CSCE 5290: Natural Language Processing

Group 3 Project Report

<u>Job Description Based Resume Matcher Using Text Summarization, Keyword Extraction</u>

1. Project Title and Team Members

Job Description Based Resume Matcher Using Sentement Analysis,TextSummarization, Keyword Extraction

Suhas Siddarajgari Tellatakula	11626111
Vamshi Telukuntla	11618397
Srikanth Mamillapalli	11551359
Sairohithvarma Kantem	11606743

GitHub link: https://github.com/Vams98/NLP-project#nlp-project

2. Goals and Objectives:

- ### Motivation:

  Now-a-days, candidates applying to jobs online by uploading resumes is increasing rapidly. But unfortunately, only few of them manage to secure dream jobs in respective dream companies/firms. This is because most of the candidates’ resumes will not get picked in the first place. Students frequently apply to numerous jobs with single resume irrespective of job description. There is need to be aware of creating job description tailored resumes. So, we are trying to build a solution which can analyse and understand the job descriptions by generating the sentiments and a short summary for every job description. Also, it generates the top keywords from the resumes dataset.

- ## Significance:

As companies receive plethora of applications for one job opening, it will be very difficult for the companies to go through each resume and filter the candidates. Hence, they employ artificial intelligencebased resume pickers which look for specific keywords in the resumes. Creating job description based tailored resumes is one of the crucial steps in applying for a job. A recent survey indicates that 34% of people do not know how to prepare unique resumes based on job description. Hence, this tool can be extremely helpful for candidates to check their resume relevancy to the applied job and increase their chances of landing dream job.

- ## Objectives:

With this project, we are aiming to solve above problem by taking advantage of the Natural Language Processing methodologies employing text summarization, label classification and keyword extraction techniques. The objectives of the project are as follows:

➢ To summarize multiple paragraph job/role descriptions into a simple yet fully understandable output by executing and finalizing best out of different text summarization techniques like pyTextRank, Spacy pipeline.

➢ To classify the job description based on the tone using lexicon based semantic analyzer.

➢ To classify the jobs/roles under few tags/groups based on the above generated short summary employing different concept/tags classification mechanisms.

➢ To identify and filter out the important keywords from a document by trying out and choosing best one out of different keyword extraction methods using TfIdfVectorizer.

➢ To provide the relevancy or matching score between job descriptions and resumes based on a machine learning model built on the generated summary and keywords extracted from resumes.

▪ Features:

Once the project is fully developed based on the above objectives, the solution is expected to have following features:

➢ Tool provides a brief summary from a posted job description/role requirement.

➢ Tool provides a sentiment analysis for the job descriptions .

➢ Tool can extract important relevant keywords from a resume document.

➢ Tool suggests best fit candidates' resumes to a given job description or generates matching score between job description and resume.

3. Related Work (Background):

Title 1: Automated Resume Evaluation System using NLP

Link: https://ieeexplore.ieee.org/abstract/document/9036842

Summary:

For most businesses, the hiring process is essential, and traditional hiring practices have lost their effectiveness as online hiring grows in popularity. The traditional methods of manually screening applications and creating a shortlist of qualified applicants for interviews take a long time and a lot of work. Job hunting has improved in both accessibility and sophistication in the age of technology. The selection of a candidate based solely on their resume has not, however, been totally automated. This study suggests a model that ranks resumes in accordance with the preferences and requirements of the employer by extracting useful information from them.

To accomplish the desired result, the suggested model has three segments. Using NLP approaches, the first section transforms unstructured resumes into structured data. The extraction phase, which makes up the second section, is when pertinent data from the resume is taken out and given an identifying value. In the final section, the resumes are ranked in accordance with the values assigned. The model streamlines the hiring procedure and saves time and effort by segmenting the entire process into four parts.

The suggested technique provides an automated way for recruiters to find qualified candidates by gleaning important data from their resumes. The model transforms unstructured data into structured data using NLP techniques, making it simpler to retrieve pertinent information. The model then gives the information that was retrieved an identifying value and ranks the resumes according to the values given. This strategy enables a quicker and more effective hiring procedure, which is advantageous to both the business and job seekers.

Title 2 : Ontology Based Job and Resume Matcher

Link:
https://www.cmpe.boun.edu.tr/~gungort/papers/Ontology%20Based%20Job%20and%20Resume%20Matcher.pdf

Summary:

An innovative method for enhancing the job matching process is presented in this paper in the form of an ontology-based job and resume matcher system. The authors suggest a system that uses an ontology-based model for matching, pulls qualifications and experience from both resumes and job postings. Using ontologies allows for a more accurate and efficient matching process by giving the knowledge domain a structured representation. The method provides a better user experience, requires little input from job seekers, and improves the match between the candidate and the job.

The authors provide a thorough analysis of the system's performance, proving that their strategy outperforms current practices. The system's 87% precision rating shows that the matching procedure was highly accurate. The system's architecture and the numerous parts that make up the system are also fully described by the authors. Researchers and developers interested in creating comparable systems or researching the usage of ontologies in the context of job matching will find this information to be useful.

The study offers an innovative method that uses ontologies to increase the speed and accuracy of the matching process, making a significant addition to the field of job matching and recruitment overall. The evaluation of the system gives solid proof of its efficacy, and the publication is a useful resource for academics and developersin the field due to the authors' in-depth description of the system's design and components. The essay makes a strong argument for the use of ontology-based systems in future job matching systems by illustrating their potential in the field of recruitment.

Title 3: An Intelligent framework for E-Recruitment System Based on Text Categorization and Semantic Analysis

Link: https://ieeexplore.ieee.org/abstract/document/9544102

Summary:

An new strategy is to employ NLP technology to create an autonomous text classification system for job and resume categorization in the online hiring process. Recruiters and job searchers can use the suggested system to identify appropriate job offers and resumes, respectively. Both parties depend on the system's correctness because incorrect categorization could lead to a mismatch between talents and employment requirements. To extract pertinent information from the resumes, the system uses POS tagging, tokenization, and lemmatization of the data. The Phrase Matcher is used to rate resumes based on the information provided by the recruiter, suggesting to job seekers that their skills are inadequate and giving the recruiter access to the best resumes. The system is effective in classifying resumes based on job descriptions, as shown by the results of the system evaluation, which showed better precision and recall.

The suggested method takes a novel approach to categorizing resume data on a large dataset of job descriptions by utilizing Word Order Similarity between Sentences and Domain Adaptation for the sensitive nature of resume content. The technology can offer job seekers customized job recommendations by grouping individuals based on the data in their resumes. The system's results and analyses are displayed, giving information about how well the strategy works to increase the effectiveness of the hiring process. The proposed system's capacity to spot job seekers' skill gaps and recommend them is a critical component in assisting them in improving their employability. Also, the system's capability to give the recruiter the best resumes can considerably lessen their workload so they can concentrate on the most relevant Resumes. Overall, the proposed system is a promising strategy that has the potential to change online recruitment by enhancing its precision, efficacy, and efficiency.

Title 4: A novel firefly driven scheme for resume parsing and matching based on entity linking paradigm

Link: https://www.tandfonline.com/doi/abs/10.1080/09720529.2020.1721879

Summary:

In order to increase the effectiveness and efficiency of the hiring process, the paper outlines a data-driven HR approach that makes use of NLP approaches. In order to effectively match resumes with job requirements, the authors specifically created a resume parser that evaluates critical recruitment parameters and incorporates a pie chart presentation into the algorithmic structure. The firefly ranking algorithm was used to assess the effectiveness and accuracy of the suggested approach, which produced an overall accuracy of 94.19%.

The paper's contribution is the creation of a strong resume parser that effectively matches resumes with job requirements, overcoming the drawbacks of traditional hiring practices. The parser can now more accurately extract crucial data from unstructured resumes thanks to the application of NLP techniques. Also, by supplying a useful tool for matching resumes with job requirements, the inclusion of a pie chart display in the algorithmic framework improves the parser's efficiency. The findings indicate that the proposed strategy has a great deal of potential to increase the effectiveness and efficiency of the hiring process.

Title 5: Resume Parsing Framework for E-recruitment

Link: https://ieeexplore.ieee.org/abstract/document/9721762

Summary:

In order to accurately extract information from resumes for hiring reasons, the article suggests a paradigm for resume parsing. Due to the wide range of formats used in resumes and the requirement for substantial annotated data, the existing methods, such as rule-based, supervised, and semantics-based methods, have their limits. In the suggested framework, resumes' raw content is extracted, and blocks are divided using text block categorization. Afterwards, named entity recognition is used to extract the entities, and an ontology is then used to enrich them. The suggested approach accurately extracts information from resumes that aids in the decision-making process and addresses the shortcomings of existing methodologies.

The suggested approach makes a substantial contribution to the development of recommender systems for e-recruitment. A more effective and efficient hiring procedure may result from the precise information extraction from resumes utilizing the suggested methodology. The framework is a useful advancement in the resume parsing field since it can manage the constraints imposed by current approaches. The retrieved entities are enriched using ontology, which improves the framework's accuracy. The suggested framework has the potential to considerably enhance the hiring procedure overall.

Title 6: Learning-Based Matched Representation System for Job Recommendation

Link: https://www.mdpi.com/1943792

Summary:

In this paper, a recommender system that helps job searchers locate relevant positions based on their resumes is proposed. When faced with multiple job offers, the conventional job recommendation algorithms were unable to suggest positions that appropriately matched the job seekers' profiles. The suggested method, however, uses content-based filtering to analyze and compare the similarity between the job seeker's talents and explicit elements of the job description, and then recommends the top-n positions to the job seekers. In order to match CV skills to advertised positions, the system first obtained first-hand information by extracting job descriptions from Indeed from key Saudi Arabian cities. It then examined the top talents demanded in job offers. Using decision support tools, the success and error rates of recommendations were compared to actual results.

This study is essential because it offers a solution to the issue that job searchers confront when trying to find openings that match their qualifications and expertise. The recommender system's content-based filtering method circumvents the drawback of conventional job suggestion techniques. By guaranteeing that job offers and resumes are appropriately matched, the proposed method benefits both recruiters and job seekers by saving time and money. Also, a more accurate evaluation of the effectiveness of the system can be obtained by using the decision support metrics when comparing the results of the system to reality. Considering the Saudi Arabian job market in particular, this paper offers a significant contribution to the field of job recommendation systems.

Title 7: A survey of job recommender systems

Link: https://academicjournals.org/journal/IJPS/article-full-text-pdf/B19DCA416592.pdf

Summary:

In order to create individualized recommender systems for candidates and job matching, the article presents an overview of e-recruiting procedures and current recommendation techniques. Traditional information retrieval methods are no longer sufficient due to the rise of internet-based recruiting platforms, which causes many candidates to lose out on job possibilities. The recommender system technology, which addresses concerns with information overload, has proved successful in e-commerce applications. Consequently, to enhance the functionality of e-recruiting, the essay emphasizes the significance of recommender systems.

The article provides a thorough analysis of current recommendation methods and e-recruiting procedures. It offers insights into the many methods used in personalized recommender systems, such as content-based, collaborative, hybrid, and knowledge-based strategies. It is an educational read for scholars and practitioners in the field because the writers also cover the benefits and limits of each technique. Overall, this research advances knowledge of recommender systems' function in e-recruiting and the potential advantages they may offer to both job seekers and recruiters.

Title 8: Matching Resumes to Jobs via Deep Siamese Network

Link: https://dl.acm.org/doi/abs/10.1145/3184558.3186942

Summary:
The difficult problem of proposing positions to job-seeking people by matching their resumes to job descriptions is the main topic of this study. The authors suggest a siamese adaption of convolutional neural networks to address this issue since it can accurately capture the underlying semantics and project similar job descriptions and resumes closer to one another in a semantic space. The method isevaluated on a sizable dataset of more than 5 million pairs of resumes and job descriptions, and the experimental findings show that the proposed method outperforms the state-of-the-art approaches.

This study is useful because it addresses a crucial problem in job searching by putting forth a cutting-edge method that successfully matches resumes to job descriptions. A promising method that can capture the underlying semantics of the data and increase the accuracy of the job recommendation system is the use of siamese adaption of convolutional neural networks. Strong proof that the suggested strategy outperforms current methods and might be used for real-world job recommendation systems is shown by the authors' extensive experiment.

Title 9: Hybrid Job and Resume Matcher

Link: : https://ieeexplore.ieee.org/abstract/document/9558932

Summary:

The article presents a research project that expands on a system that was previously created for scoring resumes and matching them with job adverts. Using machine learning and deep learning approaches, the authors suggest a strategy that is more effective and powerful for extracting information from resumes and job adverts that have several forms. Compared to the earlier rule-based approaches, which were only efficient when the structure of resumes and job advertisements was understood, this represents a substantial advancement.

The process of assessing resumes and connecting them to job advertisements could be completely transformed by the suggested approach. The information extraction process, which is a crucial stage in the matching and scoring system, can be made more accurate and efficient with the help of machine learning and deep learning techniques. Recruiters, HR managers, and job seekers who want to enhance the efficacy and efficiency of their job search can all benefit from this research effort. The research work and its possible benefits are summarized succinctly in this abstract, but it would be fascinating to read the whole report to learn more about the specifics of the suggested method and its assessment.

Title10: Smart Talents Recruiter-Resume Ranking and Recommendation System

Link: https://ieeexplore.ieee.org/abstract/document/8913392

Summary:

The creation of a brand-new applicant recommendation system known as the Smart Applicant Ranker is discussed in the article. Utilizing ontologies and two ranking algorithms, the system is made to assist recruiters in entering job requirements and matching them with the best comparable prospects. The usage of Semantic Web technology helps IT recruitment organizations find expertise in an efficient and economical manner.

The importance of candidate recommendation systems in the context of tactical and financial cost-effectiveness is highlighted in the essay. This issue can be promptly solved by the Smart Applicant Ranker, which can assist recruiters in finding the best qualified applicants for a specific job requirement rapidly. The use of two ranking algorithms and an ontology is a promising strategy that may improve the precision and potency of the recommendation system.

The post does a fantastic job of introducing the Smart Applicant Ranker and some of its possible advantages. More information regarding the ontology and ranking algorithms utilized in the system, as well as the evaluation findings to show its efficacy, would be fascinating to see. The article nonetheless makes a significant contribution to the subject of candidate recommendation systems and might be helpful to both employers and job seekers**.**

Title11: Implicit Skills Extraction Using Document Embedding and Its Use in Job Recommendation

Link: https://ojs.aaai.org/index.php/AAAI/article/view/7038

Summary:

The approach described in the article uses a combination of natural language processing (NLP) methods and the idea of implicit talents to match resumes to job descriptions. The approach for extracting implicit abilities, which are those that are not expressly listed in a job description but may be implicit given the region, industry, or role, is suggested in this study. The technique produces the final set of implicit abilities using a Doc2Vec model trained on 1.1 million JDs and a number of weighting techniques. The suggested strategy gets a mean reciprocal rank of 0.88, outperforming a baseline method that uses solely explicit skills.

The article makes a significant contribution to the subject of job recommender systems because it discusses the difficulty of matching unconventional and ad hoc applications with semi- or unstructured job descriptions. The unique strategy for extracting implicit talents that has been put out may help the job recommender system's precision. The robustness of the suggested strategy is further enhanced by the use of various weighting methods and similarity metrics.

The success of the suggested technique is shown by the article's clear explanation of the methodology employed and the evaluation findings attained. To train the Doc2Vec model, additional information regarding the datasets utilized and the computational resources needed would be fascinating to observe. Nevertheless, the article makes a significant contribution to the field of job recommender systems and may be helpful to both employers and job seekers who want to increase the speed and efficacy of their job search.

Title12: AN AUTOMATED RESUME SCREENING SYSTEM USING NATURAL LANGUAGE PROCESSING AND SIMILARITY

Link: https://www.intelcomp-design.com/paper/2etit2020/2etit2020-99-103.pdf

Summary:

The article offers a remedy for the difficulties recruiting businesses encounter while sifting through the numerous applications they receive for a job posting. The suggested approach is to construct a summarized form of each application by using natural language processing techniques to extract pertinent data from unstructured resumes. This facilitates the screening process and allows recruiters to analyze each resume more thoroughly in less time.

To compare each resume to the job description and determine ranking scores, the method makes use of a vectorization model and cosine similarity. The proposed strategy provides a more accurate assessment of candidate suitability than the conventional method of manually evaluating resumes, according to the article's encouraging results.

