iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.62698

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Comparative Analysis of Identification of Customer Name Variations Using Machine Learning Techniques

Vaishnavi Katkar[1], Sandhya Rakhunde[2], Kaveri Raut[3], Simran Godhwani[4]
Department of Computer Engineering, SCTR's Pune Institute of Computer Technology, Pune,India

Abstract: ***The demand for automated validation of new customer accounts, particularly in resolving variations in customer names within extensive databases, is on the rise. This surge in demand is a direct response to the substantial scale at which data for new customers is being input into CRM systems, necessitating manual validation and correction processes. Usually, this process takes a lot of time for manual validation and account creation. The primary goal of this research paper is to identify machine learning methods capable of effectively addressing the challenge of resolving name variations in extensive databases. Therefore, we have developed an architecture which has the capability to identify the variations in customer names and find the correct matched account according to input by sales/operations team members, saving their manual account validation time and help to speed up the process of the sales/op team.***

Our main aim of this exploration is to build a web application based on the higher prediction accuracy of some powerful machine learning algorithm. We have used official sales-operations team data to create the required training dataset as per our problem statement for training our ML model. With an accuracy of 80% and more, the prediction result of SVM and logistic regression shows a significant improvement of accuracy which drives us to develop an Interactive Web Application for identifying customer name variations as per business requirement.

Keywords: Machine Learning Models, Approximate and Exact String-Matching Algorithms, SVM, Logistic Regression, Fuzzy String Matching, Flask, ReactJS, Python.

I. INTRODUCTION

In vast databases, customer names can have multiple variations, and resolving these variations in customer names within extensive databases is our business requirement. The large volume of fresh customer data being entered into CRM systems, which calls for manual validation and correction procedures which we need to automate using the machine learning approach and string-matching algorithms. The primary objective of this research paper is to identify machine learning methods capable of effectively addressing the challenge of resolving name variations in extensive databases.

To achieve this objective, we propose a systematic approach. Initially, we studied the given data thoroughlyand created the required dataset to train our ML model. Before model training, we created a training dataset using fuzzy string-matching algorithms, exact andapproximate string-matching algorithms, and high computational speed to identify matches with a very high accuracy rate. On this training dataset, we applied various ML classification models to get high accuracy inidentifying customer name variations and display the most accurately matched customer account or name to the user as per the user input requirement. In cases where such matches are not found in the system, the user needs to create a new account for that customer, as no previous account exists for that customer. Multiple customer names (subgroups) imply the same common customer's name (parent name) must be mapped based on parameters or attributes from the dataset. example: subgroups like TCS, Tata Motors, Tata Play, Tata Technology, Tata Consultancy, etc. are part of the same parent company, "TATA Groups." This mapping is done based on relevant features from the training dataset, along with exact and approximate string-matching algorithms, and is used to identify variations in customer names. To automate the process and identify the most matched account for customer names, we have used various machine learning models for the classification purpose of customer names. We tried with SVM, logistic regression, and random forest and comparatively studied their accuracy and processing time rate to get high computational speed to identify matches with a very high accuracy rate. We created the seamless and interactive User Interface (UI) for easy access to the entire functionality of our solution for the Sales and Operations Team of Veritas. I created a trainedML model and deployed it on a local server, using Python and a Flask-created backend to provide all the services required to process the user input and display the required output.

This research paper concludes by presenting a potential solution to tackle the problem of resolving variations in customer names within extensive databases by automating the manual validation and correction processes using ML techniques along with string matching algorithms. This entire solution is provided as a full-stack web application with end-to-end processing and can be easily used by the Sales-Operations team of Veritas. Our research not only provides insights into the current state of relevant work in this area but also highlights practical implementation, valuable directions for future research, and limitations.

II. RELEVANT WORK

This section outlines significant contributions in the domain of machine learning (ML) and natural language processing (NLP) that align with our project's aim of identifying customer name variations. The study by Mukku Bhagya Sri et al. [4] marks a significant advancement in string matching algorithms, exploring both established and novel approaches such as the Enhanced Knuth-Morris-Pratt (KMP), Enhanced Boyer Moore, and Enhanced Rabin Karp algorithms. Their research, emphasizing the enhanced KMP for its superior accuracy and efficiency in processing text documents, underscores the critical role of sophisticated string matching in accurate customer name identification. Further exploration by Koloud Al-Khamaiseh and Shadi ALShagarin [5] provides a comprehensive survey on string matching algorithms, categorizing them into exact and approximate matching techniques. Their analysis sheds light on the complexities of aligning patterns with text beginnings, utilizing a sliding window mechanism, and highlights the importance of continuous innovation in the field to address existing challenges and improve solution frameworks.

