

CLASSIFICATION AND RECOMMENDATION OF EXPERTS FROM GITHUB, GITLAB PLATFORMS

Project report submitted to the SASTRA Deemed to be University in

Partial fulfilment of the requirements for the award of the degree of

Master of Science in Computer Science

Submitted by

Menaka S K

221058016

June 2021



Department of Computer Science and Engineering

SRINIVASA RAMANUJAN CENTRE

KUMBAKONAM, TAMIL NADU, INDIA – 612001



Department of Computer Science and Engineering
SRINIVASA RAMANUJAN CENTRE
KUMBAKONAM - 612001

Bonafide Certificate

This is to certify that the project report titled “**CLASSIFICATION AND RECOMMENDATION OF EXPERTS FROM GITHUB, GITLAB PLATFORMS**” submitted in partial fulfilment of the requirements for the award of the degree of M.Sc., Computer Science to the Srinivasa Ramanujan Centre, SASTRA Deemed to be University, is a bonafide record of the work done by **Menaka S K** (Reg. No.221058016) during the final semester of the academic year 2020-21, in the **Srinivasa Ramanujan Centre**, under my supervision. This project report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

Signature of Project Supervisor

: 

Name with Affiliation

: S.Swaminathan, Assistant Professor II, Department of CSE.

Date

: 21.06.2021

Project Viva-voce held on 21.06.2021

Examiner 1

Examiner 2



Department of Computer Science and Engineering

SRINIVASA RAMANUJAN CENTRE

KUMBAKONAM - 612001

Declaration

I declare that the project report titled “**CLASSIFICATION AND RECOMMENDATION OF EXPERTS FROM GITHUB, GITLAB PLATFORMS**” submitted by me is an original work done by me under the guidance of Prof S.Swaminathan, Assistant professor II, Department of CSE, Srinivasa Ramanujan Centre, SASTRA Deemed to be University during the final semester of the academic year 2020-21, in the Srinivasa Ramanujan Centre. The work is original and wherever I have used materials from other sources, I have given due credit and cited them in the text of the report. This project report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

Signature of the candidate

:

Name of the candidate

:

Menaka S.K

Date

:

21.06.2021

ACKNOWLEDGEMENT

I pay my sincere pranam to **God ALMIGHTY** for his grace and infinite mercy and for showing on me his choicest blessings.

First, I would like to express my sincere thanks to our honorable Chancellor **Prof.R.Sethuraman**, Vice-Chancellor **Dr.S.Vaidhyasubramaniam** and Registrar **Dr.R.Chandramouli** for giving an opportunity to be a student of this esteemed institution.

I express my deepest thanks to **Dr. V. Ramaswamy**, Dean and **Dr. A. Alli Rani** Associate Dean, Srinivasa Ramanujan Centre, for their moral support and suggestions when required without any reservations.

I exhibit my gratitude to **Dr. S. Sivagurunathan**, Head of Department, Computer Science and Engineering, Srinivasa Ramanujan Centre, for his constant support and valuable suggestion for completion of the project.

I also take this opportunity to express our deep sense of gratitude to my project supervisor **S.Swaminathan**, Assistant Professor II, Department of CSE, for his cordial support, valuable information and meticulous guidance which enabled me to complete this project successfully.

I would like to place on record to the project coordinator **Dr.V.Kalaichelvi**, Senior Assistant Professor, Department of CSE for her benevolent approach and pain taking efforts. I owe my sincere thanks to all faculty members in the department who have directly or indirectly helped me in completing this project.

Without the support of our parents and friends this project would never have become a reality. I owe my sincere thanks to all of them.

I dedicate this work to all my well-wishers, with love and affection.

Table of Contents

Chapter No.	Contents	Page No.
1	Introduction	1
	Problem Statement	2
	Existing Methodology	2
	Proposed Methodology	2
	Motivation	3
2	Objective	4
3	Experimental Works / Methodology	6
4	Result and Discussion	26
5	Conclusion and Further Work	31
6	Reference	33
7	Appendix	35

List of Figures

Figure No.	Title	Page No.
3.1.1	Home Page	9
3.1.2	Signup Page	9
3.1.3	Login Page for recruiter	10
3.1.4	Recruiter Page	10
3.1.5	Search for candidates for respective position	11
3.1.6	Recommendation of candidates	11
3.1.7	Recruiter sending mail to candidate	12
3.1.8	Mail Received (confirmation)	12
3.2.1	Candidate's Login Page	13
3.2.2	Candidate Filling the resume	13
3.2.4	Candidate Position Recommendation	14
3.3	Random Forest Classifier workflow	15
3.4	Voting average	16
3.4.2	Under-fitting and Over-fitting	25

List of Tables

Table No.	Table name	Page No.
1	Important Metrics Random Forest Classifier	17
2	Important Metrics Naive Bayes Classifier	18
3	Distribution of developers among various technical roles	23
4	Confusion Matrix	27
5	Error Rate	29
6	Accuracy Rate	29

ABSTRACT

Abstract

The world is blooming with huge technologies and this made everyone in the universe work smart, as there is a saying that ‘Necessity is the Mother of Invention’ according to the quote, necessity involved in mankind is hugely responsible for all those inventions of new technologies. In the same way, Machine Learning comes into the picture in which the machine mimics humans to some extension. This project is based on suppressing the risks faced by many of the recruiters during the interview process, in which every time the recruiters have to scan every resume with their naked eye and will ask questions related to some technical domains in which the candidate may or may not be familiar. To avoid this kind of misconception and focusing on the candidates experienced part make the candidate who approaching for the interview more confident. In order to achieve this, some new approaches have been involved in this project. The main objective is to collect some of the candidate details, from that information to predict which field does the candidate is experienced with more hands-on projects according to the prediction result the respective candidate may get the interview call from the recruiter in a specific domain and the candidate can attend the interview very confidently with the help of their previous projects.

CHAPTER 1
INTRODUCTION

1.1 Introduction

Today's newer software development concentrates over the high degree of technical specializations. These requirements make some companies to focus on emerging cross-functional members, such as mobile app developers, frontend, backend, full-stack developers, etc. The recruiters usually expect expert developers and sometimes it is hard to recruit the developers under such constraints. Some of the social platforms help the recruiters to easily analyze that the person is suitable for the specified role with a strong belief by visualizing the projects worked by the candidate. To identify the technical roles the machine learning algorithms like the Random Forest classifier, Naive Bayes are used which classifies the roles of developers based on multiple required features.

1.2 Problem Statement

The common problem involved in recruitment is to analyze the developers who excel in technical platforms and it not easy to analyze it with the help of a resume. It can be conquered with the use of open-source platforms like GitLab, GitHub, Bit Bucket, etc. It is an effortless task for the recruiters to make a transparent exploration of the global developers. The approach of machine learning makes the system predict the accurate results than by humans.

1.3 Existing Methodology

The way of filtering the candidates to which position they are belonging to, scanning the entire resume with the naked eye, not at all easy and so the new approach is being proposed from the candidates who has some level of experience in hands-on projects thus the candidate will be quite strong at the field to which they are interested with. Thus by saving time as well get rid of misconception of the candidates and assigning them to various field of work in which they are not satisfied to work.

1.4 Proposed Methodology

The process of scanning the resume every time with a naked eye and identifying that which position does someone will be recommended to fit is not an easy task to handle by the recruiters. In this project, the candidate will be allowed to fill the GUI resume, and then based on the input skills they are classified under five categories namely, Frontend Developer, Backend Developer, Mobile App Developer, Full Stack Developer, and Data Scientist. The recruiters may search for a

particular category and a list of matched candidates with that particular category will be recommended.

1.5 Motivation

This project helps the recruiter to pick up the developers in the easiest way, Random forest classifier is used to classify the candidates from the five categories such as Frontend Developer, Backend Developer, Mobile App Developer, Full-stack Developer, and Data Scientist. The recruiters may simply do register themselves and are allowed to search for a candidate from the GUI (Graphical User Interface) and after clicking on the search button the matched candidates or developers who are already submitted their resume will be automatically recommended to the candidates and the recruiters can view the candidate's skills, roles, project URL and so on. These details raise a thought to the recruiters to select the most expert developers from the recommended candidates.

