Smart Bridge
Remote Smart Internship Program

Project Report
On
Analysis Of
Amazon Cell Phone Reviews

Internship ID - RSIP Career Basic AI 093
Project ID - SPS_PRO_180
Duration - 1 month

Submitted By -
Suryansh Choudhary

Table of Contents

1. Introduction

1.1 Overview

In recent years E-Commerce has exploded everywhere in the world, and the majority of the population prefers to buy products through these websites. Consequently large amounts of data in the form of reviews are produced which helps prospective buyers choose the right product. Furthermore, these reviews contain opinionated contents which can be useful for the company to identify the areas which need to be enhanced.

However it is impractical for the user to read each and every review about the product. Moreover, reading only a few reviews may present a biased idea about the product. It is quite possible that some of the reviews lack credible sources, which the users have no means to differentiate. Besides the reviews and ratings provided little to assess the specific features of the product. Due to all the above constraints, the user is unable to make a fully informed decision about the product.

1.2 Purpose

Our project aims at building a model to predict the helpfulness of the review and the rating based on the review text. Corpus-based and knowledge-based methods can be used to determine the semantic similarity of review text. We used Natural language processing for the analysis of the sentiment (positive or a negative) of the given review.

Sentiment analysis can be used to extract customer reviews from different sources on the internet. This technique implements various algorithms to analyze the corpus of data and make sense out of it. This technique helps to identify the orientation of a sentence thereby recognizing the element of positivity or negativity in it. Automated opinion mining can be implemented through a machine learning based approach. Opinion mining uses natural language processing to extract the subjective information from the data.

The Sentiment Analysis is used to classify the polarity (positivity or negativity behind the opinion) of the reviews. There are three main categories in sentiment analysis: sentence-level, unit aspect-level and document level sentiment analysis.

2. Literature Survey

2.1 Existing Problem

When a person thinks of buying a product, his or her next immediate action would be to search for the product on the internet. Internet gives him/her a lot of choices based on brand, price, model, features, colors, quality, rating, discounts and many more. Introduction of new products, new fashion, new model, new brand, new business, new technology, new services, new marketing strategies happen daily. These may leave a consumer perplexed and confused when having to make a choice. When it's tough to make a choice, we tend to get the feedback from the people who have already bought and used those products. Customers register their review on online shopping sites.

Amazon is the 1st site in the list of the top 10 online shopping sites. Amazon gives overall ratings for each product based on all ratings from the reviewers of the product. But from that overall rating a person cannot necessarily conclude on quality of all the features of the product.

Getting feedback on major features of the product will help the manufacturer or seller to improve the quality and marketing of the product. The ultimate solution for these problems is to read through the text reviews to know specifically which feature of the product is lacking customer satisfaction. But a product may have thousands of reviews which would make this task tedious. So, we need a system to do that for us.

2.2 Solution

We started building a model to predict the helpfulness of the review and the rating based on the review text. Currently consumers who write opinion and experience online are increasing. If the consumer reads the whole review it can spend much time. But if it is read without some evaluation it will be biased. Sentiment classification aims to overcome this problem by automatically classifying user review by positive or negative opinion.

This project aims at building a model to predict the helpfulness of the review and the rating based on the review text. Corpus-based and knowledge-based methods can be used to determine the semantic similarity of review text. We will be using Natural language processing to analyse the sentiment (positive or negative) of the given review. A simple web application is integrated to the model built.

