



***Dissertation on***

**“Authorship Attribution”**

*Submitted in partial fulfillment of the requirements for the award of degree of*

**Bachelor of Technology  
in  
Computer Science & Engineering**

***Submitted by:***

<b>Daivik U D</b>	<b>01FB15ECS085</b>
<b>Sushmitha Somashekar</b>	<b>01FB15ECS319</b>
<b>Tejaswini K</b>	<b>01FB15ECS325</b>

***Under the guidance of***

**Internal Guide**

**Dr. Jayashree R**

**Professor**

**January – May 2019**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
FACULTY OF ENGINEERING  
PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)  
100ft Ring Road, Bengaluru – 560 085, Karnataka, India



**PES UNIVERSITY**  
(Established under Karnataka Act No. 16 of 2013)  
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

**FACULTY OF ENGINEERING**

## **CERTIFICATE**

*This is to certify that the dissertation entitled*

**‘Authorship Attribution’**

*is a bonafide work carried out by*

<b>Daivik U D</b>	<b>01FB15ECS085</b>
<b>Sushmitha Somashekar</b>	<b>01FB15ECS319</b>
<b>Tejaswini K</b>	<b>01FB15ECS325</b>

In partial fulfilment for the completion of eighth semester project work in the Program of Study Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Jan. 2019 – May. 2019. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 8<sup>th</sup> Semester academic requirements in respect of project work.

Signature  
Dr. Jayashree R  
Professor

Signature  
Dr. Shylaja S S  
Chairperson

Signature  
Dr. B. K. Keshavan  
Dean of Faculty

**External Viva**

**Name of the Examiners**

**Signature with Date**

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

## DECLARATION

We hereby declare that the project entitled “**Authorship Attribution**” has been carried out by us under the guidance of Dr. Jayashree R, Associate Professor and submitted in partial fulfillment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester January – May 2019. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

01FB15ECS085

Daivik U D

01FB15ECS319

Sushmitha Somashekar

01FB15ECS325

Tejaswini K

# ACKNOWLEDGEMENT

We would like to thank Dr. Jayashree R for her continuous support and guidance without which the completion of the project would not have been possible.

We are grateful to the project coordinators, Ms. Preet Kanwal and Ms. Sangeeta for organizing, managing and helping out with the entire process.

We express our deep sense of gratitude to Dr. Shylaja SS, Head of Department, Computer Science and Engineering, for providing us with so many opportunities to showcase our talent.

We would like to thank Dr. B K Keshavan, Dean of Faculty for his determined work in ensuring students had the best possible faculty to prepare them for whatever activity they decide to pursue in the future.

We are grateful to Dr. K N Balasubramanya Murthy, Vice Chancellor, for his continued work towards ensuring a vibrant learning environment within PES University existed for students to excel.

We would like to thank Prof. Jawahar Doreswamy, CEO and Pro Chancellor for his work towards bettering PES University year upon year and ensuring the infrastructure for learning always existed.

We would like to thank Dr. M R Doreswamy, Chairman and Chancellor of PES University for his vital role in setting up this institution over 25 years ago, and ensuring it becomes a top college in India.

Finally, we would like to thank our parents, without whose continued support, we would not have been able to achieve any of the things we have so far.

# ABSTRACT

Authorship Attribution (AA) is a well-contemplated issue among Natural Language Processing scientists which goes back to the earliest endeavors at quantitative examination of content records. The field of Authorship Attribution started at some point in the late nineteenth century with the principal endeavors to evaluate composing style. Enthusiasm for the field was principally created by the craving for progressively target avocations for the initiation of questioned works by creators, for example, Shakespeare and Bacon. Measurable strategies were overwhelming in the field when of the early investigations of Zipf and Yule of relative word frequencies, which found that the recurrence with which a word showed up in a content could be approximated with a Poisson circulation.

In our Project the user is directed to the main page of the wesbite after signup and login. On validation of the credentials and entering the main page, we can create the dataset by clicking on “Create Dataset”. This creates a random, shuffled dataset of sentences from the list of books in our dataset. We then apply “Logistic Regression” or “Neural Network” to train the dataset. The results are observed in the form of graphs and stored in a CSV file. After training the model, for prediction, we submit the line/lines from the known/unknown articles in the Query Box to find the author name.

# TABLE OF CONTENTS

Chapter No.	Title	Page No
1.	INTRODUCTION	1
2.	PROBLEM DEFINITION	2
3.	LITERATURE SURVEY	3
3.1	Author Attribution with CNN	4
3.2	Authorship Attribution with Topic Models	4
3.3	Authorship attribution using PCA and CNN	5
3.4	Machine learning approach to authorship attribution of literary texts	6
4.	CUSTOMER REQUIREMENT SPECIFICATION	9
4.1	Introduction	9
4.1.1	Scope	9
4.2	Product Perspective	9
4.2.1	User Characteristics	10
4.3	Architecture Diagram	11
4.4	Requirements List	11
4.4.1	Scenario 1	11
4.4.2	Scenario 2	12
4.4.3	Scenario 3	12
4.4.4	Scenario 4	12
4.4.5	Scenario 5	13
4.5	External Interface Requirements	13
4.5.1	Hardware Requirements	13
4.5.2	Software Requirements	13
4.6	User Interfaces	14
4.7	Performance Requirements	14
4.8	Help	14
4.9	Other Requirements	14
4.9.1	Site Adaptation Requirements	14
5.	HIGH LEVEL DESIGN	16
5.1	Use Case Diagram	16
5.2	User Interfaces	17
5.2.1	Home Page	17

5.2.2 Sign Up	18
5.2.3 Login	19
5.2.4 Main page	20
6. LOW LEVEL DESIGN	21
6.1 Introduction	21
6.2 Design Description	21
6.2.1 Home module	21
6.2.1.1 Home()	21
6.2.1.2 Main()	22
6.2.2 Logistic Regression	23
6.2.2.1 process()	23
6.2.3 Neural Network	23
7. TEST STRATEGY AND TEST PLAN DOCUMENT	24
7.1 Introduction	24
7.2 Test Strategies	24
7.3 Performance Criteria	25
7.4 Test Environment	25
7.5 Roles and Responsibilities	26
7.6 Test Schedule	26
7.7 Test Tools Used	26
7.8 Acceptance Criteria	26
8. IMPLEMENTATION/CODE	27
8.1 Tools and Technology	27
8.1.1 tkinter	27
8.1.2 numpy	27
8.1.3 sqlite3	27
8.1.4 csv	27
8.1.5 pandas	28
8.1.6 keras	28
8.1.7 sklearn	28
8.1.8 Nltk	28
8.2 Working	28
8.2.1 Home.py	28
8.2.2 createdatasets.py	29
8.2.3 LogisticRegression.py	30

8.2.4 NeuralNetwork.py	31
8.2.5 Predict Author.py	32
9. TESTING	33
9.1 Test Data	33
9.2 Test Case List	33
10. RESULT AND DISCUSSION	37
11. SNAPSHOTS	39
12. CONCLUSION	41
13. FURTHER ENHANCEMENT	42
14. BIBLIOGRAPHY/REFERENCES	43





## LIST OF TABLES

Table No.	Title	Page No.
4.1	Scenario 1	11
4.2	Scenario 2	12
4.3	Scenario 3	12
4.4	Scenario 4	12
4.5	Scenario 5	13
4.6	Software Requirements	13
6.1	Data Members	21
6.2	Data Members	22
9.1	Testcase list	33

## LIST OF FIGURES

Figure No.	Title	Page No.
4.1	Architecture Diagram	16
5.1	Use Case Diagram	16
5.2	Home Page	17
5.3	Sign Up Page	18
5.4	Login Page	19
5.5	Main Page	20
8.1	Code	29
8.2	Code	30
8.3	Code	31
8.4	Code	32
8.5	Code	32
10.1	Screenshot of Logistic Regression	37
10.2	Screenshot of Neural Network	38
11.1	Snapshot of home page	39
11.2	Snapshot after creating dataset	39
11.3	Snapshot of predicted result	40
11.4	Snapshot of NN and LogReg graphs	40

## **CHAPTER- 1**

### **INTRODUCTION**

Authorship Attribution (AA) is a well-contemplated issue among Natural Language Processing scientists which goes back to the earliest endeavors at quantitative examination of content records.

The field of Authorship Attribution started at some point in the late nineteenth century with the principal endeavors to evaluate composing style. Enthusiasm for the field was principally created by the craving for progressively target avocations for the initiation of questioned works by creators, for example, Shakespeare.

Measurable strategies were overwhelming in the field when of the early investigations of Zipf and Yule of relative word frequencies, which found that the recurrence with which a word showed up in a content could be approximated with a Poisson circulation. A great investigation on the origin of the Federalist Papers by Mosteller and Wallace in 1964 by means of Bayesian examination of recurrence checks of the most ordinarily utilized words demonstrated intensely powerful on the field and lead to many years of the construction of 'stylometry,' basically point by point, manual element engineering.

The two most ordinarily referred to creation rivalries are the Ad-hoc Authorship Attribution Competition and the PAN (International Workshop on Plagiarism Detection, Author Identification, and Near-Duplicate Detection), which have been kept running in 2004 and annually from 2007 individually. There are a few basic true applications including literary theft discovery, identification of creators of dangers, verification of suicide notes, forensics, and intelligence insights, where it is widely utilized and contemplated. These patterns, just as expanded help by institutions including law authorization and government, have prompted a resurgence of the sub field and greatly improved results on open benchmarks.

Authorship Attribution is a well-known field in NLP and has made significant progress in this domain. The fundamental thought behind factually or computationally-bolstered Authorship Attribution is that by estimating some literary highlights that we can recognize writings composed by various writers. The main endeavours to measure the composition style return to nineteenth century, with the spearheading investigation of Mendenhall (1887) on the plays of Shakespeare pursued by factual examinations in the principal half of the twentieth century by Yule (1938; 1944) and Zipf (1932).

## **CHAPTER- 2**

### **PROBLEM DEFINITION**

The objective is to coordinate unknown text with its writer by means of some comparability estimation gained from marked content composed by the equivalent per-child. The purpose of the project is to develop a tool which efficiently identifies the author based on their style of writing. Most of the authors have a unique style of writing which forms the basis of this project. Identification of the appropriate author given any article as the input has the lot of importance in today's world.

## **CHAPTER- 3**

### **LITERATURE SURVEY**

Most of the related works focuses on identifying authors using various ML techniques and stylometric analysis. They give various approaches to implement authorship attribution. We have identified some suitable methods and implemented in our project. A brief summary of few papers containing their approach, methodologies, result and conclusion are discussed in the following section.

