Resume Screening using Naive Bayes Algorithm

Shantanu Milkhe
Dept. of Computer Engineering
Sardar Patel Institute of Technology, Mumbai, India

Prince Mishra
Dept. of Computer Engineering
Sardar Patel Institute of Technology, Mumbai, India

Nikhil Naik
Dept. of Computer Engineering
Sardar Patel Institute of Technology, Mumbai, India

Reeta Koshy
Dept. of Computer Engineering
Sardar Patel Institute of Technology, Mumbai, India

ABSTRACT
In the realm of document classification, the choice of algorithm plays a pivotal role in achieving accurate and efficient results. This research paper delves into a comparative analysis of three distinct algorithms: Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machines. It models the probability of a document belonging to a particular class, making it a fundamental choice for text classification. KNN, an instance-based learning approach, operates on the premise of proximity to classify documents by their similarity to labeled instances. Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. This research paper comprehensively evaluates the performance of these algorithms using a diverse and representative dataset comprising various document categories. Standard evaluation metrics, including accuracy, precision, recall, F1-score, and computational time, were employed to assess the efficacy of each algorithm. The study also explores the impact of dataset size and dimensionality on the algorithms' performance and scalability.

Keywords
Naive Bayes, KNN, Support Vector Machine

1. INTRODUCTION
In today's dynamic employment landscape, sifting through a colossal influx of resumes has become a formidable challenge for organizations. Automating resume screening offers a promising solution, where algorithms play a pivotal role. This study embarks on an exploration of three prominent algorithms—Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machine—and their potential contributions to revolutionize resume screening.

Automated resume screening transcends mere efficiency; it fosters the ideals of objectivity and fairness in the hiring process. In a world where diversity and inclusion are paramount, these algorithms offer a path to reduce unconscious biases that may inadvertently affect traditional screening methods.

Naive Bayes, grounded in probabilistic principles, is renowned for its simplicity and computational efficiency. KNN, driven by similarity metrics, leverages the wisdom that similar candidates often share similar qualifications.

The objective of the Support Vector Machine (SVM) algorithm is to optimize the delineation of an n-dimensional space through the creation of an optimal decision boundary, facilitating the classification of data points into distinct categories and enhancing the accuracy of future categorization for new data instances.

This research goes beyond algorithmic comparisons; it seeks to equip organizations with the insights required to select the most fitting algorithm for their unique resume screening needs. The study delves into the subtle nuances of each algorithm, taking into account dataset size, dimensionality, and classification performance.

As the volume of resumes continues to surge, this paper serves as a guiding compass for organizations, navigating the evolving terrain of talent acquisition. It provides a comprehensive analysis of Naive Bayes, KNN, and Support Vector Machine, addressing the quest for efficient and unbiased resume screening.

In the ensuing sections, we will delve into the methodologies, experimental outcomes, and their practical implications, offering organizations a roadmap for a seamless, objective, and equitable resume screening process.

2. Literature Survey
The research paper [1] includes the use of different machine learning algorithms for the purpose of finding out the most suitable candidates for a given job description. The dataset that was used is in excel format. The paper preprocesses the dataset which includes removing the stop words and lemmatization. The paper had done the classification of resumes with classification algorithms like Random Forest, Multinomial Naive Bayes, Logistic Regression and Linear SVM with the accuracy of 38.99%, 44.39%, 62.4% and 78.53% respectively .

The research paper [2] has used NLP, Name Entity Recognition and character Positioning for resume screening and to extract the information from the resume. The system flow involves converting the Resume to text file, preprocessing, extracting candidate skills, Education details and candidates Experience, Matching the extracted skills with a collection of skill sets defined for software engineer position. This methodology was tested on five different resumes and this model was able to extract only 33.59% percent of skills correctly.

The research paper [3], resumes play a critical role, yet 75% never reach human review due to manual screening challenges. AI-driven systems, utilizing text analysis, swiftly evaluate resumes without bias. These systems categorize applicants objectively using specific criteria and keyword matching, empowering candidates to tailor resumes. For instance, in industrial and systems engineering, diverse concentrations exist, like operations management, logistics, and data analytics. Efficient screening methods are crucial to expedite hiring amidst high application volumes.

The research paper [4] focused on creating the model that can be used to predict the suitable job position for the given resume. It classifies the resumes into different categories or classes. The output of vectorization is given as input to train the model. The model has used a wide range of classifier algorithms to classify the resume. Once the candidate uploads the resume the system predicts or suggests the top 5 suitable job profiles for the resume along with confident scores.