CHAPTER 2

OBJECTIVE

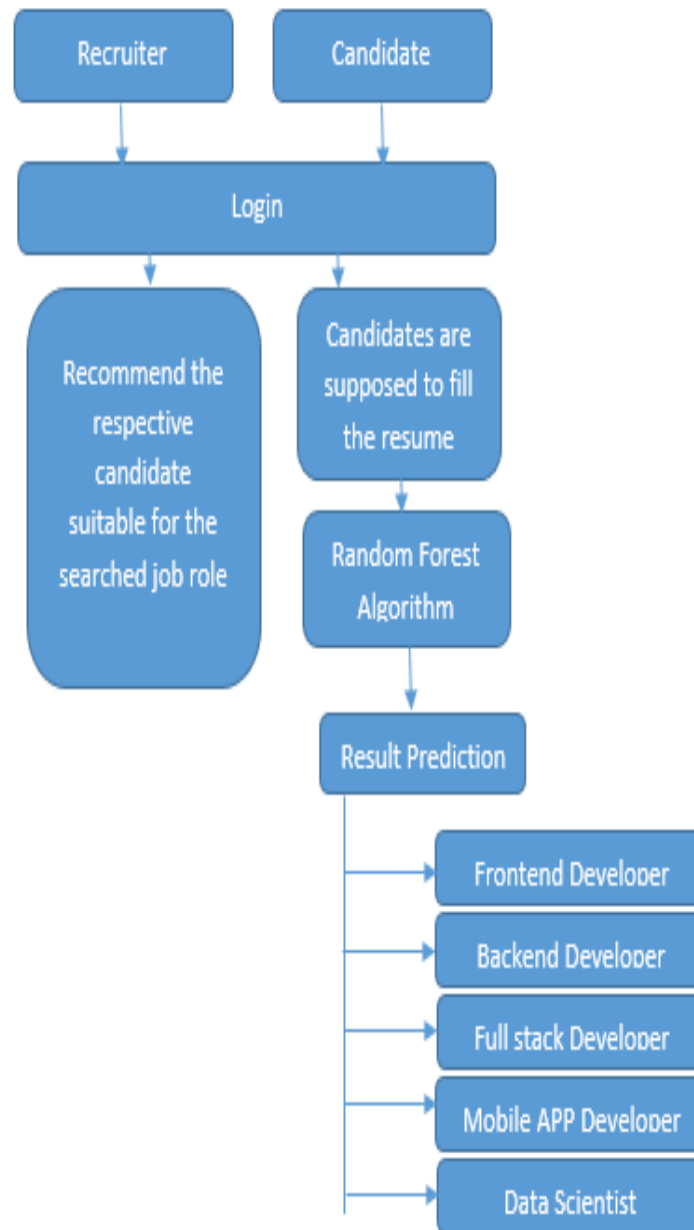
2. Objective

The technique involved here is a classification-based approach a couple of commonly used classification algorithms are going to be used in this project by comparing with both of its accuracy levels the relevant algorithm is to be considered for future predictions. The concept of machine learning plays a vital role in predicting accurate results by various machine learning techniques. The classification technique involved here able to classify the candidates whom resumes are submitted into five distinct categories namely, Frontend Developer, Backend Developer, Full-stack Developer, Mobile App Developer, and Data Scientist. These predicted candidates are recommended to the recruiters if they are trying to search for the candidates from the list box. The correctly matched candidates with job roles are viewed by the recruiters and they may be allowed to send mail to the candidates to appear for an interview in the future. This concept makes the company to less train the candidates in the respective because they already have hands-on experience in involved in some projects.

