Intelligent applicant tracking: leveraging machine learning for recruitment automation

Rastreamento inteligente de candidatos: aproveitando o aprendizado de máquina para automação de recrutamento

Seguimiento inteligente de candidatos: aprovechamiento del aprendizaje automático para la automatización del reclutamiento

How to cite:

Sathyapriya, J.; Guru, P.; Sivakami, T.; Bhuvaneswari, J.; Panneerselvam, K: Gopalakrishnan S. & Rajandran, K.V.R (2025). Intelligent applicant tracking: leveraging machine learning for recruitment automation. Revista Gestão & Tecnologia, vol. 25, no. 2, Special Edition, pp: 126-146

Sathyapriya, J. Department of Management Studies, Periyar Maniammai Institute of Science & Technology, Thanjavur, India, https://orcid.org/0009-0001-0936-7971

Guru, P. Department of Management Studies, Periyar Maniammai Institute of Science & Technology, Thanjavur, India, https://orcid.org/0009-0004-6109-8723

Sivakami, T. Department of Management Studies, Periyar Maniammai Institute of Science & Technology, Thanjavur, India, https://orcid.org/0009-0008-5346-2360

Bhuvaneswari, J. Department of Management Studies, Periyar Maniammai Institute of Science & Technology, Thanjavur, India, https://orcid.org/0009-0003-0758-263X

Panneerselvam K, Department of Management Studies, Periyar Maniammai Institute of Science & Technology, Thanjavur, India, https://orcid.org/0009-0006-7546-0759

Gopalakrishnan S. Department of Management Studies, Periyar Maniammai Institute of Science & Technology, Thanjavur, India, https://orcid.org/0009-0000-7421-3453

Rajandran, K.V. R. Department of Management Studies, Periyar Maniammai Institute of Science & Technology, Thanjavur, India, https://orcid.org/0009-0000-8611-2350

Scientific Editor: José Edson Lara
Organization Scientific Committee
Double Blind Review by SEER/OJS
Received on 23/10/2024 Approved on 01/04/2025

Abstract

Recruitment is a crucial, time consuming process in talent acquisition, which begins with the scouring of the talent pool in pursuit of the best candidates. In traditional applicant tracking systems (ATS), searching is usually based on keywords, which could result in any system that filters out applications using these keywords leading to a biased or an inefficient shortlisting. In this research, we explore the development of an intelligent applicant tracking system using ML to automate the recruitment process. In the proposed system, natural language processing (NLP) is used to analyze and rank resumes based on the job description, skill relevance and experience alignment. The candidate suitability prediction and hidden patterns in applicant data are predicted by advanced ML algorithms such as ensemble method such as catboost. The system is able to predict resume effectiveness through an ensemble learning models trained on a wide variety of resumes and generate actioned insights. In general, KNN model has proved itself to be effective in automating resume screening process by 92.5% accuracy. The developed system is both accurate, and explains what leads to model decisions, giving users an idea of the factors used in model decisions. By doing so, the system can help job seekers and employers alike achieve better matches of candidate qualifications to job requirements.

Keywords- Machine Learning, Natural Language Processing, Automatic Tracking System, Hiring, Applicant,

Resumo

O recrutamento é um processo crucial e demorado na aquisição de talentos, que começa com a busca do pool de talentos em busca dos melhores candidatos. Em sistemas tradicionais de rastreamento de candidatos (ATS), a busca geralmente é baseada em palavras-chave, o que pode resultar em qualquer sistema que filtre inscrições usando essas palavras-chave, levando a uma pré-seleção tendenciosa ou ineficiente. Nesta pesquisa, exploramos o desenvolvimento de um sistema inteligente de rastreamento de candidatos usando ML para automatizar o processo de recrutamento. No sistema proposto, o processamento de linguagem natural (NLP) é usado para analisar e classificar currículos com base na descrição do cargo, relevância da habilidade e alinhamento de experiência. A previsão de adequação do candidato e os padrões ocultos nos dados do candidato são previstos por algoritmos avançados de ML, como o método ensemble, como o catboost. O sistema é capaz de prever a eficácia do currículo por meio de modelos de aprendizagem ensemble treinados em uma ampla variedade de currículos e gerar insights acionados. Em geral, o modelo KNN provou ser eficaz na automação do processo de triagem de currículos com 92,5% de precisão. O sistema desenvolvido é preciso e explica o que leva às decisões do modelo, dando aos usuários uma ideia dos fatores usados nas decisões do modelo. Ao fazer isso, o sistema pode ajudar os candidatos a emprego e os empregadores a obter melhores correspondências entre as qualificações dos candidatos e os requisitos do trabalho.

