

Més que un team

INDEX

1. Overview
2. Exploratory Data Analysis
 - 2.1 Data Summary
 - 2.2 Other Visualization
3. Fraud Detection
 - 3.1 Cycles in Letter of Recommendation
 - 3.2 Detection of Exaggerated Letter of Recommendation
 - 3.3 Correlation of Letter of Recommendations with Resume
4. Data Analysis (Connectivity)
 - 4.1 Letter of Recommendations
 - 4.2 Work Experience and Education
5. Profile
 - 5.1 Years of Experience
 - 5.2 Number of Companies Worked with
 - 5.3 Last Position
6. HR Decision Support Dashboard
7. Conclusion
8. Annexure

OVERVIEW:

AI Resume Analyzer:

An AI Resume Analyzer is a tool that uses artificial intelligence to scan and evaluate resumes quickly and efficiently. It applies natural language processing (NLP) to analyze the structure, skills, and qualifications listed, ensuring they match job requirements. This tool can help recruiters save time by automatically ranking candidates based on their qualifications, and experience, detecting fraud entries, and relevance to the job description. It also provides insights such as highlighting missing keywords, suggesting improvements, and ensuring resumes are ATS (Applicant Tracking System) compliant. For job seekers, it enhances their chances of getting noticed by tailoring their resume to specific roles.

Task:

The task involves developing AI-powered tools to assist "Satya," an AI system designed for HR decision-making. Specifically, the goals include **Resume and Recommendation Screening:** Helping Satya analyze resumes and recommendation letters for accuracy and authenticity. **Fraud Detection:** Building mechanisms to detect fraudulent claims in resumes and reciprocal endorsements in recommendations, including spotting vague language and suspicious patterns. **Data Analysis:** Uncovering meaningful insights about candidates' professional networks and connections, and identifying key influencers or well-connected individuals. **HR Decision Support Dashboard:** Designing a user-friendly dashboard that includes fraud alerts and a candidate ranking system, helping HR teams make informed hiring decisions, for which we were given 1000 entries of applicants having unique IDs and recommendation letters.

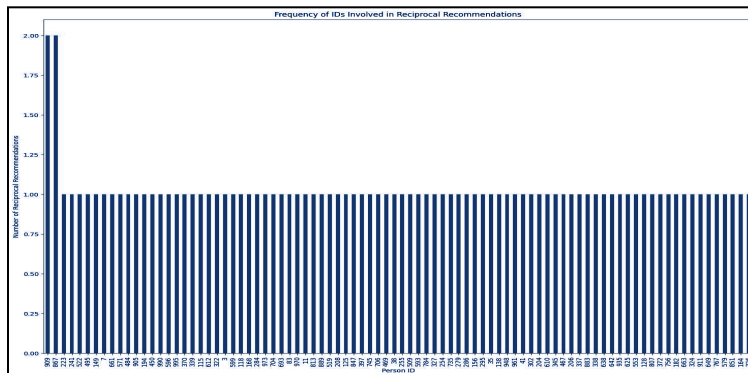
Approach:

We have divided the entire task into three primary scores which are Fraudulent Score, Connectivity Score, and Profile Score. For the very first score, we have employed the **DistilBERT** large language model which segregates the LORs that seem to have some kind of evident exaggeration. We have also found those IDs that have reciprocal recommendations and flagged them in binary format and found similarities between the LORs received and the resume. For the connectivity quantification of an ID, we have developed various features like the number of genuine LORs and the number of genuine LORs given alongside the overlap between the education or a certain work experience between the i^{th} and the j^{th} ID person. Finally, for the Profile Score, we have quantified the features like the number of years of experience and number of companies along with the last position held by the person alongside the Domain of the person.

