**Amazon Kindle Store Reviews Analysis Using IBM Watson Services**

A MINI PROJECT REPORT

Submitted to

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD**

In partial fulfillment of the requirements for the award of the degree of

### BACHELOR OF TECHNOLOGY

### IN

### COMPUTER SCIENCE AND ENGINEERING

Submitted by

### AITHA SAI PRIYA 18UK1A0501

**SRAVANI KARNE 18UK1A0553**

### GUDIKANDULA BHARATHCHANDRA 18UK1A0574

**DONGALA ANJIREDDY 18UK1A0514**

Under the esteemed guidance of

**Dr.P.MAHIPAL REDDY**

(Assistant Professor)

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTU Hyderabad & Approved by AICTE, New Delhi) Bollikunta , Warangal – 506005

**2018-2022**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTU Hyderabad & Approved by AICTE, New Delhi)

Bollikunta , Warangal – 506005

**2018 – 2022**

**CERTIFICATE**

This is to certify that the Mini Project Report entitled **“Amazon Kindle Store Reviews Analysis Using IBM Watson Services”** is being submitted by ***A.SAIPRIYA(H.NO:18UK1A0501),K.SRAVANI(H.NO:18UK1A053),G.BHARATHCHANDRA(H.NO:18UK1A0574),D.ANJIREDDY(H.NO:18UK1A0514)*** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2021-22, is a record of work carried out by them under the guidance and supervision.

#### Dr.B.MAHIPAL REDDY Dr. R. NAVEEN KUMAR

#### Project Guide Head of the Department

#### EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this mini project in the institute.

We extend our heartfelt thanks to **Dr. R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the mini project.

We express heartfelt thanks to **Mr. SHIVAM**, Program Manager, SmartBridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the mini project and for their support in completing the mini project, mini project and internship.

We express heartfelt thanks to the guide, **Dr.P. MAHIPAL REDDY** Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion of this mini project.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the this.

**INDEX**

**TITLE Page .No**

**CHAPTER 1: INTRODUCTION**

* 1. Introduction 01
  2. Overview

1.3 Purpose

**CHAPTER 2: LITERATURE SURVEY**

* 1. Existing problem 02

2.2 Proposed solution

**CHAPTER 3: THEORTICAL ANALYSIS**

3.1 Block diagram 03

3.2 Hardware/Software designing

**CHAPTER 4: EXPERIMENTAL INVESTIGATIONS** 04

4.1 Methods And Implementation 05

**CHAPTER 5: FLOW CHART** 14  **CHAPTER 6: RESULTS** 15

**CHAPTER 7: ADVANTAGES AND DISADVANTAGES** 17

**CHAPTER 8: APPLICATIONS**  18

**CHAPTER 9: CONCLUSION**  20

**CHAPTER 10: FUTURE SCOPE** 21

**CHAPTER 11: BIBILOGRAPHY** 24

**APPENDIX**  25

### ABSTRACT

The world we see nowadays is becoming more digitized. In this digitalized world e-commerce is taking the ascendancy by making products available within the reach of customers where the customer doesn’t have to go out of their house. Nowadays people are relying ononline products so the importance ofa reviewis going higher. For selecting a product, a customer needs to go through thousands of reviews to understand a product. But in this prosperous day of deep learning, going through thousands of reviews would be much easier if a model is used to polarize those reviews and learn from it. We used deep learning techniques on a large scale amazon dataset to polarize it and get satisfactory accuracy.

### Keywords—e-commerce, deep learning, feature extraction, text classification,sentiment analysis

**1.INTRODUCTION**

**1.1 Introduction**

Amazon Kindle Store is an e-book e-commerce store for all book reading hobbyists. Online reviews are a category of product information created by the users based on personal handling experience. Online shopping websites endow with platforms for consumers to review products and carve up opinions. Sentiment analysis or opinion mining is nothing but classification of emotions in the reviews text into positive, negative and neutral. Amazon Kindle Store is an e-book e-commerce store for all book reading hobbyists. Online reviews are a category of product information created by the users based on personal handling experience. Online shopping websites endow with platforms for consumers to review products and carve up opinions. Sentiment analysis or opinion mining is nothing but classification of emotions in the reviews text into positive, negative and neutral

**1.2 Overview**

With everything shifting online, reviews have started giving utmost importance to Sentiment Analysis. Honestly, it’s their only gateway to thoroughly understanding their customer-base, including their expectations from the customer review. Social Media listening can help organisations from any domain understand the grievances and concerns of their customers – which eventually helps the organisations scale up their services. Sentiment Analysis helps brands tackle the exact problems or concerns of their customers. Sentiment analysis gives an organisation the much-needed insights on their customers. Organisations can now adjust their marketing strategies depending on how the customers are responding to it.

Sentiment Analysis also helps organisations measure the ROI of their marketing campaigns and improve their customer service. Since sentiment analysis gives the organisations a sneak peek into their customer’s emotions, they can be aware of any crisis that’s to come well in time – and manage it accordingly.



**1.3 Purpose**

Our aim for the project is to make use of Deep learning using python and extract necessary libraries from it for the sentiment analysis of the amazon kindle store reviews. The reviews dataset from the amazon kindle is to be pre-processed first by removing the punctuation marks and by applying the tokenization and attention masking step. This step is necessary to make the machine familiarise with the input text. The model is then trained under these texts and the model is built using the Convolutional neural networks. The main idea behind the project is to demonstrate the use of sentiment analysis for the customer feedback on each book from amazon kindle store.

**2.LITERATURE SURVEY**

Deep learning is a field of machine learning concerned with algorithms inspired by the structure of the brain called artificial neural networks. It has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

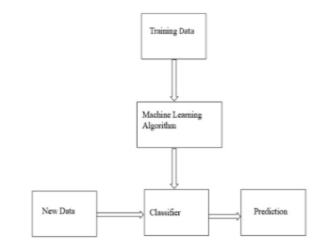
Each layer of its neural network builds on its previous layer with added data like a retailer, sender, user, social media event, credit score, IP address, and a host of other features that may take years to connect together if processed by a human being. Deep learning algorithms are trained to not just create patterns from all transactions, but also know when a pattern is signaling the need for a fraudulent investigation. The final layer relays a signal to an analyst who may freeze the user’s account until all pending investigations are finalized.

