SENTIMENTAL ANALYSIS USING AUTOENCODERS

Mini project report submitted in fulfillment of the Academic requirements

For the award of degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

(2017-2021)

BY

A.RAJENDER REDDY	**17241A12C2**
N.SRUNITH KUMAR	**17241A12F3**
N. PRUDHVI THORAN	**18245A1213**
MOHAMMAD FURQAN SAUDAGAR	**17241A12E9**

Under the guidance of

V. Vinisha

Assistant Professor, Dept of IT

DEPARTMENT OF INFORMATION TECHNOLOGY

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY, HYDERABAD

(AUTONOMOUS)

SENTIMENTAL ANALYSIS USING AUTOENCODERS

Abstract:

Social media has proven to be an effective way of communicating information and expressing opinions with others . As many people use social media every day, a lot of reviews, feedbacks, article is created and broadcast. Sentimental analysis which is also called as opinion mining is the study of people's mindset, their choices and their decisions for choosing a thing or taking a decision. The people's opinions and feelings may be in any language. It means it may be your native language also. Sentimental analysis have recently attracted many researchers in the deep learning community. As per the psychology, whenever we need to take a decision or to do something , we generally want to hear others opinions . This inspired us to start working on this. At last we are going to classify whether a given review is positive or negative.

CONTENTS

INTRODUCTION

1.1 Introduction to the project:

Sentiment Analysis is the process of finding and validating whether a meaning of a sentence is positive, negative or neutral. Extracting sentiments from the social media and recommending others to make the people's work easier of opting things.

Auto encoders have gained a lot of attention in recent years as a building block of Deep Learning. They act as the algorithms by constructing outputs from the given inputs. Using loss function, we will take sure that outputs are not so different from inputs. In a neural network implementation of auto encoders, the hidden layer is taken as the learned feature. There will be some difficulty with plain auto encoders much effort has to be spend on regularizations and loss function in order to prevent them against over fitting. Little attention for the loss function, which is a critical problem for modelling textual data. The loss functions are squared Euclidean distance and element-wise KL Divergence. The problem with those loss functions are they try to reconstruct all dimensions of input independently .However, we argue that this is not the optimal approach when we work with text classification. There are two reasons for this. First one is- In natural language the distribution of word frequencies obeys the power-law. According to power law, few of the most frequent words will be responsible for most of the probability mass of word occurrences. Due to this, the Auto encoder puts much more effort on reconstructing the most frequent words and ignores the less frequent ones. This may lead to a bad performance and less accuracy especially when the class distribution doesn't has the frequent words so well. For sentiment analysis, this problem is especially severe and it effects too. Because it is obvious that the useful and important features only occupy a small fraction of the whole vocabulary and Reconstructing irrelevant words such as 'chef' or 'tasty' very well is not likely to help learn more useful representations to classify the sentiment of restaurant reviews. The second reason for this is reconstructing all the words from the input text explicitly is also very expensive, because the latent space has to contain all the necessities of the inputs carried by the words. It really costs high, here we need to store the

words which are useful and which are not useful also. As the size of the vocabulary of the input can be of the size of five to ten thousand even for a medium sized dataset, the size of the hidden layer that has to be choosen should be very large such that it can learn all the inputs and yields better output, the large hidden layer will be a great disadvantage to the capacity of the model and makes it too difficult for scaling large problems.

1.2 Project Overview:

As we are from IT background, thinking everything in a positive and different way is an important trait of everybody. As we are dealing with information and we will be doing it too, we have chosen a type of recommendation system which can be used to recommend others to follow. In this present situation everything became virtual and people are making use of others opinion to do anything. They are trusting the others more than they trust themselves. This is the main idea behind the choosing of this project.

First thing, we do is we will load the data and completes the data pre processing techniques and then converts the words into vector and pass it to the auto encoder and atlast passing it through machine learning algorithm.

1.2.1 Existing System:

There are many existing systems for sentimental analysis. Now, we will be discussing some of the existing solutions for this. The one among them is knowledge based techniques. In this technique, the motto is to classify texts by effectively classifying the text based on the words such as good, bad etc. In this technique it gives less priority to particular words. The next existing solution for sentimental analysis is statistical methods. Statistical methods draws some applications from machine learning algorithms such as support vector machines, bag of words concept, semantic analysis etc and also draws some applications from deep learning also. For me, the opinion of the speaker should not be changed and should be grammatically followed. There is one more technique also named Hybrid techniques.

1.2.2 Proposed System:

We are going to implement it using auto encoders. Auto encoders is one of the applications of Artificial Neural Networks. This implementation gives a good accurate results compared to RNN's and CNN's.

Advantages of Proposed System:

By this system it will help others to decide whether a review is good or not.
The output of auto encoder is passed to a classifier so that it builds a model. By this we can save our time and obtain accurate results.

Chapter-II

LITERATURE SURVEY

My journey with my teammates into the project has started with the name of our project i.e Sentimental Analysis. Sentimental analysis is the concept of mining the text that identifies the main subject of the text. The applications of sentimental analysis is found in the areas such as helping the business to know whether their product is completely reached to the people or not, and in movie reviews and also social media and online conversations too. However, analyzing the social media streams is sometimes against the security policies and in that cases sentiment analysis is restricted to basic analysis and count based mostly metrics. With the recent advances in deep learning, the flexibility of algorithms to analyze text has improved significantly. Inventive use of advanced AI techniques is an efficient tool for doing in-depth analysis. We to tend to believe it's vital to classify incoming client spoken language a couple of complete supported following lines:

1. The main features of the brand's products and the costumer care service that the companies are providing.

2. The feedback of the users regarding those products.

Sentiment analysis which is also called as opinion mining or feeling AI infers to the study of employment of linguistic communication process, text analysis, and also bio science to consistently determining, extracting the useful things and studying the emotional feelings whether it is in the phrase format or subjective

format. Sentiment analysis is widely applied in the areas such as voice of clients or audience such as reviews and survey responses, online and social media.