Lisna Zahrotun's work [6] on comparing different similarity metrics, such as Jaccard similarity and cosine similarity, in text mining and clustering elucidates the efficacy of these metrics in clustering document titles. This study demonstrates the importance of selecting appropriate similarity measures and clustering parameters to optimize the text clustering process, contributing valuable insights towards refining customer name variation identification methods. Additionally, Zhaoyang Zhang's detailed review [7] of key string-matching algorithms like KMP, Boyer-Moore, Bitap, and BNDM emphasizes the strategic importance of optimizing search efficiency through preprocessing and heuristic rules. Zhang's analysis points to ongoing research needs to enhance algorithm performance while minimizing resource consumption, aligning with the objectives of our project in efficiently handling customer data variations. Collectively, these studies lay a robust foundation for our investigation into leveraging ML and NLP for enhanced identification of customer name variations. They not only highlight the technological advancementsin string matching and similarity metrics but also stress the necessity for scalable, accurate, and resource-efficient methodologies in addressing the complexities of customer data management.

III. PROPOSED ARCHITECTURE

Supervised Learning algorithms learns the pattern from pre-existing data and try to predict new results based on the previous learning. ML algorithms are used toidentify existing data like probability-based, function-based, rule-based, tree-based, instance-based,etc. Following Fig.1. indicates the ample architecture of our proposed Model. According to our model membersof the sales-operations team are required to provide a customer account name then the model will find thematching accounts and predict which of them is exactmatched or No-matched. If an account required with matches is found then further sales ticket processing is done else sales-operations team need to create a new account for that customer.

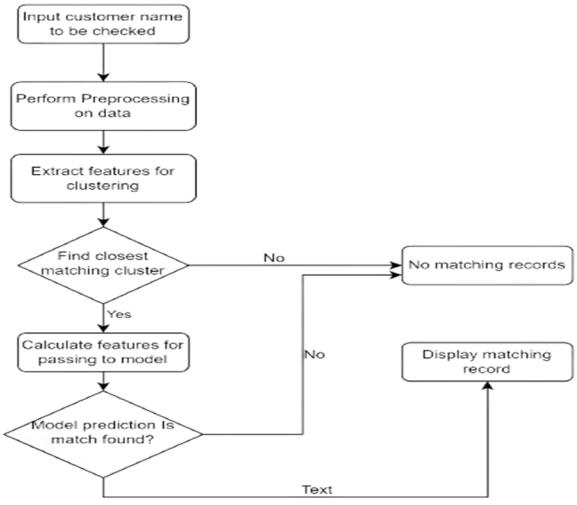


Fig. 1. Company Name Classification

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

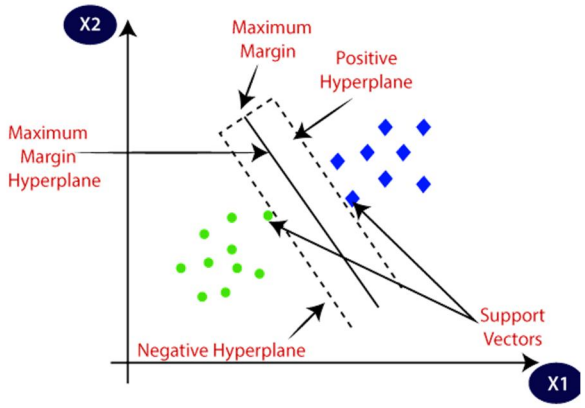
IV. METHODOLOGIES

A. Support Vector Machine (SVM):

Support Vector Machine (SVM) is a supervised learningalgorithm used for binary classification tasks, making it ideal for our goal of predicting whether a customer name is a match or no match. SVM excels at avoiding overfitting by establishing a linear hyperplane with the maximum margin to separate data points into distinct classes, leading to better generalization and higher accuracy.

