

RECRUITMENT PROCESS USING ARTIFICIAL INTELLIGENCE

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

G.L.V CHAITANYA RAJU (38110093)
RAMANI CHANDRA BABU (38110097)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

MARCH - 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of G.L.V CHAITANYA RAJU (38110093), RAMANI CHANDRA BABU (38110097) who carried out the project entitled "**RECRUITMENT PROCESS USING ARTIFICIAL INTELLIGENCE**" under my supervision from October 2021 to May 2022

Internal Guide

(Mr. Murari Devakannan kamalesh .M.E(Ph.D))

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner External Examiner

DECLARATION

I G.L.V CHAITANYA RAJU, RAMANI CHANDRA BABU hereby declare that the Project Report entitled **RECRUITMENT PROCESS USING ARTIFICIAL INTELLIGENCE** done by me under the guidance of Mr.Murari Devakannan kamalesh .M.E(Ph.D) (Internal) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree In COMPUTER SCIENCE AND TECHNOLOGY.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr.Murari Devakannan kamalesh .M.E(Ph.D)** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

An industry or a business or a firm requires quality personnel for accomplishment of objectives framed by them in order to survive in this competitive era. They all are in the beginning of fourth industrial revolution. To remain competitive in this digital world all search for bright, potential and dynamic employees. Organizations with an effective recruitment strategy will be able to employ suitable individual in order to manage the digital world and developing business environment. So the recruitment strategy is the prime factor for every organization in hiring skilled employees who could be more efficient and effective in accomplishing the job objectives. The recruitment strategy as it is a major function of organization apparently takes help of data analysis for decision making process. The data analysis is known as “Artificial Intelligence” which plays a crucial role in recruitment decision. Artificial intelligence in a most basic terminology and is a human develop intelligent machines. AI will work and react like human and its ultimate goal is to facilitate computers to carry out the work as normally done by people. AI leads with an incredible speed and accuracy. The major objective of this paper is to study how Artificial Intelligence influences the recruitment strategy.

CONTENT

Chapter	List of content	Page no.
I	Abstract	1
ii	List of figures	3
1	Introduction	4
2	Literature Survey	7
3	System Requirements	12
	3.1 Hardware Requirements	
	3.2 Software Requirements	
	3.3 Language Specification	
	3.4 History of python	
	3.5 Application of python	13
	3.6 Features of python	14
	3.7 Feasibility of study	
	3.7.1 Economical feasibility	15
	3.7.2 Technical feasibility	
	3.7.3 Social feasibility	
	3.8 Python GUI	16
	3.8.1 Tkinter Programming	
	3.8.2 Tkinter Widgets	
	3.8.3 Standard Attributes	17
4	System Analysis	19
	4.1 Purpose	
	4.2 Scope	
	4.3 Exsisting System	
	4.4 Proposed system	20
5	System Design	22
	5.1 Input design	
	5.2 Output design	
6	Modules Description	28
7	System implementation	31
	7.1 System architecture	
8	System testing	32
	8.1 Test plan	
	8.2 Verification	
	8.3 Validation	
	8.4 Basics of software testing	
	8.5 Black box testing	
	8.6 White box testing	33
	8.7 Types of testing	

	8.7.1 Unit testing	
	8.7.2 Integration testing	34
	8.7.3 Functional testing	
	8.7.4 System testing	
	8.7.5 Stress testing	
	8.7.6 Performance testing	37
	8.7.7 Usability testing	
	8.7.8 Acceptance testing	
	8.7.9 Regression testing	
9	Result And Discussion	39
	Conclusion	45
Iv	References	46

LIST OF FIGURES:

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	Tkinter Window	17
7.1	System Architecture	31

CHAPTER 1

INTRODUCTION

Organizations develop and sustain by innovating new ideas to compete in the digital era. Innovation leads to decrease in manpower and increase in machine handling. Organizations will also need to train its employees to handle machines, software or any equipment for that matter. Now organizations are trying to implement recruitment techniques which lead to acquire talented employees. Artificial Intelligence term was coined by John McCarthy (1950) soon after when he published a paper entitled “Computing Machinery and intelligence” and this paper lead to open the doors to the filed termed as AI. Now, HR managers implement Artificial Intelligence technology to recruit, retain and inspire the proficient manpower which leads to success and growth for both the employer and employee. AI in recruiting has a major role in talent acquisition (Madeline Laurano Co-Founder & Chief Research Office). Nearly 30% of companies implementing AI for recruiting activity which can lesser down time, cost reduction and can place right talent in the right position. Artificial Intelligence is defined as “An area of study in the field of computer science. Artificial intelligence is concerned with the development of computers able to engage in human-like thought processes such as learning, reasoning, and self-correction”. Artificial intelligence is all about fast thinking with lot of knowledge, think as human, logical reasoning, etc., as it is a part of computer science leading to efficiency in problem solving by giving solutions.

1.1 OBJECTIVE

OBJECTIVES

- To observe how artificial intelligence is used currently in recruiting process.
- To understand the importance of artificial intelligence in recruitment.

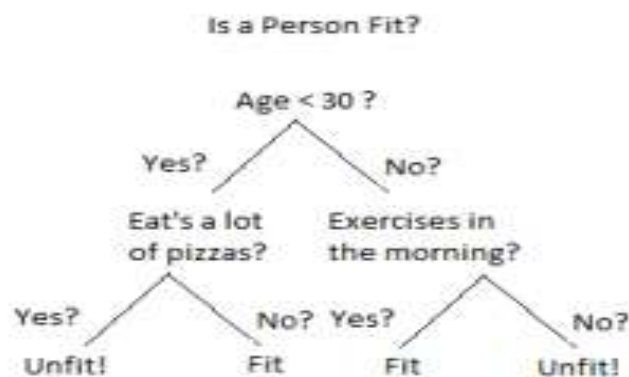
1.2 PROPOSED ALGORITHMS

PROPOSED ALGORITHM

Decision Trees for Classification: A Machine Learning Algorithm

Introduction

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.



An example of a decision tree can be explained using above binary tree. Let's say you want to predict whether a person is fit given their information like age, eating habit, and physical activity, etc. The decision nodes here are questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot

of pizzas'? And the leaves, which are outcomes like either 'fit', or 'unfit'. In this case this was a binary classification problem (a yes no type problem).

There are two main types of Decision Trees:

Classification trees (Yes/No types)

What we've seen above is an example of classification tree, where the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is Categorical.

Regression trees (Continuous data types)

Here the decision or the outcome variable is Continuous, e.g. a number like 123.

