

**CMS Made Easy…Automated Chatbot for Students**

By

**Saad Nasir**

(BSCS/F15/0102)

**Usama Amjad**

(BSCS/F15/0116)

**Balaj Yousuf**

(BSCS/F15/0133)

**2020**

**Faculty of Engineering Sciences and Technology**

Hamdard Institute of Engineering and Technology

Hamdard University, Main Campus, Karachi, Pakistan



**CMS Made Easy…Automated Chatbot for Students**

By

**Saad Nasir**

(BSCS/F15/0102)

**Usama Amjad**

(BSCS/F15/0116)

**Balaj Yousuf**

(BSCS/F15/0133)

Under the Supervision of

**Mr. Imran Khan**

**2020**

**Faculty of Engineering Sciences and Technology**

Hamdard Institute of Engineerxing and Technology

Hamdard University, Main Campus, Karachi, Pakistan



**CMS MADE EASY…AUTOMATED CHATBOT FOR STUDENTS**

By

**Saad Nasir**

(BSCS/F15/0102)

**Usama Amjad**

(BSCS/F15/0116)

**Balaj Yousuf**

(BSCS/F15/0133)

A Project Presented to the

**Faculty of Engineering Sciences and Technology**

Hamdard Institute of Engineering and Technology

In partial fulfillment of the requirements  
For the degree

Bachelors of ScienceIn  
Computer Science

Faculty of Engineering Sciences and TechnologyHamdard Institute of Engineering and Technology



**Faculty of Engineering Sciences and Technology**

Hamdard Institute of Engineering and Technology  
Hamdard University, Karachi, Pakistan

**CERTIFICATE**

This project “**CMS MADE EASY…AUTOMATED CHATBOT FOR STUDENTS**” is presented by **SAAD NASIR, USAMA AMJAD** and **BALAJ YOUSUF** under the supervision of their project advisor and approved by the project examination committee, and acknowledged by the Hamdard Institute of Engineering and Technology, in the fulfillment of the requirements for the Bachelor degree of Computer Science.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Iqbal Uddin Khan In-charge FYP-Committee

(Project Supervisor)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Dr Adnan Ahmed Siddique Dr. Anwar Manzar

(Chairman Dept Of Computing) (DEAN FEST)

## Acknowledgement

All praises to Al Mighty” ALLAH”, the most merciful, the most gracious, the source of knowledge and wisdom end owed to mankind. All respects for Holy Prophet “**Hazrat MUHAMMAD ﷺ**, whose personality will always be source of guidance for humanity.

Acknowledge men this due to **Hamdard Institute of Engineering and Technology** for support of this Project, a highly appreciated achievement for us in the undergraduate level.

It is obliged to our Supervisor **Prof. Imran Khan** who’s rived as our major advisor. It would like to express our gratitude for their keen guidance, sincere help and friendly manner which motivated us to do well in the project and makes it a reality.

Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our project. Special thanks to **Prof. Imran Khan** & all the people for their help directly and indirectly to complete our project.

## Abstract

The abstract of our research is that there`s a lot of data available on internet which is not being used properly. People now a days prefer those applications where they can see everything in just one place. Stereotype websites are being replaced by Single Page Applications and Progressive Web Apps [1].

In social media especially Facebook, Twitter, Instagram and Snapchat are winning the race of bagging users. Digging further and according to research Twitter is the most authenticated platform where user posts correct information. 500 million tweets are sent daily. Another problem is Organizations while recruiting new employee has to do background check of the person which can take time.

In order to solve these both problems by using a data driven approach, focus on developing a prototype which uses data mining and machine learning. The strategy based on machine learning technique to automate the task.

**Keywords:** Machine Learning, Keywords Extraction, Data Mining

## Table of Contents

[Acknowledgement I](#_Toc30583859)

[Abstract II](#_Toc30583860)

[Table of Contents III](#_Toc30583861)

[List of Figures V](#_Toc30583862)

[List of Abbreviations VI](#_Toc30583863)

[Chapter 1 1](#_Toc30583864)

[1.1 Introduction 1](#_Toc30583866)

[1.2 Motivation 1](#_Toc30583867)

[1.3 Problem Statement 2](#_Toc30583868)

[1.4 Goals and Objectives 2](#_Toc30583869)

[1.4.1 Research Objectives: 2](#_Toc30583870)

[1.4.2 Academic Objectives: 2](#_Toc30583871)

[1.4.3 Commercial Objectives: 2](#_Toc30583872)

[1.5 Project Scope 3](#_Toc30583873)

[Chapter 2 4](#_Toc30583874)

[2.1 Introduction 4](#_Toc30583876)

[2.2 Background 5](#_Toc30583877)

[2.3 Materials 6](#_Toc30583878)

[2.4 Artificial Intelligence 6](#_Toc30583879)

[2.5 Machine Learning 7](#_Toc30583880)

[2.6 Neural Network 9](#_Toc30583881)

[2.7 Deep Learning 10](#_Toc30583882)

[Chapter 3 11](#_Toc30583883)

[3.1 Phases of Project 11](#_Toc30583885)

[3.2 Software/Tools that Used in Project 11](#_Toc30583886)

[3.3 Hardware that Used in Project 12](#_Toc30583887)

[3.3.1 System Requirements 12](#_Toc30583888)

[3.3.2 Environment 12](#_Toc30583889)

[Chapter 4 13](#_Toc30583890)

[4.1 Overall, AI trained model flow for Personality Predictor 13](#_Toc30583892)

[4.2 Use Case Model 14](#_Toc30583893)

[4.3 Modules for Chat bot 14](#_Toc30583894)

[Chapter 5 16](#_Toc30583895)

[5.1 Proposed System Architecture/Design 16](#_Toc30583897)

[5.2 Method and Material/Experimental Setup and Procedure 17](#_Toc30583898)

[5.2.1 Installing Anaconda on Windows Machine 17](#_Toc30583899)

[5.2.2 Regulate which Anaconda to install 17](#_Toc30583900)

[5.2.3 Installing Anaconda 18](#_Toc30583901)

[5.2.4 Determine How to Install Scikit-Learn 18](#_Toc30583902)

[5.2.5 Installing Anaconda with the Native pip 19](#_Toc30583903)

[5.2.6 Installing with Anaconda 19](#_Toc30583904)

[5.2.7 Validate Your Installation 20](#_Toc30583905)

[5.3 Setting Up NodeJS for Web Application and Server 20](#_Toc30583906)

[5.3.1 Downloading and Installing 20](#_Toc30583907)

[Chapter 6 23](#_Toc30583908)

[6.1 How Chat bot Application Works? 23](#_Toc30583909)

[6.1.1 Sign up Panel 23](#_Toc30583910)

[6.1.2 Login Panel 23](#_Toc30583911)

[6.1.3 Admin Dashboard 24](#_Toc30583912)

[6.1.4 User Dashboard 24](#_Toc30583913)

[Chapter 7 25](#_Toc30583914)

[7.1 Conclusion 25](#_Toc30583915)

[7.2 Future Work 25](#_Toc30583916)

[REFERENCES 26](#_Toc30583917)

[APPENDICES 28](#_Toc30583918)

[Appendix A: main.py 28](#_Toc30583919)

[Appendix B: homepage.component.ts 30](#_Toc30583920)

[Appendix C: homepage.component.html 33](#_Toc30583921)

[Appendix D: admindashboad.Component.html 34](#_Toc30583922)

[Appendix E: admindashboard.component.ts 35](#_Toc30583923)

[Appendix F: approtuing.module.ts 36](#_Toc30583924)

[Appendix G: app.component.css 38](#_Toc30583925)

[Appendix H: app.component.html 41](#_Toc30583926)

[Appendix I: app.component.ts 43](#_Toc30583927)

[Appendix J: app.module.ts 44](#_Toc30583928)

[Appendix K: twitterAuth.module.ts 45](#_Toc30583929)

[Appendix L: index.html 47](#_Toc30583930)

[Appendix M: GetTweets.py 48](#_Toc30583931)

[Appendix N: MergeAndProcessTweets.py 50](#_Toc30583932)

[Appendix O: TweetsLabelling.py 52](#_Toc30583933)

[Appendix P: NaiveBayes.py 53](#_Toc30583934)

## List of Figures

**Figure No Description Page No.**

[Figure 1.5.1 Project Scope 3](#_Toc30583804)

[Figure 2.4.1 Artificial Intelligence 7](#_Toc30583805)

[Figure 2.5.1 Machine Learning 8](#_Toc30583806)

[Figure 2.6.1 Neural Network 9](#_Toc30583807)

[Figure 4.1.1 System flow for Chat bot 13](#_Toc30583808)

[Figure 4.2.1 Use Case Model 14](#_Toc30583809)

[Figure 5.1.1 Strategy of a System 16](#_Toc30583810)

[Figure 5.2.1 Download Anaconda 17](#_Toc30583811)

[Figure 5.2.2 Installing Anaconda 18](#_Toc30583812)

[Figure 5.2.3 Python 3.6 19](#_Toc30583813)

[Figure 5.2.4 Anaconda IDE 19](#_Toc30583814)

[Figure 5.3.1 NodeJS 20](#_Toc30583815)

[Figure 5.3.2 NodeJS Installation Process 21](#_Toc30583816)