In general, the proposed method seems to be a viable strategy to increase the effectiveness and fairness of the hiring process, helping both hiring organizations and job searchers.

Title13: Analyzing CV/resume using natural language processing and machine learning

Link: : https://dspace.bracu.ac.bd/xmlui/handle/10361/9480

Summary:

The article suggests a strategy for extracting crucial information from resumes or curriculum vitae's semi-structured text and ranking it in accordance with the preferences and needs of affiliated companies. Segmentation, data extraction, and evaluation are the three sections of the model. The first segment entails categorizing the resume, and the second segment entails employing HTML conversion to extract structured data from unstructured data. The decision tree method is used to categorize input into various groups according to qualifications in the final section, and the data with positive weight is used to train the system.

The article offers a thorough method for screening resumes and offers a solution to one of the major issues with the hiring procedure. The performance evaluation proves the efficacy of the suggested model, which is based on efficient segmentation, data extraction, and evaluation procedures. The emphasis on decision tree algorithms and other classifier algorithms in the study draws attention to the possibilities for using machine learning methods in the recruitment sector. The paper is organized nicely overall and offers a convincing solution to the issue of resume screening.

Title14: Competence-Level Prediction and Resume & Job Description Matching Using Context-Aware Transformer Models

Link: : https://arxiv.org/abs/2011.02998

Summary:

In order to improve the selection of qualified candidates while drastically lowering the time and labor required to screen a mountain of job applications, the article offers a thorough study on resume classification. The authors present a dataset of 6,492 resumes for clinical research coordinators (CRCs) that have been carefully classified by specialists into four levels of experience. The inter-annotator agreement had a strong Kappa value of 61%, proving the dataset's dependability.

The authors suggest two brand-new transformer-based methods for categorizing resumes. Using both the resume and the job description as inputs, the second model (T2) predicts if the applicant would be suitable for the position. The first model (T1) classifies resumes to a CRC level. Impressive results of 73.3% for T1 and 79.2% for T2 are obtained by the best models using section encoding and multi-head attention decoding.

Prediction mistakes primarily occur between adjacent CRC levels, which are challenging for even experts to discern, according to the study of the results. This result emphasizes the usefulness of the suggested models in actual HR platforms.

The paper makes a significant contribution to the fields of resume classification and HR recruitment overall. A considerable improvement is made in the process of screening resumes and choosing qualified candidates thanks to the high-quality dataset and creative transformer-based models. This results in time and labor cost savings. The results of the study illustrate the potential of natural language processing methods for HR recruitment and have major practical ramifications

Title15: Resume Ranking based on Job Description using SpaCy NER model

Link: https://www.academia.edu/download/64550140/IRJET-V7I516.pdf

Summary:

The essay emphasizes the difficulties of manual resume screening, which is a costly and time-consuming process for hiring managers and recruiters. The author suggests an automated method that extracts pertinent data from resumes based on job descriptions using sophisticated Natural Language Processing (NLP) techniques. The suggested methodology uses the Spacy NER model to automatically extract necessary entities and creates a graph showing each resume's score.

The automated method is quick and effective, making it simpler for recruiters to fill open positions. By doing away with the need to manually sort through a large number of resumes, the method decreases the likelihood of passing over qualified individuals. A promising way to increase the effectiveness of the hiring process is the usage of NLP techniques.

The proposed model's in-depth technical specifics are, however, lacking in the text, and the model's performance evaluation is not covered. The reader is left with doubts regarding the strategy's precision and dependability. The paper also skips over the drawbacks and inherent biases of employing automated resume screening methods.

Overall, the research offers a viable method for applying NLP techniques to automate the resume screening process. To verify the efficacy of the strategy, more technical information and an analysis of the model's performance are needed. The drawbacks and inherent biases of employing automated recruitment methods must also be taken into account.

Title16: Matching Jobs and Resumes: a Deep Collaborative Filtering Task

Link: https://inria.hal.science/hal-01378589/

Summary:

The study on automatic matching of recruiters and job seekers based on recruiting agency records is presented in the article. In order to determine how well a collaborative filtering mode and a cold start mode perform when recommending job candidates to recruiters, the authors undertake tests. According to the study, whereas cold start mode produces subpar outcomes, collaborative filtering mode produces good suggestion performances. The authors offer a speculative explanation for this phenomena, speculating that the language barrier between recruiters and job searchers may be a factor in the information gap.

By demonstrating how interaction logs between job searchers and recruiters offer useful information beyond what is found in CVs and job postings, the paper makes a contribution to the area. The authors' hybrid system, Major, uses a deep neural network to match the characteristics of collaborative filtering representations. Major's experimental validation yields encouraging results, especially in cold start mode.

The publication is generally well-structured and gives a concise summary of the investigation, the research contributions, and the experimental findings. The procedures utilized are fully explained by the authors, and the results are given in a clear and intelligible way. For businesses and recruiting firms striving to enhance their hiring procedures, the study is very pertinent. The findings of this study indicate that the effectiveness of job matching algorithms can be greatly improved by include interaction logs between job seekers and recruiters.

4. Dataset:

1. As part of project increment-1, we are employing "job description.csv" for sentiment analysis and text summarization; a detailed description of the dataset is provided below.
2. The sample "resume.csv" is used to extract keywords from the "resume.csv" dataset in order to best match the candidate resume to the specified job description, a detailed explanation of the dataset is provided below.

3. “stopwords.txt” file which is used for removing the stop words as a part of preprocessing process.

1. Detailed explanation about “Job description.csv” dataset.

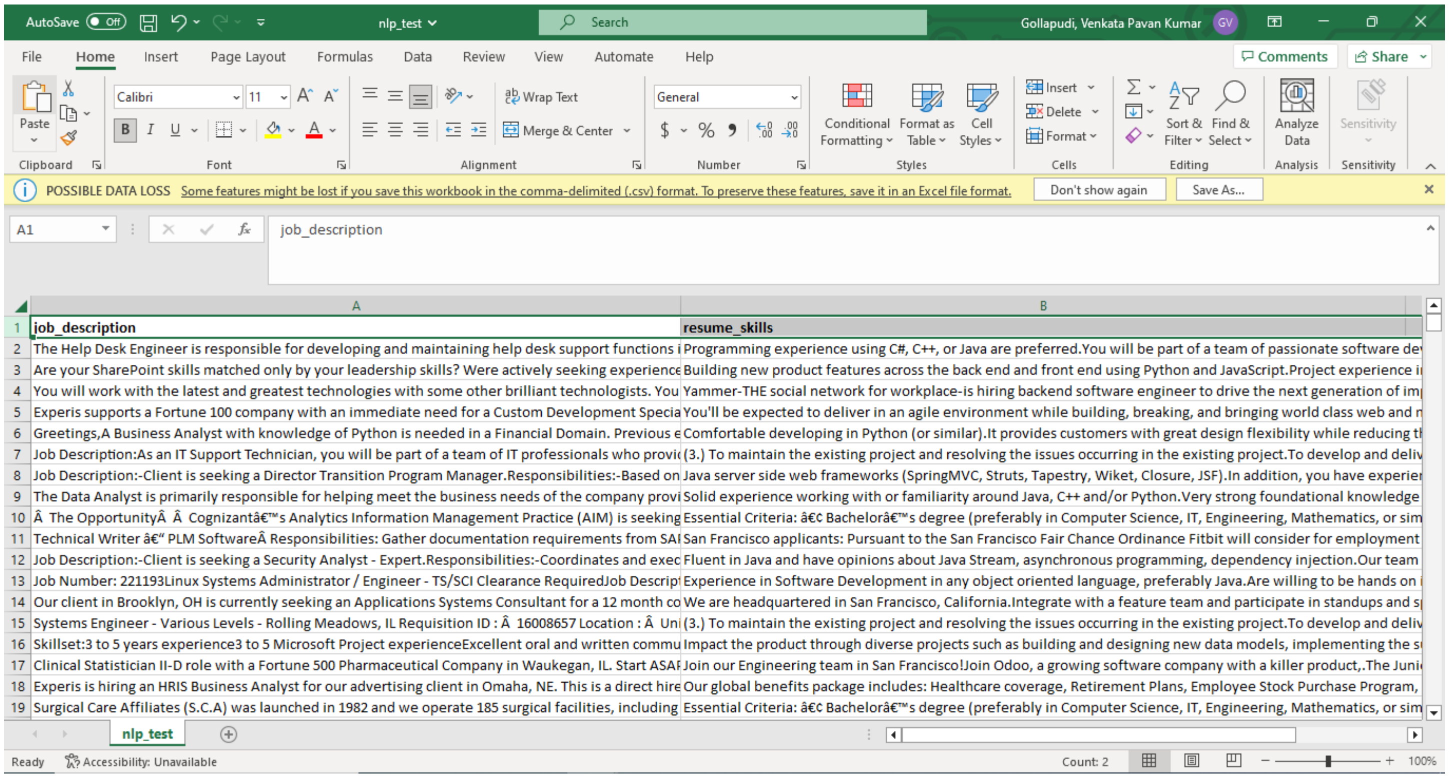
    - **Country**: Depicts the country name for a particular given “Job description”. Data type of this column is “String”.Eg: {“United states of America”}.
    - **Country code**: Depicts the country _ code for a particular given “job description”. Data type of this column is “string” of characters. Eg: {“US”}.
    - **Date added:** Depicts on which date the “Job” was posted.
    - **Has Expired:** Displays whether the job has expired or not expired. Data type of this column is “string” of characters. Eg:{“Yes”,”No”}.
    - **Job board:** Describes on which job board, job has been posted. Eg: {“Monster”,”Indeed”,”Linkedin}. Date type of this column is “string” of characters.
    - **Job Description:** Describes the Job description for a particular job that has been posted on the platform. Data type of this column is “string” of characters.
    - **Job Title:** Describes the Job Title for a particular job that has been posted on the platform. Data type of this column is “string” of characters. Eg: {“It support Technician”, “Cybercoders” etc}.
    - **Job Type:** Describes the type of job such as “Full Time” or “Contract”. Data Type of this column is “String” of characters.
    - **Location:** Describes the location of job for a particular job that has been posted. Data Type of this column is “String” of characters.
    - **Organization:** Describes the name of the organization. Data Type of this column is “String” of characters.

- **Page url:** Describes the “url” for the particular job that has been posted on the board. Data Type of this column is “String” of characters**.**
- **Salary:** Describes the “salary” for the job that has been posted. Data Type of this column is “String” of characters
- **Sector:** Describes the “sector” for the job that has been posted. Data Type of this column is “String” of characters
- **Uniq id:** Describes the “uniq id” for the job that has been posted. Data Type of this column is “String” of characters

2. Detailed explanation about “Resume.csv” dataset.

- **Id:** Describes the “Id” for the job that has been posted. Data Type of this column is “String” of characters
- **Resume Str:** Describes the “Resume_str” for the job that has been posted. Data Type of this column is “String” of characters
- **Resume html:** Describes the “Resume html” for the job that has been posted. Data Type of this column is “String” of characters
- **Category:** Describes the “Category” for the job that has been posted. Data Type of this column is “String” of characters. Eg{“HR”, ”TR”}.

From the above datasets, we extracted the IT related job descriptions and IT related resume skills and uploaded as dataset by name “nlp_test.csv”



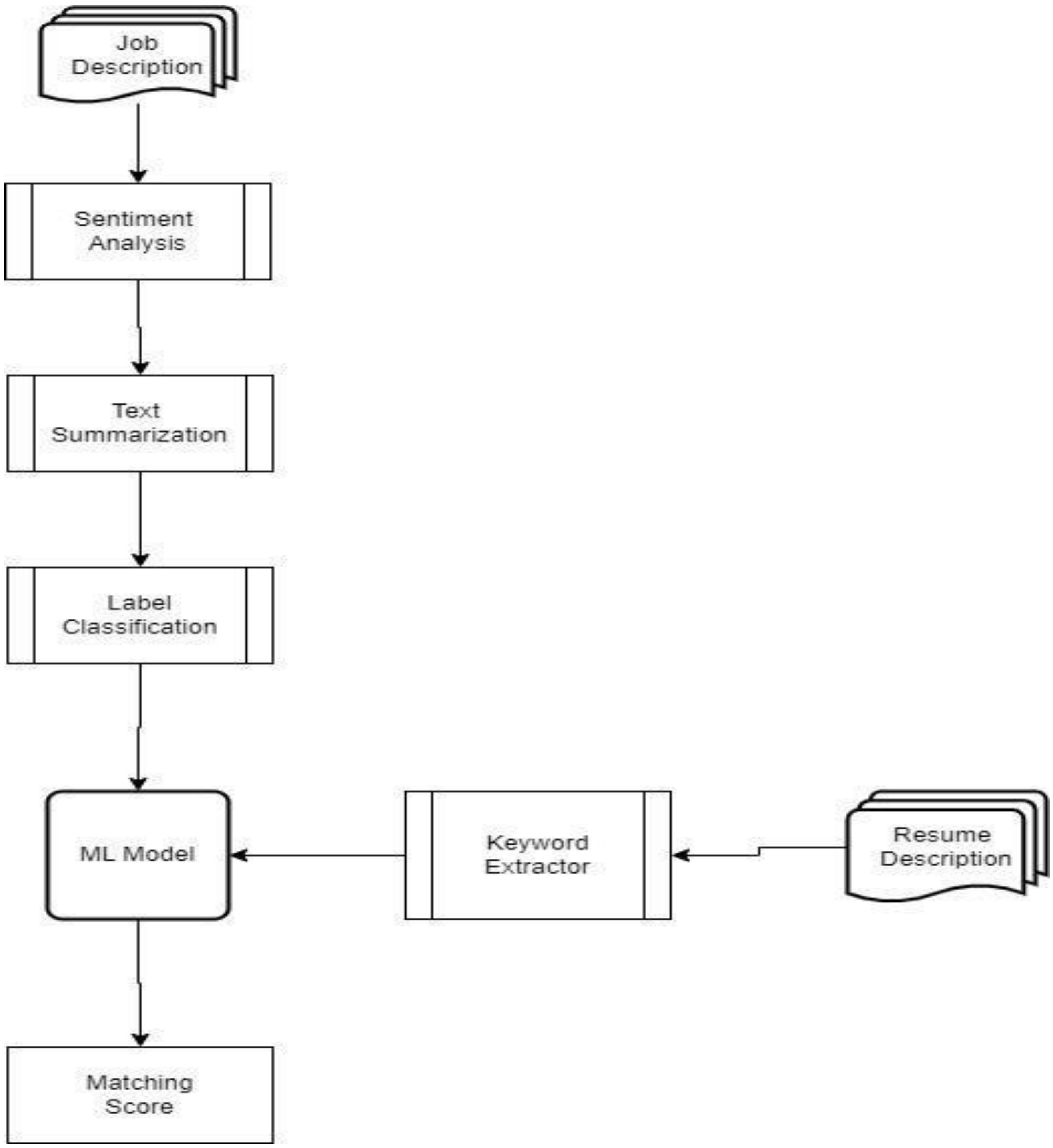
	A	B
1	job_description	resume_skills
2	The Help Desk Engineer is responsible for developing and maintaining help desk support functions i	Programming experience using C#, C++, or Java are preferred.You will be part of a team of passionate software de
3	Are your SharePoint skills matched only by your leadership skills? Were actively seeking experience	Building new product features across the back end and front end using Python and JavaScript.Project experience i
4	You will work with the latest and greatest technologies with some other brilliant technologists. You	Yammer-THE social network for workplace-is hiring backend software engineer to drive the next generation of im
5	Experis supports a Fortune 100 company with an immediate need for a Custom Development Specia	You'll be expected to deliver in an agile environment while building, breaking, and bringing world class web and
6	Greetings,A Business Analyst with knowledge of Python is needed in a Financial Domain. Previous e	Comfortable developing in Python (or similar).It provides customers with great design flexibility while reducing t
7	Job Description:As an IT Support Technician, you will be part of a team of IT professionals who provi	(3.) To maintain the existing project and resolving the issues occurring in the existing project.To develop and deli
8	Job Description:-Client is seeking a Director Transition Program Manager.Responsibilities:-Based on	Java server side web frameworks (SpringMVC, Struts, Tapestry, Wiket, Closure, JSF).In addition, you have experie
9	The Data Analyst is primarily responsible for helping meet the business needs of the company provi	Solid experience working with or familiarity around Java, C++ and/or Python.Very strong foundational knowledge
10	Â The OpportunityÂ Â Cognizantâ€™s Analytics Information Management Practice (AIM) is seeking	Essential Criteria: â€¢ Bachelorâ€™s degree (preferably in Computer Science, IT, Engineering, Mathematics, or sim
11	Technical Writer â€“ PLM SoftwareÂ Responsibilities: Gather documentation requirements from SA	San Francisco applicants: Pursuant to the San Francisco Fair Chance Ordinance Fitbit will consider for employment
12	Job Description:-Client is seeking a Security Analyst - Expert.Responsibilities:-Coordinates and exec	Fluent in Java and have opinions about Java Stream, asynchronous programming, dependency injection.Our team
13	Job Number: 221193Linux Systems Administrator / Engineer - TS/SCI Clearance RequiredJob Descrip	Experience in Software Development in any object oriented language, preferably Java.Are willing to be hands on
14	Our client in Brooklyn, OH is currently seeking an Applications Systems Consultant for a 12 month co	We are headquartered in San Francisco, California.Integrate with a feature team and participate in standups and s
15	Systems Engineer - Various Levels - Rolling Meadows, IL Requisition ID : Â 16008657 Location : Â Un	(3.) To maintain the existing project and resolving the issues occurring in the existing project.To develop and deli
16	Skillset:3 to 5 years experience3 to 5 Microsoft Project experienceExcellent oral and written commu	Impact the product through diverse projects such as building and designing new data models, implementing the s
17	Clinical Statistician II-D role with a Fortune 500 Pharmaceutical Company in Waukegan, IL. Start ASA	Join our Engineering team in San Francisco!Join Odoo, a growing software company with a killer product,.The Juni
18	Experis is hiring an HRIS Business Analyst for our advertising client in Omaha, NE. This is a direct hire	Our global benefits package includes: Healthcare coverage, Retirement Plans, Employee Stock Purchase Program,
19	Surgical Care Affiliates (S.C.A) was launched in 1982 and we operate 185 surgical facilities, including	Essential Criteria: â€¢ Bachelorâ€™s degree (preferably in Computer Science, IT, Engineering, Mathematics, or sim