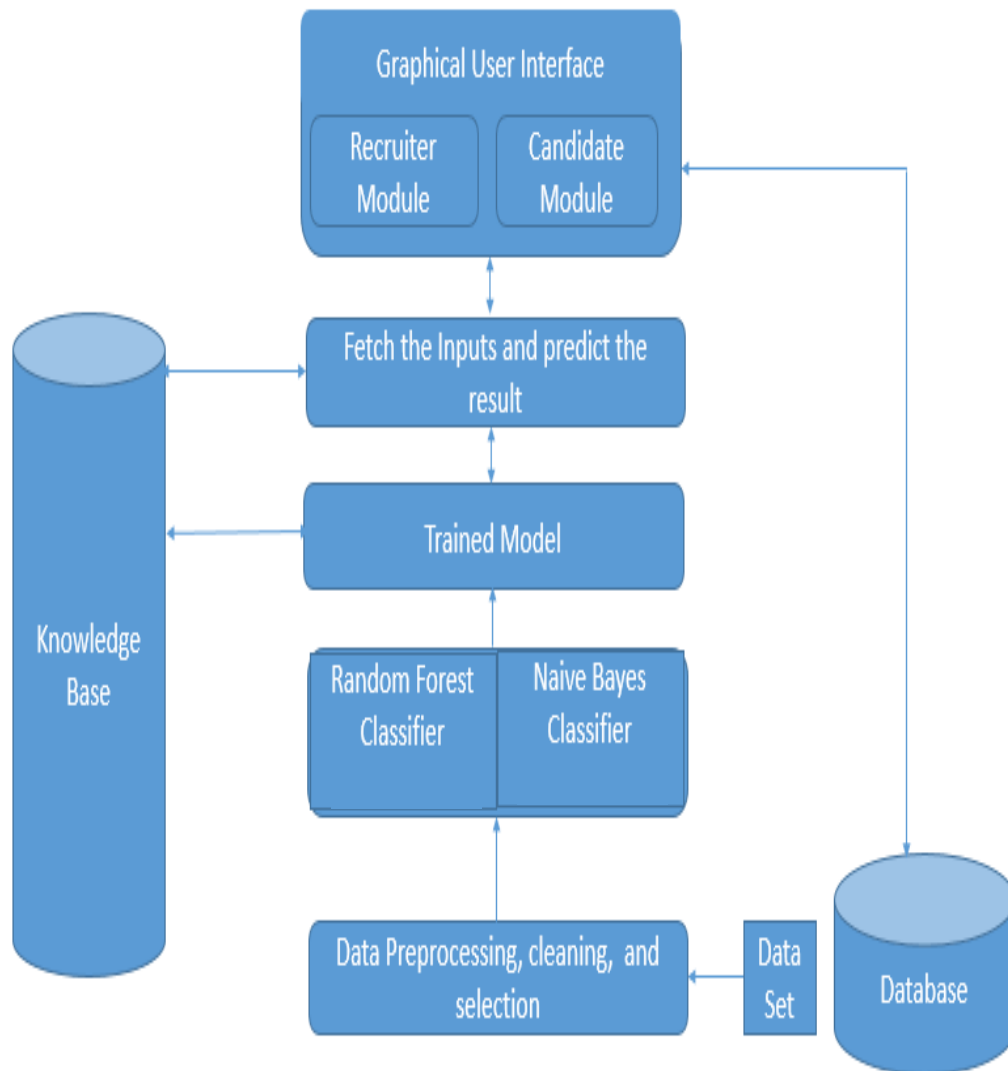
CHAPTER 3
EXPERIMENTAL WORKS / METHODOLOGY

3. Experimental Work

3.a Flowchart Diagram



3.b Architecture Diagram



3.1 Experimental Works:

3.1.1 Home Page

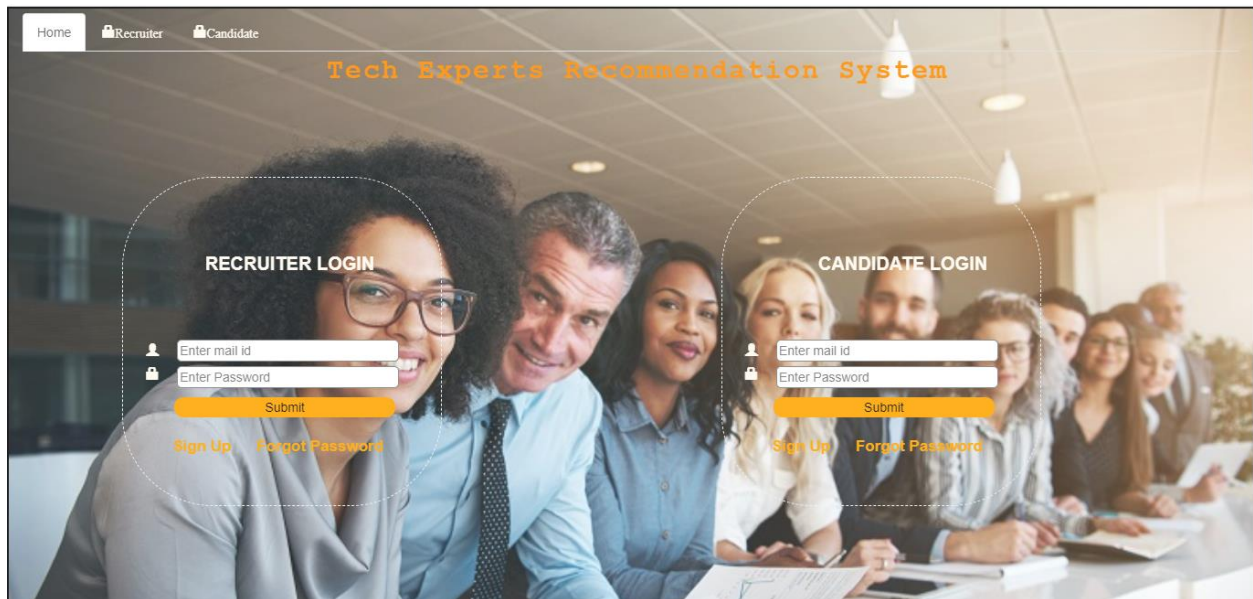


Fig 3.1.1 Home_Page

3.1.2 Signup Page

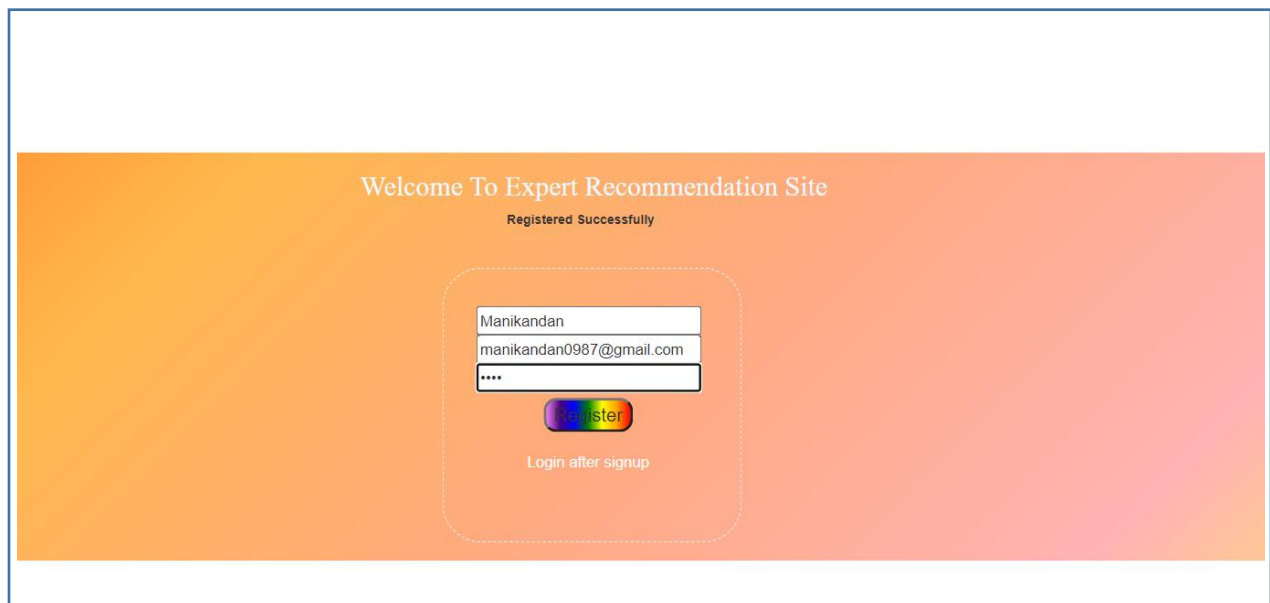


Fig 3.1.2 Signup_Page

3.1.3 Login Page for Recruiter



Fig 3.1.3 Recruiter Login

3.1.4 Recruiter's Page



Fig 3.1.4 Recruiter Page

3.1.5 Recruiter Searches for the candidate

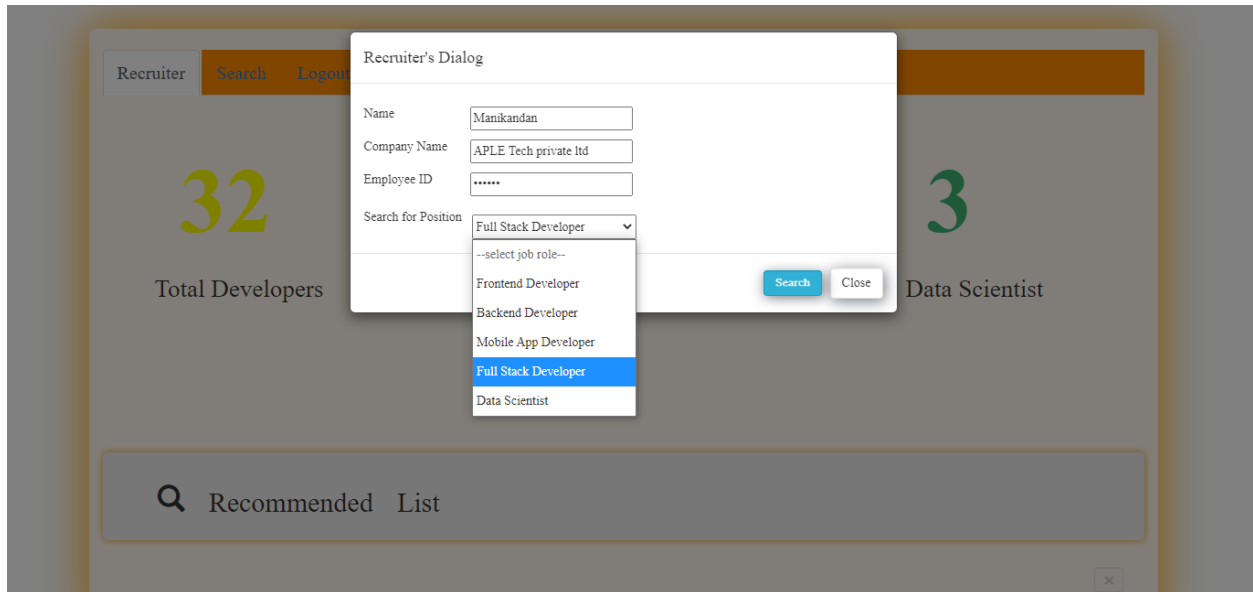


Fig 3.1.5 Recruiter searching for candidates on respective position

3.1.6 Recommendations for the Recruiter

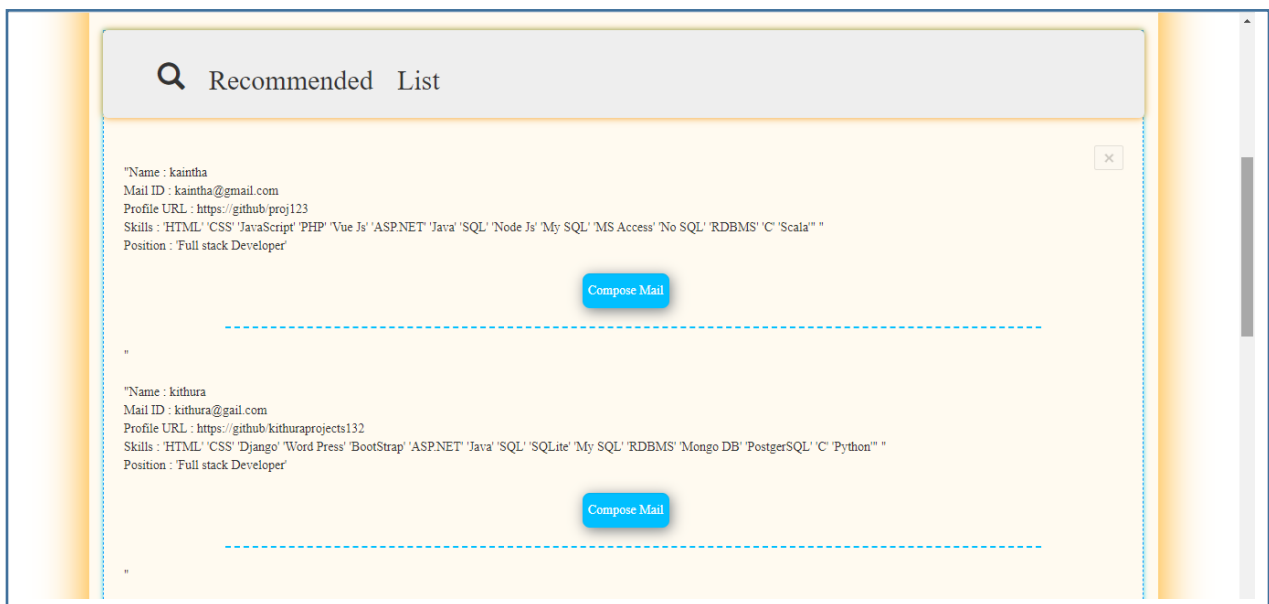


Fig 3.1.6 Recommending Candidates to recruiters

3.1.7 Recruiter sending mail to the candidate

Send Mails To Reach The Candidates

To:

From:

Password:

Mark the Time:

Mark the Date:


Subject:

Content:

3.1.7 Recruiter sending mail to candidate

3.1.8 Mail Received

Regarding Job Interview Inbox x

 **221058016@sastra.ac.in** to me 10:31 PM (0 minutes ago) ☆ ↶ ⋮

Hi Menaka, We are intrested to appoint you a interview on April 28th 2021 for the position Full stack Developer.
Thank you

