1.INTRODUCTION

1.1 INTRODUCTION

Recruiters must be able to properly screen resumes in order to hire the right individual at the right time. The process of deciding whether a candidate is qualified for a position based on his or her qualification, education, work-experience, and other information from their CV is known as resume screening. The importance of efficient and effective resume screening is at the heart of any strong recruitment strategy. The goal of resume screening is to find the best candidates for a position. In the current system, candidates must fill out a manual form with all of their resume information, which takes a long time, and then they are dissatisfied with the position that the current system prefers based on their qualifications. Our method will work in the same way as a handshake between two people. i.e. the employer prefers the best candidate available, and the candidate chooses the best position possible based on his or her talents and abilities. Our system is a resume ranking software that uses natural language processing (NLP) and machine learning. This AI-powered resume screening programme goes beyond keywords to contextually screen resumes. Following resume screening, the software rates prospects in real time depending on the recruiter's job needs. In order to match and rate candidates in real time, the software employs natural language processing and machine learning.

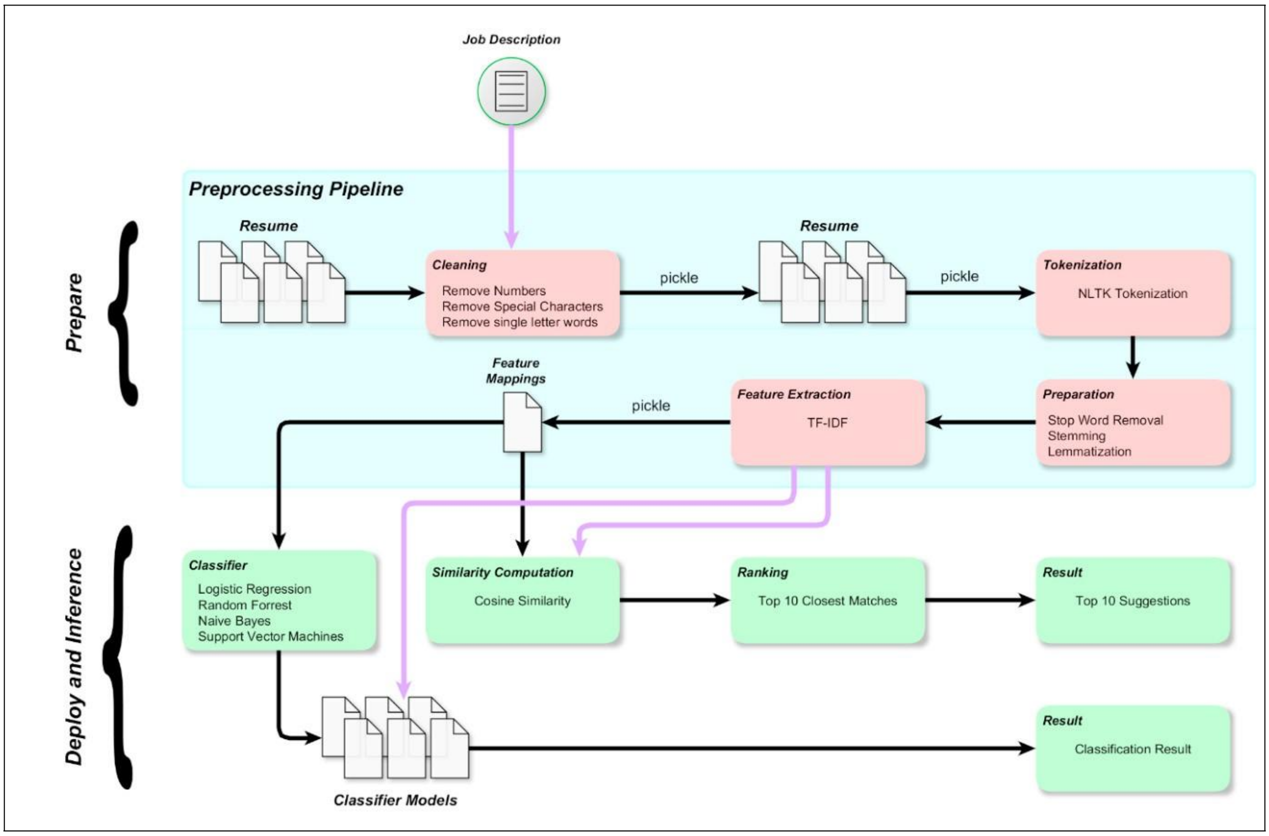
1.2 AIM OF PROJECT

For each recruitment, companies take out online ads, referrals and go through them manually. Companies often submit thousands of resumes for every posting. When companies collect resumes through online advertisements, they categorize those resumes according to their requirements. After collecting resumes, companies close advertisements and online applying portals. Then they send the collected resumes to the Hiring Team(s). It becomes very difficult for the hiring teams to read the resume and select the resume according to the requirement, there is no problem if there are one or two resumes but it is very difficult to go through 1000’s resumes and select the best one. To solve this problem, today in this article we will read and screen the resume using machine learning with Python so that we can complete days of work in few minutes**.**

2.LITERATURE SURVEY

The recruitment process in today’s world has witnessed a major change with the evolution of technologies like the intenet.The following sections summarises some of the library work performed in this domain of e-recruitment systems. The proposed solutions use various approaches with the automated screening of candidates. It discusses different machine learning algorithms and uses support vector regressions to list of ranked candidates for the given job.Our work takes a different approaches as it focuses mainly on the content of the resumes where we perform the extraction of skills and related parameters to match with the job descriptions

2.1BLOCK DIAGRAM OF RESUME SCREENING



2.2 RESUME SCREENING

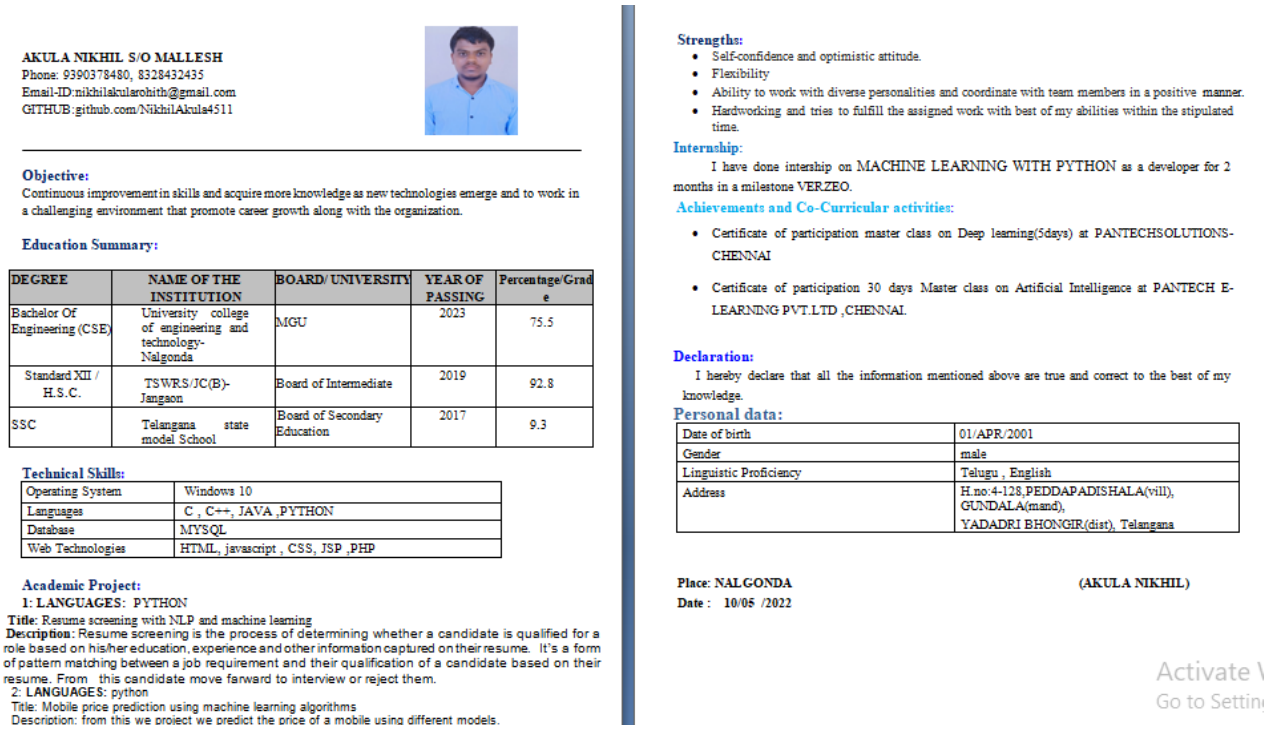
2.2.1 RESUME

A resume is a formal document that a job applicant creates to customize their qualifications for a job position.A resume is usually accompanied by a customized cover letter in which applicant express an intrest in a specific job or company and draws attention to the most relevant specifics on the resume.