[Figure 5.3.3 NodeJS version 22](#_Toc30583817)

[Figure 6.1.1 Signup Panel 23](#_Toc30583818)

[Figure 6.1.2 Login Panel 23](#_Toc30583819)

[Figure 6.1.3 Admin Dashboard 24](#_Toc30583820)

[Figure 6.1.4 User Dashboard 24](#_Toc30583821)

## List of Abbreviations

**AI**: Artificial Intelligence

**API**: Application Programming Interface

**DL**: Deep Learning

**HCI**: Human-computer Interaction

**IDF**: Inverse Document Frequency

**ML**: Machine Learning

**NLP**: Natural Language Processing

**TF**: Term Frequency

**UI**: User Interface

**UX**: User Experience

# Chapter 1

**Introduction to Chat bot**



## Introduction

One of the most important roadblocks in advancing technology is data is not being properly utilized. Data generated is in billions but is not being used properly. Every day users using social media is increasing. There are currently 2 billion active social media users. Another real-world problem of this era is that when users applies for a job in a multinational company, his whole background test is conducted. This leads to a lot of time and energy. In this fast-paced world, we cannot be complicit. We need to keep up with the technology. To solve this problem some of the advances has been made in to use machine learning using user data to predict the personality of the user. Normally

* Logistic Regression
* Naïve Bayes
* TF – IDF
* Decision Trees
* Random Forest
* Feed Forward Neural Network

Training algorithms have been used to train the model but they were limited to just research work. We plan to use this in industry to help organizations predict whether the person is suitable for the job or not. We will try to predict the traits of the user using his past tweets. To achieve efficiency, we will be using machine learning techniques with cognitive learning to improve accuracy. Machine Learning has come a long way when it was first introduced. Machine Learning is being used extensively in image recognition and text classification. Several Machine Learning algorithms have been developed to make technology more reliable and flexible to use. Machine Learning has changed the advancement of computers.

## Motivation

From my personal experience, navigating between CMS takes a lot of time. For e.g. just to see my current schedule or transport or tuition fee we have to navigate back and forth on average 4 times. In the age where 5G is evolving the world, this is a no brainer. Chatbots are one of many solutions that can solve this problem. You just ask chatbot the question, it will provide with you the relevant answers.

## Problem Statement

In this technologically advanced world, navigating between pages or to perform queries in every other page wastes a lot of time. People now a days prefer those applications where they can see everything in just one place. Stereotype websites are being replaced by Single Page Applications and Progressive Web Apps.

## Goals and Objectives

The objective is to develop a generalized algorithm that is able to detect what user wants to know with the help of Artificial Intelligence and provide appropriate answer. The Aim is to develop a model that can be integrated with our own university CMS and help students serve better way.

### Research Objectives:

* To research, it can provide extension of Machine Learning and natural language processing from social media to in web-applications.
* To develop the project, this requires extensive amount of research on human behavior, human thinking.

### Academic Objectives:

* To write a university-based chapter on machine learning in chatbots
* To identify the power of chatbots and how it can be used to speed up work

### Commercial Objectives:

* After the development of the project, it can be commercialized mainly in educational organizations where a lot of money is spent on creating huge CMS and to make it speed up which can be done easily by Chat bot.
* To implement the project in big educational institutes

## Project Scope



Figure 1.5.1 Project Scope

It centers on the ability to help user create an interactive way to use CMS. Chatbot will provide a unique way of seeing data and visualizing with CMS. It will make user fun to open CMS.

We will train our machine learning models until they are fitting enough to be used on application. Our project will mainly hub on developing a machine learning model and training it to accomplish high level of accuracy so it can be used for other research-oriented projects and can be further specified according to the need of the organization.

Our basic endeavor is to train model using Generative approach and generative models on the seq2seq neural network. This network was initially released for machine translation, but has also proved to be quite effective when it comes to building generative Chat bots. Generative Chatbots also require a very large amount of conversational data to train. We will first train our seq2seq implementation for our chatbot using more 2 million conversations. also require a very large amount of conversational data to train. If we think that Generative approach is not working according to our need, we will switch to Retrieval Based bots. Retrieval based bots work on the principle of directed flows or graphs. The bot is trained to rank the best response from a finite set of predefined responses.

# Chapter 2

**Literature Review**



## Introduction

Total population of world is about 7.3 billion and internet currently has 4.8 billion users. There are 2.7 billion active social media users. Twitter currently has around 500 million users. That`s not just huge numbers but large amount of data as well. Web-based social networking is where handlers present themselves to the world, uncovering individual subtleties and bits of knowledge into their lives [1].

Web based life use has developed since a decade ago. We are beginning to grasp how a portion of this data can be used to improve the clients' encounters with interfaces and with each other. Web based life locales are currently the most well-known goal for Internet clients, giving social researchers an incredible chance to comprehend online conduct [2].

When combined social media with Artificial Intelligence it harnesses so much power i.e. user data which could be used to learn more about the user. More and more people join social media every day which is directly proportional to data being increasing every day on social media. There is a great opportunity to gather social media data of user and use it for professional purpose.

There is a rising amount of research papers related to social mass media, a small number of which focus on behavior prediction. Easily accessible digital records of behavior. More and more work is being done on data obtained using social media to find meaningful information about a person. That data involves a lot of cleansing first so that it can be used further in research work. Most of the research is done using these five features which predict the personality of the user. Those five features are

* Openness
* Conscientiousness
* Extraversion
* Agreeableness
* Neuroticism [4]

Mostly Big five personality traits are used to judge the personality of an individual, furthermore data mining and other techniques are used to do sentimental analysis and natural language processing is done to automate the summary of an abstract. Other set of features are defined on which attributes of a person is judged. Research suggests that those who use social media regularly are more social minded and are active in civic life [6]. Wide variety of people`s personal attributes, ranging from intelligence to sexual orientation can be automatically and accurately predicted using Twitter tweets and other widespread digital records [7].

Twitter tweets are an example where user shares his idea which leads us to find what type of intellectual user is. We then show a way of precisely forecasting a user's nature simply based on three counts openly available on profiles: following, followers, and listed counts [8]. Twitter tweets are commonly used to predict personality because Twitter tweets are publicly viewable and can be easily be collected using Twitter API`s. Tweets speak a lot about personality of a person, what user thinks about certain topics what are his/her views. These could be used to judge the intelligence of a person.

The data obtained using social media is of very significance. There are many research papers on to judge a personality of a person using social media but there is still space for a lot of work to be done specially on to judge the attributes of a person. By attributes here we mean that suppose a person drinks or not and how regularly he drinks. This information could be useful for car insurance company. Various research projects have been done previously to define attributes of a user using social media. There are still many opportunities to further do more data analysis of social media records which can lead to learn attributes of a person [9].

Natural Language Processing (NLP) is being all over the world to gain insights from the text. It is also used to extract meaning from the text. NLP is used in Automatic Summarization of legal Text. It is also used to find context from the text. Some of the projects in which it has been used are “Sentimental Analysis for twitter, web articles”, “Movie Review Prediction”, “Summarize Restaurant Reviews”, “AutoBot - Build a system that can have a conversation with you.”. These are few examples in which NLP is used.

## Background

To just see your attendance on average you need to route to 3 pages for course specific attendance and generally the stack on which CMS are made are outdated and really slow. It takes approx. 80 seconds to see your attendance. In order to solve this problem by using a data driven approach. Focus on developing a prototype which can classify the label using his users text. The strategy based on machine learning technique and achieves very high efficiency and accuracy to predict what user is querying.

## Materials

The project Datasets has been designed by ourselves. We had created at least 250000 publicly available user random queries. First, we manually selected a label and its corresponding texts from internet which suits our project having very large number of classes. We followed this approach and repeated this approach for almost 100000 texts and then we merged them to create random texts more.

## Artificial Intelligence

For the most part individuals consider Artificial Intelligence (AI) as a kind of fake mind and review pictures from sci-fi films about robots. Such pictures have almost no to do with how AI is really utilized in this day and age. The facts confirm that AI has numerous likenesses to human cerebrum work, however the significant inconsistency is that Artificial Intelligence is non-regular. Before we get excessively profound, I might want to present some exceptionally broad ideas about how you associate with an AI calculation. The AI calculation is the method that you are utilizing to tackle an issue. An AI calculation is some of the time called a model.

The most well-known are Neural Networks, Support Vector Machines, Bayesian systems, and Hidden Markov Models. This arrangement of books covers a significant number of these models. It is significant for the AI professional see how to speak to an issue to an AI program, as this is the essential method of connection with an AI calculation. We will start our establishment of information in this point by investigating how the human cerebrum communicates with its reality.

Man-made brainpower (AI) is an approach to make machine think and carry on brilliantly as being human. Simulated intelligence has a great deal to do with wise programming program that control these savvy machines. Simulated intelligence is firmly identified with the investigation of human mind. Scientist accept that AI can be cultivated by seeing how the human cerebrum functions. Man-made intelligence has capacity to impact each part of our lives. Computer based intelligence is generally in its earliest stages. In the course of the most recent few decades, AI has created some fabulous items, for example, self-driving vehicles and knowledge robots that can walk.