The research paper [5] talks about a system that automatically checks job applicants' resumes using language technology and machine learning. It wants to make it easier to handle lots of resumes without needing people to do it manually, and to make sure no biases affect the process. This system takes resumes in Word or PDF formats and uses language technology, like figuring out important skills and experience from the text. It then compares these skills with what a job needs, and gives scores based on how well they match. The system shows these scores in graphs to help choose the best resumes. It's good because it handles many resumes well, saves time by reducing manual work, gives fair assessments based on skills, and makes it easy to decide.

The research paper [6] proposes a system that can classify the resumes to its suitable job position. It uses different ML classification Algorithms like Decision tree, Random Forest , KNN and Support Vector with accuracy of 83.53%, 91.38%, 81.63% and 90.62% respectively. It uses the NLP and ML techniques to classify the resume effectively to make the recruitment process easy and time efficient.

The research paper [7] explores techniques for resume screening and information extraction using Natural Language Processing (NLP) and Machine Learning (ML). It discusses methods like sentence breakdown, word tagging, and identifying important details, 6 both using simple rules and more complex statistical methods. It talks about different tools that turn resumes into organized information—some are easy to use but not very accurate, while others are better but expensive. Additionally, it explains how understanding queries and context can improve finding the right matches. Challenges like writing styles and the need for faster and better detection of resume parts are highlighted.

The research paper [8] study evaluated three classifiers—Decision Tree, KNN, and Multinomial Naive Bayes—applied to categorize text into six topics using an Amazon product review dataset. The classifiers were trained with 4500 samples and tested with 1500, involving text cleanup procedures such as removing less relevant words and simplifying them. Various methods of describing words were tested, and settings for each classifier were adjusted accordingly. The findings revealed that Multinomial Naive Bayes achieved the highest accuracy at 91.8%, followed by KNN at 82.6% and Decision Tree at 79.6%. Notably, Naive Bayes showed the quickest learning speed, while KNN took the longest time for testing. The study highlights the significant performance of a simple Naive Bayes classifier when employing appropriate text cleanup and word description methods for text categorization.

The research paper [9] the Applicant Tracking System (ATS) streamlines recruitment by electronically handling applications based on specific criteria like keywords and skills. It's akin to CRM systems but tailored for recruitment, filtering resumes from sources like job boards and internal applications. Modern ATS, integrating AI and cloud-based platforms, enhances resume sorting, while applicants use optimization techniques to align their resumes with job requirements, improving their chances of securing interviews.

The research paper [10] introduces a Naive Bayes classifier for sorting data into different categories using probabilities. It calculates the chances of data belonging to various groups and picks the most probable one. This method, based on Bayesian theory, assumes independence between attributes for simpler calculations. Developed in Python, the classifier is tested on example data, showing accurate results. It's easy to understand and works well with limited data, but struggles with considering connections between attributes. Overall, it offers a useful toolkit for sorting data into groups, ready for different real-life tasks.

The research paper [11] explores improving Naive Bayes for text sorting by choosing important features using stats methods. It compares CHI and CHIR techniques for selecting positive features. Using these methods on 20 Newsgroups data, CHIR performs better, boosting accuracy from 70-80% to 90-98% for Naive Bayes. CHIR picks fewer but better-balanced features per category than CHI, improving the model's training and accuracy. This method works well with larger data sets, showing comparable results to other sorting techniques. In summary, the study suggests using CHIR over CHI to enhance Naive Bayes' text sorting accuracy based on its better performance in the 20 Newsgroups dataset.

The research paper [12] This paper presents a system using a convolutional neural network (CNN) to sort job candidates as selected or rejected, aiming to make recruitment screening faster and more effective. It uses CNN because it's good at understanding text. The system represents resumes as word patterns and trains the CNN with this data after cleaning it up. It compares CNN's performance with other methods and ranks resumes based on matching job skills. The results show that it achieves 74% accuracy on a Bangladeshi job portal, doing better than other methods. It also talks about how this system can be used with other recruitment tools and improved with more data and better computer systems.

The research paper [13] study introduces a resume screening system using Python and Natural Language Processing (NLP) to sort resumes into job categories. It uses a KNeighboursClassifier model trained on resume text to achieve this. The process involves cleaning up the text by removing unnecessary things like web links, punctuations, and common words. Then, it turns the text into a format the computer can understand and trains a model to do the sorting. Impressively, the system shows a 7% high accuracy of 99% in categorizing resumes into 25 different job categories, with most categories having very accurate results. The study suggests that using this system can automate manual resume screening, making it easier for recruiters and helping to avoid biases in hiring decisions.