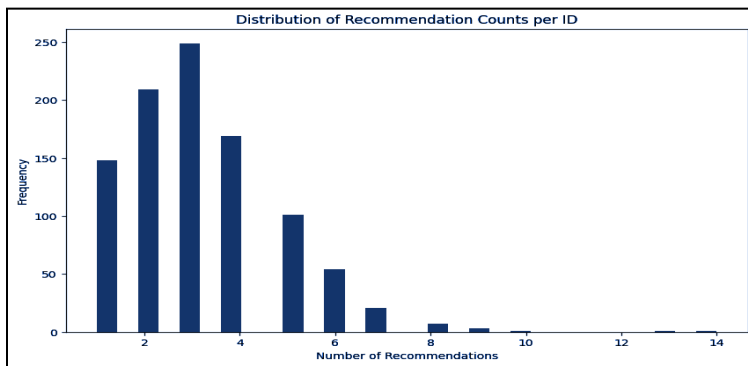
In total, these 3 scores help us determine if the candidate is the perfect fit for a given role and according to the demands of HR for a particular role and needs.

EXPLORATORY DATA ANALYSIS

DATA SUMMARY



The corresponding bar graph shows the frequency of IDs involved in Reciprocal Recommendations with two IDs, 909 and 867, having 2 reciprocal recommendations, while others have 1. Those IDs not engaged in reciprocal recommendations are dropped from the bar graph.

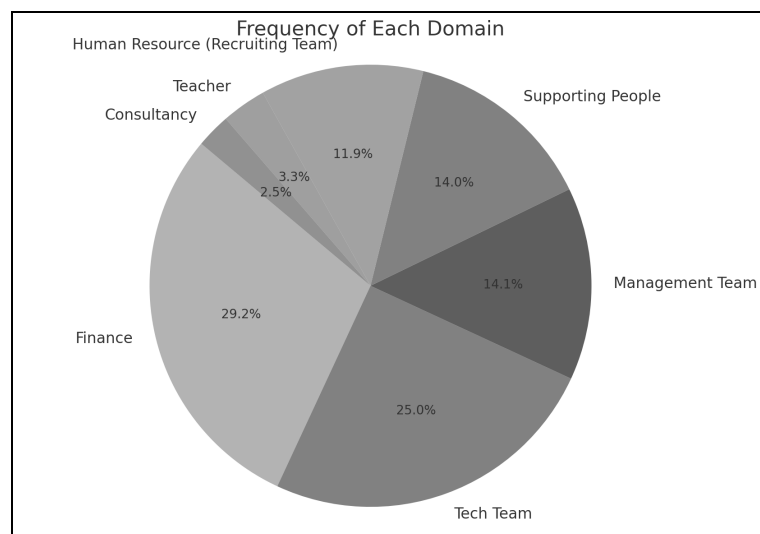


This graph gives us the distribution of the number of LORs received by an ID across all the 0-999 IDs available.

OTHER VISUALIZATIONS



A nice way to visualize the information about this is with a word cloud where the frequency of each tag is shown with font size and color. The word cloud shows the most common tags in the letter of recommendation for IDs 1 and 2 having 6 texts.



This pie chart depicts the percentage of ID in a particular domain. We have a total of 8 domains which are devised based on the structure of a company.

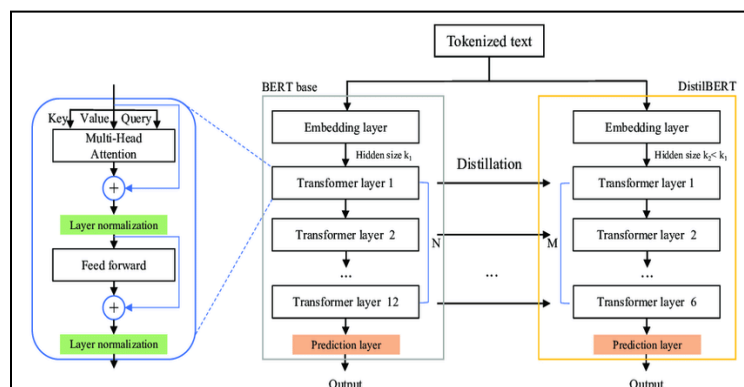
FRAUD DETECTION

We have utilized the Large Language Models and other architectures to identify potential exaggerations in Letters of Recommendation (LORs) by analyzing language patterns that indicate inflated claims. The model segregates these LORs, flagging those that appear suspicious. Additionally, we identified instances of reciprocal recommendations, where two individuals endorse each other, and flagged these cases in binary format. Finally, we compared the LORs to resumes to detect inconsistencies, ensuring that the content in both is aligned and helping flag possible fraudulent submissions.

