EXAMINING AND PREDICTING HELPFULNESS OF REVIEWS BASED ON LSTM

EXAMINING AND PREDICTING HELPFULNESS OF REVIEWS BASED ON LSTM

A MAJOR PROJECT REPORT

Submitted by

Saran Reddy	**RA1611003020024**
Sravan Akuthota	**RA1611003020032**
Aravind Yalla	**RA1611003020047**

Under the guidance of

Ms.A.ARUNA

(Assistant Professor(O.G), Department of Computer Science and Engineering)

in fulfillment for the award of the degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM CAMPUS, CHENNAI -600089

MAY 2020

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University U/S 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled **"EXAMINING AND PREDICTING HELPFULNESS OF REVIEWS BASED ON LSTM"** is the bonafide work of **Saran Reddy – RA1611003020024 | Sravan Akuthota – RA1611003020032 | Aravind Yalla – RA1611003020047** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an occasion on this or any other candidate.

SIGNATURE

Ms.A.ARUNA
Assistant Professor(O.G),
Computer Science and Engineering,
SRM Institute of Science and Technology,
Ramapuram Campus, Chennai.

SIGNATURE

Dr. N. KANNAN., Ph.D.,
Professor and Head
Computer Science and Engineering,
SRM Institute of Science and Technology,
Ramapuram Campus, Chennai.

Submitted for the project viva-voce held on...........................at SRM Institute of Science and Technology, Ramapuram Campus, Chennai -600089.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM, CHENNAI - 89

DECLARATION

We hereby declare that the entire work contained in this project report titled "**EXAMINING AND PREDICTING HELPFULNESS OF REVIEWS BASED ON LSTM**" has been carried out by Saran Reddy – RA1611003020024 | Sravan Akuthota – RA1611003020032 | Aravind Yalla – RA1611003020047 at SRM Institute of Science and Technology, Ramapuram Campus, Chennai-600089, under the guidance of **Ms.A.ARUNA., Assistant Professor(O.G)**, Department of Computer Science and Engineering.

Place: Chennai

Saran Reddy
Sravan Akuthota A. Sravan
Aravind Yalla Y. Aravind

Date:

Own Work Declaration

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : B.Tech | Computer Science and Engineering

Student Name **:** Saran Reddy | Sravan Akuthota | Aravind Yalla

Registration Number **:** RA1611003020024 | RA1611003020032 | RA1611003020047

Title of Work **:** Examining and Predicting Helpfulness of Reviews based on LSTM

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

RA1611003020024	RA1611003020032	RA1611003020047
	A. Sravan	Y. Aravind

ACKNOWLEDGEMENT

We place on regard of our deep sense of gratitude to our lionized Chairman **Dr.R.SHIVAKUMAR** for providing us with the requisite infrastructure throughout the course.

We take the opportunity to extend our hearty and sincere thanks to our Dean, **Dr.G.SELVAKUMAR, M.Tech, PhD.,** for maneuvering us into accomplishing the project.

We take the privilege to extend our hearty and sincere guidance to the Professor and Head of the Department, **Dr.N.KANNAN, M.Tech, PhD.,** for his suggestions, support and encouragement towards the completion of the project with perfection.

We convey our hearty and sincere thanks to our Project Coordinator **Dr.V.SELLAM, M.Tech, PhD.,** for her fortification. We express our hearty and sincere thanks to our guide **Ms.A.ARUNA., Assistant Professor(O.G) ,** Computer Science and Engineering Department for her/his sustained encouragement, consecutive criticism and constant guidance throughout this project work.

Our thanks to the teaching and non-teaching staff of the Computer Science and Engineering Department of SRM Institute of Science and Technology, Ramapuram Campus, who provided necessary resources for our project.

PROJECT STUDENT NAMES

Saran Reddy

A. Sravan **Sravan Akuthota**

Y. Aravind **Aravind Yalla**

ABSTRACT

Understanding the reviews gave by the client gives us the characteristics of the survey and causes us to locate the most accommodating reviews from generally accessible plenteous audits. These online surveys give us an extraordinary asset to conclude whether to go with an item or not. However, how we choose the review is useful and to what degree it tends to be proposed to that client. There are a few sites that give supportive surveys considering just two situations the one with photographs or a lengthy review. Be that as it may, we moved toward this issue with an arrangement of points of view, right off the bat the dataset handling where we wash down the information as this is the will be the essential advance of preparing by cleaning the delimited content utilizing JSON. After that, the component determination is done to the unstructured information by catching the impact auxiliary highlights utilizing the TF*IDF strategy to discover the heaviness of catchphrase and apportion the significance to that watchword relies upon its hour's rehashes in the sentence. Furtherly, a model determination is accomplished by two classes specifically, Probabilistic Measures and Resampling Methods. At long last, the key advance forecast of support is finished by the LSTM a Recurrent neural system prepared to utilize the backpropagation that beats the angle issue. Regardless of essence of advantaged connection and other social edges, our outcomes suggest that end clients esteem catchphrases and pertinence over sentence structure and coherence. Moreover, it likewise implies that clients may have a restricted "consideration" for perusing each review. Along these lines, we propose this paper to suggest helpful review

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

WORD	**ABBRIVATION**
RNN	Recurrent neural Network
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
GRU	Gated Recurrent Units
DBSCAN	Density-based spatial clustering of applications with noise
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
NLTK	Natural Language Processing Toolkit
API	Application Programming Interface
WSGI	Web server Gateway Interface
URL	Uniform Resource Locator
HTTP	Hyper Text Transfer Protocol
XML	Extended markup Language
SDLC	Software Development Lifecycle

CHAPTER-1

INTRODUCTION

1.1 Overview:

The content of written feedback differs considerably; high-quality feedback can collect a significant amount of valuable votes from the audience over time. The freshness of the review is an essential quality attribute, and a model that correctly forecasts the amount of valuable votes that the written review would earn in the future that bring considerable commercial benefit to businesses. Knowing the value of the analysis in advance, companies may offer high-quality, fresh feedback to their clients to obtain insight into their goods and services. It is particularly difficult to predict the usefulness of the review due to the scarcity of labelled data, with only less than 10% of the reviews released. The Yelp data set has a large usefulness measure (more than three available votes). Furthermore, unlike sentimental research, where positive and negative sentimental words play an significant role in the process of classification, there are no apparent features that explicitly indicate the utility of the text.

1.2 Problem Statement:

- We discuss the outcomes of the test, and allows us the chance to vote for funding. If the consumer wishes to do so, he or she can upvote or downvote a comment to his or her helpfulness. For eg, 3/5 would mean that three participants believed that the research was beneficial in thinking otherwise. There may be a number of people who have read the analysis but have not expressed their opinions.
- We solve the problem by posing it as a matter of classification. Analysis is labelled as "helpful" or "unhelpful;" instead, different algorithms predict labels. Here we're going to use the Random Forest Bag-of-Words as a reference. In addition, we analyzed the basic RNN and LSTM tests and compared the results.

- It is important to consider how users view product reviews as they may include insight about what affects buying choices of the customer. We address this topic by the study of the data and the certainty of its usefulness. We continue to use deep learning frameworks such as RNN and LSTM as a binary classification problem.

1.3 Objective

- The preliminary objective is Learning the reviews received by the consumer gives us the consistency of the review and allows us to identify the most valuable comments from the widely accessible feedback. These online reviews are a fantastic tool for us to determine whether or not to go for the Product.
- Prediction of helpfulness is done by the LSTM a Recurrent neural network trained by using the backpropagation which overcomes the gradient problem

1.4 Organisation of the Report:

The report consists of 8 chapters which together cover all the details regarding the Examining and Predicting Helpfulness of reviews based on LSTM:

The first chapter introduces the project and explains in detail the overview and the project statement. It goes in depth about the issue at hand and also talks about the objective of the project and how it helps in improving the problem situation.

The second chapter of the report deals with the literature survey of the project. It introduces multiple related projects which either use the same technology or are in the same field. It also points at the scope of improvement or issues in the existing projects. At the end of the chapter, all the literature surveys are summarized.

The third chapter of the report talks about the system design that is used in this project. The introduction mentions the technology and procedure used in Examining and Predicting Helpfulness of reviews. The System Architecture explains the step by step working and the different stages involved using a workflow diagram. It also mentions the system requirements for the implementation and summarizes the working of the project.

The fourth chapter involves the description of different modules used in the working of this system. It lists the module names and goes in depth and explains the working of each module using a flowchart or a workflow diagram if available.

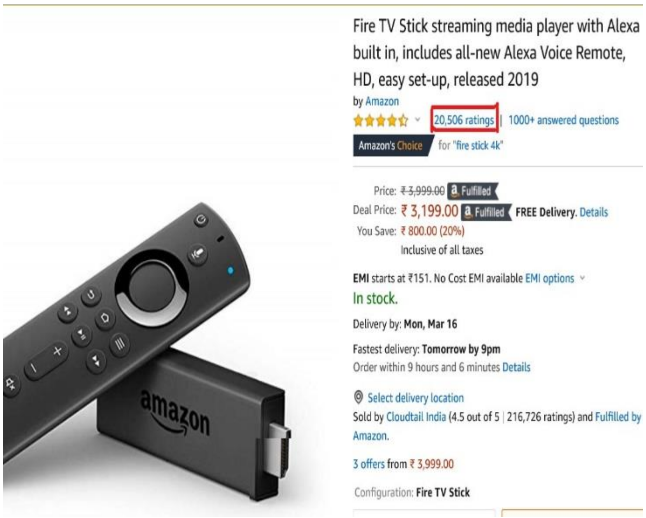
The fifth chapter of the report involves the implementation of the system or project. In this chapter, the overview of the platform used to run the program is provided. It also mentions the simulation parameters required to run the code on the platform. Finally, it contains some sample code and screenshots of the output received.

CHAPTER-2

LITERATURE SURVEY

2.1 Introduction:

Customer feedback System field teeming in today's e-commerce environment. However, not all the region comments were viewed equally by the machine. Some tend to be additional than others and the region unit additionally important to the user's buying request. It would be helpful to learn what makes a successful analysis. Totally different from a practical study, this would be an actual agent's mission. 2 comments could both look insightful, but usually one of them was viewed as helpful and the alternative was not.



Fire TV Stick streaming media player with Alexa built in, includes all-new Alexa Voice Remote, HD, easy set-up, released 2019

by Amazon

20,506 ratings | 1000+ answered questions

Amazon's Choice for "fire stick 4k"

Price: ~~₹ 3,999.00~~ Fulfilled

Deal Price: ₹ 3,199.00 Fulfilled FREE Delivery. Details

You Save: ₹ 800.00 (20%)

Inclusive of all taxes

EMI starts at ₹151. No Cost EMI available EMI options

In stock.

Delivery by: Mon, Mar 16

Fastest delivery: Tomorrow by 9pm

Order within 9 hours and 6 minutes Details

Select delivery location

Sold by Cloudtail India (4.5 out of 5 | 216,726 ratings) and Fulfilled by Amazon.

3 offers from ₹ 3,999.00

Configuration: Fire TV Stick

Fig. 1. Screenshot of an amazon product with 20,506 reviews

We would like to value the performance of knowledge purification, feature choice and LSTM on such a task. In our downside environment, we continue to discuss details regarding on-line ratings, which offers an related incentive for input. If the user chooses to upvote whether it's useful or not. contemplate a situation, wherever seven out of ten notice it useful which suggests seven individuals notice it useful and 3 thought the review wasn't useful. There may well be more UN agency saw the review however didn't categorical their views. we tend to approach {the downside the matter} by casting it as a classification problem.

Considering any comment from the stack of all comments anywhere they are marked as helpful or not, we appear to determine the labels by various algorithms. Here we can use knowledge purification and have the option of the TF*IDF technique as a benchmark.

2.2 Existing System:

- The effect of text opinion on the utility of the article is more important for positive and comparative reviews than for standard reviews. Moreover, the ambiguity of the text of the comments has a major inverted U-shaped association with the utility of the article, and the connection is greater with standard comments relative to positive and comparable reviews.
- Text opinion has a detrimental impact on the quality of the article, and the impact is greater for positive reviews than for comparable and standard reviews. Finally, we use a random forest approach to estimate the utility of the analysis depending on its numerical and textual characteristics. The new method has shown that the duration of analysis is the most important element in assessing the effectiveness of online feedback.
- For the text mood of the online comments, the previous research indicated that it had an asymmetrical effect on the utility of the study. In fact, poor feedback have had a greater degree of customer support than optimistic feedback.

2.3 Issues in Existing System:

- Semantic Features are not considered

  Semantic features are the essential semantic components of meaning for every lexical object. An individual semantic attribute constitutes one component of the intension of a phrase, which is the intrinsic significance or idea evoked. It is believed that the linguistic sense of a phrase derives from similarities and major variations with certain words. Semantic features allow linguistics to clarify how terms that share such features can be part of the same semantic domain. Correspondingly, the difference between the sense of terms is clarified by various semantic characteristics.

- Low Reliability

  Reliability applies to whether or not you get the same response when using an tool to calculate it more than once. In basic words, the value of study means that the analysis process delivers reliable and accurate findings.

- Performance of evaluation parameters cannot be improved

  Performance is the execution of a mission requiring the implementation of expertise with the parameters which cannot give improved results.

- High Complexity

  Complexity characterizes the complexity of a system or model whose components communicate in a number of ways and comply with local rules, meaning that there is no logical, higher guideline for defining the multiple possible interactions.

- With no lexical Analysis

  The Lexical Analysis is the first step in the compiler phase. This requires the modified source code from the language pre-processors that are written as phrases. The lexical analyser splits the document into a set of tokens by removing some document space or remarks from the source code.

2.4 Summary of Literature Survey

- Detection of misleading reviews on internet

Detection of misleading reviews on internet goods utilizing Decision Tree and Data Gain Internet Feedback are one of the main reasons for consumers to buy any product or to receive service from a variety of sources of evidence that can be utilized to evaluate the consumer sentiment on the items. Fake reviews will be intentionally published, Economic motives Most false comments will surface on such websites.

- Reliable Fake Review Detection

Reliable Fake Review Identification by Modeling Temporal and Behavioural Trends Fake reviews have been a big problem in online review networks, where malicious users exploit the impression of an item (e.g. a restaurant) by creating false reviews. This method collectively measures the credibility of consumers as well as the truthfulness It is challenging to differentiate fake comments from real ones based on ranking ratings.

- Spotting Fake Reviews from the Review Sequence

Methodological Framework to Establish Ground Reality between Real and Bogus Online Reviews With the advent between misleading comments, academics in recent years have been researching how to differentiate between valid and false online reviews. Within this field of study, though, the development of the ground reality has become a difficult issue. Hindering the creation of a ground reality of high standard. A false review can be a duplication or near-duplicate of an actual one on the same thing.

- Text sentiment on review helpfulness

  Text opinion has a detrimental impact on the quality of the article, and the impact is greater for positive reviews than for comparable and standard reviews. Lastly, we use a random forest method to assess the utility of the analysis, based on its numerical and textual properties. The new method has shown that the duration of analysis is the most important element in assessing the effectiveness of online feedback. For the online commentary's text mood, previous work suggested that it had an asymmetric impact on the effectiveness of the study; Deficient input has historically provided a higher degree of consumer service than positive feedback.

- Informativeness and classification thresholds by search products and experience products

Differences between search goods and experience items influence the understanding of customers about review helpfulness. We examine how review informativeness influences review helpfulness through product type. We affirm multiple classification criteria for search and experience goods. Increased classification efficiency by our suggested variables and criteria. Both the number of attributes and the total duration of attributes assess the informativeness of the test. Differences between quest items and interaction goods balance user expectations of value-added analysis. Confirm various classification criteria for search and knowledge items. Increased classification efficiency by our suggested variables and limits Both the amount of attributes and the total duration of attributes assess informativeness.