For our project, SVM was trained on a supervised algorithm using the training dataset. We performed feature engineering to identify the most efficient set of features, optimizing the model's accuracy and predictionefficiency. Compared to other algorithms like Logistic Regression and Random Forest, SVM showed superior performance due to its ability to capture non-linear and polynomial relationships between the variables. SVM will generally perform better on linear dependencies, otherwise you need a nonlinear kernel and choice of kernel may change results.

Fig. 2. Representation of of SVM working Process



The implementation involved typical ML development phases such as data preprocessing, dataset splitting, model training, and evaluation using accuracy metrics. Given its exceptional accuracy and capability to handle variations in customer names, SVM emerged as the most effective algorithm for our project.

B. Logistic Regression:

Logistic regression is a supervised learning algorithm primarily used for binary classification tasks, making it suitable for predicting whether a customer name is a match or no-match. This method establishes a probabilistic relationship between input features and the likelihood of a binary outcome, utilizing the logistic function to map the linear combination of features into aprobability range of 0 to 1. The decision boundary in logistic regression is set at the probability value of 0.5, separating the dataset into the two classes. Unlike Support Vector Machines (SVM), which focus on maximizing the margin to separate classes, logistic regression does not aim for the "best" margin. Instead, itdetermines decision boundaries based on different weights and probabilities near the optimal point. This method is straightforward to implement, interpret, and train efficiently.

However, logistic regression has some limitations. It assumes linearity between the dependent and independent variables, which may not always hold in real-world scenarios. The model may struggle with nonlinear problems due to its linear decision surface, making it less effective for complex patterns in the data.Given these challenges, SVM was chosen over logistic regression for our project due to its superior performance in handling non-linear relationships and providing higher accuracy.

C. Random Forest

Random Forest is a versatile supervised learning algorithm used for both classification and regression tasks. It aggregates predictions from multiple decision trees, each trained on a random subset of data and features. This ensemble method improves the model's robustness and accuracy by combining the diverse trees' outputs through voting (for classification) or averaging (for regression). This approach enhances the model's generalization capability and effectively mitigates overfitting concerns often associated with individual decision trees.

In contrast to Support Vector Machines (SVM), Random Forest does not rely on a linear hyperplane for classification. Instead, it leverages the flexibility of decision trees to adapt to complex patterns in the data, making it well-suited for handling large datasets with high dimensionality. Random Forest's ability to manage multiclass problems and diverse data sets makes it a valuable algorithm in various machine learning applications.

The key advantage of Random Forest is its resistance to overfitting, achieved by increasing the number of treesin the forest. This ensemble technique results in higher predictive accuracy. Despite these strengths, Random Forest may not perform as well as SVM on certain sparse datasets where data is easy to classify. However, it remains a strong contender in situations where data complexity and variability require a robust approach.

D. K-means Clustering

K-means clustering is an unsupervised learning method used to group similar data points into clusters. It is particularly useful in understanding data by organizing itinto distinct clusters. The algorithm partitions a dataset into K clusters by minimizing the sum of squared distances between data points and their assigned cluster centroid. The aim is to form internally homogeneous clusters that are distinct from each other. K-means uses a centroid-based approach, calculating distances to assign data points to the closest cluster centroid. This clustering method is advantageous for its simplicity and efficiency, making it widely used in data mining and machine learning applications. In the context of our project, we use K-means clustering to group records based on three key features: Phonetic Encoding, Named Entity Recognition (NER) based on Location, and Important Word Extraction from Company Names. This approach allows us to create clusters of similar company names. Once an input name is assigned to a cluster, our ML model, specifically SVM, is applied to all records within the cluster to predict whether the input name matches or does not match other records. By clustering the data, we can focus on processing records within a specific cluster, reducing computational load and potentially increasing prediction accuracy. Thistargeted approach enhances the overall efficiency and performance of the project.

E. Key String Matching Algorithms and SimilarityMeasures

Cosine Similarity: Measures the cosine of the angle between vectors, representing term frequency in strings. Widely used in text analysis to determine similarity based on word frequency distributions.