Successful resume highlight specific accomplishments applicants have achieved in former positions, such as cutting costs,transcending sales goals, increasing profits and building out teams.

The most determined applicants rewrite their resumes to suit the occasion, concentrating on skills and experience that fit the job for which they were applying.

Sample resume as follows:



AKULA NIKHIL S/O MALLESH
Phone: 9390378480, 8328432435
Email-ID:nikhilakularohith@gmail.com
GITHUB:github.com/NikhilAkula4511

Objective:

Continuous improvement in skills and acquire more knowledge as new technologies emerge and to work in a challenging environment that promote career growth along with the organization.

Education Summary:

DEGREE	NAME OF THE INSTITUTION	BOARD/ UNIVERSITY	YEAR OF PASSING	Percentage/Grade
Bachelor Of Engineering (CSE)	University college of engineering and technology-Nalgonda	MGU	2023	75.5
Standard XII / H.S.C.	TSWRS/JC(B)-Jangaon	Board of Intermediate	2019	92.8
SSC	Telangana state model School	Board of Secondary Education	2017	9.3

Technical Skills:

Operating System	Windows 10
Languages	C , C++, JAVA ,PYTHON
Database	MYSQL
Web Technologies	HTML, javascript , CSS, JSP ,PHP

Academic Project:

1: LANGUAGES: PYTHON

Title: Resume screening with NLP and machine learning

Description: Resume screening is the process of determining whether a candidate is qualified for a role based on his/her education, experience and other information captured on their resume. It's a form of pattern matching between a job requirement and their qualification of a candidate based on their resume. From this candidate move farward to interview or reject them.

2: LANGUAGES: python

Title: Mobile price prediction using machine learning algorithms

Description: from this we project we predict the price of a mobile using different models.

Strengths:

- Self-confidence and optimistic attitude.
- Flexibility
- Ability to work with diverse personalities and coordinate with team members in a positive manner.
- Hardworking and tries to fulfill the assigned work with best of my abilities within the stipulated time.

Internship:

I have done intership on MACHINE LEARNING WITH PYTHON as a developer for 2 months in a milestone VERZEO.

Achievements and Co-Curricular activities:

- Certificate of participation master class on Deep learning(5days) at PANTECHSOLUTIONS-CHENNAI
- Certificate of participation 30 days Master class on Artificial Intelligence at PANTECH E-LEARNING PVT.LTD ,CHENNAI.

Declaration:

I hereby declare that all the information mentioned above are true and correct to the best of my knowledge.

Personal data:

Date of birth	01/APR/2001
Gender	male
Linguistic Proficiency	Telugu , English
Address	H.no:4-128,PEDDAPADISHALA(vill), GUNDALA(mand), YADADRI BHONGIR(dist), Telangana

Place: NALGONDA **(AKULA NIKHIL)**
Date : 10/05 /2022

Fig: RESUME OF A CANDIDATE WHO APPLIED FOR A JOB

2.2.2 RESUME SCREENING

Resume screening is the process of determining whether a candidate is qualified for a role based on her education , experience , and other information captured on their resume. In a nutshell , it's a form of pattern matching between a job's requierements and the qualifications of a candidate based on their resume.

The goal of screening resumes is to decide whether to move a candidate forward – usually onto an interview – or to reject them.

The resume screening process

1. Screening for must-haves

What you are looking for is a set of skills, education or experience, or anything else the candidate needs to perform the job. This stringent list is the fulcrum of your job description. If a candidate does not have one of these, you eliminate them from the hiring process.

2. Screening for happy-to-haves

Your happy-to-have list can be a sum of things that add an advantage to the candidate because it can enhance their performance on the job role. For example, a candidate from a similar domain as the one you are hiring for may have a better chance at successfully performing on the role.

3. Screening based on general impressions

As you scan through resume or candidate information, look for anything that stands out. It could be the candidates' proficiency in a certain language, attention to detail, leadership skills, etc. These might not directly influence your hiring decision but can tell you if you can confidently advance a candidate to the next round or not.

Choosing the right people for the job is the biggest responsibility of every business since choosing the right set of people can accelerate business growth exponentially. We will discuss here an example of such a business, which we know as the IT department. We know that the IT department falls short of growing markets. Due to many big projects with big companies, their team does not have time to read resumes and choose the best resume according to their requirements. To solve this type of problem, the company always chooses a third party whose job is to make the resume ADVA NCED NLP PROJ ECT PYTHON STRUCTURED DATA TEXT USE CASES as per the requirement. These companies are known by the name of Hiring Service Organization. It's all about the information resume screen. The work of selecting the best talent, assignments, online coding contests among many others is also known as resume screen. Due to lack of time, big companies do not have enough time to open resumes, due to which they have to take the help of any other company. For which they have to pay money. Which is a very serious problem. To solve this problem, the company wants to start the work of the resume screen itself by using a machine learning algorithm.

3.Methodology

3.1 Existing System

Screening of resumes is done by some of the company's employees who are going to recruit, i.e., every resume is checked individually, and if the resume is fit for the required job description, then the resume will be selected. It might be done based on the capabilities they seek, the candidates' work experience, or other factors that are relevant to the job profile.

3.1.1 Disadvantages

a) Time consuming all resumes must be referred manually, which takes a long time.
b) Recruiters are under a lot of pressure Even if all resumes are manually referred to, the procedure will take twice as long if there is no correct fit for the job profile.
c) Unnecessary resource allocation Recruiters could be working on other projects instead of spending so much time on resume checking.
d) Inefficient Once a requirement is identified, recruiters do not go through all the resumes.
e) Errors Due to many resumes and the little time available for processing, some mistakes may be made

3.2 Proposed system

As per the article named Resume Screening using Machine Learning, the fundamental working procedure is resumes should be in CSV format. The screening process should begin with removing garbage terms (unwanted/repeated words). The remaining words are then screened, and skill points are granted. And the skill points will be assigned in the appropriate order. Finally, a graph will be displayed due to the skill points, allowing eligible candidates for the job role to be selected.

3.2.1Advantages

- A large number of resumes can be scanned at a time.
- The correct fit for the job role can be identified easily.
- It takes less time to complete the process.

3.3 Introduction to Python

Python is a general purpose, dynamic, high level, and interpreted programming language It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development. Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development. Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc. We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable. Python makes the development and debugging fast because there is no compilation step included in Python development and edit-test-debug cycle is very fast.

PYTHON’S FEATURE SET:

The factors that played an important role in moulding the final form of the language and are given by

1. Easy to code: Python is very easy to code. Compared to other popular languages like Java and C++, it is easier to code in Python.
2. Easy to read: Being a high-level language, Python code is quite like English. Looking at it, you can tell what the code is supposed to do. Also, since it is dynamically-typed, it mandates indentation. This aids readability.
3. Expressive: Suppose we have two languages A and B, and all programs that can be made in A can be made in B using local transformations. However, there are some programs that can be made in B, but not in A, using local transformations. Then, B is said to be more expressive than A. Python provides us with a myriad of constructs that help us focus on the solution rather than on the syntax.
4. Free and Open-Source: Firstly, Python is freely available. You can download it from the link. Secondly, it is open source. This means that its source code is available to

the public. You can download it, change it, use it, and distribute it. This is called FLOSS (Free/Libre and Open Source Software.

5. High- Level: This means that as programmers, we don’t need to remember the system architecture. Nor do we need to manage the memory. This makes it more programmer- friendly and is one of the key python features.

6. Portable: We can take one code and run it on any machine, there is no need to write different code for different machines. This makes Python a portable language.

7. Interpreted: If you are any familiar with languages like C++ or Java, you must first compile it, and then run it. But in Python, there is no need to compile it. Internally, its source code is converted into an immediate form called bytecode. So, all you need to do is to run your Python code without worrying about linking to libraries, and a few other things. By interpreted, we mean the source code is executed line by line, and not all at once. Because of this, it is easier to debug your code. Also, interpreting makes it just slightly slower than Java, but that does not matter compared to the benefits it has to offer.

8. Object-Oriented: A programming language that can model the real world is said to be object-oriented. It focuses on objects and combines data and functions. Contrarily, a procedure-oriented language revolves around functions, which are code that can be reused. Python supports both procedure-oriented and object-oriented programming which is one of the key python features. It also supports multiple inheritance, unlike Java. A class is a blueprint for such an object. It is an abstract data type and holds no values. 9. Extensible: If needed, you can write some of your Python code in other languages like C++. This makes Python an extensible language, meaning that it can be extended to other languages.

10. Embeddable: We just saw that we can put code in other languages in our Python source code. However, it is also possible to put our Python code in a source code in a different language like C++s. This allows us to integrate scripting capabilities into our program of the other language.

11. Large Standard Library: Python downloads with a large library that you can use so you don’t have to write your own code for every single thing. There are libraries for regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and a lot of other functionality.

12. GUI Programming: It also supports graphical user interface.

13. Dynamically Typed Python is dynamically-typed. This means that the type for a value is decided at runtime, not in advance.This is why we don’t need to specify the type of data while declaring it.