3. Theoretical Analysis

3.1 Block Diagram

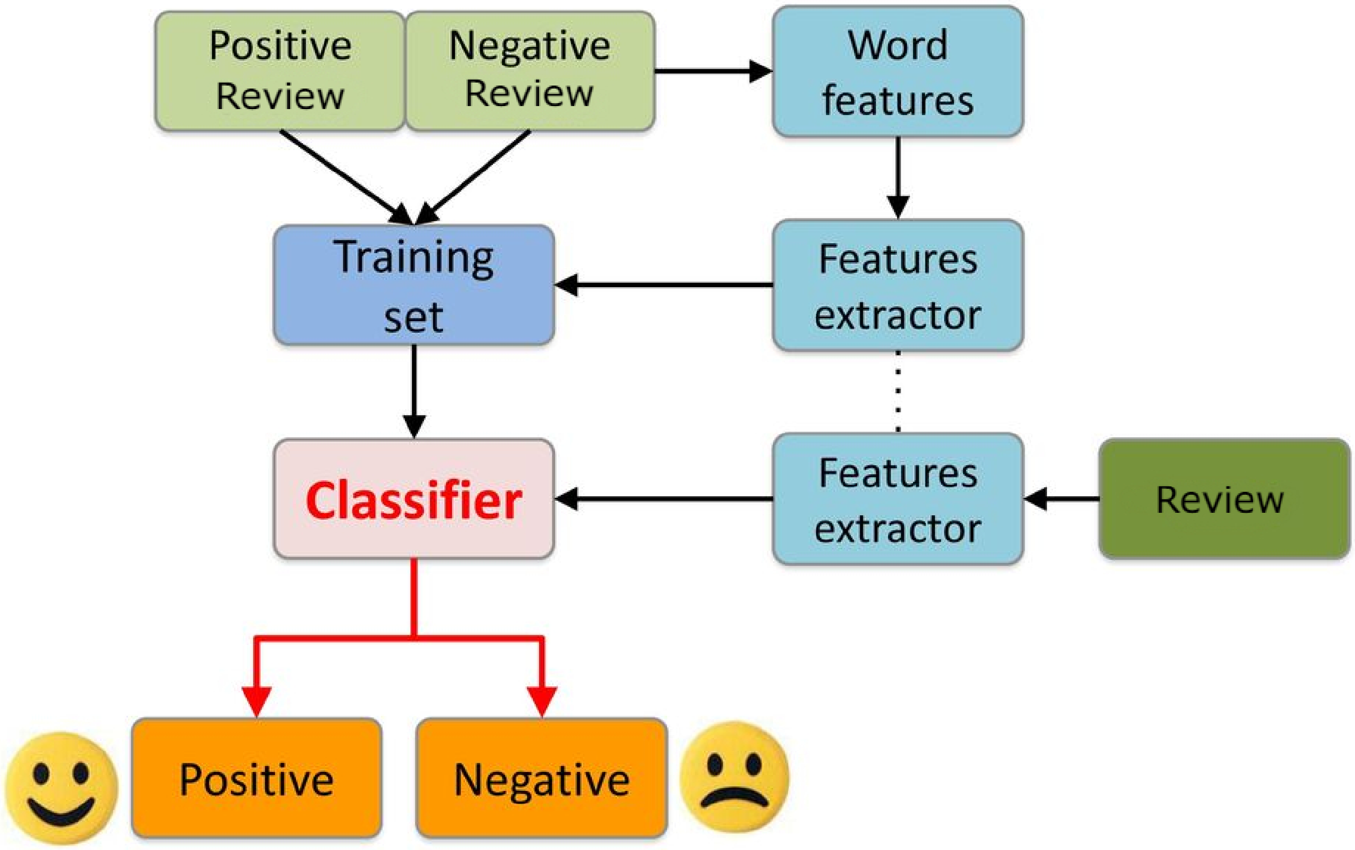

Fig 3.1.1 Block Diagram

3.2 Software Designing

For software designing we would need a compatible operating system which can be used to run the following language scripts: Python, Java Script and HTML. We would also need the following software:

- Anaconda
- Spyder
- TensorFlow
- Keras
- Flask
- Sublime Text

4. Experimental Investigations

1. **Data Collection:** The employed data set contains customer reviews about the Unlocked Mobile phones, which are taken from Amazon.com. The dataset is taken from “http://www.kaggle.com”. The data set contains the following information or attributes from the ‘unlocked phone’ category of Amazon.com. The given data is stored in a .csv file of size 24.2 MBs. This data is used to predict that what rating a customer would probably give, based on the sentiments hidden in their reviews.
2. **Preprocessing:** Preprocessing can increase the performance and accuracy of a classifier. The dataset is comprised of exclusively English reviews. Reviews contain information which are not clearly expressive and needed to be removed.
    a. **Remove Punctuation:** All punctuations are removed as they are not needed for analysis.
    b. **Remove Stop words:** Some words are called stop words. These pronouns, prepositions, conjunctions have no specific meanings. “I", "a", "an",” is”, "are", "as", "at", "from", "in", "this", "on", "or", "to", "was", "what", "will”, “with" etc. are examples of stop words, so these types of words have been removed. All text is converted into lower case.
    c. **Stemming:** It converts word into its grammatical root form. Stemming technique converts words like “teach”, ”teacher”, ” teaching”, ” teaches” to root word teach. M.F. Porter stemming algorithm is used for this task. It minimizes the feature set and increases the accuracy of classification.
3. **Model Building & Training:** We initialize the model and then we add input layer, hidden layer and output layer. We configured the learning process, trained and tested the model, and optimized the model. By training the model we achieved an accuracy of 95.7%. The model is then saved.
4. **Predicting:** The model is then loaded and various inputs are given to predict the responses. The predicted responses tell us how accurate our model is.
5. **HTML for Web UI:** A HTML file is made to accept review from the user and then model is deployed to display whether the review is positive or negative on the screen.
6. **Flask file:** A flask file is made to coordinate between the HTML script and the model. Once the file is run, it assigns a local host on the internet browser and displays its HTML file.