Fig 3.1.8 Mail Received

3.2.1 Candidate's Login

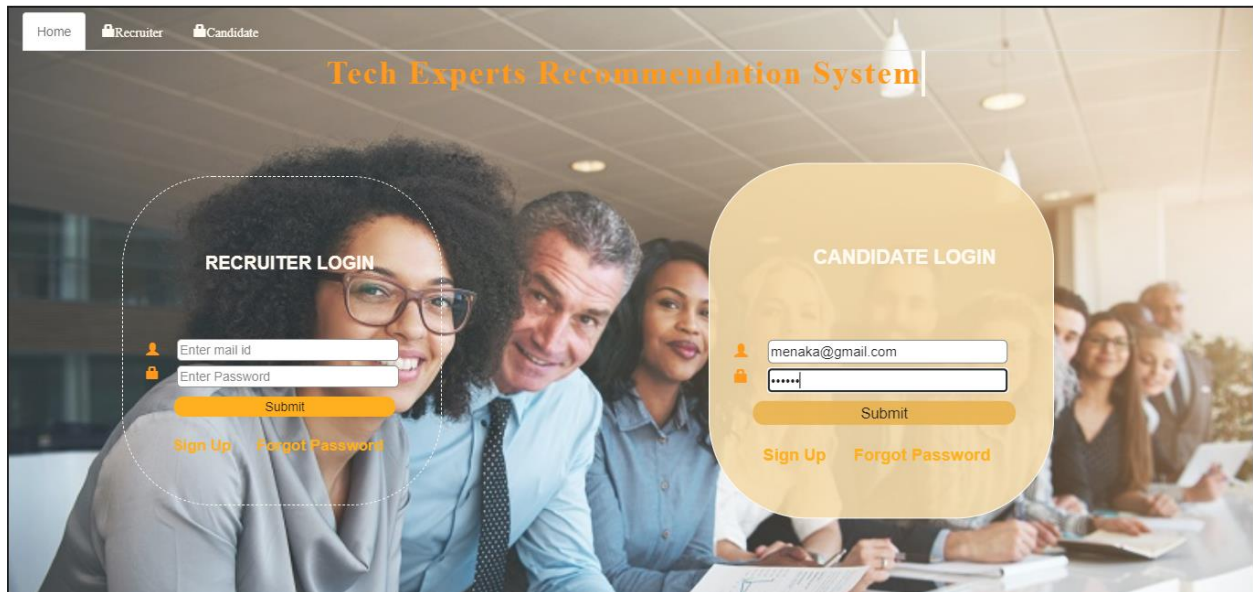


Fig 3.2.1 Candidate's Login

3.2.1 Candidate filling the resume

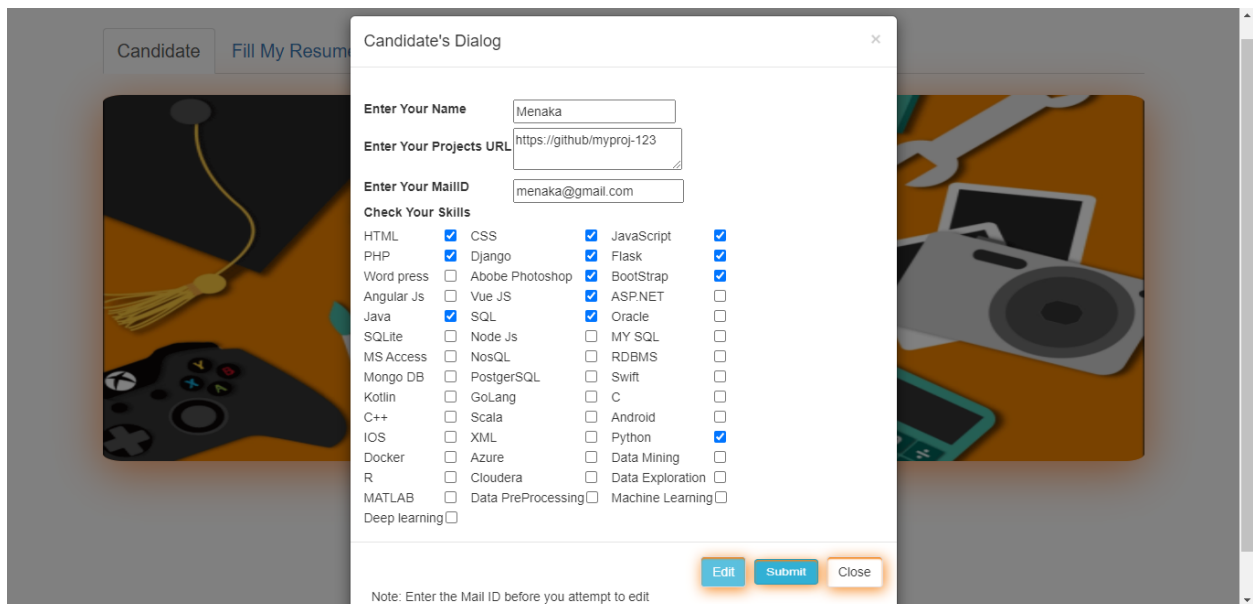


Fig 3.2.2 Filling the Resume

3.2.3 Candidate click on after Resume completion

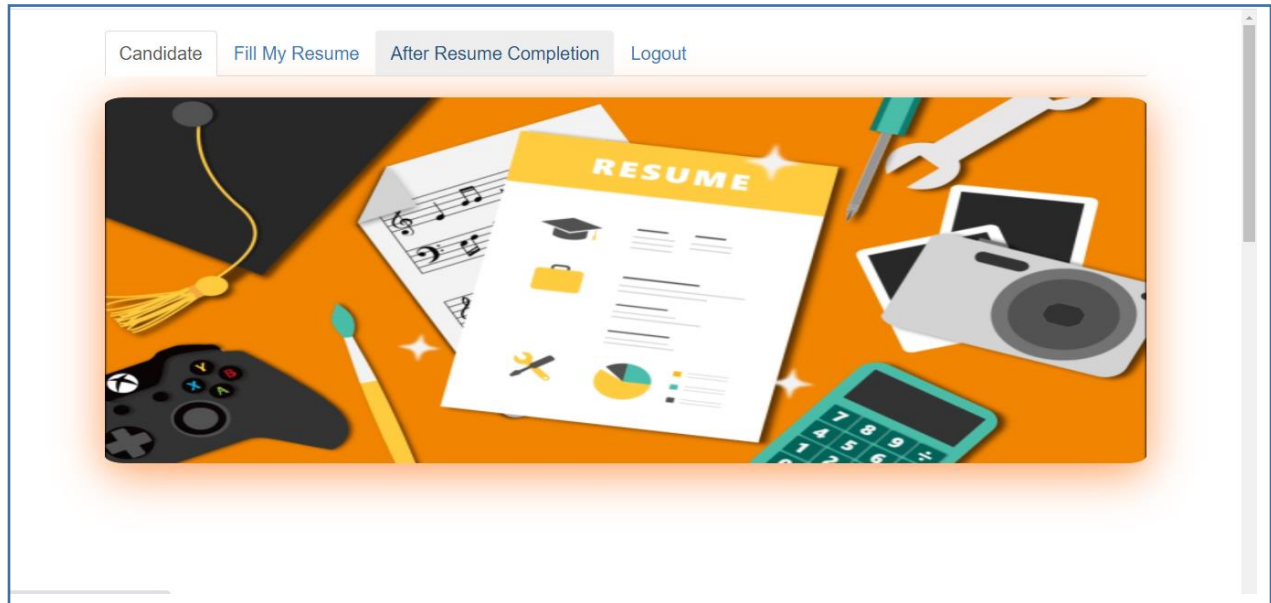


Fig 3.2.3 After Resume submission tab

3.2.4 Candidate Recommended with the position

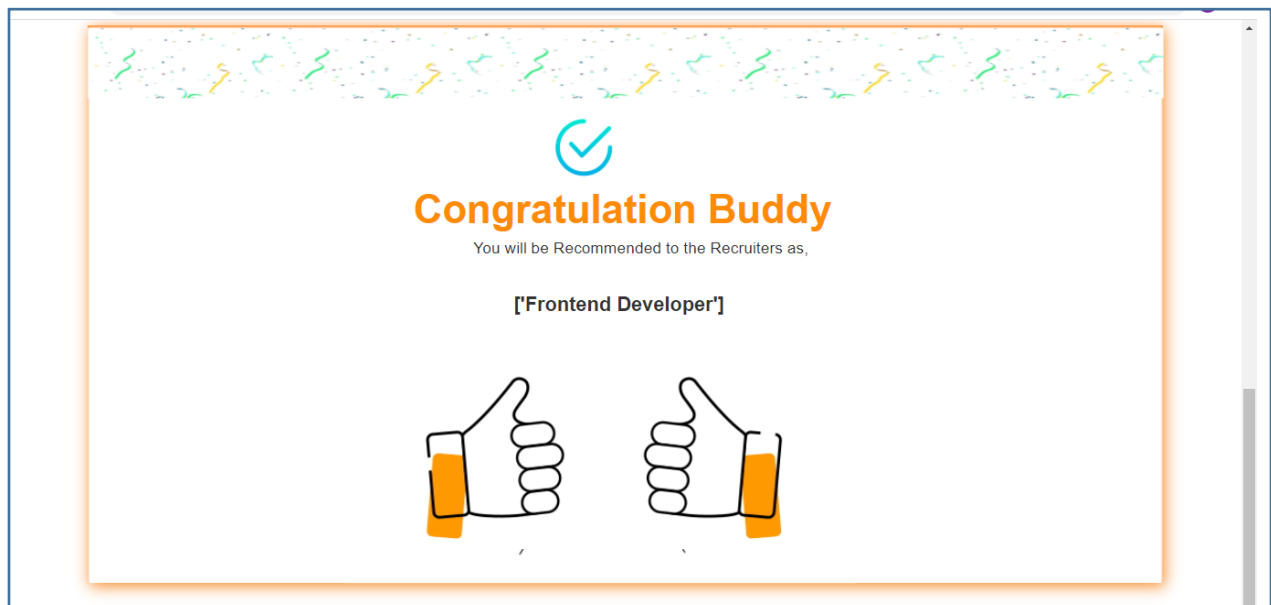


Fig 3.2.4 Candidate Recommended with the job roles

3.3 Methodology:

This main aim is to recommend the experts from the GitHub and GitLab platforms, this concept is achieved through the implementation of Random Forest Classifier and Naïve Bayes Classifier algorithms.

3.3 a) Random Forest Algorithm

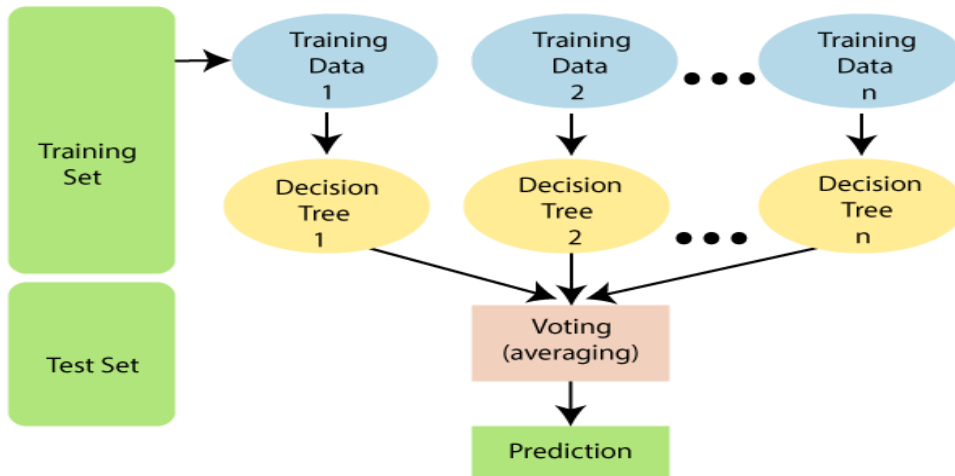


Fig 3.3 Random Forest Classifier Workflow

This algorithm is a collection of multiple decision trees that carry part of the training dataset for validation and then take the remaining part of the test dataset for result evaluation. The results are predicted in such a way that it will take each decision tree's results and then predict the final classification result by averaging the results of the decision trees.

3.3.1 How Random Forest Classifier works?

The Random Forest algorithm works on the basis of numerous decision trees and this decision tree yield to a result that may be a binary class or multiple class labels and this number of predicted results are further filtered by the technique called voting average in which the majority of the votes of the predicted class is considered as our model output.

Voting Average:

The process of model's prediction is based on various decision trees predictions which is further filtered by using **voting average** technique to predict the relevant class as the outcome and it is shown in Fig 1.1.