- Readers Objectives and Review Cues

With the rise of e-commerce, online user feedback have become extremely valuable sources of knowledge to help customers make purchasing decisions. However, the proliferation of online user feedback has triggered overloading of details, rendering it challenging for customers to pick credible feedback. To the development of an online shopping industry, It is essential for product reviewers to compose more valuable reviews and to make it simpler for customers to receive positive feedback by recognizing variables that decide the effectiveness of online reviews. For this study, Amazon has collected online user feedback. Com uses a crawler for Web info. Additional details on research material was also gathered by performing an overview of the sentiment for the mining review article. Our findings indicate that the utility of reviews is determined by both peripheral metrics, including review ranking and reputation of reviewers, and core signals, such as the quality of reviews. Centered on the hypotheses of double cycles, We notice that users rely on various types of knowledge for analysis, based on their intent of reading feedback: online feedback may be used to check for details or to compare alternatives. Our results provide online business owners with fresh ideas about how to handle public ratings about their websites.

- Determinants of online reviews

More and more consumers are gravitating to read product feedback online before taking buying decisions. Since there are a variety of comments written regularly that range in effectiveness, a great deal of consideration is given to calculating their helpfulness. The goal of this paper is to examine the different determinants of the importance of the reviews and also to explore the moderating effect of the form of product, That is, knowledge or the quest for products in accordance with the utility of online feedback. Results of the analysis suggest that the credibility of the reviewers, the transparency of the name of the reviewers and the scope of the examination favorably impact the effectiveness of an online review. For such determinants there are the moderating impact of product form on helpfulness. In other terms, the amount of reviews for a company and the declaration of reviewer identification have a larger impact on the utility of experienced products, whilst the credibility of reviewer, the severity of reviews and the scope of reviews are more relevant for helpfulness in the quest products. There are relationship results for average review ranking and average review profundity for a company with customer helpfulness on product sales.

- Consumer Behaviour Perspective

In order to make it easy for users to select helpful feedback from large and varied comments, several e-commerce platforms build voting mechanisms to determine their utility and filter out all recommendations depending on the outcome of the vote. To maximize the social feedback framework it is important to consider the process of user understanding of helpfulness. This can be explored in two ways. One is the website style of the analysis section.

Similar to the nature of the evaluation platform, there is a list of reviews received by prior users, about which selective summary details is included on the main web page. In order to obtain further information about a specific article, the program usually guides readers to a new page or reveals secret portions of the article text. Such a configuration of the evaluation method is directly connected to the other dimension of user reading habits and the cognitive cycle. In the first case, the general knowledge of the reviews will not involve focus, yet the special features of the individual reviews can attract the interest of readers. A more thorough analysis of the textual material of a specific article would involve attentive reading and interpretation by the user. Online product reviews are typically published by users with favorable or poor user experience with their purchases. Compared to the item summary offered by the retailers, online product comments provide more detailed qualitative input on the characteristics of the products and customer interface, representing a more actual understanding of their transactions.

- Study of Customer Review

Customer ratings on a broad variety of goods and services are increasingly accessible online. They supplement other details offered by online stores such as product specifications, consumer recommendations and tailored guidance given by automatic recommendation systems. While studies have shown the advantages of consumer feedback to an online retailer, A still unknown problem is what allows customer feedback valuable to the user in the decision-making phase. Drawing on the concept of the search and evaluation of products from the knowledge economy, we build and evaluate a model of customer service analysis. The product form limits the impact of the evaluation extremity on the utility of the analysis. Of seasoned products, Reviews with high ratings are less successful than reviews with weak ratings. For both product categories, the breadth of the research has a beneficial effect on the usefulness of the report, however the product segment restricts the influence of the nature of the report on the usefulness of the study. Test scope has a more beneficial impact on the quality of the quest products analysis than for experience goods

- Modeling and Predicting

Internet comments are a powerful tool for prospective buyers to make buying choices. Nevertheless, the sheer quantity of accessible feedback, as well as the wide differences in the content of the review, pose a significant barrier to the successful usage of the feedback, since the most valuable reviews that be lost in a big number of poor-quality reviews. While the most important feedback can be lost under a big number of poor-quality reviews. The purpose of this paper is to establish models and algorithms for predicting the utility of the feedback, and provide a framework for the discovery of the most valuable feedback for the products concerned. First, we demonstrate that the utility of a analysis depends on three critical factors. The reviewer's experience, the prose style of the article, and the timeliness of the article. Depending on the study of such variables, we are proposing a nonlinear regression model for forecasting usefulness.

CHAPTER-3
SYSTEM DESIGN

3.1 Introduction

User feedback prevail in today's e-commerce world. And not all comments have the same experience. Others are more powerful than the others, and are more important to the buying choice of a customer. Understanding what makes a valuable analysis should benefit. Unlike an study of emotions, and for a human person this role may be challenging. Two comments might seem very insightful, but one was mostly viewed as beneficial and the other was not. We would like to measure NLP's success and profound learning on such a mission. We analyze data from feedback in our issue environment, which offers an opportunity for helpfulness votes. A consumer will upvote or downvote a comment on their helpfulness if he wants to. Three-fifths, for example, would indicate three people felt the analysis was beneficial and two felt otherwise. There may be several people who have read the analysis but have not shared their opinions. We address the topic by presenting it as a classification problem. Through analysis is labelled "helpful" or "unhelpful;" different algorithms then forecast the labels. We'll use the Random Forest Bag-of-Words as a reference here. We also worked on the data simple-RNN and LSTM, and contrasted the performance.

The Long-Term Memory Network (LSTM) is a neural network that is built with Backpropagation Over Time to resolve the gradient problem that is vanishing. As such, It can be used to build a broad range of recurrent networks that can be efficiently used to solve complex sequence problems in machine learning and to produce state-of-the-art tests. Rather than neurons, LSTM networks have memory blocks that are linked across layers.

LSTMs may be used for modelling univariate time-series forecasting issues. There are issues that consist of a single collection of observations where one model has to learn from a collection of previous observations in order to determine the next value in the sequence.

Long-term memory is an artificial persistent neural network architecture used in the area of deep learning. In comparison to standard neural feed-forward networks, LSTM has input connections. Not only does it handle single data items, it even manages whole data sets.

The LSTM model will learn a function that maps a set of past observations as data for output observation. As such, it is important to turn the set of observations into several examples from which the LSTM can benefit.

Both RNNs have feedback loops on the repeat sheet. This helps them to maintain their knowledge of 'vision' over time. Nonetheless, it could be challenging to train conventional RNNs to solve problems involving long-term, time-dependent learning. That is how the gradient of the loss function decays indefinitely over time(called the gradient problem of vanishing). LSTM networks are a form of RNN that uses specific units rather than regular units. LSTM devices have a 'information reservoir' that can keep awareness in information for a long period. The gate series is used to monitor when knowledge reaches the brain, when he's published, and when he's lost. This concept allows one to learn more about longer-term dependences. The GRUs are identical to the LSTMs, but they use a simpler form. They do use a series of gates to regulate the movement of information, but they do not use different memory cells, so they use fewer gates.

The Long Short-Term Memory network is a form of a Recurrent Neural Network (RNN). RNNs are using previous time events to warn later ones. For eg, the model needs to use knowledge about past events to determine what kind of occurrence is occurring in a scene. RNNs operate well if the question needs only recent knowledge to execute the role in hand. When the problem needs long-term dependencies, RNN will have a rough time modeling it. The LSTM was built to know how to depeat long term dependencies. It remembers the information for long periods

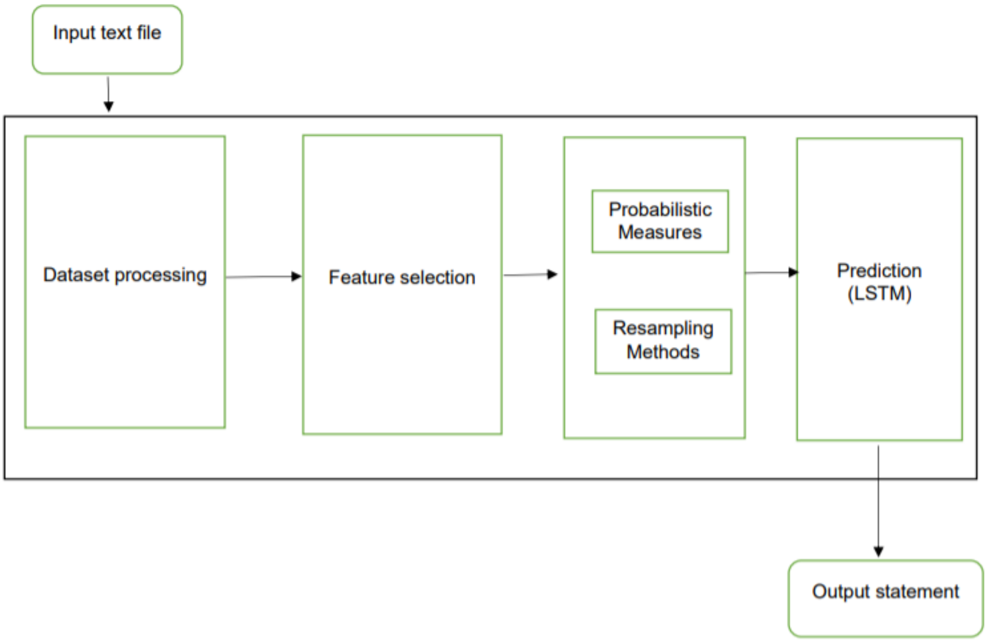
Business testing unit field teeming in the e-commerce environment of today. Nonetheless, not all unit area feedback viewed fairly. Some tend to be different and specific area unit important to a user's buying request than others. Knowing what constitutes a useful review would be helpful. This function would be completely different from helpful analyzes for an agent of an person. two Comments might appear insightful any time, but one was usually viewed as helpful and the other was not.

We would like to value information cleaning efficiency, choice of feature and LSTM on such a job. In our drawback, we prefer to analyze details regarding on-line ratings, Which gives associated input possibilities. If the consumer considers whether it is helpful or not, upvote. Consider a scenario where seven out of ten reports are helpful, meaning that seven citizens feel it is helpful and 3 find the analysis was not useful. There might well be other UN organization who saw the analysis but their opinions were not categorical. We prefer to address the {downside of the matter} by classifying it as a problem.

They appear to forecast the labels by various algorithms, taking into account any comment from the pile of all comments that each would be marked as useful or not. Here we are going to use knowledge cleaning and use the TF*IDF technique as a reference option.

3.2 System Architecture

3.2.1 Workflow Diagram:

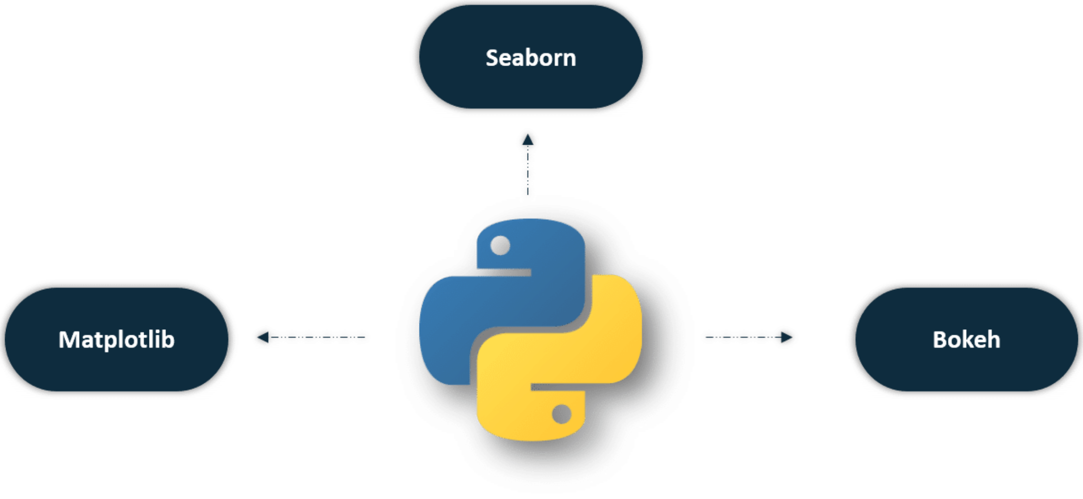


3.2.2 Description:

1. Python:

Python is a programming language that is considered to be of good standard and commonly used. Python's architecture philosophy, developed by Guido van Rossum and first published in 1991, emphasizes the readability of code by the widespread use of large whitespace. Language constructs and object-oriented techniques are being built to help programmers build fast, usable code for large and small projects. Python is quite a piece of a puzzle. It embraces several programming paradigms, including organized (particularly procedural), object-oriented and functional programming. Despite of its extensive reference library, Python is also referred to as the language used in the battery.

Python is a wonderful visualization device, too. It provides libraries such as Matplotlib, Seaborn and Bokeh with beautiful visualizations.



Python is still the most popular language for computer learning and deep learning. As a matter of fact, nowadays, all the top corporations are investing in Python in order to integrate deep learning into the back-end.

Python Line Structure

The Python coding form consists of visual lines as well as symbolic lines or sentences. The actual line of the Python scheme is a series of characters, and the end of the line completes the series of the line relative to certain other languages, such as C and C++, where the semi-colon is used to signify the termination of the expression. In the other side, the logical line consists of one or two separate lines. The use of a semi-colon in Python is not restricted or is not obligatory. The token NEWLINE is the top of the conceptual line. A conceptual line consisting solely of holes, points, or buttons is known to be a nil line, so the user ignores it.

As we showed in Python, a new line explicitly shows that a new paragraph has started. Although, Python does provide a way to split a statement into a multi-line statement or to merge several statements into a single logical line. It will further boost the readability of the file.

Here are the two methods to divide a line into two or more lines:

Explicit Line Joining

When you explicitly cross a row, we use a backward dash to break a statement into a multi-paragraph claim.

Implicit Line Joining

Statements that occur within],}, [{or) (brackets may be separated into two or three individual lines by using a backslash.

Several Comments in One line

In Python, many sentences may be rendered in the same line using a semi-colon; nevertheless, most programmers do not consider that to be a reasonable idea, because it limits the readability of the language.

Whitespaces and Indentation

Unlike other programming languages, Python uses indentation to mark a text row. According to the Python Code Development Guideline or PEP8, we 're going to have an indent scale of four.

Many programming languages have indentation for proper file layout and do not implement it. But it's mandatory in Python. That's why indentation in Python is so important.

Comments are included in each programming language to enhance the readability of the code. Likewise, in Python, the software continues to get more complex, one of the best methods to maintain the code straightforward is to use Python 's comments. It is considered a good thing to include annotations and comments in the python syntax as it renders the code much more accessible and clearer to other programmers, which is helpful while many programmers are operating on the same project at the same time.

The programming can just clarify how it functions, not why it does, because Python 's comments will do that. With the comments of Python, we will have evidence for various definitions of our own words. Comments are merely specified lines of codes that boost the readability of the text and render it self-explanatory. There are several choices for responses depending on the type of input that we would like to include with our application. Below are a variety of statements you can consider in our Python programs:

The one-line Python comments are called with a # sign. Such sentences stop at the end of the individual line, which means that all characters must continue after the # word. (which persists to the end of the line) are part of the sentence.

Python has a text string (or docstring) functionality, which is typically the first declaration used in functions and modules. Instead of being overlooked as standard statements by the Python Parser, docstrings will directly be reached at runtime using the dot operator.