The research paper [14] This paper introduces a system that combines Machine Learning (ML) and Natural Language Processing (NLP) to screen resumes and suggest candidates for jobs, aiming to ease recruiters' workload and provide feedback to candidates. It uses methods like named entity recognition and classifiers such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) to analyze resumes, matching skills in resumes to job descriptions for screening purposes. Among various ML models tested, Multinomial Naive Bayes achieves the highest accuracy at 91%. However, it notes limitations in current systems such as static models without room for improvement and accuracy leveling off with larger datasets. The paper suggests potential improvements by exploring more advanced ML techniques like neural networks and incorporating continuous training with new data for better accuracy over time. Overall, this system could streamline the recruitment process by automating resume screening and providing guidance to candidates for resume improvement.

3. PROPOSED METHODOLOGY

3.1 Data Collection

Resumes in this dataset were queried from Indeed.com with keyword 'data scientist', location 'Vermont'. It contains a total of 125 Resumes (33 invited for the Interview and 92 Candidates not invited). If a resume is 'not flagged', the applicant can submit a modified resume version at a later date. If it is 'flagged', the applicant is invited to interview.

3.2 Data Preprocessing

3.2.1Importing Data

Data is imported in CSV format. and the CSV structure of the data is as follows:

Table 1. Data format

resume_id	class	resume_text
…	…	…

resume_id - Candidate CV - ID

class - Target Feature. It indicates if the Candidate was invited for an interview or not.

resume_text - Resume text content scrapped file each file. We later remove the "resume_id" because it's not important for our model build.

3.2.2Exploratory Data Analysis

It is essential to assess the data's quality. The resumes were examined for inaccuracies and gained insights into data patterns. The outcomes of the preceding analysis are as follows:

Table 2. Table captions should be placed above the table

	Columns	Non-null Count	Dtype
0	resume_text	125	object
1	class	125	object

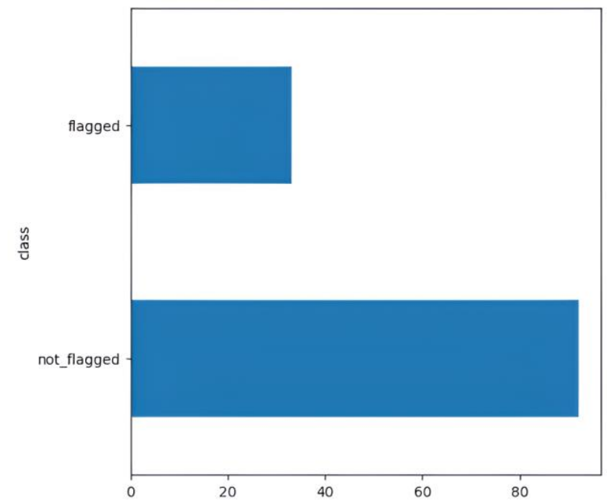

Fig 1: Bar chart of number of flagged and not flagged resumes

3.2.3Data Cleaning

Using class fields labeled as "flagged" and "not_flagged" can be less programmer-friendly and may potentially lead to issues during TF-IDF vectorization. Therefore, the resumes opt to represent "flagged" as 1 and "not_flagged" as 0.

The dataset, characterized by its extensive record count, remains in an unprocessed and unsorted state. In order to enhance its quality, a data cleansing procedure will be executed. This procedure entails the elimination of any extraneous spaces within the dataset, a conversion of all text to lowercase for consistency, and the exclusion of stop words. Stop words, including common terms such as "are," "we," and "is," which carry minimal importance in sentence construction, will be omitted. The resultant refined dataset will be segregated into two primary categories, "query" and "description," and encompasses a total of 10,000 entries.

3.2.4Data Visualization

The resumes were compared of the candidates who were invited to the interview and who were not invited.