3.4 System Requirements

3.4.1 Hardware requierements

Processor : Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz 2.30 GHz
RAM : 4.00 GB
Hard Disk: 1TB

3.4.2 Software requirements

Operating system: Windows 10 pro
Language : python 3.9
Software : Anaconda (jupyter notebook)

4.Modules Design

4.1 Modules Description

4.1.1 Data collection

Data collection is the process of gathering data for use in business decision-making, strategic planning, research and other purposes. It's a crucial part of data analytics applications and research projects: Effective data collection provides the information that's needed to answer questions, analyze business performance or other outcomes, and predict future trends, actions and scenario.

We have publically available data from Kaggle. You can download the data using the below link. https://www.kaggle.com/gauravduttakiit/resume-dataset.

4.1.2 Exploratory Data Analysis

Exploratory data analysis (EDA) is a (mainly) visual approach and philosophy that focuses on the initial ways by which one should explore a data set or experiment. Two main aspects of EDA are:

1. **Openness**, meaning a person exploring the data should be open to all possibilities prior to its exploration.
2. **Skepticism**, meaning one must ensure that the obvious story the data tells is not misleading.

There is no formal set of techniques that are used in EDA. Remember, EDA is an approach to how we analyze data, not a specific set of methods set in stone. It's a philosophy and art more so than a science.

Its purpose is to take a general view of some given data without making any assumptions about it. We are trying to get a feel for the data and what it might mean, as opposed to reject or accept some sort of premise around it, before we begin its exploration.

In other words, with EDA we let the data speak for itself instead of trying to force the data into some sort of predetermined model.

Nevertheless, some techniques are used to help us get a feel for the data. For instance, we can categorize data, quantify some of its basic aspects, or visualize it.

For instance, raw data can be plotted using histograms or other visualization techniques. Sometimes, the data is juxtaposed in a manner that helps us spot important patterns within or between data sets.

Let’s have a quick view of the data we have.

resumeDataSet.head()

There are only two columns we have in the data.

Below is the definition of each column.

Category: Type of Job Resume fits for.

Resume: Resume of candidates

resumeDataSet.shape

There are 962 observations we have in the data. Each observation represents the complete details of each candidate so we have 962 resumes for screening

4.1.3 Data Preprocessing

The resume's provided as input would be shortlisted in this procedure to remove any special or garbage characters from the resumes. All unique characters, numerals, and words with only single letters are eliminated during cleaning. After these processes, we had a clean dataset with no unique characters, numerals, or single letter words. NLTK tokenizers are used to break the dataset into tokens. Stop word removal, lemmatization and vectorization are among the preprocessing operations performed on the tokenized dataset. The data is masked in the following ways:

- Masking the strings such as \w
- Masking the escape letters like \n
- Masking all the numbers
- By substituting an empty string for all single-letter words
- Stop words are removed
- Lemmatization is performed

Removing Stop Words:

Stop words such as and, the, was, and others appear very often in words and limit the process which determines prediction, thus they are removed. Filtering the Stop Words consists of the following steps:

1. The input words are tokenized into individual tokens and saved in an array.

2. Each word now corresponds to the Stop Words list in the NLTK library:

(a) import stopwords from nltk.corpus

(b) SW[] = set(stopwords.words('english'))

(c) It returns 180 stop words, which may be confirmed using the (len(StopWords)) function and displayed using the print (StopWords) function.

3. When the words appear in the StopWords list, they are removed from the main sentence array.

4. Repeat the above sequence of steps until the tokenized array's last entry is not matched.

5. There are no stop words in the resultant array. Lemmatization: Lemmatization reduces derived phrases to make entirely sure that the underlying word is accurately associated with the language. The routine phases of lemmatization are as follows:

- Convert the text corpus into a list of words

The extraction of features is the next phase. We used the TfIdf (Term Frequency, Inverse Document Frequency) to extract features from a preprocessed dataset. The cleansed data was transferred, and Tf-Idf was used to extract features. Taking the input as a numerical vector, processing a machine learningbased classification model or learning algorithms takes place. The input text with varying length was not processed by MLbased classifiers. As a result, during the preparation procedures, the texts are changed to the required equal length vector form. There are several methods for extracting characteristics, including tf-idf, and others. Using the scikit learn library function, we generated tf-idf for each term. To calculate a tf-idf vector,we use TfidfVectorizer:

1) Sub-linear df is set to True to utilise a logarithmic form for frequency.

2) Min df is the minimal number of documents in which a word must appear in order to be saved.

3) The norm is set to l2 to ensure that all feature vectors have the same euclidean norm.

4) The gramme range is set to (n1, n2), with n1 equaling 1 and n2 equaling 2. It means that both unigrams and bigrams are taken into account. The model takes a job description and a list of CVs as input and returns a list of CVs that are most similar to the job description.

4.1.3 Model Building

In this phase data science team needs to develop data sets for training, testing, and production purposes. These data sets enable data scientist to develop analytical method and train it, while holding aside some of data for testing the model.

Team develops datasets for testing, training, and production purposes. In addition, in this phase, the team builds and executes models based on work done in the model planning phase. The team also considers whether its existing tools will suffice for running the models, or if it will need more robust environment for executing models and workflows (Example – fast hardware and parallel processing).

We will be using the ‘One vs Rest’ method with ‘KNeighborsClassifier’ to build this multiclass classification model. We will use 80% data for training and 20% data for validation. Let’s split the data now into training and test set.

```
X_train,X_test,y_train,y_test = train_test_split(WordFeatures,requiredTarget,random_state=0, test_size=0.2)
```

```
print(X_train.shape) print(X_test.shape)
```

4.1.5 Results

Let see the results we have..

```
print('Accuracy of KNeighbors Classifier on training set: {:.2f}'.format(clf.score(X_train, y_train)))
```

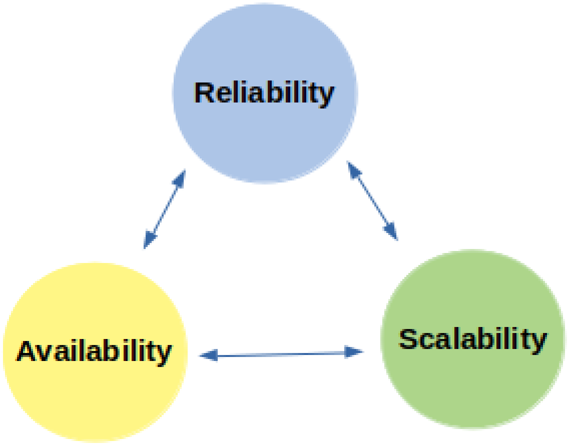
```
print('Accuracy of KNeighbors Classifier on test set: {:.2f}'.format(clf.score(X_test, y_test)))
```

5.IMPLEMENTATION

5.1 SYSTEM DESIGN

System Design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements.

It's a wide field of study in Engineering and includes various concepts and principles that will help you in designing scalable systems. These concepts are extensively asked in the Interview Rounds for SDE 2 and SDE 3 Positions at various tech companies. These senior roles demand a better understanding of how you solve a particular design problem, how you respond when there is more than expected traffic on your system, how you design the database of your system and many more. All these decisions are required to be taken carefully keeping in mind Scalability, Reliability, Availability, and Maintainability



Reliability in System Design

A system is Reliable when it can meet the end-user requirement. When you are designing a system you should have planned to implement a set of features and services in your system. If your system can serve all those features without wearing out then your System can be considered to be **Reliable**.

A **Fault Tolerant** system can be one that can continue to be functioning reliably even in the presence of faults. **Faults** are the errors that arise in a particular component of the system. An occurrence of fault doesn't guarantee Failure of the System.

Failure is the state when the system is not able to perform as expected. It is no longer able to provide certain services to the end-users.

Availability in System Design

Availability is a characteristic of a System which aims to ensure an agreed level of Operational Performance, also known as **uptime**. It is essential for a system to ensure high availability in order to serve the user's requests.

The extent of Availability varies from system to system. Suppose you are designing a Social Media Application then high availability is not much of a need. A delay of a few seconds can be tolerated. Getting to view the post of your favorite celebrity on Instagram with a delay of 5 to 10 seconds will not be much of an issue. But if you are designing a system for hospitals, Data Centers, or Banking, then you should ensure that your system is highly available. Because a delay in the service can lead to a huge loss.

There are various principles you should follow in order to ensure the availability of your system :

- Your System should not have a Single Point of Failure. Basically, your system should not be dependent on a single service in order to process all of its requests. Because when that service fails then your entire system can be jeopardized and end up becoming unavailable.
- Detecting the Failure and resolving it at that point.