It gives programmers an simple way to add short notes to any python module, function, class, method. For using the function, we use the triple quotes at beginning of the document strings or declaration and triple quotes at end of the document reaction. Docstrings can be both one-liner and one-liner.

Unlike other programming languages that allow multi-line comments, such as C, Java, and more, there is no special function for multi-line comments in Python. But this doesn't mean that it's completely difficult to render multi-line statements in Python.

There are two instances that can be made of statements that can extend over a number of lines in the Python language.

Comments on Python Block: we may use a few single-line comments for the entire segment. Usually, this type of statement is created to explain the code block that follows the statement on the website. The statement of Python Block is the only way to compose a true comment that can stretch several lines. The Python PEP8 style guide is accepted and favoured because Block comments are skipped by a Python interpreter or a parser. Nonetheless, nothing prohibits programmers from using the second 'non-real' method of writing multi-line comments in the Python, which is described below.

Using docstrings: docstrings are widely used as multi-line comment in the Python by many programmers, Because it's the closest thing to get a multi-line comment feature in Python. While it is not wrong to use the docstrings when we want to create multi-line statements, It is essential to bear in mind that there is a substantial difference between the teachings and the remarks. The comment in Python are totally overlooked by the Python parser, while the docstrings, when used in the Python framework, may be viewed at runtime.

Python Data Types

Knowledge about how data is interpreted and governed in that language is one of the most essential aspects in understanding through programming language. People are also drawn to Python because of their ease of usage and the amount of versatile methods it provides. Each of such functions is clicking intuitively.

> In Python, unlike dynamically typed languages such as C or Java, there is no need to define the data form of the element. In dynamically typed languages, such as Python, the interpreter itself determines the data form of the Python variable depending on the type of meaning applied to the variable.

Benefits of Python

- Build Universal Language

- Allow both low and high-level programming;
- Interoperability of the program
- Creation of the fastest life cycle and more efficient encoding system
- Not much hardware is required as a single container contains many data forms and form may not need its own features.
- Studying Simple and Open Source Software
- User-friendly data layout and rate
- Wide and expandable servers.
- Easy & Iot assistance

Based on the three languages listed above, we have chosen to use the Python as a programming language to create this web-based e-voting framework. The primary reason for this is

- Simple to know, it can also be used by unexperienced programmers.
- Graphic mode
- Wide and comprehensive regular collections
- Python applications are really similar to pseudo-code systems. This makes it easy and must-have for beginner programmers because of its intuitive nature as compared to C++ Java, Perl, and so on.

2.Scikit Learn Package:

Scikit-learn (formerly scikits.learn, and also known as sklearn) is a Python programming language free machine learning software. This supports various classification, regression and clustering algorithms, including gradient boosting, vector support, k-means, random forests and DBSCAN, and is built to work with NumPy's computational and science Python libraries.

3. NumPy Package:

NumPy is a Python programming language library that provides support for small, multi-dimensional arrays and matrices, along with a large variety of high-level mathematical functions to operate on these arrays. Numeric, the ancestor to NumPy, was initially created by Jim Hugunin and input from a number of other players. In 2005, Travis Oliphant created NumPy, incorporating the functionality of the opposing Numarray in Numeric, with extensive modifications. NumPy is an open source project that has many contributors to it.

4. Pandas:

In computer science, pandas are a python programming language software library for storing and analyzing results. Specifically, it involves the statistical models and methods for the study of numerical tables and time series. It is a free software that is released under a three-class BSD licenses. The name derives from the term "panel tests," an econometric definition for data sets including observations over various time periods for the same individuals.

- Data Frame item for data analysis with an automated indexing feature.
- Ways of reading and writing data from in-memory data systems and file formats;
- Code synchronization and automated retrieval of incomplete evidence;
- Reshaping and pivoting of data sets.
- Label-based splitting, graceful indexing, and a subset of broad data sets.Insert and delete column data structure.
- Engine-by-engine type that executes split operations on data sets.
- Select the data you intend to merge and enter.
- Indexing the hierarchical axis when dealing with high-dimensional data in a low-dimensional data system.
- Time series functionality: Date range creation and frequency transfer, moving window data, moving window linear analysis, date change and time lag.
- Allows the analysis of the tests.

5. NLTK-Python Natural Language Translation

Python NLTK Natural Language Analysis is one of the leading methods to work with human language data and Python NLTK is used in natural language research. The Natural Language Toolkit is basically an acronym for the NLTK. Throughout this post, you will learn how to tokenize the details, NLTK is intended to support NLP or related research and guidance, Like computational linguistics, cognitive sciences, artificial intelligence, information analysis, and computer learning. NLTK has been commonly used as a instructional tool, as an independent research system, and as a medium for prototyping and improving test frameworks.

5. bootstrap

bootstrap is a free , open source CSS platform for sensitive, mobile-first, front-end web creation. It includes CSS-based and JavaScript-based design models for typography, shapes, icons, navigation, and other device functions. bootstrap is an open source toolkit for creating Mark-up language, CSS, and JavaScript. Quickly test your ideas or build your whole project with our Sass variables and mixers, Flexible grid layout, robust pre-built models and powerful jQuery plug-ins.

6. Genism

Gensim is the Python System for Simulation, Record Indexing and Connection Extraction for Big Businesses. The main group is the Natural Language Processing (NLP) and Information Retrieval (IR) Community. Both algorithms are memory independent w.r.t. corpus scale (can handle inputs greater than RAM, distributed, out-of-core). Quick linking your own input corpus / data source (trivial streaming API) to other VectorSpace algorithms (trivial transformation API) Efficient multi core implementation of similar algorithms, such as Latent Semantic Analysis (LSA / LSI / SVD), Latent Dirichlet Allocation (LDA), Random Projections (RP), Hierarchical Dirichlet Process (HDP) or Word2vec deep learning.

7. Spacy

SpaCy is an integrated natural language processing framework in Python and Cython. It is based on the very latest science and has been developed from day one to be used in real goods. SpaCy comes with pre-trained mathematical models and term vectors and officially supports 50 + languages tokenization. It features state-of-the-art speed, convolutionary neural network models for labeling, parsing and entity recognition, and fast deep learning integration. It's commercial open-source program that was published under the MIT license.

9 **Selecting the Online Technology Platform**

Frameworks provides a framework for development of software. This allows it easy for web-based computer programmers to create reliable, practical and workable enterprise-wide mobile apps. Automate the delivery of repetitive tasks or modifications for different operations, increase development time and test time, and enable programmers to concentrate more on code logic instead of daily work.

Python framework classified as two types:

1. Full-Stack System

   Full-stack solutions offer comprehensive resources to engineers, including essential elements such as structural clearance, device generators and program architecture.

   - django
   - flask
   - turboGears
   - zope2
   - cubicWeb
   - grok
   - pylon

2. Non Full Stack

a. Non full stack applications do not offer additional features and highlights to customers. Designers have to provide a variety of coding and other things manually.

b. Micro Frame work

   - Miniaturized scale systems are the lightweight , compact and quick to operate. They are laconic, and they have clear paperwork. URL forwarding is often a full networking process. Miniaturized scale systems using WSGI which run via HTTP request or response. We are a reasonable option for small undertakings or part of the larger undertaking.
   - cherryPy
   - flask
   - bottle
   - pyramid
   - bobo

c. Asynchronous Framework

   In cases where the level of demand involves a large amount of research or an project needs lengthy periods of response, a non-concurrent solution would be efficient.

We have agreed to use the Flash Microframework for our e-Voting program.

Flask is a lightweight, simple-to-use web application platform for Python WSGI. It is developed to allow fast and easy launch, with the potential to scale to complicated applications. It began as a simple tool and jinja wrapper, and turned out to be one of the most-Python web application frameworks.

This is perceived as more Pythonic than the Django web framework on the ground because, in comparable situations, the same Flash operating program is quite succinct.

Flask gives recommendations, but does not meet with the conditions or specifications of the product. It's up to the developer to pick the devices and resources they choose to use. There are several network extensions that make up the network, including a simple new feature.

Apps that utilize the Flask program include Pinterest Facebook and the Flask Community website itself.

Important features that make up the microframework of our Choosing microframework

- Simple to start
- No Boilerplate code required
- built-in development server and quick debugger
- offers usability, versatility and fine-grained power.
- Design of Jinja2

3.3 System Requirements:

Hardware

- Intel Xeon Processor
- 200 GB Free Hard disk
- 16 GB Ram
- Internet Connectivity

Software

- Eclipse
- Python
- Numpy
- Sci-learn
- NLTK
- Web Technologies

- Bootstrap
- Genism
- Spacy

3.3 Summary

It uses input file from the consumer instead performs dataset processing, Data sets which have some apparent errors render the results of processed data in Dataset Processing and Feature Selection Processes and finds the Structural features of the analysis often abuse various types of content frequency steps. (individual tokens, words, and paragraphs) and then it is used in Model selection, In this step of selecting one of the models as the final model that solves the question and ultimately using it to simulate LSTM-based judgment in the prediction node, Long-term memory (LSTM) is an artificial recurring neural network (RNN) design used in the field of deep learning. The LSTM network is a persistent neural network that is equipped utilizing Backpropagation Over Time. That's over distance. The rising LSTM unit consists of a cell, an input gate, an output gate, and a missing gate. The three gates control the flow of knowledge into and out of the container. LSTM networks are well adapted for detection, interpretation and forecasting based on time series results, since there might be unexplained time lags between important events in time series.

The GRUs are identical to the LSTMs, but they use a simpler form. These do have a series of gates to monitor knowledge transfer, but do not use different memory cells, and have fewer gates. It can be used to solve complicated machine learning series problems and to produce state of the art results. LSTM networks have memory blocks instead of neurons connected through layers.

LSTMs may be used to model univariate time series forecasting problem. There are problems consisting of a specific series of observation where a model has to benefit from a collection of previous observations in order to evaluate the next value in the chain. Long short-term memory is an abstract memory, Persistent neural network design used in deep learning. Unlike normal neural feed-forward networks, the LSTM has an input relation. It applies not only to individual data artifacts but also to the entire data collection.

They appear to forecast the labels by various algorithms, taking into account any comment from the pile of all comments that each would be marked as useful or not. Here we are going to use knowledge cleaning and use the TF*IDF technique as a reference option.

On this project, we would like to evaluate the progress and meaningful learning of the NLP. In the world of our problems, we review input results, which allows us the chance to vote for support. When a customer decides to, he will upvote or downvote a message on their helpfulness. For eg, three-fifths may mean three people considered the study to be helpful and two thought otherwise. A variety of people might have read the article but have not voiced their thoughts. We address the subject by posing it as a question of classification. Research is called "helpful" or "unhelpful;" The marks are then estimated by other algorithm. Here we are going to use Random Forest Bag-of-Words as a guide. We also focused on simple-RNN and LSTM simulations, similar to efficiency measures.

The LSTM model involves understanding a method that maps a series of past observations as data for output observation. As such, the sequence of results includes a translation into a number of instances from which the LSTM can gain access.

CHAPTER 4
MODULE DISCRIPTION

4.1 Introduction

Data cleaning and function selection would be used with TF*IDF, basic Recurrent neural network (RNN) and long short term memory (LSTM) with specific parameters to choose best output pattern.

Data has several forms; all the data is messy. There are numerous data formats like missing data, unstructured data and data without regular structure. We need techniques to cleanse to boost the standard. Three steps in a row visualizes the data processing pipeline, from cleanup to machine learning to information visualization.

The standard of results from you get from information depends on the expiration of that data. Data cleansing provides a great history in databases and a key step that are called as extract, transform, load (ETL). The dataset retrieving, accessing and cleaning is described in the experimentation.

The recurrent neural networks are the deep neural network designed for the text classification from the experiences. The experiences are gained from the learnings. These learnings are collected from the protype of the hidden layers in the neural networks where the gradient problem outcomes but we handled by carefully executing the review and to the maximum extent the efficiency is reduced. The LSTM neural network are proposed to find the solution for the NLP problems. The LSTM neural networks work on the principle of weighted sum where it is used as activation that make the network function

Examining and forecasting utility of LSTM algorithm related feedback. This method contains four key modules: module for data set collection, function selection, module for model selection, and module for prediction.

4.2 Dataset Processing

Data arrives in several different ways, chaotic for everything. If we're thinking about lost data, unstructured data or data that needs normal structure, you need methods of cleaning up data before you can process it to enhance its consistency.

Data sets can be accessible in several ways, but most of them are processed as delimited text files. Such data sets, as seen in previous instances, delimit their fields with a symbol, usually a comma, but in some cases with white line (line, button, etc.). These raw data sets are especially vulnerable to error, since they lack any details showing their context and therefore allow data scientists to manually analyze the data collection.

So-called "self-description formats" will significantly improve our capacity to accurately interpret the details. That involve the XML and JSON formats. These data formats require the data to be inserted into the metadata such that it can be completely self-described inside one package. They often require complex data formats that are more difficult to explain with simple flat text files (such as data or relationship variant arrays inside the data).

We obtained a subset of data for product analysis. We concentrate on the three types of goods. Checking our strategy we use the first three data groups. Remember that additional datasets are used to create word embedding models of various vector lengths from the variable length study article.

The data sets that have such obvious errors make the results of the data processed somewhat questionable. Loss findings contribute to missing details or inaccurate conclusions which can result in incorrect tests. Hence, data cleaning is a crucial phase in the data processing system. Data can come from different sources, too. While-source can be accurate in isolation, a mixture of data can involve accuracy and quality in processing. For eg, one data collection might have a specific unit of measurement for a given area than another, allowing it to be standardized.

Data Processing is the activity of transforming data from a specific medium to a far more functional and acceptable type, i.e. To make it more useful and more insightful.

Using Machine Learning techniques, mathematical modelling and statistical information, the whole procedure can be automated.

Removal of unnecessary results Which involves removing duplicate or null or obsolete values from the dataset. Duplicate findings more often arise through data processing, so unnecessary results are those that do not directly answer the particular question you are attempting to solve.

Errors that arise by calculation, transfer of data or other similar situations are referred to as systematic errors. Structural errors include typos in the name of the features, the same feature with a different name, mislabelled classes, i.e. various groups that should be the same or inaccurate capitalization.

Outliers that trigger problems with some types of models. Linear regression models, for instance, are less robust than decision tree models. Generally speaking, we do not reject outsiders until we have a valid reason to remove them. Eliminating them sometimes reduces the efficiency, often it doesn't. So, one must have a reasonable excuse to exclude the observer, such as ambiguous figures that are impossible to be part of the actual evidence.

Managing lost data Missing data is a challenging machine learning problem. We can't completely overlook or erase the missing information. They ought to be handled with care, since they could be a symptom of something important.

Pre-processing involves the changes that have been made to our data before it is put into an algorithm. Data Pre-processing is a method used to transform raw data into a clean data package. In other terms, if data is collected from multiple sources, it is processed through a raw form which can not be analysed.

4.3 **Feature Selection module**:

App Selection is a process in which you automatically or manually pick the features that most contribute to your predicted feature or results that you're interested in. Using un related features in the data will minimize the precision of the model and enable the model to learn from irrelevant features.

For the most part, the technical characteristics of the analysis take account of various forms of material level steps. Our insight into the usage of structural features is to express the meaning of different sentences, sentences and paragraphs. The conceptual characteristics used in our analyses include the length of the study, the number of clauses, the number of letters, the number of all capitalized terms and the number of question marks. The length of the study tells one intuitively whether the test is relevant or not. Commonly, shorter sentences are less insightful than longer remarks. Both the capitalized words are kept as we think they might be of great interest to some of the details of the analysis. In comparison, multiple question marks in the text are generally not helpful, because they may call for clarification rather than for honest suggestions.