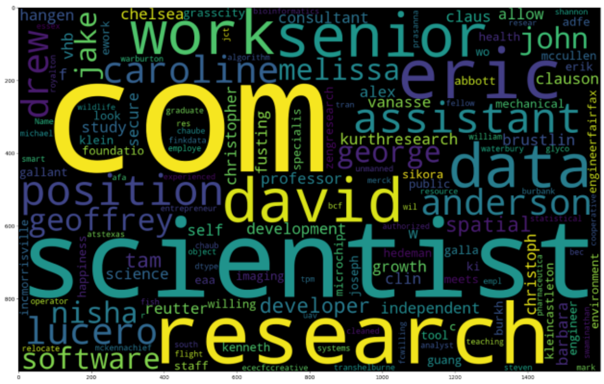

Fig 2: Common words in resumes of invited candidates

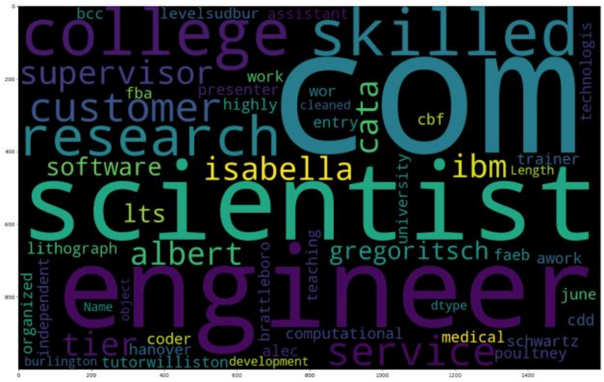

Fig 3: Common words in the resumes of candidates who were not invited

3.3 Data Transformation

Count Vectorization is employed to convert raw resume texts into a structured format by representing them as vectors of term or token counts. This methodology involves creating a matrix where each row corresponds to an individual resume and each column represents a unique term or token found within the corpus. The cell values in this matrix indicate the frequency of each term's occurrence within a specific resume.

The significance of Count Vectorization lies in its ability to translate textual information into a numerical form that algorithms can process. This transformation is essential for the task of resume screening, enabling the application of machine learning and statistical techniques to analyze and make sense of large volumes of text.

3.4 Training Model

3.4.1 Data Splitting

Next, the dataset was partitioned into training and testing subsets to assess the model's performance effectively. Utilizing the train_test_split function from the sklearn.model_selection module, we randomly split the data. Here, 65% of the data is allocated for training the model, while the remaining 35% is reserved for evaluating its performance.

3.4.2 Model Selection and Training

In the pursuit of effective resume screening, the Multinomial Naive Bayes algorithm emerges as a promising solution. This classifier is well-suited for text classification tasks and is particularly adept at handling the count vectors generated by the Count Vectorizer.

After employing the Multinomial Naive Bayes classifier, the resume screening process was further enhanced by training the data with Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) classifiers. These additional classifiers offer distinct approaches to classification, allowing their strengths to be leveraged and potentially improving the accuracy and robustness of the screening system. The combination of these classifiers enables more informed and precise decision-making when evaluating job applicants' resumes.

4. MODEL EVALUATION USING CONFUSION MATRIX

A Confusion Matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

Here are some basic definitions needed to understand the CM Table:

– True Positives (TP): Instances correctly identified as positive by a classification model.

– False Positives (FP): Instances incorrectly identified as positive by a classification model.

– True Negatives (TN): Instances correctly identified as negative by a classification model.

– False Negatives(FN): Instances incorrectly identified as negative by a classification

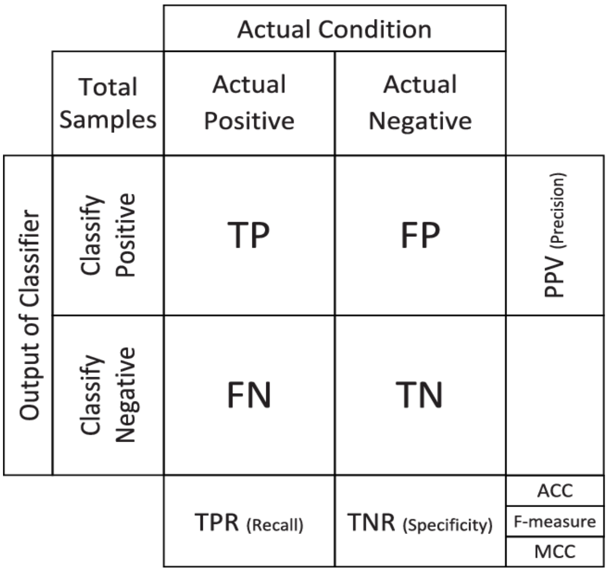

Fig 4: Testing Data

Once the model is trained we predict the performance of the model on train data and test data for each algorithm