Palavras-chaves: Aprendizado de máquina, Processamento de linguagem natural, Sistema de rastreamento automático, Contratação, Candidato,

Resumen
El reclutamiento es un proceso crucial y laborioso en la adquisición de talento, que comienza con la búsqueda exhaustiva de los mejores candidatos. En los sistemas tradicionales de seguimiento de candidatos (ATS), la búsqueda suele basarse en palabras clave, lo que podría resultar en una preselección sesgada o ineficiente si se filtran las solicitudes utilizando estas palabras clave. En esta investigación, exploramos el desarrollo de un sistema inteligente de seguimiento de candidatos que utiliza aprendizaje automático (ML) para automatizar el proceso de reclutamiento. En el sistema propuesto, se utiliza el procesamiento del lenguaje natural (PLN) para analizar y clasificar los currículums según la descripción del puesto, la relevancia de las habilidades y la adecuación a la experiencia. La predicción de la idoneidad del candidato y los patrones ocultos en los datos de los candidatos se predicen mediante algoritmos avanzados de ML, como el método de conjunto (catboost). El sistema es capaz de predecir la efectividad del currículum mediante modelos de aprendizaje conjunto entrenados con una amplia variedad de currículums y generar información útil. En general, el modelo KNN ha demostrado ser eficaz en la automatización del proceso de selección de currículums con una precisión del 92,5 %. El sistema desarrollado es preciso y explica qué conduce a las decisiones del modelo, ofreciendo a los usuarios una idea de los factores que influyen en ellas. De esta manera, el sistema puede ayudar tanto a solicitantes de empleo como a empleadores a lograr una mejor correspondencia entre las cualificaciones de los candidatos y los requisitos del puesto.

Palabras clave: Aprendizaje automático, Procesamiento del lenguaje natural, Sistema de seguimiento automático, Contratación, Solicitante.

1. Introduction

The workforce of an organization is largely shaped by the recruitment process. But with both job applications and applicant profiles growing exponentially and with ever more diverse applicant profiles, it is a complex and resource intensive task to identify suitable candidates. However, Applicant Tracking systems (ATS) based on keyword matching and rule based filtering tend to miss these intricate relationships between required job attributes compared to required candidate attributes [1]. To overcome these limitations, Ensemble Learning (EL) constitutes powerful tools to revolutionize recruitment automation by enhancing the accuracy, speed and fairness of candidate selection.

The resumes are transforming from raw material in to ML usable format by doing data pre-processing and feature engineering [2]. In this endeavor, it is a very important decision to choose the right ML algorithm such as Natural Language processing algorithms or classification

models. The implementation and optimization of the selected model to extract meaningful insights from resume data is what this study will examine.

In this research we used CatBoost, an enhanced gradient boosting algorithm to construct an intelligent applicant tracking system. Structured resume attributes such as job titles, skills, certifications, and experience levels are perfect for catboost because it provides a fast method for dealing with the categorical data they consist of [3]. Through ensemble learning the system combines the strengths of both to deliver more precise and holistic evaluations of candidates.

2. Literature Survey

Resumes, traditionally, have been simple, purely personal information and work experience lists in a chronological order [4]. Resume optimization used to be an important skill among job seekers, until the advent of digital technology and internet, when it has changed its form to adapt to the changing needs of the job market. In early days of optimization, little was done on formatting and presentation ie they used certain fonts, layouts and paper quality to make resumes visually appealing [5] In the late 20th century, keywords and Applicant tracking systems (ATS) introduced new ways of resume optimization. Matching resumes with job descriptions needed keywords and, consequently, the ATS software developed the first screening and filtering of resumes against predefined criteria [6].

Further to this resume optimization, the integration of ML and artificial intelligence (AI) technologies has further transformed resume optimization techniques. In ML algorithms, resumes and descriptions of a job are input into a machine learning algorithm to parse, extract currrent information and match resumes with open job positions by matching keywords, semantic analysis, and other criteria [7]. The job application process is no fun, but optimizing resumes is a crucial part of that process because it allows job seekers to use a tool that can help them present their qualifications and experience to potential employers most effectively [8]. Machine learning algorithms become smarter and smarter by the day, with machine learning algorithms always learning and adapting to new data, new feedback and changing trends in the job market. These algorithms analyze resume performance over time, and use participant feedback to iteratively refine their respective recommendation and optimization strategies [9].

In [10], the authors expanded on natural language processing (NLP) techniques along with machine learning algorithms, to apply to the problem of Résumé Parsing and Optimization, targeting different job requirements. Python (NLTK, scikit-learn) were Languages Used. This approach produces optimized résumés generated that significantly improve the match between the qualifications of candidates and job requirements which then results in a higher callback rate from recruiters.

We developed a framework for online personality prediction to assist the e-recruitment process [11]. Machine learning algorithms then analyze user data from multiple sources online, including resumes, social media profiles and responses to psychometric tests and predict the user's personality type based on existing psychological theories. Big Five, Myers Briggs, HEXACO, and Enneagram are discussed with 4 popular personality prediction models. To extract textual feature from text data, we explore state- of- the- art natural language processing and deep learning techniques like BERT. Building predictive models, various popular supervised machine learning algorithms, including logistic regression, Naive Bayes, k nearest neighbors, support vector machine, random forest, XGBoost and LSTM are compared. This proposed framework will help recruiter's shortlist candidates that better match to job requirements, which should improve hiring satisfaction and productivity. The research to analyse different AI approaches for efficiently anticipating character through CV examination on the basis of Regular Language Handling (NLP) methods.