F. Feature Extraction Techniques for Training MachineLearning Models

1) *Simple Ratio:* Simple Ratio is a string matching algorithm that computes the similarity ratio between two strings. It is based on the ratio of the number of matching characters to the total number of characters in the strings.
2) *Partial Ratio:* Partial Ratio is a string matching algorithm that calculates the similarity between two strings by considering partial matches. It identifies the ratio of matching characters between two strings, taking into account substrings.
3) *Token Set Ratio:* Token Set Ratio is a string matching algorithm that considers sets of tokens (words) in the strings. It computes the similarity ratio by comparing the intersection and union of token sets, allowing for partial matches and reordering of tokens.
4) *Word Match Percentage:* Word Match Percentage calculates the percentage of matching words between two strings. It focuses on the commonality of words in the strings, providing a measure of similarity based on word occurrences.
5) *Last Word Match:* Last Word Match identifies whether the last words of two strings match. It is a binary indicator that is 1 if the last words match and 0 otherwise.
6) *Character Matching Percentage:* Character Matching Percentage calculates the percentage of matching characters between two strings, providing a fine-grained measure of similarity.
7) *Ngrams:* ngrams involve breaking a string into a sequence of contiguous substrings of length 'n'. This technique is used to capture local patterns or features within a string, enhancing the understanding of similarities between strings. Common choices for 'n' include unigrams (single characters), bigrams (pairs of characters), and trigrams (triplets of characters).

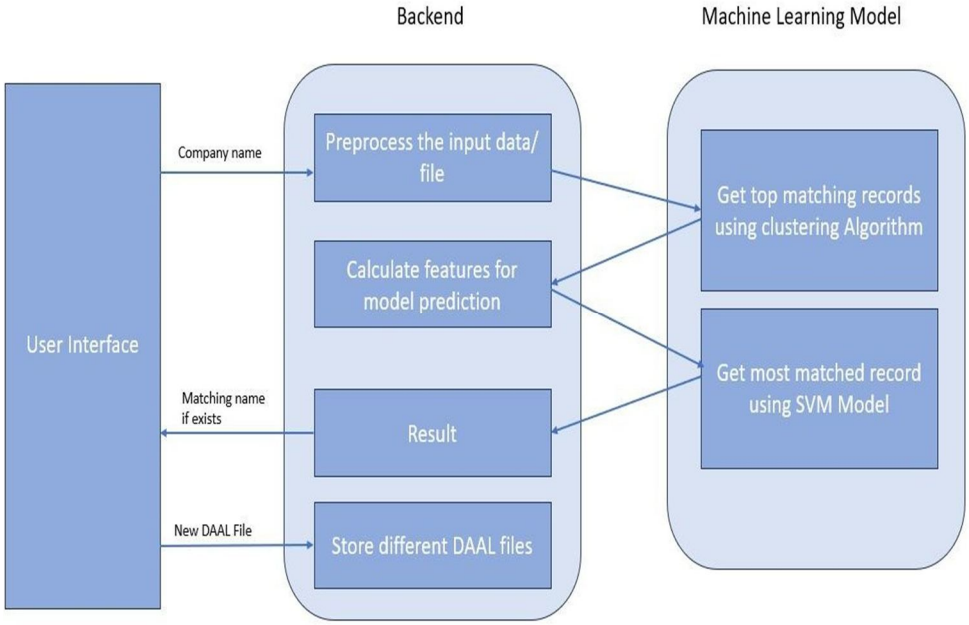
V. DESIGN AND IMPLEMENTATION

As we have mentioned earlier our goal is to develop a Web-based application which can predict whether a customer name matched or not based on provided DAAL file and input data. In order to find the top matching 3 records from a given DAAL file of 20 lakhs + entries we used fuzzy string matching, approximate and exact string matching algorithms. In order to find the most suitable ML algorithm that is capable of predicting customer name matches or not more precisely, we have tested some sort of powerful machinelearning models like SVM, Logistic regression, Randomforest along with string matching algorithms.

For the successful evaluation of these models, we have used some machine learning libraries such as Scikit-learn, numpy, matplotlib, and pandas. In order to get rid of the overfitting problem we split up the dataset into two subparts: one is for testing and another is for training. Based on different training dataset sizes we have achieved the accuracy rate for every defined Model. However, preprocessing of our own created training dataset can produce higher prediction accuracy. Data normalization is an effective way to increase the accuracy of certain machine learning models and some machine learning models do not perform well without data Normalization.