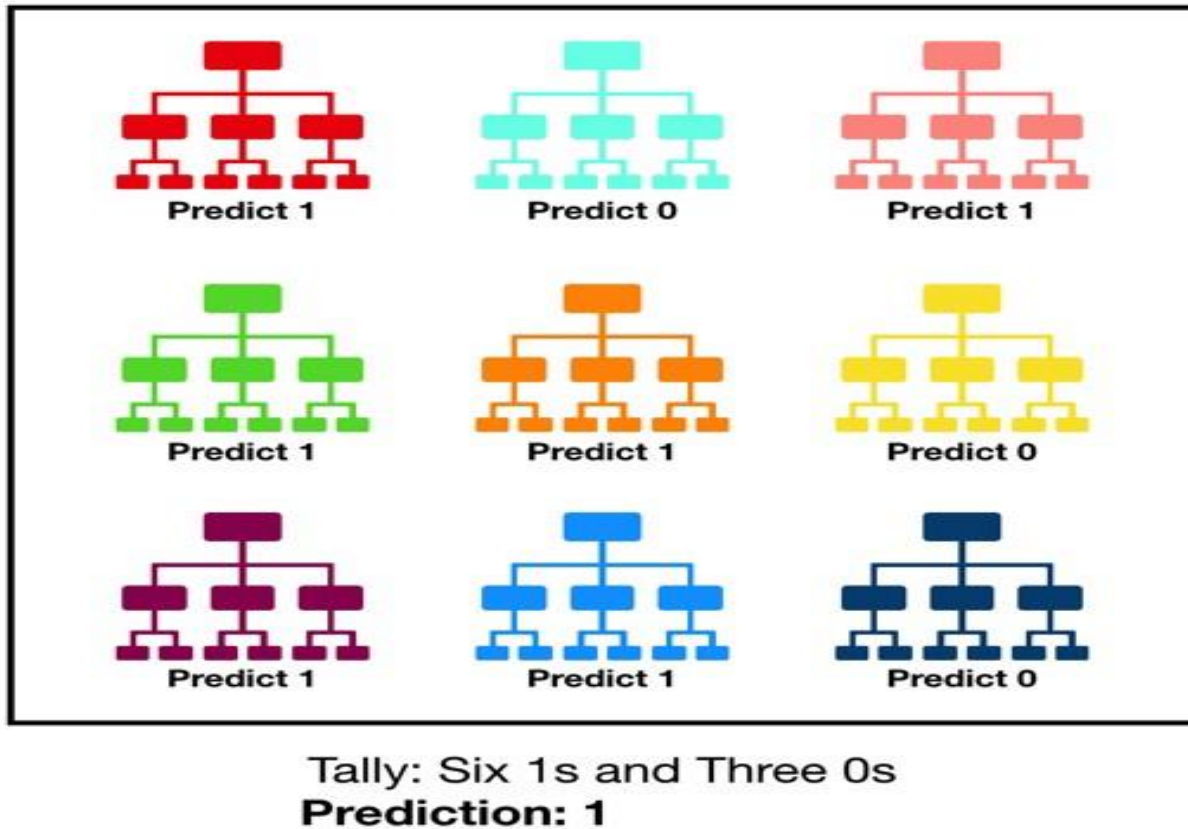


Fig: 3.3 Voting average

The important metrics like precision, recall, and F1-score are measured by the performance of the algorithm, it can be calculated from the formulae that are discussed in the conclusion.

The classification report is given below for Random Forest Classifier,

Important Metrics obtained from the algorithm.	
Prediction	0.9567
Recall	0.9711
F1 Score	0.9615

Table 1. Important metrics for Random Forest Classifier

3.3 b) Naive Bayes algorithm:

The second type of algorithm used in this project is Naive Bayes which is also a kind of machine learning classifier algorithm that is used for humongous volumes of data, even if someone is working with data that has a large amount of data records the best way is to use Naive Bayes. Naive Bayes is based on the concept called Bayes_theorem.

Bayes_Theorem:

The Bayesian algorithm works on conditional_probability. The conditional_probability something that give raise to the already appeared instances. The conditional_probability can give rise the probability of an incident or the event using its prior knowledge that are available.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

The diagram illustrates the components of Bayes' Theorem with arrows pointing from text descriptions to the formula:

- The probability of A being true "given that" B is true.** (points to $P(A|B)$)
- The probability of B being true "given that" A is true.** (points to $P(B|A)$)
- The probability of A being true.** (points to $P(A)$)
- The probability of B being true.** (points to $P(B)$)

3.3.2.1 Bayesian Algorithm:

Step 1 - Train the classifier with training features and labeled featured data.

Step 2 - Compute the probability $\text{prob}(i)$ either by using intuition or based on experts opinion or histogram based estimation.

Step 3 - Compute $\text{prob}(i/x)$ where i is the target label and x are the features to be passed to the model to train.

Step 4 - Find the maximum $\text{prob}(i/x)$ and assign the unknown instance to that class.

The classification report is given below for Naive Bayes Classifier

Important Metrics obtained from the algorithm.	
Prediction	0.6730
Recall	0.68311
F1 Score	0.6731

Table 2. Important metrics for Naive Bayes classifier

3.4 Data Collection Methods:

Basically, two methods are used to collect the data from the open-source platforms.

- **A)** Collecting the list of users of open source platforms from the Stack Exchange Data Explorer (SEDE)
- SEDE is a tool that is publicly used by numerous people in the world that allows the users to query data that are available in the Stack Exchange Data Explorer.
- **B)** Manual search for every skill sets acquired by the users and entering it into the dataset

Method 1:

To collect the list of users of the category Data Scientist in the Stack Exchange Data Explorer Platform (SEDE).

Step 1: Go to Stack Exchange Data Explorer website.