AutoSave Off
nlp_test
Search
Gollapudi, Venkata Pavan Kumar
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Automate
Help
Comments
Share
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.
Don't show again
Save As...
A1
job_description
nlp_test
Ready
Accessibility: Unavailable
Count: 2
100%

5.Implementation and Detail design of Features:

Detailed Workflow

According to the project's workflow, we are performing 70% of the project in the project increment 1. First, we downloaded and read the job descriptions dataset. And perform the pre-processing before we work on the dataset. We drop the unwanted rows and columns and rows having null values. We added sentiment analyzer based on the comments from Project proposal. We performed 2 kinds of sentiment analyzer namely VaderSentiment and textblob techniques. Then we implemented the spacy pipeline and added textrank to extract brief summary for every job description. We implemented the TfIdf Vectorizer on resume dataset to extract the top keywords from resume descriptions.



Sentiment Analyzer

A sentiment analyzer that uses pre-built dictionaries or lexicons of words and their corresponding sentiment scores is known as a lexicon-based sentiment analyzer. Using the sentiment scores of the words it includes, this method seeks to determine the sentiment of a given passage of text. The working involves following steps:

- The first step is to preprocess the text data by tokenizing the text into individual words, removing stop words, and converting the text to lowercase.
- The next step is to develop a vocabulary or lexicon of terms and the sentiment scores assigned to them. The lexicon can be manually curated, crowdsourced sentiment annotations can be used, or pre-built lexicons like the AFINN lexicon or the NRC Emotion Lexicon can be used to accomplish this.
- Each word in the text is given a sentiment score based on its inclusion in the lexicon after the lexicon has been developed. For instance, the sentiment score for the word "happy" might be +1 whereas the sentiment score for the word "sad" might be -1.
- The final step is to aggregate the sentiment scores of all the words in the text to arrive at an overall sentiment score for the text. This can be done using various methods, such as taking the average of all the scores or using more sophisticated techniques such as weighted scoring.

Text Summarization

In extractive summarizing, a subset of the most crucial phrases or sentences from the original text are chosen to construct the summary. This is often accomplished by locating passages that contain words, named entities, or other significant elements that are indicative of the text's overall content. The summary is created by joining these selected sentences together.

In abstractive summarizing, the summary is produced by employing natural language generation techniques to produce new sentences that effectively convey the main ideas of the original text. This necessitates a deeper comprehension of the text's subject matter as well as the capacity to create meaningful sentences that express the same ideas as the original text.

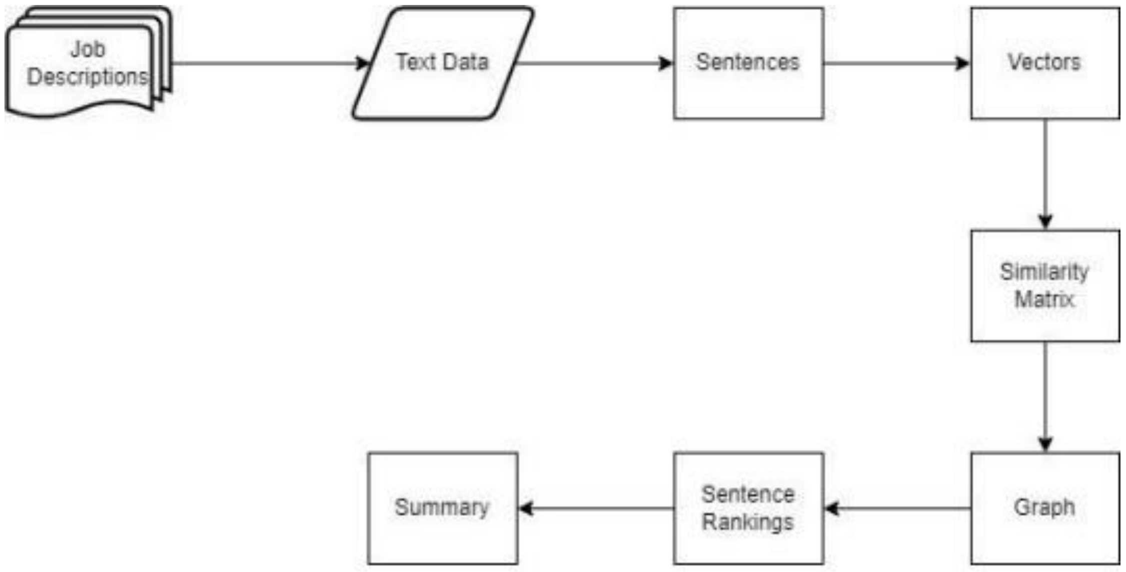


Fig 2. Text Summarization Process

Keyword Extraction

The open-source NLP library Spacy offers a number of tools for keyword extraction. The following steps are commonly involved in keyword extraction using Spacy:

- Preprocessing the text data involves breaking it down into individual words, getting rid of unnecessary words, and changing the text's case to lowercase. The Spacy tokenizer and pre-made language models can be used for this.
- The text is then subjected to part-of-speech (POS) tagging. POS tagging includes classifying each word in the text according to its syntactic type, such as a noun, verb,

adjective, or adverb. The POS tagger from Spacy can be used for this.

- Spacy can be used for entity recognition in addition to POS tagging. This entails locating named characters, groups, places, dates, or other entities in the text. These named entities might be seen as crucial textual keywords.
- After POS tagging and entity recognition have been used to preprocess and evaluate the text, the last step is to extract the most significant keywords. The built-in features of Spacy, like as phrase matching and noun chunking, can be used for this, as well as more sophisticated methods like TF-IDF or TextRank.

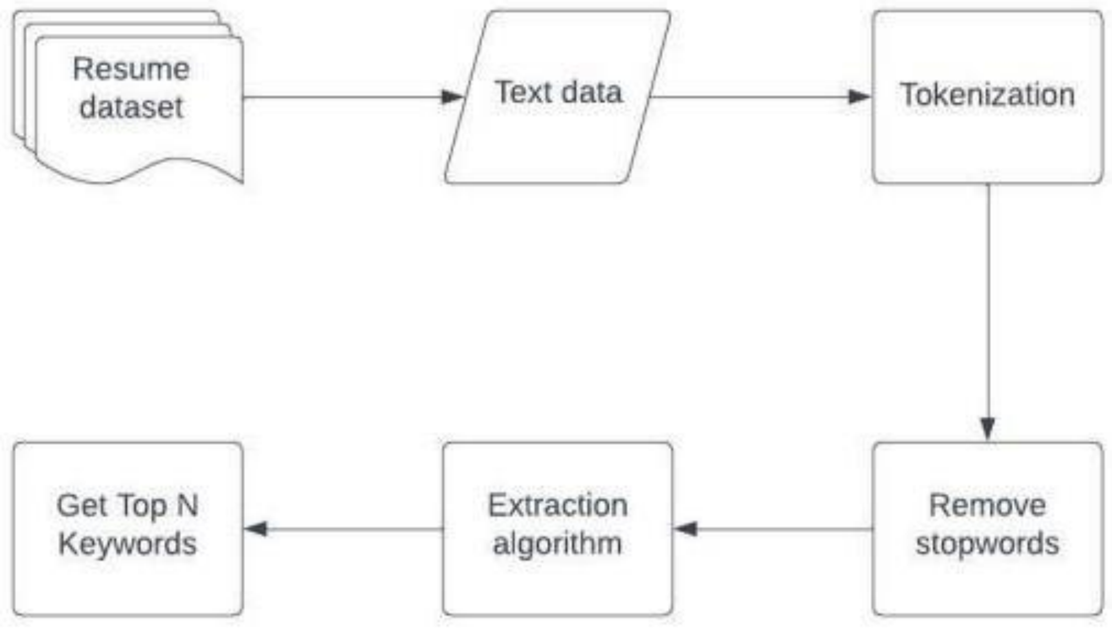


Fig 3. Keyword extraction Process

Label Classification:

Pandas is a Python package designed primarily for data processing and analysis. It offers highly optimized and efficient data structures for working with structured data types as tables, time series data, and panel data.

Matplotlib is a popular Python data visualization package. It includes a variety of tools for constructing various types of plots, including line plots, scatter plots, bar plots, histograms, and more. Matplotlib is widely used for visualizing data and findings in scientific computing, data analysis, and machine learning.
Working Involves following Steps

- The first step is to extract the Text Data from the Dataset
- In the second stage we are pre-processing the text data of Job dataset by removing stop words, punctutation.
- Third step involves Removing the null values from the dataset.
- Fourth step Involves Extracting the Labels from the Job Dataset such as Job Type (i.e Full Time, PartTime) and Sector(It/software,Staffing)

Finally we are plotting the Extracted Label such as "Job_type" and "sector" Into Graphically by using matplotlib library.

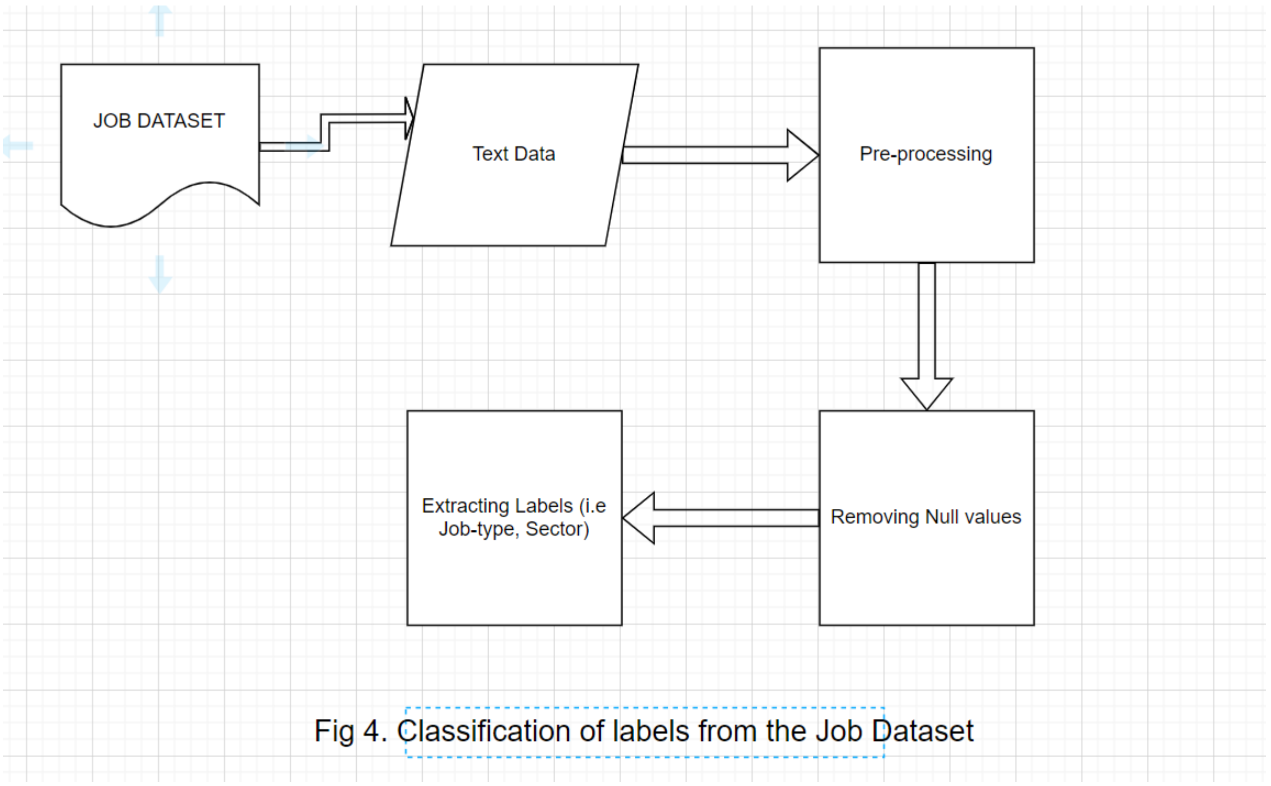


Fig 4. Classification of labels from the Job Dataset

Multi-Label Classification Using Machine Learning Model

Logistic Regression:

Logistic Regression is a statistical strategy used to predict one of two possible outcomes based on input data in binary classification tasks. It is a well-known machine learning method that is frequently utilized in a variety of fields including finance, healthcare, and marketing.

The input features are utilized to compute a weighted sum, which is then sent through a sigmoid function to get a probability score between 0 and 1. This probability score denotes the possibility of the input falling into one of two categories. To transfer the output to the probability range, the sigmoid function is utilized.

The logistic function, also known as the sigmoid function, is defined as:

sigmoid(z) = 1 / (1 + e^-z)

where z is the weighted sum of the input features.

The maximum likelihood estimation approach can be used to train the logistic regression model, which entails determining the parameters that maximize the likelihood of the observed data. In logistic regression, the objective function is the log-likelihood function, which is denoted as:

log L = sum(yi log(pi) + (1 - yi) log(1 - pi))
where yi is the real class label (0 or 1) for the i-th sample, pi is the projected probability of the i-th sample belonging to the positive class, and the total is calculated across all samples.

Once the model is trained, it can be used to predict the class label of new input data by calculating the probability score using the input features and comparing it to a threshold value. If the probability score is greater than the threshold value, the input is classified as belonging to the positive class, otherwise, it is classified as belonging to the negative class.
Working Involves following steps:

- The first step is to extract the Text Data from the Dataset
- In the second stage we are pre-processing the text data of Job dataset

by removing stop words, punctutation.