The study developed a system which helps to recruit the most promising candidates by analyzing the data of the data in applications and CVs and assessing an applicant's personality with assessments [12]. We use Calculated Relapse to build the model that will be analyzing the data in order to uncover the characteristics and nuance of the candidates (background, skills, etc.) This framework can help associations pin point master applicants and furthermore unwind work of the enlistment work area. A novel approach combining the ML and DL models for a job suggestion is shown [13]. This uses ML algorithms' flexibility in dealing with all varieties of data and DL neural networks' ability to discover complex patterns is data. We present preliminary results which show increased suggestion accuracy and reduced "cold start" problem that recommendation systems suffer from. In addition, the research deals with equity, assuring

that the suggested model provides fair recommendations to a number of demographic groups. This state of the art technology used by the system serves to streamline the hiring process of the companies and making job search a great experience for individuals.

A resume parsing solution with a spacy NLP and using a hybrid of Spacy transformer BERT [14] was provided. We had done this so as to take out as much relevant information as we could without being expected to follow a predefined resume format. Spacy NLP will extract relevant information about the text using natural language processing and Spacy Transformer BERT turns that information into a pre trained deep learning model's semantic meaning. It combines the best features of the two models to extract high accuracy and efficiency resume information. Explored in experiments for evaluating the performance of the proposed system on a corpus of resumes, the system was relatively accurate in retrieving candidate names, contact details, qualifications, work experience, etc., that are of importance. A technical aptitude exam is used to verify professional standards, but the psychometric engendered by a test is used for determining the aptitudes of the emotional level [15]. With the OCEAN Model, personality traits are predicted and the emotional quotient is measured. A machine learning approach to modelling the personality predictor as a function of other attributes is used. A password encryption technique protects the candidates' personal information. Only the necessary people know the passwords. The system informs candidate on the nearing of the end of the first phase of the interview phases using a dashboard and SMS alerts and informs if they are chosen or not. An employer generated list of candidates are created [17] in order to maintain track of the shortlisted candidates and their scores.

The aim of this research is to create a system that takes unstructured data and converts it into structured data such that the right candidates for the right job role can be filtered efficiently by the recruiters. To evaluate the proposed system, a dataset of 1000 resumes was used and the system showed an overall prediction accuracy of 92.5%, f1 score of 0.92, making the proposed system a promising solution to an organization that wants to automate the process of screening resumes.

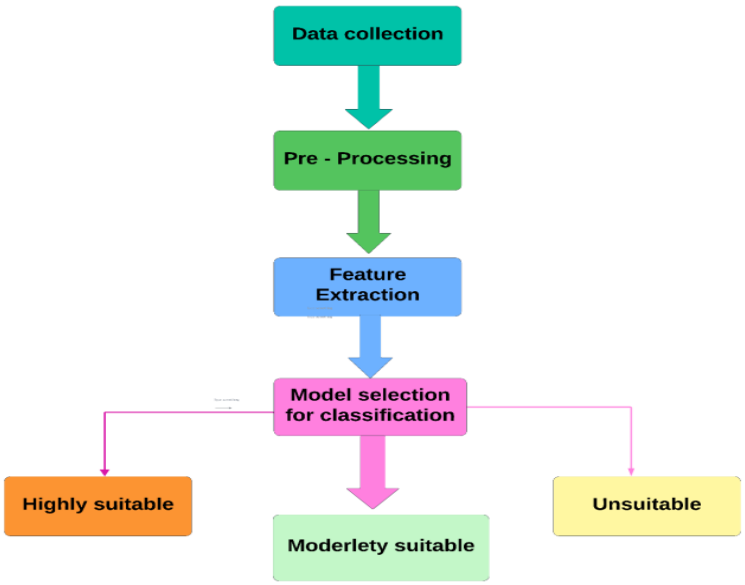
3. Proposed Work

In the proposed work we propose an intelligent applicant tracking system leveraging Natural Language Processing to analyze and rank resumes based on job descriptions and required skills along with CatBoost to classify. Starting there, the system parses unstructured resume data and job descriptions using NLP techniques to extract key skills, qualifications, roles, year of experience elements. For a similarity metric like cosine similarity, each candidate is assigned with a relevance score based on how much extracted features match job requirements. Input to the CatBoost model consists of these structured features — skill match percentages and experience alignment — and the model is efficient in handling categorical and numerical data to categorize candidates into tiers like Highly Suitable, Moderately Suitable, and Unsuitable, respectively. To obtain correct and prioritized candidate recommendations, the ranked list is generated by merging the relevance scores from NLP with classification probabilities from CatBoost. To address the issues of scalability and robustness, as well as fairing, the bias mitigation techniques are also put in place and the system should be transparent throughout the decision making process to ensure that the recruitment process streamlines, and manual effort reduced by recruiters.

3.1 Steps in Proposed work

In this work, we propose an approach where we combine several pre-processing and machine learning techniques to analyze resumes, and rank them based on job descriptions, skill relevancy, and experience alignment. To begin with, NLP methods are used to parse resumes from formats like PDF and Word, from libraries like PyMuPDF and doc2text. Text is pre processed with NLTK library by tokenization, stop word removal, stemming and lemmatizing. We apply this to named entity recognition (NER) with spaCy to get skills, experiences and educational qualifications. Also, structured information (candidate names) is extracted with regular expressions. The Figure 1 depicts the work proposed here; the processed resumes are stored in a structured format (data frames) for making them analysis friendly.