4.1 Naive Bayes classifier

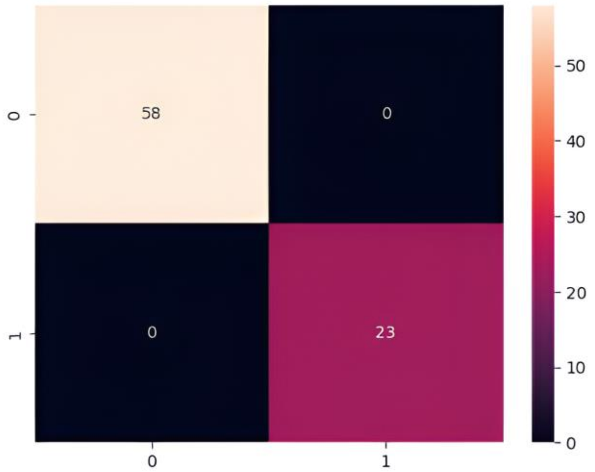

Fig 5: Training Data

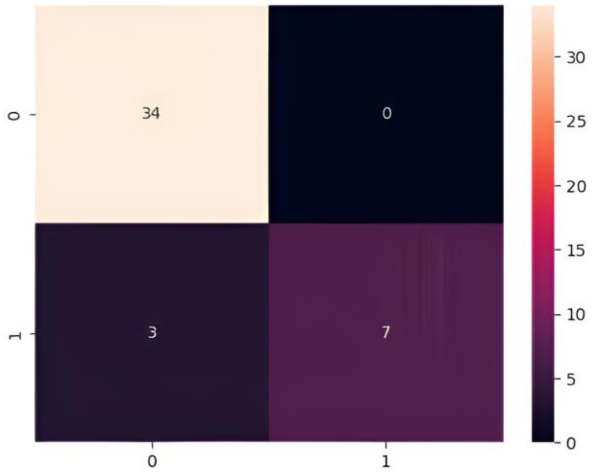

Fig 6: Testing Data

During the training phase, the model displayed exceptional accuracy, with 58 true positives, signifying its ability to correctly identify suitable resumes without any false positives. Notably, there were no false negatives in the training data, indicating the model's proficiency in recognizing all unsuitable resumes. Additionally, 23 true negatives reinforced its effectiveness in distinguishing unsuitable candidates.

In the test dataset, the model's performance remained robust, with 34 true positives and zero false positives, maintaining precision. However, there were three false negatives, highlighting areas for potential improvement in recall. Seven true negatives were also achieved, further validating the model's aptitude in identifying unsuitable resumes. This research underscores the reliability of Naive Bayes classifiers in resume screening, with implications for enhancing HR and recruitment processes by minimizing Type I errors and optimizing recall.

4.2 Support Vector Machine classifier

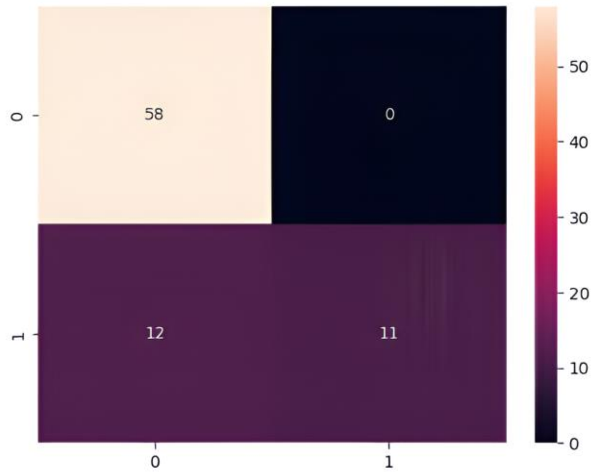

Fig 7: Training Data

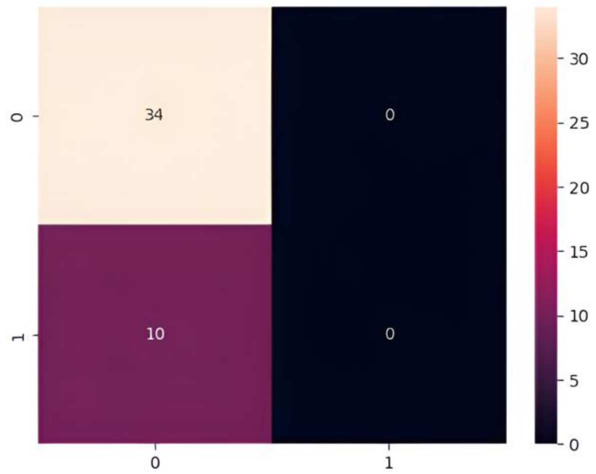

Fig 8: Testing Data

Our findings on the training data indicate that the SVM model exhibited 58 true positives, signifying its capability to accurately classify suitable resumes. Importantly, there were no false positives, underscoring the model's precision in avoiding incorrect categorizations of unsuitable resumes. However, there were 12 false negatives, indicating instances where the model misclassified suitable resumes as unsuitable. Additionally, 11 true negatives were achieved, showcasing its ability to correctly identify unsuitable resumes during training.

When the model was applied to the test dataset, it maintained respectable performance with 34 true positives and zero false positives, emphasizing its precision in recognizing suitable candidates. However, the presence of 10 false negatives suggests potential room for improvement in recall. It's noteworthy that there were no true negatives in the test data, which calls for further exploration of the model's applicability to new, unseen data.