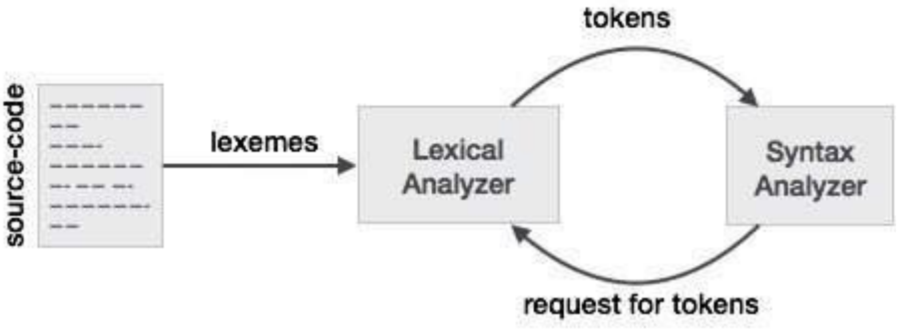
Both unigram and bigram characteristics are calculated for each sample using the TF-IDF method, which gives more weight to less frequent than high-frequency words. We incorporate all lexical, logical and textual characteristics and use them as different roles in the classification process.

The TF*IDF algorithm is used to calculate the keyword in every text and to give value to the keyword depending on the amount of times it occurs in the document. More precisely, it checks the importance of the term across the world, which is referred to as corpus.

In natural language processing and other linguistic studies, a crucial criterion is to figure out how understandable a document is until some pre-processing and standardization is carried out. There are common ways to figure out how often a paragraph or an whole document should be understood. The readability rating shows the complexity degree of comprehension of a given sentence or paragraph

The Lexical Review is the compiler's first step. It use to takes the updated source code from the language pre processors which are written in the form of phrases. The lexical analyser splits the text into a sequence of tokens by deleting a whitespace or a statement from the source code.

If the token is called null by the lexical analyser, an error may occur. The lexical analyser works in the same way as the syntax analyser. It reads the character streams from the source code, checks the correct tokens and, where necessary, passes the details to the syntax analyser.

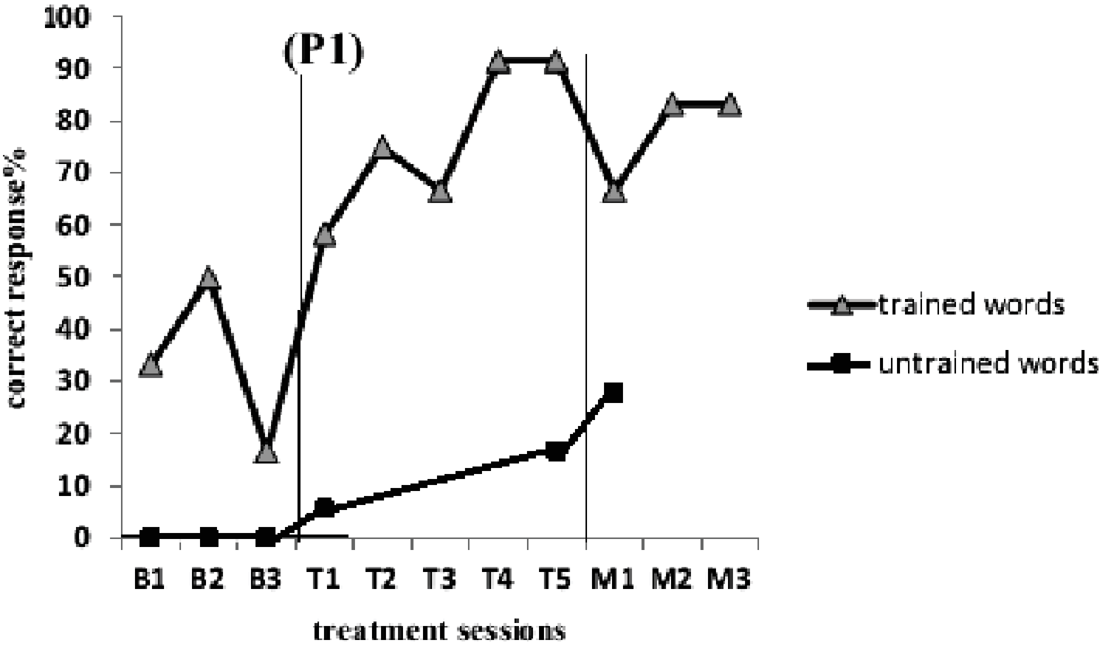


Lexical analysis

Lexemes is referred to as the alphanumeric series of characters in a symbol. There are several predefined standards to be accepted as a valid symbol for every lexem. The rules are described by the rules of grammar, by way of a sequence. A pattern indicates what a token may be, and such patterns are represented by standard expressions.

Structured modelling is a software development methodology that utilizes graphical diagrams to create and view device requirements that are readily interpreted by users. Such diagrams explain the measures that need to be taken and the details required to satisfy the design purpose of a specific program.

Semantic features are the essential semantic components of meaning for every lexical object. An individual semantic feature constitutes one component of the intension of a word, which is the inherent meaning or concept evoked. It is proposed that the linguistic sense of a phrase derives from similarities and major variations with certain words. Semantic features allow linguistics to understand how terms that share similar features can be part of the same semantic domain.



Sample semantic analysis

In order to obtain optimal outcomes from the paradigm applicable to Machine Learning initiatives, the data structure must be in the right way. Described Machine Learning includes information in a particular format, e.g. Random Forest algorithm does not consider any null values, so it is essential to run random forest algorithm null values from the original raw data collection.

4.4 **Model selection module:**

Growing feature selection module in the Machine Learning Studio (classic) uses datasets as inputs. The module then applies well-known statistical methods to data columns which are given as inputs. Output is a collection of indicators that will help you determine the columns that provide the highest value for details.

Form discovery is a critical concern in many fields of research and engineering. In the framework of a data collection, model selection includes choosing a mathematical model that better fits the characteristics and regularities of the results.

Selecting a model is the method of selecting a product final machine learning model from the collection of applicant machine learning models for the training data set

Selecting a model is the method of selecting a product one of the models to solve the question as the final one. Plan compilation differs from the assessment of the project.

For starters, we evaluate or examine candidate models in order to choose the correct one, so this is model selection. And once the product has been selected, it will be checked to decide how well it will perform in general; this is a product assessment.

Some algorithms require thorough data analysis in order to properly address the problem structure of the learning algorithm.

We then need to move a step forward and understand model selection as a selection method between product creation pipelines. There are two key types of techniques to estimate the optimal product selection case:

Probabilistic Measures: Pick a model by in-sample error and difficulty.

Resampling Methods: Use a standard utilizing an approximate out-of-sample mistake.

4.5 **Prediction module:**

Prediction 'applies to the performance of an algorithm after it has been educated in a historical dataset and introduced to fresh data while determining the probability of a given result, such as whether or not the consumer is going to churn.

The long Short Term Memory Network (LSTM) is a recurrent neural network that is fitted with Backpropagation Over Time and overcomes the gradient problem of extinction. As such, it can be used to build massive repeated networks that can be used efficiently to solve complicated sequence problems in machine learning and to produce state-of-the-art tests. Except the cortex, LSTM networks provide layer-connected memory blocks.

LSTMs will be used to model univariate time series forecasting issues. There are problems that comprise of a particular set of observations, and a model is needed to take account of the selection of previous observations in order to determine the next element in the sequence. The LSTM model will learn a system that maps a set of past observations as inputs for output observation. As such, a set of findings has to be converted into a variety of instances that the LSTM may benefit from.

The neural network of the long short term memory is a form of recurrent neural network (RNN). RNNs use past-time events to alert future occurrences. For example, in order to decide what kind of thing is going on in a picture, the designer needs to use information from past events. RNNs perform well if the problem just requires fresh information in order to serve the current position. Where a long-term dependence solution is required, RNN should not have modelled it. The LSTM was built to consider long-term dependency. It's been keeping the details for a long time.

CHAPTER-5

SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

As the project involves semi-supervised learning, the platforms used have to support the necessary packages involved. A laptop or device that has enough RAM will accommodate package loads.

The packages are listed as follows:

- Pandas
- Time
- Stream listener
- Numpy
- Matplotlib
- Spacy
- Gensim

From the sklearn package, the sub-packages SVM, train_test_split, confusion matrix and unique_labels.

Python is regarded as a high-level, general-purpose language with packages that can be imported in-built. Most of the Python implementations (including CPython) include a read eval print loop (REPL) that allows the user to function as a command line interpreter (CLI) from which the user enters the statements sequentially and receives the answers immediately. Many plugins, including IDLE, IPython, incorporate additional functionality such as enhanced auto-completion, session state control, and syntax highlighting.

There are web browser-based IDEs, as well as conventional desktop-based interactive programming environments; Sage Math (designed to create Python-based science and math programs); PythonAnywhere, a browser-based Hosting and IDE platform; and Canopy IDE, a commercial Python IDE focusing on scientific computing.

The foundation of garbage collection and dynamics can be used to explain python as a script. It promotes through programming paradigms, including organized (especially procedural), object oriented and functional programming. Thanks to a broad digital collection, Python is often known as the "batteries included" language.

Most of the operating systems provide Python-accessible interpreters. CPython, an open source software framework, is being created and sponsored by a worldwide programmer group. The Python Software Foundation, a non-profit organization, provides and oversees support for the development of Python and CPython.

Project Jupyter is non-profit organization founded for the creation of open-source applications, open-source principles and shared programming resources through multiple languages.

5.2 Overview of the platform

The development and dissemination of live data, estimates, visualizations and text documents can be created by Jupyter Notebook which is open source software application. Project Jupyter is non profit project founded for the creation of open-source applications, open-source principles and shared programming resources through various languages.

Jupyter Notebooks is the IPython spin-off program utilized for the IPython Notebook framework itself. The name, Jupyter, comes from the three programming languages it supports: Julia, Python, and R. Jupyter sails with the base of the IPython, which can then be used to write Python programs, although there are also over 100 other kernels that can be used as well.

The notebook applies the console-based approach to interactive computing in a qualitatively specific manner , providing a web-based interface for monitoring the whole computational process: application creation, recording and implementation, and data sharing. The notebook Jupyter combines two components:

Notebook documents: a summary of all the details used in the web application, including mathematical inputs and outputs, informative language, data, photographs and rich media description of items.

Online application is a browser based interactive document authoring method that includes detailed information, mathematics, computation, and their rich media production.

The functionality of Jupyter Notebook help to establish immersive computing beyond the traditional console method.

The key features of the Jupyter Notebook include:

- JavaScript editing in-browser, with automated highlighting syntax, indenting, and completion / introspection list.

● Caliber to perform the algorithm from the window, with the effect of the calculation added to the application that generated it.

● Display the statistical results using rich media representations such as Text, LaTeX, PNG, SVG, etc. Publication – For example, reliable data from the matplotlib database may be accessed online.

- Use the Markdown mark-up language in the rich text editing client, which will include feedback on the script, is not restricted to plain text.
- ● Capacity to easily use mathematical notation using LaTeX in the markdown containers and made natively by MathJax.

Notebook documents contain digital application inputs and outputs, as well as extra text preceding the code but not planned for implementation. Notebook files may also act as a full session computational archive, Interlacing the executable code with insightful vocabulary, mathematics, and rich description of the corresponding objects. Such documents are locally JSON files that are filled with the.ipynb suffix.

Notebooks may converted to a variety of static formats utilizing the nbconvert method, including HTML (for blog articles, for example), restructured Text, Latex, pdf, and slide displays. Furthermore, the Jupyter Notebook Viewer (nbviewer) will represent any notebook text with the ".ipynb" extension. It software loads a notebook URL folder and makes a unchangeable web page. The findings can instead be transmitted to peers or as a public blog post, without allowing any users to install the Jupyter Notebook themselves.

In turn, Nbviewer is literally nbconvert as a web server, meaning that static transformations can be done using nbconvert without depending on nbviewer.

Since Jupyter is used in a web browser, certain people are rightly worried about accessing confidential data. However, if the regular installation guidelines have been followed, Jupyter is currently operating on the owner 's device. If the URL in the address bar begins with the "http:/localhost:" or "http:/127.0.0.1:," the domain is the owner's device. Jupyter does not submit user data somewhere else — and as an open source, many users may test if privacy is genuine.

The standard workflow in a notebook is very close to the normal IPython session, with the addition that in-place cell editing is necessary many times until the required results are achieved, instead of needing to re-run different scripts with the magic command percent run.

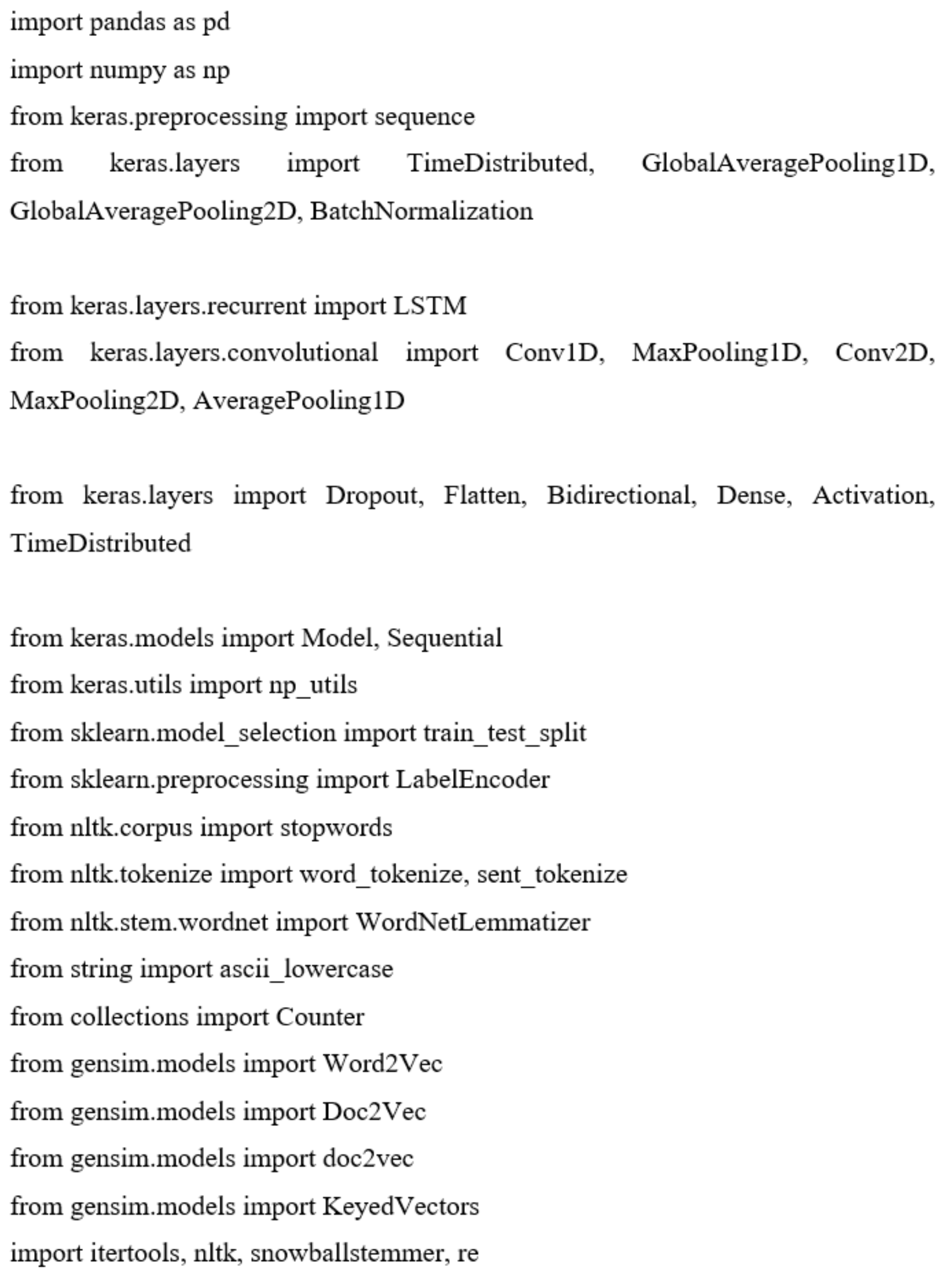
The analytical problem is typically done in sections, organizing related ideas into cells and going on until the previous pieces function properly. It is much more convenient for interactive exploration than to break up a computation into scripts that, as was previously required, must be run together, particularly if parts of them take a long time to run.