Cycles in Letter of Recommendations

We have found all the IDs that are involved in reciprocal cycles and flagged them with binary numbers, which is 1 when the corresponding LOR is engaged in a cycle and 0 when it is not. This feature will be later used by assigning weights to it into a statistical foundational model.

Detection of Exaggerated Letter of Recommendation



To detect the level of exaggeration in an LOR we have used DistilBERT in which every line of the text file is passed to calculate the semantic embedding. These vectors capture the meaning of the sentences and are now ready for further analysis. Vectors are compared for similarity using a distance metric like cosine

similarity or Euclidean distance and KNN clustering is applied to these vectors. After clustering, similar vectors (e.g., recommendation letters with similar language) will be grouped together, and outliers or unusual patterns (e.g., exaggerated claims or fabricated credentials) may appear in separate clusters. Based on the distances, ratios are assigned and quantile ranges helped to divide the data equally, thus assigning a score for the same.

Correlation of Letter of Recommendations with Resume

We have parsed the CV and the letter of recommendation and created embeddings for both of them. These embeddings can capture both the context and the **semantics** of both files. We compute the cosine similarity and normalize them to get the overall similarity between the letter of recommendation and the resume. If the similarity score is closer to 1 then we may assertively say that the LOR of the candidate aligns with his job title in the CV but if the score is less than a threshold value this implies that the person has been recommended for some other position or given a generic LOR deviating its chances towards a fraudulent profile. Finally using Information gains through Decision trees we found the weightage of each of the 3 features which then helped us calculate the fraudulent score.

DATA ANALYSIS (CONNECTIVITY)

In this section, we have carried out three operations in order to understand how strong the connections of a particular person are.

Letter of Recommendations:

One of the basic and important factors is the number of genuine letters of recommendation received by the person along with the number of genuine LORs written/given by the person. This helps us find how many connections does it has.

Work Experience and Education

We have quantified the strong and meaningful connectedness between two people based on the overlap in their work experience in the same company and same department or the overlap of the education taken by both the IDs.

We employed Qwen2 LLM in order to parse and get important information out of the resume since normal parsing was being restricted due to a change in the format of writing the core part of the CV.

Bert Name Entity Recognition

It was observed that the connection of a person can be quantified as higher or lower based on overlaps in education and common companies they have worked in. We used the Bert Name

Entity Recognition Model to get a list of companies a person with a certain ID has worked in which are mentioned in their resume. Similarly we have also extracted the education section so that we can find overlap entries which may further contribute towards potential connectivity score

Influence Score

The third and one of the most important features that contribute to the connectivity score is the Influence Score.

1. Domain Influence Ratio:

- How it was formed:

$$\text{Domain Influence Ratio} = \text{Domain-specific In-degree} / (\text{Cross-domain In-degree} + 1)$$

This formula computes the ratio of recommendations the individual receives from their own domain compared to those from other domains. Adding 1 to the denominator avoids division by zero when an individual has no cross-domain recommendations.

- Intuition:

This metric reflects how concentrated or specialized an individual's influence is within their own domain. A high Domain Influence Ratio means the individual is primarily recognized by peers within their own field, suggesting domain-specific expertise. A low ratio indicates that the individual's influence extends beyond their own field, with more recommendations coming from other domains, suggesting interdisciplinary influence.