Types:

The first thing that we need to do for any sentimental analysis is classifying and understanding the context of the text that is used according to the situation and determining whether the expressed sentence in a document is positive, negative, or neutral. Coming to the next level of analysis, In this step we will look after the classification of the emotional states such as angry, sad, and happy. There are many forms of sentiment analysis. Some of them are facet primarily based sentiment analysis, Grading type sentiment analysis (positive, negative, neutral), polyglot sentiment analysis and detection of various emotions. As of now, don't know any depth information regarding the auto-encoders. We've got started assembling the connected theory relating to the auto encoders.

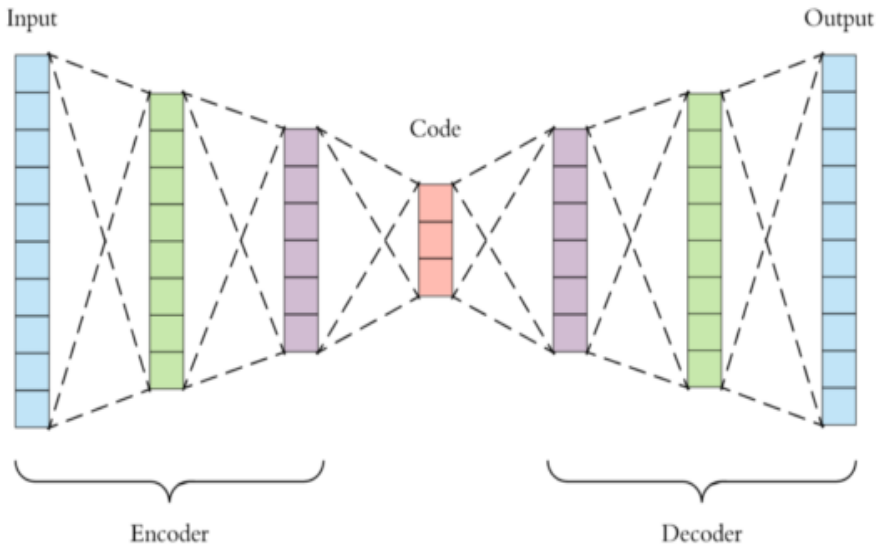
AUTOENCODERS:

An auto encoder is known to be a kind of artificial neural network. The main aim of auto encoders is it will learn the economical information codings in a unsupervised fashion. When we want to deploy some image or anything which is of highly dimensional to internet. There will be the problem of taking high time or space. Here with the help of auto encoders we can reduce the dimension of the image or anything or deploy it which takes less time and space. Many assumptions and other features exist to the essential one with the aim of forcing the learned representations (the hidden representations) of the input to assume helpful properties. Auto encoders square measure is responsible for finding several issues, from facial recognition to feat of the linguistics which means vocabulary of words. Auto encoders remodels the unsupervised learning method to a supervised one by the method of self reconstruction. This allows us to use all the tools which are developed for supervised learning like back propagation to expeditiously train the auto encoders. An auto encoder is one of the applications of artificial neural networks .It just understands the input and copies it to the output. It has an internal layer (an encoder) which describes a code which is used to read the input, and also a output layer. It has two main

parts: an encoder that maps the input into the code, and a decoder that maps the code for the reconstruction of the original input it may be a image or text.

The idea of auto encoders has became very famous in the field of neural networks, and the first application based on auto encoders was made in 1980's . The most important application of auto encoders are dimensionality reduction and feature learning, But recently we are using this application for studying generative models also.

Basic Architecture



Schema of a basic Autoencoder

- **Encoder:** This part of the network is used for compressing the input into a latent space representation. The encoder layer encodes the input whether it is text or image as a compressed representation in a reduced dimension. The compressed image or text is the another version of the original image or text.
- **Code:** This part of the network is the important one in the architecture of auto encoder. It is responsible to represent the compressed format of the input which is fed to the decoder.
- **Decoder:** This part of the network is for decoding the encoded image or text back to the original dimension (original input). The decoded image or text is a lossy

reconstruction of the original image or text and it is reconstructed from the latent space representation provided.

In the simple auto encoder case, there will be one hidden layer, the encoder part of an auto encoder takes the input and maps it to code that means latent representation. Code or latent representation represents the same. The activation functions used here are such as a sigmoid function or a RELU function. Weights and biases will be assigned to the networks, and then updated iteratively during training through a method called back-propagation.