#### **3.1 Author Attribution with CNN's**

Factual strategies including Naive Bayes, compression models, and various distribution similarities have been utilized to investigate n-gram means Authorship Attribution. Most as of late, a plenty of models more recognizable to AI specialists than etymologists, for example, Support Vector Machines, Latent Dirichlet distribution, Decision Trees, and Neural Networks have been connected to various kinds of word embeddings with progress.

The methodology is to use Pre-prepared feature extractors that could be adequately cross-connected to different tasks in vision and language processing. The results obtained are firstly, a self-gathered dataset was an accumulation of open area books sourced from Project Gutenberg. The dataset involves eight books composed by six separate writers and is somewhat unequal for the authors with more than one included work, in spite of the fact that not very so. They were chosen for homogeneity of theme and time of distribution with the goal that the system would be compelled to essentially depend after composition style disambiguation as opposed to increasingly shallow highlights. The execution of the model on the dataset was consoling in that it effectively outperformed the dimension of an irregular and even a fairly robust classifier. Two systems over the dataset, one with backpropagation empowered just up to the convolutional layer and another through the input were prepared.

Secondly, the dataset of 2012 PAN author identification challenge was used.[1] The documents concern a solitary point (or genre at any rate), so they are not inconsequentially detachable dependent on the consideration of very specific words, an issue which tormented early initiation recognizable proof examinations. Here, the dropout probability,  $p=0.5$  to  $p=0.75$ , diminished overfitting sufficiently long to improve sentence arrangement exactness by about 3%. The design and filter bank sizes were proportional to those of the initially depicted model. A heat map of the confusion matrix produced by the final system over the test set was created to verify the results.

### **3.2 Authorship Attribution with Topic Models**

Origin attribution has pulled in much consideration because of its numerous applications in, for instance, PC criminology, criminal law, military insight, and humanities look into. The customary issue, which is the focal point of this article, is to credit mysterious test writings to one of a lot of realized competitor writers, whose preparation writings are provided ahead of time (i.e., directed characterization). Though the vast majority of the early work on origin attribution concentrated on formal writings with just a couple of applicant writers, analysts have as of late directed their concentration toward situations including casual writings and tens to thousands of writers. In parallel, point models have picked up fame as a methods for finding topics in such huge content corpora. This article investigates initiation attribution with theme models, broadening the work introduced by Seroussi and partners by detailing extra trial results and utilizations of point based writer portrayals that go past customary origin attribution. The Methods used are ,

- Methods Based on LDA
- Methods Based on AT (Author Topic Distributions)
- Methods Based on AT-FA
- Methods Based on DADT

In rundown, we found that the DADT-based probabilistic methodology (DADT-P) yielded solid execution on the five informational collections we considered, outflanking the Token SVM gauge in four out of the five cases.[2] We demonstrated that DADT-P is more appropriate for initiation attribution than techniques dependent on LDA and AT (with or without invented creators), and then

utilizing DADT for dimensionality reduction. Despite the fact that our results exhibit that isolating archive words from creator words is a decent way to deal with origin attribution, depending just on unigrams is a constraint (which is shared by LDA, AT, and DADT).

DADT's improved execution in examination with strategies dependent on LDA and AT includes some major disadvantages of more parameters to tune. In any case, the most significant parameter is the quantity of themes—we found that the earlier qualities that yielded great outcomes on the little informational indexes likewise gotten great execution on the extensive informational collections moving forward without any more tuning. We offered a basic formula to decide the quantity of subjects for DADT-P: First keep running AT-P to locate the general number of themes (which is identical to running DADTP without archive points), and afterward tune the report/creator subject parity.

### **3.3 Authorship Attribution Using Principal Component Analysis And Competitive Neural Networks**

The starting point of non-customary origin attribution, or stylometry, is regularly said to be Augustus de Morgan's proposal that specific creators of the Bible may be discernible from each other. A trademark distinction in the circulation of various estimated words in Shakespeare and Marlowe were about unclear, they were both altogether and reliably not the same as Bacon. The thing that matters was principally seen in the general recurrence of three- and four-letter words: Shakespeare utilized progressively four - letter words and Bacon increasingly three-letter words.

G. U. Yule conceived an element known as "Yule's trademark K," which evaluated "vocabulary richness" by contrasting word frequencies with that normal by a Poisson conveyance, yet like Mendenhall's word lengths, this excessively was later observed to be an inconsistent marker of style. In the historical backdrop of origin ponders, it is demonstrated that Burrows technique for principle component analysis (PCA) is exceptionally effective to remove the redundant information. The principle component analysis (PCA) essentially involves computing the frequency of each of a list of function words and performing principle component analysis (PCA) to find the linear combination of variables that best accounts for the variations in the data. Rather than analyze this result statistically, the transformed data are simply plotted. Two-dimensional plots of the first two



principal components supply us with a means to inspect visually for trends, which occur as clusters of points. Later, cluster analysis may follow this step. This simple but effective method continues to be used today, partly because of the ease with which the results are communicated and interpreted. He confirmed this suspicion independently, demonstrating that Thompson was much more prone to use position words such as “up,” “down,” “over,” and “back,” than Baum.[3] This was not demonstrated using complex statistical techniques; rather, function word frequencies were tallied, the authors’ tallies compared, PCA used to reduce the dimensionality of the data, and the resulting plots inspected: the two authors’ works form obvious clusters. In this paper instead of cluster analysis of the two-dimensional plots, the author attribution will be found by the use of artificial neural networks with output neurons competing on the data of first principal components.

The examination depicted in this paper concerning creator recognizable proof analysis demonstrates that the technique for primary component analysis (PCA), when pursued by a artificial neural system is an proficient tool. In this way a progression of future analyses ought to incorporate more extensive scope of creators, meaning of new arrangements of literary descriptors, and test for different sorts and structures of neural systems, and search the possibility of inheritance through interpretation into different dialects.

### **3.4 .Machine learning approach to authorship attribution of literary texts**

Stylometry indicates quantitative analysis of composed content that yields data about the style it is formed with and through that about the author of this content. In this way as the fundamental stylometric assignments, having a place inside data recovery space, there are viewed as author characterisation, similitude recognition, lastly, considered as the most significant, author recognizable proof. Author characterisation obtains details about the author, for example, gender, training, social foundation and so forth. Similarly recognition includes contrasting writings of a few authors so as to discover, in the event that they exist, a few properties in normal. Author recognizable proof (or attribution) implies ascribing an obscure content to an essayist basing on some element trademark or measure. It very well may be utilized when a few people guarantee to have thought of some content or when nobody is capable or willing to distinguish the genuine author of this content. Stylometry is frequently utilized for discovery of counterfeiting, discovering

authors of secretly distributed writings, for questioned authorship of writing or in criminal examinations inside scientific semantic space. Two basic issues of the stylometric analysis are: determination of descriptors that describe messages and authors, and diagnostic strategies connected to the assignment.

Markovian Models consider a content as a succession of characters that relates to a Markov chain . In probabilistic model of natural language letters show up with some likelihood, contingent upon which characters go before them. In the least complex model there is viewed as just the quick antecedent which offers ascend to the first request Markov chain. In this manner for all sets of letters in the letter set there are acquired networks of change frequencies of one letter into another. These measurements are determined for all writings by known writers and for some unattributed message as the genuine creator there is chosen the one with the most noteworthy likelihood . Techniques, for example, Linear Discriminant Analysis, Principal Component Analysis or bunch analysis mean to lessen the dimensionality for information and if techniques connected to writings of both known and obscure writers give the equivalent result, the topic of recognition of author is settled.

The research described in this paper concerning stylometric analysis shows beyond doubt how efficient a tool Artificial Neural Networks can be when applied in classification tasks[4]. However ends with regards to the decision of literary descriptors utilized as highlights for acknowledgment process, in view of on results displayed in the past segment and prompting some subjective explanation that syntactic traits are progressively powerful in attribution.

## **CHAPTER- 4**

# **CUSTOMER REQUIREMENT SPECIFICATION**

## **4.1 Introduction**

Purpose of this is to explain the functionalities required to be implemented while building an Authorship Attribution website. Target audience are authors, critics and readers. It will evaluate the design and the features of the system and help come up with optimal solutions to fulfil the requirements of users. This document is intended for both the customers and the developers of the system, as it will help them in further stages of development.

### **4.1.1 Scope**

The authorship attribution website is a platform where given an input text by the user, it predicts the author's name based on the writing style through Machine Learning algorithms. The limitation is, if there is no unique way of writing amongst authors, it becomes difficult to identify the relevant author. It consists of a web application for the user:

1. To have a collection of the books written by 8 authors.
2. To predict the correct author for the given input text of author's book.
3. To manage the articles in a database used for initial training.

## **4.2 Product Perspective**

Other related projects are:

- Classifying the author as male/female based on the text.
- Predicting the success of the text/novel.
- Source Code Author Attribution using Author's Programming Style and Code Smells.

Our product is independent and totally self-contained. It is not a part of any larger system.

Software platforms used for development are macOS, Windows and Linux.

Software platforms used for deployment are macOS, Windows and Linux.

### **4.2.1 User Characteristics**

1. Authors: To identify their style of writing compared to others.
2. Readers: To appreciate the different genres of articles/novels, Smaller search space for identifying personal interest.
3. Critics: To compare authors of the same domain and identify flaws in one's writing.

### **4.2.2 General Constraints, Assumptions and Dependencies**

Assumptions:

The input for prediction of the author's name has to be inserted as only as a text.

General Constraints:

- The number of concurrent users can be upto 100 (Default is 100 in Apache Server but can be changed later in the settings)
- Works on MacOS, Windows, Linux Operating Systems.

Dependencies:

- Python configuration needs to be modified to include \_tkinter module. Python configuration needs to be changed to include the directory that contains Tkinter.py in its default module search path.

### **4.2.3 Risk**

Program Risks:

1. Wrong Identification of an Author because of similar style of writing.
2. Wrong Identification of an Author due to new data of a new author, which does not exist in the database.

Product Risks:

1. Customers would be able to identify the authors with better fluency, relevance to the topics which would discourage them from reading articles/novels by a below average author.