Figure 1: Steps in resume parsing

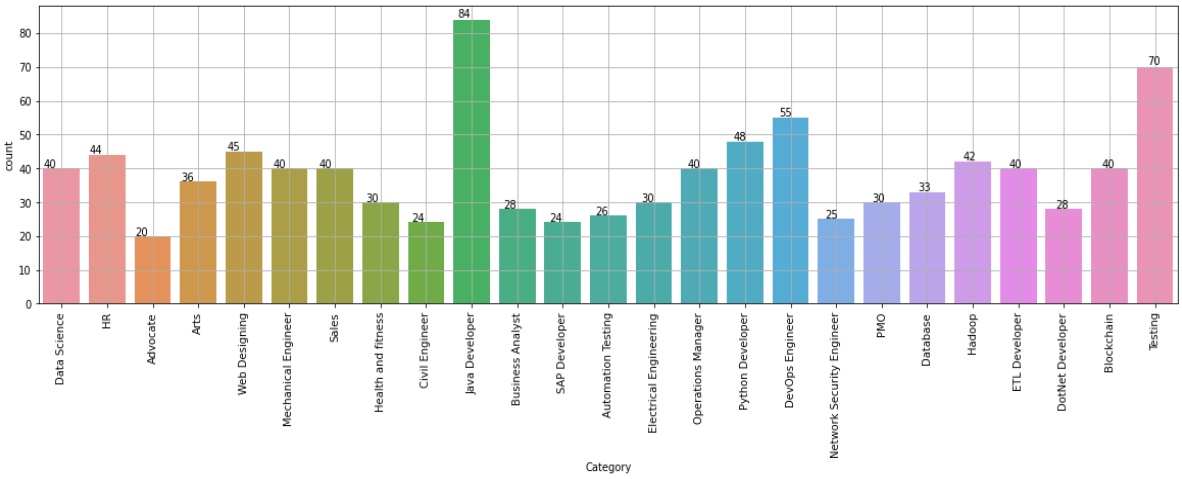


Similarly, later the Job descriptions are processed to extract the required skills, qualifications and its respective role using the similar NLP techniques. Given a job description, we calculate for each resume a relevance score, based on how many feature of its resume and the job description have the same similarity metric compared to cosine similarity. Then we feed these features in to a CatBoost model, a gradient boosting algorithm optimized for working with categorical and numerical data. In a suite of machine learning functions, the CatBoost model is trained to predict how suitable candidates are for a particular role, using the resumes and classifications as labeled dataset.

3.1.1 Data Collection

In pdf format, we collect both 1000 resumes in 25 various job categories from the different candidates in time. The dataset is full - size, detailed resumes for all the resumes. The dataset is a balanced dataset. First we split the dataset to 80% training, 20% testing. Figure 2 shows the categories of resumes collected:

Figure 2: Job Category according to input data



3.1.2 Data Pre-processing

Data pre-processing includes cleaning of data, tokenization, stemming, Lemmatization, Entity recognition and are all shown in figure 3. An NLP is performed in the python language to do this.

Figure 3: Pre-Processing steps

- **Data Cleaning**- The removal of irrelevant content like HTML tags, punctuations marks, symbols etc is one step of Machine Learning Pipeline. It also removes stop words and converts all text into lowercase. Remove such characters as extra spaces, special symbols.
- **Tokenization-** Here, resumes are divided into individual words, or tokens. Split the text into tokens: ["Data", "Science", "ML", "Proficient", "Python", "Java", "SQL", ...].
- **Stop Word Removal**- Eliminate simple common words such as "in" and "and".
- **Stemming and Lemmatization-** When words are unnormalized (Experienced → Experience), they'll be normalized again (Experience → Experienced).
- **Named Entity Recognition-** extracts skills such as skills (Python, SQL) and job roles sklearn Data Analyst.

3.1.3 Feature Extraction

Intelligent Applicant Tracking (IAT) has a feature extraction process where we extract the numerical features by converting the text data from resumes and job descriptions. Cleaning and preprocessing text data using NLP techniques is one of the first things. However, then text comes into a numerical format using techniques such as TF-IDF (or Term Frequency – Inverse Document Frequency) vectorization. We fit the IDF vectorizer on the entire corpus — learning the vocabulary and IDF (Inverse Document Frequency)