The main functionality of an auto encoder is to reduce the multi dimensional data into less dimensional so that it can effectively deal with the large datasets which is the present day trend. We may have a doubt that PCA (principal component analysis) is already present and why do we need auto encoders and all. The auto encoders came up with a new functionality. Principal component analysis is used only for linear representations but auto encoders supports non-linear representations also. This is why we prefer auto encoders to PCA (Principal component analysis. There is a possibility to make auto encoders to function as PCA. It can be done by using linear as the activation function. The maths behind the auto encoders is very easy to understand. First let us divide the whole architecture of auto encoder into two parts namely the encoder, and the decoder.

$$\phi : \mathcal{X} \to \mathcal{F}$$

$$\psi : \mathcal{F} \to \mathcal{X}$$

$$\phi, \psi = \underset{\phi,\psi}{\arg\min} \left\| X - (\psi \circ \phi) X \right\|^2$$

Here, Let us understand the above equations. The first equation which is denoted by phi tells about the input format which is encoding part. The input data is represented by X, and it is made reduced to a code or a latent representation F, which is present at the middle that we call it as bottleneck. The decoding function, which is denoted by ψ, maps the compressed form of input present at the latent space F to the output. As expected the output will be same as the input

that we have given. The thing we are doing here is nothing but the recreation of the input text or image by compression. Here z is the latent dimension which is represented after passing through the activation function.

$$\mathbf{z} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

The decoding portion x' is represented in the same way but with different activation functions and different strategies based on the type of inputs. The activation function that we have choosed depends on the type of inputs. If the input is 0 and 1 the RELU activation function is used and so on.

$$\mathbf{x}' = \sigma'(\mathbf{W}'\mathbf{z} + \mathbf{b}')$$

The main thing is the discussion about the loss. The loss function is based on these equations. The loss function is derived in terms of Euclidean distance.

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')\|^2$$

Here the thing that we have observed is the inputs and outputs are equal. Here we can have a doubt that whether it belongs to supervised learning or unsupervised learning. So, we can name this type of learning is semi supervised learning. The main thing that we need to make sure is that to choose a good type of auto encoder based on the inputs and their functionalities. If we take very less number of nodes at the latent space, the ability to read the input and recreating the same input will not actually work and the images that we obtain at the output layer may be blurred. If we use too many nodes, then we need not think about compression much.

Types of auto encoders:

Denoising Autoencoders

There are many different kinds of auto encoders .One among them is denoising auto encoders. It is also the most widely used auto encoder. The main functionality of this auto encoder is it adds some noise to the training data before training but they becomes natural and correctly compares the error to the original image during instruction. This helps not to make the image so complex and noisy.

Sparse auto encoders

The sparse auto encoder, the intuitive counter, has a greater latent measurement of input or output dimensions. The sparse auto encoder contain the large latent space. A sparsity constant is added while training in the loss function.

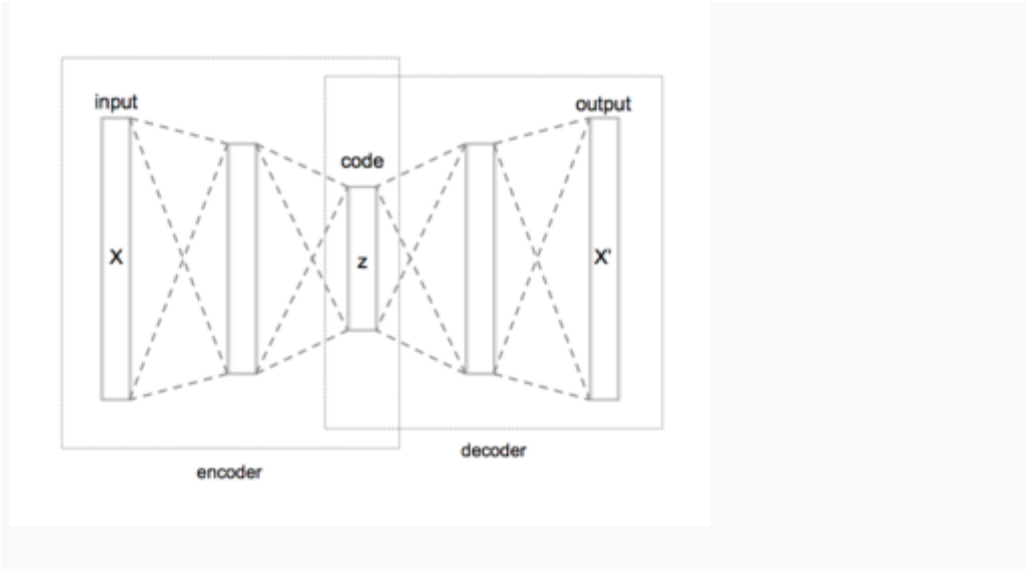
Contractive auto encoders

Contractive auto encoding is very similar to the last two approaches, but in this case, we didn't adjust the structure and honestly add structured loss feature. This could be an idea as a nervous form of hillside slope.

Variational auto-encoders

We can simply say that VAE’s are the extension of the traditional(basic one) auto encoders. The design of VAE’s is similar to the design of the traditional one which allows us to randomly sample random space. The random samples will be present here. Those random samples will be decoded to get a desired output.

Advantages of Depth:



Auto encoders are generally trained by using a single input layer and a single output layer which means a single layered decoder and a single layered encoder.

- There are many advantages of having deeper networks. They are
- Due to this depth, the training data will be reduced because of the more intermediate layers.
- The other advantage of deep auto encoders is that they yields good results when compared to the linear (single layered input and output) auto encoders.

Applications:

The main applications of auto encoders since the 80s are

Dimensionality Reduction

Information Retrieval

Relation with PCA

Anomaly Detection

Image Processing

Drug discovery

Population synthesis

Popularity prediction

Machine Translation

Chapter-III

SYSTEM REQUIREMENTS

3.1 Hardware System Configurations

The hardware requirement is nothing but it tells what hardware things we are going to use. The features of our choosen hardware requirements are as follows.

- Processor :- Intel core i3 1.7GHz
- Hard disk :-100 GB.
- RAM :-4 GB.