## 4.3 Architecture Diagram

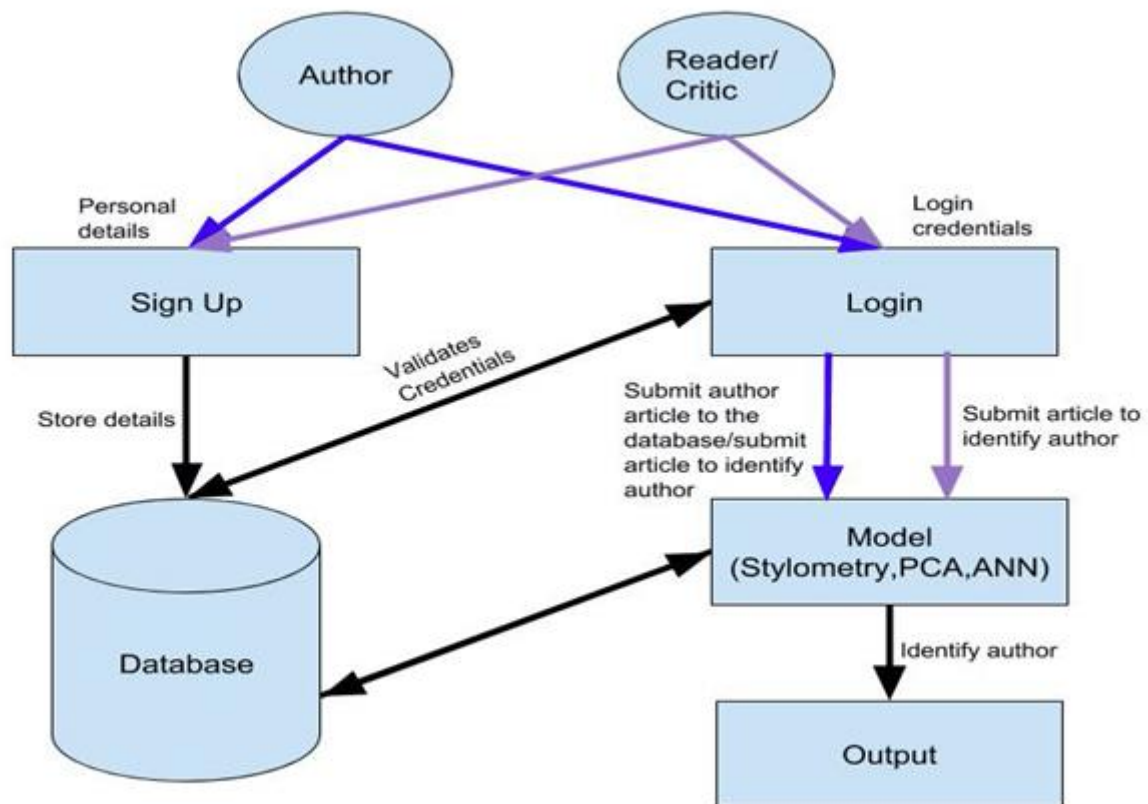


Fig 4.1 Architecture Diagram

## 4.4 Requirements List

### 4.4.1 Scenario 1

<u>Reqmt#</u>	<u>Requirement</u>
<u>SCE – 1</u>	<u>Sign Up as a User</u>

Table 4.1 Scenario 1

#### **4.4.2 Scenario 2**

<u>Reqmt #</u>	<u>Requirement</u>
<u>SCE – 2</u>	<u>Login as a User</u>

Table 4.2 Scenario 2

#### **4.4.3 Scenario 3**

<u>Reqmt#</u>	<u>Requirement</u>
<u>SCE– 3</u>	<u>Submitting Books by an Author to save in the database</u>

Table 4.3 Scenario 3

#### **4.4.4 Scenario 4**

<u>Reqmt #</u>	<u>Requirement</u>
<u>SCE– 4</u>	<u>Upload the input text from the books by an Author/Critic/Reader to predict author name</u>

Table 4.4 Scenario 4

#### **4.4.5 Scenario 5**

<u>Reqmt #</u>	<u>Requirement</u>
<u>SCE- 5</u>	<u>Author/Critic/Reader enters the article in the text area to predict the author name</u>

Table 4.5 Scenario 5

### **4.5 External Interface Requirements**

#### **4.5.1 Hardware Requirements**

- We need a laptop or a computer system having windows or MacOS or Linux operating systems.
- The system requires minimum 2GB of RAM size.

#### **4.5.2 Software Requirements**

<b>Name</b>	<b>Version/Release Number</b>
Database like MYSQL to store the training data.	8.0
User interface module like tkinter	Compatible with python 3.7
Python editor (Jupyter notebook) to code the project.	3.7
Windows/MacOS/Linux Operating Systems.	Windows 10/ v10.12.6/ v4.20.10

Table 4.6 Software Requirements

## **4.6 User Interfaces**

- The Authorship Attribution website broadly comprises of Home, Sign Up, Log In, Our Tool and About Us webpages.
- The home page has a few articles/novels written by certain authors. The viewers can have a glimpse of the content.
- A statistical count of the number of authors, articles, readers, critics will be maintained and displayed.
- The Sign Up page requests the User to enter personal details and create an account. The list of people is maintained in the database for Login credentials validation.
- The Log In page for the User ensures Login credentials validation and successful Login to our main page.
- The main page is where the content is entered in the text area in order to identify the relevant author based on his/her style of writing through Machine Learning Algorithms.
- On selecting “Create Dataset”, the randomly shuffled dataset is created from the books of authors available. “Created Dataset successfully” is displayed in the Notification area.
- On selecting “Browse”, the author\_data.csv file, which is the dataset created, can be uploaded for processing the Machine Learning Algorithms.
- On selecting “Logistic Regression”, the model built would run accordingly, display graphs and results on the terminal. The results are also saved in a CSV file in the same directory. “Logistic Regression Successful” is displayed in the Notification area.
- On selecting “Neural Network”, the model built would run accordingly, display graphs and results on the terminal. The results are also saved in a CSV file in the same directory. “Neural Network Successful” is displayed in the Notification area.
- On entering the input text for author name prediction in “Enter the text”, the model is ready for prediction. Once we “Submit”, the respective author name is notified in the Notification area.
- About Us page comprises the basic terms that readers/critics should know, tells our customers why we are important to them, receive any comments from the customers and more.



## **4.7 Performance Requirements**

- The number of concurrent users can be approximately 50-100 that is supported in Apache server.
- Number of files is a single input text file that is given by the User.
- The records in the database used for training data consist of articles and authors of 8 authors.

## **4.8 Help**

A brief description of the user manual is given on the homepage of the website.

## **4.9 Other Requirements**

### **4.9.1 Site Adaptation Requirements**

We aim to make sure the website runs across different browsers like Google Chrome, Mozilla Firefox, Safari. The website is not dependent on any other external application or software.

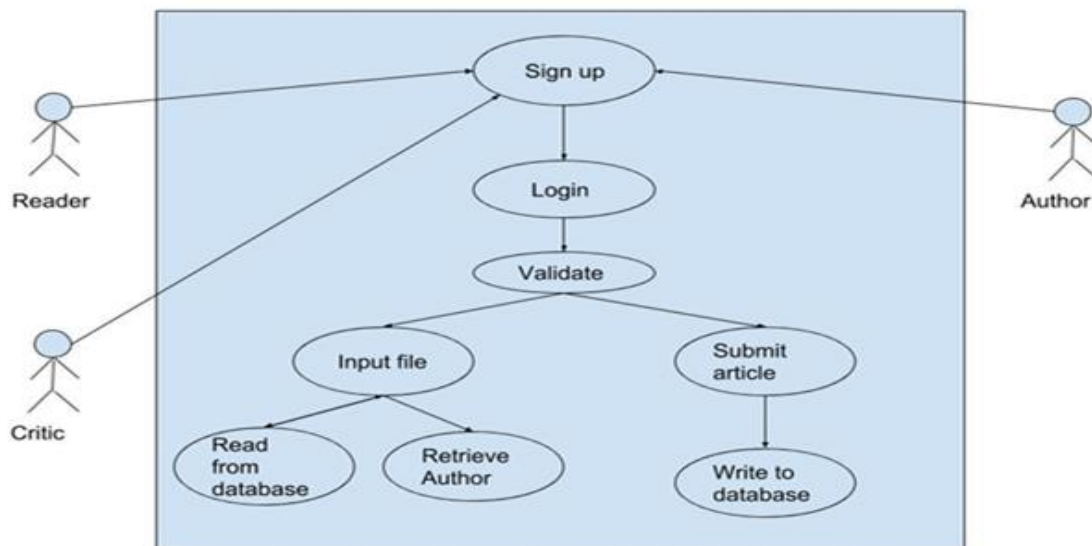
## **CHAPTER- 5**

### **HIGH LEVEL DESIGN**

High-level design (HLD) clarifies the engineering that would be utilized for building up a product item. The design chart gives an outline of a whole framework, recognizing the principle segments that would be created for the item and their interfaces. The HLD utilizes conceivably nontechnical to somewhat specialized terms that ought to be justifiable to the heads of the framework.

#### **5.1 Use Case Diagram**

The use case diagram depicts how the author, reader and critic can use the tool to identify



authors.

Fig 5.1 Use Case Diagram

## 5.2 User Interfaces

### 5.2.1 Home Page

Home page is the first page that gets loaded once you start running the tool. It has a list of options where the user can login or a new user can sign up. The user can also quit by selecting the quit option.