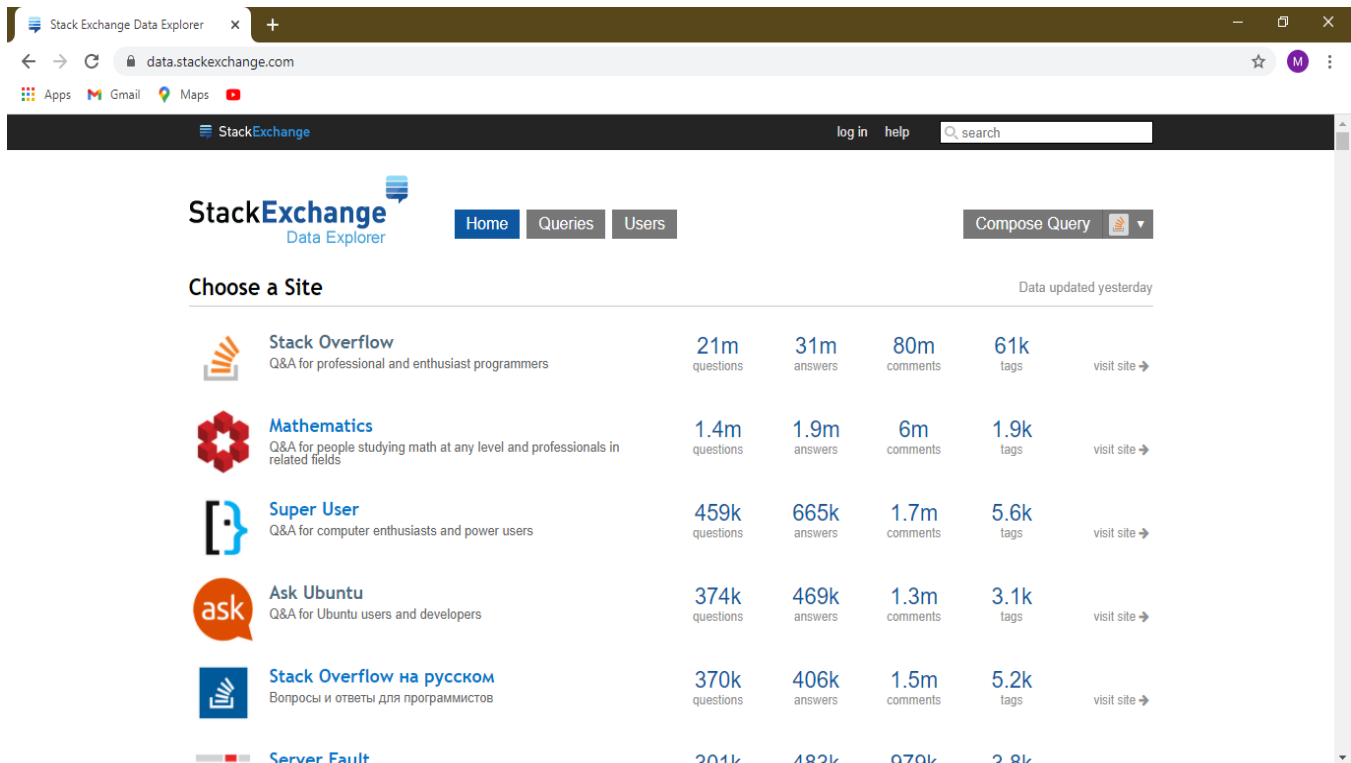


Fig 2. SEDE

Step 2: Click on the Data Scientist Profile

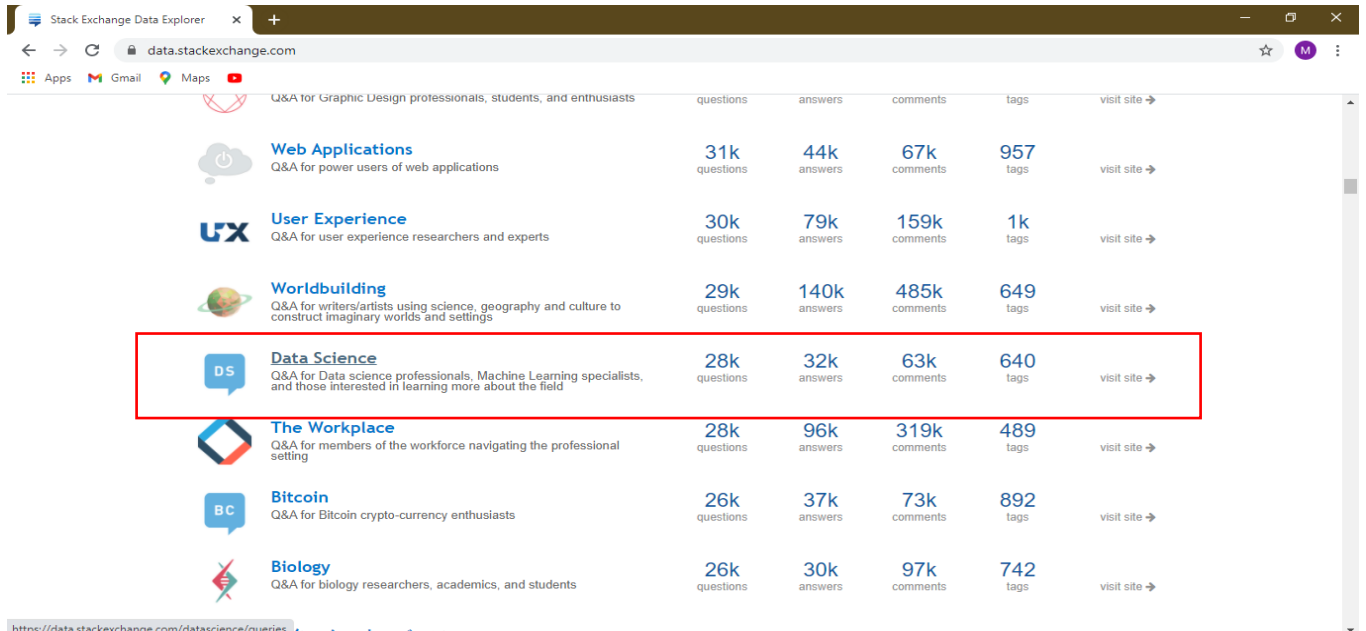


Fig 3. Selecting the Role

The next step is to click on the compose query option and the required SQL query to get exact details of the users who had a prior knowledge of using GITHUB and GITLAB resources.

Step 3: Querying the list of data scientist.

The screenshot shows a web browser window with the URL `data.stackexchange.com/datascience/query/new`. The page has a header with the "Data Science" logo and a description: "Q&A for Data science professionals, Machine Learning specialists, and those interested in learning more about the field". Below the header, there is a form to "Enter a title for your query" and a link to "edit description". The main content area is divided into two parts: a query editor on the left and a database schema on the right.

The query editor contains the following SQL query:

```
1 select * from users where websiteurl like '%gitlab%';
```

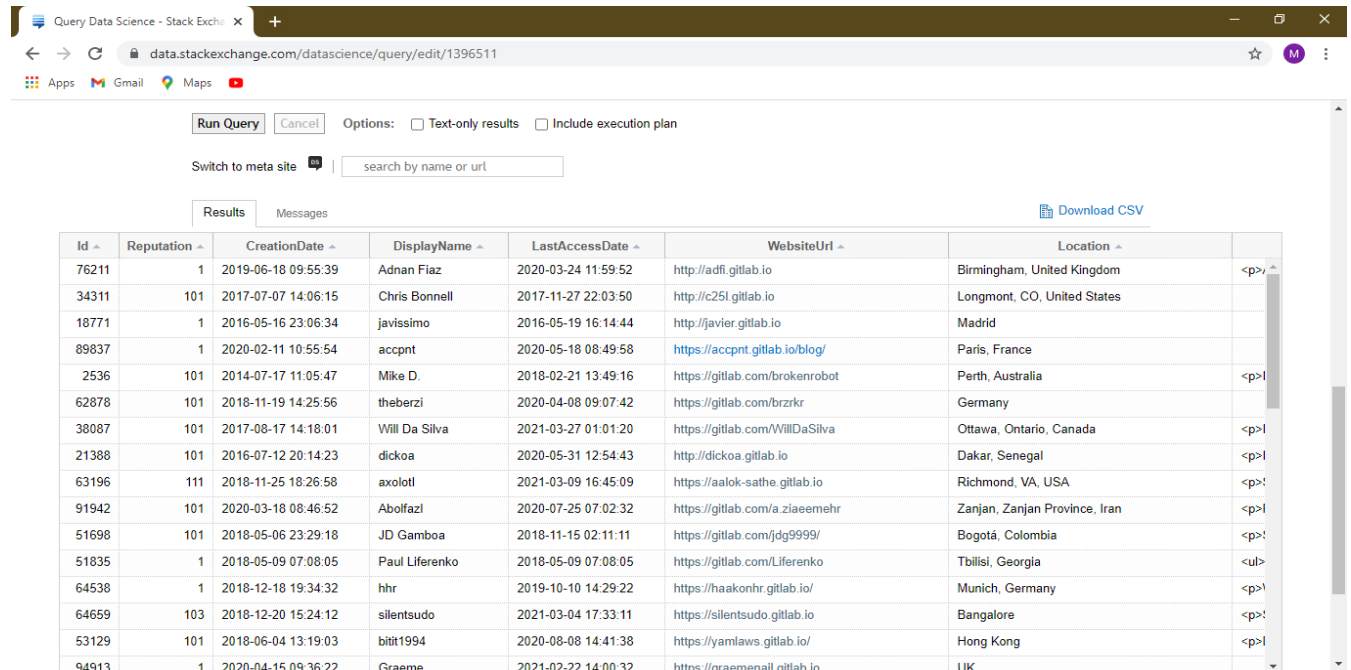
The database schema on the right lists the following fields and their data types:

Field	Data Type
Text	nvarchar (max)
ContentLicense	varchar (12)
PostHistoryTypes	
Id	tinyint
Name	nvarchar (50)
PostLinks	
Id	int
CreationDate	datetime
PostId	int
RelatedPostId	int
LinkTypeId	tinyint

Below the schema, there is a section for "Revisions" with the text "Waiting for you to make your first edit..." and a shaded area for editing.

Fig 5. Querying in SEDE

Step 4: Download the CSV file.



The screenshot shows a web browser window with the URL `data.stackexchange.com/datascience/query/edit/1396511`. The page displays a table of user data with columns: Id, Reputation, CreationDate, DisplayName, LastAccessDate, WebsiteUrl, and Location. A 'Download CSV' link is visible in the top right corner of the table area.

Id	Reputation	CreationDate	DisplayName	LastAccessDate	WebsiteUrl	Location
76211	1	2019-06-18 09:55:39	Adnan Fiaz	2020-03-24 11:59:52	http://adfi.gitlab.io	Birmingham, United Kingdom
34311	101	2017-07-07 14:06:15	Chris Bonnell	2017-11-27 22:03:50	http://c25i.gitlab.io	Longmont, CO, United States
18771	1	2016-05-16 23:06:34	javissimo	2016-05-19 16:14:44	http://javier.gitlab.io	Madrid
89837	1	2020-02-11 10:55:54	accpnt	2020-05-18 08:49:58	https://accpnt.gitlab.io/blog/	Paris, France
2536	101	2014-07-17 11:05:47	Mike D.	2018-02-21 13:49:16	https://gitlab.com/brokenrobot	Perth, Australia
62878	101	2018-11-19 14:25:56	theberzi	2020-04-08 09:07:42	https://gitlab.com/bzrkkr	Germany
38087	101	2017-08-17 14:18:01	Will Da Silva	2021-03-27 01:01:20	https://gitlab.com/WillDaSilva	Ottawa, Ontario, Canada
21388	101	2016-07-12 20:14:23	clckoa	2020-05-31 12:54:43	http://clckoa.gitlab.io	Dakar, Senegal
63196	111	2018-11-25 18:26:58	axolotl	2021-03-09 16:45:09	https://aalok-sathe.gitlab.io	Richmond, VA, USA
91942	101	2020-03-18 08:46:52	Abolfazl	2020-07-25 07:02:32	https://gitlab.com/a.ziaemehr	Zanjan, Zanjan Province, Iran
51698	101	2018-05-06 23:29:18	JD Gamboa	2018-11-15 02:11:11	https://gitlab.com/jdg9999/	Bogotá, Colombia
51835	1	2018-05-09 07:08:05	Paul Liferenko	2018-05-09 07:08:05	https://gitlab.com/Liferenko	Tbilisi, Georgia
64538	1	2018-12-18 19:34:32	hhr	2019-10-10 14:29:22	https://haakonhr.gitlab.io/	Munich, Germany
64659	103	2018-12-20 15:24:12	silentsudo	2021-03-04 17:33:11	https://silentsudo.gitlab.io	Bangalore
53129	101	2018-06-04 13:19:03	bitit1994	2020-08-08 14:41:38	https://iyamlaws.gitlab.io/	Hong Kong
94913	1	2020-04-15 09:36:22	Graeme	2021-02-22 14:00:32	https://graemenail.gitlab.io	UK

Fig 6. Download the CSV

Download the CSV file by clicking on the top right link. The complete CSV file is downloaded into the system.

Method 2:

The collected CSV file contains the URL of their own profile which helps to extract the users details from the open source platform.

Id	Reputation	CreationDate	DisplayName	LastAccessDate	WebsiteUrl	Location
76211	1	2019-06-18 09:55:39	Adnan Fiaz	2020-03-24 11:59:52	http://adfi.gitlab.io	Birmingham, United Kingdom

By using the WebsiteUrl column go to the relevant page in which collect the user details like which field they are really interested in (HTML, CSS, Js, Android application Etc.). Use this details to predict the user relevant job roles to be recommended for the recruiters.

3.4.1 Data Cleaning and Pre-Processing

In this technique all the data are cleaned, invalid data are eliminated, null values are replaced with modes of the respective column, for instance, the column name like NoSQL, RDBMS, PostgreSQL, Swift, C, C++, Android, Windows, Python, and MATLAB has nulls its value is replaced by applying fill with the null method.