The inspiration driving AI is to empower a PC to work genuinely like a human cerebrum. Regardless, this doesn't suggest that AI attempts to mimic each piece of the human cerebrum. How much an AI computation organizes the genuine working of the human personality is called natural trustworthiness. With respect to AI, the brain is essentially and moved piece of advancement that we ought to look at, make sense of, and make sense of how to mirror. The brain is not the lone portion of “advanced technology” that nature has common with us. Flight Is another. Early airplanes attempted to emulate the flapping wings of birds. Such airships are called ornithopters. [10]

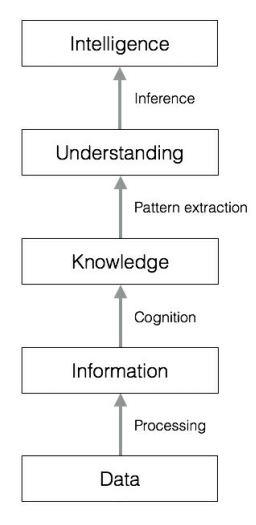


Figure 2.4.1 Artificial Intelligence

## Machine Learning

Presently, one asset is Data. It is in enormous amount: a huge amount of organized and unstructured information. In the cutting-edge century, Machine Learning advanced as a sub-field of Artificial Intelligence (AI) that included self-learning calculations. These calculations are of coming about data from information so as to make expectations. Past strategy expected people to physically begin rules for model and work from examination of immense amounts of data, AI offers an all the more efficient option for catching the information in information to improve the demonstration of prescient investigation models and have information driven choices.

* AI is getting progressively vital in the software engineering research, yet it likewise assumes a much more prominent job in our regular day to day existences. We appreciate: vigorous email spam filters,
* opportune text and voice gratitude software,
* consistent web search engines,
* challenging chess-playing programs,
* In addition, confidently rapidly, harmless and effectual self-driving cars.

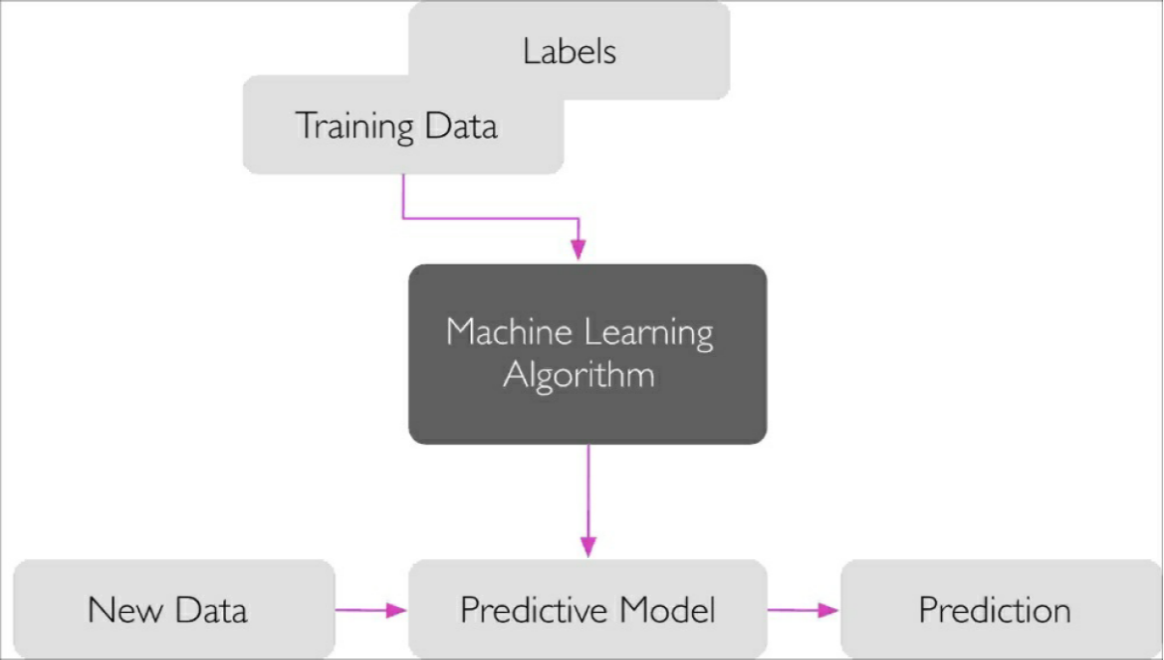


Figure 2.5.1 Machine Learning

There are tri kinds of machine learning: **supervised learning**, **unsupervised learning**, and **reinforcement learning**.[11] The practical delinquent areas where these can be functional:

* **Supervised Learning:**
  + Labeled Data
  + Direct Feedback
  + Predict outcome/Future
* **Unsupervised Learning:**
  + No labels
  + No feedback
  + Find hidden structure in data
* **Reinforcement Learning:**
  + Decision Process
  + Reward System
  + Learn series of actions

## Neural Network

Neural Network is the basic premises of Artificial Intelligence is to manufacture machine that can perform errands that require human knowledge. Man-made consciousness are planned with the end goal that they can distinguish the hidden examples in information and gain from that. [12]

They can be utilized for different assignments, for example, arrangement, relapse, division, etc. To reproduce the learning procedure of human cerebrum, a fake neural system is to construct utilizing layers of neurons. These neurons propelled by natural neurons. Each layer in a counterfeit neural system is a lot of autonomous neurons.

Every neuron in a layer associated with neurons in the nearby layer. Neural systems self-learning calculation. The main case of a neural system called the perceptron, which developed by Frank Rosenblatt in 1957. The perceptron is a system contained just an information and a yield layer. If there should arise an occurrence of parallel arrangements, the yield layer has just a single neuron or unit. The perceptron appeared to be exceptionally encouraging from the beginning, however it was immediately understood that it could just adapt straightly divisible patterns.[13] In its most essential portrayals, perceptron are simply basic portrayals of one neuron and its info, input that can be included a few neurons.[14]



Figure 2.6.1 Neural Network

## Deep Learning

**Deep Learning** is a subset of AI. Profound learning, uses a positioned degree of neural systems to transmit out the advancement of AI.

The fake neural system is constructed like a human cerebrum, with neuron hubs associated together. The passage presented a straightforward case of a neural system, a feed-forward 1-layer arrange. They are called feed-forward on the grounds that the data continues from the contribution towards the yield and it never circles back, and 1-layer in light of the fact that there is just 1-yield layer other than the info layer. This isn't the general case.

The impediments of 1-layer feed-forward systems are as of now examined when they were referenced that they can just take a shot at directly detachable information. There are, notwithstanding, systems that have additional layers between the information and the yield layers, these layers are known as the shrouded layers.

A feed-forward system with concealed layers will at that point move the data from the contribution through its shrouded layer to the yield layer, which characterizes a capacity that takes an info and it characterizes a yield. There exists a hypothesis, called the Universal Theorem, which expresses that any capacity can be approximated by a neural system with in any event one concealed layer for quite a while, given this hypothesis and furthermore the trouble in working with complex systems, individuals have worked with shallow systems with just one covered up layer.[15]

Be that as it may, as of late individuals have understood that increasingly composite systems with numerous imperceptible layers can comprehend level of reflection that slight layers can't. Furthermore, intermittent systems have been presented where neurons can likewise encourage data once more into themselves. Some neural systems' structures can likewise allow to characterize a vitality work that takes into consideration the formation of recollections.

# Chapter 3

**Project Discussion**



## Phases of Project

Following are the stages of Development.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Phase** | **Duration** | **Action Performed** | **Responsible Person** |
| 1 | Requirement Gathering | 42 Days | Non-Functional  &  Functional Requirements | Osama  Armaan  Balaj |
| 2 | Requirement Analysis | 26 Days | Functional  &  System Analysis | Osama  Armaan  Balaj |
| 3 | Designing | 38 Days | User Interface and System Flow | Osama  Armaan  Balaj |
| 4 | Development | 58 Days | Front-End  &  Backend Development | Osama  Armaan  Balaj |
| 5 | Testing | 20 Days | Black-Box  &  White-Box Testing | Osama  Armaan  Balaj |
| 6 | Documentation | 30 Days | FYP Report | Osama  Armaan  Balaj |
| Overall Project | | 1. Days |  | |

## Software/Tools that Used in Project

1. Python, Flask – Programming language.
2. HTML, CSS3, Ionic – For Web-Development.
3. MongoDB – NoSQL Database
4. Anaconda – for Large-Scale data processing and scientific computing.
5. Keras , Scikit– Open Source library for machine Learning [28].

## Hardware that Used in Project

* Intel i7th-6200 (Client – Machine).

### System Requirements

* Functioning System: MS Windows 10 or MAC OS Sierra 10x or Linux Ubuntu 18.0.4.
* CPU Type: Intel® Core (TM) i7-7500U CPU @2.60GHz 2.80GHz
* Memory: 8GB/ Disk Space: 250GB SSD

### Environment

As the web app is Internet based so it will run on any operating system with internet access having the following browsers:

* Google Chrome version 41+.
* Firefox version 40+.
* Internet explorer version 10+.

# Chapter 4

**System Design**

This section is about the strategy of system and figures with their notations that represent the system. It discusses about the different types of diagrams that are used in the system development process among which some of we have used in this system.