2. Degree Centrality:

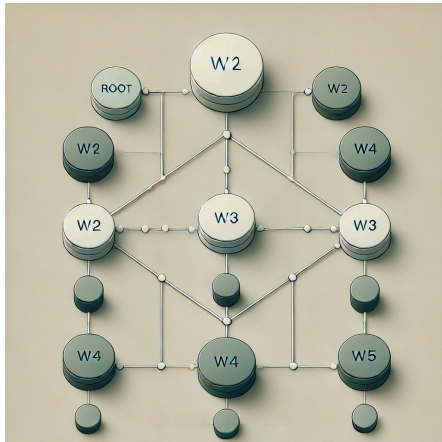
- How it was formed:

$$\text{Degree Centrality} = \text{In-degree} + \text{Out-degree}$$

This is the sum of the In-degree (how many people recommend the individual) and Out-degree (how many people the individual has recommended).

- Intuition:

Degree Centrality measures the total number of connections an individual has in the network. It reflects overall connectivity, both in terms of how many others recommend them and how active they are in recommending others. A higher Degree Centrality suggests the individual is well-connected and actively participates in their network, potentially indicating influence.



3. Domain-specific In-degree:

- How it was formed:

Domain-specific In-degree = Number of recommendations received from people in the same domain

This counts the number of recommendations the individual receives from others within their own domain (e.g., finance professionals recommending other finance professionals).

- Intuition:

This metric focuses specifically on influence within a specific domain or field. A high value indicates that the individual is highly regarded by their peers, reflecting domain-specific expertise and leadership.

4. Cross-domain In-degree:

- How it was formed:

Cross-domain In-degree = Number of recommendations received from people in different domains

This counts the number of recommendations the individual receives from people outside their domain.

- Intuition:

Cross-domain In-degree* reflects the individual's *interdisciplinary influence*—how much they are recognized by people outside their own field. A high Cross-domain In-degree suggests that the individual has a broad influence that spans multiple domains, indicating versatility and cross-functional leadership.

5. Final Influence Score:

- How it was formed:

The Final Influence Score is a weighted sum of the normalized values of the four features:

Final Influence Score = Domain Influence Ratio * W_1 + Degree Centrality * W_2 + Domain-specific In-degree * W_3 + Cross-domain In-degree * W_4

where (W_1), (W_2), (W_3), and (W_4) are the feature importances derived from the Decision Tree model.

- Intuition:

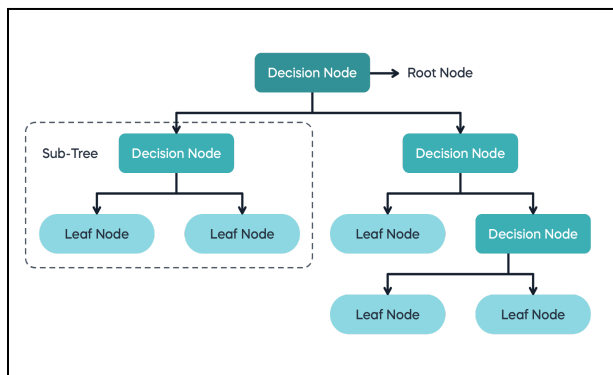
The Final Influence Score integrates all the relevant features, weighting each according to its importance in determining overall influence. The features that contributed more to splitting the data (according to the Decision Tree) are weighted more heavily in calculating this score. The score is designed to reflect the individual's overall influence, taking into account both their domain-specific and cross-domain connectivity, as well as their total network involvement.

Summary of the intuition:

- Domain Influence Ratio tells us how focused someone's influence is within their own domain versus across domains.
- Degree Centrality indicates how connected they are in the network, both in terms of being recommended and recommending others.
- Domain-specific In-degree shows their reputation within their own field.
- Cross-domain In-degree shows their recognition and influence outside of their field.

The final Influence Score is a composite measure that blends all these factors, with more important factors (based on the Decision Tree) having more weight.

Why decision trees?



Decision trees can handle complex, non-linear relationships between features, which is particularly useful when adjusting weights to find optimal outcomes. This is beneficial when weights are not directly proportional to the outcome, as the tree can learn specific splits that best represent interactions between different parameters. Decision trees offer an inherent way to assess the importance of each feature (or weight in your case). By looking at how often a

feature is used in decision splits and the reduction in impurity it provides, you can rank the significance of each weight. Decision trees are robust to missing data and outliers, which means if certain weights are unavailable or have extreme values, the tree can still make optimal decisions by focusing on other, more stable variables.