Working

Now that we know what a Decision Tree is, we'll see how it works internally. There are many algorithms out there which construct Decision Trees, but one of the best is called as ID3 Algorithm. ID3 Stands for Iterative Dichotomiser 3.

Before discussing the ID3 algorithm, we'll go through few definitions.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

[1] Requirements validation techniques: An empirical study

Requirements validation makes sure that the requirements written in software requirements specification (SRS) must be complete and

consistent and are according to the customer's needs. It ensures the validity of user requirements by eliminating ambiguities and inconsistencies from SRS. Several techniques for requirements validation have been discussed in the literature. This paper gives an overview of requirements validation techniques which have been practicing in industry, which includes requirements inspections, requirements prototyping, requirements testing and viewpoint-oriented requirements validation. This paper also highlights pros and cons of these techniques. In requirements testing, special attention is given to TCD inspections.

[2]A review on software requirements validation and consistency management

Requirements need to be validated at the early phase of the software development to avoid errors such as inconsistency, incompleteness and incorrectness. Drawn from this argument, a requirements validation process needs to consider Consistency, Completeness and Correctness ("3 Cs") for the production of a quality software specifications. This paper provides a review of requirements validation and consistency management based on the existing literature in order to identify the gaps in the existing knowledge on the process of software requirements specifications. This paper begins with a review of the definitions of the 3Cs, upon which the understanding of the 3Cs is derived. Next comprehensive review of related works on the identified consistency management techniques: traceability and analysis approaches are then presented. This is supported with a heat map representations of the related research on the types of contributions, techniques, specifications and semantics used in consistency management. Since semi-formal specifications were found as the most

common representation of the requirements, the types of models used as semi-formal specifications to represent the requirements were also discussed. Overall, this paper identifies the various gaps existing within the process of validating and managing the consistency of requirements to avoid re-inventing the wheels in the diverse and comprehensive knowledge of requirements engineering.

[3] Improving software quality using machine learning

Software is an entity that keeps on progressing and endures continuous changes, in order to boost its functionality and maintain its effectiveness. During the development of software, even with advanced planning, well documentation and proper process control, are problems that are countered. These defects influence the quality of software in one way or the other which may result into failure. Therefore, in today's neck to neck competition, it is our requirement to control and minimize these defects in software engineering. Software prediction models are typically used to map the patterns of classes of software that are prone to change. This paper highlights the significant analysis in the area's subject to learn and stimulate the association between the metric specifying t'e object orientation & the concept of change proneness. This would often lead us to rigorous testing so as to find all kinds of possibilities in the data set. We have two views to be addressed: (1) Parameters quantification that affects the quality, functionality and productivity of the software. (2) Machine learning technologies are used for predicting software Here, the focus of the research paper is to equate and compare all of learning methods corresponding to performance parameter with its statistical method &

methodology which would often results enhanced. Data points are the basis for prediction of models.

[4] Software requirement analysis: Research challenges and technical approaches

Requirement analysis is one of the key challenges in software development projects. Customer requirement specification and management entails various impacts to software projects and still is an improvement area on both academic and industrial fields. Models like CMMI also uncovers requirement development and management and specifies the specific goals and practices for them. In this paper, key challenges and issues of requirement management are listed with respect to a standardization activity, namely CMMI.

[5] Requirement validation model for virtual distributed system

Requirement engineering plays very important role in almost every field of computing and development. Many different techniques or models have been proposed for requirement engineering process. All these models follow different techniques to solve the requirement engineering issues. There are a number of general activities common to all processes which are requirement elicitation, requirement analysis, requirement specification, requirement validation and requirement management. Currently many researches have been proposed for various requirement validation techniques. All these techniques proposed by researchers majorly focus on centralized projects. In this research paper, a requirement validation framework has been proposed for distributed virtual projects. Main purpose of this framework is to enhance the quality of the distributed virtual systems

by providing easy and systemic way for validating the requirements of such systems. Distributed Virtual Environments are those systems which are designed for users who are located at different geographical areas and uses different networks. These are designed without any requirement specification from end users. Our proposed framework provides a systematic way to requirement engineers through which they could analyze the requirements that come from user end on the basis of some factors like completeness, correctness etc. For developing proposed framework, three approaches have been followed; Prototyping, Review and Test Based. The successful implementation of proposed requirement on distributed virtual environment can have a good influence for building quality software in future

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

System: Pentium Dual Core

Hard Disk: 120 GB

Monitor: 15" LED

Input Devices: Keyboard, Mouse

Ram: 4 GB

3.2 SOFTWARE REQUIREMENTS

Operating system: Windows 7/10

Coding Language: Python

3.3 LANGUAGE SPECIFICATION

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This **tutorial** gives enough understanding on **Python programming** language.

3.4. HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

3.5. APPLICATION OF PYTHON

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

3.6 FEATURES OF PYTHON

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

3.7 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

3.7.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.7.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.7.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

3.8 PYTHON GUI

This project is implemented using Python GUI. Python provides various options for developing graphical user interfaces (GUIs). Most important are listed below.

- **Tkinter** – Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.
- **wxPython** – This is an open-source Python interface for wxWindows
- **Jpython** – Jpython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine <http://www.jython.org>.

There are many other interfaces available, which you can find them on the net

3.8.1 Tkinter Programming

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the Tkinter module.

- ❑ Create the GUI application main window.
- ❑ Add one or more of the above-mentioned widgets to the GUI application.
- ❑ Add the PhotoImage.
- ❑ Enter the main event loop to take action against each event triggered by the user.

```
#!/usr/bin/python
import Tkinter

top = Tkinter.Tk()

# Code to add widgets will go here...
Top.mainloop
```

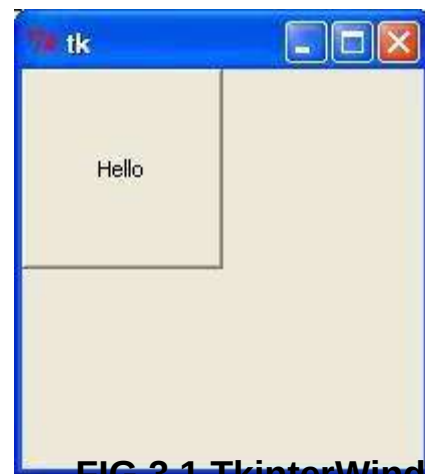


FIG 3.1 TkinterWindow

3.8.2 Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in TABLE 4.3

3.8.3 Standard Attributes

Let us take a look at how some of their common attributes such as sizes, colors and fonts are specified.