- Third step involves Removing the null values from the dataset.
- In the Fourth stage splitting the Dataset into Train and Test Dataset.
- Training the model train dataset.
- Predicting the values such as label(sector) for the job description text data.
- Calculating the Accuracy, precision, recall and F1 score for the model

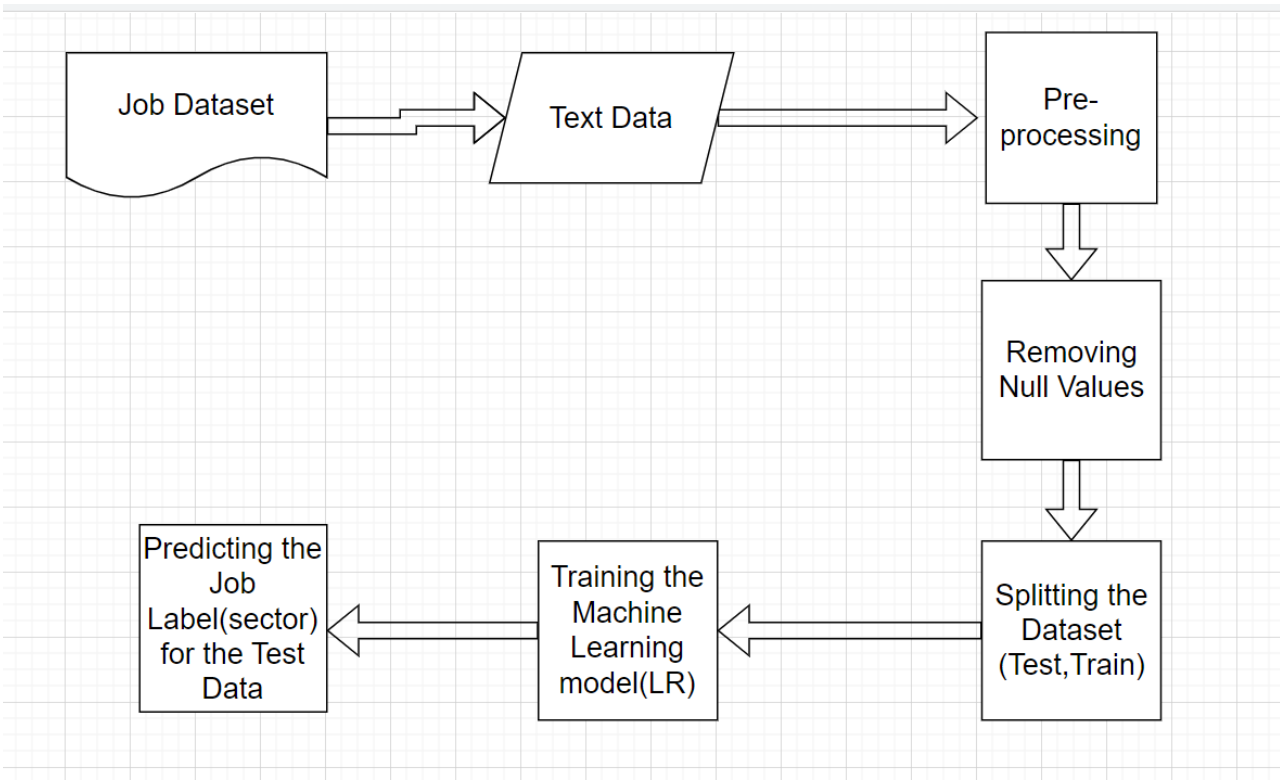

Job Dataset
Text Data
Pre-processing
Removing Null Values
Splitting the Dataset (Test,Train)
Training the Machine Learning model(LR)
Predicting the Job Label(sector) for the Test Data

Fig 5 Predicting Job label(sector) by training the model

Similarity Score calculation

Cosine Similarity

Cosine similarity is a measure of how similar two vectors in an n-dimensional space are. It's commonly utilized in domains including natural language processing, image processing, and recommendation.

The cosine similarity between two vectors, A and B, is derived as the
cosine of the angle between the two vectors, which may be obtained
using the dot product of the two vectors and their magnitudes. It is mathematically defined as follows:

cosine_similarity(A, B) = (A . B) / (||A|| ||B||)

where A . B is the dot product of vectors A and B, and ||A|| and ||B|| are the magnitudes of vectors A and B, respectively.

Cosine similarity spans from -1 to 1, with -1 indicating that the two vectors are entirely distinct, 0 indicating that the two vectors are orthogonal, and 1 indicating that the two vectors are identical.

Cosine similarity is used in recommendation systems to discover comparable goods or people based on their preferences or behavior. Cosine similarity, for example, may be used to find other users with similar likes and propose movies that they have rated highly based on a user's rating history of movies
Working Involves Following Steps:

- In the First Step Extracting the Text Data from Resume Dataset and as Well as Job Data.
- Removing the stop words, punctuation and Null values from the two Datasets.
- Calculating the cosine similarity between the two text data
- Displaying the similarity score.

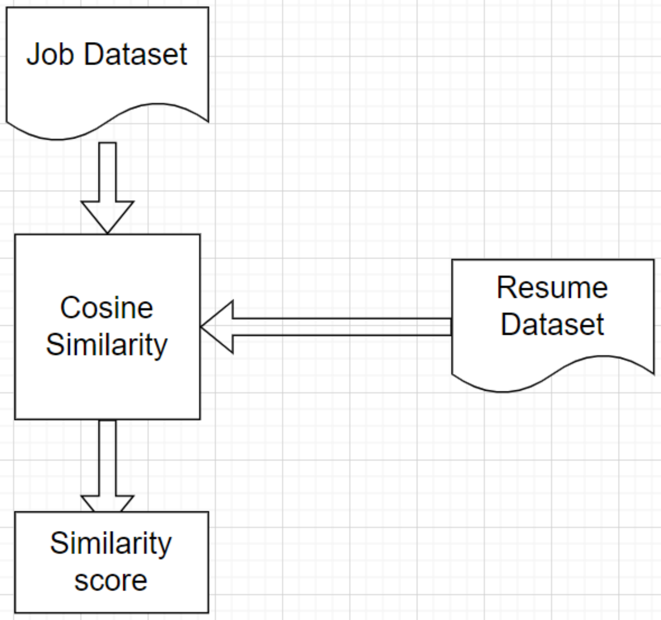


Fig 6 Similarity score calulation

<u>Similarity Score Calculation By fitting the machine Learning model</u>

<u>Naive Bayes Classifier</u>

Naive Bayes is a probabilistic classification technique used in machine learning to perform classification problems. It is based on the Bayes theorem and the assumption of input feature independence.
The purpose of Naive Bayes classification is to anticipate a new input's class label based on its attributes. The algorithm uses the Bayes theorem to compute the probability of each class given the input features:

$$P(class|features) = (P(features|class) * P(class)) / P(features)$$

P(class|features) represents the probability of the class given the input features, P(features|class) represents the probability of the features given the class, P(class) represents the prior probability of the class, and P(features) represents the probability of the input features.

The assumption of Naive Bayes is that the input characteristics are independent of one another, which implies that the probability of each feature given the class may be determined individually. This assumption simplifies the calculation of the conditional probability and minimizes the algorithm's computing complexity.

The following are the three most prevalent types of Naive Bayes classifiers:

Gaussian Naive Bayes: For continuous input features, this classifier assumes that the input features are normally distributed.

Multinomial Naive Bayes: This classifier is used to classify discrete input characteristics such as text.

Bernoulli Naive Bayes: This classifier is used to detect spam using binary input characteristics.
To train a Naive Bayes classifier, the algorithm must learn the prior probability of each class from the training data, as well as the conditional probability of each feature given the class. This can be accomplished through the use of maximum likelihood estimate or Bayesian estimation.

Once trained, the classifier can predict the class label of new input data by calculating the probability of each class given the input features and selecting the class with the highest probability.

Decision Tree

Decision Tree is a well-known machine learning technique that may be used for classification and regression applications. It is a basic and interpretable paradigm that is simple to perceive and comprehend.

A decision tree is a tree-like model that depicts a set of options and their potential outcomes. It is made up of root nodes, internal nodes, and leaf nodes. The root node symbolizes the decision-making process's beginning point, and each sub node reflects a choice based on one of the input characteristics. The final choice or conclusion is represented by the leaf nodes.

The decision tree is constructed by recursively splitting the data based on the input characteristics in such a way that the output subsets are as pure as feasible with regard to the target variable. A purity metric, such as entropy or Gini impurity, is used to determine the purity of a subset.

Entropy is a measure of an example's impurity or disorder, and it is defined as:
$E(S) = -sum(p(i) * log2(p(i)))$

where S represents the collection of examples, i represents the class label, and p(i) is the proportion of cases in S that belong to class i.

Gini impurity is another impurity metric that is frequently used in decision trees. It is defined as follows:

$G(S) = 1 - sum(p(i)^2)$

where S represents the collection of examples, i represents the class label, and p(i) is the proportion of cases in S that belong to class i.

The algorithm chooses the input feature that produces the purest subsets based on the purity metric and builds a new internal node based on that feature. This method is continued for each subset recursively until a stopping requirement, such as a maximum depth or a minimum number of instances per leaf node, is fulfilled.

The approach traverses the tree from the root node to the leaf node depending on the values of the input characteristics to predict the class label of a new input using a decision tree. The final class label represents the majority class among the leaf node samples.

RandomForest

Random Forest Classifier is an ensemble learning approach that uses numerous decision trees to increase the classification model's accuracy and resilience. It is a machine learning method that is commonly used for categorization jobs.

Random Forest's main idea is to generate a huge number of decision trees, each one based on a random subset of the input characteristics and a random portion of the training data. Because of the unpredictability, each tree is unique and captures a distinct element of the data. The ultimate forecast is then created by averaging all of the different tree predictions.

The algorithm follows the following steps to build the Random Forest Classifier:

Select a subset of the input features at random from the total features.

Choose a chunk of the training data at random.

Create a decision tree using a purity measure such as Gini impurity or entropy based on the specified features and training data.

Repeat steps 1–3 several times to construct a forest of decision trees.

To forecast the class label of a new input, run it through the forest's trees and aggregate the results.

Because each tree is built on a random subset of the training data and input features, the Random Forest Classifier reduces overfitting. It also handles missing values and noisy data effectively, because the majority voting of the trees may balance out individual tree mistakes.

Random Forest Classifier also gives helpful information on feature relevance, which is a measure of how beneficial each feature is for classification. This data can be used to choose features and preprocess data.

Working Involves Following Steps:

- ➢ Above Four machine models are used here to predict the similarity score between the data extracted from Job Data as well as Resume Data
- ➢ In the First Stage We extracted the Text Data from Job Data and Resume
    - o Data and Removed the stop words,punctuation and removed Null values.
    - o Split the Data into Train and Test set for each of the machine learning model (LR,Naïve Bayes,Decision Tree, Randomforest)
    - o Trained the model with the trained dataset.
    - o Predicted similarity score for the data in job data and resume data
    - o Finally displayed the results of precision,recall values for each of Machine Learning Model(i.e Naïve Bayes, Decision Tree, Random Forest,Logistic Regression).

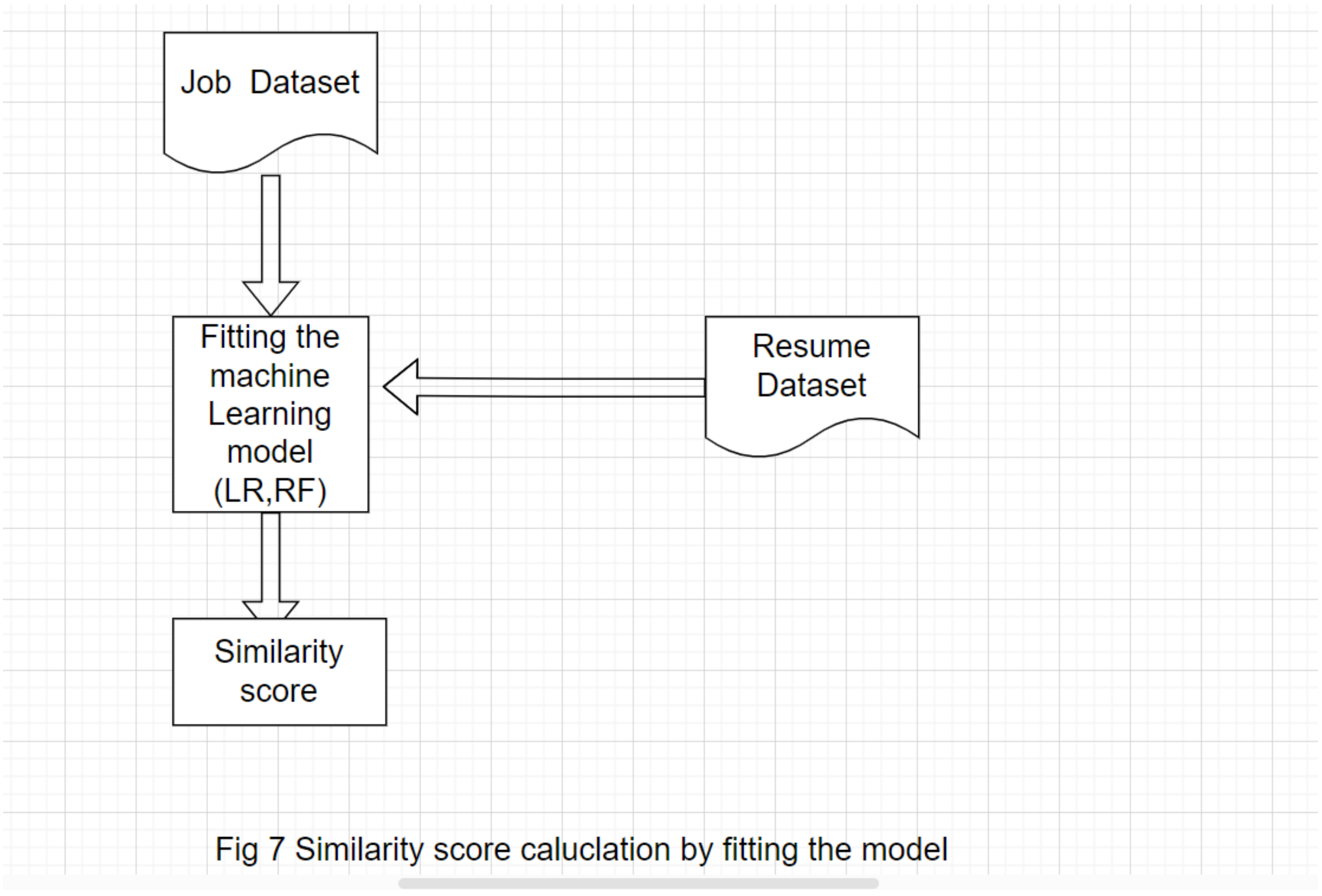
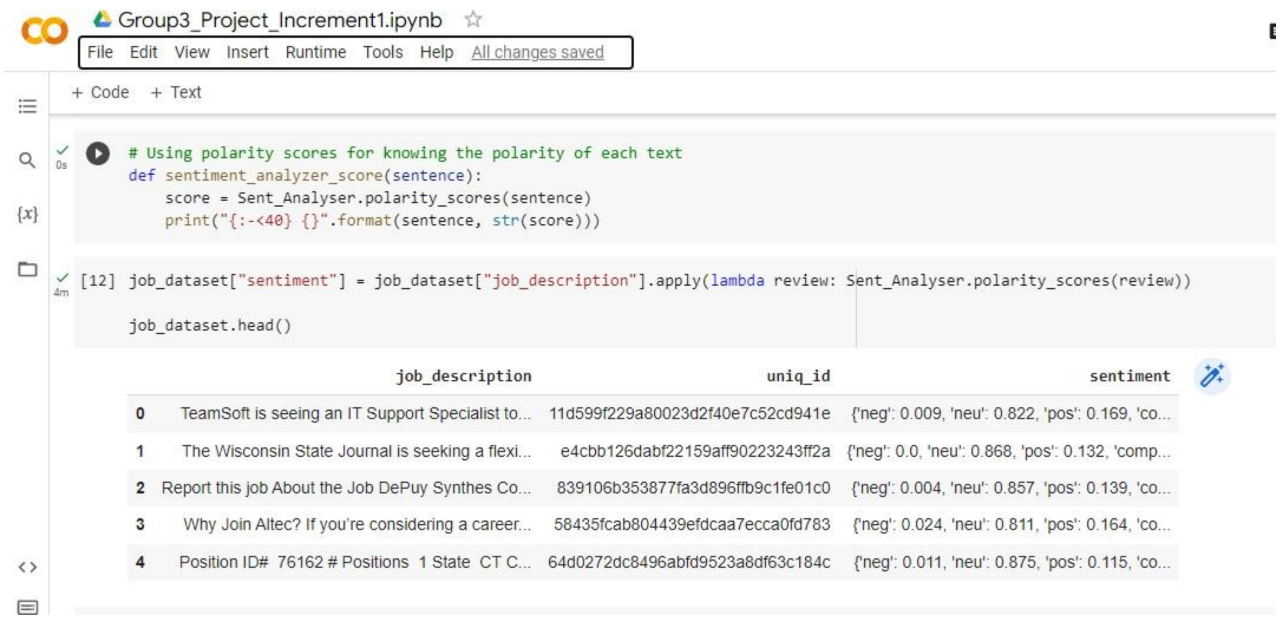


Fig 7 Similarity score caluclation by fitting the model

6. Analysis and Preliminary Results:

In the project increment 1, we mainly performed 3 NLP techniques namely sentiment analysis, text summarization and keyword extraction.