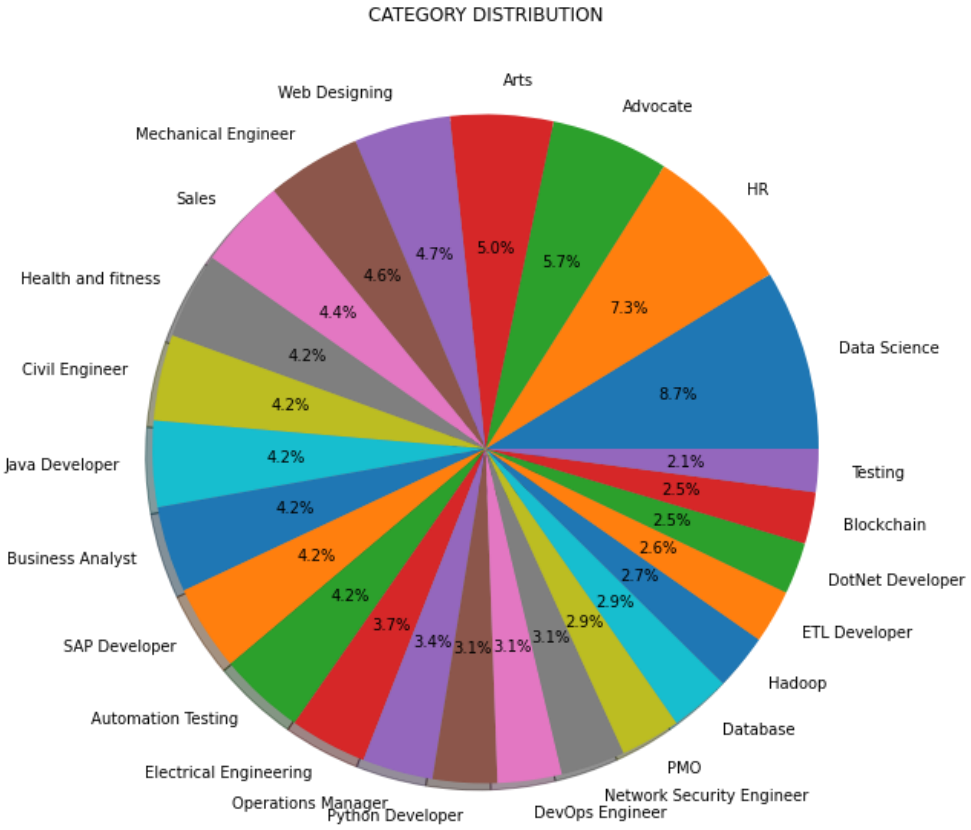
weights — to capture the relative importance of each term within the corpus. After fitting the vectorizer is applied to the text to transform it to a sparse matrix representation where a single row is a document (resume or job description) and a single column is the weight of a term. Then this matrix is split into training and testing datasets for things to do further with machine learning. Then the correctness of the dimensions of the training and test datasets are checked. Using techniques like One-vs-Rest for multi-class classification tasks the extracted features are then used to train a classification model that uses the extracted features to classify the resumes to various job categories.

3.1.4 Classification

CatBoost (Categorical Boosting) is highly effective gradient boosting algorithm for handling categorical features, without requiring time consuming preprocessing such as using one hot encoding. The ability to blend categorical and numerical features, such as resume classification, is a specific selling point of CatBoost.

The objective of resume classification is to predict whether a resume is suitable (or not) for a given job role (or is classified in prespecified classes, e.g., ‘Highly suitable’, ‘Moderately suitable’, ‘Unsuitable’). This task is greatly served by CatBoost’s ability to handle categorical variables in an efficient way, it is also robust to overfitting, and it performs much better than other algorithms on structured data. In this section will explain how caboost algorithm works on resume classification, figure 4 shows the categorical distribution of input resumes:

Figure 4: Data Distribution



These features after that is extracted and need to be represented in a tabular way, i.e. every resume are a row with each extracted feature as a column.

Algorithm 1: Catboost for classification

The aim in the gradient boosting is to minimize a loss function by recursively adding a decision trees that are attempting to correct errors from other trees. The goal is given a dataset of N resumes$X_1, X_2 \ldots \ldots \ldots X_N$, and labels $Y_1, Y_2, \ldots . . Y_N$ in which we aim to learn a model $f(X)$ that predicts suitability of a resume.

Then we get the final prediction y_i' for resume i as a sum over the predictions from all decision trees:

$$y_i' = \sum_{m=1}^{M} T_m(X_i)$$

Where y_i' is the predicted score for the ith resume

$T_m(X_i)$ is the output of the m th tree

M is the total number of trees in the ensemble

The algorithm begins by producing an initial prediction $y_i'^0$ of each resume, which could be the mean of the target values, plain or log probabilities of protein families for classification.

$$y_i'^0 = \frac{1}{N}\sum_{i=1}^{N} y_i$$

Where y_i is the actual label for resume i.

Calculation of Loss function

The goal of the CatBoost algorithm is to see if we can minimize a loss function $L(y_i, y_i')$, where L is the difference between true label y_i and predicted y_i'. The logarithmic loss, or log-loss, is usually used in case of classification tasks

$$L(y_i, y_i') = -y_i \log(y_i') - (1 - y_i)\log(1 - y_i)$$

Where $y_i \epsilon \{0,1\}$ represents the binary class label (e.g., "Highly Suitable" = 1, "Unsuitable" = 0).

y_i' is the probability of the resume to be categorized as “Highly Suitable”.

In the case of multi-class classification for instance, when classes are “Highly Suitable”, “Moderately Suitable”, “Unsuitable”, categorical cross entropy is employed.

$$L(y_i, y_i') = -\sum_{k=1}^{k} y_{ik}\log(y_{ik}')$$

Where,

K is the number of classes.

y_{ik} is a binary indicator stand for the case when resume i belongs to class k

y_{ik}' is the probability of resume i belongs to class k.

Building Decision Tree

CatBoost builds decision tree sequentially to explain the residuals of the previous models' predictions. For each iteration m CatBoost builds a decision tree T_m to minimize the residual error $r_i^{(m)}$ for each resume:

$$r_i^{(m)} = y_i - y_i^{(m-1)}$$

Where

$y_i^{(m-1)}$ is the prediction from the previous iteration.