❖ Dimensions

- ❖ Colors
- ❖ Anchors
- ❖ Relief styles
- ❖ Bitmaps
- ❖ Cursors

Tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

1. **pack() method** :It organizes the widgets in blocks before placing in the parent widget.
2. **grid() method** :It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3. **place() method** :It organizes the widgets by placing them on specific positions directed by the programmer.

CHAPTER 4

SYSTEM ANALYSIS

4.1 PURPOSE

The purpose of this document is detection of traffic sign using machine learning algorithms. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project – such as interface, functional requirements and performance requirements.

4.2 SCOPE

The scope of this SRS document persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRS to the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

4.3 EXISTING SYSTEM

Most of previous literatures have placed arguments regarding job loss due to automation or AI as an intelligent machine that can potentially become a threat to human as they threaten human efficiency. However it was really necessary for this research from the perspective of Human Resources because not all the jobs are susceptible to alteration due to the fourth industrial wave. According to WEF (WEF,) reports jobs with the requirements of unique skillsets will be in demand despite the threat of redundancy. Moreover, Ireland being the less inclined country among its

peer European countries have rise to the idea of exploring further into the domain. In this paper the researcher has tried to find the correlation between the bulk hiring drives and the effectiveness of implementing AI into finding top performing talents that will be interested in the organizational development for the long run. Due to the serious gap in knowledge access and relevant resources. It was a challenge to find sufficient data and link it with the research objectives.

DISADVANTAGE OF EXISTING SYSTEM

- Low retention
- Low recruitment

4.4 PROPOSED SYSTEM

Artificial intelligence is a technology which can work smartly as equal to human brain in different situations. It gains attention and importance in automating recruiting system when compared to traditional recruitment methods. Recruitment is the central activity of all organization to work on. Now, recruitment industry is taking up growth by implementing smart way to recruit i.e., recruiting through artificial intelligence. And numerous industries are focusing on the changes taking place in recruitment process. AI technology has tremendous impact on recruitment activity as it enables the recruiter to align all some unstructured candidate bio-data, construct profile into uniformity, identify and match skill sets required for the industry. In today's era, recruiters think AI technology is competing with them for recruitment activity. But, it is human built software to make the work at ease while the process is carried on. To conclude the role of AI it is the combination of humans and AI that leads to data maintenance, save the

cost and time to the organizations with more accuracy and access in total recruitment process.

ADVANTAGES OF PROPOSED SYSTEM

- Time saving
- Cost saving
- Hire with Quality

CHAPTER 5

SYSTEM DESIGN

5.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

5.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action

5.3 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

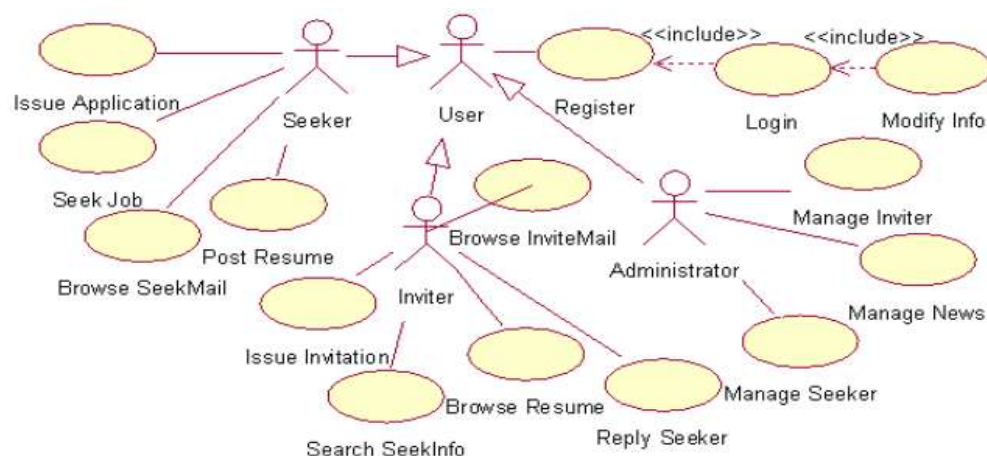
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the mode modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

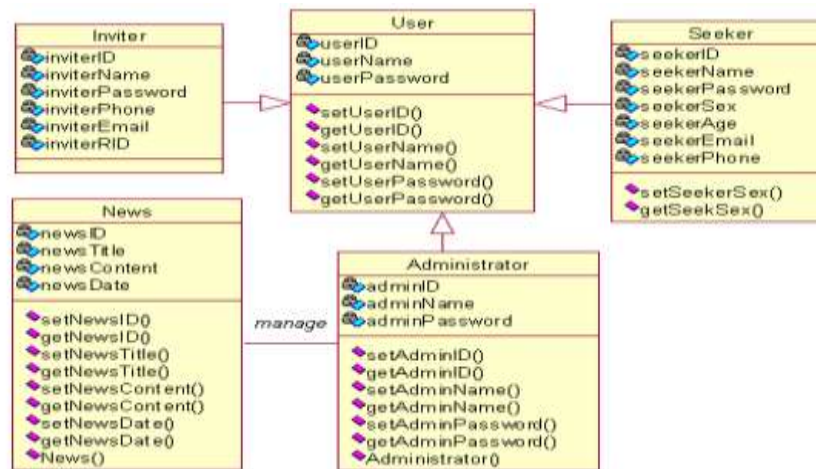
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



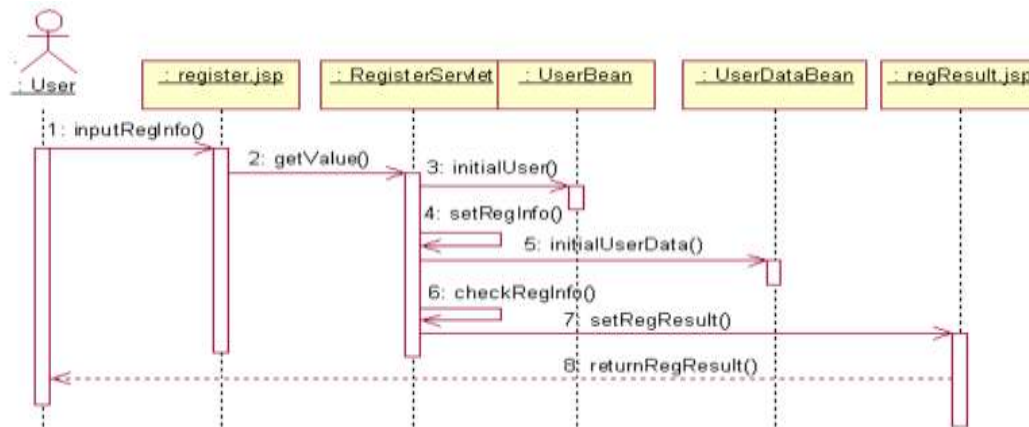
CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