We first performed the VaderSentimentAnalyzer on the job descriptions in job data set and the results are following:

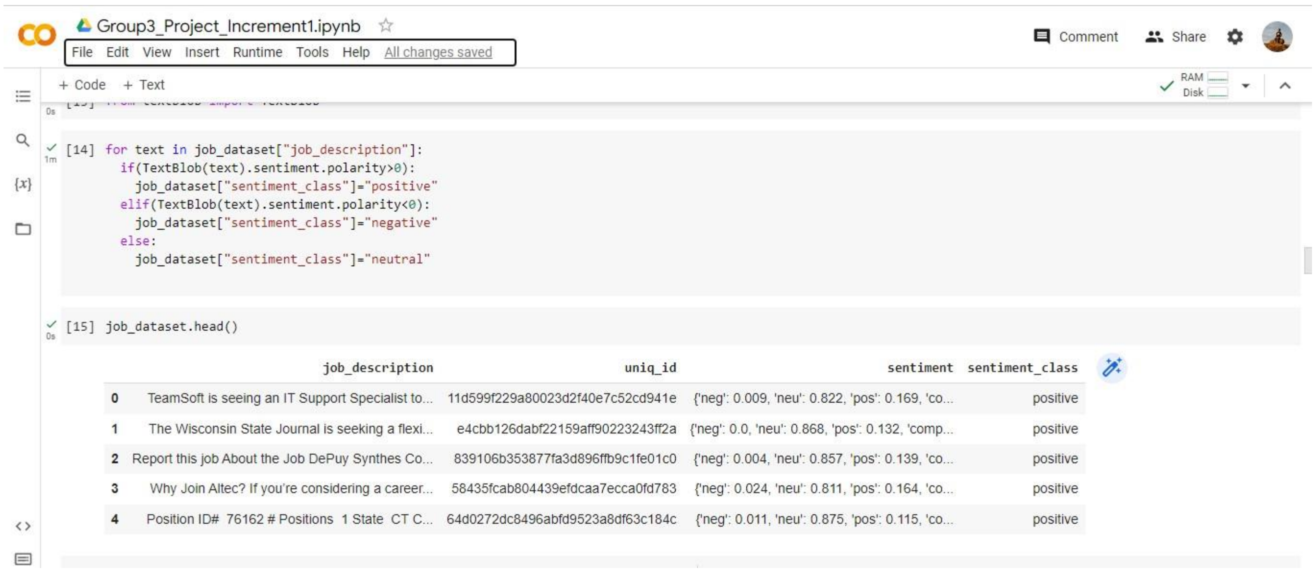

Group3_Project_Increment1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
# Using polarity scores for knowing the polarity of each text
def sentiment_analyzer_score(sentence):
    score = Sent_Analyser.polarity_scores(sentence)
    print("{:-<40} {}".format(sentence, str(score)))
```

```
[12] job_dataset["sentiment"] = job_dataset["job_description"].apply(lambda review: Sent_Analyser.polarity_scores(review))

     job_dataset.head()
```

	job_description	uniq_id	sentiment
0	TeamSoft is seeing an IT Support Specialist to...	11d599f229a80023d2f40e7c52cd941e	{'neg': 0.009, 'neu': 0.822, 'pos': 0.169, 'co...
1	The Wisconsin State Journal is seeking a flexi...	e4cbb126dabf22159aff90223243ff2a	{'neg': 0.0, 'neu': 0.868, 'pos': 0.132, 'comp...
2	Report this job About the Job DePuy Synthes Co...	839106b353877fa3d896ffb9c1fe01c0	{'neg': 0.004, 'neu': 0.857, 'pos': 0.139, 'co...
3	Why Join Altec? If you're considering a career...	58435fcab804439efdcaa7ecca0fd783	{'neg': 0.024, 'neu': 0.811, 'pos': 0.164, 'co...
4	Position ID# 76162 # Positions 1 State CT C...	64d0272dc8496abfd9523a8df63c184c	{'neg': 0.011, 'neu': 0.875, 'pos': 0.115, 'co...

We also performed sentiment analysis using textblob and the resulted classification is as follows:

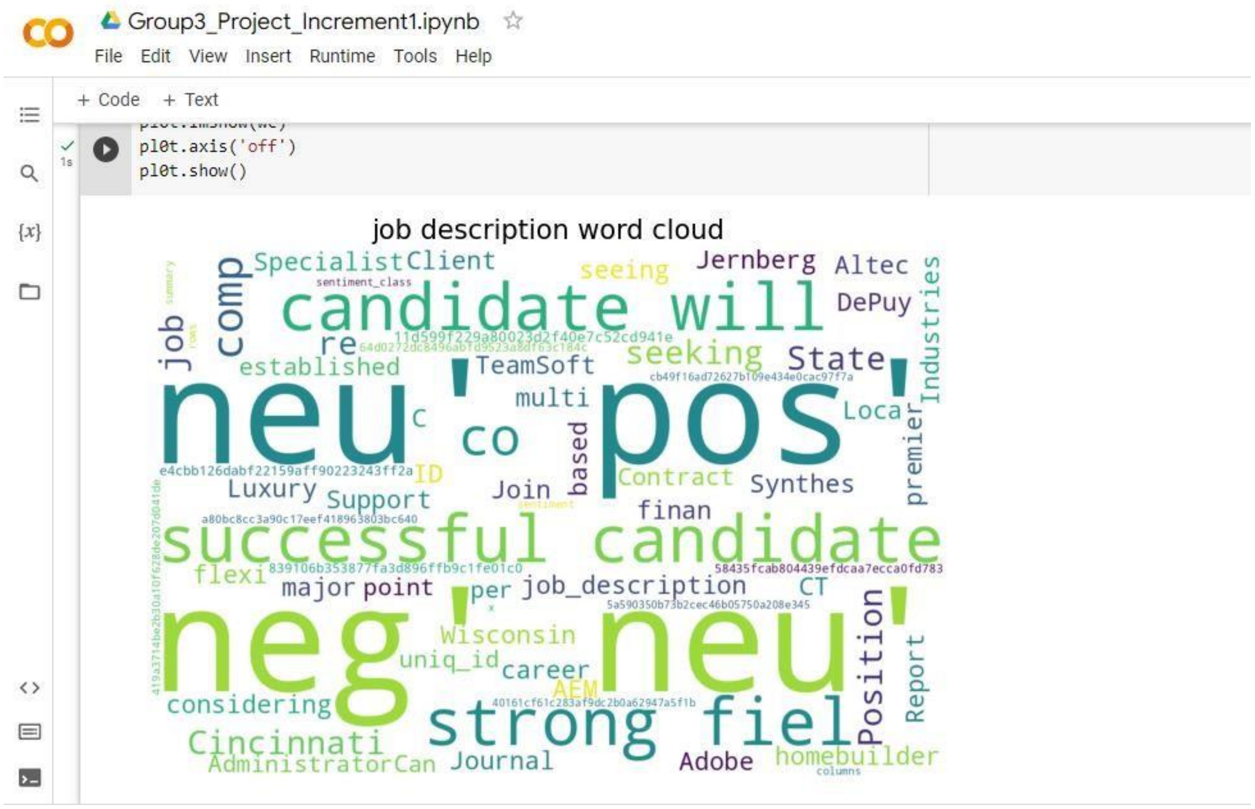

Group3_Project_Increment1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk

```
[14] for text in job_dataset["job_description"]:
       if(TextBlob(text).sentiment.polarity>0):
         job_dataset["sentiment_class"]="positive"
       elif(TextBlob(text).sentiment.polarity<0):
         job_dataset["sentiment_class"]="negative"
       else:
         job_dataset["sentiment_class"]="neutral"
```

```
[15] job_dataset.head()
```

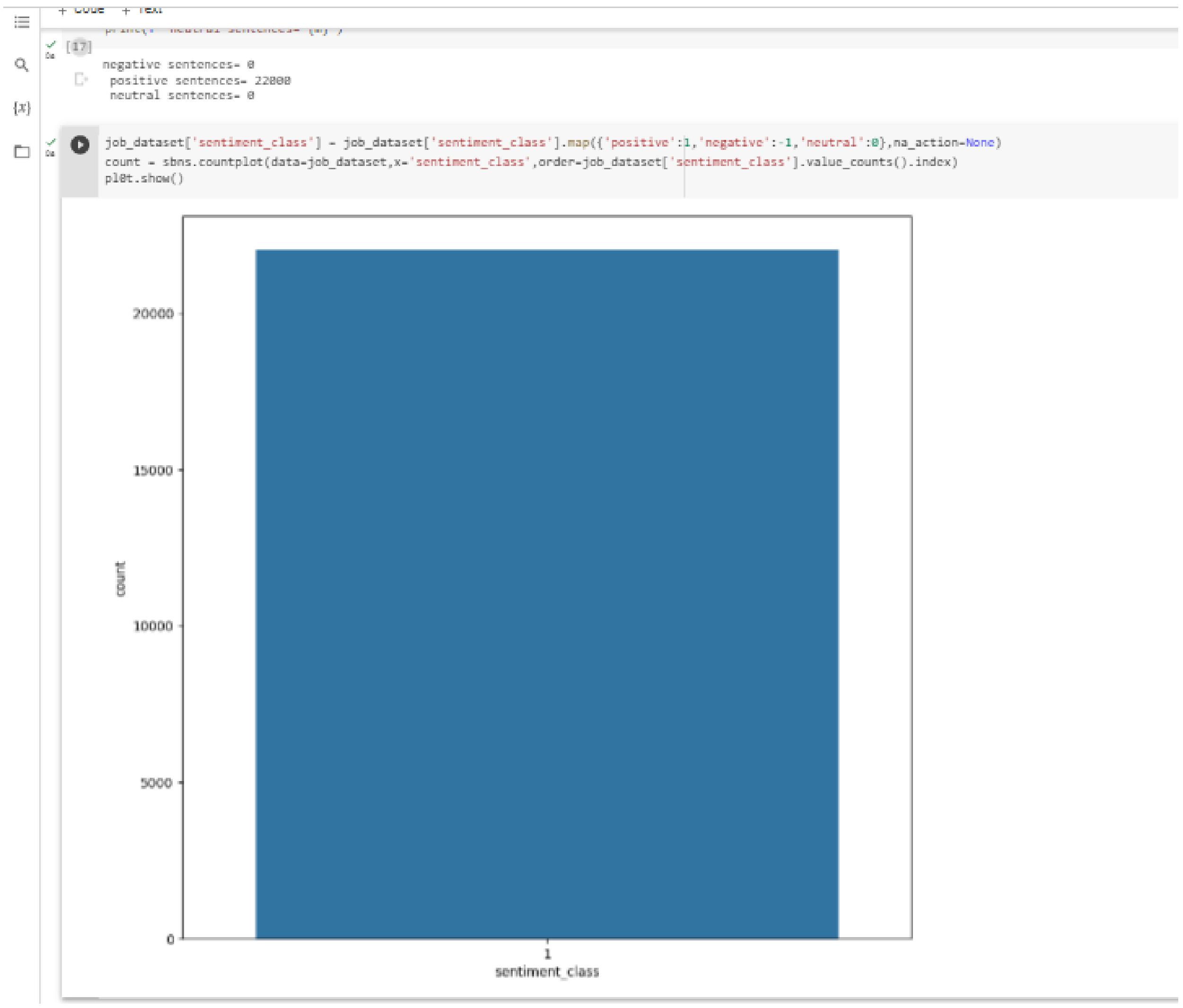
	job_description	uniq_id	sentiment	sentiment_class
0	TeamSoft is seeing an IT Support Specialist to...	11d599f229a80023d2f40e7c52cd941e	{'neg': 0.009, 'neu': 0.822, 'pos': 0.169, 'co...	positive
1	The Wisconsin State Journal is seeking a flexi...	e4cbb126dabf22159aff90223243ff2a	{'neg': 0.0, 'neu': 0.868, 'pos': 0.132, 'comp...	positive
2	Report this job About the Job DePuy Synthes Co...	839106b353877fa3d896ffb9c1fe01c0	{'neg': 0.004, 'neu': 0.857, 'pos': 0.139, 'co...	positive
3	Why Join Altec? If you're considering a career...	58435fcab804439efdcaa7ecca0fd783	{'neg': 0.024, 'neu': 0.811, 'pos': 0.164, 'co...	positive
4	Position ID# 76162 # Positions 1 State CT C...	64d0272dc8496abfd9523a8df63c184c	{'neg': 0.011, 'neu': 0.875, 'pos': 0.115, 'co...	positive

The word cloud resulted from the job description is:



```
plt.imshow(wc)
plt.axis('off')
plt.show()
```

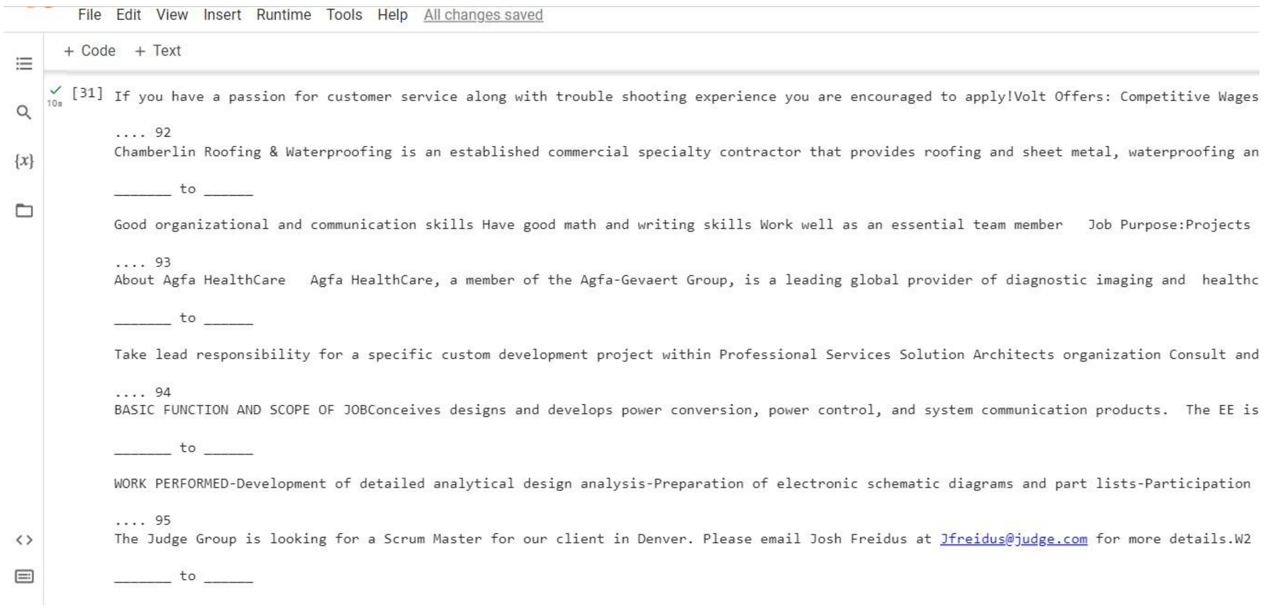
Also, we can see that almost all job descriptions are positive toned only:



```
negative sentences= 0
 positive sentences= 22000
 neutral sentences= 0
```

```
job_dataset['sentiment_class'] = job_dataset['sentiment_class'].map({'positive':1,'negative':-1,'neutral':0},na_action=None)
count = sbns.countplot(data=job_dataset,x='sentiment_class',order=job_dataset['sentiment_class'].value_counts().index)
pl0t.show()
```

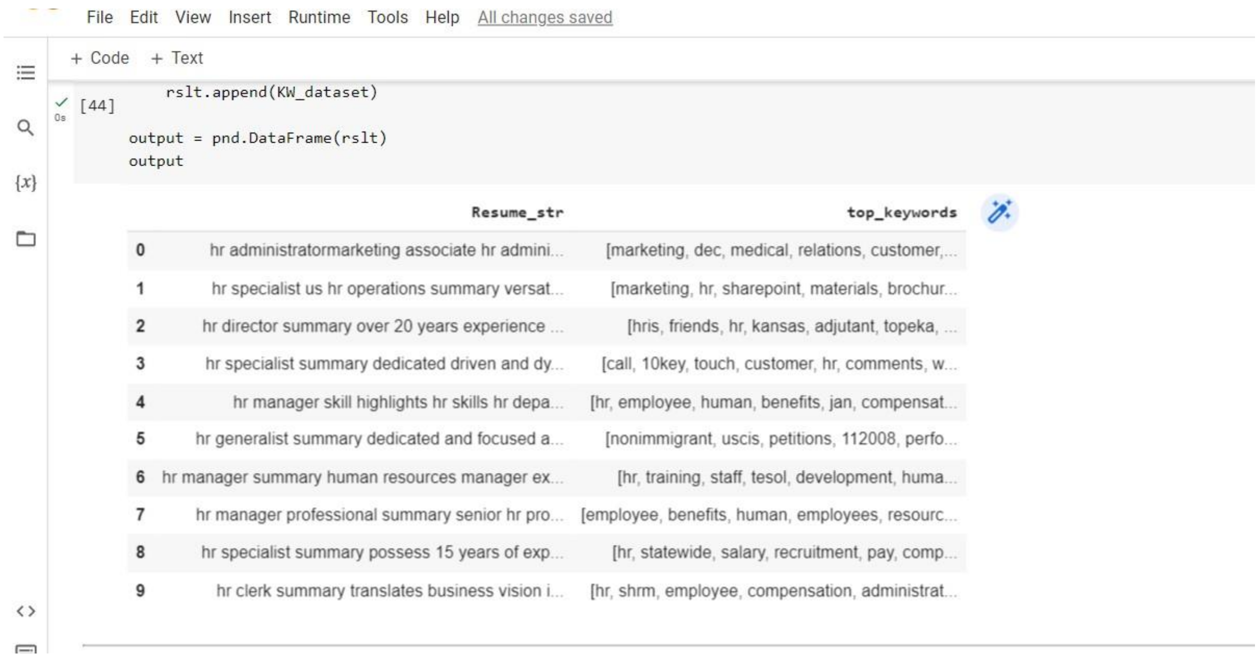
The results from text summarization on the first 100 job descriptions is as shown below:


File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[31]
```
If you have a passion for customer service along with trouble shooting experience you are encouraged to apply!Volt Offers: Competitive Wages

.... 92
Chamberlin Roofing & Waterproofing is an established commercial specialty contractor that provides roofing and sheet metal, waterproofing an

_______ to ______

Good organizational and communication skills Have good math and writing skills Work well as an essential team member   Job Purpose:Projects

.... 93
About Agfa HealthCare   Agfa HealthCare, a member of the Agfa-Gevaert Group, is a leading global provider of diagnostic imaging and  healthc

_______ to ______

Take lead responsibility for a specific custom development project within Professional Services Solution Architects organization Consult and

.... 94
BASIC FUNCTION AND SCOPE OF JOBConceives designs and develops power conversion, power control, and system communication products.  The EE is

_______ to ______

WORK PERFORMED-Development of detailed analytical design analysis-Preparation of electronic schematic diagrams and part lists-Participation

.... 95
The Judge Group is looking for a Scrum Master for our client in Denver. Please email Josh Freidus at Jfreidus@judge.com for more details.W2

_______ to ______
```

Also, the output from the keyword extraction on the resume dataset is as follows:

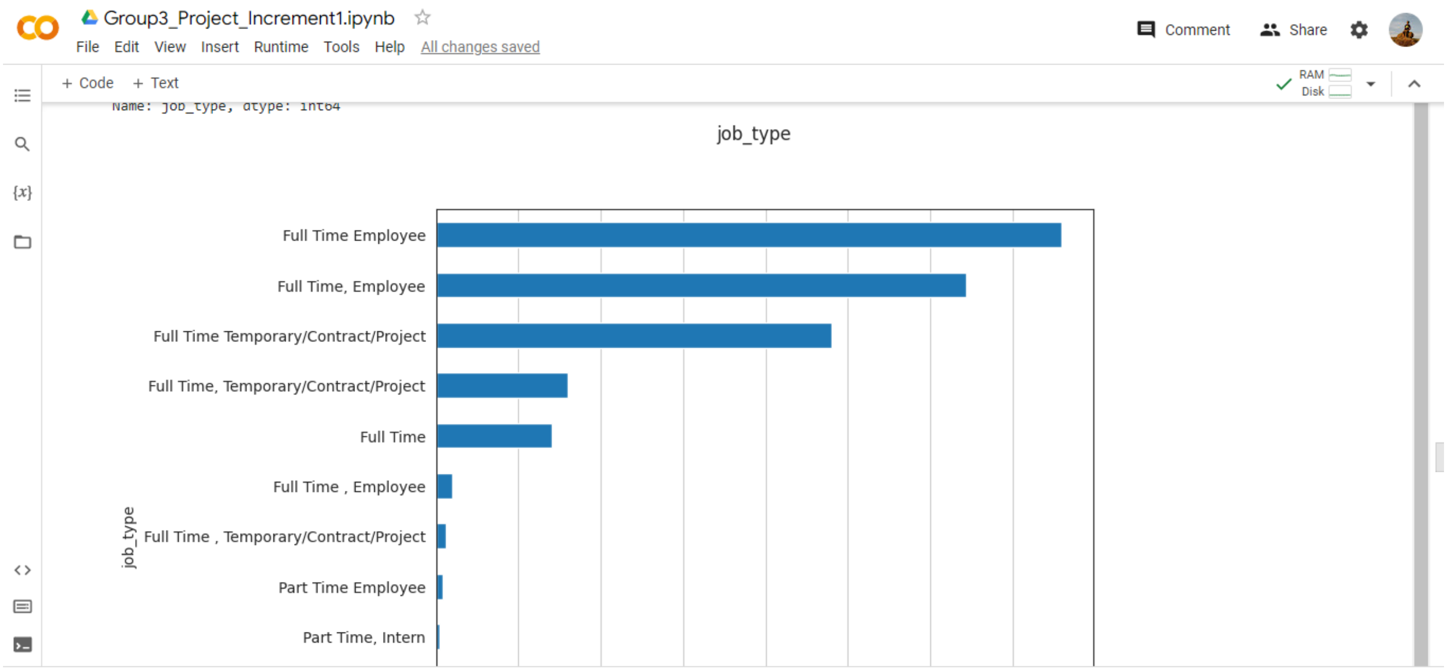

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[44]
```
    rslt.append(KW_dataset)

output = pnd.DataFrame(rslt)
output
```

	Resume_str	top_keywords
0	hr administratormarketing associate hr admini...	[marketing, dec, medical, relations, customer,...
1	hr specialist us hr operations summary versat...	[marketing, hr, sharepoint, materials, brochur...
2	hr director summary over 20 years experience ...	[hris, friends, hr, kansas, adjutant, topeka, ...
3	hr specialist summary dedicated driven and dy...	[call, 10key, touch, customer, hr, comments, w...
4	hr manager skill highlights hr skills hr depa...	[hr, employee, human, benefits, jan, compensat...
5	hr generalist summary dedicated and focused a...	[nonimmigrant, uscis, petitions, 112008, perfo...
6	hr manager summary human resources manager ex...	[hr, training, staff, tesol, development, huma...
7	hr manager professional summary senior hr pro...	[employee, benefits, human, employees, resourc...
8	hr specialist summary possess 15 years of exp...	[hr, statewide, salary, recruitment, pay, comp...
9	hr clerk summary translates business vision i...	[hr, shrm, employee, compensation, administrat...

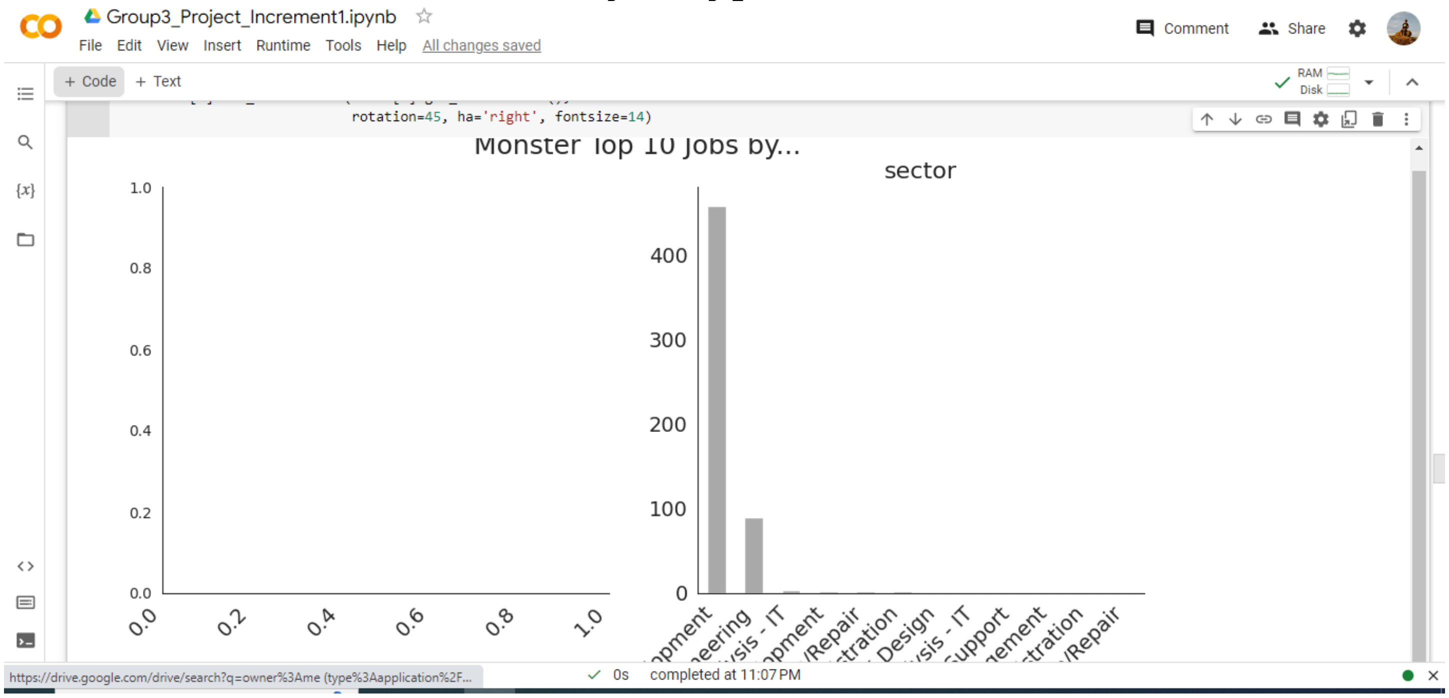
Here , we can observe the labels under which the job descriptions are classified. As there are huge number of job descriptions belonging to different sectors, we have limited our dataset to IT related sector and labelled the data.



```
Name: job_type, dtype: int64
```

Group3_Project_Increment1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
job_type
Full Time Employee
Full Time, Employee
Full Time Temporary/Contract/Project
Full Time, Temporary/Contract/Project
Full Time
Full Time , Employee
Full Time , Temporary/Contract/Project
Part Time Employee
Part Time, Intern
job_type

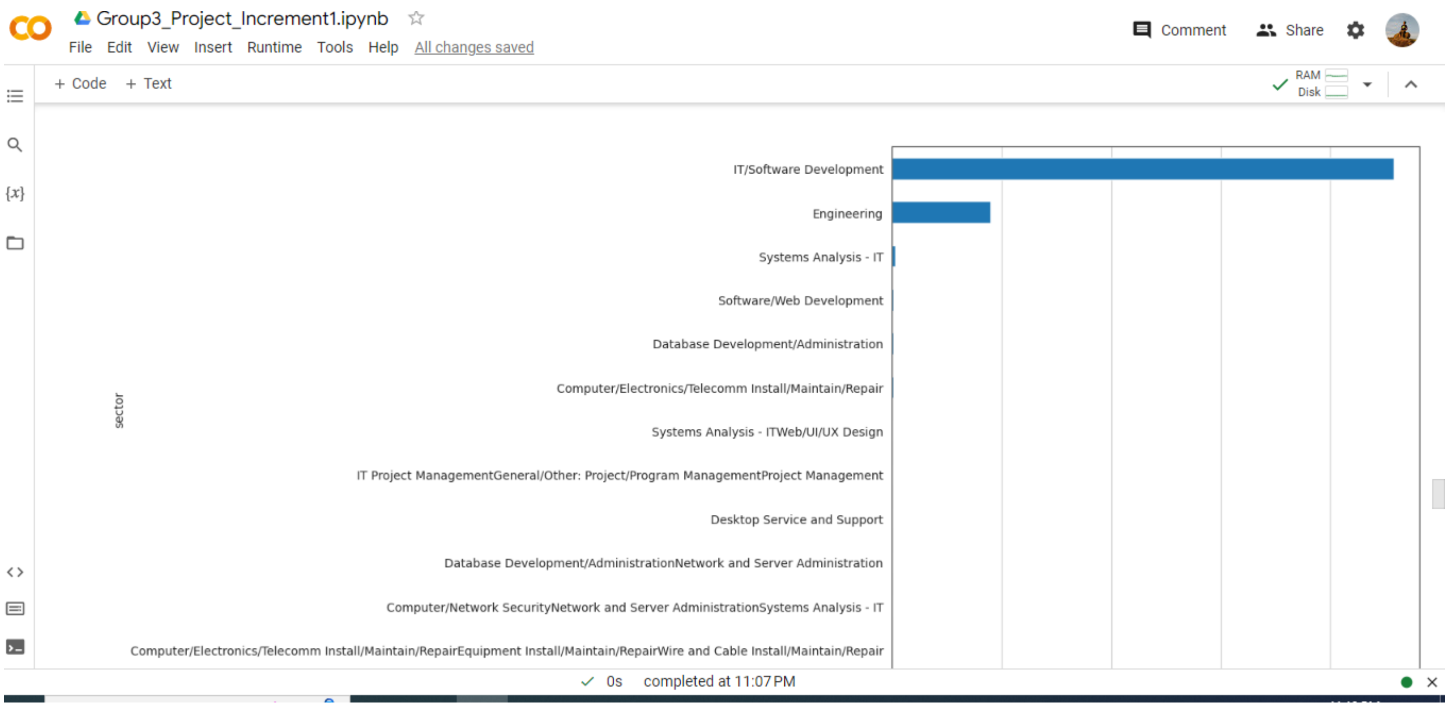
This is labelled result based on job_types.



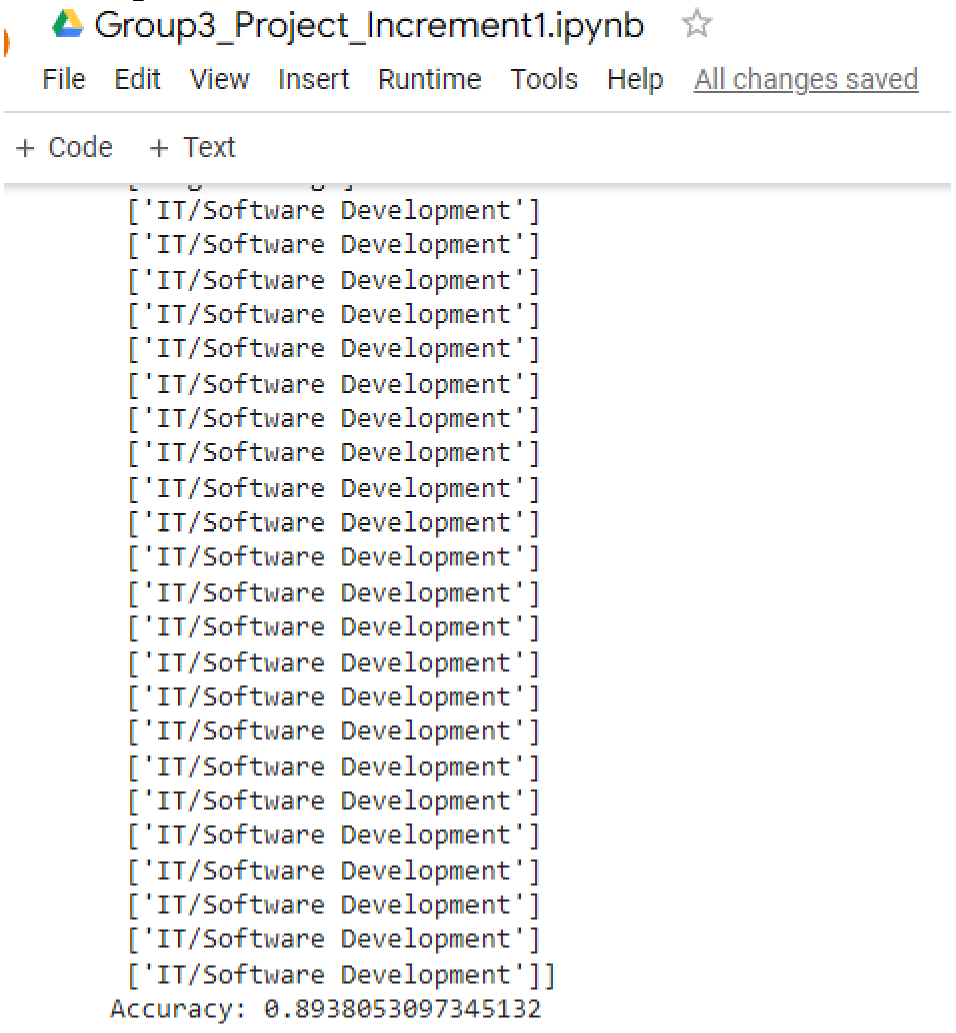
```
rotation=45, ha='right', fontsize=14)
```

Group3_Project_Increment1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Monster Top 10 Jobs by...
sector
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
400
300
200
100
0
https://drive.google.com/drive/search?q=owner%3Ame (type%3Aapplication%2F...
0s completed at 11:07 PM

This is the result obtained by labelling on job sector.

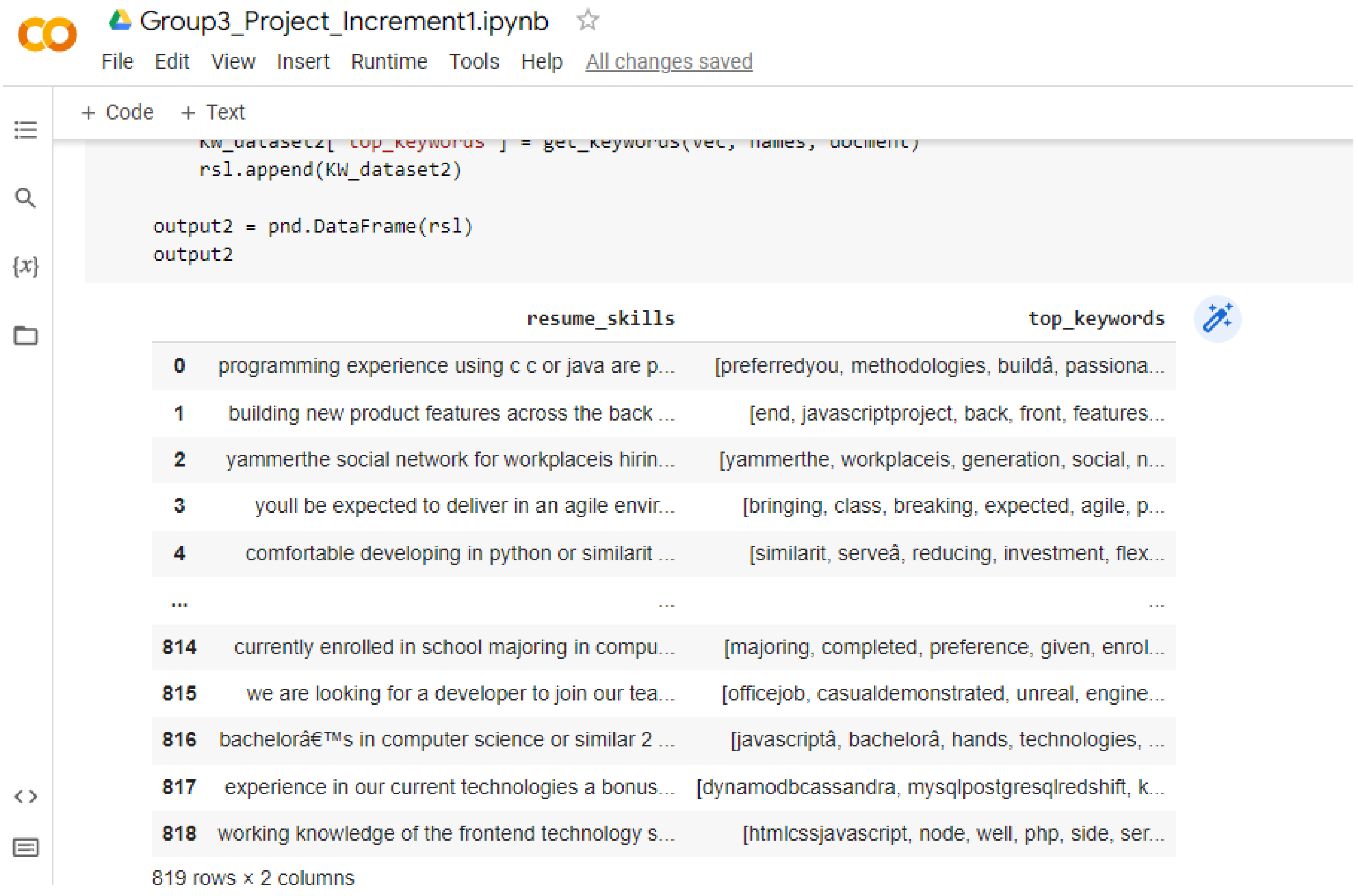


We can notice there are more records which falls under IT/software developemet label.


Group3_Project_Increment1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']
 ['IT/Software Development']]
Accuracy: 0.8938053097345132
```