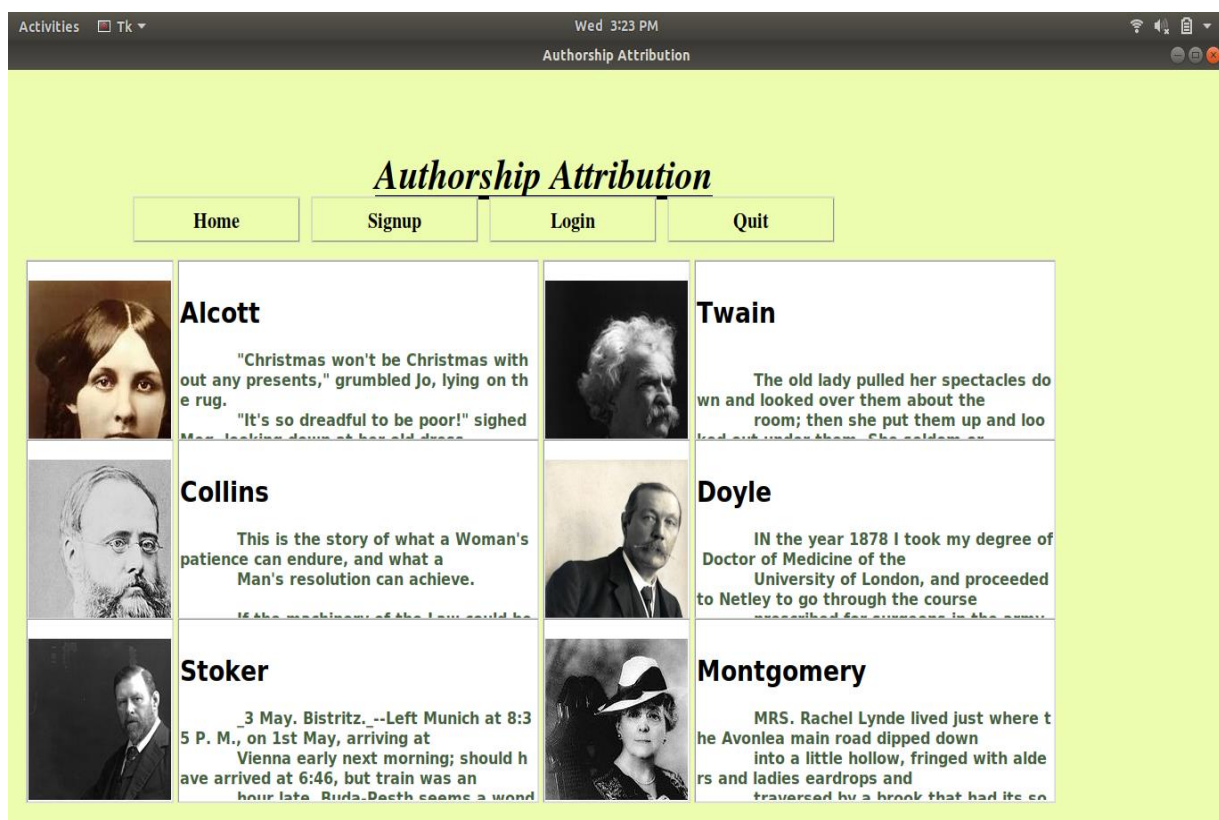
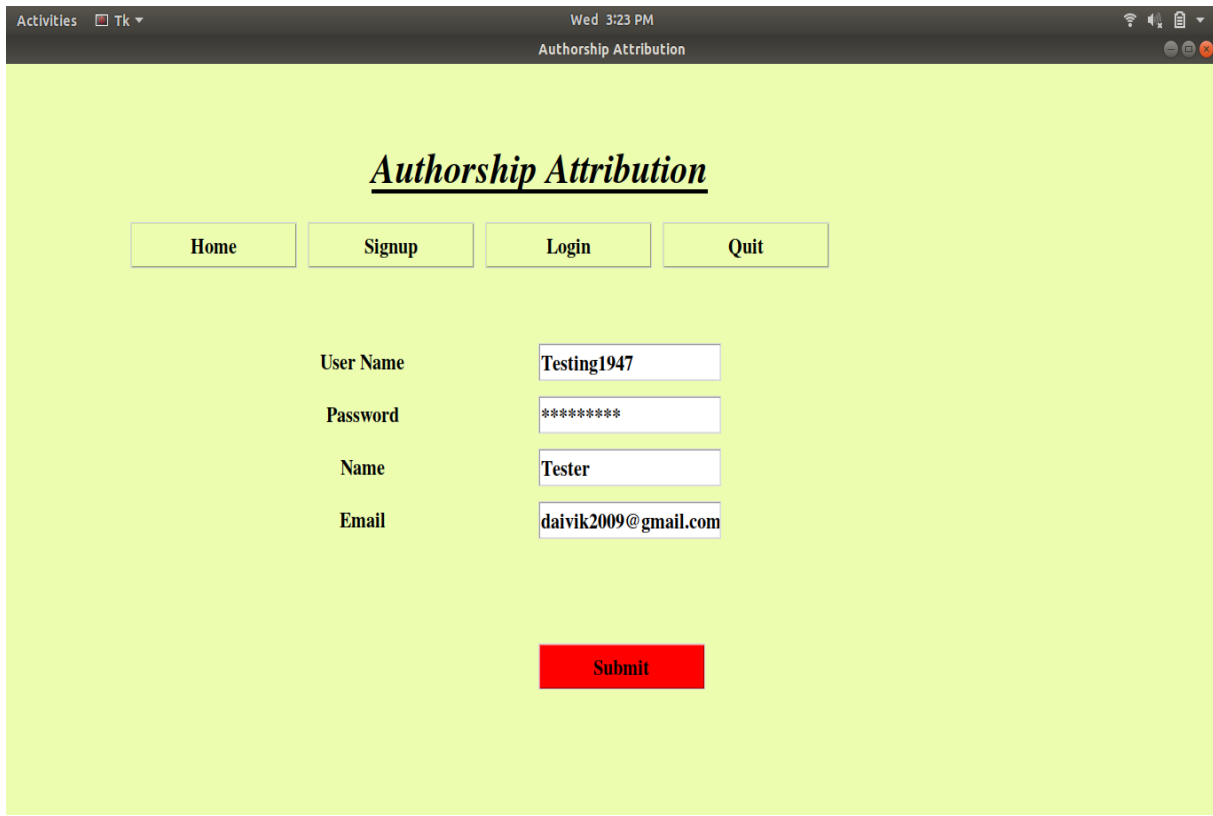


Fig 5.2 Home Page

### 5.2.2 Sign Up

This is a sign up page where a new user can register by signing up providing the following details as mentioned in the diagram below.



The screenshot shows a web browser window titled "Authorship Attribution". The page has a light green background. At the top, there is a navigation bar with four buttons: "Home", "Signup", "Login", and "Quit". Below the navigation bar, the title "Authorship Attribution" is displayed in a large, bold, italicized font. Underneath the title, there are four input fields for registration details: "User Name" (containing "Testing1947"), "Password" (containing "\*\*\*\*\*"), "Name" (containing "Tester"), and "Email" (containing "daivik2009@gmail.com"). A red "Submit" button is located at the bottom of the form.

Fig 5.3 Sign Up page

### 5.2.3 Login

Login page contains user name and password. The user with valid credentials can successfully login.

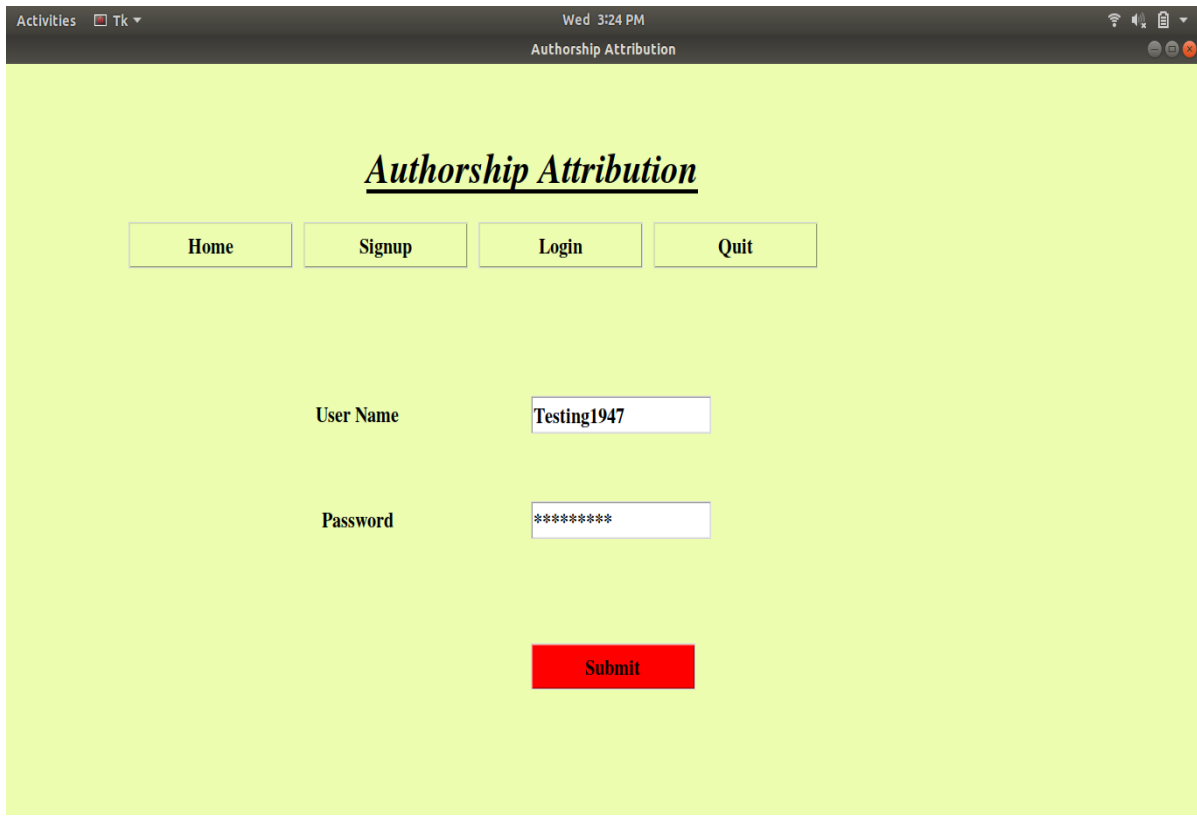


Fig 5.4 login page

### 5.2.4Mainpage

Once the user signs up and logs in he is directed to the main page. This is the main page where we can create dataset apply Logistic regression or Neural network to train the dataset and submit lines from the articles to find the author. This is the main user interface for the tool that we are developing. After training the model the user enters the sentences in the enter text field to get to know who the author is. The main functionality of the project happens through this interface.

Activities Tk ▾ Fri 10:57 PM Authorship Attribution

## *Authorship Attribution*

Select Dataset

Enter the text

Notification :

Fig 5.5 Main page

## **CHAPTER- 6**

# **LOW LEVEL DESIGN**

## **6.1 Introduction**

### **6.1.1 Overview**

Low-level design (LLD) is a segment level design process that follows a well ordered refinement process. This procedure can be utilized for designing information structures, required software architecture, source code and eventually, execution calculations. By and large, the information association might be characterized amid prerequisite investigation and afterward refined amid information design work.

### **6.1.2 Purpose**

The main objective of LLD is to give insight into the code and how it is structured. It describes about the class diagrams and their relation between each classes. It gives a logical solution to how the problem is solved. The code can be implemented directly using the low level diagram with a very less changes or modifications.