Deep learning is used across all industries for a number of different tasks. Commercial apps that use image recognition, open-source platforms with consumer recommendation apps, and medical research tools that explore the possibility of reusing drugs for new ailments are a few of the examples of deep learning incorporation.

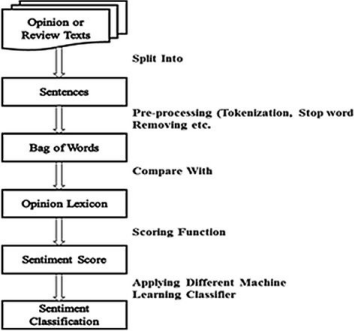
**3.THEORETICAL ANALYSIS**

While selecting the algorithm that gives an accurate prediction we went through a lot of algorithms which gave the results abruptly accurately and from them we selected only one algorithm for the prediction problem that is convolutional neural network.

**3.1 Block Diagram**

****

**Fig: Outline of Classification in Machine Learning**

****

**Fig: Block diagram of REVIEW SENTIMENT ANALYSIS**

**3.2 Software Designing**

● Jupyter Notebook Environment

● Spyder Ide

● Machine Learning Algorithms

● Python (pandas, numpy,keras,Tensorflow,CNN layers)

● HTML

● Flask

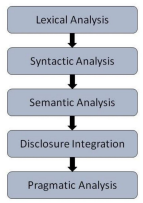
We developed this loan status prediction by using the Python language which is an interpreted and high level programming language and using the Machine Learning algorithms. For coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientic programming in the python language. For creating a user interface for the prediction we used Flask. It is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a web page is HTML by creating the templates to use in the functions of the Flask and HTML.

**4. EXPERIMENTAL INVESTIGATIONS**

In this paper, the dataset we used for the purpose of doing sentiment analysis is derived from https://www.kaggle.com/bharadwaj6/kindle-reviews . In this we have rows and columns.It contains the books names and the respective review for all the books and also the other information related to the book and also the user. To review the comments we make use of the Natural Language Processing (NLP) algorithm it is a Deep learning algorithm which can process the sentences and remove the all punctuations and other string processing is done by this algorithm.

**Steps in NLP:**

There are five general steps.



● **Lexical Analysis** − It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of text into paragraphs, sentences, and words.

● **Syntactic Analysis (Parsing)** − It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as “The school goes to boy” is rejected by English syntactic analyzers.

● **Semantic Analysis** − It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentences such as “hot ice-cream”.

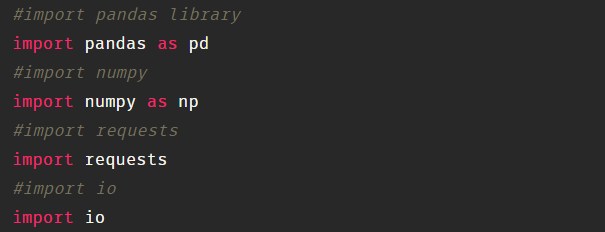
● **Discourse Integration** − The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentences.

● **Pragmatic Analysis** − During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

**4.1 Methods And Implementation**

The first step is usually importing the libraries that will be needed in the program.

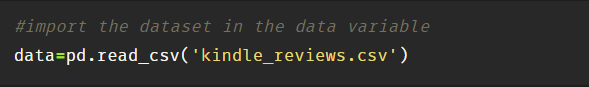
Import numpy and pandas to your Python script:



### Read The Dataset

You might have your data in .csv files; excel files or .test files or something else. But the goal is the same in all cases. If you want to analyses that data using pandas, the first step will be to read it into a data structure that’s compatible with pandas.

Let’s load a .csv data file into pandas. There is a function for it, called read\_csv(). We will need to locate the directory of the CSV file at first (it’s more efficient to keep the dataset in the same directory as your program).



Dataset consists of 10 columns (982619 rows × 10 columns).

Columns are:

1. Unnamed: 0
2. asin - ID of the product, like B000FA64PK
3. helpful - helpfulness rating of the review - example: 2/3.
4. overall - rating of the product.
5. reviewText - text of the review (heading).
6. reviewTime - time of the review (raw).
7. reviewerID - ID of the reviewer, like A3SPTOKDG7WBLN
8. reviewerName - name of the reviewer.
9. summary - summary of the review (description).
10. UnixReviewTime - unix timestamp.

To check the first five rows of the dataset, we have a function call **head( ).**

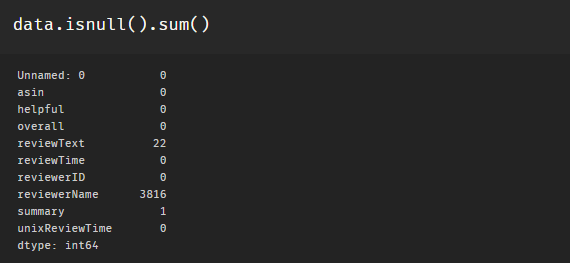


This head ( ) function returns the first 5 rows for the object based on position. It is useful for quickly testing if your object has the right type of data in it.

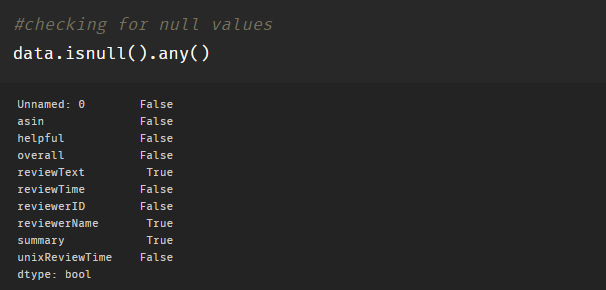
### Checking For Null Values Or Taking Care Of Missing Data

Sometimes you may find some data are missing in the dataset. We need to be equipped to handle the problem when we come across them. One of the most common ideas to handle the problem is to take a mean of all the values of the same column and have it to replace the missing data.