One of the main features of the Jupyter notebook is its ability to view plots that are the performance of running code cells. To support this feature, the IPython kernel is programmed to run smoothly with the matplotlib plotting library. The fundamental incorporation of the plotting collection is a cornerstone of the kernel.

The fingerprint of of trustworthy notebook is retained to avoid untrusted code from operating on behalf of users when notebooks are opened. This fingerprint is verified by the server notebook when a notebook is accessed. If no similar fingerprints are detected, Javascript and HTML data will not be seen until the cells are regenerated by re-executing them.

5.3 Implementation Details

5.3.1 Sample Coding



```
import pandas as pd
import numpy as np
from keras.preprocessing import sequence
from keras.layers import TimeDistributed, GlobalAveragePooling1D,
GlobalAveragePooling2D, BatchNormalization

from keras.layers.recurrent import LSTM
from keras.layers.convolutional import Conv1D, MaxPooling1D, Conv2D,
MaxPooling2D, AveragePooling1D

from keras.layers import Dropout, Flatten, Bidirectional, Dense, Activation,
TimeDistributed

from keras.models import Model, Sequential
from keras.utils import np_utils
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.stem.wordnet import WordNetLemmatizer
from string import ascii_lowercase
from collections import Counter
from gensim.models import Word2Vec
from gensim.models import Doc2Vec
from gensim.models import doc2vec
from gensim.models import KeyedVectors
import itertools, nltk, snowballstemmer, re
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_extraction.text import CountVectorizer
import spacy
from sklearn.metrics import accuracy_score
from keras.callbacks import ModelCheckpoint
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import train_test_split
LabeledSentence = doc2vec.LabeledSentence

class LabeledLineSentence(object):
    def __init__(self, sources):
        self.sources = sources
        flipped = {}
        # make sure that keys are unique
        for key, value in sources.items():
            if value not in flipped:
                flipped[value] = [key]
            else:
                raise Exception('Non-unique prefix encountered')

    def __iter__(self):
        for source, prefix in self.sources.items():
            with utils.smart_open(source) as fin:
                for item_no, line in enumerate(fin):
                    yield LabeledSentence(utils.to_unicode(line).split(), [prefix + '_%s' %
item_no])

    def to_array(self):
        self.sentences = []
        for source, prefix in self.sources.items():
            with utils.smart_open(source) as fin:
                for item_no, line in enumerate(fin):
                    self.sentences.append(LabeledSentence(utils.to_unicode(line).split(),
[prefix + '_%s' % item_no]))
        return self.sentences

    def sentences_perm(self):
        shuffled = list(self.sentences)
        random.shuffle(shuffled)
        return shuffled

data = pd.read_csv("deceptive-opinion.csv")
data.head()
data['polarity'] = np.where(data['polarity']=='positive', 1, 0)
data['deceptive'] = np.where(data['deceptive']=='truthful', 1, 0)

def create_class(c):
    if c['polarity'] == 1 and c['deceptive'] == 1:
        return [1,1]
    elif c['polarity'] == 1 and c['deceptive'] == 0:
        return [1,0]
    elif c['polarity'] == 0 and c['deceptive'] == 1:
        return [0,1]
    else:
        return [0,0]

def specific_class(c):
    if c['polarity'] == 1 and c['deceptive'] == 1:
        return "TRUE_POSITIVE"
    elif c['polarity'] == 1 and c['deceptive'] == 0:
        return "FALSE_POSITIVE"
    elif c['polarity'] == 0 and c['deceptive'] == 1:
        return "TRUE_NEGATIVE"
    else:
        return "FALSE_NEGATIVE"

data['final_class'] = data.apply(create_class, axis=1)
data['given_class'] = data.apply(specific_class, axis=1)
data
Y = data['given_class']
Y
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
encoded_Y
dummy_y = np_utils.to_categorical(encoded_Y)
textData = pd.DataFrame(list(data['text']))

stemmer = snowballstemmer.EnglishStemmer()
stop = stopwords.words('english')


stop.extend(['may','also','zero','one','two','three','four','five','six','seven','eight','nine','t
en','across','among','beside','however','yet','within']+list(ascii_lowercase))

stoplist = stemmer.stemWords(stop)
stoplist = set(stoplist)
stop = set(sorted(stop + list(stoplist)))

textData[0].replace('[!"#%\'()*+,-./:;<=>?@\[\]^_`{|}~1234567890’”“‘\\\]','
',inplace=True,regex=True)

wordlist = filter(None, " ".join(list(set(list(itertools.chain(*textData[0].str.split('
')))))).split(" "))
```

```
  data['stemmed_text_data'] = [' '.join(filter(None,filter(lambda word: word not in stop,
  line))) for line in textData[0].str.lower().str.split(' ')]
  minimum_count = 1

  str_frequencies =
  pd.DataFrame(list(Counter(filter(None,list(itertools.chain(*data['stemmed_text_dat
  a'].str.split(' '))))).items()),columns=['word','count'])

  low_frequency_words = set(str_frequencies[str_frequencies['count'] <
  minimum_count]['word'])

  data['stemmed_text_data'] = [' '.join(filter(None,filter(lambda word: word not in
  low_frequency_words, line))) for line in data['stemmed_text_data'].str.split(' ')]

  data['stemmed_text_data'] = [" ".join(stemmer.stemWords(re.sub('[!"#%\'()*+,-
  ./:;<=>?@\[\]^_`{|}~1234567890’”“′‘\\\]',' ', next_text).split(' '))) for next_text in
  data['stemmed_text_data']]

lmtzr = WordNetLemmatizer()

w = re.compile("\w+",re.I)

def label_sentences(df, input_point):
  labeled_sentences = []
  list_sen = []
  for index, datapoint in df.iterrows():
    tokenized_words = re.findall(w,datapoint[input_point].lower())
    labeled_sentences.append(LabeledSentence(words=tokenized_words,
tags=['SENT_%s' %index]))
    list_sen.append(tokenized_words)
  return labeled_sentences, list_sen
```

```
def train_doc2vec_model(labeled_sentences):
   model = Doc2Vec(min_count=1, window=9, size=512, sample=1e-4, negative=5,
workers=7)
   model.build_vocab(labeled_sentences)
   pretrained_weights = model.wv.syn0
   vocab_size, embedding_size = pretrained_weights.shape
   model.train(labeled_sentences, total_examples=vocab_size, epochs=400)
   return model

textData = data['stemmed_text_data'].to_frame().reset_index()
sen, corpus = label_sentences(textData, 'stemmed_text_data')
corpus
sen

doc2vec_model = train_doc2vec_model(sen)
doc2vec_model.save("doc2vec_model_opinion_corpus.d2v")
doc2vec_model = Doc2Vec.load("doc2vec_model_opinion_corpus.d2v")

 tfidf1 = TfidfVectorizer(tokenizer=lambda i:i, lowercase=False, ngram_range=(1,1))
 result_train1 = tfidf1.fit_transform(corpus)
 tfidf1

 tfidf2 = TfidfVectorizer(tokenizer=lambda i:i, lowercase=False, ngram_range=(1,2))
 result_train2 = tfidf2.fit_transform(corpus)

 tfidf3 = TfidfVectorizer(tokenizer=lambda i:i, lowercase=False, ngram_range=(1,3))
 result_train3 = tfidf3.fit_transform(corpus)

 svd = TruncatedSVD(n_components=512, n_iter=40, random_state=34)
 tfidf_data1 = svd.fit_transform(result_train1)
 tfidf_data2 = svd.fit_transform(result_train2)
 tfidf_data3 = svd.fit_transform(result_train3)
```

```
nlp = spacy.load('en')
temp_textData = pd.DataFrame(list(data['text']))

overall_pos_tags_tokens = []
overall_pos = []
overall_tokens = []
overall_dep = []

for i in range(1600):
    doc = nlp(temp_textData[0][i])
    given_pos_tags_tokens = []
    given_pos = []
    given_tokens = []
    given_dep = []
    for token in doc:
        output = "%s_%s" % (token.pos_, token.tag_)
        given_pos_tags_tokens.append(output)
        given_pos.append(token.pos_)
        given_tokens.append(token.tag_)
        given_dep.append(token.dep_)

  overall_pos_tags_tokens.append(given_pos_tags_tokens)
  overall_pos.append(given_pos)
  overall_tokens.append(given_tokens)
  overall_dep.append(given_dep)

count = CountVectorizer(tokenizer=lambda i:i, lowercase=False)
pos_tags_data = count.fit_transform(overall_pos_tags_tokens).todense()
pos_data = count.fit_transform(overall_pos).todense()

tokens_data = count.fit_transform(overall_tokens).todense()
```

```
dep_data = count.fit_transform(overall_dep).todense()
min_max_scaler = MinMaxScaler()
normalized_pos_tags_data = min_max_scaler.fit_transform(pos_tags_data)
normalized_pos_data = min_max_scaler.fit_transform(pos_data)
normalized_tokens_data = min_max_scaler.fit_transform(tokens_data)
normalized_dep_data = min_max_scaler.fit_transform(dep_data)

final_pos_tags_data = np.zeros(shape=(1600, 512)).astype(np.float32)
final_pos_data = np.zeros(shape=(1600, 512)).astype(np.float32)

final_tokens_data = np.zeros(shape=(1600, 512)).astype(np.float32)
final_dep_data = np.zeros(shape=(1600, 512)).astype(np.float32)
final_pos_tags_data[:normalized_pos_tags_data.shape[0],:normalized_pos_tags_da
ta.shape[1]] = normalized_pos_tags_data

final_pos_data[:normalized_pos_data.shape[0],:normalized_pos_data.shape[1]] =
normalized_pos_data
final_tokens_data[:normalized_tokens_data.shape[0],:normalized_tokens_data.shap
e[1]] = normalized_tokens_data
final_dep_data[:normalized_dep_data.shape[0],:normalized_dep_data.shape[1]] =
normalized_dep_data

maxlength = []
for i in range(0,len(sen)):
   maxlength.append(len(sen[i][0]))

print(max(maxlength))

def vectorize_comments(df,d2v_model):
   y = []
   comments = []
   for i in range(0,df.shape[0]):
        label = 'SENT_%s' %i
        comments.append(d2v_model.docvecs[label])
    df['vectorized_comments'] = comments
    return df

textData = vectorize_comments(textData,doc2vec_model)
print (textData.head(2))

X_train,          X_test,          y_train,          y_test          =
train_test_split(textData["vectorized_comments"].T.tolist(),
                                                  dummy_y,
                                                  test_size=0.1,
                                                  random_state=56)

X = np.array(textData["vectorized_comments"].T.tolist()).reshape((1,1600,512))
y = np.array(dummy_y).reshape((1600,4))

X_train2 = np.array(X_train).reshape((1,1440,512))
y_train2 = np.array(y_train).reshape((1,1440,4))

X_test2 = np.array(X_test).reshape((1,160,512))
y_test2 = np.array(y_test).reshape((1,160,4))

Xtemp = textData["vectorized_comments"].T.tolist()
ytemp = data['given_class']
training_indices = []
testing_indices = []

skf = StratifiedKFold(n_splits=10)
skf.get_n_splits(Xtemp, ytemp)

for train_index, test_index in skf.split(Xtemp, ytemp):
  training_indices.append(train_index)
  testing_indices.append(test_index)

def extractTrainingAndTestingData(givenIndex):
  X_train3 = np.zeros(shape=(1440, max(maxlength)+10, 512)).astype(np.float32)
  Y_train3 = np.zeros(shape=(1440, 4)).astype(np.float32)
  X_test3 = np.zeros(shape=(160, max(maxlength)+10, 512)).astype(np.float32)
  Y_test3 = np.zeros(shape=(160, 4)).astype(np.float32)

  empty_word = np.zeros(512).astype(np.float32)

  count_i = 0
  for i in training_indices[givenIndex]:
    len1 = len(sen[i][0])
    average_vector1 = np.zeros(512).astype(np.float32)
    average_vector2 = np.zeros(512).astype(np.float32)
    average_vector3 = np.zeros(512).astype(np.float32)
    for j in range(max(maxlength)+10):
      if j < len1:
        X_train3[count_i,j,:] = doc2vec_model[sen[i][0][j]]
        average_vector1 += result_train1[i, tfidf1.vocabulary_[sen[i][0][j]]] *
doc2vec_model[sen[i][0][j]]
        average_vector2 += result_train2[i, tfidf2.vocabulary_[sen[i][0][j]]] *
doc2vec_model[sen[i][0][j]]
        average_vector3 += result_train3[i, tfidf3.vocabulary_[sen[i][0][j]]] *
doc2vec_model[sen[i][0][j]]
      elif j == len1:
        X_train3[count_i,j,:] = tfidf_data1[i]
      elif j == len1 + 1:
        X_train3[count_i,j,:] = tfidf_data2[i]
      elif j == len1+2:
        X_train3[count_i,j,:] = tfidf_data3[i]
```