## **6.2 Design Description**

### **6.2.1 Home module**

#### **6.2.1.1 Home() Data members**

Data Type	Data Name	Access Modifiers	Initial Value	Description
object	window	global	-	it is used to create create object for TK() class from tkinter module
string	bgcolor	private	#ECFD0	Used to specify background color
string	fgcolor	private	black	Used to specify foreground color

Fig 6.1 Data members

### **Inner Methods**

#### **login()**

- Calls Login method when the user selects login option from home page

#### **home()**

- Calls Home method when the user selects home option

#### **signup()**

- Calls Signup method when the user selects signup option from home page

### **6.2.1.2 Main()**

#### **Data members**

Data Type	Data Name	Access Modifiers	Initial Value	Description
object	window	global	-	it is used to create create object for TK() class from tkinter module
string	bgcolor	private	#ECFD B0	Used to specify background color
string	fgcolor	private	black	Used to specify foreground color

#### 6.2 Data Members

### **Inner Methods**

#### **submit()**

- Submit the value once the dataset is selected

#### **browse()**

- Function to browse for selecting the dataset

#### **createdataset()**

- Function to create the dataset from the raw data

#### **nnet()**

- Function to call neural network module

#### **logreg()**

- Function to call logistic regression module



## **6.2.2 Logistic Regression**

### **6.2.2.1 process()**

This function implements logistic regression for the dataset. It takes the path of the created dataset as input, performs, tf idf vectorization on the data, passes it to PCA and then perform logistic regression. It will split the data for training and testing and train the model first. Then use test data for testing the model.

## **6.2.3 NeuralNetwork.py**

### **6.2.3.1 process()**

This function implements neural network on the created dataset. The input to the function is the path of the dataset, if performs tf idf vectorization, passes it to PCA for dimensionality reduction and performs neural network on the data. Relu activation function can be used since it produces better accuracy

## **CHAPTER- 7**

# **TEST STRATEGY AND TEST PLAN DOCUMENT**

## **7.1 Introduction**

Purpose of this is to explain the how the testing of the entire application will happen to ensure that there is no flaw in the application. Target audience are authors, critics and readers. It will evaluate the implementation of the overall project. This document is intended for both the customers and the developers of the system, as it will provide the way to make sure that nothing breaks down.

## **7.2 Test Strategies**

The test strategies which have been planned to be used in the project include:

Types of Testing to be used for testing the functionality:

1. White Box Testing: to ensure each Authorship Attribution Framework works as intended. All the main types of white box testing are done here –

a. Static Testing:

i. Done by humans to find any errors in the AA code, check if code conforms to planned AA design, check error handling capabilities of the framework using test cases with all different input data types. This is done through desk checking, code walkthroughs and formal inspection once every piece of code of the AA framework is developed.

ii. By Static Analysis Tools to detect coding errors like not freeing memory after use, unused variables and unreachable code among other errors. It needs to be done as soon as a module of code is developed. Manually examine each class of our AA tool to find errors in AA code, further possible optimizations and check error handling capabilities of our AA tool. Use of static analysis tools pylint to detect static coding errors in tool.

b. Structural Testing:

i. Code Functional Testing: Test scripts to test the functionality of every function of every written module of the AA Library are created and added to the test suite to detect any errors in the code functionality considering the internal data structures.

ii. Code Coverage Testing: Is to be done after most of the code base has been developed to ensure

that a higher percentage of the AA framework code is covered by the test cases.

2. Black Box Testing: is to be done to ensure the requirements of the various classes of the AA framework work as intended. We are to incorporate this in our project by using

a. Requirements based testing: to be done to test the various requirements specified in the SRS of the GA framework. Eg: Test if User is able to provide his/her input text for prediction.

b. Positive/negative testing: to be done to test that the code works under positive and negative conditions.

c. User Documentation Testing: to test the documentation created for end users so as to allow users to easily. Need to review code regularly.

### 3. Error Handling Tests

To ensure that all possible error conditions are handled by raising valid exceptions

Eg: 1. Login Validation of input.

2. Numerical/Audio/Video data provided.

3. Accessing the prediction results without entering the text for prediction.

4. GUI Tests

- To verify form validation tests – form does not support on invalid input conditions.

- To verify Model evaluation of the User Interface on selecting “Logistic Regression” and “Neural Network” buttons.

## **7.3 Performance Criteria**

This section shall clearly define the Performance criteria as defined in Contract or CRS.

1. The website system should be able to support as many as 1000 concurrent users with average response times of not more than 5 seconds.

2. The website system should be very user friendly.

## **7.4 Test Environment**

OS: Ubuntu 16.04, Windows, Mac.

Hardware: Linux system with a fast processor(2GHz) and at least 4GB RAM with large memory.

Software: Defined Python3, python modules.

## **7.5 Roles and Responsibilities**

This section shall identify roles and responsibilities of all parties involved in testing. Daivik, Sushmitha and Tejaswini to work parallel on different testing tasks as when different modules are developed.

## **7.6 Test Schedule**

This section shall describe the schedule planned for testing activities.

March 1st – 10 April: Basic Class Design, Test case generation.

15 April – 25 April: White and Black Box Testing.

18 April – 22 April: System/Acceptance Testing.

## **7.7 Test Tools Used**

This section shall describe the test tools required / used for testing the product.

1. pyTest: Can be used for various unit tests, integration testing.
2. Coverage: For carrying out code coverage tests.
3. Pylint: For static python code testing.

## **7.8 Acceptance Criteria**

This section shall describe the acceptance criteria for releasing the product to the next phase.

1. The website system should be able to support as many as 100 concurrent users with average response times of not more than 5 seconds.
2. The website system should be very user friendly.
3. The API should be very well documented

## **CHAPTER- 8**

### **IMPLEMENTATION/ PSEUDO CODE**

#### **8.1 Tools and Technology**

Some of the important tools and technologies used in the project are explained below:

##### **8.1.1 tkinter**

tkinter is a module which is an interface for TK graphical user interface toolkit. The TK is instantiated in the beginning without arguments and it creates a main window which acts as a global window for the application. The TK has button, canvas, radio button, scale etc. We have used some of these features for our project.

##### **8.1.2 numpy**

numpy is the most popular library in python which is used for scientific computing. Its uses include:

A useful N-dimensional array object

Broadcast functions

Tools which maybe required for integrating C and Fortran Codebase

Fourier transform, linear algebra and rand function capabilities.

##### **8.1.3 sqlite3**

SQLite3 can be incorporated with Python utilizing sqlite3 module. It furnishes a SQL interface consistent with the DB-API 2.0 particular portrayed by PEP 249. To utilize sqlite3 module, you should initially make an association object that speaks to the database and after that alternatively you can make a cursor object, which will help you in executing all the SQL explanations.

##### **8.1.4 csv**

The csv module is used to read and write data in the tabular form in csv format. It is the solution available for importing and exporting databases and spreadsheets. The csv object iterates over each data entry and reads the data from the csv file.

### **8.1.5 pandas**

pandas is a Python package giving quick, adaptable, and expressive information structures intended to make working with organized (forbidden, multidimensional, possibly heterogeneous) and time arrangement information both simple and natural. It expects to be the key abnormal state building hinder for doing down to earth, genuine information examination in Python.

### **8.1.6 keras**

keras is a library written in python. It is a open source library containing various implementations of frequently used neural network building blocks such as layers optimizers etc. keras has high importance when we need deep learning library that allows for fast prototyping and run seamlessly on CPU and GPU.

### **8.1.7 sklearn**

Scikit-learn gives a scope of directed and unsupervised learning calculations by means of a steady interface in Python. The main vision of the library is a level of robustness and support required for use in production systems.

### **8.1.8 Nltk**

The Natural Language Toolkit is more frequently used to handle the free flowing English Language. This toolkit is written in Python programming language. This toolkit is a suite of libraries and program for symbolic and statistical natural language processing.

## **8.2 Working**

### **8.2.1 Home.py**

The execution of the application starts through home.py. This program creates the UI through tkinter module. The home page starts executing and the direction of program flow continues according to the user input depending on whether he signs up or logs in. The home page window calls the main window page once the user logs in. In the main window page the user can create dataset, perform logistic regression or neural network and submit text sentences to identify the author of the

text.

```

151
152 def Home():
153     global window
154     bgcolor="#E6FDB0"
155     fgcolor="black"
156     window = tk.Tk()
157     window.title("Authorship Attribution")
158     window.geometry('1280x720')
159     window.configure(background=bgcolor)
160
161     window.grid_rowconfigure(0, weight=1)
162     window.grid_columnconfigure(0, weight=1)
163
164     def login():
165         print("Login")
166         window.destroy()
167         Login()
168
169     def home():
170         print("Home")
171         window.destroy()
172         Home()
173
174     def signup():
175         print("Signup")
176         window.destroy()
177         Signup()
178
179     message1 = tk.Label(window, text="Authorship Attribution", bg=bgcolor, fg=fgcolor, width=50, height=3, font=('times', 30, 'italic bold underline'))
180     message1.place(x=100, y=10)
181
182
183     home = tk.Button(window, text="Home", command=home, fg=fgcolor, bg=bgcolor, width=16, height=1, activebackground = "Red", font=('times', 15, ' bold '))
184     home.place(x=140, y=150)
185
186     signup = tk.Button(window, text="Signup", command=signup, fg=fgcolor, bg=bgcolor, width=16, height=1, activebackground = "Red", font=('times', 15, ' bold '))
187     signup.place(x=340, y=150)
188
189
190     login = tk.Button(window, text="Login", command=login, fg=fgcolor, bg=bgcolor, width=16, height=1, activebackground = "Red", font=('times', 15, ' bold '))
191     login.place(x=540, y=150)

```

Fig 8.1 code

## 8.2.2 createdatasets.py

This program reads all the data from text files of each author when the user submits the path of the text files through browse option. It splits every line of the text file and stores the data of each line along with the author in a csv file. The data is randomly shuffled to train the model better. The dataset consists of books from 8 different authors , Alcott, Twain, Collins, Doyle, Stoker, Montgomery, Austen, Bronte. The book names along with their page number count are , Dracula: 15031, Emma: 15937, Hound\_of\_the\_Baskervilles: 7181, Huckleberry\_Finn: 11219, Jane\_Eyre: 20290, Little\_Women: 20269, Pride\_and\_Prejudice: 12721, Sign\_of\_the\_Four: 4455, Study\_in\_Scarlet: 4548, Tom\_Sawyer: 8203, Woman\_in\_White: 25631. Total lines will be 1,65,350.