4.3 KNN classifier

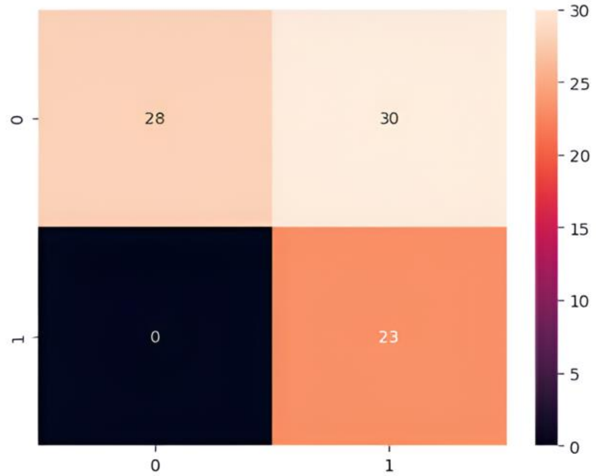

Fig 9: Training Data

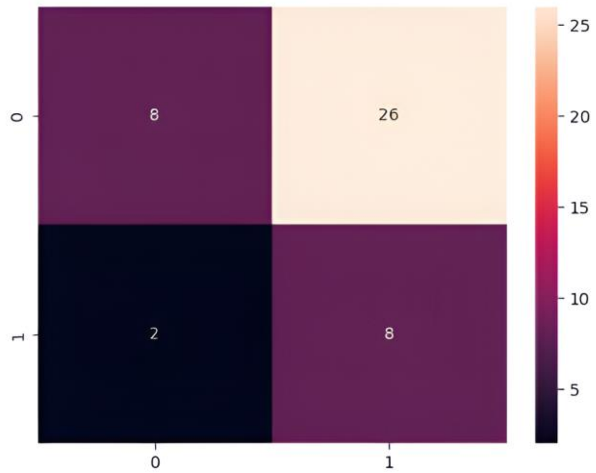

Fig 10: Testing Data

In the training dataset, the KNN classifier displayed a notable 28 true positives, showcasing its proficiency in correctly identifying suitable resumes. However, it also exhibited 30 false positives, indicating a tendency to misclassify some unsuitable resumes as suitable. On the positive side, there were no false negatives in the training data, indicating that the model successfully recognized all unsuitable resumes. Additionally, 23 true negatives reflected its ability to correctly classify unsuitable candidates during training.

When evaluated on the test dataset, the KNN model showed a different set of results. It achieved 8 true positives, implying its capability to identify suitable resumes in this new data. However, there were 26 false positives, raising concerns about its precision when classifying unsuitable resumes. The presence of 2 false negatives indicated instances where the model misclassified suitable resumes as unsuitable. Furthermore, it achieved 8 true negatives, showcasing its ability to correctly distinguish unsuitable resumes in this context.

5. RESULTS & DISCUSSION

A Confusion Matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

Table 3. Confusion Matrix details for Train and Test Data

Algorithm	Dataset	TP	FP	FN	TN
KNN Classification	Train	28	30	0	23
	Test	8	26	2	8
SVM classification	Train	58	0	12	11
	Test	34	0	10	0
Naive Bayes Classification	Train	58	0	0	23
	Test	34	0	3	7

Table 4. Comparing algorithms based on performance metrics

Algorithm	Precision	Accuracy	F1-score
KNN Classification	0.67	0.36	0.36
SVM classification	0.60	0.77	0.67
Naive Bayes Classification	0.94	0.93	0.93

The performance metrics of three classification algorithms, namely K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Naive Bayes, were examined to gain insights into their suitability for various classification tasks.

KNN classification demonstrated mixed results across the training and test datasets. In the training phase, KNN achieved 28 true positives, indicating its ability to accurately identify suitable resumes. However, the model also recorded 30 false positives, which signals a significant tendency to misclassify unsuitable resumes as suitable. This imbalance negatively impacted its precision and highlights the model's susceptibility to making Type I errors. On the positive side, KNN did not produce any false negatives during training, demonstrating its capacity to recognize all unsuitable resumes. Additionally, the model recorded 23 true negatives, reflecting its strength in correctly identifying unsuitable candidates during training.

The test phase revealed some limitations in KNN's generalization to new data. The model achieved 8 true positives, which underscores its ability to identify some suitable resumes, but it also recorded 26 false positives, showing a continued struggle with precision. These false positives could lead to the inclusion of unsuitable candidates in the selection process. Moreover, the model produced 2 false negatives, misclassifying suitable resumes as unsuitable, further suggesting that its recall is not optimal. On a positive note, KNN achieved 8 true negatives, confirming some level of proficiency in distinguishing unsuitable resumes in the test dataset. However, the substantial number of false positives, particularly in the test set, raises concerns about KNN's reliability for resume screening tasks without further refinement or optimization of hyperparameters.