Scalability in System Design –

Scalability refers to the ability of the System to cope up with the increasing load. While designing the system you should keep in mind the load experienced by it. It's said that if you have to design a system for load **X** then you should plan to design it for **10X** and Test it for **100X**. There can be a situation where your system can experience an increasing load. Suppose you are designing an E-commerce application then you can expect a spike in the load during a Flash Sale or when a new Product is Launched for sale. In that case, your system should be smart enough to handle the increasing load efficiently and that makes it **Scalable**.
In order to ensure scalability you should be able to compute the load that your system will experience. There are various factors that describe the Load on the System:

- Number of requests coming to your system for getting processed per day
- Number of Database calls made from your system
- Amount of Cache Hit or Miss requests to your system
- Users currently active on your system

5.2 UML Diagrams

The **Unified Modeling Language™** (**UML®**) is a standard visual modeling language intended to be used for

- modeling business and similar processes,
- analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

UML can be applied to diverse **application domains** (e.g., banking, finance, internet, aerospace, healthcare, etc.) It can be used with all major object and component **software development methods** and for various **implementation platforms** (e.g., J2EE, .NET).

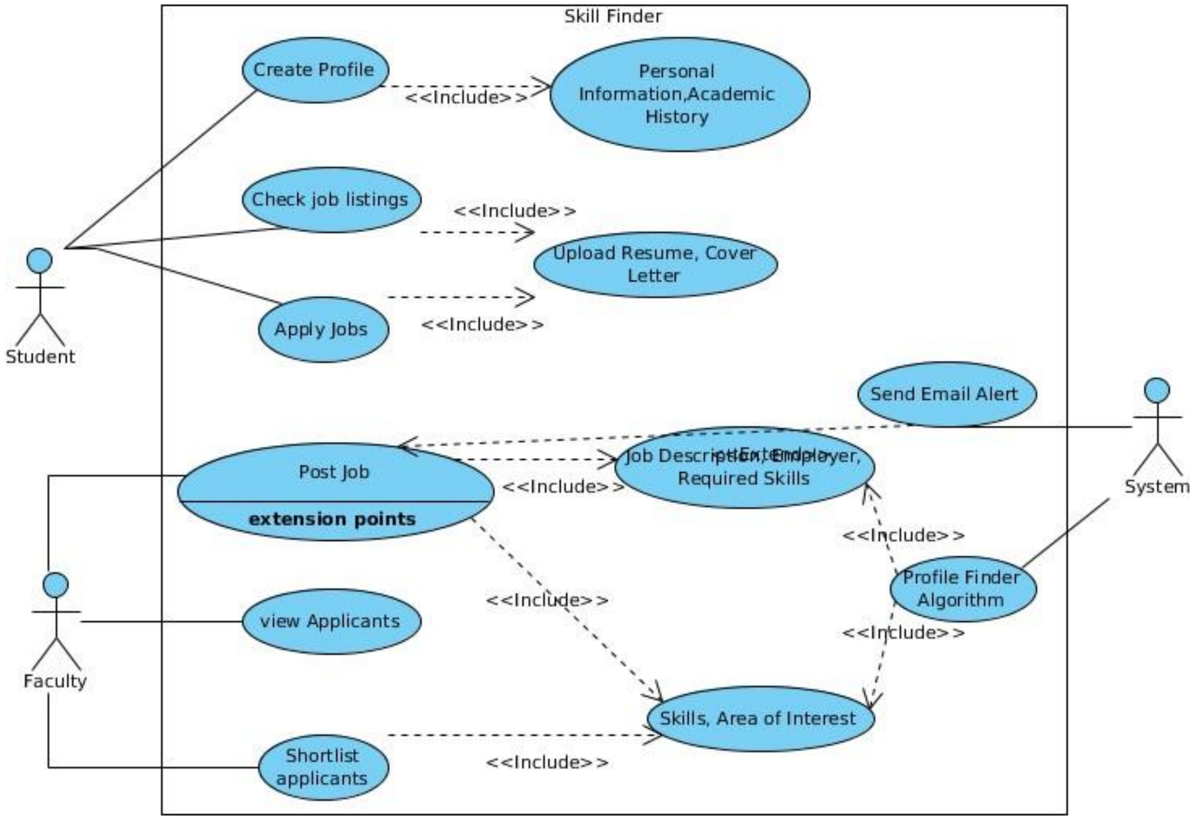
UML is a standard modeling **language**, not a **software development process**. UML 1.4.2 Specification explained that process:

- provides guidance as to the order of a team's activities,
- specifies what artifacts should be developed,
- directs the tasks of individual developers and the team as a whole, and
- offers criteria for monitoring and measuring a project's products and activities.

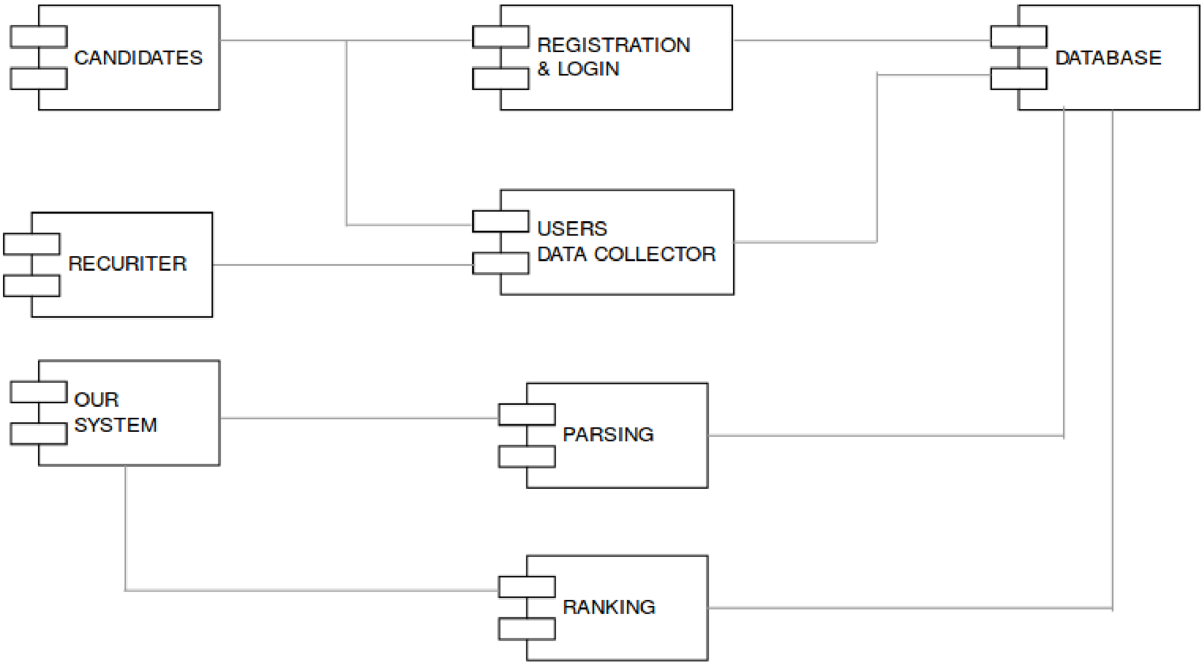
UML is intentionally **process independent** and could be applied in the context of different processes. Still, it is most suitable for use case driven, iterative and incremental development processes. An example of such process is **Rational Unified Process** (RUP).

UML is not complete and it is not completely visual. Given some UML diagram, we can't be sure to understand depicted part or behavior of the system from the diagram alone. Some information could be intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams.