PROFILE SCORE:

Another important scoring metric to determine if a profile is good or not is the Profile Score. We have quantified the work experience of the profile as an important factor in determining what all a candidate brings to the table.

Using LLM Model Qwen2 we have found the number of years of work experience of all the people from ID_0 to ID_999 and normalized the entire dataset to get a score. The higher the score higher the work experience of the corresponding ID while lower means lower the number of years of work experience.

Apart from this Profile will depend a lot on the skills of a particular person. Now since the skills required depend from one Job Description to another we have implemented the keyword searching functionality on the HR: Dashboard where the HR can filter those resumes that have a particular set of keywords.

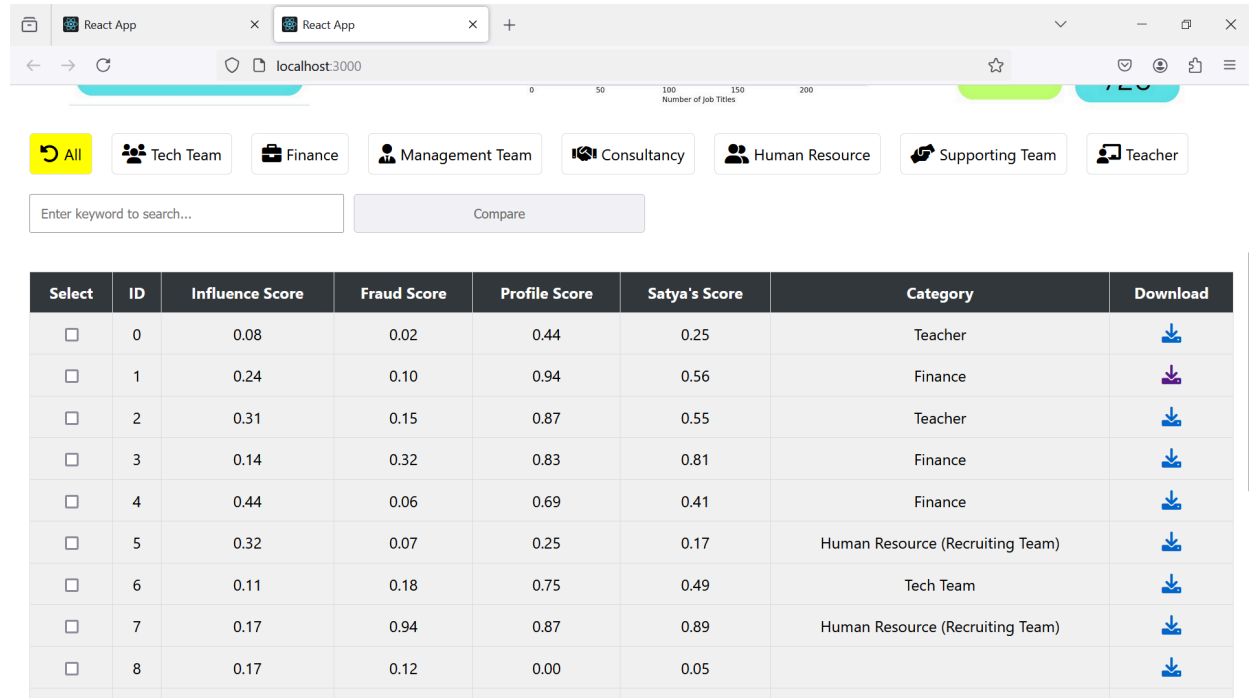
HR SUPPORT DASHBOARD

The following website puts together all the statistics regarding a CV and assists HR in terms of finding the most suitable candidate.