3.2 Software System Configurations

The software requirement is nothing but it tells what software products we are going to use. For eg like the platform used to code. The software products should contain the versions such as

- Operating system :- Windows 10
- Coding Language :- Python.
- Command Prompt :- Anaconda Prompt
- Libraries :- Keras, Sklearn, pandas, numpy, beautifulsoup

Chapter-4

SYSTEM DESIGN

4.1 Introduction to UML

Unified modeling language (**UML)** is a standardized modelling language which enables the developers to specify, visualize, construct and document artifacts of software system**.** UML is a way of representation in a pictorial or graphical form which is used to represent the input data and the actions performed on the system and output generated by the system.

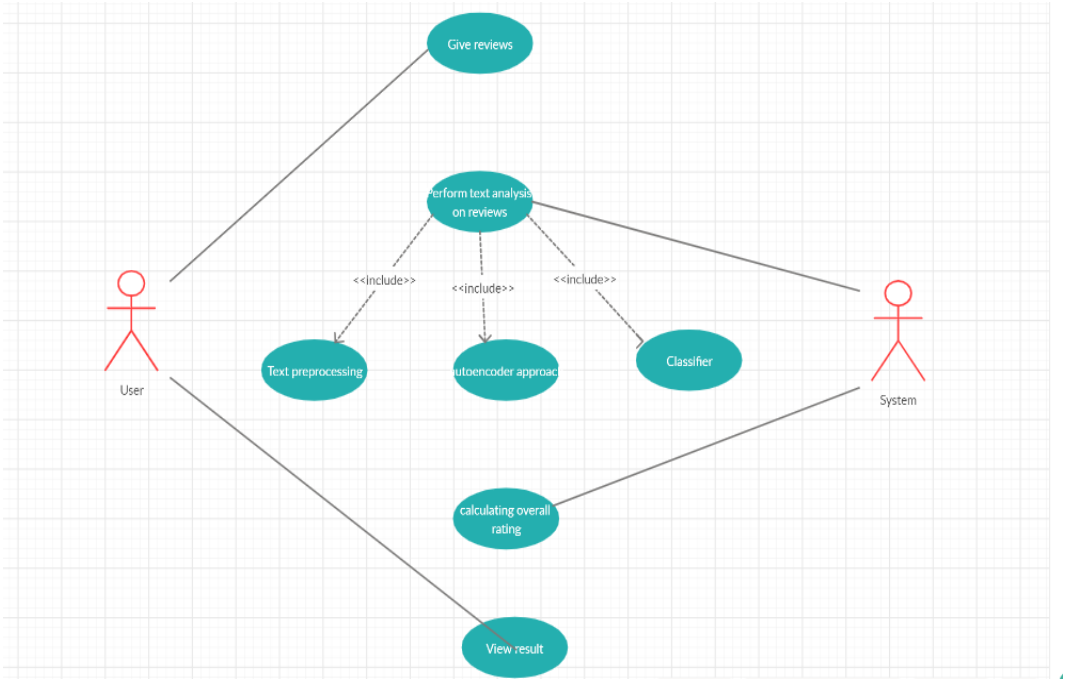
4.2 UML DIAGRAMS

4.2.1 USECASE DIAGRAM

In this use-case diagram there are two actors user and the system

The actions performed by user are giving reviews, view the rating

The actions performed by system are performing text analysis on reviews, which includes text-preprocessing, includes auto encoders approach, Logistic Regression classifier, calculation of rating.



4.2.2. CLASS DIAGRAM:

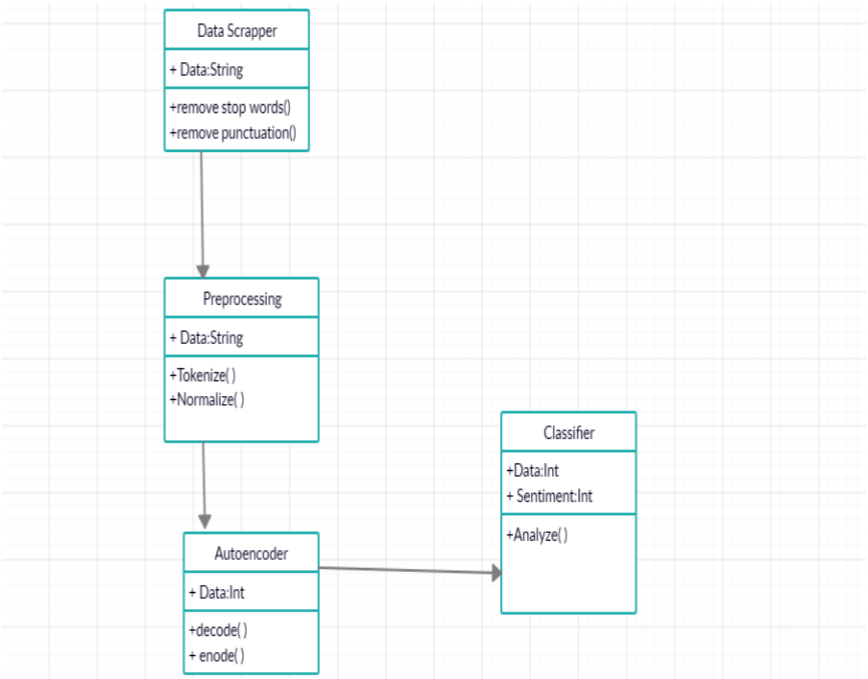
Here datascrapper, Preprocessing, Autoencoder, Classifier are the four classes that we have defined