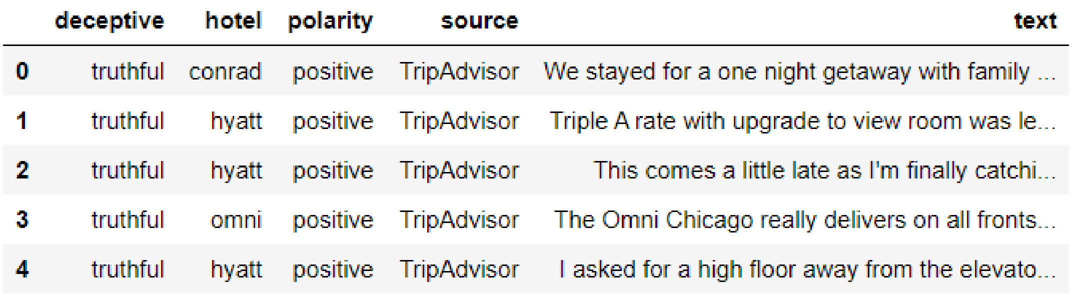
```
     elif j == len1+3:
       X_train3[count_i,j,:] = average_vector1
     elif j == len1+4:
       X_train3[count_i,j,:] = average_vector2
     elif j == len1+5:
       X_train3[count_i,j,:] = average_vector3
     elif j == len1+6:
       X_train3[count_i,j,:] = final_pos_tags_data[i]
     elif j == len1+7:
       X_train3[count_i,j,:] = final_pos_data[i]
     elif j == len1+8:
       X_train3[count_i,j,:] = final_tokens_data[i]
     elif j == len1+9:
       X_train3[count_i,j,:] = final_dep_data[i]
     else:
       X_train3[count_i,j,:] = empty_word

   Y_train3[count_i,:] = dummy_y[i]
   count_i += 1

count_i = 0
for i in testing_indices[givenIndex]:
  len1 = len(sen[i][0])
  average_vector1 = np.zeros(512).astype(np.float32)
  average_vector2 = np.zeros(512).astype(np.float32)
  average_vector3 = np.zeros(512).astype(np.float32)
  for j in range(max(maxlength)+10):
    if j < len1:
      X_test3[count_i,j,:] = doc2vec_model[sen[i][0][j]]
      average_vector1 += result_train1[i, tfidf1.vocabulary_[sen[i][0][j]]] *
doc2vec_model[sen[i][0][j]]
```

```
            average_vector2 += result_train2[i, tfidf2.vocabulary_[sen[i][0][j]]] * doc2vec_model[sen[i][0][j]]
            average_vector3 += result_train3[i, tfidf3.vocabulary_[sen[i][0][j]]] * doc2vec_model[sen[i][0][j]]
        #elif j >= len1 and j < len1 + 379:
        #   X_test3[count_i,j,:] = glove_data[i, j-len1, :]
        elif j == len1:
            X_test3[count_i,j,:] = tfidf_data1[i]
        elif j == len1 + 1:
            X_test3[count_i,j,:] = tfidf_data2[i]
        elif j == len1+2:
            X_test3[count_i,j,:] = tfidf_data3[i]
        elif j == len1+3:
            X_test3[count_i,j,:] = average_vector1
        elif j == len1+4:
            X_test3[count_i,j,:] = average_vector2
        elif j == len1+5:
            X_test3[count_i,j,:] = average_vector3
        elif j == len1+6:
            X_test3[count_i,j,:] = final_pos_tags_data[i]
        elif j == len1+7:
            X_test3[count_i,j,:] = final_pos_data[i]
        elif j == len1+8:
            X_test3[count_i,j,:] = final_tokens_data[i]
        elif j == len1+9:
            X_test3[count_i,j,:] = final_dep_data[i]
        else:
            X_test3[count_i,j,:] = empty_word

    Y_test3[count_i,:] = dummy_y[i]
    count_i += 1
```

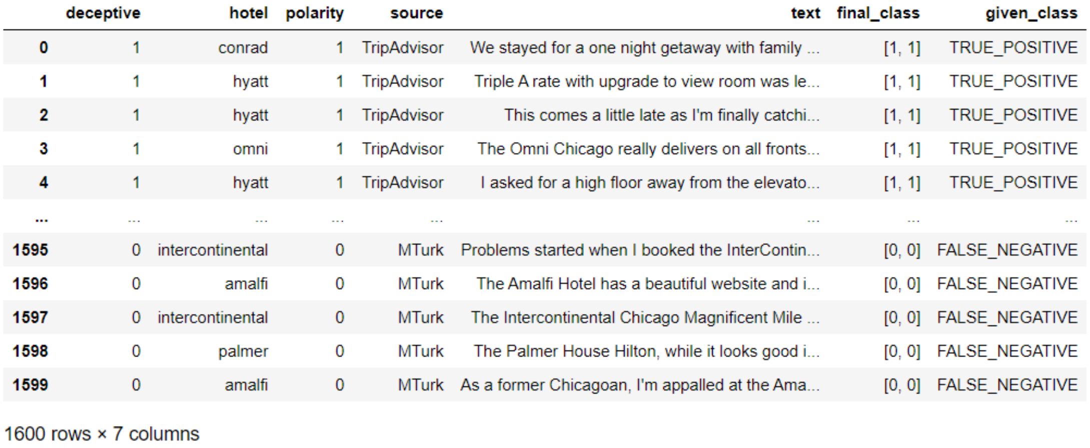
```
      return X_train3, X_test3, Y_train3, Y_test3
      model = Sequential()
    model.add(Conv1D(filters=128, kernel_size=9, padding='same', activation='relu',
    input_shape=(max(maxlength)+10,512)))
    model.add(Dropout(0.25))
    model.add(MaxPooling1D(pool_size=2))
    model.add(Dropout(0.25))
    model.add(Conv1D(filters=128, kernel_size=7, padding='same', activation='relu'))
    model.add(Dropout(0.25))
    model.add(MaxPooling1D(pool_size=2))
    model.add(Dropout(0.25))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same', activation='relu'))
    model.add(Dropout(0.25))

    model.add(Bidirectional(LSTM(50, dropout=0.25, recurrent_dropout=0.2)))
    model.add(Dense(4, activation='softmax'))
    model.compile(loss='binary_crossentropy', optimizer='Adam', metrics=['accuracy'])
    print(model.summary())

final_accuracies = []

filename = 'weights.best.from_scratch%s.hdf5' % 9

checkpointer = ModelCheckpoint(filepath=filename, verbose=1,
save_best_only=True)

X_train3, X_test3, Y_train3, Y_test3 = extractTrainingAndTestingData(9)

model.fit(X_train3, Y_train3, epochs=10, batch_size=512,
callbacks=[checkpointer], validation_data=(X_test3, Y_test3))

model.load_weights(filename)
```

```
for i in range(1):
   filename = 'weights.best.from_scratch%s.hdf5' % i
   checkpointer = ModelCheckpoint(filepath=filename, verbose=1, save_best_only=True)
   X_train3, X_test3, Y_train3, Y_test3 = extractTrainingAndTestingData(i)
   model.fit(X_train3, Y_train3, epochs=10, batch_size=512, callbacks=[checkpointer], validation_data=(X_test3, Y_test3))
   model.load_weights(filename)
   predicted = np.rint(model.predict(X_test3))
   final_accuracies.append(accuracy_score(Y_test3, predicted))
   print(accuracy_score(Y_test3, predicted))

print(sum(final_accuracies) / len(final_accuracies))
Y_test
```

5.3.2 Screenshots :

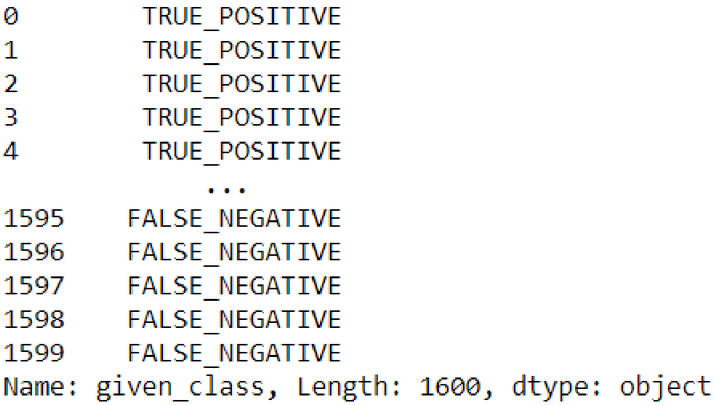


	deceptive	hotel	polarity	source	text
0	truthful	conrad	positive	TripAdvisor	We stayed for a one night getaway with family ...
1	truthful	hyatt	positive	TripAdvisor	Triple A rate with upgrade to view room was le...
2	truthful	hyatt	positive	TripAdvisor	This comes a little late as I'm finally catchi...
3	truthful	omni	positive	TripAdvisor	The Omni Chicago really delivers on all fronts...
4	truthful	hyatt	positive	TripAdvisor	I asked for a high floor away from the elevato...

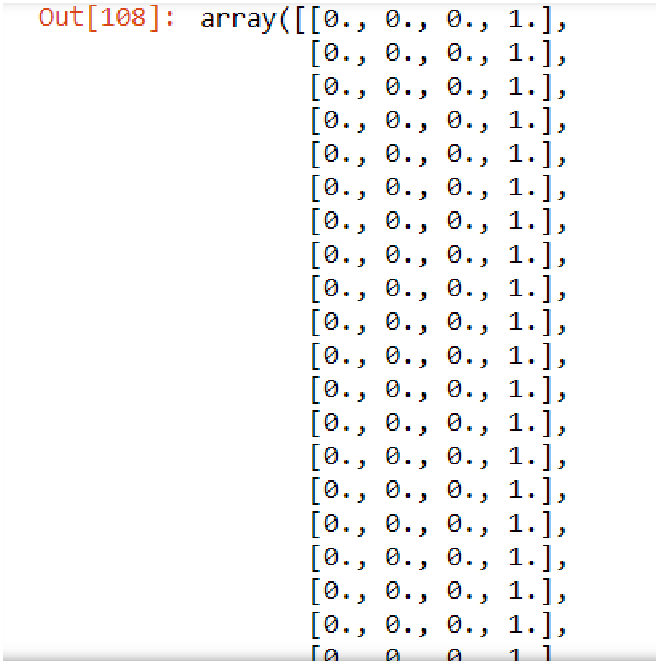


	deceptive	hotel	polarity	source	text	final_class	given_class
0	1	conrad	1	TripAdvisor	We stayed for a one night getaway with family ...	[1, 1]	TRUE_POSITIVE
1	1	hyatt	1	TripAdvisor	Triple A rate with upgrade to view room was le...	[1, 1]	TRUE_POSITIVE
2	1	hyatt	1	TripAdvisor	This comes a little late as I'm finally catchi...	[1, 1]	TRUE_POSITIVE
3	1	omni	1	TripAdvisor	The Omni Chicago really delivers on all fronts...	[1, 1]	TRUE_POSITIVE
4	1	hyatt	1	TripAdvisor	I asked for a high floor away from the elevato...	[1, 1]	TRUE_POSITIVE
...	...	...	...	...	...	...	...
1595	0	intercontinental	0	MTurk	Problems started when I booked the InterContin...	[0, 0]	FALSE_NEGATIVE
1596	0	amalfi	0	MTurk	The Amalfi Hotel has a beautiful website and i...	[0, 0]	FALSE_NEGATIVE
1597	0	intercontinental	0	MTurk	The Intercontinental Chicago Magnificent Mile ...	[0, 0]	FALSE_NEGATIVE
1598	0	palmer	0	MTurk	The Palmer House Hilton, while it looks good i...	[0, 0]	FALSE_NEGATIVE
1599	0	amalfi	0	MTurk	As a former Chicagoan, I'm appalled at the Ama...	[0, 0]	FALSE_NEGATIVE

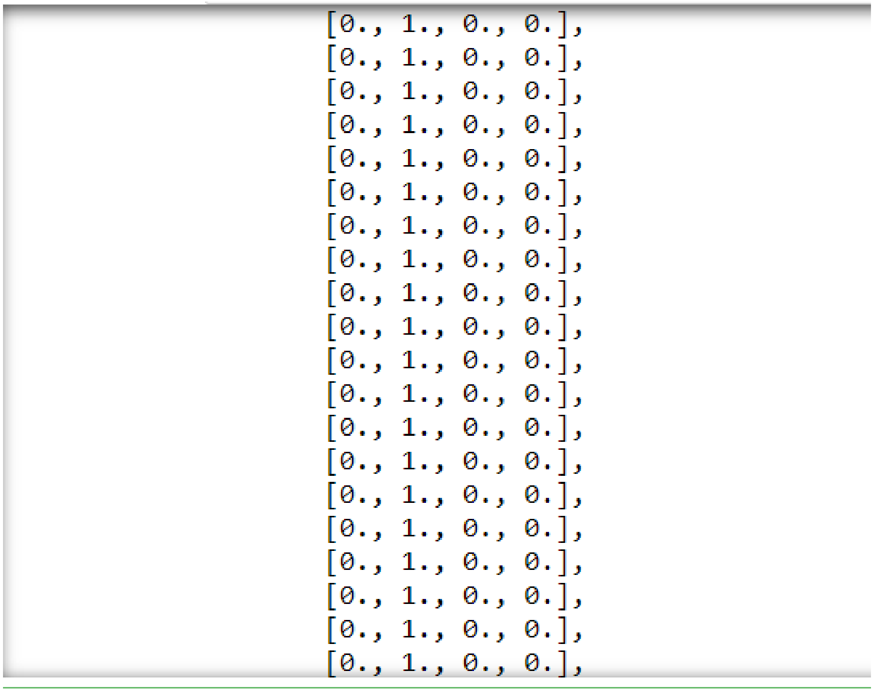
1600 rows × 7 columns



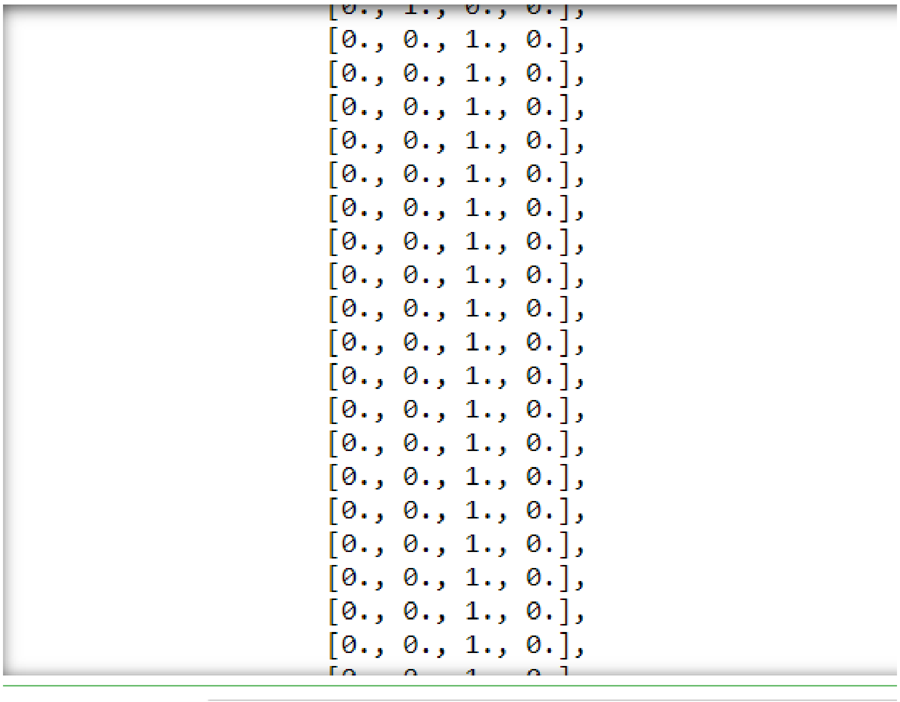
```
0         TRUE_POSITIVE
1         TRUE_POSITIVE
2         TRUE_POSITIVE
3         TRUE_POSITIVE
4         TRUE_POSITIVE
              ...
1595     FALSE_NEGATIVE
1596     FALSE_NEGATIVE
1597     FALSE_NEGATIVE
1598     FALSE_NEGATIVE
1599     FALSE_NEGATIVE
Name: given_class, Length: 1600, dtype: object
```