### Authorship Attribution

```

2 from nltk import tokenize
3 import numpy as np
4 import random
5 import pandas as pd
6 import nltk
7 nltk.download('punkt')
8 def split_text(filepath, min_char):
9
10     file = open(filepath, "r", encoding="utf8")
11     text = file.read().replace('\n', ' ')
12     text = text.replace('.', '.').replace('.', '.').replace('?', '?').replace('!', '!')
13     text = text.replace('--', ' ').replace('...', ' ').replace('_', ' ')
14     file.close()
15
16     sentences = tokenize.sent_tokenize(text)
17
18     sentences = [sent for sent in sentences if len(sent) >= min_char]
19
20     return list(sentences)
21
22 def process():
23
24     min_char = 5
25
26     alcott = split_text('Books/Little_Women.txt', min_char = min_char)
27     austen = split_text('Books/Pride_and_Prejudice.txt', min_char = min_char)\
28         + split_text('Books/Emma.txt', min_char = min_char)
29     bronte = split_text('Books/Jane_Eyre.txt', min_char = min_char)
30     collins = split_text('Books/Woman_in_White.txt', min_char = min_char)
31     doyle = split_text('Books/Study_in_Scarlet.txt', min_char = min_char)\
32         + split_text('Books/Sign_of_the_Four.txt', min_char = min_char)\
33         + split_text('Books/Hound_of_the_Baskervilles.txt', min_char = min_char)
34     montgomery = split_text('Books/Anne_of_Green_Gables.txt', min_char = min_char)\
35         + split_text('Books/Anne_of_Avonlea.txt', min_char = min_char)
36     stoker = split_text('Books/Dracula.txt', min_char = min_char)
37     twain = split_text('Books/Tom_Sawyer.txt', min_char = min_char)\
38         + split_text('Books/Huckleberry_Finn.txt', min_char = min_char)
39
40

```

Fig 8.2 Code

### 8.2.3 LogisticRegression.py

After creating the feature(text) and label(author) lists, we permute and shuffle the dataset for randomization of the dataset. Each author is assigned a label, say 0 to 7. We use TF IDF Vectorizer for the extraction of features of text data. This results in vectorized dataset formation. PCA is affected by scale, so we have to scale the highlights in our information before applying PCA. Using StandardScaler enables us to institutionalize the dataset's highlights onto unit scale (mean = 0 and fluctuation = 1), which is a necessity for the ideal execution of many AI calculations. This converts the text data into the right format. Usually, the split between training and testing set is: 80% training and 20% test. Here, we chose 6/7th for training data and 1/7th for testing data. Perform logistic regression on the data after all the processing. Lastly, we predict the test results and save the model for later use. MSE VALUE, MAE VALUE, R-SQUARED VALUE, RMSE VALUE, ACCURACY VALUE for the Neural Network is observed in the csv file and plotted as a graph.



```

73
74     train_img = scaler.transform(train_img)
75     test_img = scaler.transform(test_img)
76
77
78     from sklearn.decomposition import PCA
79
80     pca = PCA(.95)
81
82     pca.fit(train_img)
83
84     train_img = pca.transform(train_img)
85     test_img = pca.transform(test_img)
86
87     print(train_img.shape)
88
89
90     logisticRegr = LogisticRegression(solver = 'lbfgs')
91
92     logisticRegr.fit(train_img, train_lbl)
93
94
95     y_pred=logisticRegr.predict(test_img)
96     print(y_pred)
97
98     result2=open("results/resultLogisticRegression.csv","w")
99     result2.write("ID,Predicted Value" + "\n")
100     for j in range(len(y_pred)):
101         result2.write(str(j+1) + "," + str(adata[y_pred[j]]) + "\n")
102     result2.close()
103
104     mse=mean_squared_error(test_lbl, y_pred)
105     mae=mean_absolute_error(test_lbl, y_pred)
106     r2=r2_score(test_lbl, y_pred)
107
108
109     print("-----")
110     print("MSE VALUE FOR Logistic Regression IS %f " % mse)
111     print("MAE VALUE FOR Logistic Regression IS %f " % mae)
112     print("R-SQUARED VALUE FOR Logistic Regression IS %f " % r2)

```

Fig 8.3 Code

## 8.2.4 NeuralNetwork.py

After creating the feature(text) and label(author) lists, we permute and shuffle the dataset for randomization of the dataset. Each author is assigned a label, say 0 to 7. We use TF IDF Vectorizer for the extraction of features of text data. This results in vectorized dataset formation. PCA is affected by scale, so we have to scale the highlights in our information before applying PCA. Using StandardScaler enables us to institutionalize the dataset's highlights onto unit scale (mean = 0 and fluctuation = 1), which is a necessity for the ideal execution of many AI calculations. This converts the text data into the right format. Usually, the split between training and testing set is: 80% training and 20% test. Here, we chose 6/7th for training data and 1/7th for testing data. After processing data implement neural network model. We have chosen relu as activation function because it has better accuracy. Lastly, we predict the test results and save the model for later use. MSE VALUE, MAE VALUE, R-SQUARED VALUE, RMSE VALUE, ACCURACY VALUE for the Neural Network is observed in the csv file and plotted as a graph.

## Authorship Attribution

```

116 scaler.fit(X_train)
117
118 X_train = scaler.transform(X_train)
119 X_test = scaler.transform(X_test)
120
121
122 from sklearn.decomposition import PCA
123
124 pca = PCA(.95)
125
126 pca.fit(X_train)
127
128 X_train = pca.transform(X_train)
129 X_test = pca.transform(X_test)
130
131
132 print(X_train.shape)
133
134 model = Sequential()
135 model.add(Dense(activation="relu", input_dim=X_train.shape[1], units=1, kernel_initializer="uniform"))
136
137 model.add(Dense(8, activation="sigmoid"))
138
139
140 model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
141
142 print(model.summary())
143 history = model.fit(X_train, y_train, epochs=100, batch_size=10, validation_split=0.2, callbacks=[EarlyStopping(monitor='val_loss', patience=7, min_delta=0.0001)])
144
145
146 y_pred = model.predict(X_test)
147 print(y_pred)
148 print("ytest")
149 print(y_test)
150
151 model.save('model.h5')
152 print(y_pred)
153 labels = ["Alcott", "Austen", "Bronte", "Collins", "Doyle", "Montgomery", "Stoker", "Twain"]
154 for v in y_pred:
155     print(v, labels[np.argmax(v)])

```

Fig 8.4 Code

## 8.2.5 Predict Author.py

Once we train using logistic regression or neural network model, then we use decision tree classifier to predict the name of the author. Decision tree classifier is used because it has only binary branches and hence classification becomes little easy and faster. Once the author name is predicted the name of the author is returned to the main window screen.

```

26
27
28 X = pad_sequences(sequences, maxlen=max_len)
29
30 X_train=X
31 y_train=y
32
33 seq = tokenizer.texts_to_sequences(input)
34 padded = pad_sequences(seq, maxlen=max_len)
35
36
37 model2= DecisionTreeClassifier()
38 model2.fit(X_train, y_train)
39 y_pred = model2.predict(padded)
40 labels = ["Alcott", "Austen", "Bronte", "Collins", "Doyle", "Montgomery", "Stoker", "Twain"]
41 print(y_pred, labels[int(y_pred)])
42
43 return labels[int(y_pred)]
44

```

Fig 8.5 Code

## **CHAPTER- 9**

# **TESTING**

### **9.1 Test Data**

Test Data includes set of author text data shuffled in random fashion. Random row is selected from the authordata.csv file for prediction of the author. Using this author name can be identified. The results can be checked with the test-author mapping in the csv file.

### **9.2 Test Case List**

We have followed the gherkins syntax to write our test cases in the below table.

Test case Number	Testcase	Output
<b><u>1.</u></b>	Scenario Outline: Executing the python file on the terminal When the User opens the terminal And runs the python file	The user interface opens with the home page
<b><u>2.</u></b>	Scenario Outline: Clicking on the Home button Given the user is on any other page And when the user clicks on the home button	The home page appears and the popular articles of different authors appear
<b><u>3.</u></b>	Scenario Outline: Clicking on the Sign Up button When the user clicks on the Sign Up button	The Sign Up page should appear with all the required fields
<b><u>4.</u></b>	Scenario Outline: Registering a User in the Sign Up page Given the user is on the Sign Up page	The details are inserted in the database and is seen on the terminal and the

	When the user enters the details And clicks on Submit	user is registered
<u>5.</u>	Scenario Outline: Registering a User in the Sign Up page with the username that already exists Given the user is on the Sign Up page When the user enters the details and gives the same user name which already exists And clicks on Submit	An appropriate error message appears
<u>6.</u>	Scenario Outline: Existing user trying to log in When the user clicks on the login button Then the login page should appear with the respective fields When the user enters the details And clicks on submit	The credentials are validated from the database and login is successful when the user enters the right details and the login is unsuccessful when the user enters the wrong details. After logging in successfully, the Main page appears
<u>7.</u>	Scenario Outline: New user trying to log in When the user clicks on the login button Then the login page should appear with the respective fields When the user enters the new/non existing details And clicks on submit	Login is unsuccessful and an appropriate error message appears
<u>8.</u>	Scenario Outline: Clicking on create dataset Given the user has logged in When the user clicks on the create dataset button	The dataset is created and the pop-up message 'Created dataset successfully' appears and the message also appears in the

		notification field
<b><u>9.</u></b>	<p>Scenario Outline:          Select the dataset          Given the user has logged in          When the user clicks on the browse button          Then a pop-up to choose the file appears          When the user selects the dataset file that was created in the csv format          And clicks on open</p>	<p>The dataset is selected and the file selected appears in the field</p>
<b><u>10.</u></b>	<p>Scenario Outline:          Running logistic regression          Given the user has logged in          When the user clicks on the logistic regression button</p>	<p>Logistic regression starts running for the dataset selected and it can be seen on the terminal. The various error values are calculated and the result matrix also appears on the terminal. An appropriate message appears when the logistic regression is completed successfully and it also appears in the notification field.</p>
<b><u>11.</u></b>	<p>Scenario Outline:          Running Neural Network          Given the user has logged in          When the user clicks on the neural network button</p>	<p>Neural network starts running for the dataset selected and it can be seen on the terminal. The various error values are calculated and other calculations appear on the terminal. An appropriate message appears when the neural network is</p>

		completed successfully and it also appears in the notification field.
<b><u>12.</u></b>	Scenario Outline: Entering the text to be classified Given the user has logged in When the user copies the text from the dataset and pastes it in the 'Enter the text' field And clicks on submit	The Result/Author is predicted and is displayed in the notification field.
<b><u>13.</u></b>	Scenario Outline: Clicking on logout Given the user has logged in When the user clicks on the logout button	It logs out and goes to the initial Login/Sign Up page
<b><u>14.</u></b>	Scenario Outline: Clicking on Quit When the user clicks on the Quit button	The page is Quit.