We will be using **isnull().sum()** method to see which column has missing values.**isnull().sum()** will give you the total count of null values present in each column.



We can also check the null values in dataset using isnull ().any (). By using **isnull ().any (),** a column is checked for NULL values and a Boolean series is returned by the isnull () method which stores True for every NaN value and False for a Not null value.

****

Word “True” refers to that a column has missing values i.e., in our dataset Price column has missing values. We can replace the missing values by mean, median or mode by using fillna method.

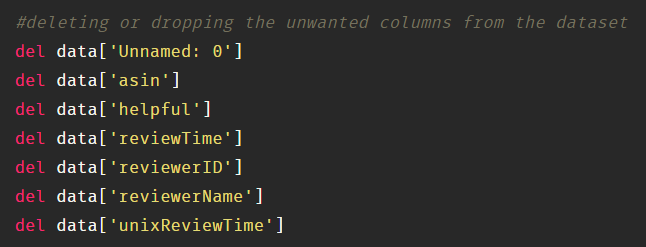
In our dataset reviewText, reviewName, Summary, to handle these null values we are going to use a method called **dropna( ).**

The **dropna()** function is used to remove/drop null values/missing values in dataset either in row or column.

### Delete Or Drop The Columns

### Deletion or dropping is one of the primary operations when it comes to data analysis. Very often we see that a particular attribute in the data frame is not at all useful for us while working on a specific analysis(example: name, ID etc), rather having it may lead to problems and unnecessary change in the prediction.

del is used to delete a specific column by referencing the column name as shown below.

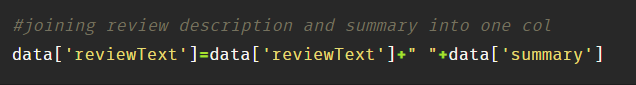


Deleting ‘Unnamed: 0’, 'asin', reviewTime’, ‘reviewerID’, helpful', ‘unixReviewTime’ columns from the dataset. To drop it the syntax looks like. Here drop is a function; we need to pass the columns names inside the function. And the axis=1 represents the Colum wise operation.

### Joining Review Description And Summary Into One Column.

In our dataset, we have **reviewText** and **summary** columns.We concatenating these 2 columns and place them in reviewText as a single column.

So that we can proceed for processing of the main textual information.



### Now dropping a summary column from data.

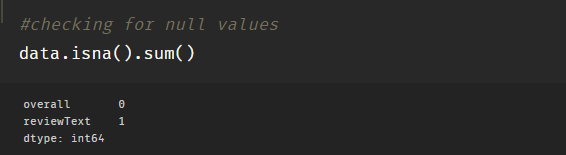


### Here axis refers to the dimension of the array, in this case, axis=0 is the dimension that points to the row-wise operation and axis=1 is the dimension that points to the column side.

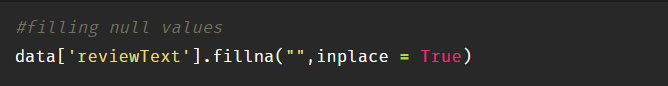
When inplace = True, the data is modified in place, which means it will return nothing and the dataframe is now updated with the column been removed

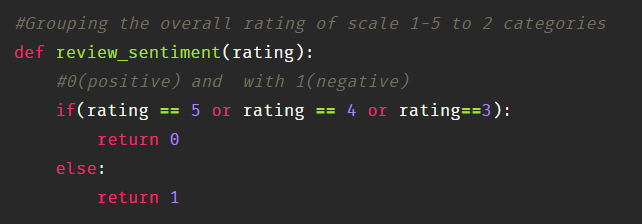
When inplace = False, which is the default, then the operation is performed and it returns a copy of the object.

Checking for null values now for both ReviewText and overall columns



Since there is only one null value, replace it with blank space.



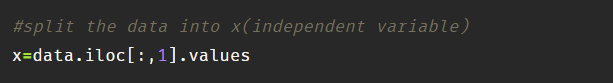


### Split The Dataset Into Dependent And Independent Variable

The concept of the dependent variable (y) and independent variables(x) is important to understand. Here, Dependent variable is nothing but output in the dataset and the independent variable is all inputs in the dataset. We can denote with any symbol (alphabets). In our dataset, if you look closely, the one column (ReviewText) determines the outcome of the first column (Overall). So, we can say that Overall is the dependent variable, the value of which is determined by the ReviewText.

To read the columns, we will use iloc of pandas (used to fix the indexes for selection) which takes two parameters — [row selection, column selection].

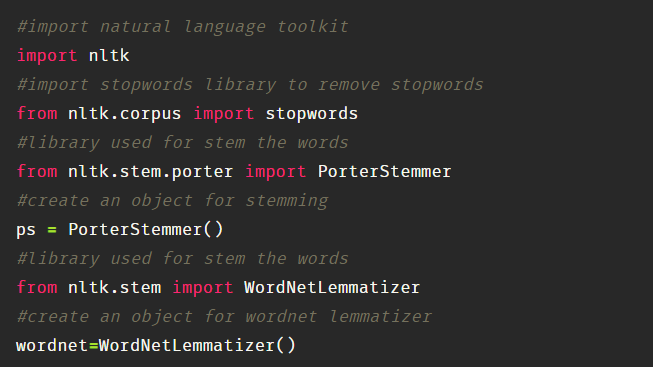
Let’s split our dataset into independent and dependent variables.



### Processing Textual Data

We will be using different libraries for the steps that are included in this step. **“Re”** is the library which is used to replace the selected special characters with the desired parameter. **“NLTK”** – Natural language Tool Kit is the library used for stemming using a special class in the library.

**Import required Libraries**

****

**Remove Punctuations and Numbers**

We will begin by removing punctuation, number, converting each word into lower case, taking root of the words and removing the stopwords.