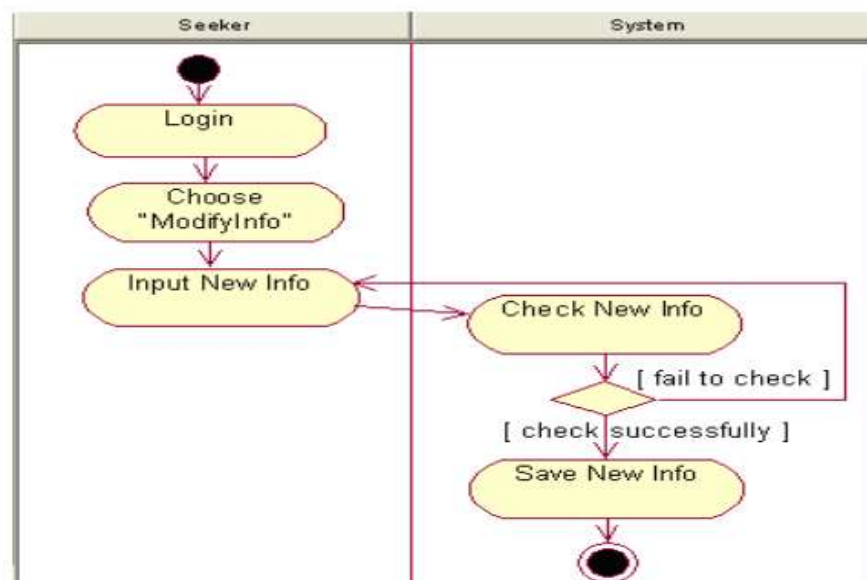
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



CHAPTER 6

6.1 MODULES

- Data Sourcing Module
- Resume Screening Module
- Candidate Matching Module
- Checking Reference Module
- Legal Aspect Module

MODULES DESCRIPTION

6.1.1 Data Sourcing Module

This means that today, as a recruiter, you will need to analyze social networks to identify correctly who a candidate truly is. Moreover, you can program AI based bots to automatically source candidates for you based on your criteria. Your data strategy needs to incorporate social networks to automatically cross reference information from a CV that will be scanned and analyzed in order to ensure that it tells the truth about the candidate's experience. An AI algorithm will be able to source and analyze hundreds of

millions of profiles. This is obviously something a human would not be able to do. AI is about human augmented and here the mundane task to source and check is taken care of as well as the communication with candidates whilst the process is ongoing. This will ensure engagement from all.

6.1.2 Resume Screening Module

The screening process is also about to improve. As you will have identified valuable candidates based on your criteria, you will not waste any time. Your potential talents would have a know-how in your industry and will be able to start rather quickly. Candidate ranking will be more efficient, and the right candidate will come out quicker. Facial recognition will ensure that you are talking to the right candidate and false identity will then be a problem of the past.

6.1.3 Candidate Matching Module

Candidate matching will move towards a personalized candidate experience and with the help of a strongly backed AI algorithm you will be able to tailor a candidate experience to your own needs. Finding someone with 5 to 10 years' experience with strong skills, network and know-how will become an easy task because the variety and varcity of data will allow detailed and precise analysis and extract of data. Candidate experience will be optimized so that you will truly understand what engages your candidate. You will also be able to know when to intervene to ensure longer service. For example, if you see that every 3 years your candidate changes jobs and responsibility you will be able to predict and offer something new before he/she reaches that critical timing.

6.1.4 Checking Reference Module

Checking references will be simpler. Your AI algorithm will use all types of data available. It will be able to cross reference people and build a network with starfish type of connections. You will then either contact their references, if possible, or check what has been written or said on social networks and make your own opinion.

6.1.5 Legal Aspect Module

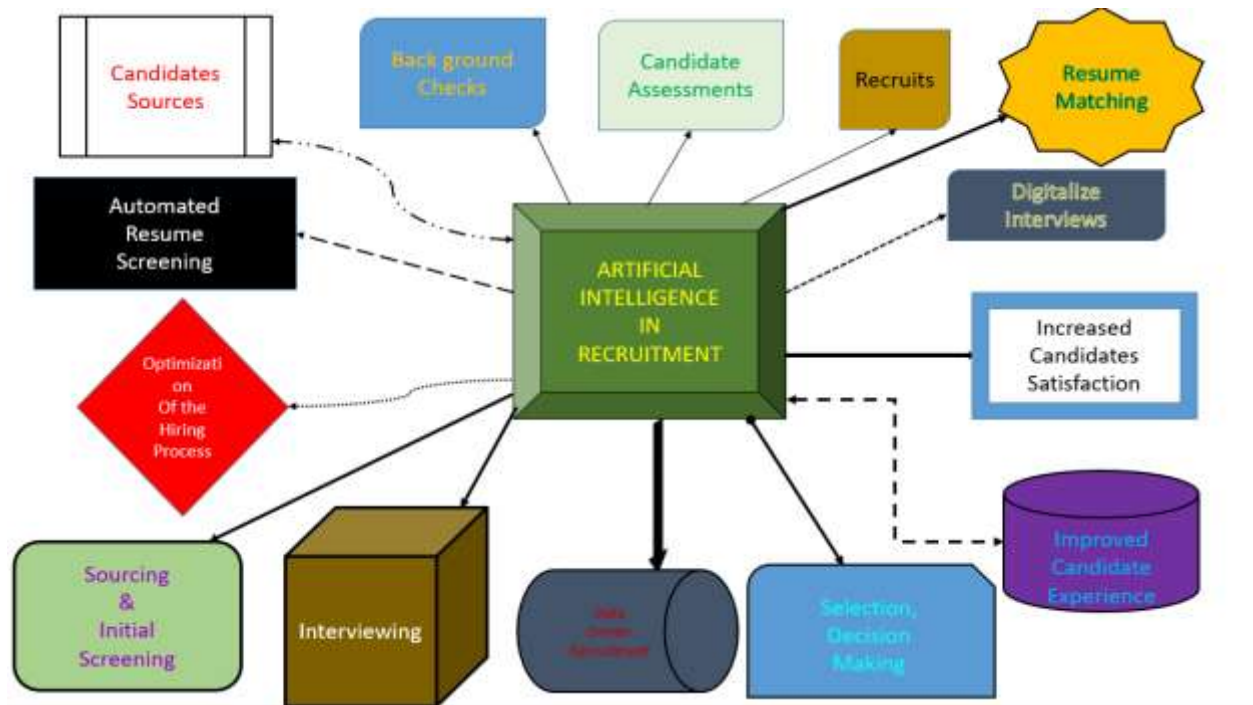
Once all these steps have been completed, the legal aspect will become automated. The legal team will be augmented as they will have in their possession all data available automatically gathered during the recruitment process. Contracts will take into consideration all information discussed during the recruiting process and will be written with human input to ensure it matches all legal requirements.