In creating and implementing our solution, we first carefully examined the given data to understand it betterand prepare a suitable training dataset for our ML model. We designed a specialized dataset, making sureit meets the needs of our solution. We used fuzzy and approximate string-matching methods with fast computation to build this dataset. Next, we applied various ML models to accurately identify customer name variations using relevant features and matching techniques. We explored SVM, logistic regression, and random forest models, comparing them for accuracy andprocessing speed. To make our solution user-friendly, we developed a simple and interactive User Interface(UI). We deployed our trained ML model on a local server using Python and a Flask-created backend. Additionally, we created a web application to easily showcase our solution.

Fig. 3. System Architecture Diagram



VI. RESULTS

Identifying gaps is a crucial step as it helps in understanding the specific challenges of the project. In this section we will discuss our results which we have achieved after experimental design. This table providesa concise summary of the accuracy of Support Vector Machine (SVM) classifiers using different kernel functions across datasets with varying numbers of attributes.

High Prediction Accuracy The application of SVM with custom kernels optimized through feature engineering techniques resulted in a prediction accuracy peak of 81.6% for datasets with 7 to 8 attributes. This high accuracy rate substantiates the efficacy of our model in identifying and resolving name variations in customer data.

Efficient Data Processing Our system demonstrated the ability to process extensive datasets rapidly, significantly reducing the time required for account validation. This outcome is pivotal in illustrating the system's capacity to handle voluminous data, a necessityin the context of growing organizational databases.

User-Friendly Interface The developed web application showcased an intuitive user interface, enabling users to navigate and utilize the system with ease. This achievement underlines our commitment to delivering a solution that is not only powerful in terms of computational performance but also accessible and practical for end-users.

Table I. Accuracy for Different Kernels used IN SVM

No. of attributes used	Line ar SVM	RBF non-linear SVM	Poly SVM	Custom(poly + gamma 0.001)	Average
6 attrib.	72	74.4	71.2	75.2	73.2
7 attrib.	71.2	76.8	76	81.6	76.4
7 attrib.	76.8	72.8	76	80	76.4
13 attrib.	77.6	73.6	78.4	78.4	77
10 attrib.	73.6	75.2	76	80	76.2
8 attrib.	72.8	75.2	76	81.6	76.4

(7 attributes- final model) 'Simple Ratio', 'partial Ratio', 'Token Set Ratio', 'word match percentage', 'last word match', 'character_matching_percentage', 'ngrams'

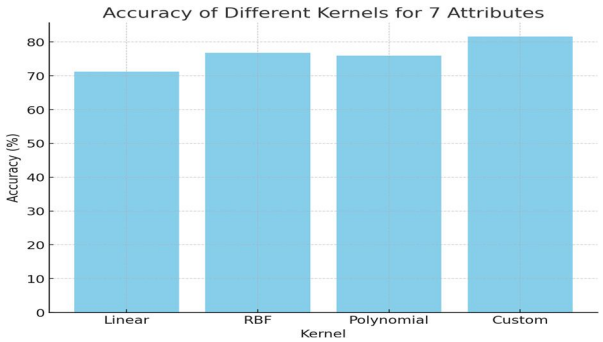
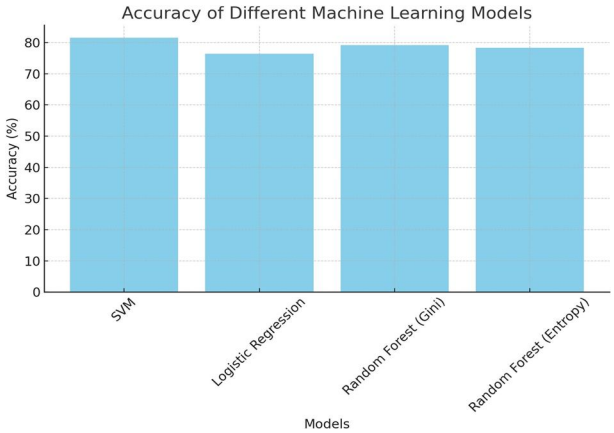