SVM classification demonstrated stronger performance, with a precision of 0.60, an accuracy of 0.77, and an F1-score of 0.67. These metrics suggest that SVM is more balanced than KNN, particularly in its ability to handle both true positives and true negatives. During the training phase, the SVM model achieved 58 true positives, reflecting its capability to correctly classify suitable resumes. It also avoided any false positives, which underscores its precision in avoiding incorrect classifications. However, the model recorded 12 false negatives, indicating cases where it misclassified suitable resumes as unsuitable, thus reducing recall. Additionally, the model recorded 11 true negatives, reinforcing its ability to correctly reject unsuitable candidates during training.

In the test phase, SVM maintained respectable performance with 34 true positives and zero false positives, again emphasizing its strength in precision. However, the model's recall showed room for improvement, as 10 false negatives were observed, which could result in suitable resumes being overlooked. It is also noteworthy that there were no true negatives in the test dataset, suggesting that the SVM model might require additional fine-tuning to generalize well on unseen data, particularly when faced with a diverse range of candidate profiles. This observation calls for further exploration of feature selection or regularization techniques to enhance the model's applicability in real-world resume screening scenarios.

Naive Bayes classification emerged as the strongest performer, with a precision of 0.94, accuracy of 0.93, and an F1-score of 0.93. These metrics highlight its effectiveness in balancing precision and recall. During the training phase, Naive Bayes achieved 58 true positives and 23 true negatives, with no false positives or false negatives, showcasing its exceptional performance in both identifying suitable candidates and rejecting unsuitable ones. In the test phase, the model recorded 34 true positives and zero false positives, maintaining high precision. While the presence of three false negatives suggests a slight room for improving recall, Naive Bayes still outperformed the other models, making it a highly reliable choice for automating resume screening tasks.

Naive Bayes' performance reinforces its suitability for handling textual data, making it a particularly effective model for resume screening applications where accurately classifying relevant candidates is crucial. The minimal number of false positives ensures that unsuitable resumes are not mistakenly passed, while its high recall reduces the chances of overlooking qualified candidates.

This research paper on resume screening reveals Naive Bayes as a promising algorithm for automating classification. Comparative analysis with k-NN and SVM highlights algorithmic influence on screening outcomes. Naive Bayes demonstrates competitive results, emphasizing its efficacy in handling textual data.

6. LIMITATIONS & BOUNDARY CONDITIONS

While the Naive Bayes classifier has proven to be a robust and computationally efficient algorithm for document classification, it is essential to acknowledge certain limitations that may impact its performance in real-world scenarios. Firstly, the algorithm relies on the assumption of feature independence, treating each term in a document as independent of others. This oversimplified assumption may not always align with the complex linguistic structures and dependencies present in natural language, potentially leading to suboptimal classification results. Additionally, Naive Bayes is known to be sensitive to the quality and representativeness of the training data. In situations where the training set is small, noisy, or imbalanced, the classifier may exhibit biased behavior and struggle to generalize effectively to unseen documents. Furthermore, the model is unable to capture relationships or dependencies between features, limiting its ability to discern intricate contextual nuances within the text. Lastly, the algorithm may face challenges when handling continuous data and is prone to the 'zero probability' problem when encountering terms in the test set that were not present in the training data. It is imperative for researchers and practitioners to consider these limitations when employing Naive Bayes for document classification tasks, ensuring a nuanced interpretation of the results and exploring complementary approaches to address these challenges.

While exploring the limitations of the Naive Bayes classifier for document classification, it is crucial to consider specific boundary conditions that may influence its applicability. Firstly, the algorithm performs optimally when applied to well-defined and discretely categorized document classes. Instances where document categories overlap or exhibit ambiguous boundaries may pose challenges to the classifier's effectiveness. Moreover, the Naive Bayes model assumes a static feature set and may not readily adapt to dynamically evolving document structures or topics. The boundary conditions also extend to the scale of the dataset; extremely large datasets may strain the algorithm's computational efficiency, necessitating considerations for optimization techniques or parallelization strategies. Additionally, the performance of Naive Bayes is contingent on the representativeness of the training set; in scenarios where the document corpus undergoes substantial changes over time, periodic retraining may be essential to maintain classification accuracy. These identified boundary conditions underscore the importance of thoughtful consideration and adaptation of the Naive Bayes classifier within the specific contextual constraints of document classification tasks

7. CONCLUSION

In conclusion, this research paper has provided a comprehensive analysis of three classification algorithms – K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Naive Bayes – in the context of text classification. Our study aimed to evaluate the suitability of these algorithms for various text classification tasks and provide insights into their respective performances.