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1SYSTEM ARCHITECTURE

Describing the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system.



CHAPTER 8

SYSTEM TESTING

8.1 Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

8.2 Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

8.3 Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

8.4 Basics of software testing

There are two basics of software testing: black box testing and white box testing.

8.5 Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

8.6 White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

8.7 Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

8.7.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

8.7.2 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

8.7.3 Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

8.7.4 System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

8.7.5 Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

8.7.6 Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

8.7.7 Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How

pleasing is it to use its design? This falls under the class of black box testing.

8.7.8 Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

8.7.9 Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing

REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system

depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that it has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system,

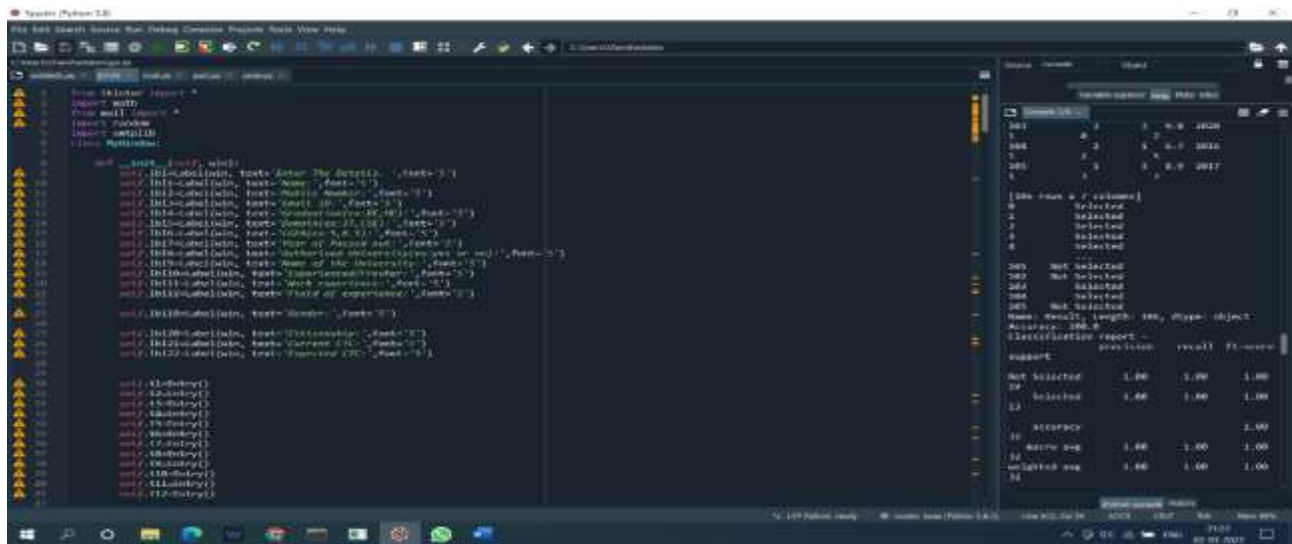
analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

CHAPTERV –9

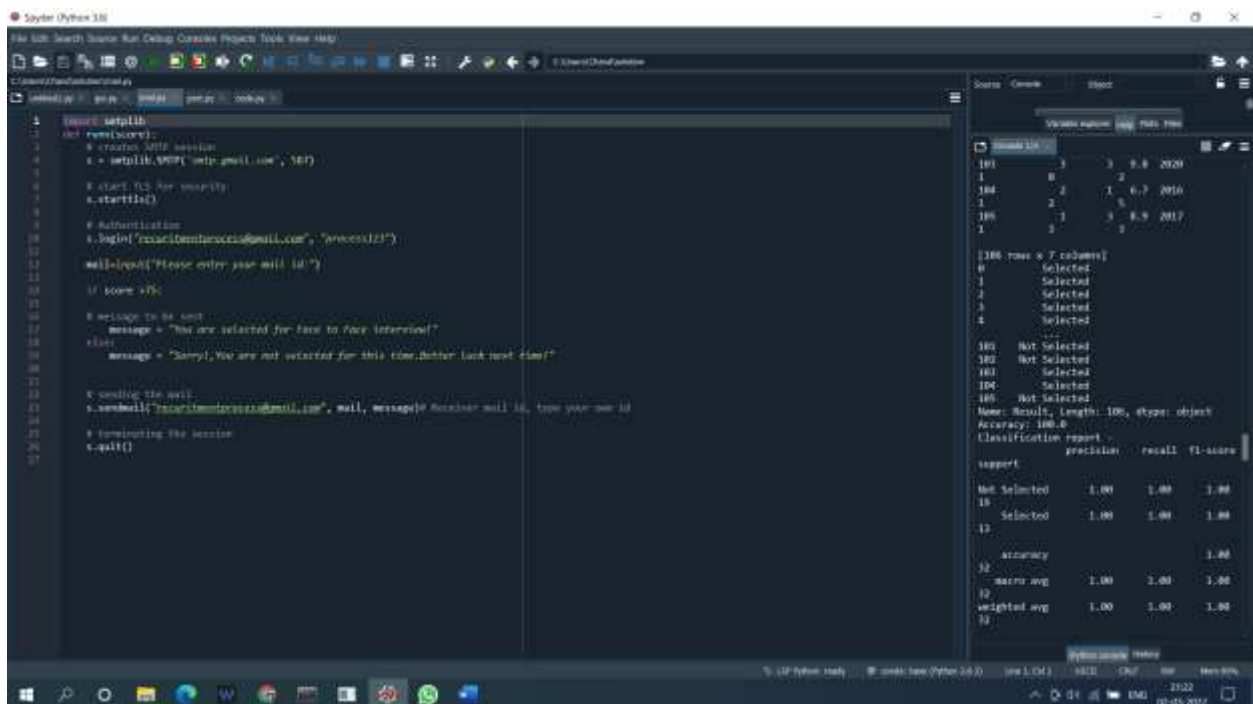
RESULT AND DISCUSSION

SOURCE CODE

Gui.py



mail.py



Part.py

```

1 import pandas as pd
2 from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
3 from sklearn.model_selection import train_test_split # Import train_test_split function
4 from sklearn.metrics import r2_score # Import r2_score metric
5 from sklearn.metrics import classification_report, confusion_matrix # Import classification report and confusion matrix functions
6 df = pd.read_csv('data.csv')
7
8 features = df[['Graduation', 'GPA', 'Hours', 'Intensity', 'Experience', 'G-Cat-Iacs']]
9 target = df['Result']
10
11 print(features)
12 print(target)
13
14 # Split dataset into training set and test set
15 X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.3, random_state=1) # 70% training and 30% test
16 # Create a Decision Tree Classifier object
17 clf = DecisionTreeClassifier()
18 # Train Decision Tree Classifier
19 clf = clf.fit(X_train, y_train)
20 # Predict the response for test dataset
21 y_pred = clf.predict(X_test)
22 # Print the result
23 print("Accuracy: ", accuracy_score(y_test, y_pred)*100)
24
25 print("Classification report: ", y_test, classification_report(y_test, y_pred))
26 print(type(y_test))