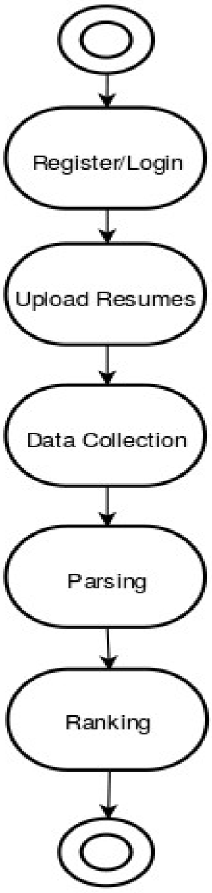
5.2.1 USE CASE DIAGRAM



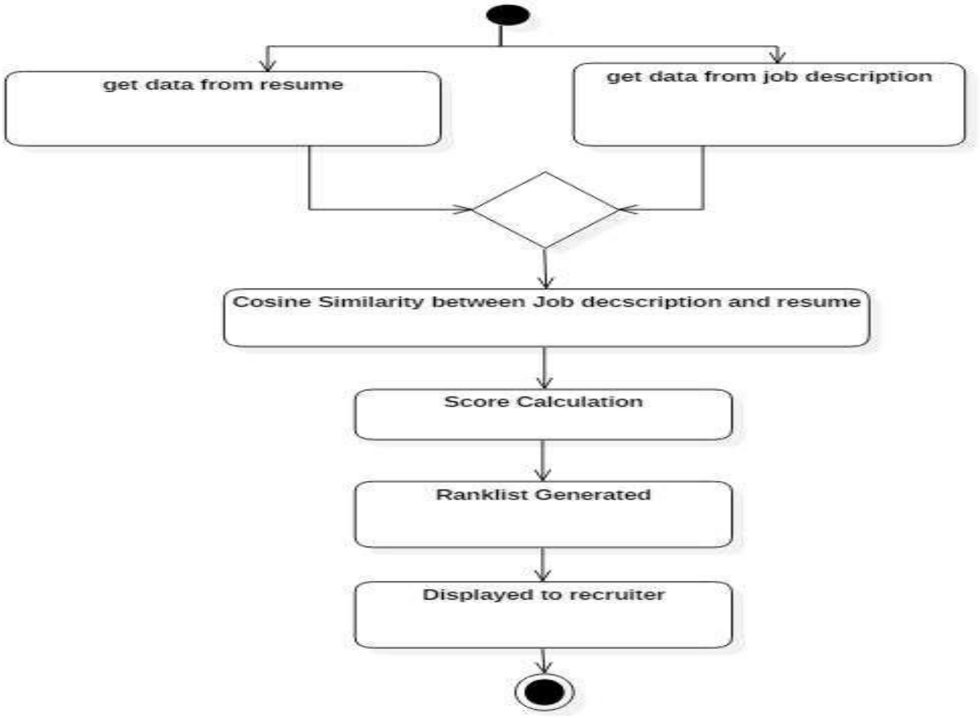
5.2.2 Component Diagram



5.2.3 State chart Diagram



5.2.4 Activity Diagram


get data from resume
get data from job description
Cosine Similarity between Job decscription and resume
Score Calculation
Ranklist Generated
Displayed to recruiter

6. Result Analysis

By displaying a resume list in order of relevance to the position, the technique ranks CVs according to their match with the job description, making it easy for recruiters. This would allow the recruiter to categorise the resumes according to the job requirements and quickly locate the CVs that best match the job description. The approach would aid the recruiter in expediting profile shortlisting while also ensuring the shortlisting process's authenticity, since they would be able to examine a large number of resumes in a short period of time, also with the proper fit, which a human would not be able to perform in near real time. This would help to make the process of recruiting individuals more efficient and successful in terms of selecting the best candidates. This would also assist the recruiter in reducing the time and resources required in locating the best candidates, making the process more costeffective

7. System Testing

7.1 Introduction

system Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. **System Testing** is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS). System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing. **System Testing is a black-box testing**.

7.2 Testing methodologies

7.2.1 Unit Testing

Unit Testing is a software testing technique by means of which individual units of software i.e. group of computer program modules, usage procedures, and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent module is tested to determine if there is an issue by the developer himself. It is correlated with the functional correctness of the independent modules. Unit Testing is defined as a type of software testing where individual components of a software are tested. Unit Testing of the software product is carried out during the development of an application. An individual component may be either an individual function or a procedure. Unit Testing is typically performed by the developer. In SDLC or V Model, Unit testing is the first level of testing done before integration testing. Unit testing is such a type of testing technique that is usually performed by developers. Although due to the reluctance of developers to test, quality assurance engineers also do unit testing.

7.2.1.1 Block Box Testing

Black box testing is a type of software testing in which the functionality of the software is not known. The testing is done without the internal knowledge of the products.

Black box testing can be done in the following ways:

1. Syntax Driven Testing – This type of testing is applied to systems that can be syntactically represented by some language. For example- compilers, language that can be represented by a context-free grammar. In this, the test cases are generated so that each grammar rule is used at least once.

2. Equivalence partitioning – It is often seen that many types of inputs work similarly so instead of giving all of them separately we can group them and test only one input of each group. The idea is to partition the input domain of the system into several equivalence classes such that each member of the class works similarly, i.e., if a test case in one class results in some error, other members of the class would also result in the same error.
The technique involves two steps:

1. **Identification of equivalence class –** Partition any input domain into a minimum of two sets: **valid values** and **invalid values**. For example, if the valid range is 0 to 100 then select one valid input like 49 and one invalids like 104.
2. **Generating test cases –** (i) To each valid and invalid class of input assigns a unique identification number. (ii) Write a test case covering all valid and invalid test cases considering that no two invalid inputs mask each other. To calculate the square root of a number, the equivalence classes will be:

3. **(a) Valid inputs:**
   - The whole number which is a perfect square- output will be an integer.
   - The whole number which is not a perfect square- output will be a decimal number.
   - Positive decimals
   - Negative numbers(integer or decimal).
   - Characters other that numbers like “a”,”!”,”;”,etc.

7.2.1.2 White Box Testing

White box testing techniques analyze the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

Working process of white box testing:

- **Input:** Requirements, Functional specifications, design documents, source code.
- **Processing:** Performing risk analysis for guiding through the entire process.
- **Proper test planning:** Designing test cases so as to cover entire code. Execute rinse-repeat until error-free software is reached. Also, the results are communicated.
- **Output:** Preparing final report of the entire testing process.

7.2.2 Integration Testing

Integration testing is the process of testing the interface between two software units or modules. It focuses on determining the correctness of the interface. The purpose of integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit tested, integration testing is performed.

Integration test approaches – There are four types of integration testing approaches. Those approaches are the following:

1. Big-Bang Integration Testing – It is the simplest integration testing approach, where all the modules are combined and the functionality is verified after the completion of individual module testing. In simple words, all the modules of the system are simply put together and tested. This approach is practicable only for very small systems. If an error is found during the integration testing, it is very difficult to localize the error as the error may potentially belong to any of the modules being integrated. So, debugging errors reported during big bang integration testing is very expensive to fix.

Advantages:

- It is convenient for small systems.

Disadvantages:

- There will be quite a lot of delay because you would have to wait for all the modules to be integrated.
- High risk critical modules are not isolated and tested on priority since all modules are tested at once.

2. Bottom-Up Integration Testing – In bottom-up testing, each module at lower levels is tested with higher modules until all modules are tested. The primary purpose of this integration testing is that each subsystem tests the interfaces among various modules making up the subsystem. This integration testing uses test drivers to drive and pass appropriate data to the lower level modules.

Advantages:

- In bottom-up testing, no stubs are required.
- A principle advantage of this integration testing is that several disjoint subsystems can be tested simultaneously.

Disadvantages:

- Driver modules must be produced.
- In this testing, the complexity that occurs when the system is made up of a large number of small subsystems.

3. Top-Down Integration Testing – Top-down integration testing technique is used in order to simulate the behavior of the lower-level modules that are not yet integrated. In this integration testing, testing takes place from top to bottom. First, high-level modules are tested and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended.

Advantages:

- Separately debugged module.
- Few or no drivers needed.
- It is more stable and accurate at the aggregate level.

Disadvantages:

- Needs many Stubs.
- Modules at lower level are tested inadequately.

7.2.3 Functional Testing

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. It focuses on simulation of actual system usage but does not develop any system structure assumptions.

It is basically defined as a type of testing which verifies that each function of the software application works in conformance with the requirement and specification. This testing is not concerned about the source code of the application. Each functionality of the software application is tested by providing appropriate test input, expecting the output and comparing the actual output with the expected output. This testing focuses on checking of user interface, APIs, database, security, client or server application and functionality of the Application Under Test.

- Functional testing can be manual or automated.

7.3 Validation

Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e. it checks what we are developing is the right product.

Activities involved in validation:

1. Black box testing
2. White box testing
3. Unit testing
4. Integration testing

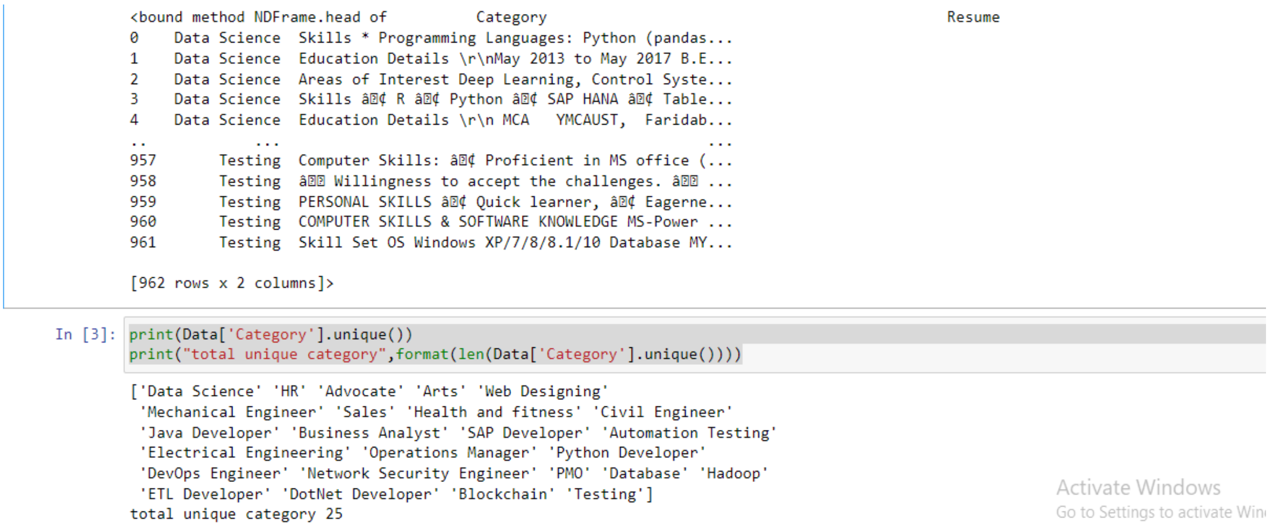
9. SAMPLE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
Data=pd.read_csv("C:/Users/nikhil/Documents/nikhil.csv")
print(Data.head)
print(Data['Category'].unique())
print("total unique category",format(len(Data['Category'].unique())))
print(Data['Category'].value_counts())
import seaborn as sns
plt.figure(figsize=(10,10))
sns.countplot(y='Category',data=Data)
plt.show()