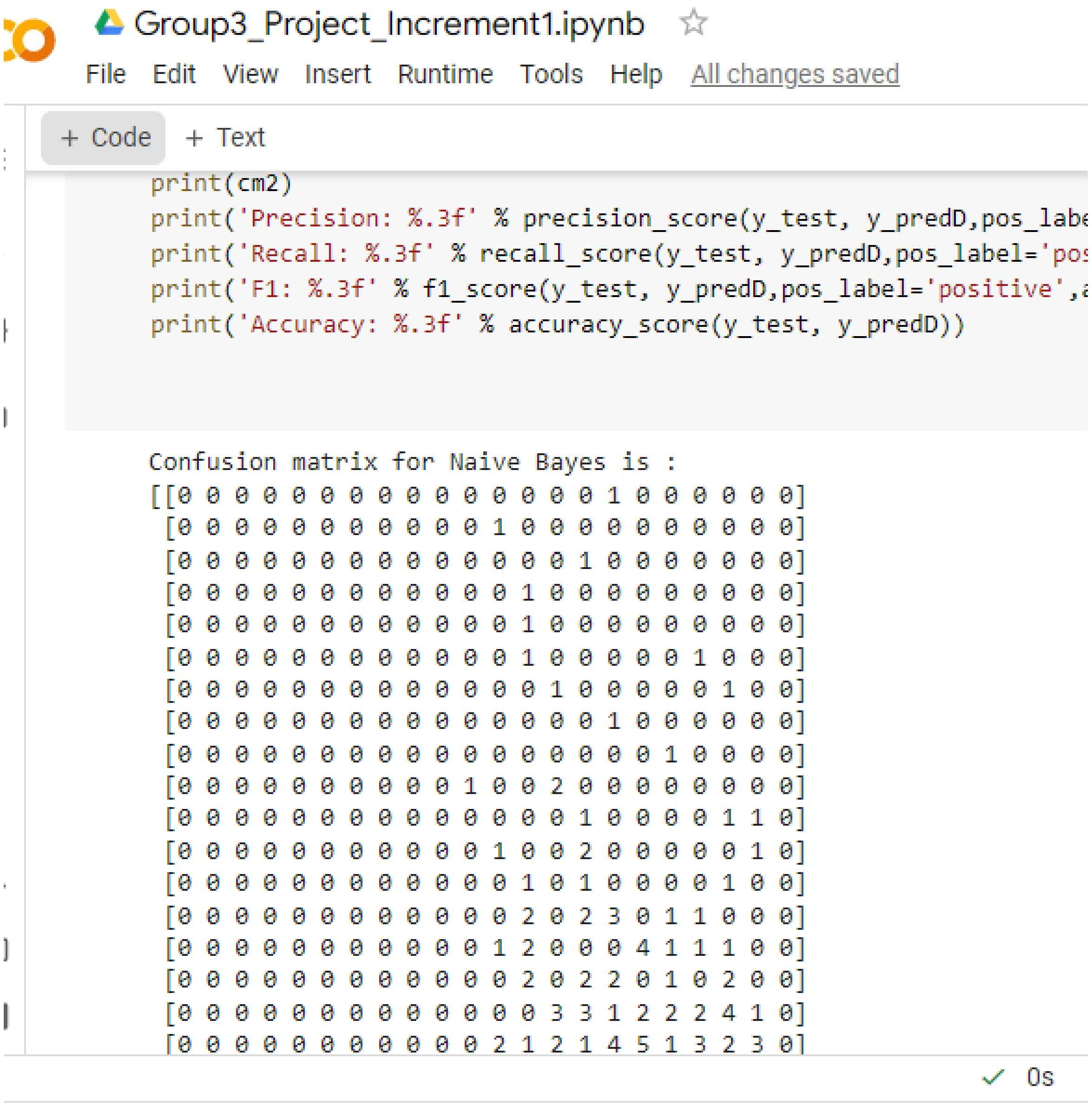
This is where we perform and extract the keywords from IT related resume skills.


Group3_Project_Increment1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
        KW_dataset2['top_keywords'] = get_keywords(vec, names, document)
        rsl.append(KW_dataset2)

    output2 = pnd.DataFrame(rsl)
    output2
```

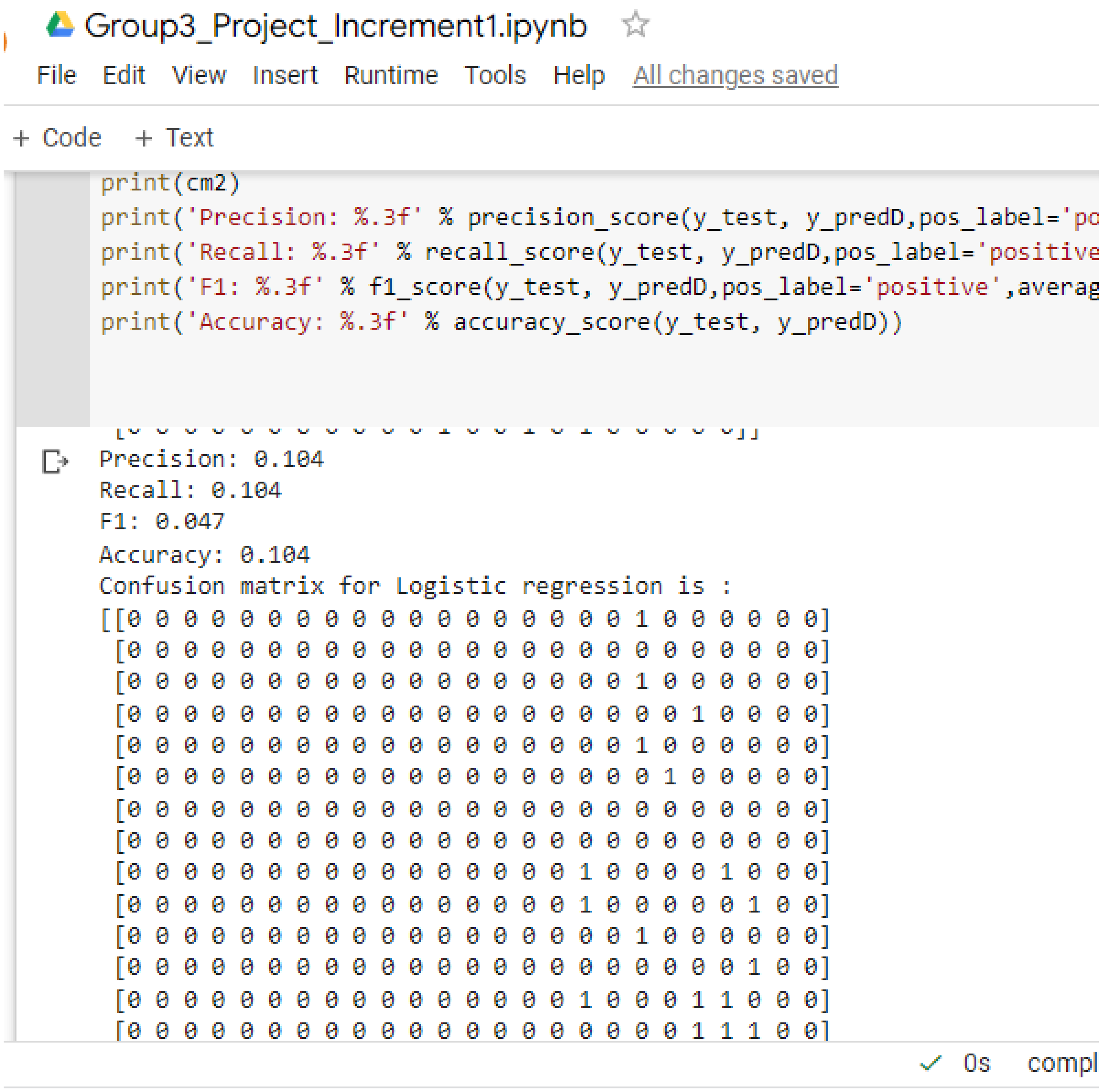
	resume_skills	top_keywords
0	programming experience using c c or java are p...	[preferredyou, methodologies, buildâ, passiona...
1	building new product features across the back ...	[end, javascriptproject, back, front, features...
2	yammerthe social network for workplaceis hirin...	[yammerthe, workplaceis, generation, social, n...
3	youll be expected to deliver in an agile envir...	[bringing, class, breaking, expected, agile, p...
4	comfortable developing in python or similarit ...	[similarit, serveâ, reducing, investment, flex...
...	...	...
814	currently enrolled in school majoring in compu...	[majoring, completed, preference, given, enrol...
815	we are looking for a developer to join our tea...	[officejob, casualdemonstrated, unreal, engine...
816	bachelorâ€™s in computer science or similar 2 ...	[javascriptâ, bachelorâ, hands, technologies, ...
817	experience in our current technologies a bonus...	[dynamodbcassandra, mysqlpostgresqlredshift, k...
818	working knowledge of the frontend technology s...	[htmlcssjavascript, node, well, php, side, ser...

819 rows × 2 columns


Group3_Project_Increment1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
✓ 0s

```
print(cm2)
print('Precision: %.3f' % precision_score(y_test, y_predD,pos_labe
print('Recall: %.3f' % recall_score(y_test, y_predD,pos_label='pos
print('F1: %.3f' % f1_score(y_test, y_predD,pos_label='positive',a
print('Accuracy: %.3f' % accuracy_score(y_test, y_predD))
```