In datascrapper removing stopwords( )and removing punctuations( ) are methods

In Preprocessing Tokenize( ) and Normalize methods are present

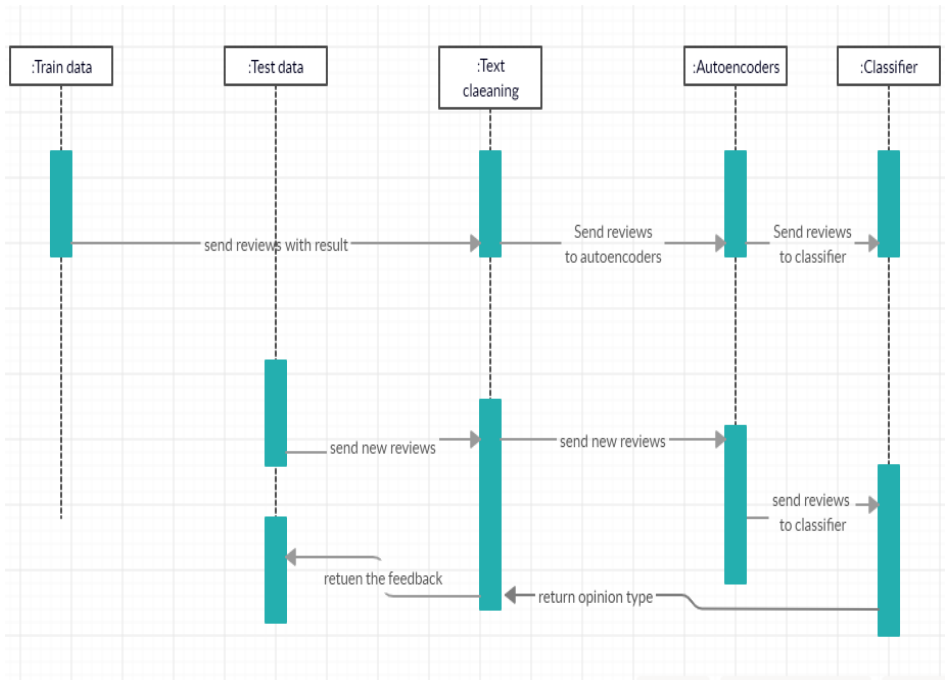
In Autoencoder encode( ) and decode( ) methods and In classifier analyzing

All are declared as public.



4.2.3 SEQUENTIAL DIAGRAM

In this diagram,trainingdata,testing data, feature extraction,auto,classifier are objects



4.3 **INPUT DESIGN**:

The input is the review will be given to the model and after analyzing the review . The input is nothing but a sentence which is a group of words related to restaurant

Eg The food is tasty

The food is terrible etc..

4.4 **OUTPUT DESIGN:**

The output will be in the format whether the given review is positive or negative after doing some processing.

Eg: The feedback is positive

The feedback is negative

CHAPTER 5

SOFTWARE ENVIRONMENT

5.1. **PYTHON**

- Python is high level, interpreted, interactive language.
- It is very easy to build modules in python.
- It is very easy to learn and implement python.
- Python provides inbuilt libraries by which the time of the programmers gets saved.
- Python also supports object oriented programming language.

5.2. **PYTHON LIBRARIES**

Python provides many inbuilt libraries which makes our work easier, and faster. In this project we have used many libraries such as keras, sklearn, pandas, numpy, beautifulsoup, porterstemmer.

Keras for data preprocessing.

Sklearn for classifier.

Pandas for reading data and making dataframes.

BeautifulSoup for removing HTML tags, punctuations etc

Porterstemmer for getting root of a word.

CHAPTER-6

IMPLEMENTATION

The main steps for any machine learning project is

i. Gathering data

ii.Preprocessing

iii.Training a model

The first thing for any machine learning algorithm is to import the required packages

The second thing is to load the datasets

1. Loading Datasets:

As our project is related to recommendation system. We have choosed restaurant reviews dataset which contains two columns Review and Label

There are 1000 rows. None of the row is empty. This is a good sign for accuracy metric. The dataset which we took is in tsv format. TSV stands for Tab-separated values file. TSV is a file extension for a tab-delimited file. It is used with spreadsheet software. TSV stands for Tab Separated Values. These files are used for raw data and can be imported into and exported from spreadsheet software in a multi-purpose way.

The next process is

2. Data Pre-processing:

This is the main part of any machine learning algorithm. In our project, as it is word data we need to convert it into categorical data. This is the main step.

Cleaning data:

Beautiful soup is a Python library for removing all the HTML tags from the data.. It commonly saves the time of programmers.

By this we have removed all the html tags, removes the numbers in the string and also the punctuations.

Porter Stemmer:

Stemming is the process of producing the root word without gerund forms. Stemming programs are commonly referred to as stemming algorithms.

Now our data is free from all the tags, punctuations, ing forms, numbers, stopwords, uppercase and lowercase misconceptions etc..

We have divided the words to the format of tokens. We have used tokenizer to accomplish the task.

Now the biggest step of our project to convert this into categorical data.

Word Embeddings:

We have many techniques for this from various libraries such as One-Hot-Encoding, Label Encoding, GLOVE

We have choosed Count Vectorizer to do this job. Finally we are ready with our categorical data.

3.Passing to auto encoder:

We have selected the sparsity autoencoder which will add some sparsity constraint to the autoencoder and gives good results.

