it is based on the latter. Bayes theorem, formulated by Thomas Bayes, calculates the probability of an event occurring based on the prior knowledge of conditions related to an event.

```
# fitting naive bayes to the training set
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import confusion_matrix

classifier = MultinomialNB();
classifier.fit(x_train, y_train)

# predicting test set results
y_pred = classifier.predict(x_test)
```

```
1  from sklearn.metrics import accuracy_score,classification_report
2
3  nb_clas=classification_report(y_test,y_pred)
4  acc=accuracy_score(y_test,y_pred)
5  print(nb_clas)
```

```
              precision    recall  f1-score   support

           0       0.50      0.50      0.50        10
           1       0.00      0.00      0.00         6
           2       0.17      0.14      0.15        37
           3       0.21      0.14      0.17        85
           4       0.73      0.81      0.77       323

    accuracy                           0.62       461
   macro avg       0.32      0.32      0.32       461
weighted avg       0.57      0.62      0.59       461
```

CONCLUSION

- This study presents a method to predict user ratings from user reviews. The proposed method uses a huge volume dataset with the same ranked ratings to learn the model and predicts ratings based on this oracle model. We obtained 62% accuracy for the rating prediction task.
- As a future study, it is planned to repeat this method for different OSS and also to measure community-based success of different applications with different metrics.
- NLP is a powerful solution that can take the product review system to the next level. The information obtained by these tools can be used by site owners to enhance the product with a focus on specific aspects, to classify product reviews on the basis of how they work i.e. positive or negative. This can be also used to make targeted advertising work properly.
- Online reviews provide a wealth of insights for a business, but can be intensive to read through and digest. There are many ways to try to automate this task. Currently, the leading approaches use deep learning models trained on online review data. The models best suited to this application are able to extract many different kinds of keywords, predict their sentiment, and classify them into relevant categories, which allows businesses to improve operations, make better decisions and elevate the customer experience with data.