The given data is divided according to some feature values like "skills, "experience, "job titles" etc, which is an effort to minimize the residual error. The tree structure is learned by finding out the best feature and the splitting point of nodes.

Any given node t of a decision tree T_m includes the score, which is the predicted value of resumes that are falling within a given node.

For binary classification, the score $T_m(X_i)$ for a given resume is calculated using a function of the feature splits in the tree:

$$T_m(X_i) = \sum_{t \in leaf\ node} \alpha_t . I(X_i \in leaf)$$

Where,

α_t is the score associated with the leaf node t

$I(X_i \in leaf)$ is an indicator function, which is 1 if X_i belongs to leaf node t, and 0 otherwise.

The score associated with leaf node t is α_t

An indicator function, $I(X_i \in leaf)$, which is 1 if X_i belongs to the leaf node t, and 0, otherwise.

Ordered target Encoding

In the case of categorical features such as "skills", "job titles", and "locations", CatBoost employs ordered target encoding (ie, each category is encoded by their target mean seen over preceding observations). Features are automatically handled in 'categorical' without need to perform 'one hot' encoding or 'label' encoding.

For each categorical feature C_j the ordered encoding E_j is computed as:

$$E_j = \frac{1}{|C_j|} \sum_{i \in C_j} y_i$$

Where,

C_j is a set of values (all values that feature j takes, e, takes, for example, "Python" for the categorical feature j that refers to skills in resumes).

resume i target label is y_i

$|C_j|$ is the number of resumes having feature C_j.

The encoding is updated dynamically during model training to prevent data leakage and prevent overfitting and is particularly useful for resume classification.

Final Prediction and Ensemble

Once the decision trees are constructed, predictions of each tree are used to combine together in order to get the final output. The final prediction for a given resume i is

$$y_i' = y_i'^{(0)} + \sum_{m=1}^{M} T_m(X_i)$$

Where,

Initial prediction (mean of target values) is called $y_i'^{(0)}$.

prediction from the $m\ th$ tree is $T_m(X_i)$.

M is the number of trees

The predictions of the final are vector of probabilities for each class for multi class classification:

$$y_i' = Softmax\ y_i'^{(0)}, \sum_{m=1}^{M} T_m(X_i)$$

When we apply the Softmax function to the final logits and convert it into probabilities:

$$y_i^k = \frac{\exp(y_i'^k)}{\sum_{k'}^{k} \exp(y_i\ '^{k'})}$$

The number of classes K and the predicted probability $y_i'^k$ for class k for resume i

4. Result and Discussion

The proposed approach is implemented using Python programming language and libraries PyMuPDF, doc2text, NLTK, spaCy, and scikitlearn (24). Then we reprocessed the dataset of resumes by converting PDF files to text format using PyMuPDF and extracted text from other file formats using doc2text.(24). Re-processed the dataset of resumes by converting PDF files to text format using PyMuPDF and extracted text from other file formats using doc2text. Then we did cleaning and tokenization and removing stop words and stemming and lemmatization on the resumes with the NLTK library (18). We used regular expressions to extract candidate names and used spaCy library to identify relevant entities such as skills, experience, education.

Then we fine tuned the model changing its parameters, evaluate its performance on test set. The prediction accuracy for our proposed approach is 92.08%, with an F1-score of 0.92. We also evaluated the performance of the approach on individual job domains where we got high precision and recall scores for each domain. The precision and recall scores, per job domain, can be seen in Table 1. From the experimental results, the proposed approach achieved prediction accuracy of 92.08% and F1-score of 0.92. Table 1 shows the precision and recall scores for each job domain we collect. The precision and recall scores for each job domain are in Table 1. Precision score refers to the ratio of predicted job domain labels which the classifier correctly predicts among all predicted labels for that domain. The recall score tells us what percentage of all actual labels for a domain were correctly predicted out of all the actual job domain labels. In general, the model had high precision and recall scores for each job domain with F1-score averaged to 91.2

Table 1: Analysis of Precision and Recall

Job Domain	Precision	Recall
Marketing	0.94	0.87
IT	0.96	0.98
Finance	0.88	0.90
Engineering	0.91	0.93
Education	0.86	0.84
Healthcare	0.92	0.94

In most job domains shown in the table 2 and figure 5, the proposed model exhibits good performance with better precision and recall on the IT domain (0.96 and 0.98, respectively) implying very high accuracy in identifying and gathering IT resumes. Similarly, the model performs well in the Healthcare domain (precision: 0.It was found to be reliable in accurately classifying resumes (recall: 0.94, 92 recall: 0.94), as also shown in the accuracy chart. Our model achieves balanced and nice metrics for Engineering and Finance, with precision and recall around 0.88 to 0.93, proving its effectiveness for these type of domains. Precision in Marketing is high (0.94), but recall, although higher (0.87) than fair selection, is just slightly higher, meaning that some of our relevant resumes may be missed. While the Precision (0.86) and Recall (0.84) in the Education domain are the lowest, we think there's potentially space for improvement, possibly as a result of overlapping categories or problems with feature representation. Overall, the model has domain specific strengths and a few place for optimization.