4. Selecting a model:

We have selected logistic regression classifier and trained the model. The results that we received from the auto encoder is passed to the logistic regression classifier and tested on the test data

6.1 SAMPLE CODE:

```
import numpy as ns

import pandas as pn

df=pn.read_csv('/kaggle/input/reviews/Restaurant_Reviews.tsv', sep='\t')

df.head(5)

df.isna().count()
```

```
# Packages we need to import
import re
import collections as clt
#Packages for data cleaning
from sklearn.model_selection import train_test_split as tt
from nltk.corpus import stopwords as stpw
from keras.preprocessing.sequence import pad_sequences as pds
# Packages for modeling
from keras import models as mdl
from keras import layers as ly
from keras import regularizers as reg
def clean_reviews(review):
    # 1. Removing html tags
    reviewed_text = bes(review).get_text()
    # 2. Retaining only alphabets.
    reviewed_text = re.sub("[^a-zA-Z]"," ",reviewed_text)
    # 3. Converting to lower case and splitting
    word_tokens= reviewed_text.lower().split()
    # 4. Remove stopwords
    stop_words= set(stpw.words("english"))
    word_tokens= [word for word in word_tokens if not word in stop_words]
    cleaned_review=" ".join(word_tokens)
    return cleaned_review
```

```
import re
from bs4 import BeautifulSoup as bes
df['Review']=df['Review'].apply(clean_reviews)
df.head(5)
#cleaning the text
import nltk as nlt
from nltk.corpus import stopwords as stpw
from nltk.stem.porter import PorterStemmer as pstm
corpus = []
for i in range(0,1000):
    review = re.sub('[^a-zA-Z]', ' ', df['Review'][i])
    review = review.lower()
    review = review.split()
    ps = pstm()
    review = [ps.stem(word) for word in review if not word in set(stpw.words('english'))]
    review = ' '.join(review)
    corpus.append(review)
from sklearn.feature_extraction.text import CountVectorizer as cvr
cv = cvr(max_features = 1565)
Z= cv.fit_transform(corpus).todense()
p = df.iloc[:,1].values
#changing to dataframe
```

```
import pandas as pn

df3=pn.DataFrame(Z)

df3.head(6)

from keras.layers import Input as inp

from keras.layers import Dense as den

from keras.models import Model as mdl

from keras import regularizers as reg

encoding_dim = 32

input_layer = inp(shape=(1565,))

# add a Dense layer with a L1 activity regularizer

encoded = den(encoding_dim, activation='relu',

              activity_regularizer=reg.l1(10e-5))(input_layer)

decoded = den(1565, activation='sigmoid')(encoded)

auto = mdl(input_layer, decoded)

auto.compile(optimizer='adadelta',loss='mse')

auto.fit(df3, df3,epochs=100)

from keras.models import  Sequential

hidden_representation = Sequential()

hidden_representation.add(auto.layers[0])

hid_rep = hidden_representation.predict(df3)

from sklearn.model_selection import train_test_split as tt

X = hid_rep[:]

y = df['Liked']
```

```
X_train, X_test, y_train, y_test = tt(X, y, test_size=0.20, random_state=42)

from sklearn.linear_model import LogisticRegression as LR

#from sklearn.naive_bayes import GaussianNB as GNB

from sklearn.metrics import classification_report, accuracy_score as cr,ascore

clf = LR().fit(X_train, y_train)

pred_y = clf.predict(X_test)

cm=cmatrix(y_test, y_pred)

print (cr(y_test, pred_y))

print (ascore(y_test, pred_y))

#for visualization

from sklearn.metrics import confusion_matrix as cmatrix

import seaborn as sb

import pandas as pn

import matplotlib.pyplot as plot

df_cm = pn.DataFrame(cm, range(2),range(2))

#plt.figure(figsize = (10,7))

sb.set(font_scale=1.4)#for label size

sb.heatmap(df_cm, annot=True,annot_kws={"size": 16})# font size

print("Accuracy Score is :", ascore(y_test, pred_y))
```

CHAPTER 7

SYSTEM TESTING

Testing is the process where the software product that we have build is whether meeting its requirements or not. The product is working under all circumstances or not. This process of validating the product is called testing. Testing can be done in various ways. It can be checked modules wise or units wise and etc .System testing is the type of technique which ensures that whether the whole system is working in a synchronous manner and functioning well. The test data that we choose to ensure its correctness should be able to meet all the requirements. The implementation that means working of the product will be checked against its requirements and make sure that its users not to feel any difficulty to use it. Unit testing and system testing will be done during this phase. Without testing no software or hardware project will be deployed. After the software has been developed, before giving handling it to the user the software will be tested whether it is meeting all its requirements or not. This testing includes various methods and strategies by which one can ensure that the software is reliable and good. The program will be tested logically and pattern of execution of the program will be checked. Thus the code will be checked for possible test data and outputs will be verified.

MODULE TESTING: In this type each module will be tested independently without effecting other modules. This strategy helps us to identify error and rectifying it without affecting the other modules. Whenever the program isn't satisfying the desired functionality, it must be corrected to induce the desired result. Each module within the system is tested separately.