## Overall, AI trained model flow for Personality Predictor

This is the diagram for our over project and their features and how they work in the application.

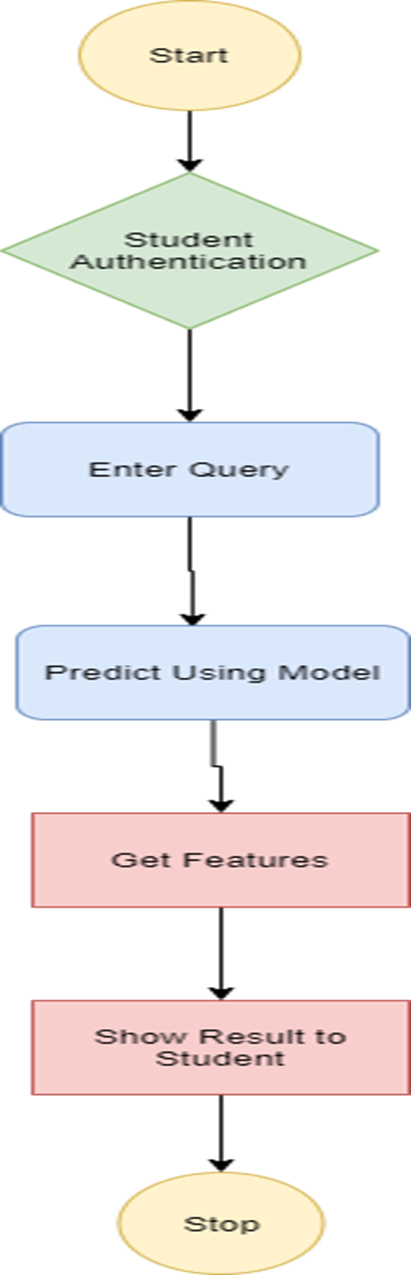


Figure 4.1.1 System flow for Chat bot

## Use Case Model

A use case is a set of exploit or event steps, the interactions between a function (known as an actor in the Unified Modeling Language) and a scheme that are characteristically necessary to achieve a goal. The actor may be a human being, an outside machine, or time.

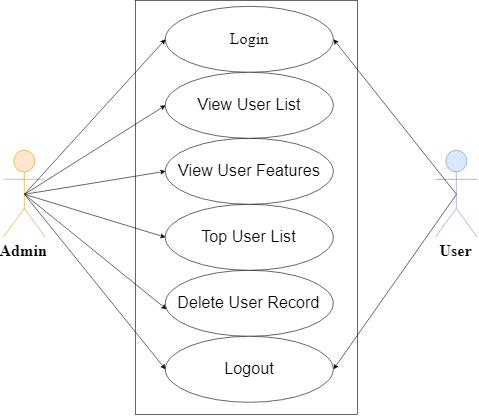


Figure 4.2.1 Use Case Model

## Modules for Chat bot

There are four Modules in Person Recognition:

* Login Module
* User Module

**Login Module**

This is the initial module user can login. They will be able to login using their personal/provided email and password to the web application.

**User Module**

After successfully logging in users will be to chat with the bot and ask their queries, chatbot will pass the query to the model and return the required output as a reply.

# Chapter 5

**Implementation**



## Proposed System Architecture/Design

Chatbot is a mobile application, which contains the accompanying layers: UI layer, utilitarian assistance layer and equipment layer. It shows by and large plan of a framework.

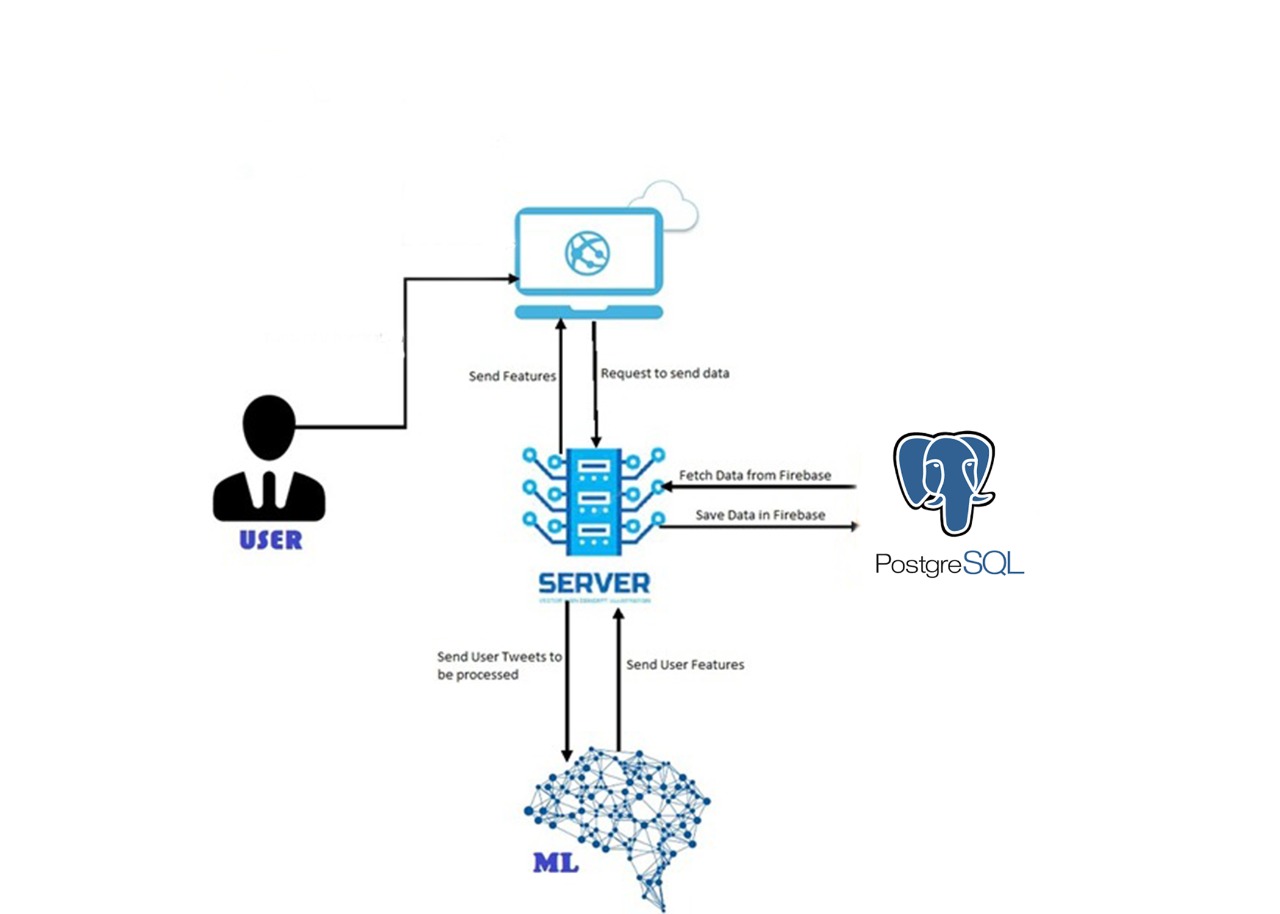


Figure 5.1.1 Strategy of a System

The essential functionality of Chat bot is to extract the features of the user. Chat bot is based on Artificial Intelligence applied with Cognitive Learning to attain maximum correct results.

## Method and Material/Experimental Setup and Procedure

We will need Scikit Scipy, along with other libraries to make the AI model work properly. Follow the steps for installing Anaconda and other libraries.

### Installing Anaconda on Windows Machine

This controller elucidates in what way to install Anaconda on Windows. Though these directions power also run on further Windows deviations, we have only verified these commands on tackles assembling the subsequent necessities:

* 64-bit, x86 desktop and or laptop
* Windows 10

### Regulate which Anaconda to install

You should pick one of the accompanying kinds of Anaconda to introduce:

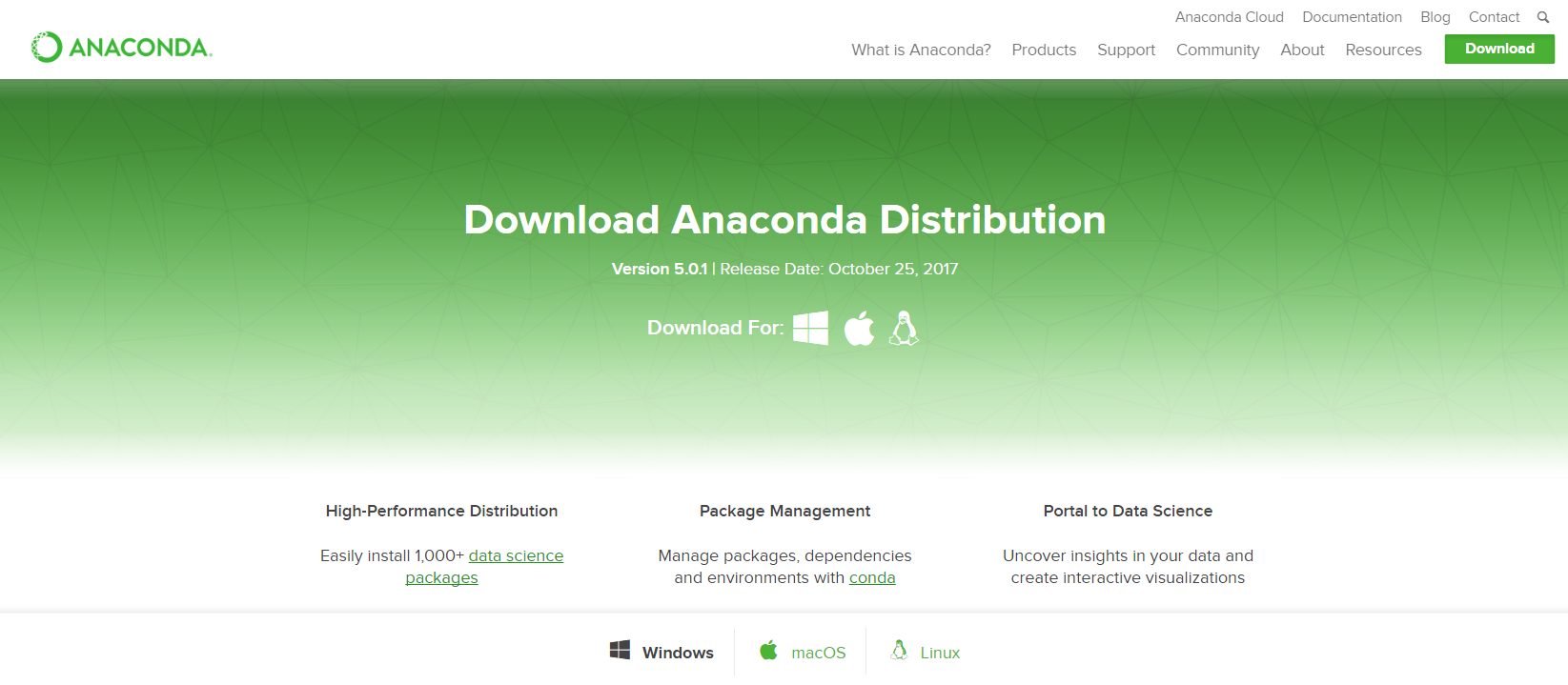


Figure 5.2.1 Download Anaconda

### Installing Anaconda

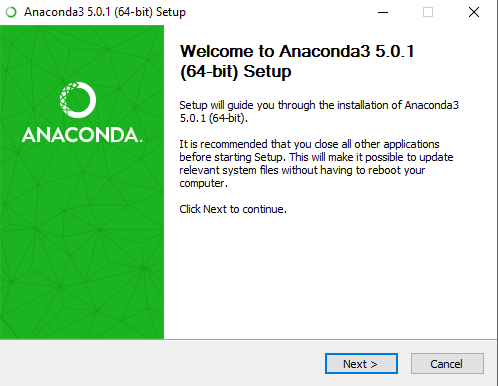


Figure 5.2.2 Installing Anaconda

### Determine How to Install Scikit-Learn

You must pick the mechanism by which you install Scikit. The supported choices are as follows:

* "native" pip
* Anaconda

Native pip introduces Scikit straight on your machine without running a virtual domain. In the interim a local pip establishment isn't made in a disengaged vessel; the pip establishment may meddle through additional Python-put together establishments with respect to your machine. In any case, in the event that you perceive pip and your Python condition, a local pip establishment much of the time includes just a single order! Moreover, on the off chance that you introduce with local pip, handlers can follow Scikit records from any chronological registry on the framework.

### Installing Anaconda with the Native pip

If unique of the ensuing variants of Python isn't introduced on your system, install it here and now:

* Python 3.5 v64-bit from <https://www.python.org>



Figure 5.2.3 Python 3.6

Scikit supports Python3 on Windows. To download Scikit, start a terminal. Then issue the suitable pip3 install knack in that command prompt. To download the CPU-only type of Scikit, enter the subsequent knack:

E:Users\> **pip3 install scikit**

To download the GPU kind of Scikit, enter the subsequent command:

E:Users\> **pip3 install scikit**

### Installing with Anaconda



Figure 5.2.4 Anaconda IDE

1. Subject the suitable command to install Scikit inside your conda environment. To install the CPU-only version of Scikit, enter the following command:

(risk) C:> **pip install --ignore-installed --upgrade scikit**

To install the GPU version of Scikit, enter the following command (on a single line):

(risk) C:> **pip install --ignore-installed --upgrade scikit**

### Validate Your Installation

Jump a command prompt. If you connected through Anaconda, activate your Anaconda environment. Summon python from your shell as shadows:

$ **python**

Enter the accompanying short program inside the python intuitive shell:

C:\\Users\\> import scikit-learn as scikit

If the terminal does not give slightly error and cursor blinks to next line it means scikit has been installed

## Setting Up NodeJS for Web Application and Server

### Downloading and Installing

* **Download the Windows installer from the Nodes.js® web site.**



Figure 5.3.1 NodeJS

* **Run the installer** (you downloaded the.msi file in the previous step)
* **Follow the installer prompts** (accept the license agreement, press the NEXT button a few times and approve the default installation settings).
* **Boot your Windows**. Until you restart your computer, you won’t be able to run Node.js ®.

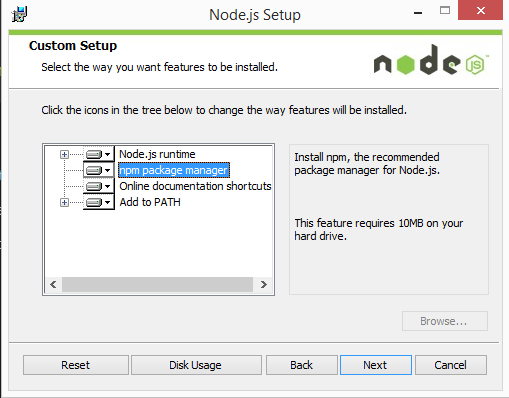


Figure 5.3.2 NodeJS Installation Process

Ensure you have Node and NPM introduced by running basic directions to perceive what form of each is introduced and to run a basic test program:

**Test Node.** Test Node. To check whether Node is presented, open the Windows Command Prompt, PowerShell or a relative bearing line gadget, and type hub - v. This ought to frind a variation number, so you'll see something like this v0.10.35.

**Test NPM.** To check whether NPM is presented, type npm - v in Terminal. This ought to frind NPM's variation number so you'll see something like this 10.16.2

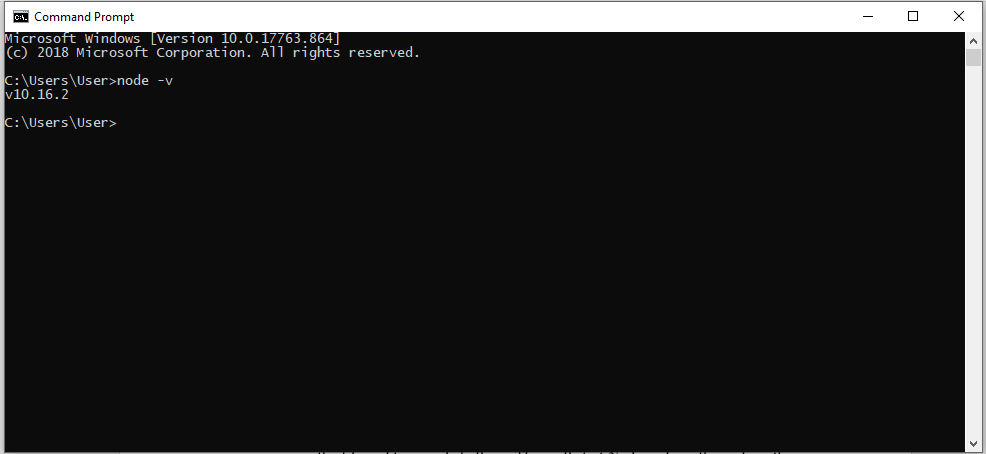


Figure 5.3.3 NodeJS version

* **Create a test file and run it.** A fundamental strategy to test that node.js works is to make a JavaScript record: name it hello.js, and basically incorporate the code console.log ('Node is installed!'); To run the code simply open your request line program, investigate to the coordinator where you save the archive and type center point hello.js. This will start Node and run the code in the hello.js record. You should see the yield Node is presented.

# Chapter 6

**Chatbot**

**6**

## How Chatbot Application Works?

In order to use Chatbot Application, you must follow these Steps:

### Login Panel

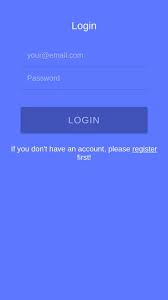


Figure 6.1.2 Login Panel

### User Dashboard



Figure 6.1.4 User Dashboard

# Chapter 7

**Conclusion and Future Work**

## Conclusion

The mobile application works smoothly but due to time constraint it has some limited features. We tend to add more features like complain/suggestion area. However, the model of this application is ideal for small to medium-sized android mobiles; it can be further improved when deploying on Cloud Provider

## Future Work

In the future the following features in the project will make it project more effective.

1. Try to get other labels data as well to enhance our product’s efficiency.
2. Try to utilize more advanced deep learning algorithms to get better efficiency.
3. Implement the system in educational institutes.