```

Variables explorer: 100.0, 1.00, 1.00, 1.00

Classification report:

	precision	recall	f1-score
Not Selected	1.00	1.00	1.00
Selected	1.00	1.00	1.00
accuracy			1.00
macro avg	1.00	1.00	1.00
weighted avg	1.00	1.00	1.00

Code.py

```

1 # Importing tkinter module
2
3 Created on Wed Mar 22 10:45:00 2022.
4
5 @author: criss
6
7
8 from tkinter import *
9
10 # Python program to create a simple GUI
11 # Simple Quiz using Tkinter
12
13 # Import everything from tkinter
14 from tkinter import *
15
16 # And import messages as tk from tkinter
17 from tkinter import messagebox as mb
18
19 # Import json to use json file for data
20 import json
21
22 # Class to define the components of the GUI
23 class Quiz:
24     # This is the first method which is called when a
25     # new object of the class is initialized. This method
26     # sets the question count to 0, and initializes all the
27     # other methods to display the content and make all the
28     # functionalities available.
29     def __init__(self):
30         # Set question number to 0
31         self.q_no = 0
32
33         # Assigns ques to the display_question function to update later.
34         self.display_title()
35         self.display_question()
36
37         # Set selected holds an integer value which is used for
38         # selected option. In a quiz it is.
39         self.qnt_selected = IntVar()
40
41         # A distinctive width button for the current question and used to

```

Variables explorer: 100.0, 1.00, 1.00, 1.00

Classification report:

	precision	recall	f1-score
Not Selected	1.00	1.00	1.00
Selected	1.00	1.00	1.00
accuracy			1.00
macro avg	1.00	1.00	1.00
weighted avg	1.00	1.00	1.00

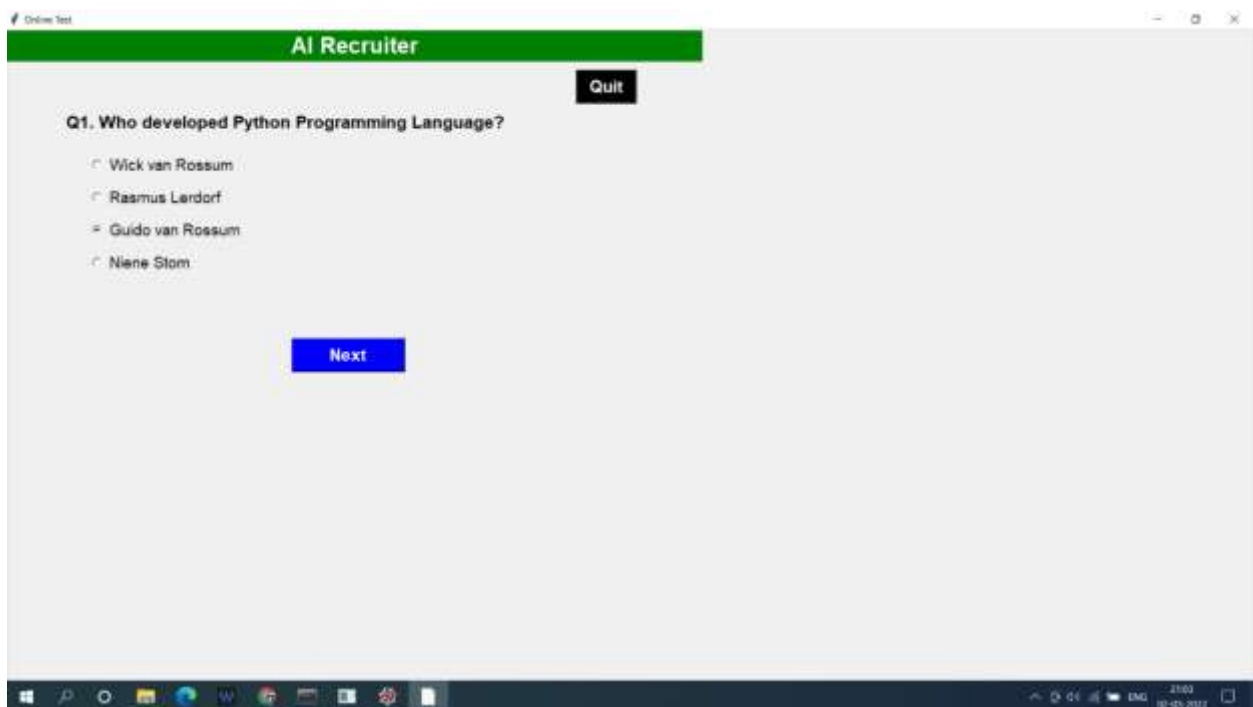
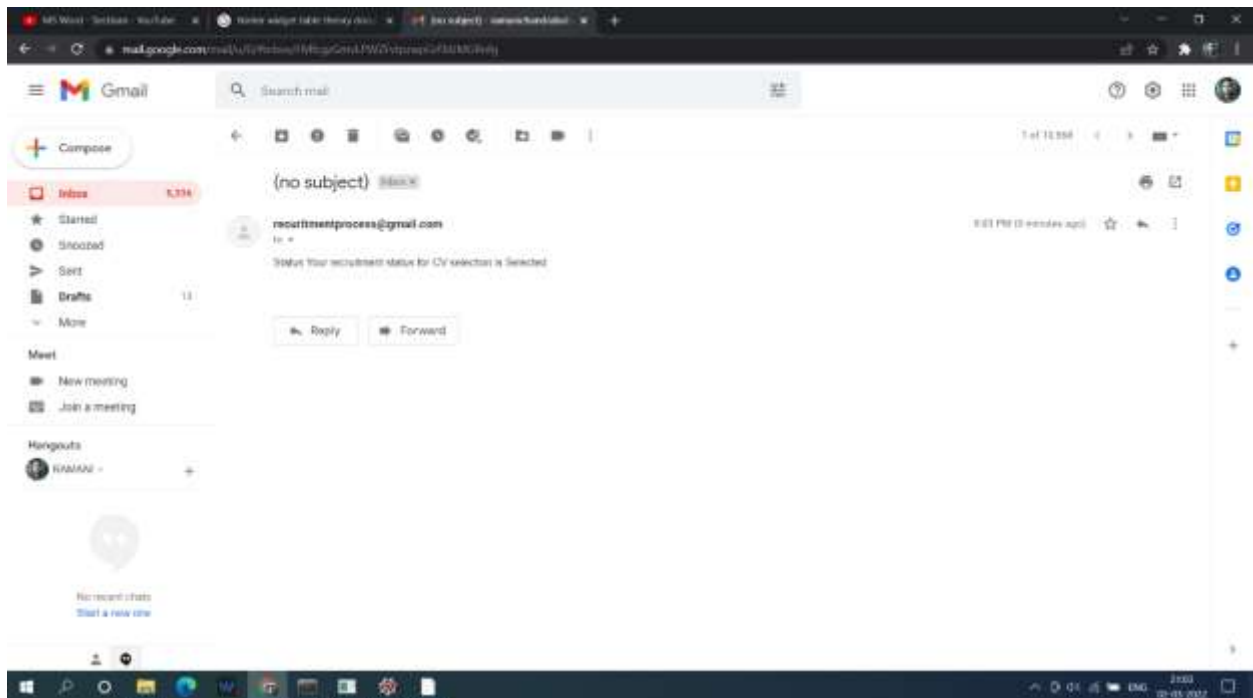
SCREENSHOTS