Remove Punctuations, Numbers Punctuations, Numbers doesn’t help much in processing the given text, so we will be using re library to replace all the punctuations numbers with space while excluding alphabets. Now we are declaring a variable called temp and assigning the independent variable. Then using re library we are substituting all the other special characters with a space excluding alphabets ([^a-zA-Z], this indicates except this replace everything with space).

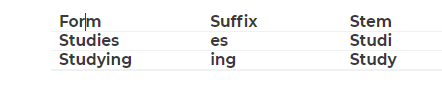
**Convert each word into its lower case**

Every word should be lowercased to have uniformity in the data. For example, it useless to have some words in different cases (eg ‘good’ and ‘GOOD’).

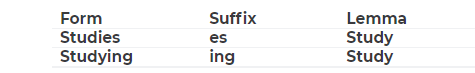
**Stemming or Lemmatization**

Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word.

For example, we may have a suffix rule that, based on a list of known suffixes, cuts them off. In the English language, we have suffixes like “-ed” and “-ing” which may be useful to cut off in order to map the words “cook,” “cooking,” and “cooked” all to the same stem of “cook.”

****

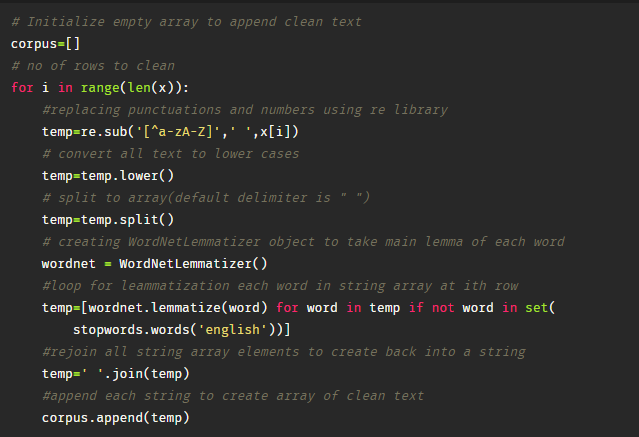
Lemmatization takes into consideration the morphological analysis of the words. To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma. Again, you can see how it works with the same example words.



To lemmatize each word first we have to split the review into a list and then apply lemmatization functionality. We use WordNetLemmatizer for Lemmatization of the word purpose.

And also we need to remove stopwords. **Stopwords** are the English word which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For example, the words like the, he, have, that, what etc.

**The whole code looks like.**

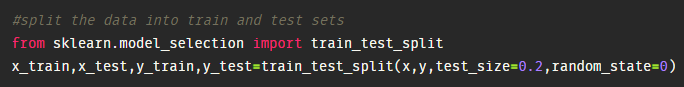
****

### Split The Dataset Into Train Set And Test Set

Scikit library provides a tool, called the Model Selection library. There is a class in the library which is, ‘train\_test\_split.’ Using this we can easily split the dataset into the training and the testing datasets in various proportions.

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

* Train Dataset: Used to fit the machine learning model.
* Test Dataset: Used to evaluate the fit machine learning model.



X is the bag of words; y is 0 or 1 (positive or negative).

In general, you can allocate 80% of the dataset to the training set and the remaining 20% to test set. We will create 4 sets— X\_train (training part of the matrix of features), X\_test (test part of the matrix of features), Y\_train (training part of the dependent variables associated with the X train sets, and therefore also the same indices), Y\_test (test part of the dependent variables associated with the X test sets, and therefore also the same indices.

There are a few other parameters that we need to understand before we use the class:

test\_size — this parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset.

train\_size — you have to specify this parameter only if you’re not specifying the test\_size. This is the same as test\_size, but instead, you can give the class what percent of the dataset you want to split as the training set.

random\_state — here you pass an integer, which will act as the seed for the random number generator during the split. Or, you can also pass an instance of the Random\_state class, which will become the number generator. If you don’t pass anything, the Random\_state instance used by np.random will be used instead.

### Model Building

The neural network model is to be built by adding different network layers

In this milestone, we start building our model by:

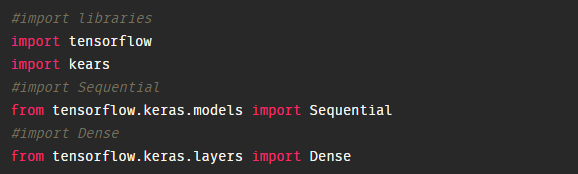
1.Initializing the mode

2.Adding Input layers

3.Adding Dense layers

4.Adding Output layers

### Import Required Libraries



### Initializing The Model

Keras has 2 ways to define a neural network:

Sequential

Function API

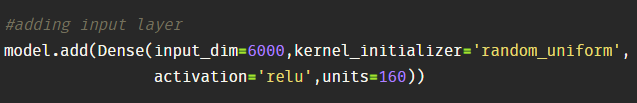
The Sequential class is used to define linear initializations of network layers which then, collectively, constitute a model. In our example below, we will use the Sequential constructor to create a model, which will then have layers added to it using the add () method.



### Add Input Layers

This layer accepts input features. No computation is performed at this layer, nodes here just pass on the information (features) to the hidden layer.

This step is to add a dense layer (input layer) where you will be specifying the number of inputs to the neural network, activation function and weights initializer and number of connection to the hidden layer as the arguments. We use add () method to add dense layers.



In the above code, input\_dim is nothing no of max\_features that were considered in the bag of words. In our project we considered 6000 features as X, so we need to give 6000 features as input\_dim.

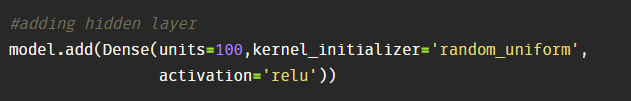
**Kernel\_initializer** is the weight initialization; initialization function is the network initialization function which sets all the weights and biases of a network to values suitable as a starting point for training. While model training the weights will be updated with the help of arguments that we passed. In this scenario, we are using initialization of weights randomly.