Upon analyzing the performance metrics, it becomes evident that Naive Bayes stands out as the most robust and effective algorithm for text classification. With a remarkable precision of 0.94, Naive Bayes consistently demonstrated its ability to accurately classify relevant instances. Its high accuracy of 0.93 and F1-score of 0.93 further reinforce its superior performance.

While KNN and SVM also have their merits and can be effective in specific scenarios, the results of this study point towards Naive Bayes as the algorithm of choice for text classification tasks. Its simplicity, efficiency, and strong performance make it particularly well-suited for tasks such as resume screening, sentiment analysis, and spam detection.

8. FUTURE WORK

– Feature Engineering and Selection: Explore additional features that could enhance the performance of the classifiers. Investigate the impact of different feature combinations or the creation of new features derived from the existing ones. Feature selection techniques could also be explored to identify the most relevant attributes for resume classification.

– Hybrid Models: Investigate the possibility of combining the strengths of multiple algorithms in an ensemble or hybrid model. For example, you could explore the performance of a voting classifier that combines the predictions of Naive Bayes, k-NN, and SVM to leverage their individual strengths and potentially improve overall accuracy.

In practical applications, the choice of classification algorithm should always consider the specific requirements of the task and dataset characteristics. However, the results of this research emphasize the relevance and effectiveness of Naive Bayes, further validating its position as a top choice in the field of text classification

9. REFERENCES

[1] Pradeep KumarRoy, Sarabjeet SinghChowdhary, RockyBhatia, "A Machine Learning approach for automation of Resume Recommendation system",Procedia Computer Science 167 (2020) 2318–2327.

[2] D. Pant, D. Pokhrel, and P. Poudyal, "Automatic Software Engineering Position Resume Screening using Natural Language Processing, Word Matching, Character Positioning, and Regex," in Proceedings of the 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET), Hammamet, Tunisia, 22-25 March 2022, IEEE, DOI: 10.1109/IC_ASET53395.2022.9765916.

[3] Roberto Salazar, "Resume Screening with Python - Analyzing Candidates Resumes for Jobs Openings", May 3, 2020

[4] Bharadwaj, S. et al. (2022) 'Resume screening using NLP and LSTM', 2022 International Conference on Inventive Computation Technologies (ICICT)

[5] Harsha, T.M. et al. (2022) 'Automated resume screener using natural language processing(nlp)', 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)

[6] Surendiran, B. et al. (2023) 'Resume classification using ML Techniques', 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT).

[7] A. K. Sinha, Md. Amir Khusru Akhtar, and A. Kumar, "Resume Screening Using Natural Language Processing and Machine Learning: A Systematic Review," in Machine Learning and Information Processing, Singapore, 2021, pp. 207–214. doi: 10.1007/978-981-33-4859-2_21.

[8] Md. A. Rahman and Y. A. Akter, "Topic Classification from Text Using Decision Tree, K-NN and Multinomial Naïve Bayes," in 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1–4. doi: 10.1109/ICASERT.2019.8934502. "

[9] Applicant tracking system - Wikipedia", En.wikipedia.org, 2021.[Online]. Available: https://en.wikipedia.org/wiki/Applicant_tracking_system . [Accessed:31- Oct- 2021]

[10] F. -J. Yang, "An Implementation of Naive Bayes Classifier," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 301-306, doi: 10.1109/CSCI46756.2018.00065.

[11] M. J. Meena and K. R. Chandran, "Naïve Bayes text classification with positive features selected by statistical method," 2009 First International Conference on Advanced Computing, Chennai, India, 2009, pp. 28-33, doi: 10.1109/ICADVC.2009.5378273.

[12] M.F. Mridha, R. Basri, M.M. Monowar, and Md. A. Hamid, "A Machine Learning Approach for Screening Individual's Job Profile Using Convolutional Neural Network," in Proceedings of the International Conference on Science & Contemporary Technologies (ICSCT), Dhaka, Bangladesh, 05-07 August 2021, IEEE, DOI: 10.1109/ICSCT53883.2021.9642652.

[13] Pujari, Shradha. (2023). Resume Screening with Natural Language Processing in Python. 10.13140/RG.2.2.17882.11206.

[14] B. Kinge, S. Mandhare, P. Chavan, and S. M. Chaware, "Resume Screening Using Machine Learning and NLP: A Proposed System," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, ISSN: 2456-3307, DOI: 10.32628/CSEIT228240.