5. Flowchart

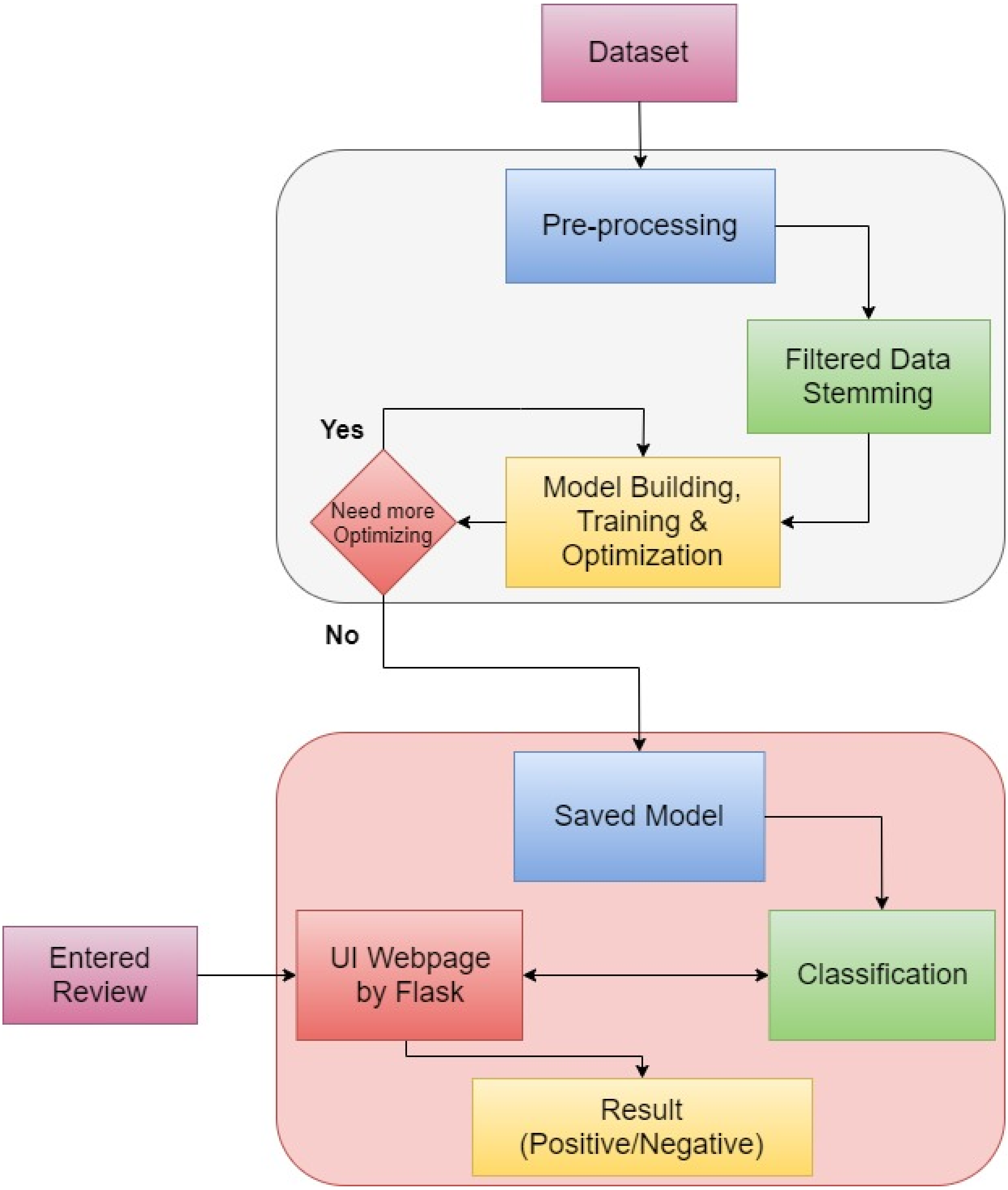

Fig 5.1 Flowchart

6. Result

The model and the flask file work very well together. We are able to distinguish a positive review from a negative one, which was our aim throughout the project. Our model is working perfectly and is predicting the correct sentiment from our reviews which was observed when we ran sample reviews through the model. We were able to achieve an accuracy of 95.7% which is a good measure for Natural Language Processing.

```
[33] entered_input = "It is a very bad product"

[34] x_intent = cv.transform([entered_input])
     y_pred = model.predict(x_intent)
     if (y_pred>0.5):
       print("It is a positive review")
     else:
       print("It is a negative review")

It is a negative review
```

Fig 6.1 Negative Review Sample

```
[37] entered_input = "It is a very good product"

[38] x_intent = cv.transform([entered_input])
     y_pred = model.predict(x_intent)
     if (y_pred>0.5):
       print("It is a positive review")
     else:
       print("It is a negative review")

It is a positive review
```

Fig 6.2 Positive Review Sample

7. Advantages & Disadvantages

7.1 Advantages

- Increases confidence in new customers.
- Brings credibility to products and the company.
- Knowing which product works best.
- Helps a company to improve their products.

7.2 Disadvantages

- One negative review of a product or business can skew a potential customer's view of them.
- We need to keep reviews current and up to date. Otherwise they will seem out of date and irrelevant.
- Disgruntled customers have the freedom to say whatever they like. This could lead to malicious or damaging information being posted.
- Lack of touch or feel of products during online shopping is a drawback.

8. Applications

1. **Business**
   a. Consumers voice
   b. Brand Reputation
   c. Online advertising
   d. On-line commerce
2. **Politic**
   a. Voting advise applications
   b. Clarification of politicians' positions
3. **Public Actions**
   a. Real-world events monitoring
   b. Legal matters
   c. Policy or government-regulation proposals
   d. Intelligent transportation systems

Sentiment Analysis Examples

1. **Brand Sentiment Analysis**
   Brand monitoring and reputation management is the most common use of sentiment analysis across different markets. It allows companies to track the perception of the brand by the customers, point out the specific details about the attitude, find patterns and trends and keep a close eye on the presentation by the influencers.