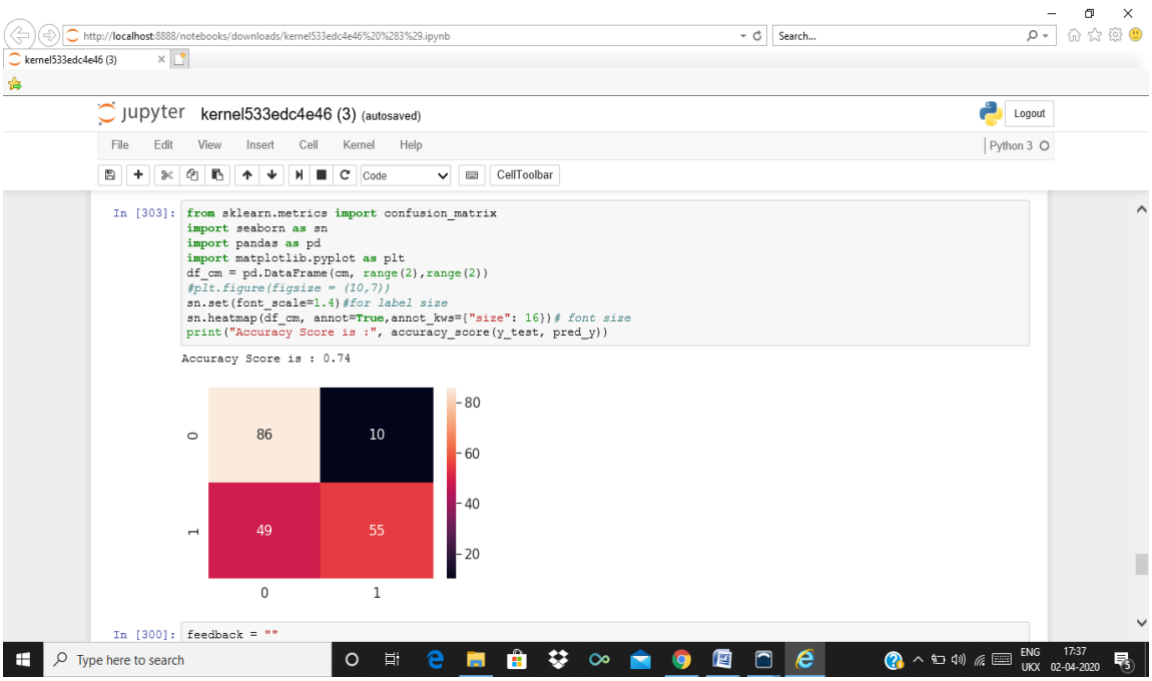
INTEGRATION TESTING: Once we have done with module testing, the next thing that we need to focus on is integration testing. In integration testing the modules are mixed and checked whether they are functioning well or not. There may be chances of occurrence of errors when modules are integrated. Those errors will be rectified during this phase.