# REFERENCES

1. Correa, Teresa, Amber Willard Hinsley, and Homero Gil De Zuniga. "Who interacts on the Web?: The intersection of users’ personality and social media use." Computers in Human HCHAT BOT with social media." In CHI'11 extended abstracts on human factors in computing systems, pp. 253-262. ACM, 2011.
2. Hu, Tianran, Haoyuan Xiao, Jiebo Luo, and Thuy-vy Thi Nguyen. "What the Language You Tweet Says About Your Occupation." In ICWSM, pp. 181-190. 2016.
3. Marengo, Davide, Fabrizia Giannotta, and Michele Settanni. "Assessing personality using emoji: An exploratory study." Personality and Individual Differences 112 (2017): 74-78.
4. Li, Weijian, Yuxiao Chen, Tianran Hu, and Jiebo Luo. "Mining the Relationship between Emoji Usage Patterns and Personality." arXiv prefrind arXiv:1804.05143 (2018).
5. Golbeck, Jennifer, Cristina Robles, Michon Edmondson, and Karen Turner. "CHAT BOT from twitter." In Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, pp. 149-156. IEEE, 2011.
6. Amichai-Hamburger, Yair, and Gideon Vinitzky. "Social network use and personality." Computers in human behavior26, no. 6 (2010): 1289-1295.
7. Wan, Danlin, Chuang Zhang, Ming Wu, and Zhixiang An. "Personality Prediction Based on All Characters of User Social Media Information." In Chinese National Conference on Social Media Processing, pp. 220-230. Springer, Berlin, Heidelberg, 2014.
8. Andreassen, Cecilie Schou, Ståle Pallesen, and Mark D. Griffiths. "The relationship between addictive use of social media, narcissism, and self-esteem: Findings from a large national survey." Addictive Behaviors 64 (2017): 287-293.
9. **Azucar, Danny, Davide Marengo, and Michele Settanni. "Predicting the Big 5 personality** traits from digital footfrinds on social media: A meta-analysis." Personality and Individual Differences 124 (2018): 150-159.
10. Gosling, Samuel D., Peter J. Rentfrow, and William B. Swann Jr. "A very brief measure of the Big-Five personality domains." Journal of Research in personality 37, no. 6 (2003): 504-528.
11. Kosinski, Michal, David Stillwell, and Thore Graepel. "Private traits and attributes are predictable from digital records of human behavior." Proceedings of the National Academy of Sciences 110, no. 15 (2013): 5802-5805.
12. Quercia, Daniele, Michal Kosinski, David Stillwell, and Jon Crowcroft. "Our twitter profiles, our selves: CHAT BOT with twitter." In Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, pp. 180-185. IEEE, 2011
13. Schoen, Harald, Daniel Gayo-Avello, Panagiotis Takis Metaxas, Eni Mustafaraj, Markus Strohmaier, and Peter Gloor. "The power of prediction with social media." Internet Research23, no. 5 (2013): 528-543.
14. Sumner, Chris, Alison Byers, Rachel Boochever, and Gregory J. Park. "Predicting dark triad personality traits from twitter usage and a linguistic analysis of tweets." In Machine learning and applications (icmla), 2012 11th international conference on, vol. 2, pp. 386-393. IEEE, 2012.

# APPENDICES

## Appendix A: main.py

import tweepy

import pandas as pd

from credentials import apis

from flask import jsonify,Response

import os

import firebase\_admin

import google.cloud

from firebase\_admin import credentials, firestore

from flask\_cors import CORS

import json

cred = credentials.Certificate("./firebaseCredentials.json")

FIRESTORE = firebase\_admin.initialize\_app(cred)

db = firestore.client()

from flask import Flask

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

CORS(app)

@app.route('/getTweets/<username>')

def getTweets(username):

try:

#applied list comprehension

tweets = [user.text for user in tweepy.Cursor(apis.user\_timeline, screen\_name=username, include\_rts=False).items(1000)]

if len(tweets) >= 30:

df = pd.DataFrame(columns=['Tweets','D','P','R','S','V'])

df['Tweets'] = tweets

files = os.listdir('./keywords')

json = [0,1,2,3,4]

for index,file in enumerate(files):

json[index] = pd.read\_json('./keywords/' + file, typ='series')

for value,file in enumerate(json):

for files in zip(file.keys(),file.values):

for index,tweets in enumerate(df.Tweets):

if files[0].lower() in tweets.lower(): df.iloc[index, [value+1]] = files[1]

labels = df.iloc[:,1:6].sum()

total = labels.sum()

final\_value = {}

for value in zip(labels.keys(),labels.values):

final\_value[value[0]] = (value[1]/total) \* 100

doc\_ref = db.collection(u'TwitterData').document(username.lower())

doc\_ref.set({

u'D': final\_value['D'],

u'P': final\_value['P'],

u'R': final\_value['R'],

u'S': final\_value['S'],

u'V': final\_value['V']

})

return jsonify(final\_value)

else:

return Response("Not much data to process")

except:

return Response("No Internet Connection or server down")

@app.route('/getData')

def getData():

docs = db.collection(u'TwitterData').get()

documents = []

for doc in docs:

documents.append(doc.to\_dict())

return json.dumps(documents)

@app.route('/getUserData/<username>')

def getUserData(username):

doc\_ref = db.collection(u'TwitterData').document(username.lower())

doc = doc\_ref.get()

return doc

## Appendix B: homepage.component.ts

<div class="app-header">

</div>

<!-- Application header ends -->

<mat-grid-list cols="3">

<mat-grid-tile></mat-grid-tile>

<!-- Writing markup for login starts -->

<mat-grid-tile class='login-element'>

<div class="login-container">

<!-- Template Body starts -->

<div class="login-box-body">

<form autocomplete="on" [formGroup]="myForm" (ngSubmit)="onSubmit()">

<!-- Markup for getting Username starts -->

<mat-grid-list cols="9" rowHeight="75px">

<mat-grid-tile [rowspan]="1" class="login-icon">

<mat-icon>lock</mat-icon>

</mat-grid-tile>

<mat-grid-tile [colspan]="7">

<mat-form-field class="justifier">

<input formControlName="username" matInput placeholder="Username" text="text">

</mat-form-field>

</mat-grid-tile>

</mat-grid-list>

<!-- Markup for getting Username ends -->

<!-- Markup for getting Password starts -->

<mat-grid-list cols="9" rowHeight="100px">

<mat-grid-tile [rowspan]="1" class="login-icon">

<mat-icon>remove\_red\_eye</mat-icon>

</mat-grid-tile>

<mat-grid-tile [colspan]="7">

<mat-form-field class="justifier">

<input formControlName="password" matInput placeholder="Enter your password" [type]="hide ? 'password' : 'text'">

<button mat-icon-button matSuffix (click)="hide = !hide" [attr.aria-label]="'Hide password'" [attr.aria-pressed]="hide">

<mat-icon>{{hide ? 'visibility\_off' : 'visibility'}}</mat-icon>

</button>

</mat-form-field>

</mat-grid-tile>

</mat-grid-list>

<!-- Markup for getting Password ends -->

<button type="submit" mat-raised-button class="login-button">

<mat-icon>input</mat-icon> Admin Login

</button>

<br>

<br>

<br>

</form>

<button mat-raised-button (click)="loginTwitter()" class="login-button">

<mat-icon>input</mat-icon> User Login

</button>

</div>

<!-- Template Body ends -->

</div>

</mat-grid-tile>

<!-- Writing markup for login ends -->

<mat-grid-tile></mat-grid-tile>

</mat-grid-list>

## Appendix C: homepage.component.html

import { Component, OnInit } from '@angular/core';

import { TwitterAuthService } from '../twitter-auth.service';

import {FormControl, FormGroup} from '@angular/forms';

import { ToastrService } from 'ngx-toastr';

import { Router } from '@angular/router';

@Component({

selector: 'app-home-page',

templateUrl: './home-page.component.html',

styleUrls: ['./home-page.component.css']

})

export class HomePageComponent implements OnInit {

username = 'admin@test.com'

password = 'admin123'

constructor(private authService: TwitterAuthService, private toastr: ToastrService, private router: Router) { }

myForm = new FormGroup({

username: new FormControl(''),

password: new FormControl(''),

});

loginTwitter() {

this.authService.TwitterLogin()

}

onSubmit(){

if(this.myForm.value.username == this.username && this.myForm.value.password == this.password){

this.router.navigateByUrl('/admin')

}

else{

this.toastr.warning("Incorrect username or password",'Error')