The salient features of this Dashboard are:

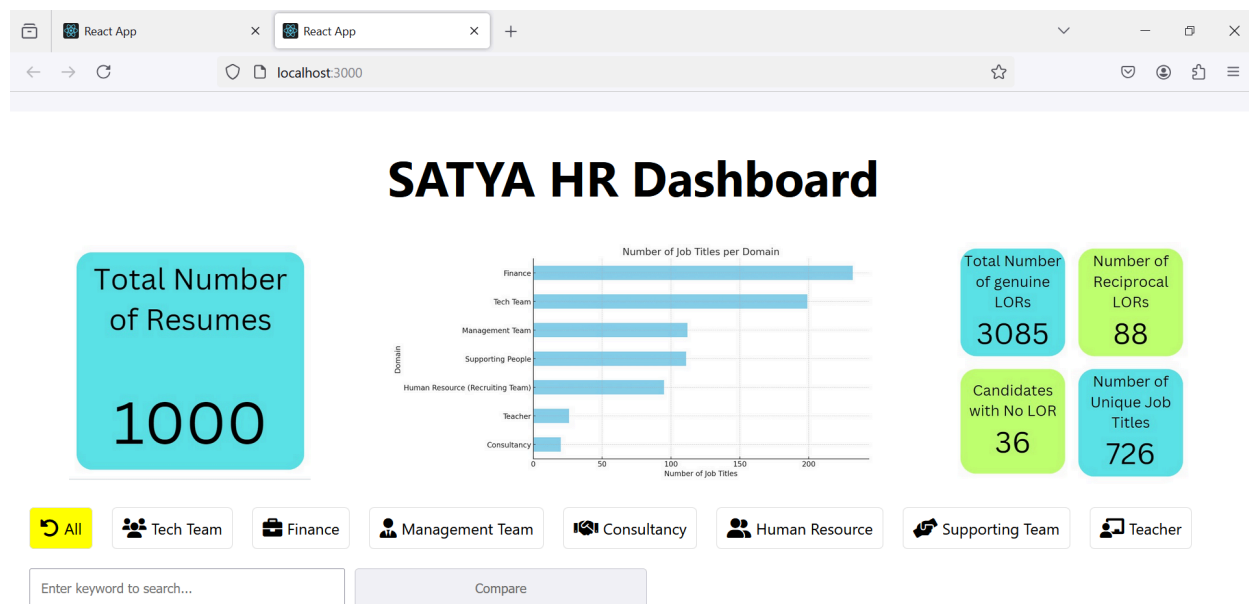
1. Keyword Searching
2. Comparing profiles of any number of IDs of your choice
3. Fraudulent Score
4. Connectivity/Influence Score
5. Z-score/Profile Score
6. Segregation based on Domains
7. Zipped CV and LORs received for a given ID

Snippets of the Live Dashboard



The screenshot shows a web application interface with a table of job titles. The table has columns for Select, ID, Influence Score, Fraud Score, Profile Score, Satya's Score, Category, and Download. The data is as follows:

Select	ID	Influence Score	Fraud Score	Profile Score	Satya's Score	Category	Download
<input type="checkbox"/>	0	0.08	0.02	0.44	0.25	Teacher	
<input type="checkbox"/>	1	0.24	0.10	0.94	0.56	Finance	
<input type="checkbox"/>	2	0.31	0.15	0.87	0.55	Teacher	
<input type="checkbox"/>	3	0.14	0.32	0.83	0.81	Finance	
<input type="checkbox"/>	4	0.44	0.06	0.69	0.41	Finance	
<input type="checkbox"/>	5	0.32	0.07	0.25	0.17	Human Resource (Recruiting Team)	
<input type="checkbox"/>	6	0.11	0.18	0.75	0.49	Tech Team	
<input type="checkbox"/>	7	0.17	0.94	0.87	0.89	Human Resource (Recruiting Team)	
<input type="checkbox"/>	8	0.17	0.12	0.00	0.05		



CONCLUSION

The Environment developed to enhance HR decision-making, leverages NLP and advanced Large Language Models to evaluate resumes and recommendation letters for authenticity, relevance, and connectivity. It calculates three primary scores: Fraudulent Score (to detect exaggeration and inconsistencies), Connectivity Score (to assess professional influence and

network strength), and Profile Score (to quantify experience and skills alignment). This comprehensive scoring system, combined with a user-friendly HR Dashboard, empowers recruiters to identify the best candidates efficiently by flagging suspicious profiles, highlighting strengths, and offering data-driven hiring insights.