**Activation** function defines the output of input or set of inputs or in other terms defines node of the output of node that is given in inputs. They basically decide to deactivate neurons or activate them to get the desired output. It also performs a nonlinear transformation on the input to get better results on a complex **neural network**. This layer accepts input features. It provides information from the outside world to the network, no computation is performed at this layer, nodes here just pass on the information (features) to the hidden layer.

**Units,** which is the number of nodes in the hidden layer. You can determine the most appropriate number through experimentation. The higher the number of dimensions the more computing resources you will need to fit the model. In our project, I’ve considered the 160 neurons in this hidden layer.

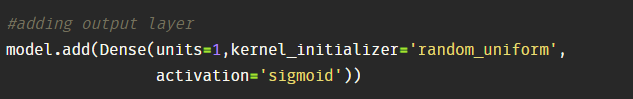
### Adding Hidden Layers

This step is to add a dense layer (Hidden layer) where you will be specifying the number of neurons to the next layer, activation function and weight initializer as the arguments. We use add ( ) method to add dense layers. In this layer, no need of mentioning input dimensions as we have mentions them in the above layer itself.



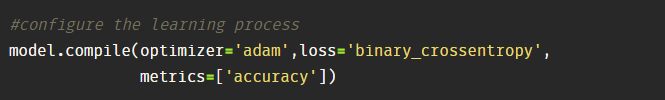
### Adding Output Layer

This step is to add a dense layer (output layer) where you will be specifying the number of classes (dependent variable), activation function and weight initializer as the arguments. We use add () method to add dense layers. In this layer, there is no need of mentioning input dimensions as we have mentions them in the above layer itself.



### Configuring The Learning Process

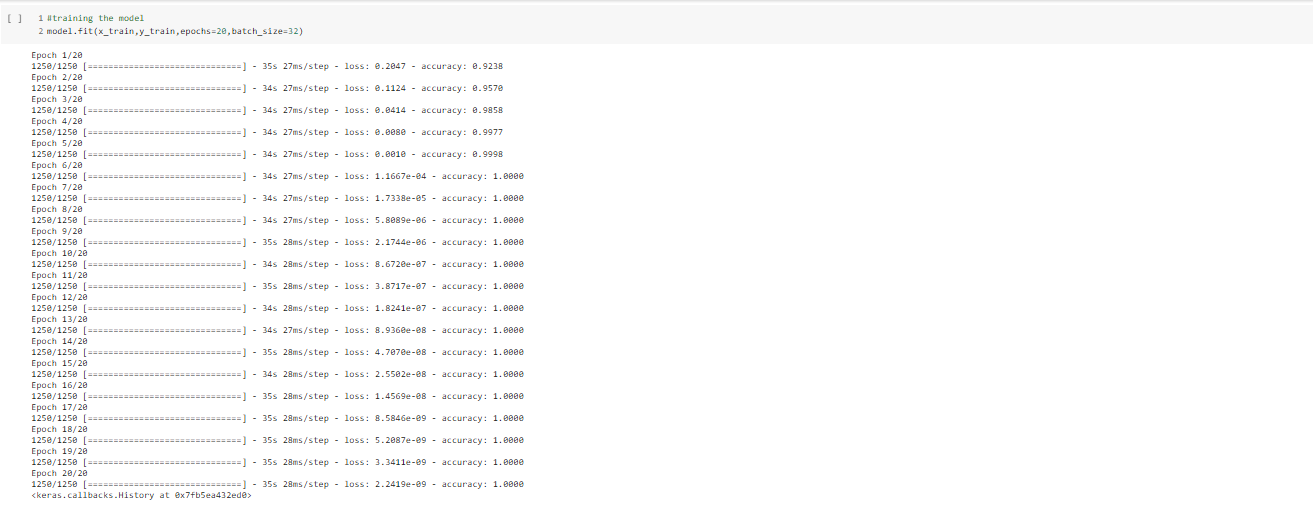
With both the training data defined and model defined, it's time to configure the learning process. This is accomplished with a call to the compile() method of the Sequential model class. Compilation requires 3 arguments: an optimizer, a loss function, and a list of metrics.



### 

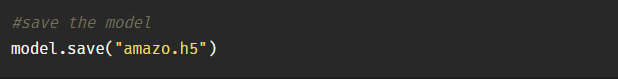
### Training The Model

At this point, we have training data and a fully configured neural network to train with said data. Training begins by calling the fit() method. The arguments are the batch size as you are using “adam” (bath gradient descent and epochs: no: of times the model should get trained).



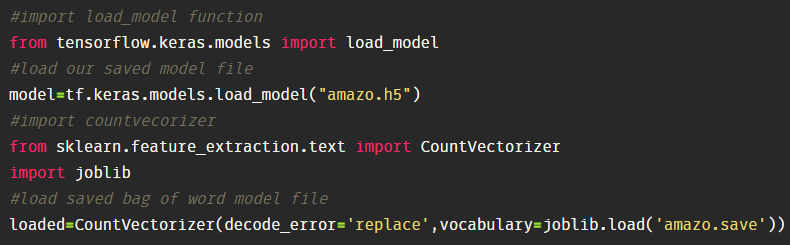
### Save The Model

Your model is to be saved for future purpose. This saved model also is integrated with an android application or web application in order to predict something.

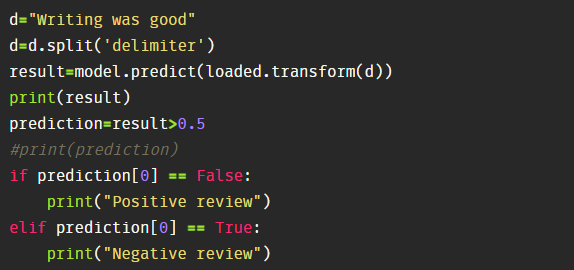


### Predictions

Import required libraries and load our saved model files



Enter input text or review and get predictions.