```
Confusion matrix for Naive Bayes is :
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 1 0 0 2 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 1 0 0 2 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 3 0 1 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 4 1 1 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 0 1 0 2 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 1 2 2 2 4 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 2 1 2 1 4 5 1 3 2 3 0]
```


Group3_Project_Increment1.ipynb

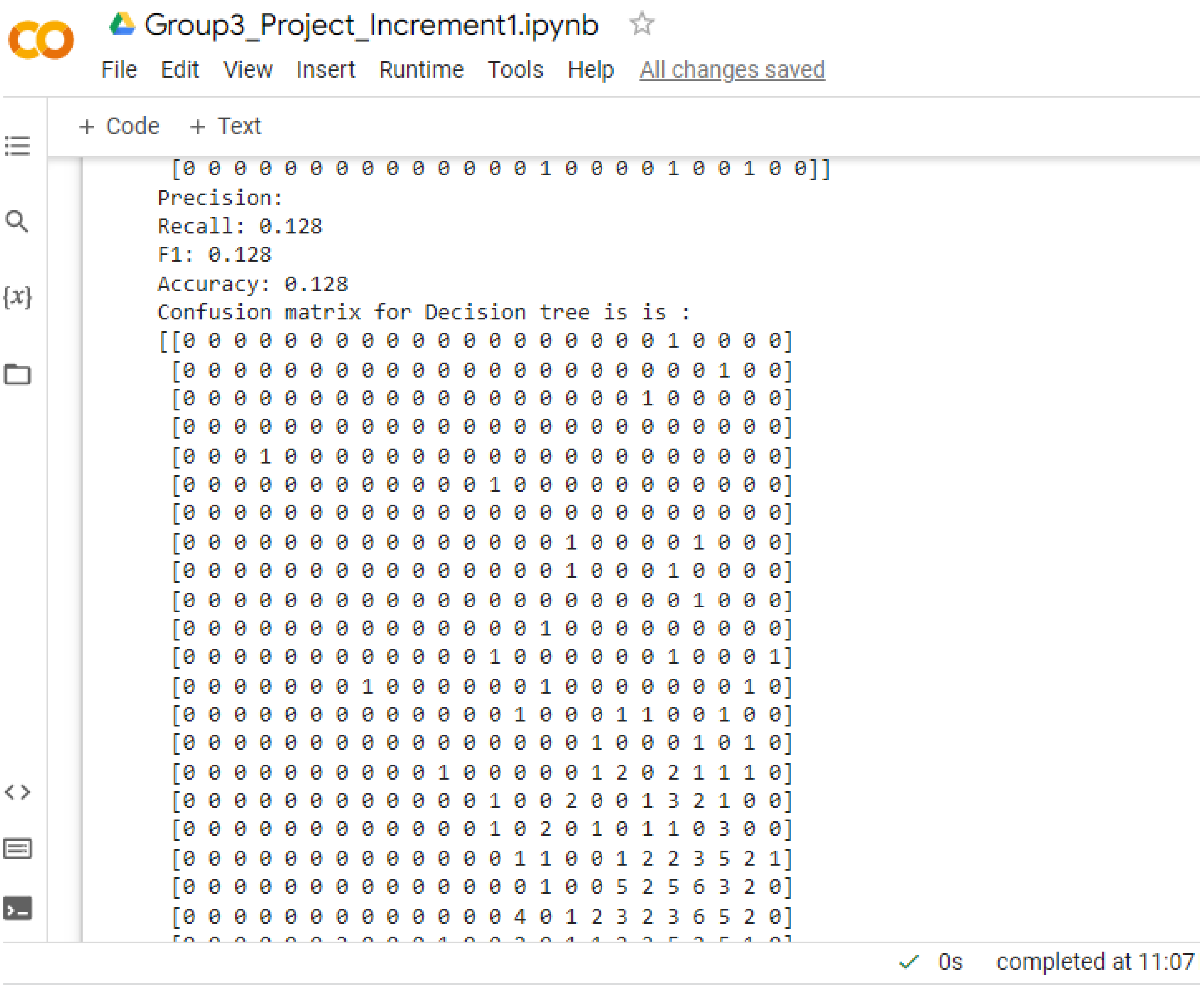
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
print(cm2)
print('Precision: %.3f' % precision_score(y_test, y_predD,pos_label='po
print('Recall: %.3f' % recall_score(y_test, y_predD,pos_label='positive
print('F1: %.3f' % f1_score(y_test, y_predD,pos_label='positive',averag
print('Accuracy: %.3f' % accuracy_score(y_test, y_predD))
```

```
Precision: 0.104
Recall: 0.104
F1: 0.047
Accuracy: 0.104
Confusion matrix for Logistic regression is :
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0]
```

✓ 0s compl



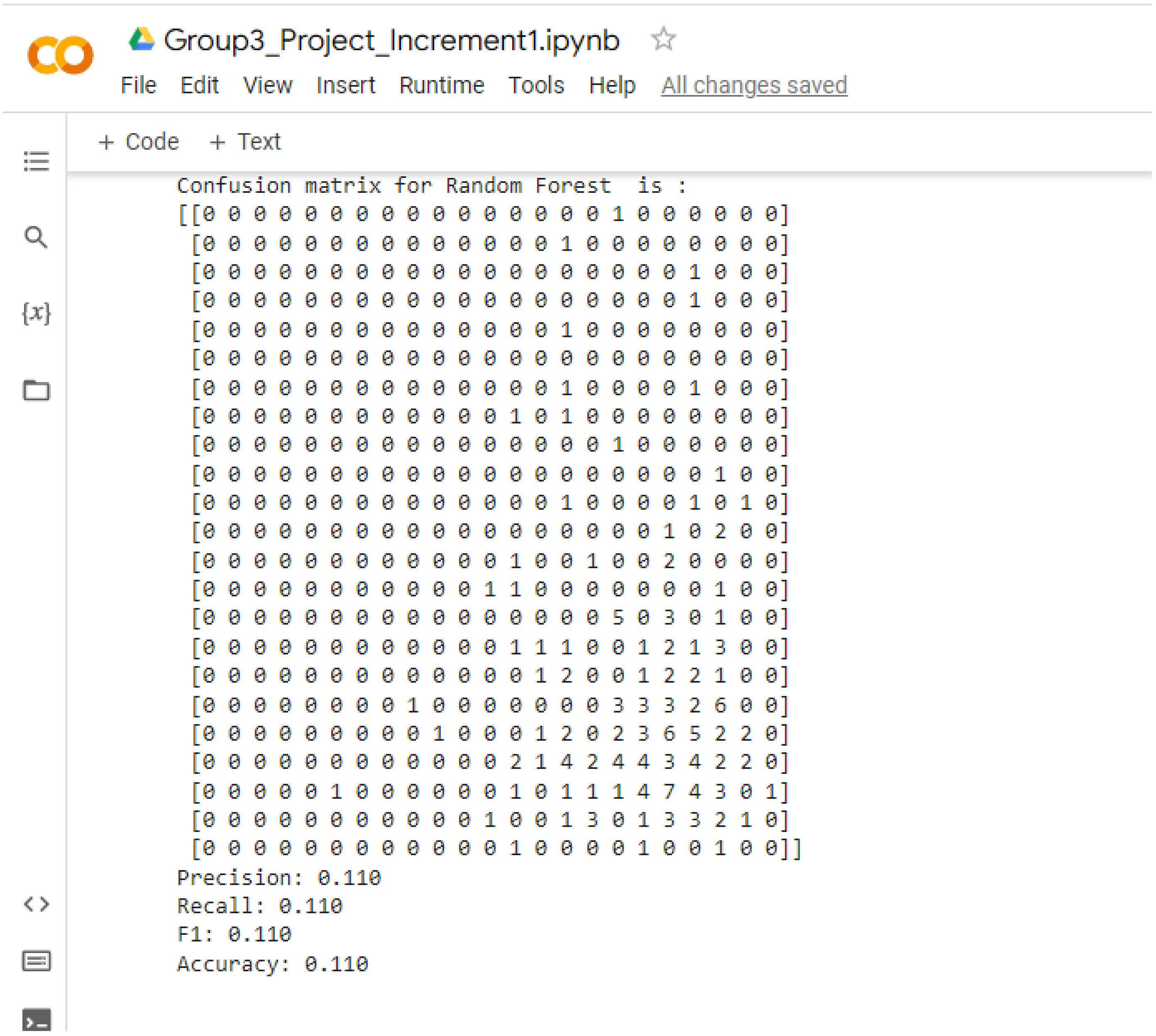
Group3_Project_Increment1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

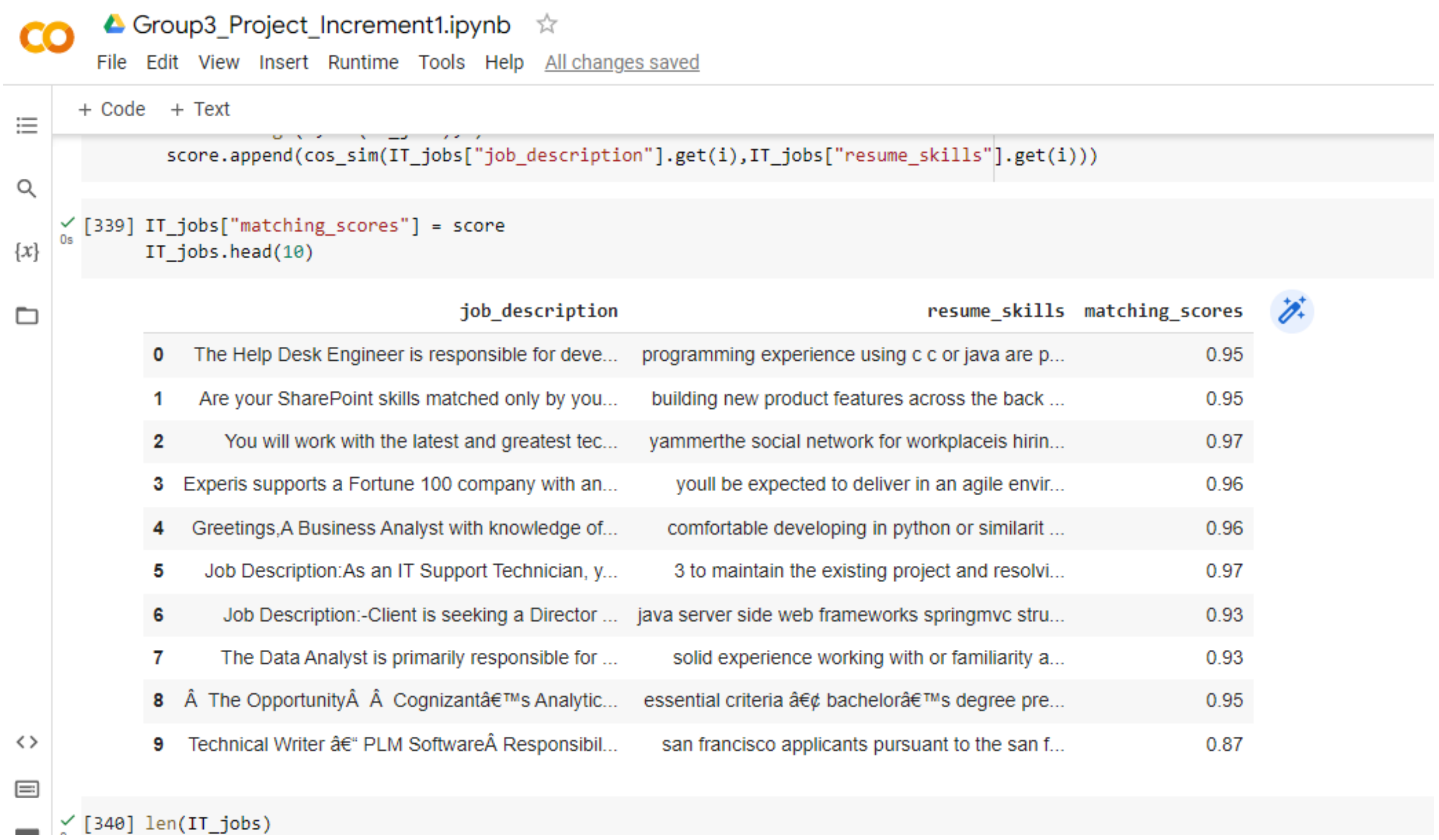
```
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0]]
Precision:
Recall: 0.128
F1: 0.128
Accuracy: 0.128
Confusion matrix for Decision tree is is :
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1]
 [0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 2 0 2 1 1 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 2 0 0 1 3 2 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 2 0 1 0 1 1 0 3 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 2 2 3 5 2 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 5 2 5 6 3 2 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 2 3 2 3 6 5 2 0]
```

✓ 0s completed at 11:07


Group3_Project_Increment1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
Confusion matrix for Random Forest  is :
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 2 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 2 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 3 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 2 1 3 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 1 2 2 1 0 0]
 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 3 3 3 2 6 0 0]
 [0 0 0 0 0 0 0 0 0 1 0 0 0 1 2 0 2 3 6 5 2 2 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 2 1 4 2 4 4 3 4 2 2 0]
 [0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 4 7 4 3 0 1]
 [0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 3 0 1 3 3 2 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0]]
Precision: 0.110
Recall: 0.110
F1: 0.110
Accuracy: 0.110
```

Above are the evealuation metrics resulted from 4 machine learning models used namely Naïve bayes, logistic regression, decision tree, random forest regression models.


Group3_Project_Increment1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

```
score.append(cos_sim(IT_jobs["job_description"].get(i),IT_jobs["resume_skills"].get(i)))

[339] IT_jobs["matching_scores"] = score
      IT_jobs.head(10)
```

	job_description	resume_skills	matching_scores
0	The Help Desk Engineer is responsible for deve...	programming experience using c c or java are p...	0.95
1	Are your SharePoint skills matched only by you...	building new product features across the back ...	0.95
2	You will work with the latest and greatest tec...	yammerthe social network for workplaceis hirin...	0.97
3	Experis supports a Fortune 100 company with an...	youll be expected to deliver in an agile envir...	0.96
4	Greetings,A Business Analyst with knowledge of...	comfortable developing in python or similarit ...	0.96
5	Job Description:As an IT Support Technician, y...	3 to maintain the existing project and resolvi...	0.97
6	Job Description:-Client is seeking a Director ...	java server side web frameworks springmvc stru...	0.93
7	The Data Analyst is primarily responsible for ...	solid experience working with or familiarity a...	0.93
8	Â The OpportunityÂ Â Cognizantâ€™s Analytic...	essential criteria â€¢ bachelorâ€™s degree pre...	0.95
9	Technical Writer â€“ PLM SoftwareÂ Responsibil...	san francisco applicants pursuant to the san f...	0.87

```
[340] len(IT_jobs)
```

Here we can notice the matching score generated from the cosine similarity on job descriptions and resume skills.

7. Project Management:

Work completed

- **Description**

  In this increment, we worked on the major parts of the project which includes the tasks like Sentiment Analysis, Text Summarization and Keyword Extraction. We first invested time on finding relevant datasets for the project. We then performed the pre-processing of these data and did a little exploratory data analysis as well. We then executed the sentiment analysis on the job descriptions and extracted short summaries from them. We also performed a bit of pre processing on the resume dataset and extracted top keywords from the resume descriptions.

- **Responsibility**

Person	Task
Suhas Siddarajgari Tellatakula	Sentiment Analysis, Text Summarization, Keyword Extraction, Report
Vamshi Telukuntla	Sentiment Analysis, Text Summarization, Keyword Extraction, Report
Srikanth Mamillapalli	Sentiment Analysis, Text Summarization, Keyword Extraction, Report
Sairohithvarma Kantem	Sentiment Analysis, Text Summarization, Keyword Extraction, Report

- **Issues/Concerns**

  Here are few issues we faced during the course of project increment 1:

- Finding relevant datasets from plethora of data from Kaggle.
- Having neutral job descriptions.
- Having null entries in datasets.
- Having improper/ unidentified notations or symbols in the datasets.

Work completed in this increment:

- **Description**

In the next increment, we aim to achieve the remaining parts of the project which includes label classification on job descriptions, building a machine learning model to suggest resumes relevant to the job descriptions. We will try out different techniques to find the best possible model to generate best accurate results.

- **Responsibility**

Person	Task
Suhas Siddarajgari Tellatakula	Label classification, Machine Learning models, report
Vamshi Telukuntla	Label classification, Machine Learning models, report
Srikanth Mamillapalli	Label classification, Machine Learning models, report
Sairohithvarma Kantem	Label classification, Machine Learning models, report

- **Issues/Concerns**

Here are few issues which we can expect as of now:

  - Considering various Machine learning models.

- Choosing the model which gives best accuracy.
- Not having enough relevant data.

8.References/Bibliography:

- https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset
- https://www.kaggle.com/code/residentmario/exploring-monster-com-job-postings
- https://towardsdatascience.com/keyword-extraction-process-in-python-with-natural-language-processing-nlp-d769a9069d5c
- https://www.topcoder.com/thrive/articles/text-summarization-in-nlp
- https://www.analyticsvidhya.com/blog/2020/11/words-that-matter-a-simple-guide-to-keyword-extraction-in-python/
- https://www.kaggle.com/code/apoorvgupta25/sentiment-analysis-using-logistics-regression/notebook

- https://www.kaggle.com/code/ahmed121ashraf131/nlp-summerization/notebook
- https://www.kaggle.com/code/akhatova/extract-keywords/notebook
- https://www.kaggle.com/code/vicely07/sentiment-analysis-of-city-of-la-job-postings
- https://www.kaggle.com/code/nezarabdilahprakasa/matching-cv-to-job-description-using-python
- https://www.kaggle.com/code/sanabdriss/nlp-extract-skills-from-job- descriptions
- https://www.kaggle.com/code/mohiteud/ml-algorithms-for-text-classification/notebook
- https://simonhessner.de/why-are-precision-recall-and-f1-score-equal-when-using-micro-averaging-in-a-multi-class-problem/