2. **Social Media Monitoring**
   Sentiment analysis can be used to automate media monitoring process and the accompanying alert system, monitor mentions or reviews of the brand on different platforms and categorize urgency of mentions according to the relevant scoring.

3. **Customer Support**
   Sentiment analysis can be used to give insight into customer's opinions regarding the product, intent Analysis for process automation and workflow management and customer prioritization

9. Conclusion

This project was a great learning experience with interesting challenges. My passion for Data Science and Artificial Intelligence drove me to choose this topic and complete it in a good way. Among the different tasks involved in the system implementation data collection and pre-processing was the most time-consuming one. Sentiment Analysis is a challenging task.

The sentiment analysis is being implementing through deep learning techniques. Deep learning consists of numerous effective and popular models, these models are used to solve the variety of problems effectively. Different studies have been discussed in this review to provide a deep knowledge of the successful growing of deep learning applications in the field of sentiment analysis. Numerous problems have been resolved by having high accuracy of both fields of sentiment analysis and deep learning.

We have created a model which can predict the Amazon mobile phone reviews whether it is a positive or negative review. We also created a UI interface to co-ordinate between our model and the HTML web page via the python file.

10. Future Scope

Future opinion-mining systems need broader and deeper common and commonsense knowledge bases. This will lead to a better understanding of natural language opinions and will more efficiently bridge the gap between multi-modal information and machine processable data. Blending scientific theories of emotion with the practical engineering goals of analyzing sentiments in natural language text will lead to more bio-inspired approaches to the design of intelligent opinion-mining systems capable of handling semantic knowledge, making analogies, learning new effective knowledge, and detecting, perceiving, and “feeling” emotions.

The major research scope areas in sentiment analysis are:

1. Spam Detection Sentiment Analysis
2. Sentiment Analysis on short Sentence like abbreviations
3. Improving sentiment word identification algorithm
4. Developing fully automatic analyzing tool
5. Effective Analysis of policy opinionated content
6. Successful handling of bi polar sentiments
7. Generation of highly content lexicon database

11. Bibliography

- ☐ Kaggle Dataset - https://www.kaggle.com/grikomsn/amazon-cell-phones-reviews
- ☐ International Journal of Computer Applications, Volume 125 – No.3, September 2015
  ISSN : 0975 – 8887
- ☐ International Journal of Advance Research in Science and Engineering, Volume 6 - Issue 7, July 2017
  ISSN(O) : 2319 - 8354
  ISSN(P) : 2319 - 8346
- ☐ https://www.lexalytics.com/applications
- ☐ Journal of Computer Science, Volume 12 - No.3, March 2016
  153 - 168
- ☐ Prominent Feature Extraction for Sentiment Analysis by Basant Agarwal and Namita Mittal, Page 89-92, January 2016
  ISSN : 2509-5706
- ☐ https://www.geeksforgeeks.org/html-tutorials/
- ☐ https://www.w3schools.com/
- ☐ https://smartbridge.teachable.com/courses/enrolled/846395
- ☐ Fourth International Conference on Recent Trends in Computer Science & Engineering
  Sentiment Analysis:A Comparative Study On Different Approaches - Page 44-49 , 2016
- ☐ https://google.com

12. Appendix

12.1 Model Code

Importing Libraries

```
[ ] import nltk
    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd
    import re
    from sklearn.feature_extraction.text import CountVectorizer
    from sklearn.model_selection import train_test_split
    from sklearn import preprocessing
    import keras
    from keras.models import Sequential
    from keras.layers import Dense
    from keras.models import load_model
    import pickle
```

```
Using TensorFlow backend.
```

Importing Dataset

```
[ ] dataset = pd.read_csv('20191226-reviews.csv')
    items = pd.read_csv('20191226-items.csv')
```