plt.savefig('c:/Users/nikhil/Pictures/Saved Picturesfrequency.jpg')
from matplotlib.gridspec import GridSpec
count=Data['Category'].value_counts()
label=Data['Category'].value_counts().keys()
plt.figure(1,figsize=(25,25))
grid=GridSpec(2,2)
cmap=plt.get_cmap('Accent')
color=[cmap(i) for i in np.linspace(0,1,5)]
plt.subplot(grid[0,1],aspect=1,title="DISTRIBUTION")
pie=plt.pie(count,labels=label,autopct='%1.1f%%')
plt.show()
plt.savefig('C:/Users/nikhil/Documents/DISTRIBUTION.jpg')
import re
def clean(text):
    text=re.sub('http\s+\s*','',text)
    text=re.sub('RT|cc','',text)
    text=re.sub('#\s+','',text)
    text=re.sub('@s+','',text)
    text=re.sub('[%s]'%re.escape(""""!"#$%&'()*+,-./:;<=>?@[]^_`{|}~""""),'',text)
    text=re.sub('\s+','',text)
    text=re.sub(r'[^x00-x7f]',r'',text)
    text=re.sub('s+','',text)

    return text
Data['clean text']=Data.Resume.apply(lambda x: clean(x))
print(Data['clean text'])
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
import string
from wordcloud import WordCloud

oneSetOfStopWords = set(stopwords.words('english')+['``',"''"])
totalWords =[]
```

```
Sentences = Data['Resume'].values
cleanSentences = ""
for i in range(0,160):
    Text = clean(Sentences[i])
    cleanSentences += Text
    Words = nltk.word_tokenize(Text)
    for word in Words:
        if word not in oneSetOfStopWords and word not in string.punctuation:
            totalWords.append(word)

wordfreqdist = nltk.FreqDist(totalWords)
mostcommon = wordfreqdist.most_common(50)
print(mostcommon)
from sklearn.preprocessing import LabelEncoder

var_mod = ['Category']
le = LabelEncoder()
for i in var_mod:
    Data[i] = le.fit_transform(Data[i])
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from scipy.sparse import hstack

Text = Data['clean text'].values
Target = Data['Category'].values

word_vectorizer = TfidfVectorizer(
    sublinear_tf=True,
    stop_words='english',
    max_features=1500)
word_vectorizer.fit(Text)
WordFeatures = word_vectorizer.transform(Text)

print ("Feature completed .....")

X_train,X_test,y_train,y_test=train_test_split(WordFeatures,Target,random_state=0,test_size=0.)
print(X_train.shape)
print(X_test.shape)
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
clf = KNeighborsClassifier()
clf.fit(X_train, y_train)
prediction = clf.predict(X_test)
print('Accuracy of KNeighbors Classifier on training set:{:.2f}'.format(clf.score(X_train,
y_train)))
print('Accuracy of KNeighbors Classifier on test set: {:.2f}'.format(clf.score(X_test, y_test)))

print("\n Classification report for classifier %s:\n%s\n" % (clf,
metrics.classification_report(y_test, prediction)))
```

9.OUT PUT SCREENS

OUTPUT SCREENS

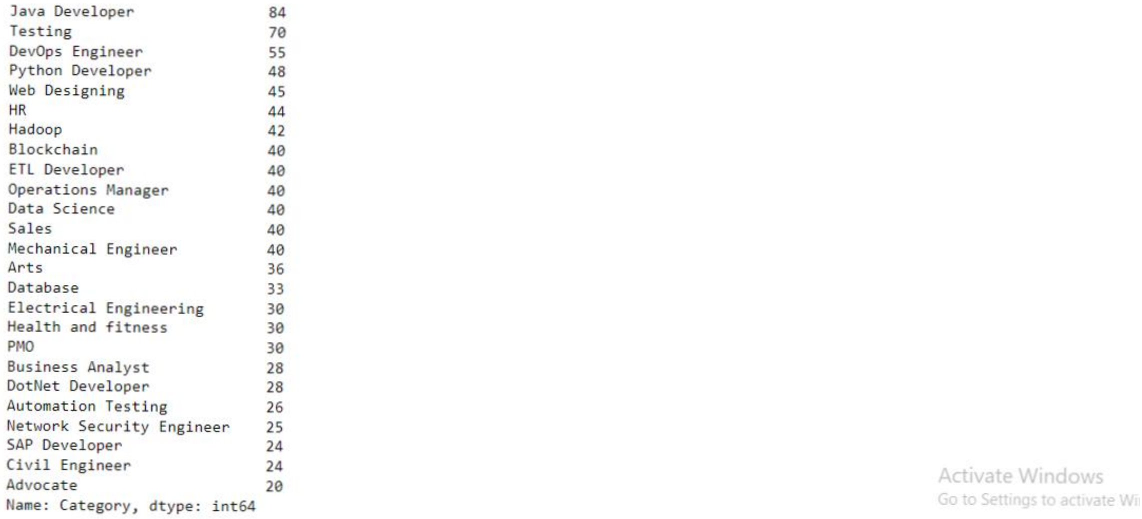


```
<bound method NDFrame.head of           Category                                             Resume
0    Data Science  Skills * Programming Languages: Python (pandas...
1    Data Science  Education Details \r\nMay 2013 to May 2017 B.E...
2    Data Science  Areas of Interest Deep Learning, Control Syste...
3    Data Science  Skills â▯¢ R â▯¢ Python â▯¢ SAP HANA â▯¢ Table...
4    Data Science  Education Details \r\n MCA   YMCAUST,  Faridab...
..            ...                                                ...
957       Testing  Computer Skills: â▯¢ Proficient in MS office (...
958       Testing  â▯▯ Willingness to accept the challenges. â▯▯ ...
959       Testing  PERSONAL SKILLS â▯¢ Quick learner, â▯¢ Eagerne...
960       Testing  COMPUTER SKILLS & SOFTWARE KNOWLEDGE MS-Power ...
961       Testing  Skill Set OS Windows XP/7/8/8.1/10 Database MY...

[962 rows x 2 columns]>
```

```
In [3]: print(Data['Category'].unique())
        print("total unique category",format(len(Data['Category'].unique())))
```

```
['Data Science' 'HR' 'Advocate' 'Arts' 'Web Designing'
 'Mechanical Engineer' 'Sales' 'Health and fitness' 'Civil Engineer'
 'Java Developer' 'Business Analyst' 'SAP Developer' 'Automation Testing'
 'Electrical Engineering' 'Operations Manager' 'Python Developer'
 'DevOps Engineer' 'Network Security Engineer' 'PMO' 'Database' 'Hadoop'
 'ETL Developer' 'DotNet Developer' 'Blockchain' 'Testing']
total unique category 25
```

Fig: displays the no. of categories and column and uniqueness in category