Table II. Accuracy of Different Machine Learning Algorithms

Model used	Accuracy (%)
SVM	81.6
Logistic Regression	76.4
Random Forest (Gini)	79.2
Random Forest (Entropy)	78.4



This table compares the accuracy of various machine learning models on a given classification task. The Support Vector Machine (SVM) model achieves the highest accuracy of 81.6%, outperforming the other listed models. Logistic Regression shows the lowest accuracy at 76.4%. The Random Forest algorithm is evaluated with two different criteria: Gini impurity and Entropy, achieving accuracies of 79.2% and 78.4%, respectively. These results underscore the SVM's effectiveness for the dataset in question, with its accuracy surpassing those of Logistic Regression and both versions of the Random Forest model.

Our developed web application that has been developed based on the highest accuracy of SVM model with the support of feature engineering Methods. Here we have used Python Web programming language i.e. Flask as backend development, JavaScript framework i.e. React JS as frontend and Python for the code implementation of Machine Learning Model. Our proposed architecture usually collects values of dataset from CSV files uploaded by users or default CSV files used and trained ML models help to predict the result (matched or Not matched) according to uploaded CSV files. During the period of prediction, users' needs to provide some information as input. so that our developed web application can predict whether the test result is either match or not. In order to find the company name users need to provide the following information in a web application. Some very necessary information like the DAAL CSV file they need to search in, Input CSV file of new records, Country Code, Company Name, etc. are required.