Fig 12.1.1 Importing Libraries & Dataset

```
[ ] dataset
```

	asin	name	rating	date	verified	title	body	helpfulVotes
0	B0000SX2UC	Janet	3	October 11, 2005	False	Def not best, but not worst	I had the Samsung A600 for awhile which is abs...	1.0
1	B0000SX2UC	Luke Wyatt	1	January 7, 2004	False	Text Messaging Doesn't Work	Due to a software issue between Nokia and Spri...	17.0
2	B0000SX2UC	Brooke	5	December 30, 2003	False	Love This Phone	This is a great, reliable phone. I also purcha...	5.0
3	B0000SX2UC	amy m. teague	3	March 18, 2004	False	Love the Phone, BUT...!	I love the phone and all, because I really did...	1.0
4	B0000SX2UC	tristazbimmer	4	August 28, 2005	False	Great phone service and options, lousy case!	The phone has been great for every purpose it ...	1.0
...	...	...	...	...	...	...	...	...
67981	B081H6STQQ	jande	5	August 16, 2019	False	Awesome Phone, but finger scanner is a big mis...	I love the camera on this phone. The screen is...	1.0
67982	B081H6STQQ	2cool4u	5	September 14, 2019	False	Simply Amazing!	I've been an Xperia user for several years and...	1.0
67983	B081H6STQQ	simon	5	July 14, 2019	False	great phon3, but many bugs need to fix. still ...	buy one more for my cousin	NaN

Fig 12.1.2 Dataset

```
[ ] dataset.drop(columns=['asin', 'name' , 'date' , 'verified' , 'helpfulVotes' ] , inplace = True)
```

```
dataset
```

	rating	title	body
0	3	Def not best, but not worst	I had the Samsung A600 for awhile which is abs...
1	1	Text Messaging Doesn't Work	Due to a software issue between Nokia and Spri...
2	5	Love This Phone	This is a great, reliable phone. I also purcha...
3	3	Love the Phone, BUT...!	I love the phone and all, because I really did...
4	4	Great phone service and options, lousy case!	The phone has been great for every purpose it ...
...	...	...	...
67981	5	Awesome Phone, but finger scanner is a big mis...	I love the camera on this phone. The screen is...
67982	5	Simply Amazing!	I've been an Xperia user for several years and...
67983	5	great phon3, but many bugs need to fix. still ...	buy one more for my cousin
67984	5	Phone is like new	Product looks and works like new. Very much re...
67985	5	Outstanding phone for the price	I love the size and style of this phone. It is...

67986 rows × 3 columns

Fig 12.1.3 Dataset after removing not required columns

```
[ ] dataset['sentiment'] = dataset['rating'].apply(lambda rating : +1 if rating > 3 else 0)
```

```
dataset
```

	rating	title	body	sentiment
0	3	Def not best, but not worst	I had the Samsung A600 for awhile which is abs...	0
1	1	Text Messaging Doesn't Work	Due to a software issue between Nokia and Spri...	0
2	5	Love This Phone	This is a great, reliable phone. I also purcha...	1
3	3	Love the Phone, BUT...!	I love the phone and all, because I really did...	0
4	4	Great phone service and options, lousy case!	The phone has been great for every purpose it ...	1
...	...	...	...	...
67981	5	Awesome Phone, but finger scanner is a big mis...	I love the camera on this phone. The screen is...	1
67982	5	Simply Amazing!	I've been an Xperia user for several years and...	1
67983	5	great phon3, but many bugs need to fix. still ...	buy one more for my cousin	1
67984	5	Phone is like new	Product looks and works like new. Very much re...	1
67985	5	Outstanding phone for the price	I love the size and style of this phone. It is...	1

67986 rows × 4 columns

Fig 12.1.4 Dataset with new column 'Sentiment'

```
[ ] dataset.isnull().any()

rating       False
title         True
body          True
sentiment    False
dtype: bool

[ ] dataset["title"].fillna(dataset["title"].mode()[0] , inplace = True)
    dataset["body"].fillna(dataset["body"].mode()[0] , inplace = True)

[ ] dataset.isnull().any()

rating       False
title        False
body         False
sentiment    False
dtype: bool
```

Fig 12.1.5 Checking for NaN values and replacing them

Text Cleaning or Preprocessing

```
[ ] nltk.download("stopwords")
    from nltk.corpus import stopwords
    from nltk.stem.porter import PorterStemmer
    ps = PorterStemmer()
    review = []

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

[ ] len(dataset)

67986
```

Fig 12.1.6 Text Cleaning or Preprocessing

Remove Punctuations & Numbers and Stemming

```
[ ] for i in range(len(dataset)):
        data = dataset["title"][i]
        data = re.sub('[^a-zA-Z]', ' ',data)
        data = data.lower()
        data = data.split()
        data = [ps.stem(word) for word in data if not word in set(stopwords.words('english'))]
        data = ' '.join(data)
        review.append(data)
```

Fig 12.1.7 Remove Punctuations & Numbers and Stemming

Creating Dependent Variables

```
[ ] cv = CountVectorizer(max_features = 3000)
    x = cv.fit_transform(review).toarray()
    y = dataset.iloc[:,3:4].values
```

Fig 12.1.8 Creating Dependent Variables

Splitting Data into Training and Test set

```
[ ] x_train,x_test,y_train,y_test = train_test_split(x,y,random_state = 0,test_size = 0.2)

[ ] x_train.shape
(54388, 3000)

[ ] x_test.shape
(13598, 3000)

[ ] y_train.shape
(54388, 1)

[ ] y_test.shape
(13598, 1)
```

Fig 12.1.9 Splitting Data into Training and Test Set

Initializing the model

```
[ ] model = Sequential()
```

Adding Input Layer

```
[ ] model.add(Dense(units = 3000, activation = "relu", kernel_initializer = "random_uniform"))
```

Adding Hidden Layer

```
[ ] model.add(Dense(units = 4500, activation = "relu", kernel_initializer = "random_uniform"))
```

Adding Output Layer

```
[ ] model.add(Dense(units = 1, activation = "sigmoid", kernel_initializer = "random_uniform"))
```

Fig 12.1.10 Initializing Model and Adding Layers

Configuring the learning process

```
[ ] model.compile(optimizer = "adam", loss = "binary_crossentropy", metrics = ["accuracy"])
```

: Training the model

```
[ ] model.fit(x_train, y_train, epochs = 20, batch_size = 32)
```

```
Epoch 1/20
54388/54388 [==============================] - 306s 6ms/step - loss: 0.3439 - accuracy: 0.8498
Epoch 2/20
54388/54388 [==============================] - 308s 6ms/step - loss: 0.2664 - accuracy: 0.8880
Epoch 3/20
54388/54388 [==============================] - 308s 6ms/step - loss: 0.2026 - accuracy: 0.9189
Epoch 4/20
54388/54388 [==============================] - 310s 6ms/step - loss: 0.1590 - accuracy: 0.9369
Epoch 5/20
54388/54388 [==============================] - 306s 6ms/step - loss: 0.1366 - accuracy: 0.9472
Epoch 6/20
54388/54388 [==============================] - 305s 6ms/step - loss: 0.1266 - accuracy: 0.9504
Epoch 7/20
54388/54388 [==============================] - 306s 6ms/step - loss: 0.1206 - accuracy: 0.9522
Epoch 8/20
54388/54388 [==============================] - 307s 6ms/step - loss: 0.1164 - accuracy: 0.9525
Epoch 9/20
54388/54388 [==============================] - 304s 6ms/step - loss: 0.1135 - accuracy: 0.9544
```