```
Java Developer               84
Testing                      70
DevOps Engineer              55
Python Developer             48
Web Designing                45
HR                           44
Hadoop                       42
Blockchain                   40
ETL Developer                40
Operations Manager           40
Data Science                 40
Sales                        40
Mechanical Engineer          40
Arts                         36
Database                     33
Electrical Engineering       30
Health and fitness           30
PMO                          30
Business Analyst             28
DotNet Developer             28
Automation Testing           26
Network Security Engineer    25
SAP Developer                24
Civil Engineer               24
Advocate                     20
Name: Category, dtype: int64
```

fig: No.of applicants in Each Job Category

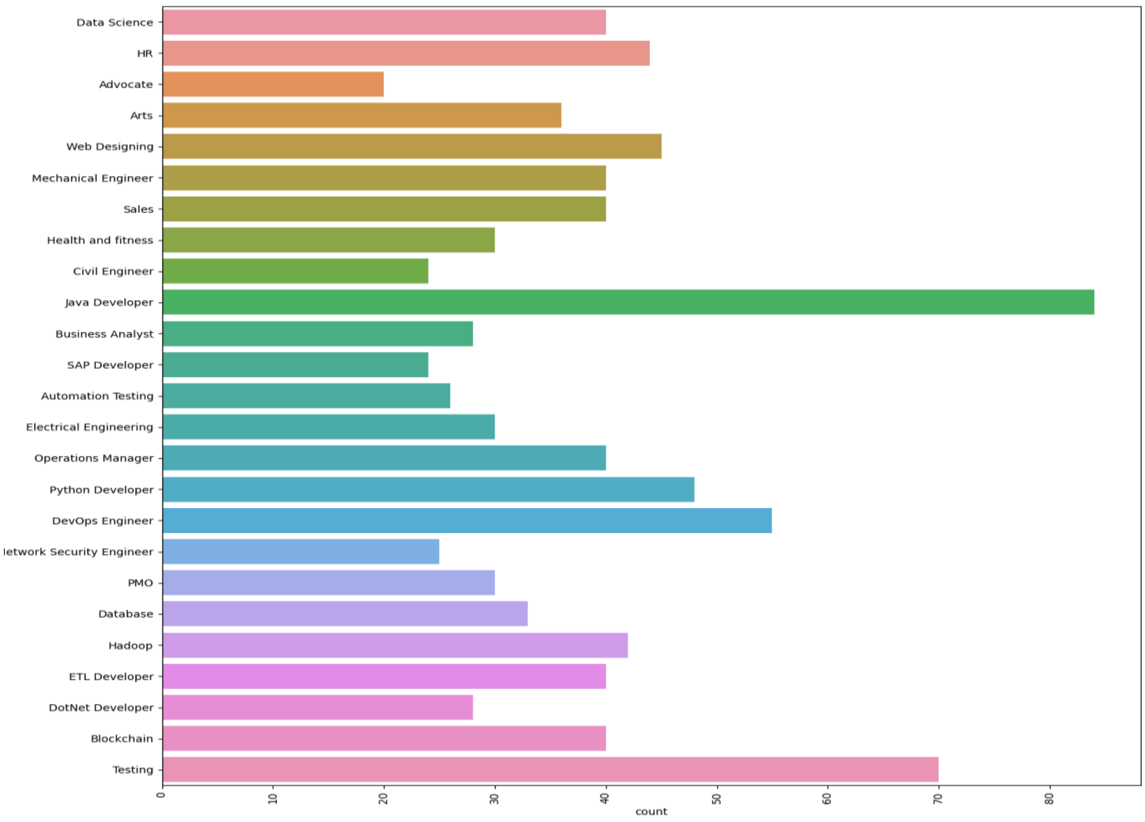
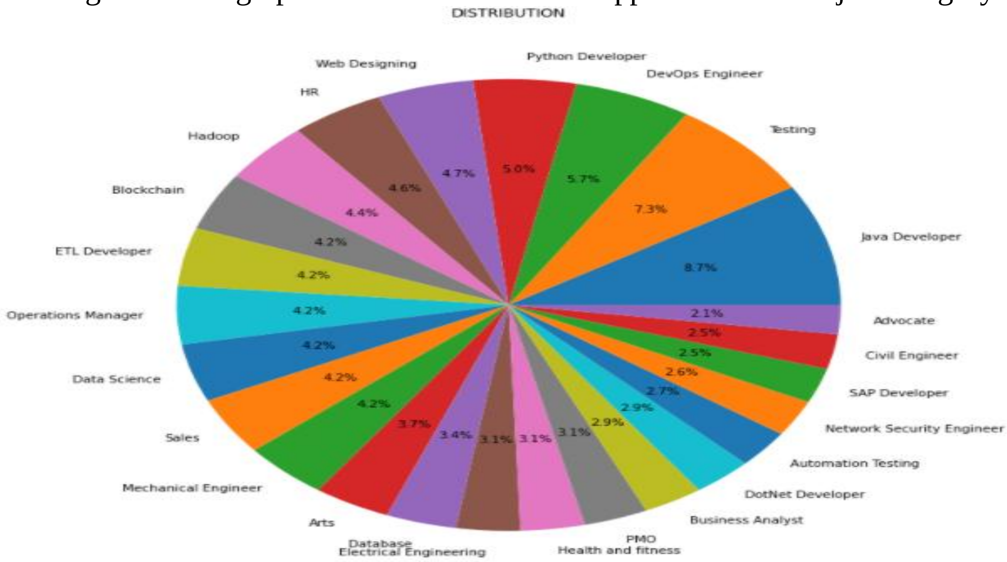
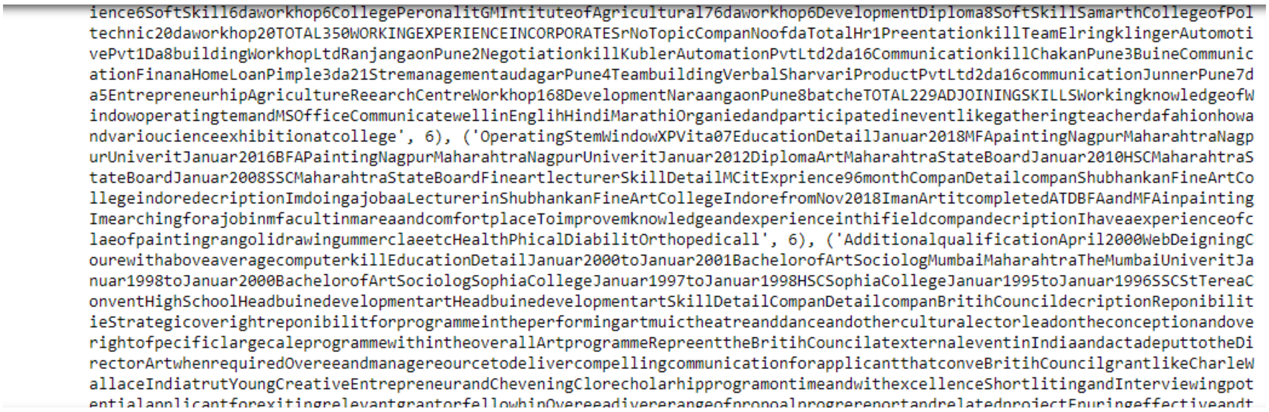


Fig: A Count graph to visualize the no. of applicants in each job category



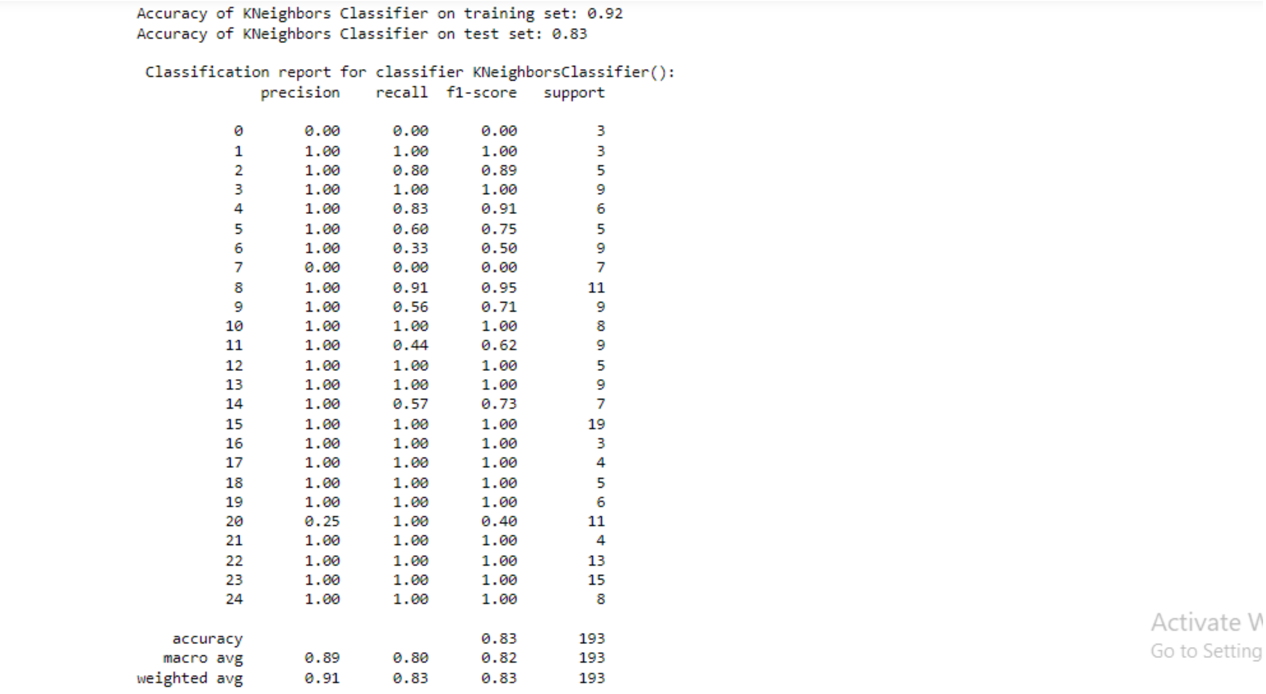
```
<Figure size 432x288 with 0 Axes>
```

Fig: pie chart to Display percentile applications each category