ANNEXURE

ANNEXURE A (COSINE SIMILARITY PREDICTIONS)

About:

Cosine similarity is a measure of quantifying the similarity between 2 vectors in a vector space. In the case of Natural Language Processing (NLP), vectors represent a document/collection of words, where each word holds a sequence of numbers (word vectorization). Cosine similarity has wide-ranging applications in fields such as computer vision, data mining, etc.

In mathematical terms, cosine similarity is the cosine of the angle between 2 vectors which can be calculated by dividing the dot product of two vectors by the product of their magnitudes.

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

Here, A_i and B_i represent the components of vectors A and B respectively.

Why Cosine Similarity?

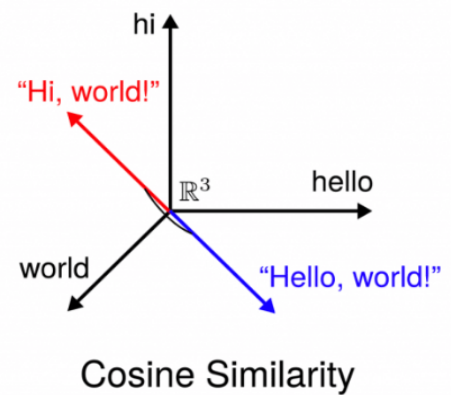
In the field of NLP, finding the text similarity between documents/sentences calls for cosine similarity as a simple and efficient method. The document/sentence is converted to a vectorized representation.

Applying cosine similarity to these two vectors gives us a value between 0 and 1. A value of 0 indicates absolutely NO similarity, and 1 means that the 2 texts are exactly alike.

Approach: We calculated the cosine similarity of the LORs received by an ID and the CV of the ID to quantify whether the LORs align with the CV or not.

if (cosine similarity > threshold) → classify as 1 {CV maps correctly with the LOR}

if (cosine similarity < threshold) → classify as 0 {LOR is either generic or different from the applied Job Title}



ANNEXURE B (CLUSTERING AND PREDICTING STRATEGY)

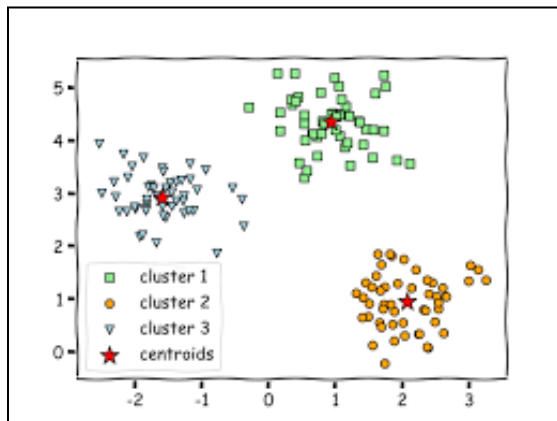
Objective:

To group the sentences which are showing the same type of bias.

How we used it for predicting the labels:

Using the different clustering techniques, we grouped together the sentences that have the same type of bias. We divided the sentences of the LOR into 2 clusters, each having a distinct type of bias.

KNN Clustering



K-Nearest Neighbors (KNN) clustering is a method used to classify sentences into clusters based on their similarity. Instead of creating centroids like in K-means, KNN classifies a point based on its distance to its 'K' nearest neighbors. The similarity is determined by calculating the distance (typically Euclidean) between sentences in a high-dimensional space. Each sentence is assigned to the most common cluster among its nearest neighbors. The value of 'K' determines the number of neighbors used to decide the cluster for a particular point.