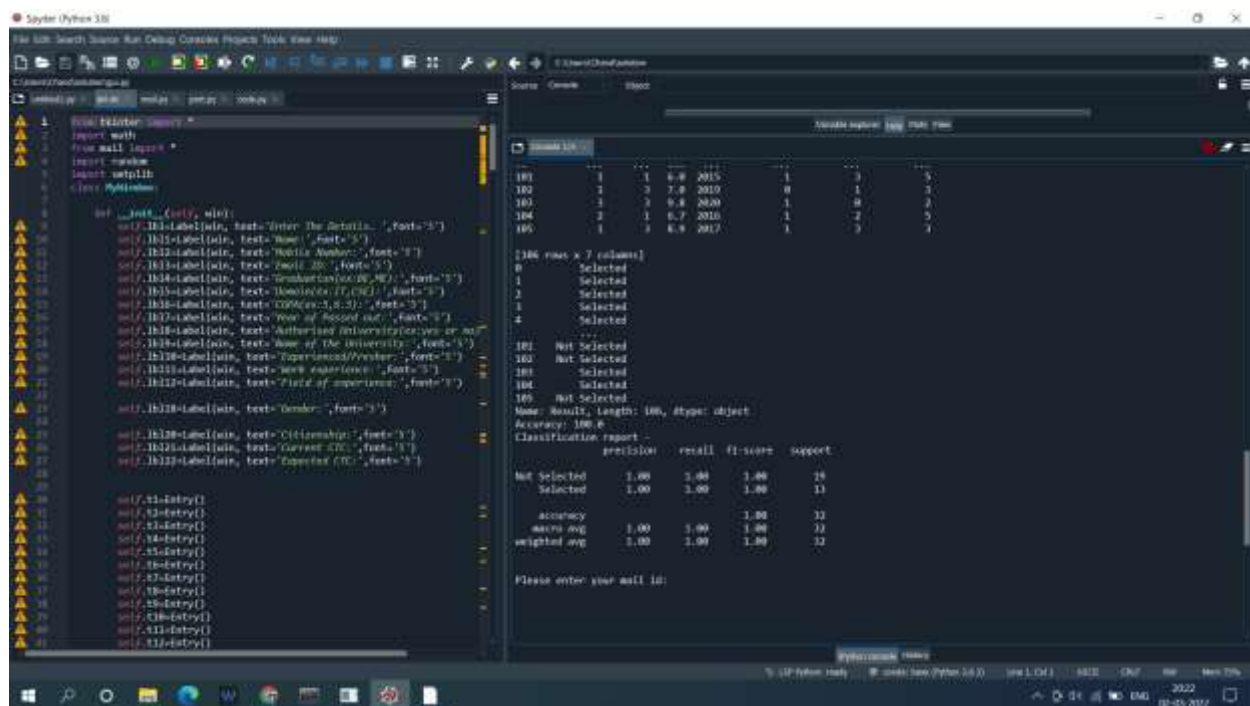
Enter The Details.

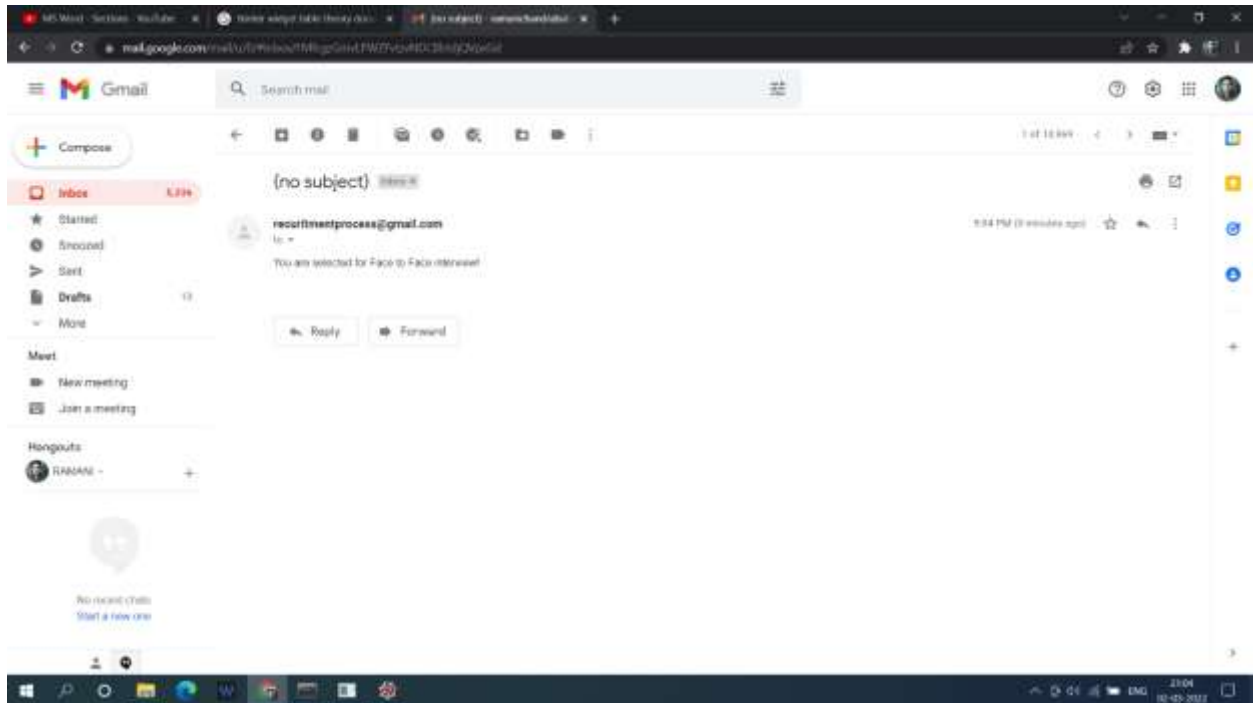
Name:		Name of the University:	
Mobile Number:		Experienced/Fresher:	
Email ID:		Work experience:	
Graduation(ex.BE,ME):		Field of experience:	
Domain(ex.IT,CSE):		Gender:	
CGPA(ex.5,6,5):		Citizenship:	
Year of Passed out:		Current CTC:	
Authorised University(ex.yes or no):		Expected CTC:	