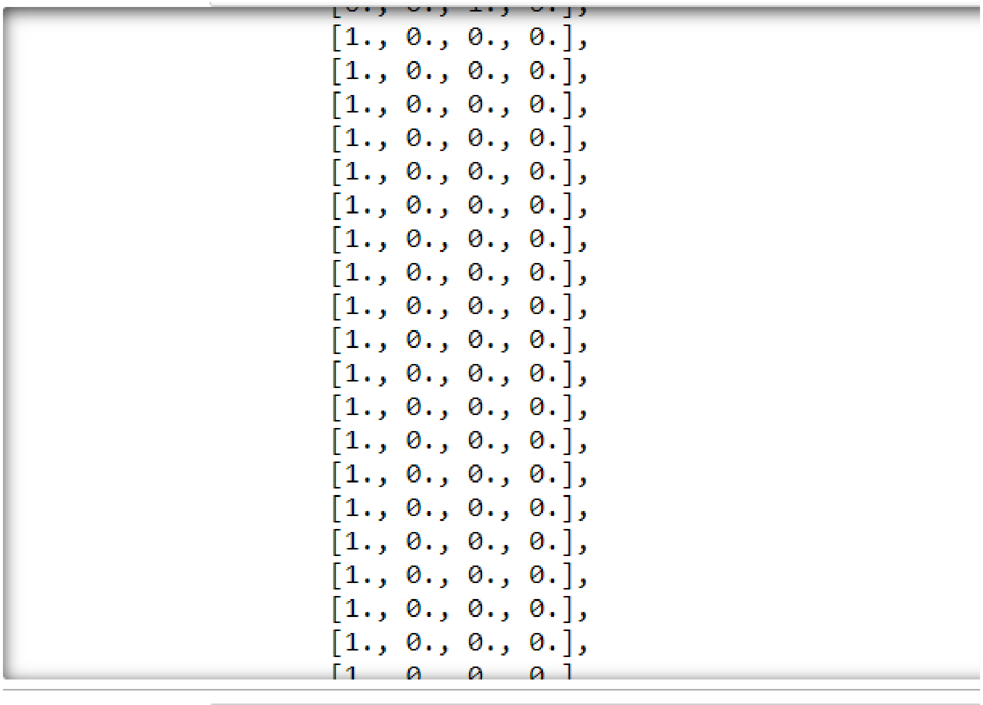
```
Out[108]: array([[0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
                 [0., 0., 0., 1.],
```



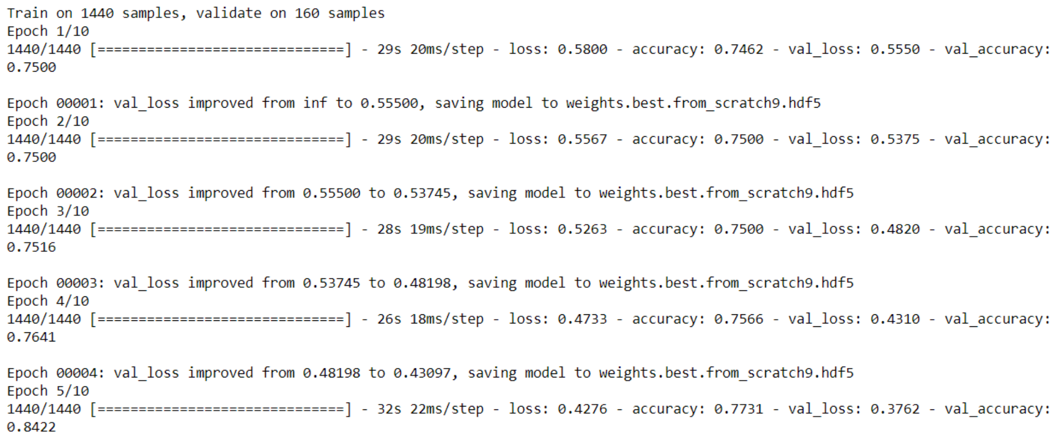
```
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
[0., 1., 0., 0.],
```



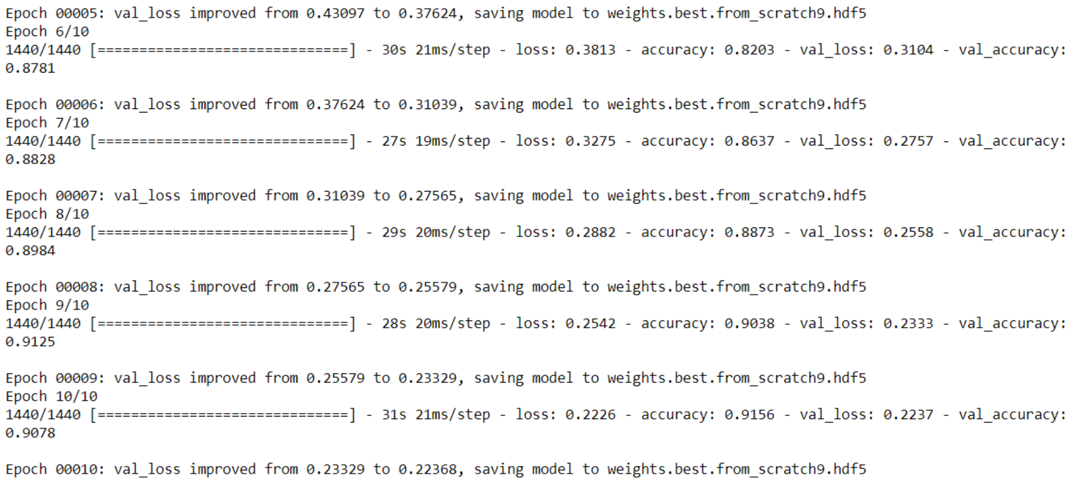
```
[0., 1., 0., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.],
[0., 0., 1., 0.]
```



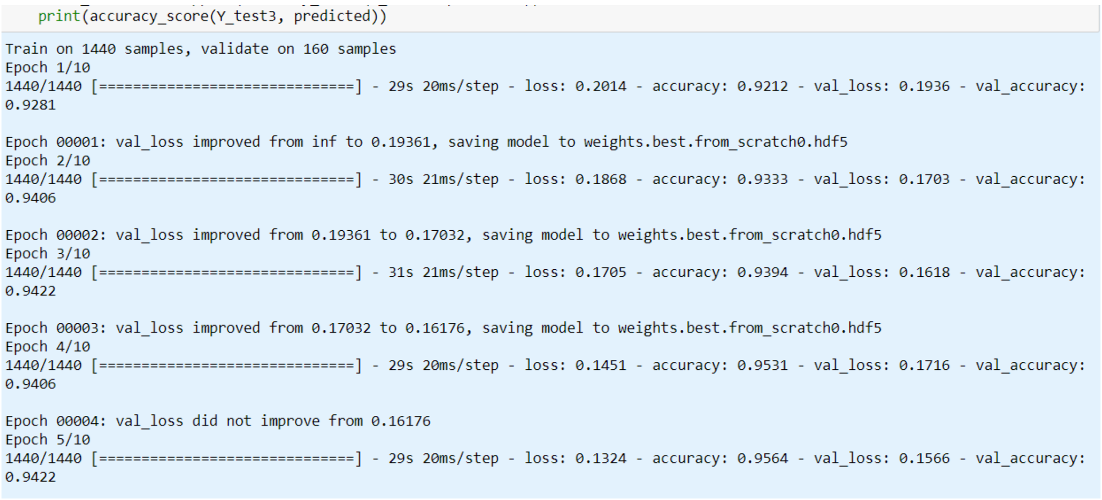
```
[0., 0., 1., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.],
[1., 0., 0., 0.]
```



```
Train on 1440 samples, validate on 160 samples
Epoch 1/10
1440/1440 [==============================] - 29s 20ms/step - loss: 0.5800 - accuracy: 0.7462 - val_loss: 0.5550 - val_accuracy:
0.7500

Epoch 00001: val_loss improved from inf to 0.55500, saving model to weights.best.from_scratch9.hdf5
Epoch 2/10
1440/1440 [==============================] - 29s 20ms/step - loss: 0.5567 - accuracy: 0.7500 - val_loss: 0.5375 - val_accuracy:
0.7500

Epoch 00002: val_loss improved from 0.55500 to 0.53745, saving model to weights.best.from_scratch9.hdf5
Epoch 3/10
1440/1440 [==============================] - 28s 19ms/step - loss: 0.5263 - accuracy: 0.7500 - val_loss: 0.4820 - val_accuracy:
0.7516

Epoch 00003: val_loss improved from 0.53745 to 0.48198, saving model to weights.best.from_scratch9.hdf5
Epoch 4/10
1440/1440 [==============================] - 26s 18ms/step - loss: 0.4733 - accuracy: 0.7566 - val_loss: 0.4310 - val_accuracy:
0.7641

Epoch 00004: val_loss improved from 0.48198 to 0.43097, saving model to weights.best.from_scratch9.hdf5
Epoch 5/10
1440/1440 [==============================] - 32s 22ms/step - loss: 0.4276 - accuracy: 0.7731 - val_loss: 0.3762 - val_accuracy:
0.8422
```

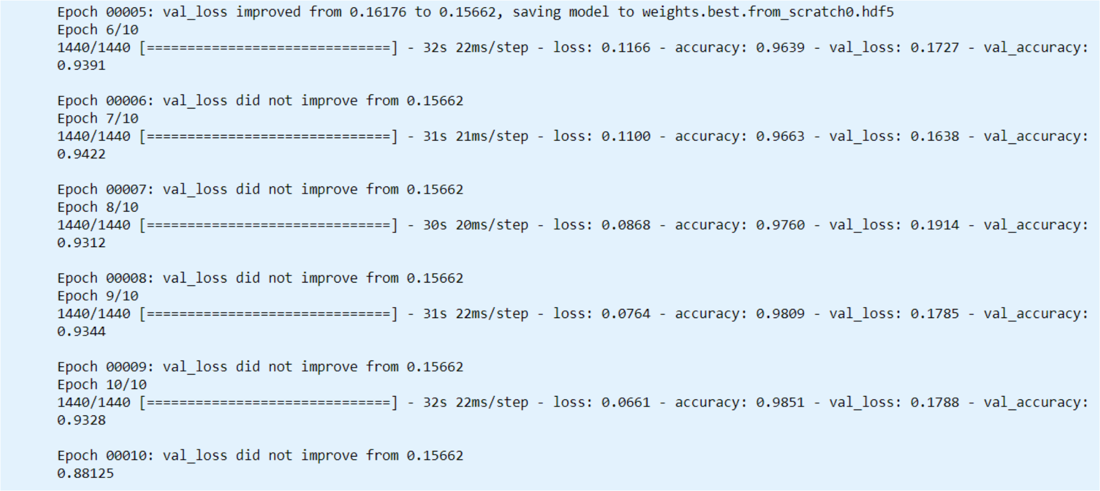


```
Epoch 00005: val_loss improved from 0.43097 to 0.37624, saving model to weights.best.from_scratch9.hdf5
Epoch 6/10
1440/1440 [==============================] - 30s 21ms/step - loss: 0.3813 - accuracy: 0.8203 - val_loss: 0.3104 - val_accuracy:
0.8781

Epoch 00006: val_loss improved from 0.37624 to 0.31039, saving model to weights.best.from_scratch9.hdf5
Epoch 7/10
1440/1440 [==============================] - 27s 19ms/step - loss: 0.3275 - accuracy: 0.8637 - val_loss: 0.2757 - val_accuracy:
0.8828

Epoch 00007: val_loss improved from 0.31039 to 0.27565, saving model to weights.best.from_scratch9.hdf5
Epoch 8/10
1440/1440 [==============================] - 29s 20ms/step - loss: 0.2882 - accuracy: 0.8873 - val_loss: 0.2558 - val_accuracy:
0.8984

Epoch 00008: val_loss improved from 0.27565 to 0.25579, saving model to weights.best.from_scratch9.hdf5
Epoch 9/10
1440/1440 [==============================] - 28s 20ms/step - loss: 0.2542 - accuracy: 0.9038 - val_loss: 0.2333 - val_accuracy:
0.9125

Epoch 00009: val_loss improved from 0.25579 to 0.23329, saving model to weights.best.from_scratch9.hdf5
Epoch 10/10
1440/1440 [==============================] - 31s 21ms/step - loss: 0.2226 - accuracy: 0.9156 - val_loss: 0.2237 - val_accuracy:
0.9078

Epoch 00010: val_loss improved from 0.23329 to 0.22368, saving model to weights.best.from_scratch9.hdf5
```

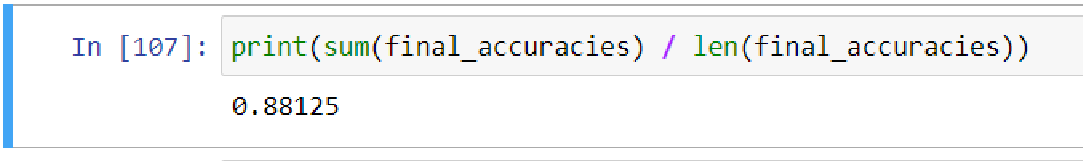


```
    print(accuracy_score(Y_test3, predicted))
```

```
Train on 1440 samples, validate on 160 samples
Epoch 1/10
1440/1440 [==============================] - 29s 20ms/step - loss: 0.2014 - accuracy: 0.9212 - val_loss: 0.1936 - val_accuracy:
0.9281

Epoch 00001: val_loss improved from inf to 0.19361, saving model to weights.best.from_scratch0.hdf5
Epoch 2/10
1440/1440 [==============================] - 30s 21ms/step - loss: 0.1868 - accuracy: 0.9333 - val_loss: 0.1703 - val_accuracy:
0.9406

Epoch 00002: val_loss improved from 0.19361 to 0.17032, saving model to weights.best.from_scratch0.hdf5
Epoch 3/10
1440/1440 [==============================] - 31s 21ms/step - loss: 0.1705 - accuracy: 0.9394 - val_loss: 0.1618 - val_accuracy:
0.9422

Epoch 00003: val_loss improved from 0.17032 to 0.16176, saving model to weights.best.from_scratch0.hdf5
Epoch 4/10
1440/1440 [==============================] - 29s 20ms/step - loss: 0.1451 - accuracy: 0.9531 - val_loss: 0.1716 - val_accuracy:
0.9406

Epoch 00004: val_loss did not improve from 0.16176
Epoch 5/10
1440/1440 [==============================] - 29s 20ms/step - loss: 0.1324 - accuracy: 0.9564 - val_loss: 0.1566 - val_accuracy:
0.9422
```



```
Epoch 00005: val_loss improved from 0.16176 to 0.15662, saving model to weights.best.from_scratch0.hdf5
Epoch 6/10
1440/1440 [==============================] - 32s 22ms/step - loss: 0.1166 - accuracy: 0.9639 - val_loss: 0.1727 - val_accuracy:
0.9391

Epoch 00006: val_loss did not improve from 0.15662
Epoch 7/10
1440/1440 [==============================] - 31s 21ms/step - loss: 0.1100 - accuracy: 0.9663 - val_loss: 0.1638 - val_accuracy:
0.9422

Epoch 00007: val_loss did not improve from 0.15662
Epoch 8/10
1440/1440 [==============================] - 30s 20ms/step - loss: 0.0868 - accuracy: 0.9760 - val_loss: 0.1914 - val_accuracy:
0.9312

Epoch 00008: val_loss did not improve from 0.15662
Epoch 9/10
1440/1440 [==============================] - 31s 22ms/step - loss: 0.0764 - accuracy: 0.9809 - val_loss: 0.1785 - val_accuracy:
0.9344

Epoch 00009: val_loss did not improve from 0.15662
Epoch 10/10
1440/1440 [==============================] - 32s 22ms/step - loss: 0.0661 - accuracy: 0.9851 - val_loss: 0.1788 - val_accuracy:
0.9328

Epoch 00010: val_loss did not improve from 0.15662
0.88125
```



```
In [107]: print(sum(final_accuracies) / len(final_accuracies))

          0.88125
```

CHAPTER-6

TESTING

Software Review is a software evaluation toward user-assembled requirements and system specifications. Testing is carried out during the life cycle of software creation at the method level or at the computer code stage of the project. The validation and verification processes necessitate the software testing baseline.

Software Validation

Validation is the method of testing whether the program meets user requisites or not. This tends to occur at the final hour of the SDLC. When the software designed actually meets the requisites it was constructed for, it is considered to be validation.