### Application Building

In this section, we will be building a web application that is integrated into the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server-side script

### 

### Create An HTML File

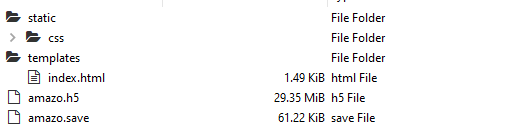
Flask Frame Work with Machine Learning Model In this section, we will be building a web application which is integrated into the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

And this flask application is building using spyder IDLE. Open spyder from the start menu.

Previously we are saved our bag of word file in save format that is **“amazo.save”.** We have also saved our Keras trained model with h5 extension is that **“amazo.h5”.**

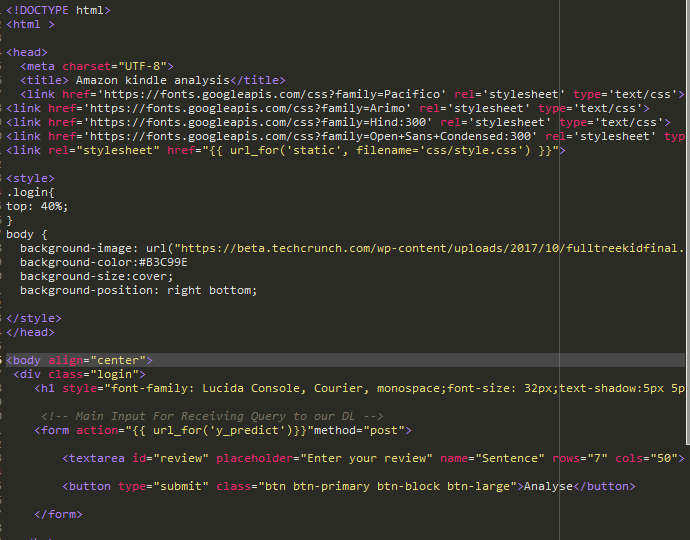
To build a web application you should know basics of “HTML, CSS, Bootstrap, flask framework and python” Create a project folder which should contains

* **A python file called app.py.**
* **Bag of word or Count Vector file (amazo.save).**
* **Trained model file (amazo.h5)**
* **Templates folder which contains index.HTML file.**
* **Static folder which contains styles.css.**

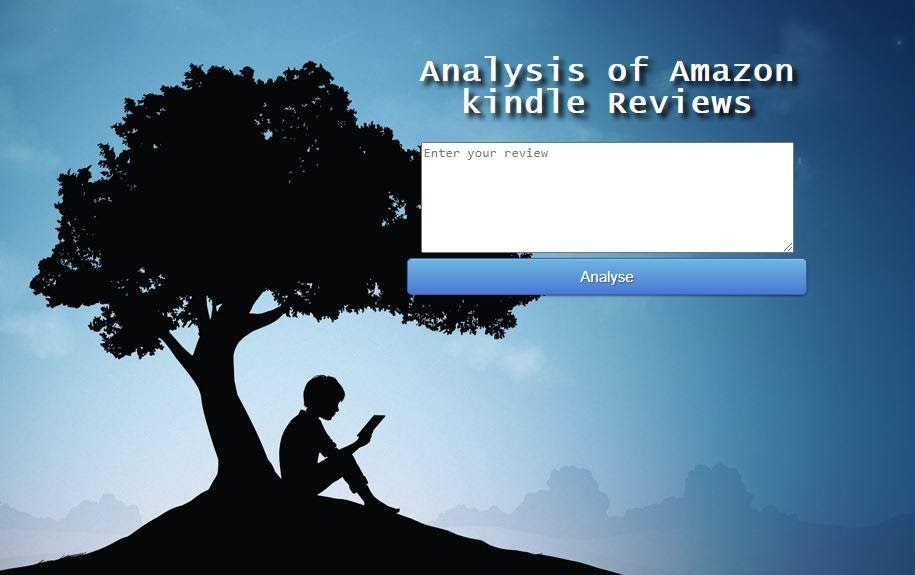
****

**Step 1: Building an Index. Html file**

This is the basic HTML page for our Project. H1 tag is used to give heading to the project. As I have mentioned there, only one input in our project is that review, based on the reviews we have to analyse the sentiment whether it is positive or negative review. So we have created 1 text input field in the html page. A button is used to send these values to the model files; this functionality will be written in the python file app.py. The model predicts the value and is displayed on the {{ prediction\_text}} field.



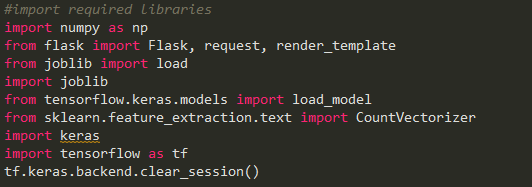
The HTML page Looks like



### Build Python Code

We will be using python for server-side scripting. Let’s see step by step process for writing backend code.

**Importing Libraries**

****

**Loading Saved model files**

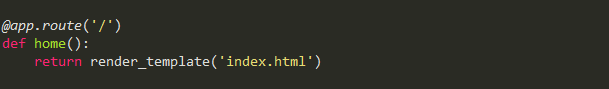
****

Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as argument Pickle library to load the model file.



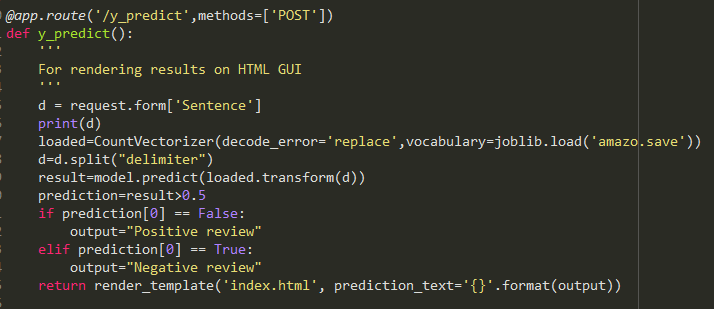
**Routing to the HTML Page**

Here we will be using declared constructor to route to the HTML page which we have created earlier.