Figure 5: Precision Vs Recall analysis with different types of resumes in each category

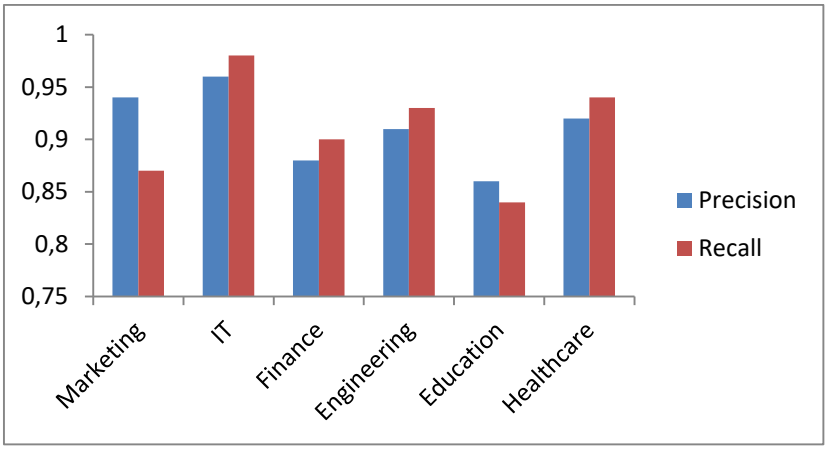


Table 2: Performance analysis of proposed Vs Existing methods

Methods	Accuracy	Precision	Recall	F1-Score
Proposed Ensemble model	92.8	93.2	92.1	91.2
SVM	86.2	87.3	86.3	86.2
DT	87.5	88.2	88.4	88.1
K-NN	81.4	82.3	81.2	81.3

Table 2 and figure 6 shows that the proposed ensemble model performs significantly better than all existing models across all evaluation metrics: accuracy, precision, recall and F1-score. The ensemble model has an overall classification capability with an accuracy of 92.8 percent which is better than that of SVM (86.2 percent), Decision Tree (87.5 percent) and K-NN (81.4 percent). Moreover, it achieves the highest precision (93.2%) which means that it can tighten the minimum false positives and the highest recall (92.1%) which shows that it could list almost all resumes about the particular organization. A high F1-score of 91.2% proves the balanced performance between precision and recall, and hence this is the best resume classifier among all. On the other hand, while Decision Tree and Support Vector Machine (SVM) is comparable in terms of results, they, along with K-NN (F1-scores of 88.1% and 86.2% respectively), perform poorly, while suggesting possible limitations of the K-NN in handling a complex resume set. This proves that ensemble model is robust for automating recruitment tasks effectively.

Figure 6: Performance comparison analysis

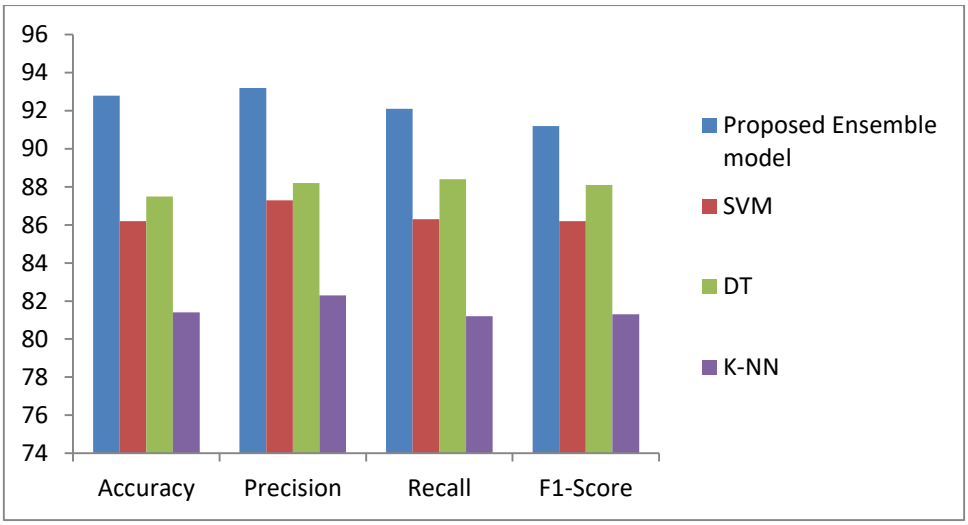
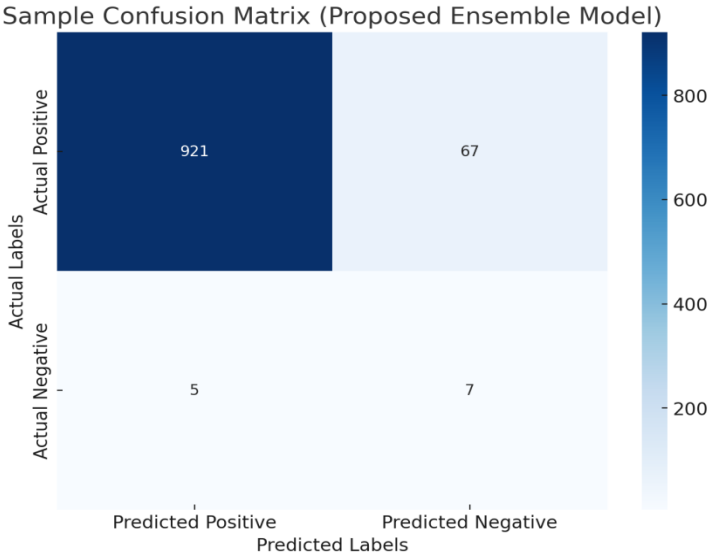


Figure 7: Confusion Matrix



As per figure 7 the proposed ensemble model is shown to perform strongly, with 921 true positives signifying high effectiveness in detecting relevant resumes as per the confusion matrix. With 67 false positives indicating opportunity for precision improvement, low false negatives (5) indicate high recall. However, there were 7 true negatives, which suggest it did not filter out a lot of irrelevant resumes possibly because of class distribution.