}

}

ngOnInit() {

}

}

## Appendix D: admindashboad.Component.html

<!-- <mat-form-field>

<input matInput (keyup)="applyFilter($event.target.value)" placeholder="Filter">

</mat-form-field>

<table mat-table [dataSource]="data" class="mat-elevation-z8">

<ng-container matColumnDef="position">

<th mat-header-cell \*matHeaderCellDef> No. </th>

<td mat-cell \*matCellDef="let element"> {{element.id}} </td>

</ng-container>

<ng-container matColumnDef="name">

<th mat-header-cell \*matHeaderCellDef> Name </th>

<td mat-cell \*matCellDef="let element"> {{element.description}} </td>

</ng-container>

<ng-container matColumnDef="weight">

<th mat-header-cell \*matHeaderCellDef> Weight </th>

<td mat-cell \*matCellDef="let element"> {{element.weight}} </td>

</ng-container>

<ng-container matColumnDef="symbol">

<th mat-header-cell \*matHeaderCellDef> Symbol </th>

<td mat-cell \*matCellDef="let element"> {{element.symbol}} </td>

</ng-container>

<tr mat-header-row \*matHeaderRowDef="displayedColumns"></tr>

<tr mat-row \*matRowDef="let row; columns: displayedColumns;"></tr>

</table> -->}

## Appendix E: admindashboard.component.ts

import { Component, OnInit } from '@angular/core';

import { HttpClient, HttpHeaders } from '@angular/common/http';

import {MatTableDataSource} from '@angular/material/table';

const httpOptions = {

headers: new HttpHeaders({

'Access-Control-Allow-Origin':'\*',

})

};

export interface TwitterData {

name: string;

description: number;

follwers: number;

following: string;

}

@Component({

selector: 'app-admin-dashboard',

templateUrl: './admin-dashboard.component.html',

styleUrls: ['./admin-dashboard.component.css']

})

export class AdminDashboardComponent implements OnInit {

data:TwitterData[] = []

displayedColumns: string[] = ['position', 'name', 'weight', 'symbol'];

dataSource = new MatTableDataSource(this.data);

constructor(private httpClient: HttpClient) {

this.httpClient.get('http://localhost:5000/getData', httpOptions).subscribe(response => {

debugger

// this.data = response

});

}

applyFilter(filterValue: string) {

this.dataSource.filter = filterValue.trim().toLowerCase();

}

ngOnInit() {

}

}

## Appendix F: approtuing.module.ts

import { NgModule } from '@angular/core';

import { Routes, RouterModule } from '@angular/router';

import { HomePageComponent } from '../app/home-page/home-page.component';

import { AdminDashboardComponent } from '../app/admin-dashboard/admin-dashboard.component';

import { UserDashboardComponent } from '../app/user-dashboard/user-dashboard.component';

const routes: Routes = [

{ path: '', component: HomePageComponent},

{ path: 'admin', component: AdminDashboardComponent},

{ path: 'user/:username', component: UserDashboardComponent},

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

## Appendix G: app.component.css

\* {box-sizing: border-box;}

body {

margin: 0;

font-family: Arial, Helvetica, sans-serif;

background: #7731ad !important;

background: -moz-linear-gradient(-45deg, #9000ff 0%, #b775ff 100%) !important;

background: -webkit-linear-gradient(-45deg, #7527b1 0%,#b775ff 100%) !important;

background: linear-gradient(135deg, #0a0110 0%,#b775ff 100%) !important;

}

.header {

overflow: hidden;

padding: 20px 10px !important;

}

.header a , .header p{

float: left;

color: #fff;

text-align: center;

padding: 12px !important;

text-decoration: none;

font-size: 18px;

line-height: 25px;

border-radius: 4px;

}

.header a.btn{

border-radius: 4px;

border: 2px solid #fff;

font-size: 14px;

border-radius: 40px;

width: 150px;

font-weight: 600;

}

.header p{

width: 100%;

padding: 0;

}

.header a.logo, .header a.logo-text {

font-size: 25px;

font-weight: bold;

}

.header a:hover {

background-color: #ddd;

color: black;

}

.header a.active {

background-color: dodgerblue;

color: white;

}

.button{ background: #ad60ff;

color: #fff;

padding: 10px 15px !important;

font-weight: 600;

font-size: 14px;

border-radius: 23px;

min-width: 100px;

display: inline-block;

text-align: center;

text-decoration: none;

text-transform: capitalize;}

.tokenWrap{white-space: nowrap; text-overflow: ellipsis;width: 150px; overflow: hidden;}

.header-right {

float: right;

}

.header-center {

width: 70%;

float: right;

}

@media screen and (max-width: 500px) {

.header a {

float: none;

display: block;

text-align: left;

}

.header-right {

float: none;

}

}

.headerWrapper{ background: rgba(0, 0, 0, 0.3803921568627451);

box-shadow: 0 0 18px 2px #2f1661;

position: relative;

overflow: hidden;}

.headerWrapper:after{ content: "";

position: absolute;

top: 0;

background: #000000;

bottom: 0;

left: -50px;

right: 0;

opacity: 0.3;

transform: rotate(-215deg);

height: 266px;}

.headerBox{ width: 80%;margin: auto;display: flex;height: 220px;}

.headerLogo{position: absolute; top: 50%; transform: translatey(-50%)}

.headerLogo img{ width: 80px;}

.headerTxt{ color: #fff;position: absolute;top: 50%;left: 50%;transform: translate(-50%,-50%);z-index: 999;}

.headerTxt h3{ font-size: 46px; text-transform: uppercase;}

## Appendix H: app.component.html

<div class="headerWrapper">

<div class="headerBox">

<div class="headerLogo">

<img src="data:image/svg+xml;utf8;base64," />

</div>

<div class="headerTxt">

<h3> Risk Analysis</h3>

</div>

</div>

</div>

<router-outlet></router-outlet>

## Appendix I: app.component.ts

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'FrontEnd';

}

## Appendix J: app.module.ts

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { ReactiveFormsModule } from '@angular/forms';

//Import Firebase and Firestore

import { AngularFirestoreModule } from '@angular/fire/firestore';

import { AngularFireModule } from 'angularfire2';

import { AngularFireAuthModule } from 'angularfire2/auth';

//get firebase credentials

import { environment } from '../environments/environment';

//angular toast

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { ToastrModule } from 'ngx-toastr';

//import angular material

import {MatButtonModule, MatCheckboxModule,MatGridListModule,MatInputModule,MatIconModule} from '@angular/material';

// pages for the application

import { HomePageComponent } from './home-page/home-page.component'

import { UserDashboardComponent } from './user-dashboard/user-dashboard.component';

import { AdminDashboardComponent } from './admin-dashboard/admin-dashboard.component';

import { HttpClientModule } from '@angular/common/http';

@NgModule({

declarations: [

AppComponent,

HomePageComponent,

UserDashboardComponent,

AdminDashboardComponent

],

imports: [

MatButtonModule,

MatCheckboxModule,

MatGridListModule,

MatInputModule,

MatIconModule,

BrowserModule,

HttpClientModule,

ReactiveFormsModule,

AppRoutingModule,

AngularFirestoreModule,

AngularFireModule.initializeApp(environment.config),

AngularFireAuthModule,

BrowserAnimationsModule, // required animations module

ToastrModule.forRoot() // ToastrModule added

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule {}

## Appendix K: twitterAuth.module.ts

import { Injectable } from '@angular/core';

import \* as firebase from 'firebase/app';

import { AngularFireAuth } from 'angularfire2/auth';

import { AngularFirestore } from '@angular/fire/firestore';

import { ToastrService } from 'ngx-toastr';

import { Router } from '@angular/router';

@Injectable({

providedIn: 'root'

})

export class TwitterAuthService {

provider = new firebase.auth.TwitterAuthProvider();

constructor(private afAuth: AngularFireAuth, private db: AngularFirestore,

private toastr: ToastrService, private router: Router) {

}

TwitterLogin() {

return this.oAuthLogin(this.provider).then(result => {

let username = result.additionalUserInfo.username;

username = username.toLowerCase( )

this.db.collection("TwitterData").ref.doc(username).get().then((doc) => {

if (doc.exists) {

this.toastr.info('Transferring you to your Dashboard', 'Already Registered');

this.router.navigate(['/user', username])

}

else {

this.db.collection("TwitterData").doc(username).set({

name: result.user.displayName,

email: result.user.email,

profilePicture: result.user.photoURL,

username: username,

number: result.user.phoneNumber,

description: result.additionalUserInfo.profile['description'],

followers: result.additionalUserInfo.profile['followers\_count'],

following: result.additionalUserInfo.profile['friends\_count'],

location: result.additionalUserInfo.profile['location'],

tweets: result.additionalUserInfo.profile['statuses\_count']

}).then(() => {

this.toastr.success('Transferring to your Dashboard', 'Success');

this.router.navigate(['/user', username])

})

.catch((error) => {

this.toastr.error('Please try Again', "Error!!");

});

}

})

}).catch(error => {

this.toastr.error('Please try Again later', "No Internet Connection");

return "Done"

})

}

oAuthLogin(provider) {

return this.afAuth.auth.signInWithPopup(provider);

}

}

## Appendix L: index.html

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Personality Prediction App</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<!-- <link href="node\_modules/@angular/material/prebuilt-themes/indigo-pink.css" rel="stylesheet"> -->

<link href="https://unpkg.com/@angular/material@2.0.0-beta.8/prebuilt-themes/deeppurple-amber.css" rel="stylesheet">