Fig 12.1.11 Configuring Learning Process and Training the Model

```
[ ] Epoch 10/20
54388/54388 [==============================] - 307s 6ms/step - loss: 0.1115 - accuracy: 0.9552
Epoch 11/20
54388/54388 [==============================] - 306s 6ms/step - loss: 0.1078 - accuracy: 0.9554
Epoch 12/20
54388/54388 [==============================] - 308s 6ms/step - loss: 0.1079 - accuracy: 0.9557
Epoch 13/20
54388/54388 [==============================] - 306s 6ms/step - loss: 0.1084 - accuracy: 0.9561
Epoch 14/20
54388/54388 [==============================] - 307s 6ms/step - loss: 0.1066 - accuracy: 0.9563
Epoch 15/20
54388/54388 [==============================] - 306s 6ms/step - loss: 0.1060 - accuracy: 0.9561
Epoch 16/20
54388/54388 [==============================] - 307s 6ms/step - loss: 0.1040 - accuracy: 0.9566
Epoch 17/20
54388/54388 [==============================] - 306s 6ms/step - loss: 0.1029 - accuracy: 0.9571
Epoch 18/20
54388/54388 [==============================] - 306s 6ms/step - loss: 0.1041 - accuracy: 0.9568
Epoch 19/20
54388/54388 [==============================] - 306s 6ms/step - loss: 0.1069 - accuracy: 0.9565
Epoch 20/20
54388/54388 [==============================] - 309s 6ms/step - loss: 0.1023 - accuracy: 0.9572
<keras.callbacks.callbacks.History at 0x7ff55709b9e8>
```

Saving Model

```
[ ] model.save('project.h5')
```

Fig 12.1.12 Training and Saving Model

Prediction

```
[ ] model = load_model('project.h5')
```

```
[ ] with open('CountVectorizer','wb') as file:
        pickle.dump(cv,file)
```

```
[ ] entered_input = "It is a very bad product"
```

```
[ ] x_intent = cv.transform([entered_input])
    y_pred = model.predict(x_intent)
    if (y_pred>0.5):
      print("It is a positive review")
    else:
      print("It is a negative review")

    It is a negative review
```

Fig 12.1.13 Prediction

```
[ ] entered_input = "It is a bad product"
```

```
[ ] x_intent = cv.transform([entered_input])
    y_pred = model.predict(x_intent)
    if (y_pred>0.5):
      print("It is a positive review")
    else:
      print("It is a negative review")

    It is a negative review
```

```
[ ] entered_input = "It is a very good product"
```

```
x_intent = cv.transform([entered_input])
y_pred = model.predict(x_intent)
if (y_pred>0.5):
  print("It is a positive review")
else:
  print("It is a negative review")

It is a positive review
```

Fig 12.1.14 Prediction Examples

12.2 App.py Code

```python
from flask import Flask, request, render_template
from keras.models import load_model
import pickle
global model, graph
import tensorflow as tf
graph = tf.get_default_graph()

with open(r'CountVectorizer','rb') as file:
    cv=pickle.load(file)

model = load_model('mymodel.h5')
app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/analyse', methods = ['POST'])
def review():
    input_text = request.form['a']

    with graph.as_default():
        x_intent = cv.transform([input_text])
        y_pred = model.predict(x_intent)
        out = ''
        if(y_pred>0.5):
            out = "Positive Review"
        else:
            out = "Negative Review"

        return render_template('index.html', review=out)
```

Fig 12.2.1 App.py Python Code

12.3 HTML Code

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>Amazon Cell Phone Review</title>

</head>

<body background="https://wallpapercave.com/wp/pTP5LzA.jpg" align="center">

    <h1>Analysis of Amazon Cell Phone Review</h1>
    <hr style="width:550px">
    <br/>
    <form action="http://localhost:5000/analyse" method='POST'>
        <p><input type="text" height="100" name="a" style="width:250px;height:100px" /></p>
        <p><input type="submit" value="Submit" style="width:80px;height:35px" /></p>
        <br/>
    </form>
    <h2><b>{{review}}</b></h2>

</body>

</html>
```

Fig 12.3.1 index.html Code

12.4 UI Webpage

Fig 12.4.1 UI Webpage