As a results, the size entire dataset is 1041 x 52, where 1041 represents the number of rows and 52 represents the number of columns respectively.

Distributions of developers among the analyzed technical roles.

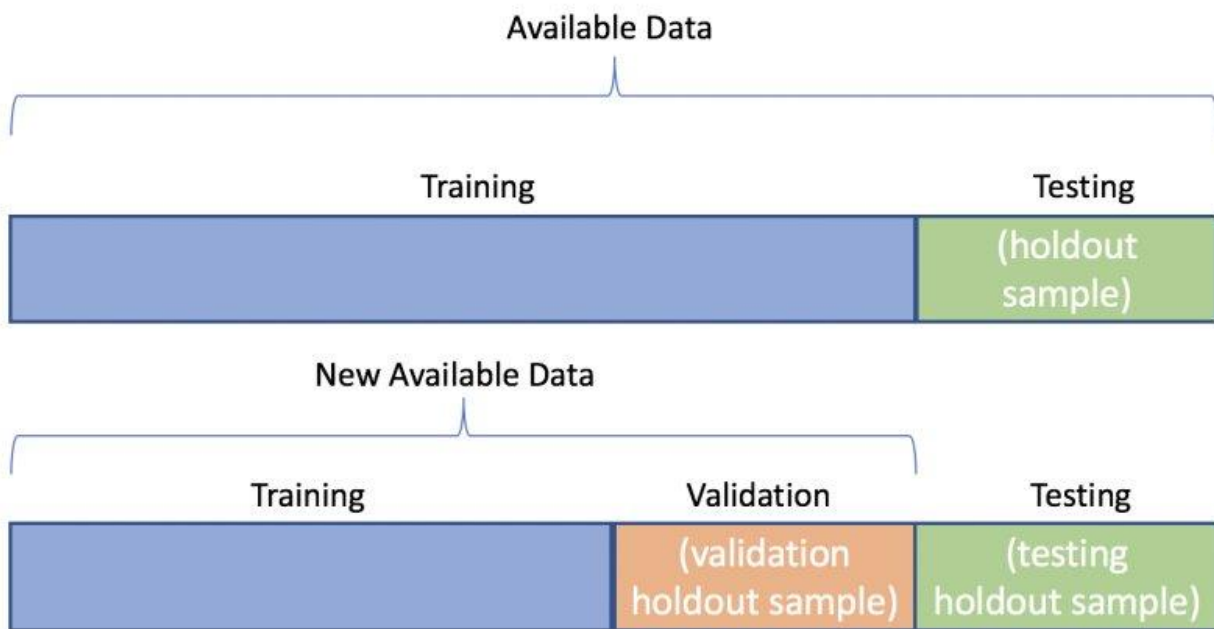
Frontend Developer	Backend Developer	Full-stack Developer	Mobile-App Developer	Data-Scientist
YES				
	YES			
YES	YES	YES		
			YES	
				YES
YES			YES	
YES				YES
	YES		YES	
	YES			YES
			YES	YES

Table 3. Distribution of developers among various technical roles

3.4.2 Train and Test dataset splitting method

The dataset is partitioned into two parts, one for training and the other for testing the results. As a result, the training dataset contains 80% of the dataset i.e.) 832 records and the remaining 20% is for the test dataset i.e.) 208 records.

3.4.2.1 How to split the dataset into training and test datasets?



3.4.2.2 How to perform training and testing in the dataset?

To perform use the follow the steps given below,

Step 1 - Import the dataset using the pandas library in python.

Step 2 - Analyse the dataset and eliminate errors, clean the dataset, etc

Step 3 - From the sklearn library import the necessary packages like model_selection for implementing train_test_split class.

Step 4 – Allocate 80% of the dataset for training and 20% for testing and implement it with the respective classification algorithms.

The state of under-fitting and over-fitting is a condition which exponentially give rise to the change in bias and the variation in variance the model.

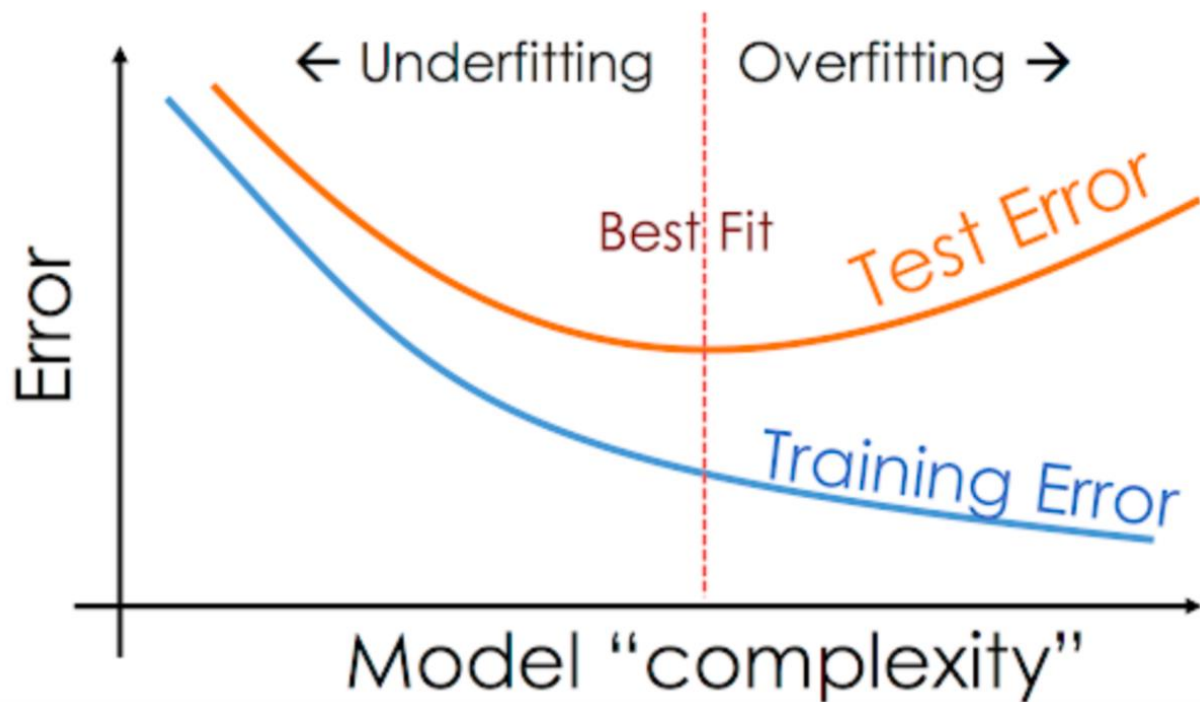


Fig.3.4.2 under-fitting and Over-fitting

Under-fitting:

In this under-fitting condition the model cant predict the inputs with newer values, that always points to an under-fitting stage of the model, which give rise to the accuracy level below average.

Over-fitting:

In this over-fitting condition the model cant predict the correct result while some newer type of inputs are given to the model and leads to accuracy of 100% which is not satisfactory.

CHAPTER 4

RESULT AND DISCUSSION

4. Result and Discussion

The respective Precision, Recall, and F1 Score are provided in Table 2 and the result candidates usually collected as shown in the figure and the front-end. The Random Forest Classifier is predicted with 97% accuracy and then it is the algorithm processed to predict with future inputs and the Precision, Recall, and F1 Score is shown in Table 3, and the Naive Bayes Classifier shown with 87% accuracy. As a result, the Random Forest algorithm predicts the developer's position from their skills this reduces the time consumption by the recruiters during their hiring process.

The back-end is a MySQL database that consists of four tables namely to store the login details, signup details, email details, recruiter who searching for the candidate, and the resume of the candidates. The front-end parts are HTML, CSS, and JavaScript, and the algorithms are implemented with python and then the API (Application Programming Interface) created using Flask framework to execute the project.

Let us consider the confusion matrix,

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Table 4. Confusion Matrix

4.1 Precision:

The Precision determines how precise or accurate our model is, it means that out of those results which are predicted as positive which of them are then positive. It is a good measure to find out when the cost of False-Positive is high. For instance, the developer may not be an expert in Frontend (actual Positive) who posses minimum skills as a frontend developer but our model is predicted as a Frontend developer.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$= \frac{\text{True Positive}}{\text{Total Predicted Positive}}$$

4.2 Recall:

The recall is a measure of how many actual positive our model captures as true positive. For instance, the candidate is a Frontend developer but the model actually predicts it as not a frontend developer.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$= \frac{\text{True Positive}}{\text{Total Actual Positive}}$$

4.3 F1 Score:

F1 score is used to maintain the balance between precision and recall. It is a better measure to use if it is needed to seek a balance between precision and recall.

$$\text{F1} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The Precision, Recall and F1- Score is given in Table 1 and Table 2.