In the above example, ‘/’ URL is bound with index.html function. Hence, when the home page of the web server is opened in the browser, the HTML page will be rendered. Whenever you enter the values from the HTML page the values can be retrieved using **POST** Method.

**Showcasing prediction on UI**

****

Here we are routing our app to y\_predict () function. This function retrieves all the values from the HTML page using Post request. We are requesting a tweet from the index.html using request function. That is stored in NumPy array. This array is passed to cla.predict function (). This function returns the prediction. And this prediction value is rendered to the text that we have mentioned in the index.html page earlier.

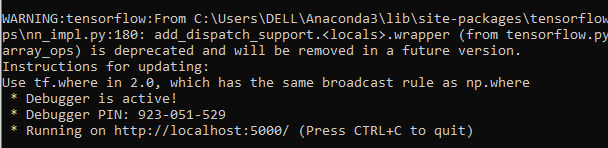
**Main Function**

This is used to run the application in localhost.



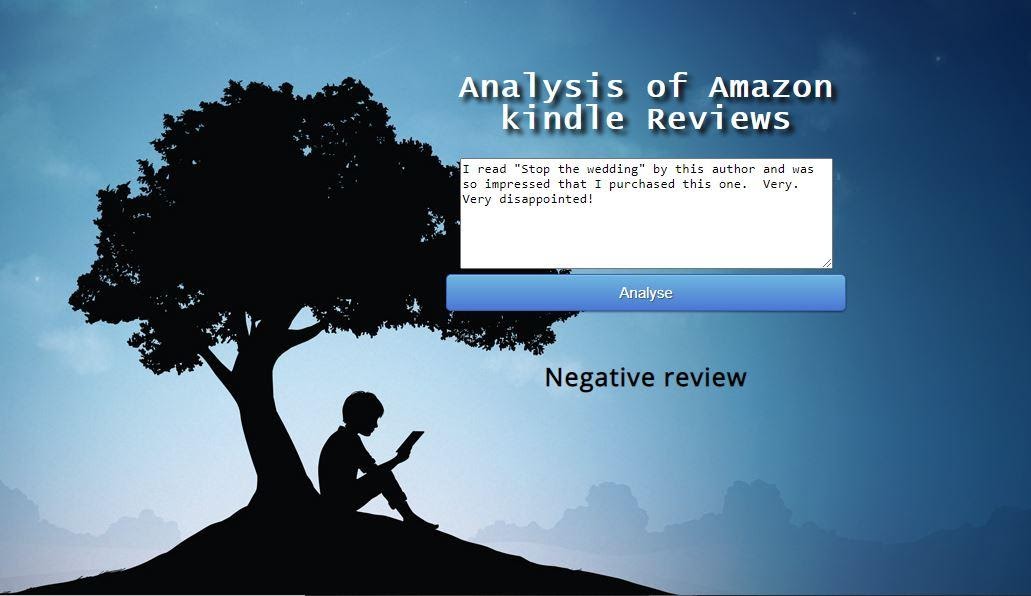
### How To Run The App.Py

* Open anaconda prompt from the start menu.
* Navigate to the folder where your app.py resides.
* Now type “python app.py” command.
* It will show the local host where your app is running on **http://127.0.0.1.5000/**
* Copy that localhost URL and open that URL in a browser. It does navigate you to them where you can view your web page.
* Enter the values, click on predict button and see the result/prediction on the web page.

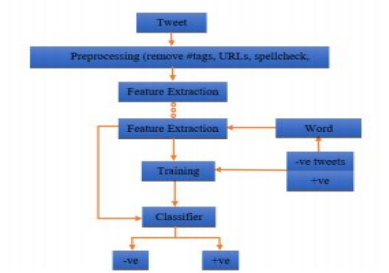


**output will look like**

****

****

**5.FLOWCHART**

****

**6.RESULT**

The following are the snapshots of the resulting model proposed by the NLP to do the sentiment analysis on the reviews directed from the amazon kindle book store.

**Fig1: Home page**

****

This is the home page. The user will enter the command which he wants to analyze

**Fig2:**

****

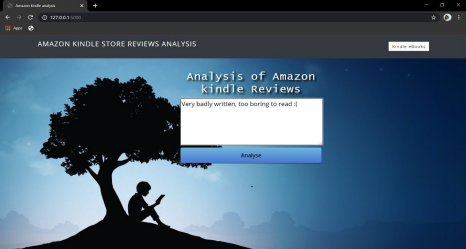
The user enters the review in the review box and presses the analyze button to know the sentiment of the review.

**Fig3:**

****

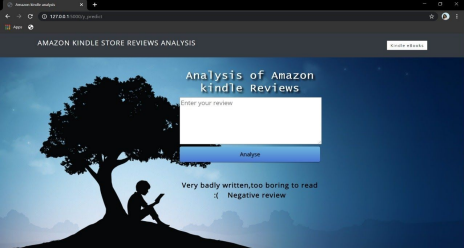
The user gets the output as the **Positive review**.

**Fig4:**

****

The user entered another review in the review box and clicked on analyse button He will be redirected for the output page I.e fig 5.

**Fig5:**

****

The user gets the output as a **Negative review**.

**7. ADVANTAGES AND DISADVANTAGES**

**ADVANTAGES:**

● Happy customers are more likely to be receptive to upselling. With sentiment analysis, you can easily identify your happiest customers. This helps you recognise chatters who might be receptive to spending more, as well as avoiding upsetting disgruntled customers with any unwelcome sales pitches.

● Users can ask comments about any book and get the direct response within seconds.

● The NLP system provides answers to sentences or questions in natural language.

● This system often has an exact answer to the questions. No unnecessary or unwanted information.

● The accuracy of the answers increases with the amount of relevant information provided in the question.

● Rapid prototyping and testing.

● Robust system in the NLP will always produce outputs, regardless of the image entered.

● It produces output in relation to the data entered.

**DISADVANTAGES:**

● Long time to implement and test a prototype.

● Manual system creation time-consuming and expensive creation process

**8. APPLICATION**

* Text-based application searching that the comment is positive or negative. extracting the emotion(positive or negative) of the given sentence by the users.
* Dialogue based application answering system.

**9.CONCLUSION**

It is completely impossible to use only raw text as input for making predictions. Hence, we saw that pe-processing step played a major role in the complete process of NLP. To get better results, accuracy and make the machine take all the text as tokens, pre-processing of data is to be done carefully looking at the type of contents present in it.

The most important thing is to be able to extract the relevant features from the given source of data. This kind of data can often come as a good complementary source in order to extract more learning features and increase the predictive power of the models. And the user is able to predict that the given comment is positive or negative.

**10. FUTURE SCOPE**

In future, the work can be extended to perform multi-class classification of reviews which will provide a delineated nature of review to the consumer, hence better judgment of the product. It can also be used to predict the rating of a product from the review. This will provide users with a reliable rating because sometimes the rating received by the product and the sentiment of the review do not provide justice to each other. The proposed extension of work will be very beneficial for the e-commerce industry as it will augment user satisfaction and trust.

**11. BIBLIOGRAPHY**

1. Gonzalez-de Santos P, et al. Fleets of robots for environmentally-safe pest control in agriculture. Precis. Agric. 2017;18:574–614. doi: 10.1007/s11119-016-9476-3.

2. Fernández-Quintanilla C, et al. Is the current state of the art of weed monitoring suitable for site-specific weed management in arable crops? Weed Res. 2018;58:259–272. doi: 10.1111/wre.12307.

3. Commonwealth of Australia. Agricultural competitiveness white paper. ISBN: 978-1-925237-73-3 (2015).

4. Slaughter DC, Giles DK, Downey D. Autonomous robotic weed control systems: A review. Comput. Electron. Agric. 2008;61:63–78. doi: 10.1016/j.compag.2007.05.008.

5. Shaner DL, Beckie HJ. The future for weed control and technology. Pest Manag. Sci. 2014;70:1329–1339. doi: 10.1002/ps.3706.

6. Bakhshipour A, Jafari A. Evaluation of support vector machines and artificial neural networks in weed detection using shape features. Comput. Electron. Agric. 2018;145:153–160. doi: 10.1016/j.compag.2017.12.032. [CrossRef] [Google Scholar]

7. dos Santos Ferreira A, Freitas DM, da Silva GG, Pistori H, Folhes MT. Weed detection in soybean crops using ConvNets. Comput. Electron. Agric. 2017;143:314–324. doi: 10.1016/j.compag.2017.10.027. [CrossRef] [Google Scholar]

8. Dyrmann M, Jørgensen RN, Midtiby HS. RoboWeedSupport - Detection of weed

locations in leaf occluded cereal crops using a fully convolutional neural network. Adv. Animal Biosci. 2017;8:842–847. doi: 10.1017/S2040470017000206. [CrossRef] [Google Scholar]

9. Wu, S. G. *et al*. A leaf recognition algorithm for plant classification using a probabilistic neural network. In *Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 11–16 (Cairo, Egypt, 2007).

10. Kumar, N. *et al*. Leafsnap: A computer vision system for automatic plant species identification. In *Proceedings of the 2012 European Conference on Computer Vision (ECCV)*, 502–516 (Berlin, Heidelberg, 2012).

**APPENDIX**

**SOURCE CODE:** Github

**HTML**

**Base.html:**

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0"> <meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Amazon kindle store review analysis</title>

<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script> <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script> <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script> <link href="{{ url\_for('static', filename='css/main.css') }}" rel="stylesheet">

<style>

body {

background-image: url("https://images3.alphacoders.com/105/thumb-1920-105099.jpg"); background-color:#B3C99E

background-size:cover;

background-repeat: no-repeat;

background-position:center bottom;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark bg-dark">

<div class="container">

<a class="navbar-brand" href="#">AMAZON KINDLE STORE REVIEW ANALYSIS</a>

<a href="https://software.farm.bot/docs/weed-detection" class="w3-btn w3-black" type="button">Help</a>

</div>

</nav>

<div class="container">

<div id="content" style="margin-top:2em">{% block content %}{% endblock %}</div> </div>

</body>

<footer>

<script src="{{ url\_for('static', filename='js/main.js') }}" type="text/javascript"></script> </footer>

</html>

**Index.html:**

{% extends "base.html" %} {% block content %}

<h1>Sentiment analysis</h1>

<div>

<br>

<form id="upload-file" method="post" enctype="multipart/form-data"> <label style="color:black" for="imageUpload" class="upload-label"> Choose...

</label>

<input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg"> </form>

<div class="image-section" style="display:none;">

<div class="img-preview">

<div id="imagePreview">

</div>

</div>

<div>

<button type="button" class="btn btn-primary btn-lg" id="btn-predict">Predict!</button> </div>

</div>

<div class="loader" style="display:none;"></div>

<h2 id="result"style="background-color:white;color:black;font-size:16px;"> <span></span>

</h2>

</div>

{% endblock %}

**APP.PY:**

import numpy as np

from flask import Flask, request, render\_template

from joblib import load

import joblib

from tensorflow.keras.models import load\_model

from sklearn.feature\_extraction.text import

CountVectorizer

app = Flask(\_\_name\_\_)

model=load\_model("amazo.h5")

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/y\_predict',methods=['POST']) def y\_predict():

'''

For rendering results on HTML GUI

'''

d = request.form['Sentence']

print(d)

loaded=CountVectorizer(decode\_error='replace',vocabulary=joblib.load('amazo.save'))

d=d.split("delimiter")

result=model.predict(loaded.transform(d)) prediction=result>0.5

if prediction[0] == False:

output="Positive review"

elif prediction[0] == True:

output="Negative review"

return render\_template('index.html', prediction\_text='{}'.format(output))

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)