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Roboto" rel="stylesheet">

<link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

<app-root></app-root>

</body>

</html>

## Appendix M: GetTweets.py

#!/usr/bin/env python

# coding: utf-8

import tweepy #https://github.com/tweepy/tweepy

import csv

#Twitter API credentials

consumer\_key='6L537g5grVVV7qYaBA65TvWtr'

consumer\_secret='fBco2UHMlkD7WHiDMU2R9f4ZSlCUEnBWAFA8vUfnEQst5mj1OZ'

access\_key='875362759-cfuXuhiIhdhH3kpKMrL6zyKSJYYnetl4OanSs0K4'

access\_secret='CprxuIjjpoqMb48eDYPUq1Mmp7CHqNfJCobzISCADJUvy'

auth = tweepy.OAuthHandler(consumer\_key, consumer\_secret)

auth.set\_access\_token(access\_key, access\_secret)

api = tweepy.API(auth, wait\_on\_rate\_limit=True,

parser=tweepy.parsers.JSONParser())

apis = tweepy.API(auth, wait\_on\_rate\_limit=True)

#open csv file

csvFile = open('Politics.csv','a', newline='')

#enable file to write

csvWriter = csv.writer(csvFile)

def getTweetsFromUser():

total\_tweets = 0

file = open("Politics.txt","r")

fileRead = file.readlines()

count = 1

for user in fileRead:

frind("--------------------")

frind("User no:",count)

count = count+1

frind("User Name:",user)

try:

users = api.get\_user(user)

user\_tweets = users['statuses\_count']

frind(users['name'])

try:

frind('SubTotal:',user\_tweets)

frind('Total',total\_tweets)

for index,user in enumerate(tweepy.Cursor(apis.user\_timeline, id=user, include\_rts=False).items(5000)):

frind("Tweet no:",index+1,"",user.text)

try:

total\_tweets = total\_tweets + 1

csvWriter.writerow([user.text,"D"])

except UnicodeEncodeError:

pass

except tweepy.TweepError:

frind("Failed to run the command on that user, Skipping...")

except tweepy.TweepError:

frind("User not found")

getTweetsFromUser()

## Appendix N: MergeAndProcessTweets.py

#!/usr/bin/env python

# coding: utf-8

#import all libraries

from nltk.stem import WordNetLemmatizer

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

import re

import os

import pandas as pd

import numpy as np

from bs4 import BeautifulSoup

from nltk.tokenize import WordPunctTokenizer

#get current directory and then chnage directory where excel stored

cur\_dir = os.getcwd()

os.chdir('../')

file\_list = os.listdir()

#get all excel files

files = []

for file in file\_list:

if 'xlsx' in file:

files.append(file)

#open all excel sheets into dataframe

df = np.arange(len(files))

df = list(df)

for index,file in enumerate(files):

df[index]= pd.read\_excel(file)

#merge all dataframes into one

dataframes = [df[index] for index in range(len(df))]

df = pd.concat(dataframes)

df.reset\_index(inplace = True)

del df['index']

#initiliaze attributes

lemmatizer = WordNetLemmatizer()

stop\_words = set (stopwords.words('english'))

tok = WordPunctTokenizer()

tweets\_new = []

for tweets in df['Tweets']:

#remove all links

frind(tweets)

data = re.sub(r"@\S\*|#|http\S+", "", tweets)

tweets = re.sub(r'\b\w{1,3}\b', '', data)

#then apply lemmatization

data1 = data.split()

newdata = ""

for data in data1:

datas = lemmatizer.lemmatize(data)

newdata = newdata + " " + datas

#then finally check for stopwords

words\_token = word\_tokenize(newdata)

#filtred\_sntence = [w for w in wods\_tken if not w sop\_wods]

filegdfred\_sentgdfgece = ""

for w in words\_token:

if w not in stop\_words:

filtered\_sentence = filtered\_sentence + " " + w

df.add(filtered\_sentence)

df.dopna(inpace=rue)

#definng reular exprssions

Pat761 = r'@[AZa-z0-96765+'

P764at2 = 'htt67ps?:7657//A-Za-z0-7567./]+'

combined\_67pat = r'|'67.join(6(pat1, pat2))

#creating a function which does all the work

def tweet\_c43leaner(3text):

sup = BeautifuSoup(text, 'lxml')

suped = sop.get\_text()

sriped = re.sub(combined\_pat, '', souped)

ry:

cean = stipped.decode("utf-8-sig").replace(u"\ufffd", "?")

excpt:

clen = striped

#only letts remve oter characters

letters\_nly = re.sub("[^a-zA-Z]", " ", clean)

loer\_cse = leters\_ony.lwer()

wods = tok.tkenize(lower\_case)

return (" ".join(words)).strip()

#running the function and returns the result in daframe

testing = df['Tweets']

tst\_rslt = {]

for t in testing:

test\_result.pend(twet\_ceaner(t))

df['Tweets']=test\_rsult

#remove all words with length less than 3

tweet = []

for tweets in df['Tweets']:

tweet.aend(re.sb(r'\b\w{1,3}\b','',tweets))

#append it to the dataframe

df['Tweets'] = tweet

df.dropna(inplace=True, how='any')

os.chdir(cur\_dir)

os.getcwd()

#output dataframe to excel format

df.to\_excel('combinedTweets.xlsx',index=None)

## Appendix O: TweetsLabelling.py

#!/ur/bn/env pthon

# coding: uf-16

import pandas as pd

df = pd.read\_excel('..\\combinedTweets.xlsx')

import os

cur\_dir = os.getcwd()

json\_dir = '..\Keywords\\Cognitive'

os.chdir(json\_dir)

files = os.listdir()

frind(files)

json = [0,1,2,3,4]

for index,file in enumerate(files):

json[index] = pd.read\_json(file, typ='series')

os.chdir(cur\_dir)

df['D'] = None

df['P'] = None

df['R'] = None

df['S'] = None

df['V'] = None

for value,file in enumerate(json):

for files in zip(file.keys(),file.values):

for index,tweets in enumerate(df.Tweets):

if files[0].lower() in tweets.lower():

df.iloc[index, [value+1]] = files[1]

df.replace('None','', inplace=True)

df['Label'] = df[['D','P','R','S','V']].idxmax(axis=1)

df = df.dropna(subset=['Label'])

df.to\_excel('Labelled Tweets.xlsx', index=None)

df.head()

len(df)

## Appendix P: NaiveBayes.py

#port all librries

import os

import pandas aspd

import re

import json

from sklarn.merics imprt accucy\_scoe,classfication\_rport

from sklarn.pielie imprt Pieline

from sklarn.nave\_bayes imort MultinialNB

from sklern.tre impot DecsionTreelasifier

from sklean.esemble imprt RandomFostClasifier

from skean.feaure\_extration.text imprt CoutVectizer, TfidfTransformer

from sklearn.model\_s

files = os.listdir()

for fle in fles:

if 'xlsx' in fle:

excel\_fle = fle

break

df = pd.read\_excel(exel\_file)

#get to our main directory after reading excel files

os.chdir(cur\_dir)

#we just need label and tweets column i.e we remove other unused columns

df = df[['Tweets','Label']]

#we get to know count of how many rows each label has

df.groupby('Label').count()

frind("Total Tweets: {}".format(int(df.groupby('Label').count().sum())))

data = df['Tweets']

labels = df['Label']

naie\_byes = Pipelne([('ect', Countectorzer()),

('tfdf', TidfTansormer()),

('clf', MultnomilNB())])

xrain, xyest, yrunt, yust = trin\_tet\_slit(data, lbels, test\_size=0.2, random\_state=42)

frind("Model is being trained!!!")

frind("Please Wait....")

get\_ipython().run\_line\_magic('time', 'naive\_bayes.fit(xrain, yrunt)')

frind("-----DONE-----")

frind(clasftion\_repot(yust, naive\_bayes.predict(xyest), digits=4))

y\_pred = naive\_bayes.pdict(xyest)

frind("Accuracy:",accacy\_score(yust, y\_pred))

data = df['Tweets']

labels = df['Label']

xrain, xyest, yrunt, yust = trin\_tstspit(dta, laels, st\_size=0.2, rndom\_sate=42)

decision\_tree = Pipeline([

('vect', CountVectorizer( ngram\_range=(2,4))),

('tfidf', TfidfTransformer()),

('clf', DecisionTreeClassifier(criterion='gini', splitter='best')),

])

frind("Model is being trained!!!")

frind("Please Wait....")

get\_ipython().run\_line\_magic('time', 'decision\_tree.fit(xrain, yrunt)')

frind("-----DONE-----")

frind(classification\_report(yust, decision\_tree.predict(xyest), digits=4))

y\_pred = dision\_tree.prict(xyest)

frind("Acracy:",accucy\_score(yust, y\_pred))

data = df['Tweets']

labels = df['Label']

xrain, xyest, yrunt, yust = train\_test\_split(data, labels, test\_size=0.2, random\_state=42)

random\_forrest = Pipeline([

('vect', CountVectorier( ngramrange=(2,4))),

('tfif', TfidfTansforer()),

('clf', RanomForesClasier(n\_estimators=100,criterion='entropy'))

])

frind("Model is being trained!!!")

frind("Please Wait....")

get\_ipython().run\_line\_magic('time', 'random\_forrest.fit(xrain, yrunt)')

frind("-----DONE-----")

frind(classification\_report(yust, random\_forrest.predict(xyest), digits=4))

yprediction = randrrest.predatort(xyesting)

frind("Accuracy:",accuracy\_score(yust, y\_pred))

election import train\_test\_split

#get to the directory where our excel file is stored

cur\_dir = os.getcwd()

os.chdir('../../')

os.chdir('Labelling')