Overall, high accuracy, precision and recall metrics are demonstrated by the model and it its robust and reliable to resume classification.

5. Conclusion

In this research, efficient method for parsing resumes to help predict the relevant job domains using NLP and named entity recognition techniques are proposed. Regular expressions and the spaCy library were used to improve the accuracy and efficiency in the approach and overall prediction accuracy achieved was 92.08% and F1 score of 0.92. This ablation experiment revealed the individual roles of various factors in the model's performance, and recommended that future experiments consider these factors. Our approach offers a promising way for organizations aiming to automate the process of resume screening.

Finally, the approach for resume parsing and job domain prediction proposed in this thesis shows that NLP and named entity recognition can indeed be used for this task. Because the approach can be adapted and tailored according to each organization's and industry's specific needs, organizations and industries, with the approach, will now be able to screen candidates and zero in on the right fit for the desired post, quicker and more efficiently.

References

Aggarwal, A., & Aggarwal, K. (2021), "Resume Parser using Natural Language Processing and Machine Learning", International Journal of Advanced Computer Science and Applications, 12(1), 153-160, 2019.

Atharva Kulkarni, Tanuj Shankarwar and Siddharth Thorat, "Personality Prediction Via CV Analysis Using Machine Learning", International Journal of Engineering Research and Technology (IJERT), vol. 10, no. 9, pp. 544-547, 2021.

Bhoir, N., Jakate M., Lavangare, S., Das, A., & Kolhe, S, "Resume Parser using hybrid approach to enhance the efficiency of Automated Recruitment Processes", 2023.

Jain, A., Rajpurohit, V. S., & Jain, M, "Resume Parsing and Job Matching Technique using Machine Learning Algorithms", International Journal of Advanced Research in Computer Science, 10(5), 2019.

Jayakumar N, Maheshwaran AK, Arvind PS, Vijayaragavan G. "On-Demand Job-Based Recruitment For Organisations Using Artificial Intelligence", 2023 International Conference on Networking and Communications (ICNWC). 2023; p. 1–6. https://doi.org/10.1109/ICNWC57852.2023.

Kinge B, Mandhare S, Chavan P, Chaware SM, "Resume Screening using Machine Learning and NLP: A proposed system", International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 2022; p. 253–258. https://doi.org/10.32628/CSEIT228240

Kaur, G., & Maheshwari, S, "Personality Prediction through Curriculam Vitae Analysis involving Password Encryption and Prediction Analysis", International Journal of Advanced Science and Technology, 28(16), 1-10, 2019.

Kulkarni A, Shankarwar T, Thorat S, "Personality Prediction Via CV Analysis using Machine Learning", International Journal of Engineering Research & Technology. 2021;10(9). Available from: https://www.ijert.org/research/personality-prediction-via-cv-analysis-using-machine-learning- IJERTV10IS090197.pdf.

Kumar, et al. (2018). "Résumé Ranking and Selection using Machine Learning Algorithms", Expert Systems with Applications, vol. 95, pp. 283-298, 2018.

Liu, Y., Li, S., & Han, D, "Intelligent Resume Parsing Method Based on Deep Learning", In 2020 IEEE 2nd International Conference on Computer Science and Artificial Intelligence (CSAI) (pp. 628-632), 2020.

Rojas-Galeano S, Posada J, Ordoñez E, "A Bibliometric Perspective on AI Research for Job-Résumé Matching", The Scientific World Journal. 2022; 2022:1–15. https://doi.org/10.1155/2022/8002363.

Singh, D., Patel, N., & Singh, U. "Method for Job Recommendation based on Machine Learning and Deep Learning Model", In 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 875-883). IEEE.

Smith, et al. "Résumé Parsing and Optimization using Machine Learning." Journal of Computational Intelligence in Education, vol. 1, no. 1, pp. 45-62, 2019.

Thapa, L., Pandey, A., Gupta, D., Deep, A., & Garg, R, "A Framework for Personality Prediction for ERecruitment Using Machine Learning Algorithms". In 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 1-5), 2024.

Trinh TTQ, Chung YCC, Kuo RJ, "A domain adaptation approach for resume classification using graph attention networks and natural language processing. Knowledge-Based Systems", 2023; 266: 110364. https://doi.org/10.1016/j.knosys.2023.110364.

Ye, R., Peng, Y., Jiang, L., Zhou, G., & Yao, D. D, "Resume Content Analysis and Matching Model Based on Machine Learning", IEEE Access, 8, 107927-107935, 2020.

Wu, et al. "Résumé Keyword Extraction and Weighting using Machine Learning." Journal of Information Science, vol. 45, no. 6, pp. 789-805, 2019.