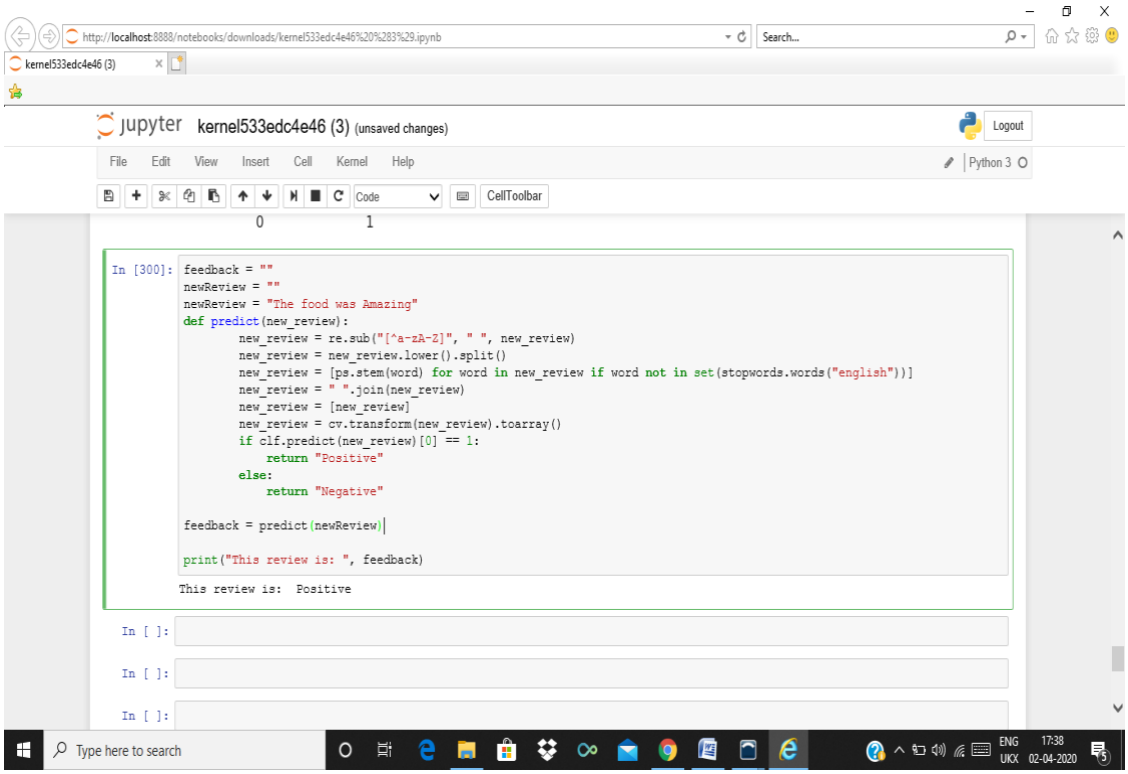
7.1 **TEST CASES**

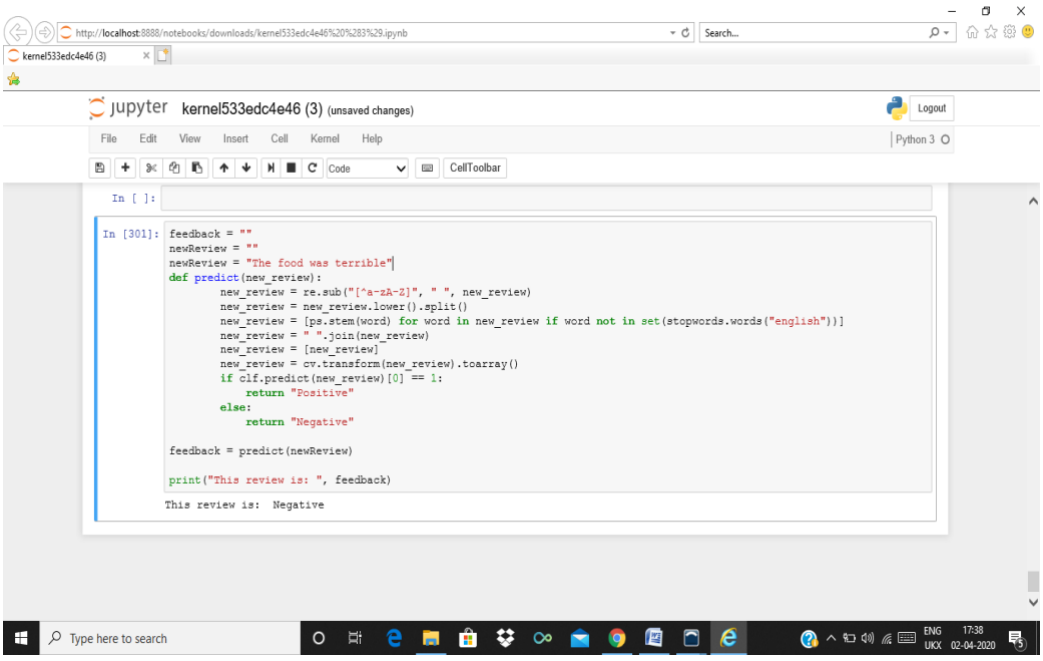
Test Case Id	**Test Case Name**	**Test Case Description**	**Test Steps**			**Pass/ Fail**
			Step	**Expected**	**Actual**	
01	Upload the restaurant reviews dataset	Verify either dataset is loaded or not	If dataset is not uploaded	It cannot display the file loaded message	File is loaded which displays the loaded message.	Pass
02	Preprocessing	Whether preprocessing on the dataset applied or not	If not applied	It cannot display the necessary data for further process	It can display the necessary data for further process	Pass

CHAPTER 8

RESULTS AND SCREENSHOTS




http://localhost:8888/notebooks/downloads/kernel533edc4e46%20%283%29.ipynb
Search...
kernel533edc4e46 (3)
jupyter kernel533edc4e46 (3) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Help
Python 3
Code
CellToolbar
0 1
In [300]:
feedback = ""
newReview = ""
newReview = "The food was Amazing"
def predict(new_review):
new_review = re.sub("[^a-zA-Z]", " ", new_review)
new_review = new_review.lower().split()
new_review = [ps.stem(word) for word in new_review if word not in set(stopwords.words("english"))]
new_review = " ".join(new_review)
new_review = [new_review]
new_review = cv.transform(new_review).toarray()
if clf.predict(new_review)[0] == 1:
return "Positive"
else:
return "Negative"
feedback = predict(newReview)
print("This review is: ", feedback)
This review is: Positive
In [ ]:
In [ ]:
In [ ]:
Type here to search
ENG
UKX
17:38
02-04-2020


http://localhost:8888/notebooks/downloads/kernel533edc4e46%20%283%29.ipynb
Search...
kernel533edc4e46 (3)
jupyter kernel533edc4e46 (3) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Help
Python 3
Code
CellToolbar
In [ ]:
In [301]:
feedback = ""
newReview = ""
newReview = "The food was terrible"
def predict(new_review):
new_review = re.sub("[^a-zA-Z]", " ", new_review)
new_review = new_review.lower().split()
new_review = [ps.stem(word) for word in new_review if word not in set(stopwords.words("english"))]
new_review = " ".join(new_review)
new_review = [new_review]
new_review = cv.transform(new_review).toarray()
if clf.predict(new_review)[0] == 1:
return "Positive"
else:
return "Negative"
feedback = predict(newReview)
print("This review is: ", feedback)
This review is: Negative
Type here to search
ENG
UKX
17:38
02-04-2020

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 **Conclusion**

From this project we can conclude that working with auto encoders is awesome and it will become the building block for deep learning projects. As we worked on the text, auto encoders found it difficult to work with the large text. The problem that we have faced with this is during word embeddings. If it is clear, then you will be getting very good results.

9.2 **Future Enhancements**

We strongly believe that there will be future studies on auto encoders and new type of auto encoders will be come into picture where there is no need of converting text data into categorical data. We also believe that improving the standard of these word embeddings can also be a way for getting more accurate results.

In future GLOVE (which is one type of word embeddings) is coming into picture to convert the text data into categorical data.

CHAPTER 10

BIBILIOGRAPHY

1. Aggarwal. Data mining: The textbook. Springer, 2015

2. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12, pp. 2493– 2537, 2011.

3.Wikipedia

4. https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726

5. http://scholar.google.co.in/scholar_url?url=https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/download/12059/11750&hl=en&sa=X&scisig=AAGBfm19mEjedi3U5cwYkFwUsy0LCM8PrA&nossl=1&oi=scholarr