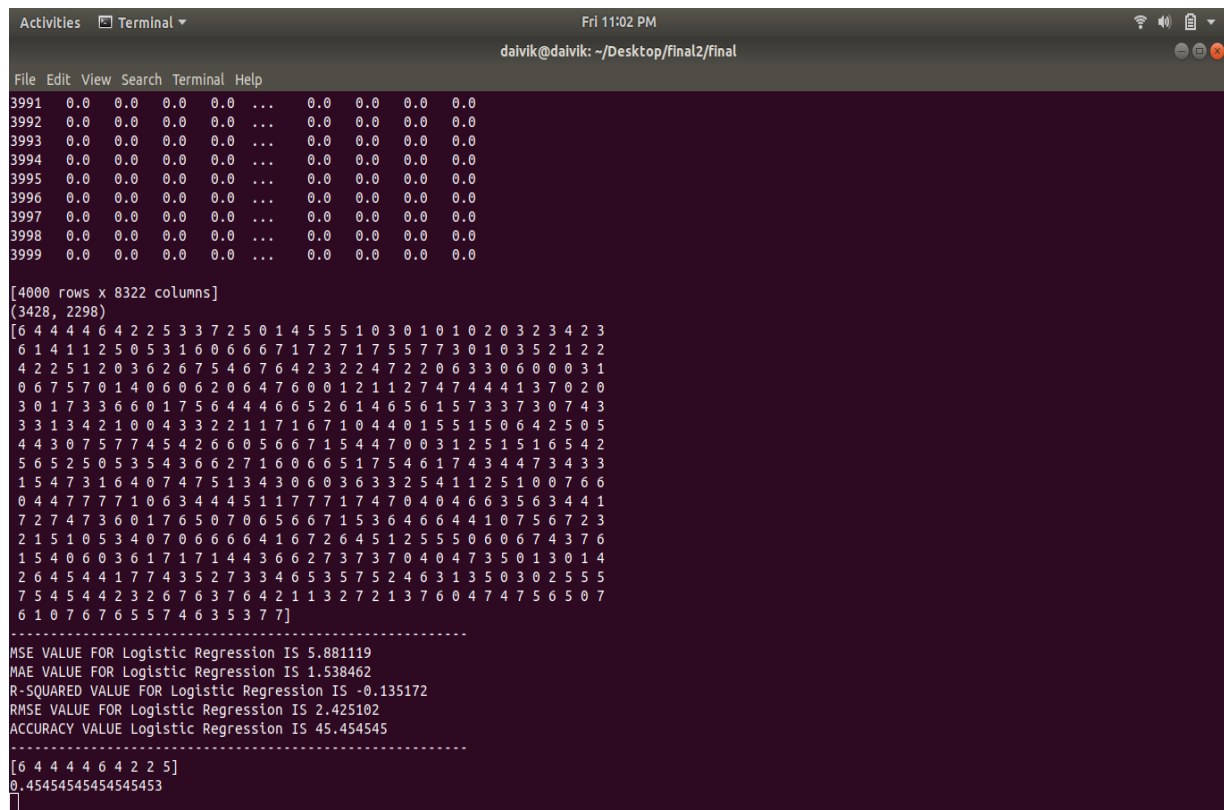
#### 9.1 Testcase list

# RESULT AND DISCUSSION

Once we start executing the program, the home page starts running, it opens the UI for further interaction with the user. For both neural network and logistic regression model MSE value, MAE value, R-squared value, RMSE value, Accuracy value is calculated.

MSE calculates the average of squared errors. MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. RMSE is a quadratic scoring rule that also measures the average magnitude of the error. Using all these metrics accuracy is calculated.

The output after performing logistic regression.



```
Activities Terminal Fri 11:02 PM
daivik@daivik: ~/Desktop/final2/final

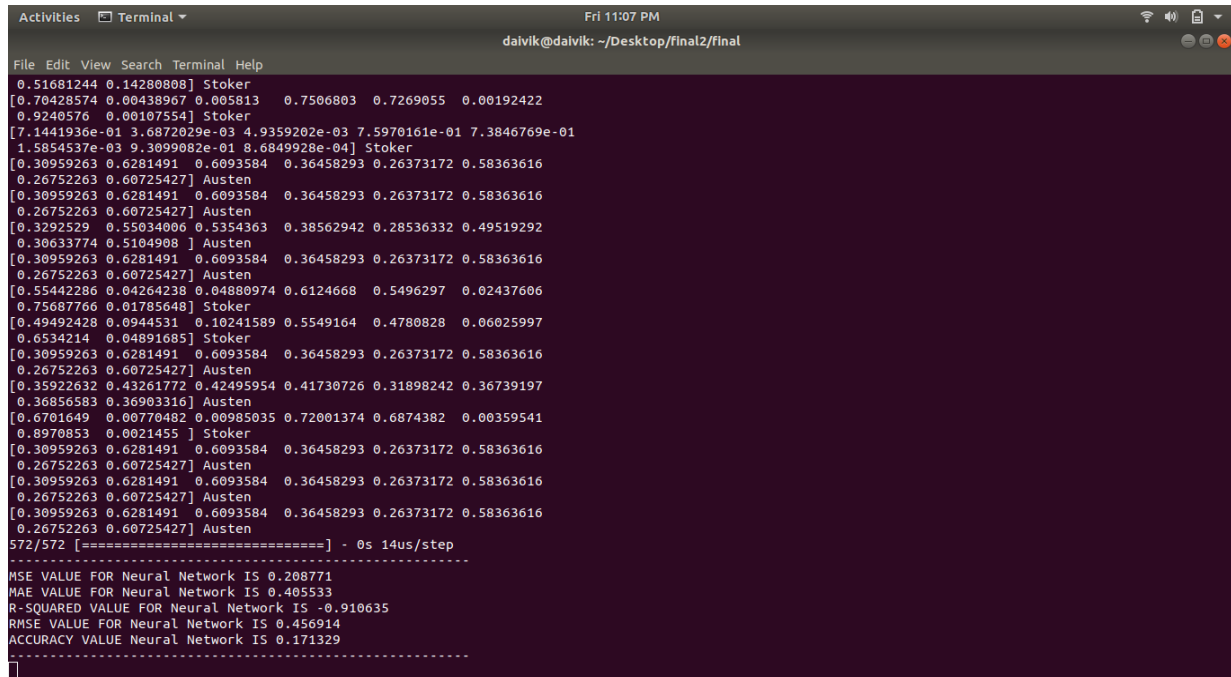
File Edit View Search Terminal Help
3991 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0
3992 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0
3993 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0
3994 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0
3995 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0
3996 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0
3997 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0
3998 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0
3999 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0

[4000 rows x 8322 columns]
(3428, 2298)
[6 4 4 4 4 6 4 2 2 5 3 3 7 2 5 0 1 4 5 5 5 1 0 3 0 1 0 1 0 2 0 3 2 3 4 2 3
6 1 4 1 1 2 5 0 5 3 1 6 0 6 6 6 7 1 7 2 7 1 7 5 5 7 7 3 0 1 0 3 5 2 1 2 2
4 2 2 5 1 2 0 3 6 2 6 7 5 4 6 7 6 4 2 3 2 2 4 7 2 2 0 6 3 3 0 6 0 0 0 3 1
0 6 7 5 7 0 1 4 0 6 0 6 2 0 6 4 7 6 0 0 1 2 1 1 2 7 4 7 4 4 4 1 3 7 0 2 0
3 0 1 7 3 3 6 6 0 1 7 5 6 4 4 4 6 6 5 2 6 1 4 6 5 6 1 5 7 3 3 7 3 0 7 4 3
3 3 1 3 4 2 1 0 0 4 3 3 2 2 1 1 7 1 6 7 1 0 4 4 0 1 5 5 1 5 0 6 4 2 5 0 5
4 4 3 0 7 5 7 7 4 5 4 2 6 6 0 5 6 6 7 1 5 4 4 7 0 0 3 1 2 5 1 5 1 6 5 4 2
5 6 5 2 5 0 5 3 5 4 3 6 6 2 7 1 6 0 6 6 5 1 7 5 4 6 1 7 4 3 4 4 7 3 4 3 3
1 5 4 7 3 1 6 4 0 7 4 7 5 1 3 4 3 0 6 0 3 6 3 3 2 5 4 1 1 2 5 1 0 0 7 6 6
0 4 4 7 7 7 7 1 0 6 3 4 4 4 5 1 1 7 7 7 1 7 4 7 0 4 0 4 6 6 3 5 6 3 4 4 1
7 2 7 4 7 3 6 0 1 7 6 5 0 7 0 6 5 6 6 7 1 5 3 6 4 6 6 4 4 1 0 7 5 6 7 2 3
2 1 5 1 0 5 3 4 0 7 0 6 6 6 6 4 1 6 7 2 6 4 5 1 2 5 5 5 0 6 0 6 7 4 3 7 6
1 5 4 0 6 0 3 6 1 7 1 7 1 4 4 3 6 6 2 7 3 7 3 7 0 4 0 4 7 3 5 0 1 3 0 1 4
2 6 4 5 4 4 1 7 7 4 3 5 2 7 3 3 4 6 5 3 5 7 5 2 4 6 3 1 3 5 0 3 0 2 5 5 5
7 5 4 5 4 4 2 3 2 6 7 6 3 7 6 4 2 1 1 3 2 7 2 1 3 7 6 0 4 7 4 7 5 6 5 0 7
6 1 0 7 6 7 6 5 5 7 4 6 3 5 3 7 7]

-----
MSE VALUE FOR Logistic Regression IS 5.881119
MAE VALUE FOR Logistic Regression IS 1.538462
R-SQUARED VALUE FOR Logistic Regression IS -0.135172
RMSE VALUE FOR Logistic Regression IS 2.425102
ACCURACY VALUE Logistic Regression IS 45.454545
-----
[6 4 4 4 4 6 4 2 2 5]
0.45454545454545453
]
```