Enter The Details.

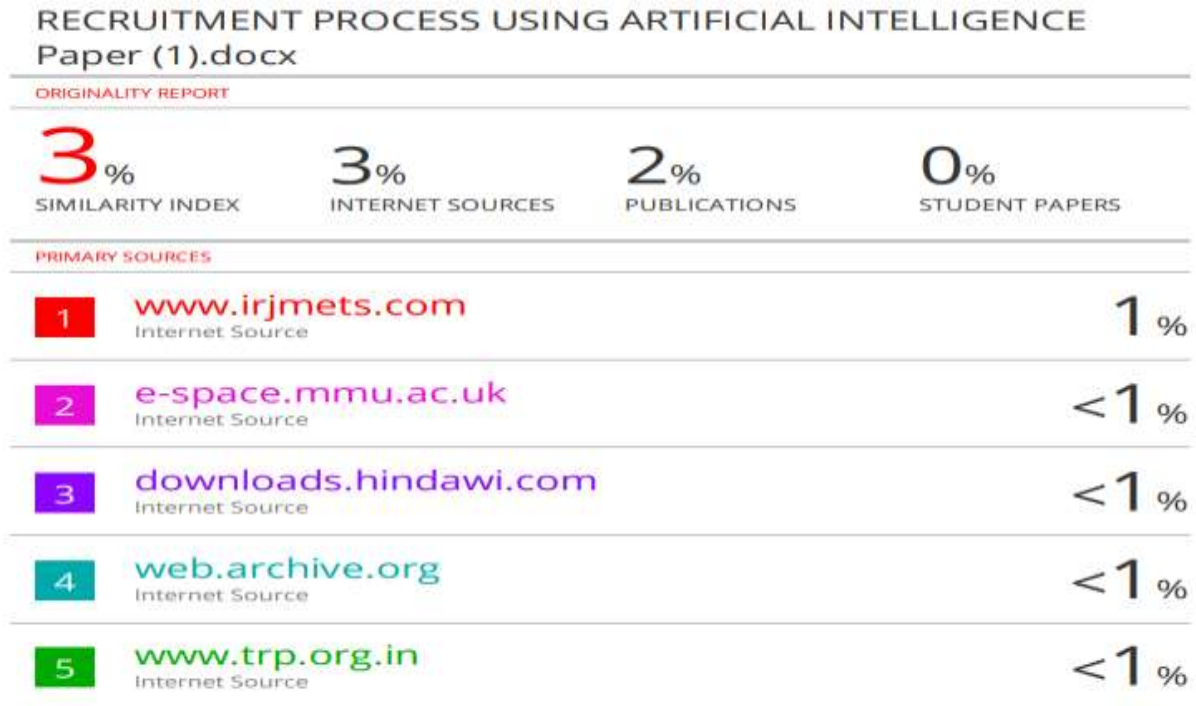
Name:	Ramesh chandra	Name of the University:	MIT
Mobile Number:	9715029441	Experienced/Fresher:	Fresher
Email ID:	satishk2001@gmail.com	Work experience:	0
Graduation(ex.BE,ME):	BE	Field of experience:	Machine Learning
Domain(ex.IT,CSE):	CSE	Gender:	Male
CGPA(ex.5,6,5):	6	Citizenship:	Indian
Year of Passed out:	2022	Current CTC:	0
Authorised University(ex.yes or no):	yes	Expected CTC:	1







Research Paper Plagiarism



CONCLUSION

Artificial intelligence has been the most apt innovative technology that facilitates in better recruitment process when implemented strategically having a significant effect on the performance of the employees. Increase in productivity, training programmes, automation of the process and the reliability of reports improves the performance of employees. Hence the adaptability of this technology results in better performance of the employees and performance indicating variables are highly significant predictors.

REFERENCES

- [1]. H.A. Bilal, M. Ilyas, Q. Tariq and M. Hummayun, "Requirements validation techniques: An empirical study", *International Journal of Computer Applications*, vol. 148, no. 14, 2016.
- [2]. M. Kamalrudin and S. Sidek, "A review on software requirements validation and consistency management", *International Journal of Software Engineering and Its Applications*, vol. 9, no. 10, pp. 39-58, 2015.
- [3]. K. Chandra, G. Kapoor, R. Kohli and A. Gupta, "Improving software quality using machine learning" in 2016 ICICCS-INBUSH, IEEE, pp. 115-118, February 2016.
- [4]. S.T. Demirel and R. Das, "Software requirement analysis: Research challenges and technical approaches", *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 1-6, 2018, March.
- [5]. T. Kiran, S. Farhan, H. Tauseef and M. A. Fahiem, "Requirement validation model for virtual distributed system", *Int. J. Multidiscip. Sci. Eng.*, vol. 4, no. 8, pp. 34-42, 2013.
- [6]. Dhirendra Pandey, Waris Khan and Vandana Pandey, *ROLE OF REQUIREMENT VALIDATION IN REQUIREMENT DEVELOPMENT*, 2012.
- [7]. U.A. Raja, "Empirical studies of requirements validation techniques", *2009 2nd International Conference on Computer Control and Communication*, pp. 1-9, 2009, February.

[8] .X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning", *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1-130, 2009.