- Validation ensures the product being synthesized is in compliance with the consumer specifications.
- The technology getting devised needs to accomplish the client requirements. Validation tests that.
- Increase emphasis is cited through validation on the user needs

Verification of Applications

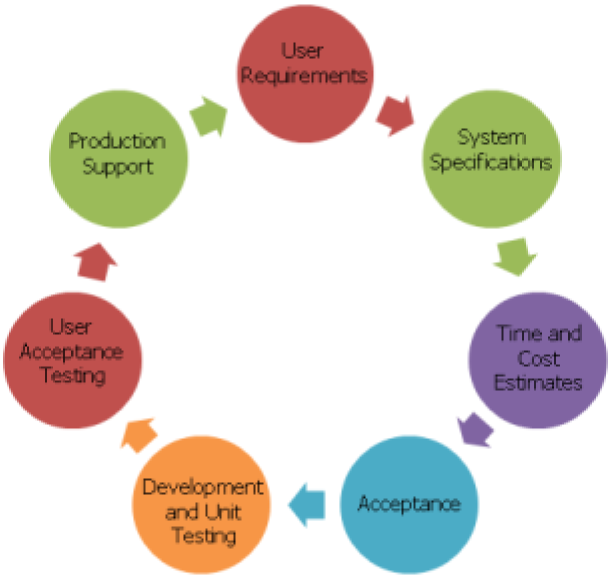
Verification is the process of ensuring that the system complies with industry expectations and is specified in accordance with the applicable specifications and methodologies.

- Inspection shall insure that the product being developed complies with the specification specifications.
- All design requirements need to be met while developing the concerned product. Verification tests that.

The clear-cut responsibility of the verification process is to check if the designed product meets design and the system features.

The goals of the study are-

- Errors-These are the real computer checks done by the professional engineers. In addition, there is a disparity between the output of the device and the expected results described as an mistake.
- Fault - A fault, or bug, is consequence of an error that can cause failure in a program.
- Failure - the system's incompetence to accomplish the desired function is considered to be a failure. Failure arises when the system errs.



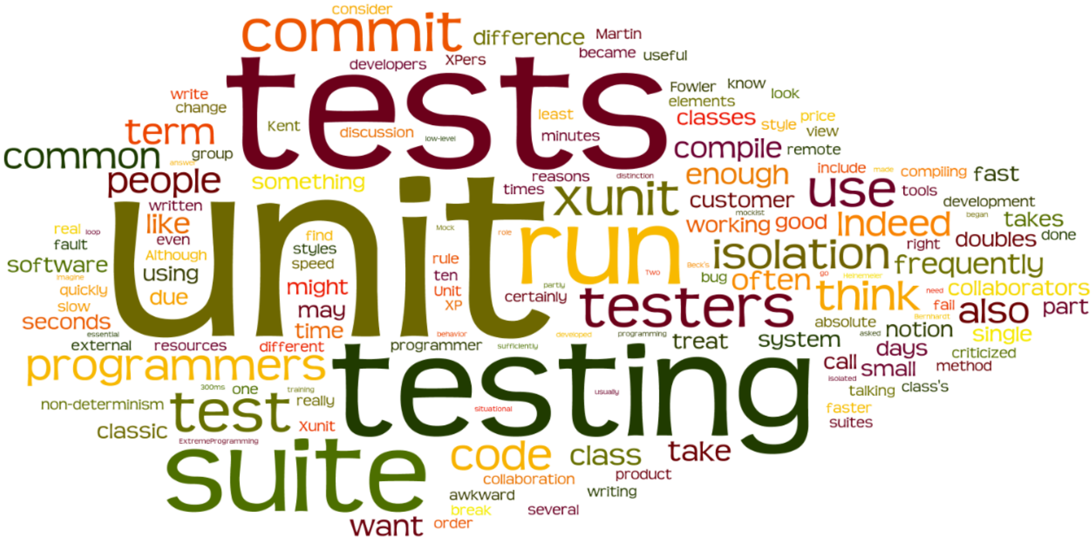
Testing Approaches

Tests may be performed on the basis of two methods

- Functionality testing
- Implementation testing

It is referred to as black-box checking when reviewing programs, without taking into consideration the real implementation. The other area is classified as white-box research, which checks not just the software but also the way it is applied.

Exhaustive testing is the method most important for optimal research. Any prospective interest is calculated in the context of output and input values. In actuality, since there is a wide spectrum of beliefs, it is impossible to determine each and every one of them.



Black Box testing

This is required to check the efficiency of the software. This is also regarded as 'comportmental' research. In this scenario, the tester has a range of input values and predicted outcomes. When the data is supplied, the software will be evaluated 'ok' if the output meets the expected results and the problem otherwise arises.
In this research phase, the tester does not know the design and structure of the application, In such a way that professional developers and end-users carry out this test on the system.

Testing methods for the Black Box:

- ● Equivalence Group – Dividing the results into the same classes. If one class item fails the test, the whole class is deemed to be successful.

- ● Boundary values – Split up data into top and lower end values. If such values pass the check, all values in between will also pass the quest.

- Cause-Effect Graphing-Only one input variable at a time is being checked in allprevious approaches. Cause (input) and Effect (output) is a testing methodology in which input value variations are routinely tested.

- Pair-wise Research-Program activity is dependent on many parameters. The various criteria for their respective values are tested pair-wise in pair-wise research.

- State-based monitoring – The condition of the system varies as input is provided. These systems will be reviewed according to their status and results.

White-box testing

Test programs and their execution are performed to improve the functionality or layout of the application. It's sometimes called structural research.

In this test phase, the tester is conscious of the design and structure of the application. Computer programmers are running a technology check.

The following are just a few white-box analysis techniques:

- Control-flow monitoring-The purpose of the control-flow monitoring is to create test cases that involve both statements and division scenarios. The criteria of the division are checked for both true and false, so that both interpretations can be obtained.

- Data-flow study – The purpose of this research methodology is to cover all data variables used in the system. It governs where the variables have been identified and indicated and where they have been used or modified.

Rates of research

Analysis on its own can be done at various SDLC rates. The assessment method is similar to the development of software. The move is verified, verified and approved until you step to the next transfer.

Separate reviews are primarily performed to insure there there are no secret bugs or issues discovered in the app. The software is being checked at various speeds.

unit testing

Once coding, the author runs multiple checks on the computer system to see whether it is error free. The research is carried out in the sense of the white box study. Unit testing helps developers to determine if individual system units work as planned and are error-free.

integration testing

Even the software units operate well separately, it is still necessary to figure out if the units can all operate together without errors. For eg, forwarding the claim and updating the details, etc.

System Testing

The software is collected as a template and reviewed in its entirety. This can be accomplished by taking one or more of the following measures:

- Regulation of features-Checks against the requirements for all features of the app.
- Results Testing-This measure indicates how good the software is. It tests the success of the system and the total period it takes for the required research to be carried out. Quality monitoring shall be carried out by way of load loading and stress testing where the system is exposed to a heavy load of users and data in various climatic condition.

- Portability & Security - These tests are carried out because the system is structured to run on several channels and to be open to a variety of individuals.

Acceptance testing

If the device is worthy of handing back to the consumer, it will be tested for user interface and reaction in the final test phase. This is important because it can be refused even though the software follows certain consumer standards, or though the customer does not like how it appears or how it works.

- Alpha research-Alpha development is conducted by the team of engineers themselves, utilizing the software as though it has been encountered in the job world. They're trying to work out how consumers can respond to some system behavior and how the software can respond to inputs.

- Beta Testing-After internal validation of the app, consumers are requested to use it for research purposes only in their production environment. It's not the product it's shipped yet. Developers expect users to pose minute problems at this point that have been missed to participate.

Regression testing

If a software program is modified with a new algorithm, function or feature, it is closely tested to decide if the additional implementation has any adverse consequences. This is also a regression study.

CHAPTER-7

RESULTS AND DISCUSSIONS

At extractTrainingAndTestingData(9)

Epoch	Loss	Accuracy	Val loss	Val accuracy
1	0.5643	0.7500	0.5523	0.7500
5	0.3895	0.8118	0.3311	0.8625
10	0.1897	0.9312	0.2668	0.9000

At extractTrainingAndTestingData(1)

Epoch	Loss	Accuracy	Val loss	Val accuracy
1	0.2003	0.9273	0.1372	0.9563
5	0.1167	0.9603	0.1398	0.9531
10	0.0548	0.9872	0.1351	0.9547

- Mean of final accuracies is 0.90625
- Training is done on 1440 samples and validation on 160 samples
- The epoch is a full overview of the data collection to be obtained from a learn ing computer.

Description of the terms of used:

- positive (P): findings are good.
- negative (N): results are not optimistic.
- true positive (TP): conclusion is optimistic and is supposed to be true.
- false negative (FN): result is optimistic but is expected to be wrong.
- true negative (TN): finding is unfavourable and is predicted to be not optimistic.
- False Positive (FP) : finding is unfavourable and is supposed to be optimistic.

Classification rate/ accuracy:

Classification rate or accuracy is given by :

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy calculation

However, there are questions about precision. It implies equal losses for all kinds of mistake. Depending on the issue, 99 percent accuracy may be fantastic, good, mediocre, poor or terrible.

CHAPTER-8

CONCLUSION AND FUTURE WORK

In this paper we focused on estimation of the reviews where the predicted performance would be derived by considering the meta-data of the reviews, in addition to the textual content or duration of the review, where it provides a more precise prediction that lets the customer determine whether or not to proceed for the product. Prediction of feedback is achieved by the long short term memory, a recurrent neural network which is trained using backpropagation method that overcomes the gradient issue and provides the findings suggested that end-users Prefer keywords and context to sentence structure and readability In fact, it also ensures that users can develop an appreciation of the commodity in less time. So, we 're writing this paper to propose helpful review

For future analysis, we may recommend studying the following issues related to the value assessment of online reviews, user-specific and explicable recommendations for helpful feedback: because various consumers could be worried with other aspects of online products, useful comment recommendations need to be more user-specific and self-explanatory. Improving the estimation of valuable feedback using unmarked data: since a limited proportion of comments may be found heuristically beneficial or unhelpful, it is therefore a fruitful analysis to accurately estimate the utility of online reviews on the basis of a limited number of classified data and a significant number of unidentified data.

REFERENCE

[1] Wan, Y., and Nakayama, M. (2012). Are Amazon.com Online Review Helpfulness Ratings Biased or Not? Life: Web-Enabled Convergence of Commerce, Work, and Social Life (M. J. Shaw, D. Zhang, and W. T. Yue. eds.), Springer Berlin Heidelberg, pp.

[2]Danescu-Niculescu-Mizil & G. Kossinets. & J. Kleinberg & L. Lee (2009) How Opinions are Received by Online Communities: A Case Study on Amazon.com Helpfulness Votes, Proceedings of the 18th international conference on World wide web, pp. 141-150. New York, NY: Association for Computing Machinery.

[3]MiaoFan,ChaoFeng,LinGuo,MingmingSun,PingLi.2019.Product-Aware Helpfulness Prediction of Online Reviews. In Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco,CA,USA,ACM,NewYork,NY,USA,7pages.

[4] Al-Smadi M, Qawasmeh O, Talafha B, Quwaider M (2015b) Human annotated Arabic dataset of book reviews for aspect based sentiment analysis. In: future internet of things and cloud (FiCloud), 2015 3rd international conference, IEEE, pp 726–730

[5] Baccianella S, Esuli A, Sebastiani F (2010) Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. LREC 10:2200–2204

[6]MiaoFan,YueFeng,MingmingSun,PingLi,HaifengWang,andJianminWang.2018.Multi-TaskNeuralLearningArchitectureforEnd-to-EndIdentificationof HelpfulReviews.InProceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).343–350

[7]Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. InProceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics,1746–1751

Format - I

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES

(**To be attached in the dissertation/ project report**)

1	Name of the Candidate **(IN BLOCK LETTERS)**	Saran Reddy | Sravan Akuthota | Aravind Yalla
2	Address of the Candidate	Bharathi Salai, Ramapuram, Chennai, Tamil Nadu 600089 **Mobile Number :** 7013530280| 7799523234| 9030623678
3	Registration Number	RA1611003020024 | RA1611003020032 | RA1611003020047
4	Date of Birth	14-04-1999 | 28-10-1998 | 25-06-1999
5	Department	Computer Science & Engineering
6	Faculty	Engineering and Technology
7	Title of the Dissertation/Project	Examining and Predicting Helpfulness of reviews based On LSTM
8	Whether the above project/dissertation is done by	I~~ndividual~~ or group : group (Strike whichever is not applicable) a) If the project/ dissertation is done in group, then how many students together completed the project : 3 b) Mention the Name & Register number of other candidates : Saran Reddy | Sravan Akuthota | Aravind Yalla RA1611003020024|RA1611003020032|RA1611003020047
9	Name and address of the Supervisor / Guide	Ms.A.Aruna arunaarulmani@gmail.com 9629664982 **Mail ID : Mobile Number :**
10	Name and address of the Co-Supervisor / Co- Guide (if any)	NA **Mail ID : Mobile Number :**

11	Software Used	Turnitin		
12	Date of Verification	16 - 05 - 2020		
13	**Plagiarism Details: (to attach the final report from the software)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1	INTRODUCTION	<1%	<1%	<1%
2	LITERATURE SURVEY	<1%	<1%	<1%
3	SYSTEM DESIGN	<2%	<2%	<2%
4	MODULE DESCRIPTION	<1%	<1%	<1%
5	SYSTEM IMPLEMENTATION	<1%	<1%	<1%
6	TESTING	<1%	<1%	<1%
7	RESULT ANALYSIS	<1%	<1%	<1%
8	CONCLUSION AND FUTURE WORK	<1%	<1%	<1%
9				
10				
Appendices		NA	NA	NA

I / We declare that the above information have been verified and found true to the best of my / our knowledge.

A. Sravan Y. Aravind **Signature of the Candidate**	**Name & Signature of the Staff (Who uses the plagiarism check software)**
Name & Signature of the Supervisor/Guide	**Name & Signature of the Co-Supervisor/Co-Guide**

Name & Signature of the HOD

a92

ORIGINALITY REPORT

10%	2%	2%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Birla Institute of Technology and Science Pilani Student Paper	1%
2	Submitted to University of Central Florida Student Paper	1%
3	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
4	Submitted to Amity University Student Paper	1%
5	Submitted to Shrewsbury Sixth Form College Student Paper	1%
6	Submitted to ABES Engineering College Student Paper	1%
7	en.wikipedia.org Internet Source	1%
8	pypi.org Internet Source	1%

9	Submitted to Rajarambapu Institute of Technology Student Paper	<1%
10	Submitted to University of Southampton Student Paper	<1%
11	Submitted to CSU, San Jose State University Student Paper	<1%
12	Submitted to Nizwa College of Technology Student Paper	<1%
13	Submitted to University of Teesside Student Paper	<1%
14	cs224d.stanford.edu Internet Source	<1%
15	Submitted to Central Queensland University Student Paper	<1%
16	thecarnivors.com Internet Source	<1%
17	Submitted to National College of Ireland Student Paper	<1%
18	Submitted to The Robert Gordon University Student Paper	<1%
19	Submitted to Loughborough University Student Paper	<1%

20	Submitted to University of Bedfordshire Student Paper	<1%
21	docs.ipswitch.com Internet Source	<1%
22	Manohar Swamynathan. "Mastering Machine Learning with Python in Six Steps", Springer Science and Business Media LLC, 2017 Publication	<1%
23	Submitted to City University of Hong Kong Student Paper	<1%
24	"Advances in Brain Inspired Cognitive Systems", Springer Science and Business Media LLC, 2018 Publication	<1%

Exclude quotes	Off	Exclude matches	< 10 words
Exclude bibliography	On		