Fig 10.1 Screenshot of Logistic Regression

The output after performing Neural Network.



```

Activities Terminal
daivik@daivik: ~/Desktop/final2/final
File Edit View Search Terminal Help
0.51681244 0.14280808] Stoker
[0.70420574 0.00438967 0.005813 0.7506803 0.7269055 0.00192422
0.9240576 0.00107554] Stoker
[7.1441938e-01 3.6872020e-03 4.9359202e-03 7.5970161e-01 7.3846769e-01
1.5854537e-03 9.3099082e-01 0.6849928e-04] Stoker
[0.30959263 0.6281491 0.6093584 0.36458293 0.26373172 0.58363616
0.26752263 0.60725427] Austen
[0.30959263 0.6281491 0.6093584 0.36458293 0.26373172 0.58363616
0.26752263 0.60725427] Austen
[0.3292529 0.55034006 0.5354363 0.38562942 0.28536332 0.49519292
0.30633774 0.5104908 ] Austen
[0.30959263 0.6281491 0.6093584 0.36458293 0.26373172 0.58363616
0.26752263 0.60725427] Austen
[0.55442286 0.04264238 0.04880974 0.6124668 0.5496297 0.02437606
0.75687766 0.01785648] Stoker
[0.49492428 0.0944531 0.10241589 0.5549164 0.4780828 0.06025997
0.6534214 0.04891685] Stoker
[0.30959263 0.6281491 0.6093584 0.36458293 0.26373172 0.58363616
0.26752263 0.60725427] Austen
[0.35922632 0.43261772 0.42495954 0.41730726 0.31898242 0.36739197
0.36856583 0.36903316] Austen
[0.6701649 0.00770482 0.00985035 0.72001374 0.6874382 0.00359541
0.8970853 0.0021455 ] Stoker
[0.30959263 0.6281491 0.6093584 0.36458293 0.26373172 0.58363616
0.26752263 0.60725427] Austen
[0.30959263 0.6281491 0.6093584 0.36458293 0.26373172 0.58363616
0.26752263 0.60725427] Austen
[0.30959263 0.6281491 0.6093584 0.36458293 0.26373172 0.58363616
0.26752263 0.60725427] Austen
572/572 [=====] - 0s 14us/step
-----
MSE VALUE FOR Neural Network IS 0.208771
MAE VALUE FOR Neural Network IS 0.405533
R-SQUARED VALUE FOR Neural Network IS -0.910635
RMSE VALUE FOR Neural Network IS 0.456914
ACCURACY VALUE Neural Network IS 0.171329
-----

```

Fig 10.2 Screenshot of Neural Network

From both the outputs we can observe that accuracy for logistic regression is comparatively more than Neural network. After training the data with any one of the model, the user can enter any sentence and the classifier will predict the other. The snapshots of the results are shared in the next chapter.



## CONFUSION MATRIX

	Montgo mery	Sto ker	Aus ten	Do yle	Tw ain	Bro nte	Alc ott	Coll ins
Montgo mery	10	2	0	5	0	1	1	0
Stoker	1	6	0	0	4	0	0	0
Austen	1	0	10	0	1	3	2	0
Doyle	1	0	0	5	5	2	4	1
Twain	0	0	1	0	4	2	0	6
Bronte	1	0	6	0	0	7	0	1
Alcott	0	0	5	0	0	0	3	0
Collins	1	3	0	1	0	1	0	5

## CHAPTER- 11

### SNAPSHOTS

The initial home page window opens once the program starts executing.

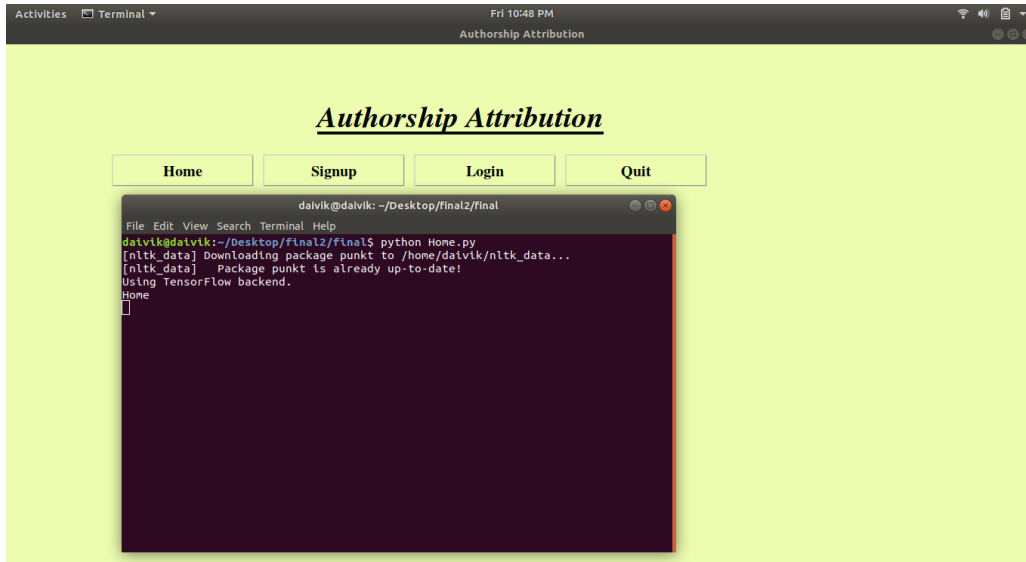


Fig 11.1 Snapshot of home page

When the user logs in and create dataset, the dataset gets generated and the user is notified.

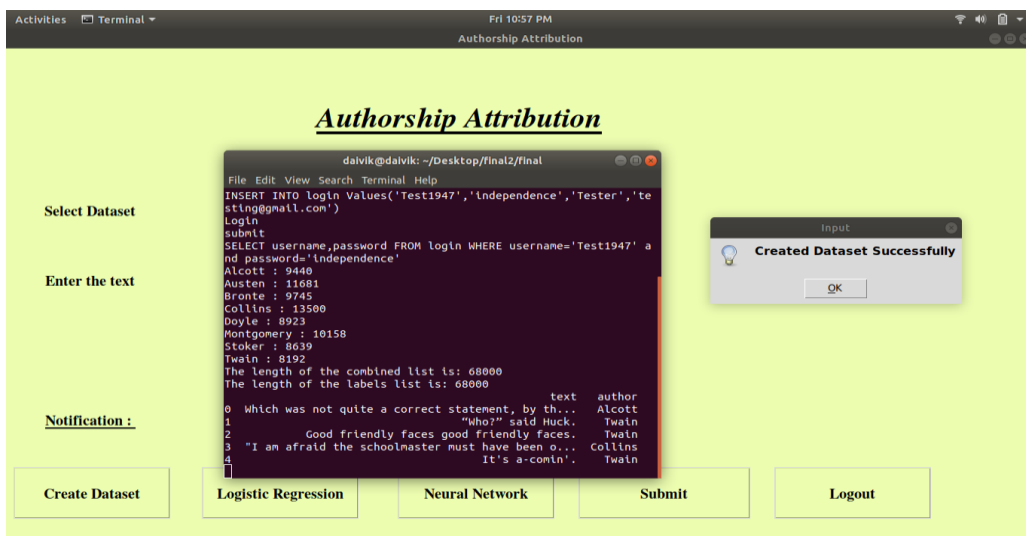


Fig 11.2 Snapshot after creating dataset

The final output predicting the author

```

Activities Terminal Fri 11:18 PM
dalvik@dalvik: ~/Desktop/final2/final

File Edit View Search Terminal Help
[0.49492428 0.0944531 0.10241589 0.5549164 0.4780828 0.06025997
0.6534214 0.04891685] Stoker
[0.30959263 0.0281491 0.0093584 0.36458293 0.26373172 0.58363616
0.26752263 0.00725427] Austen
[0.35922632 0.43261772 0.42495954 0.41730726 0.31898242 0.36739197
0.36856583 0.36903316] Austen
[0.6701649 0.00770482 0.00985035 0.72001374 0.6874382 0.00359541
0.8970853 0.0021455 ] Stoker
[0.30959263 0.0281491 0.0093584 0.36458293 0.26373172 0.58363616
0.26752263 0.00725427] Austen
[0.30959263 0.0281491 0.0093584 0.36458293 0.26373172 0.58363616
0.26752263 0.00725427] Austen
[0.30959263 0.0281491 0.0093584 0.36458293 0.26373172 0.58363616
0.26752263 0.00725427] Austen
572/572 [=====] - 0s 14us/step
-----
MSE VALUE FOR Neural Network IS 0.208771
MAE VALUE FOR Neural Network IS 0.405533
R-SQUARED VALUE FOR Neural Network IS -0.910635
RMSE VALUE FOR Neural Network IS 0.456914
ACCURACY VALUE Neural Network IS 0.171329
-----
['Which was not quite a correct statement, by the way.'
'Who?' said Huck.' 'Good friendly faces good friendly faces.' ...
'He would send a scout on ahead, and if anything made him suspicious lie snug for another week'.
'She listened attentively, with the constrained expression still on her face, and her hands still nervously clasped together in her lap.'
'Me loves evvybody,' she once said, opening her arms, with her spoon in one hand, and her mug in the other, as if eager to embrace and nourish the w
hole world.'])
[0 7 7 ... 4 3 0]
Found 8864 unique tokens.
[3] Collins
/home/dalvik/Desktop/final2/final/author_data.csv "I an afraid the schoolmaster must have been occupied with his scholars," said Miss Halcombe, "just
at the time when the woman passed through the village and returned again.

Login
submit
SELECT username,password FROM login WHERE username='Test1947' and password='Independenceeee'
dalvik@dalvik:~/Desktop/final2/final$

```

Fig 11.3 Snapshot of predicted result



Fig 11.4 Snapshot of NN and LogReg graphs

## **CHAPTER- 12**

### **CONCLUSION**

The outcome of this project is a tool, which predicts the author given a piece of text. It has a huge impact because once the model is properly trained from the dataset, further it can predict anonymous people. This project can be extended and used for plagiarism check, or to identify cyber criminals and so on. It can even be used to identify the author of some old anonymous document and the writer can be credited for his works. In this project we have used dataset of novels written by others. The model can be trained with other datasets and understand the behaviour.

## **CHAPTER- 13**

### **FURTHER ENHANCEMENT**

The project has a vast scope in the future. Implementation of the existing project can be extended to:

1. Author profiling or characterization (i.e., extracting information about the age, education, sex, etc. of the author of a given text)
2. Detection of stylistic inconsistencies (as may happen in collaborative writing)
3. Book Recommendation
4. Identifying the criminals responsible for ransom notes (to an extent)
5. Extend to many more languages

### **BIBLIOGRAPHY/REFERENCES**

- [1] [Author Attribution with CNN's, Dylan Rhodes, Conference, 2015]
- [2] [Authorship Attribution with Topic Models, Yanir Seroussi, Ingrid Zukerman, Fabian Bohnert, Conference, 2014]
- [3] [Authorship Attribution Using Principal Component Analysis And Competitive Neural Networks, Mehmet Can, Journal, 2012]
- [4] [Machine learning approach to authorship attribution of literary texts, Urszula Stańczyk, Krzysztof A. Cyran, Journal, 2007]