4.4 Error Rate:

The inaccuracy of the predicted result is said to be the error. The target value is categorical and so the error is said to be the error rate. This is the proportion of cases where the predicted results are wrong.

$$\text{Error Rate} = 1 - \text{accuracy rate}$$

Random Forest Classifier	Naive Bayes Classifier
0.03%	0.14%

Table 5. Error Rate

4.5 Accuracy:

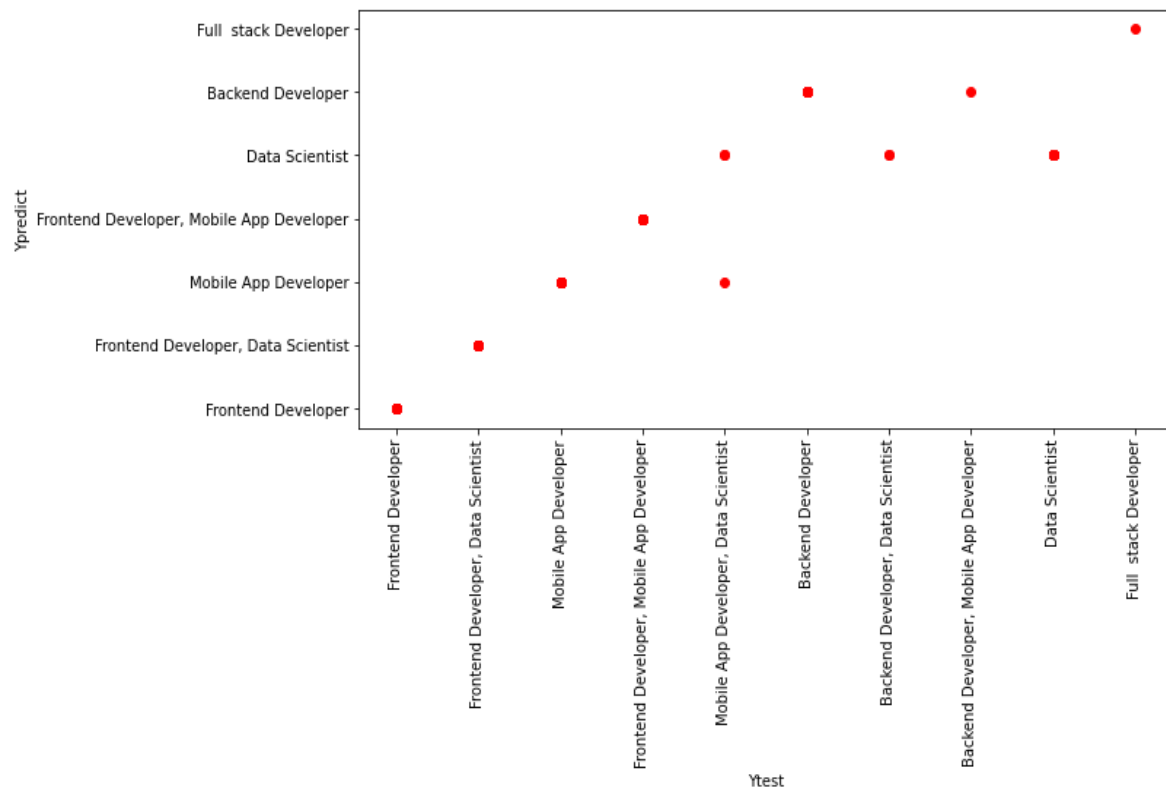
The accuracy determines the correctness of the algorithm and it is calculated as number of correct predictions by total number of predictions.

$$\text{Accuracy} = \frac{\text{Number of Correct predictions}}{\text{Total number of predictions made}}$$

Random Forest Classifier		Naive Bayes Classifier	
Accuracy	97%	Accuracy	87%

Table 6. Accuracy Rate

The below graph represents the relation between the actual result and the predicted result.



CHAPTER 5

CONCLUSION AND FURTHER WORK

5. Conclusion and Further Work

The respective job roles for the candidates for attending the interview are classified properly with 97% accuracy in Random Forest Classifier algorithm and 81% accuracy with the Naive Bayes Classifier algorithm. Finally, the model created by using Random Forest Classifier is considered for future inputs from the GUI (Graphical User Interface). These classified job roles based on the candidate's skills are further recommended to the recruiters when they do search for the respective roles. This kind of approach is very much useful to pick the candidates with hands of experience and also avoid making people work in an unsatisfactory position for a long. There are two modules one for recruiter and the other for a candidate, the recruiter module is the place where every recruiter may sign up to create their account and then login themselves to search for a candidate at the position they are looking for, and they may visualize the candidate details and able send mails to the respected best of all candidates and ask them to attend for the interview later. The second module is for the candidates who are allowed to sign up their user account and login to their respective profile in order to fill their resume online and the candidate after applying for the resume will be recommended to their recruiters with the positions predicted by the Random Forest Classifier algorithm and then they can also allow to edit their resume if needed, this predicted result is viewed by the interviewers to the process of handpicking the candidates for the respective position.

CHAPTER 6

REFERENCES

6. References

- [1] F.P Brooks Jr., The Mythical Man-Munth, Addison-Wesley, 1995.
- [2] A. Capillupi , A.Serebrenik, L. Singer, Assessing technical candidates on the social web, IEEE Softw, 30 (1) (2013) 45-51
- [3] J.E, Montandon, C. Politowski, L.L. Silva, m.T. Valente, F.Petrillo, G.Gueheneuc, what skills do IT companies look for in new developers? A study with stack overflow jonbs, Inf Software, Technology. 129 (2021) 1-6.
- [4] G.Tsoumakas, I. KataKis, I, Vlahavas, Mining Multi-label Data, Springer, PP.667-685.

CHAPTER 7
APPENDIX

7. Appendix

7.1 Implementation of Random Forest Classifier

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

import pickle
dataset=pd.read_csv("C:\\Users\\Kannan SA\\Desktop\\web_based_projects\\templates\\QueryRe
sults (3).csv")
#print(dataset.head())
dataset['SQLite']=dataset['SQLite'].fillna(dataset['SQLite'].mode()[0])
dataset['Node.js']=dataset['Node.js'].fillna(dataset['Node.js'].mode()[0])
dataset['NoSQL']=dataset['NoSQL'].fillna(dataset['NoSQL'].mode()[0])
dataset['RDBMS']=dataset['RDBMS'].fillna(dataset['RDBMS'].mode()[0])
dataset['MongoDB']=dataset['MongoDB'].fillna(dataset['MongoDB'].mode()[0])
dataset['PostgreSQL']=dataset['PostgreSQL'].fillna(dataset['PostgreSQL'].mode()[0])
dataset['Swift']=dataset['Swift'].fillna(dataset['Swift'].mode()[0])
dataset['Scala']=dataset['Scala'].fillna(dataset['Scala'].mode()[0])
dataset['XML']=dataset['XML'].fillna(dataset['XML'].mode()[0])
dataset['C/C++']=dataset['C/C++'].fillna(dataset['C/C++'].mode()[0])
dataset['Python']=dataset['Python'].fillna(dataset['Python'].mode()[0])
dataset['MATLAB']=dataset['MATLAB'].fillna(dataset['MATLAB'].mode()[0])
#dataset.drop(['Unnamed: 52'],axis=1,inplace=True)
print(dataset.isnull().sum())
#print(dataset.columns)
#print(dataset.columns[8:44])
x=dataset[dataset.columns[8:-1]]
y=dataset['Position']
#-----
#print("x ",x.columns)
#print("y",y.columns)
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2)
rff=RandomForestClassifier()
rff.fit(xtrain,ytrain)
#samp=np.array([1,1,1,0,1,0,1,0,1,1,1,0,1,0,0,0,0,0,0,1,0,0,0,1,0,1,1,0,1,1,1,1,0,0,1,0,1,1,1,0,1,
1])
ypred=rff.predict(xtest)
#ypred=rff.predict(samp.reshape(1,-1))
print("accuracy : ",accuracy_score(ytest,ypred))
```

7.2 Implementation of Naive Bayes Classifier

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

import pickle
dataset=pd.read_csv("C:\\Users\\Kannan SA\\Desktop\\web_based_projects\\templates\\QueryRe
sults (3).csv")
#print(dataset.head())
dataset['SQLite']=dataset['SQLite'].fillna(dataset['SQLite'].mode()[0])
dataset['Node.js']=dataset['Node.js'].fillna(dataset['Node.js'].mode()[0])
dataset['NoSQL']=dataset['NoSQL'].fillna(dataset['NoSQL'].mode()[0])
dataset['RDBMS']=dataset['RDBMS'].fillna(dataset['RDBMS'].mode()[0])
dataset['MongoDB']=dataset['MongoDB'].fillna(dataset['MongoDB'].mode()[0])
dataset['PostgreSQL']=dataset['PostgreSQL'].fillna(dataset['PostgreSQL'].mode()[0])
dataset['Swift']=dataset['Swift'].fillna(dataset['Swift'].mode()[0])
dataset['Scala']=dataset['Scala'].fillna(dataset['Scala'].mode()[0])
dataset['XML']=dataset['XML'].fillna(dataset['XML'].mode()[0])
dataset['C/C++']=dataset['C/C++'].fillna(dataset['C/C++'].mode()[0])
dataset['Python']=dataset['Python'].fillna(dataset['Python'].mode()[0])
dataset['MATLAB']=dataset['MATLAB'].fillna(dataset['MATLAB'].mode()[0])
#dataset.drop(['Unnamed: 52'],axis=1,inplace=True)
print(dataset.isnull().sum())
#print(dataset.columns)
#print(dataset.columns[8:44])
x=dataset[dataset.columns[8:-1]]
y=dataset['Position']

xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2)
nb=GaussianNB()
nb.fit(xtrain,ytrain)
ypred=nb.predict(xtest) # ypred=rff.predict(samp.reshape(1,-1))
print("accuracy : ",accuracy_score(ytest,ypred))
```

7.3 Plagiarism Report

ORIGINALITY REPORT			
3%	2%	1%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	www.chooseyourboss.com Internet Source	1%	
2	joaopescada.com Internet Source	1%	
3	myfik.unisza.edu.my Internet Source	<1%	
4	"Progress in Advanced Computing and Intelligent Engineering", Springer Science and Business Media LLC, 2021 Publication	<1%	
5	synote.org Internet Source	<1%	
6	A. Z. Kouzani. "Lung nodules detection by ensemble classification", 2008 IEEE International Conference on Systems Man and Cybernetics, 10/2008 Publication	<1%	
7	Zahra Shakeri Hossein Abad, Alex Shymka, Susant Pant, Ashley Currie, Guenther Ruhe. "What are Practitioners Asking about	<1%	