```
ience6SoftSkill6daworkhop6CollegePeronalitGMIntituteofAgricultural76daworkhop6DevelopmentDiploma8SoftSkillSamarthCollegeofPol
technic20daworkhop20TOTAL350WORKINGEXPERIENCEINCORPORATESrNoTopicCompanNoofdaTotalHr1PreentationkillTeamElringklingerAutomoti
vePvt1Da8buildingWorkhopLtdRanjangaonPune2NegotiationkillKublerAutomationPvtLtd2da16CommunicationkillChakanPune3BuineCommunic
ationFinanaHomeLoanPimple3da21StremanagementaudagarPune4TeambuildingVerbalSharvariProductPvtLtd2da16communicationJunnerPune7d
a5EntrepreneurhipAgricultureReearchCentreWorkhop168DevelopmentNaraangaonPune8batcheTOTAL229ADJOININGSKILLSWorkingknowledgeofW
indowoperatingtemandMSOfficeCommunicatewellinEnglihHindiMarathiOrganiedandparticipatedineventlikegatheringteacherdafahionhowa
ndvarioucienceexhibitionatcollege', 6), ('OperatingStemWindowXPVita07EducationDetailJanuar2018MFApaintingNagpurMaharahtraNagp
urUniveritJanuar2016BFAPaintingNagpurMaharahtraNagpurUniveritJanuar2012DiplomaArtMaharahtraStateBoardJanuar2010HSCMaharahtraS
tateBoardJanuar2008SSCMaharahtraStateBoardFineartlecturerSkillDetailMCitExprience96monthCompanDetailcompanShubhankanFineArtCo
llegeindoredecriptionImdoingajobaaLecturerinShubhankanFineArtCollegeIndorefromNov2018ImanArtitcompletedATDBFAandMFAinpainting
ImearchingforajobinmfacultinmareaandcomfortplaceToimprovemknowledgeandexperienceinthifieldcompandecriptionIhaveaexperienceofc
laeofpaintingrangolidrawingummerclaeetcHealthPhicalDiabilitOrthopedicall', 6), ('AdditionalqualificationApril2000WebDeigningC
ourewithaboveaveragecomputerkillEducationDetailJanuar2000toJanuar2001BachelorofArtSociologMumbaiMaharahtraTheMumbaiUniveritJa
nuar1998toJanuar2000BachelorofArtSociologSophiaCollegeJanuar1997toJanuar1998HSCSophiaCollegeJanuar1995toJanuar1996SSCStTereaC
onventHighSchoolHeadbuinedevelopmentartHeadbuinedevelopmentartSkillDetailCompanDetailcompanBritihCouncildecriptionReponibilit
ieStrategicoverightreponibilitforprogrammeintheperformingartmuictheatreanddanceandotherculturalectorleadontheconceptionandove
rightofpecificlargecaleprogrammewithintheoverallArtprogrammeRepreenttheBritihCouncilatexternaleventinIndiaandactadeputtotheDi
rectorArtwhenrequiredOvereeandmanagereourcetodelivercompellingcommunicationforapplicantthatconveBritihCouncilgrantlikeCharleW
allaceIndiatrutYoungCreativeEntrepreneurandCheveningClorecholarhipprogramontimeandwithexcellenceShortlitingandInterviewingpot
entialapplicantforexitingrelevantgrantorfellowhipOvereeadivererangeofpropoalprogrereportandrelatedprojectEnuringeffectiveandt
```

Fig: Most common words used in applications



```
Accuracy of KNeighbors Classifier on training set: 0.92
Accuracy of KNeighbors Classifier on test set: 0.83

 Classification report for classifier KNeighborsClassifier():
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       1.00      1.00      1.00         3
           2       1.00      0.80      0.89         5
           3       1.00      1.00      1.00         9
           4       1.00      0.83      0.91         6
           5       1.00      0.60      0.75         5
           6       1.00      0.33      0.50         9
           7       0.00      0.00      0.00         7
           8       1.00      0.91      0.95        11
           9       1.00      0.56      0.71         9
          10       1.00      1.00      1.00         8
          11       1.00      0.44      0.62         9
          12       1.00      1.00      1.00         5
          13       1.00      1.00      1.00         9
          14       1.00      0.57      0.73         7
          15       1.00      1.00      1.00        19
          16       1.00      1.00      1.00         3
          17       1.00      1.00      1.00         4
          18       1.00      1.00      1.00         5
          19       1.00      1.00      1.00         6
          20       0.25      1.00      0.40        11
          21       1.00      1.00      1.00         4
          22       1.00      1.00      1.00        13
          23       1.00      1.00      1.00        15
          24       1.00      1.00      1.00         8

    accuracy                           0.83       193
   macro avg       0.89      0.80      0.82       193
weighted avg       0.91      0.83      0.83       193
```

Fig: Result

Conclusion

In this paper, we presented an automated resume screening system that simplifies the e-recruitment process by eliminating the various problems faced by the recruiters as they relied on manual shortlisting of applicants for a given job position. Our system works on two fronts. Firstly, it uses Natural Language Processing to extract relevant information from the unstructured and wide-ranging formats of the resumes. It creates a summarized version of each resume which has only the entities that are pertinent to the selection process. With all the insignificant information removed, the task of the screening officials is simplified, and they can better analyses each resume with better efficiency. On the other front, our system provides the provision of ranking the applicants by using a content-based recommendation that uses the Vector Space Model and similarity to match the extracted resume features with the requirements in the job description. It calculates the similarity score value for each resume and thus creates a ranked list of top-N recommended candidates that best fit the particular job opening.

Future Enhancement

Future work for this system includes mining social networking data (e.g. Facebook, LinkedIn, GitHub profiles) of the candidates and utilizing this social behaviour information in combination with resume content to make even more improved recommendations. Another possibility is using a collaborative filtering based approach that can match the current applicant with a job according to how well other similar candidates (neighbours) are rated for it. Another scope of future work lies in the use of Latent Semantic Analysis (Berry, M., 2001) in the calculation of semantic similarity between the documents and then comparing it with the results of the term frequency based similarity approach

REFERENCES

[1] Al-Otaibi, S.T., Ykhlef, M., 2012. A survey of job recommender systems. International Journal of Physical Sciences 7, 5127–5142.

[2] Breaugh, J.A., 2009. The use of biodata for employee selection: Past research and future directions. Human Resource Management Review 19, 219–231.

[3] Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

[4] Carrer-Neto, W., Herná ndez-Alcaraz, M.L., Valencia-Garc ́ıa, R., Garc ́ıa-Sá nchez, F., 2012. Social knowledge-based recommender system. application to the movies domain. Expert Systems with applications 39, 10990–11000.

[5] Celma, O., 2010. Music recommendation, in: Music recommendation and discovery. Springer, pp. 43–85.

[6] Das, A.S., Datar, M., Garg, A., Rajaram, S., 2007. Google news personalization: scalable online collaborative filtering, in: Proceedings of the 16th international conference on World Wide Web, ACM. pp. 271–280.

[7] Diao, Q., Qiu, M., Wu, C.Y., Smola, A.J., Jiang, J., Wang, C., 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars), in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 193–202.

[8] Fa ̈rber, F., Weitzel, T., Keim, T., 2003. An automated recommendation approach to selection in personnel recruitment. AMCIS 2003 proceed- ings , 302.

[9] Golec, A., Kahya, E., 2007. A fuzzy model for competency-based employee evaluation and selection. Computers & Industrial Engineering 52, 143–161.

[10] Howard, J.L., Ferris, G.R., 1996. The employment interview context: Social and situational influences on interviewer decisions 1. Journal of applied social psychology 26, 112–136.

[11] Lin, Y., Lei, H., Addo, P.C., Li, X., 2016. Machine learned resume-job matching solution. arXiv preprint arXiv:1607.07657 , 1–8.

[12] Loper, E., Bird, S., 2002. Nltk: the natural language toolkit. arXiv preprint cs/0205028 .

[13] Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G., 2015. Recommender system application developments: a survey. Decision Support Systems 74, 12–32.

[14] Maheshwary, S., Misra, H., 2018. Matching resumes to jobs via deep siamese network, in: Companion Proceedings of the The Web Conference 2018, International World Wide Web Conferences Steering Committee. pp. 87–88.

[15] Malinowski, J., Keim, T., Wendt, O., Weitzel, T., 2006. Matching people and jobs: A bilateral recommendation approach, in: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS’06), IEEE. pp. 137c–137c.

[16] Mooney, R.J., Roy, L., 2000. Content-based book recommending using learning for text categorization, in: Proceedings of the fifth ACM conference on Digital libraries, ACM. pp. 195–204.

[17] Dataset download from kaggle link:http://www.kaggle.com/datasets/gauravduttakiit/resume-dataset/