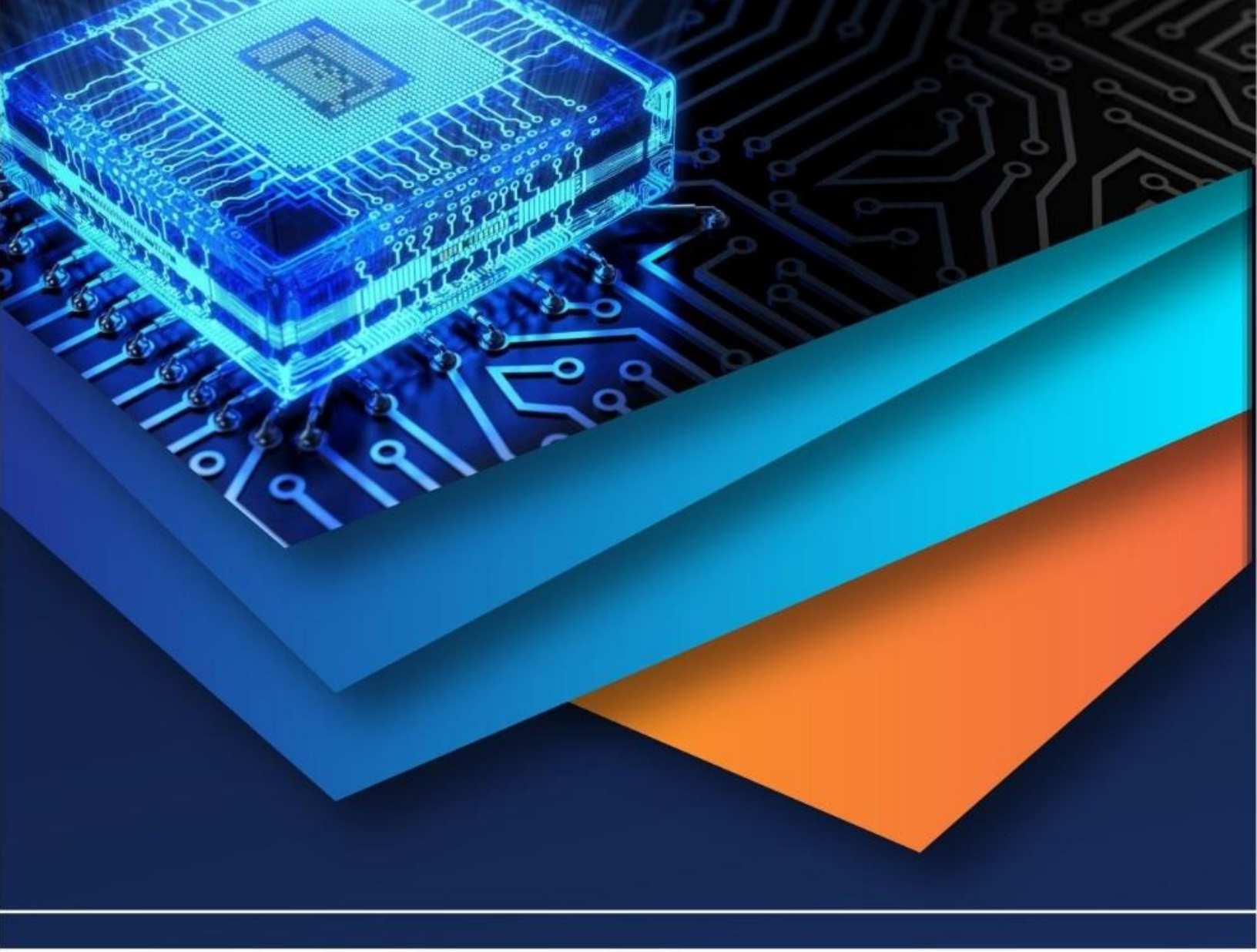
VII. CONCLUSION

In conclusion, our web application effectively identifies variations in customer names using the Support Vector Machine (SVM) model, supported by advanced feature engineering techniques. Through rigorous experimentation with multiple machine learning models such as SVM, Random Forest, and Logistic Regression, we determined SVM to consistently deliver superior accuracy. By employing sophisticated feature selection and engineering methods, we further refined our model'sperformance, ensuring its ability to accurately capture nuanced variations in customer names. We prepared our custom dataset manually to address the unique challenges of name variation identification, whichserved as a solid foundation for training our machine learning algorithms.

This solution has significant potential to enhance data accuracy and insights, which helps in work, particularly in customer relationship management (CRM) and sales operations teams. Future work includes exploring deep learning models for further accuracy improvements and expanding the dataset to ensure accurate pattern capture without underfitting.

REFERENCES

[1] H. Liu, V. Teller, and C. Friedman, "A Multi-aspect Comparison Study of Supervised Word Sense Disambiguation."
[2] M. Joshi, S. Pakhomov, T. Pedersen, and C. G.Chute, "A Comparative Study of SupervisedLearning as Applied to Acronym Expansion in Clinical Reports."
[3] S. Pakhomov, T. Pedersen, and C. Chute, "Abbreviation and Acronym Disambiguation in Clinical Discourse.
[4] Mukku Bhagya Sri, Rachita Bhavsar, Preeti Narooka, "String Matching Algorithms." International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 7 Issue 3 March 2018, Page No. 23769-23772 Index Copernicus Value (2015): 58.10, 76.25 (2016) DOI: 10.18535/ijecs/v7i3.19
[5] Koloud Al-Khamaiseh, Shadi ALShagarin, "A Survey of String Matching Algorithms", Koloud Al-Khamaiseh Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol.4, Issue 7(Version 2), July 2014, pp.144-156
[6] Lisna Zahrotun, "Comparison Jaccard similarity, Cosine Similarity and Combined Both of the Data Clustering With Shared Nearest Neighbor Method.", Computer Engineering and Applications Vol. 5, No. 1, February 2016
[7] Zhaoyang Zhang, "Review on String-MatchingAlgorithm.", SHS Web of Conferences 144, 03018(2022)
[8] Kasra Hosseini, Federico Nanni, Mariona Coll Ardanuy, "DeezyMatch: A Flexible Deep LearningApproach to Fuzzy String Matching."
[9] Shaik Asha1, Sajja Tulasi Krishna, "Semantics-Based String Matching: A Review of Machine Learning Models."
[10] Narendra Kumar, Vimal Bibhu, Mohammad Islam, Shashank Bhardwaj, "Approximate string matching Algorithms."
[11] Y. Kim, J. F. Hurdle, and S. M. Meystre, "Acronyms and Abbreviations Ambiguity in a Diverse Set of Clinical Notes."
[12] E. Ukkonen, "Approximate string-matching with q-grams and maximal matches."
[13] Maryan Rizinski, Andrej Jankov, Vignesh Sankaradas, Eugene Pinsky, Igor Mishkovski and Dimitar Trajanov, " Comparative Analysis of NLP-Based Models for Company Classification."



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY