SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 5.1, (2026), pp. 951-963
Open Access. Online & Print

DOI: 10.5281/zenodo.12511076

AN EMPIRICAL STUDY OF CLASSICAL MACHINE LEARNING METHODS FOR ENTERPRISE TEXT ANALYSIS AND CLASSIFICATION USING AZURE ML

Faraz Islam[1], Tasneem Ahmed*[2],

[1 2] Computer Application Department, Integral University, Lucknow, India-226026,

Received: 11/12/2024
Accepted: 25/02/2025

Corresponding author: Tasneem Ahmed
(tasneema@iul.ac.in)

ABSTRACT

Enterprise service management systems produce voluminous sets of short and heterogeneous textual entries that require the introduction of automated classification routines. While deep learning models based on transformers have repeatedly proved to yield high predictive accuracy, their high computational requirements and low interpretability make classical algorithms interesting for deployments in the industrial world. We performed a controlled comparative evaluation of four classical supervised binary classifiers, i.e., Averaged Perceptron, Boosted Decision Tree, Neural Networks, and Logistic Regression, in a standardized preprocessing pipeline on Azure ML Studio. Classifier performance was measured in terms of accuracy, precision, recall, and F1-Score metrics using a proprietary corpus of labeled service tickets. The findings suggest that logistic regression strikes the best compromise between classification performance, calculation efficiency, and ease of operation. The paper also shows implementation insights and pragmatic recommendations for the design of enterprise-level text analysis systems.

KEYWORDS: Text Analysis, Text Classification, Machine Learning, Averaged Perceptron, Boosted Decision Tree, Neural Network, Logistic Regression.

1. INTRODUCTION

The explosive digitization of enterprise service infrastructures has produced millions of short, unstructured textual records based on customer tickets, electronic correspondence, and support logs accumulated over successive years. Manual interrogation of such volumes of data is not possible, and this limits actionable operational intelligence. Supervised machine learning is a form of machine learning that is scaled to classify text by learning discriminating patterns on past categories of historically labeled records [1, 2]. Most existing research is based on public datasets with vocabularies, class imbalances, and topical domains vastly different from those of real-world enterprise environments; a well-controlled evaluation of classical learning algorithms on real enterprise data is therefore critical for model choice. Transformer-based architectures have achieved amazing accuracy on text classification tasks [3, 4]. Nevertheless, the process of incorporating such models into enterprise production pipelines often faces constraints imposed by computational costs and inference latencies. Classical supervised algorithms, such as logistic regression, models based on perceptrons, and boosted decision trees, still provide benefits in speed of training, resource efficiency, and interpretability [1, 2]. Classical learning methods have been the standard choice for text categorization for some time because of their ability to extract discriminative patterns from labeled data [1, 2]. Over time, a rich array of methods has developed, from probabilistic classifiers, linear classifiers, ensemble classifiers, and neural classifiers [1, 5]. The trade-offs between each of the categories of methods are conclusively different in regard to their accuracy, resilience, computational overhead, and transparency.

Modern scholarship is now briefer than ever before, biased towards deep learning and transformer- based solutions [3, 4]. Enterprise deployments continue to use classical supervised algorithms. Their success in adoption is attributed to their merits that consist of rapid training, low hardware needs, and a high level of transparency in decision-making [6, 7]. In most real-world problems, these properties prevail over progressive enhancements in predictive fidelity. Although a massive literature can be found on text classification at an individual level, the process of searching and finding an optimal algorithm to use in a specific application is not a simple task. The nature of datasets, feature representation approaches, and test procedures renders model selection challenging. In addition to that, comparative studies that test algorithms under heterogeneous experimental conditions are numerous, and these restrict their applicability in practice [8].

In this study, an attempt has been made to address these shortcomings with the objective of the structured and regulated comparison of four common supervised binary classifiers in a consistent and integrative experimental context. The aim is to find the most effective algorithm, which is best when applied to large-scale enterprise text analysis in terms of predictive measure, robustness, and a balance between computational requirements.

The main contributions of the given work are as follows:

- Coming up with a common preprocessing and feature engineering flow for business text information.
- Empirical comparison of four popular supervised machine learning algorithms.
- Holistic performance assessment based on standard classification measures.
- Subjective experience on the choice of algorithm used in scalable text analysis systems.

2. LITERATURE REVIEW

Recent research has demonstrated state-of-the-art accuracy in transformer-based models in text classification tasks. However, application in production streams in enterprises is almost always impeded by the cost and inference delays [3, 4]. However, the question is whether these models can be made cost and inference-aware, such that the delay caused by these models can be low, so that the models can be practically deployed in production streams. Classical supervised algorithms like the logistic regression, derivatives of the perceptron, and boosted decision trees continued to provide better efficiencies, interpretability, and a smaller resource footprint. It is in relatively few cases that comparative inquiries that evaluate several classical models are made under homogeneous experimental conditions, and in particular, prepare enterprises. An effort to fill this gap is given in the current work, through a rigorously systematized analysis of four highly used and common supervised learning algorithms on proprietary enterprise textual data. One of the primary points of the natural language processing and information retrieval process is automated text classification. Initial studies have demonstrated the usefulness of probabilistic algorithms (e.g., Naive Bayes, term-weighting schemes) with high-dimensional textual corpora [9, 10]. Later works focused on the performance of linear classifiers (particularly, logistic regression classifiers) that were more stable and interpretable [6].

The learning paradigm of Perceptron marked the entry of the margin-based and online learning paradigm used for text categorization. The simplest model of a perceptron that was created by Rosenblatt [11] was repelled to be refined, as the averaged perceptron, which can minimize the variance by averaging the weight vectors at every iteration in the teaching and learning steps [12]. These methods have demonstrated a high level of generalization on large texts. The methods of ensemble have become commonly used to enhance the competence of classification, especially boosting. In one of the seminal papers, Friend and Schapire (1997) demonstrated that a significant contribution a large number of weak learners can make to enhancing predictive performance [13]. Recent boosted decision trees have proven to be extremely effective in numerous text mining activities, when dealing with noisy and heterogeneous data [14, 15]. Neural network architecture added the capability to neural network models in text analysis. It has been done with good effect, using shallow feed-forward neural networks with better recall and flexibility in features of an appropriate representation [16]. More recent work has analyzed deep and convolutional neural networks of text categorization: more precise at the cost of computational complexity [17, 18].

There is relatively limited research available on fair, controlled comparisons between more than two classical models in the same experimental setting, despite a substantial amount of research into each algorithm. Classical ones are analyzed under a stable and business-oriented environment, which is particularly uncommon [19, 20]. The research is an addition to the literature, offering a systematic, application-specific appraisal of several supervised learning algorithms of enterprise-text analysis. The machine learning approaches to the field-specific applications, along with the general text classification, are also included in the current literature. Irram and Suaib conducted a thorough review of the machine learning models in medical disease detection, where the survey was narrowed down to the issue of feature selection and ensemble models in enhancing the accuracy of prediction [21]. As an example of the use of AI in biomedical imaging in healthcare applications, Bano et al. offered an AI-based system that is utilised for early cancer detection, using convolutional neural networks and support vector machines [22]. Machine learning has become a subject of more focus in the network security sector. Wasim and Ahmad suggested a light intrusion detection system on the 5G network based on hybrid ML algorithms, in which the computational efficiency is provided and high accuracy is observed [23]. Transformer-based architectures can also be tailored to special tasks; Ahmad et al. suggested Light SAED, a cross-modal transformer model to sarcasm-aware grammar classification of texts, as an example of how such task-oriented model adaptations can dramatically improve performance in challenging NLP conditions [24].

Moreover, predictive analytics in the healthcare field has also emerged as one of the new fields of study. Irram and Suaib [25] have performed an in-depth analysis of the different machine learning classifiers with medical data that includes random forests, gradient boosting processes, and neural networks with deep learning. They made the points of stringent preprocessing of data and feature-engineering (methods) in their analysis to have a plausible model performance. All these studies provide insights into the increasing flexibility of machine-learning algorithms to structured and unstructured data environments and the importance of making a systematic assessment of algorithms, especially in applied and enterprise-scale systems.

3. METHOD

The textual analysis of the enterprise-level workflow starts with the data isolation in some secure SQL Server database, then proceeds with the data transfer to the Azure Storage account, and finally transfers the data to the AzML Studio. It is processed (preprocessed) using an R script that is based on a standardized protocol to retain the English language text and remove any noise. This is succeeded by the picking of a subset of the concerned columns, followed by the process of stratified sampling used to partition the obtained data into training and testing parts. The training fractions are the ones on which the supervised classifiers are trained, and the performance measures in the training as well as the test panels to assess the supervised classifiers would then be determined. The common measures of performance would be the accuracy, precision, recall, and F1-Score. This pipeline allows processing data safely, preprocessing in a stable manner, and evaluation predictably and therefore is deployable at enterprise-scale levels [26]. The complete workflow of the enterprise text analysis is shown in Figure 1.

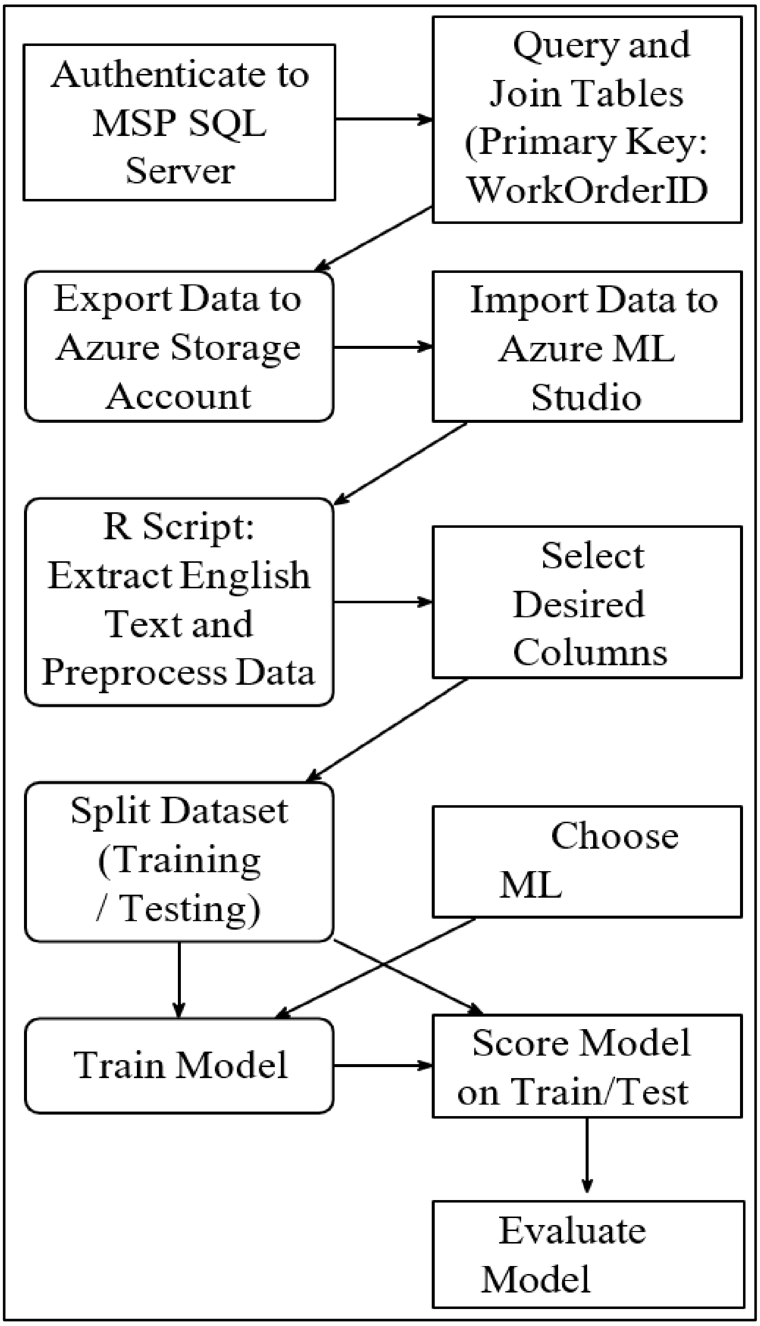

Figure 1: Enterprise text analysis workflow from data extraction to model evaluation.

Once the dataset has been materialized in the Azure ML Studio, an R script is run when the rows containing the English text are determined and scraped, and the data-cleaning standard practices are applied. Figure 2 displays the frequency of the linguistic variants in the raw dataset, hence justifying the necessity to remove English non-linguistic entries prior to the execution of the data analysis to ensure strong and useful outcomes. Then it is necessary and crucial to remove multilingual content entries, as entries of multilingual content may contain noise, thus altering the allocation of features and attenuating the precision of the models of text analysis in the downstream [27].

The second step is the feature selection, which only stores the pertinent columns to be used in reducing the model dimensionality and enhancing the model efficiency. The processed data set is then divided into training and testing data sets so as to enable supervised learning. Once the split has concluded, an appropriate machine-learning algorithm is selected according to accuracy, interpretability, and computational performance. The training is conducted on the training set, and the testing results are evaluated by the scores of the training and the testing data.

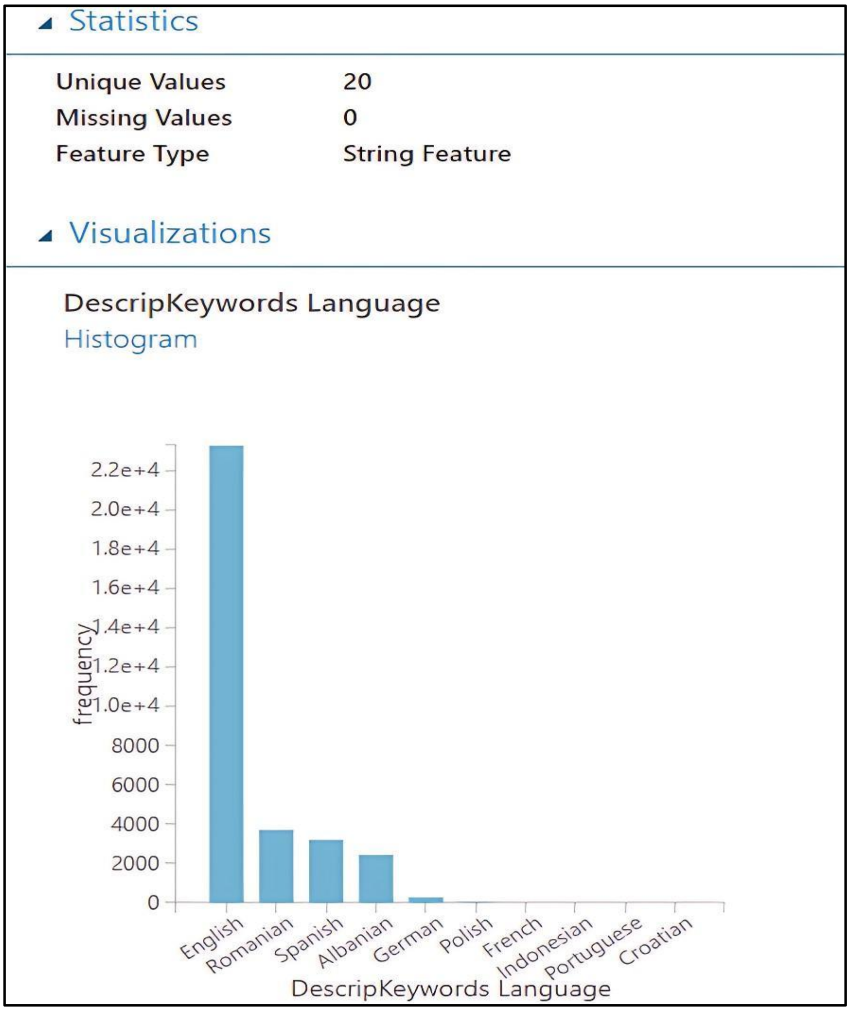

Figure 2: Distribution of different languages present in the Keywords column of the raw dataset.

Finally, assessment is done using conventional evaluation measures, including accuracy, precision, recall, F1-Score, and ROC-AUC to determine whether the model is suitable to be used in the enterprise. Its workflow provides a safe process of data processing, consistency of preprocessing, rigorous analysis, and, therefore, is a solid pipeline to analyze large volumes of enterprise text.

3.1. Dataset and Pre-processing

The data sample consists of 50,000 textual records, assigned binary values (zero or one) to them, extracted from an enterprise service-management ticketing system between the years 2019 and 2025. Class distribution is quite unbalanced, and 60 per cent is in class zero and 40 per cent in class one. Preprocessing includes normalization of text, lowering the words, eliminating punctuations and stop words, stemming using the Porter algorithm [28], and transforming the text into numerical representations of term frequencies, restricted by a maximum vocabulary size of 50,000 words, term-frequency representations [9, 7]. These measures are effective in reducing the noise, decreasing the dimensionality, and improving the performance of the model.

3.2. Machine Learning Models

Four supervised classifiers of binary classifiers were tested, which are discussed as follows:

- **Averaged Perceptron:** A linear model that iteratively updates weights based on misclassified instances. Weight averaging across iterations reduces variance and improves generalization [12].
- **Boosted Decision Tree:** An ensemble of sequentially trained weak learners where each subsequent learner focuses on correcting errors of previous ones. It enhances the precision and strength of complicated decision limits [13, 14].
- **Neural Network:** The Neural Network is a feed-forward system with multiple layers that can be used to perform non-linear modeling of relationships between feature space elements.

The neural networks are expensive in terms of hyperparameter optimization and memory space [16, 18].

- **Logistic Regression:** This is a probabilistic linear classifier, which models the membership of a particular class as a logistic regression. It is human comprehensible, easy to compute, and applicable in text classification at a large scale [6, 29].

4. EXPERIMENTAL SETUP

So as to allow the models to be comparable, all algorithms were trained and tested on the same set of feature representations to facilitate methodological fairness. This was done through stratified sampling to ensure that the class ratio remained the same in the training and the test subset [30]. The evaluation used standard metrics, i.e., accuracy, precision, recall, and f1-score [31]. All classifiers were made to have model configurations, hyperparameters, and training protocols that were standardized. Logistic Regression was chosen as a representative of the unified experimental pipeline used for the entire set of models.

The entire process of the Two-Class Logistic Regression model in the Azure machine learning studio is shown in Figure 3. The process is initiated by data importation, column choice, followed by the language identification and metadata correction, and then an R script is run that leaves only records that include English texts, and then does some initial pre-processing. This final data is further narrowed by the identification of the informative features and divided into training and test data. The Logistic Regression classifier is fit into the training data, and then the performance of the classifier will be evaluated on unseen data using standard metrics such as accuracy, precision, recall, F1-score, and AUC to determine its suitability for deployment. This managed pipeline ensures regular pre-processing, avoids data leakage, and allows model evaluation to be recreated in a cloud-computing framework [26, 30, 32].

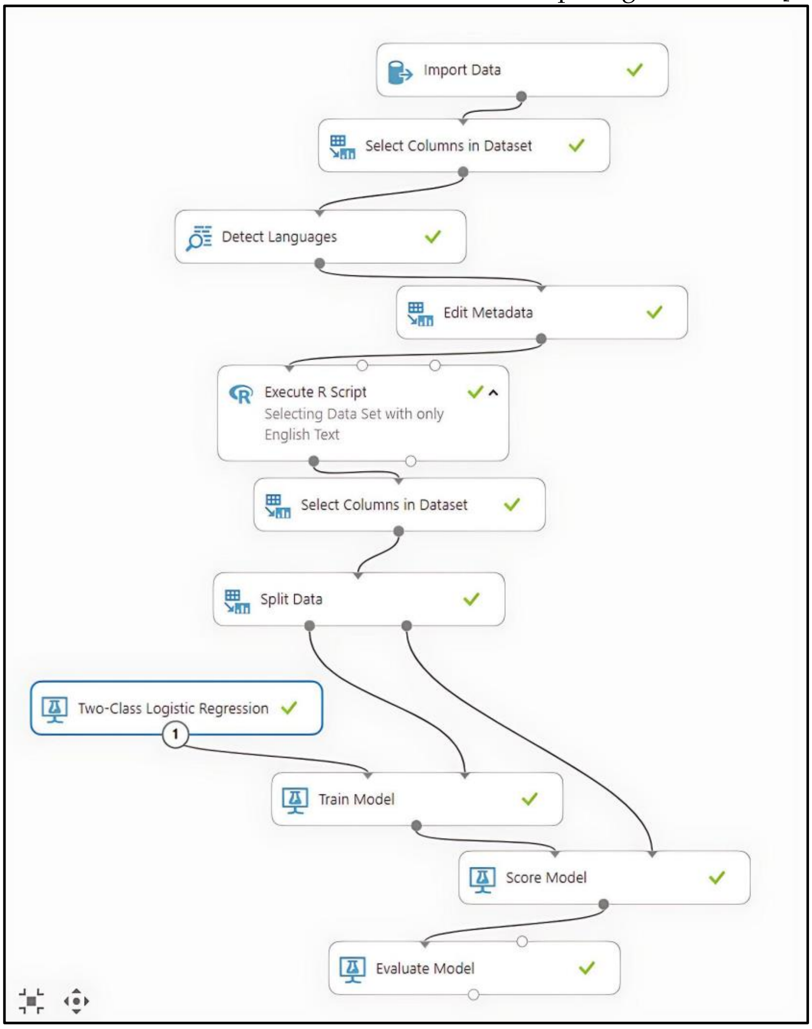

Figure 3: Workflow of the Two-Class Logistic Regression model implemented in Azure Machine Learning Studio.

Figure 4 outlines the configuration settings of the Two-Class Logistic Regression classifier that was used in Azure Machine Learning Studio. The model was trained, taking both L1 and L2 regularization into account, in order to control model complexity and control overfitting, which is important in high-dimensional text feature spaces [33, 34]. To ensure that convergence was stable, we had to have a tolerance of optimization of 10-7. Through multi-threading, computational efficiency was achieved, and an unknown categorical level made it robust when the model was utilized in inference without prior encounter with the data.

The feature weight intellectualization indicates the trained model is assigning good and bad values to various textual features and, therefore, reveals features. Positive weights boost the positive prediction of the classes, whereas negative weights reduce it. Representation of the contribution made by the sampled keywords, feature frequencies, and metadata features that were learned by pre-processing represents such learned coefficients. The offered architecture offers an effective and comprehensible training model, which is why it can be used in analysing texts at the hierarchies of meanings of the enterprise level and scales of meanings of the nuances [29, 35].

Logistic Regression Classifier

Settings

Setting	Value
Optimization Tolerance	1E-07
L1 Weight	1
L2 Weight	1
Memory Size	20
Quiet	True
Use Threads	True
Allow Unknown Levels	True
Random Number Seed	

Figure 4: Configuration settings of the Two-Class Logistic Regression classifier in Azure ML Studio.

Figure 5 describes the training settings of the Two-Class Logistic Regression model in the Azure ML Studio. The training was carried out in a single-parameter setup, and it was defined by an optimization tolerance that was controlled and memory allocated to the L-BFGS solver. The training behavior consistency and reproducibility in an experiment run is guaranteed by a clear specification of training parameters, random seed control, and support for categorical levels. Despite the fact that various classifiers have been experimented with during this study, the logistic regression configuration is a good representative example of the standardized training protocol that is used for all models [30, 36].

5. RESULTS AND DISCUSSION

The performance of classification was assessed with standard criteria. The Neural Network received the largest recall and F1 score of its superior capability in finding relevant patterns. Boosted Decision Trees showed good performance and a moderate level of interpretability. AP averaged lower scores in metrics, which highlights the weaknesses of simple linear models in the analysis of complex text data. Logistic Regression showed competitive accuracy, its precision levels, and low computational cost to be suitable for enterprise deployment. Runtime analysis showed that the Averaged Perceptron and Logistic regression took significantly less time to train and make predictions than Boosted Decision Trees and Neural Networks. The choice of a model should then be a compromise between predictive performance, computational efficiency, and interpretability. Logistic Regression provides a good tradeoff between these criteria.

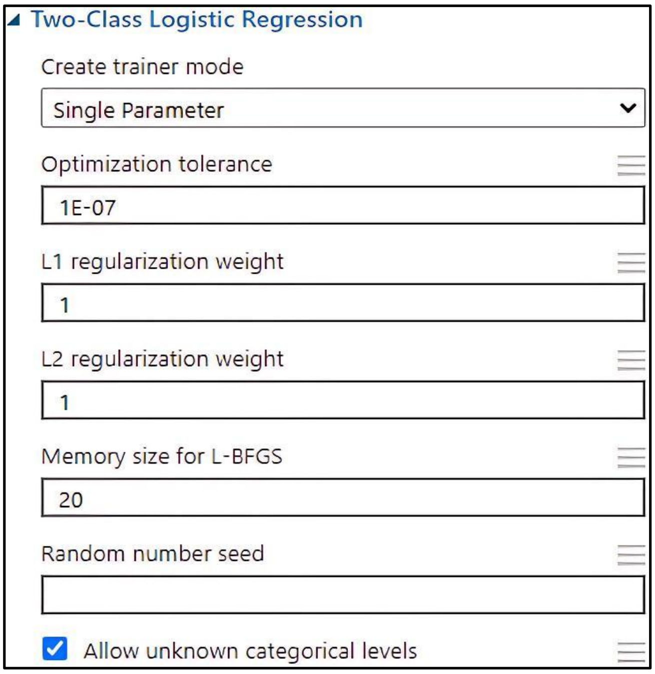

Figure 5: Training configuration for the Two-Class Logistic Regression model in Azure ML Studio.

5.1. AUC Analysis

The Area Under the Receiver Operating Characteristic Curve (AUC) provides a threshold-free measure of the discriminative capacity of a classifier, where an approach to one would represent a better ability of a classifier to discriminate between classes [37]. In this study, AUC analysis has been performed for all the evaluated models, namely Logistic Regression, Neural Network, Boosted Decision Tree, and Averaged Perceptron, by calculating cumulative values of AUC for each score threshold.

Figure 6 shows that the cumulative values of the area under the receiver operating characteristic (AUC) show relatively high discriminative capacity in the Two - Class Neural Network and Boosted Decision Tree model, where the Neural Network has the highest integrated area under the receiver operating curve (AUC), thus showing good performance for all thresholds. Logistic Regression shows moderate performance, and the Averaged Perceptron shows similar performance in the Cumulative AUC, which shows the efficiency of the averaged perceptron even though its architecture is simpler. Overall, all four classifiers have reasonably effective predictive powers under different decision thresholds, with the models yielding calibrated outputs having generally better discrimination.

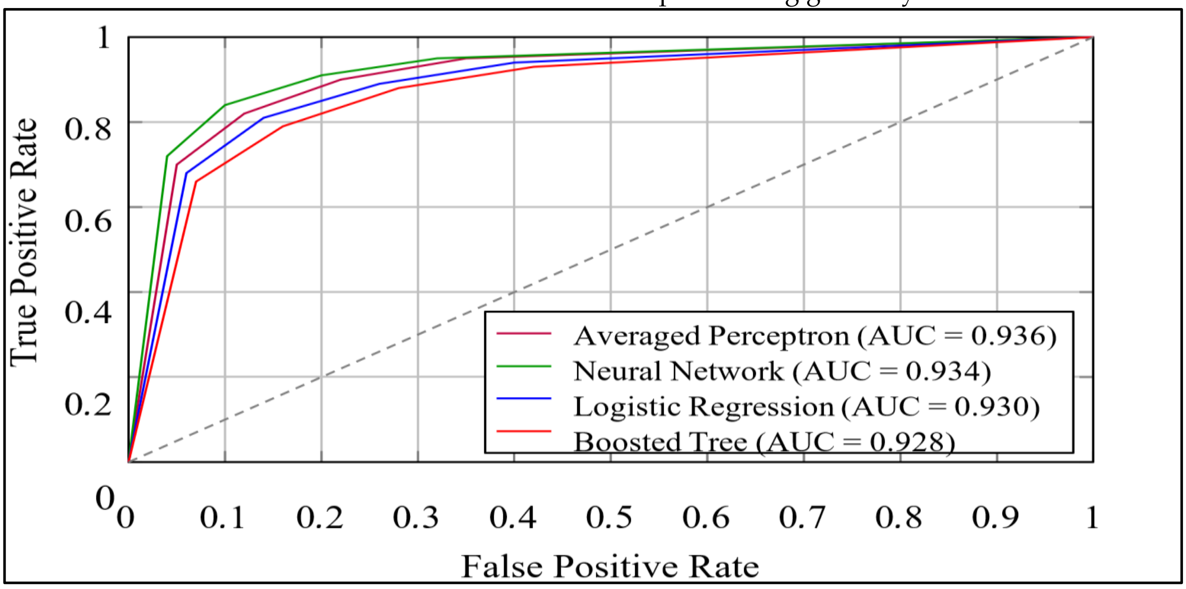

Figure 6: Comparison of cumulative AUC values for the evaluated classifiers. Cumulative AUC represents the integrated performance of each model across all decision thresholds.

5.2. Overall Classification Performance

The summarized performance of models in standard measures has been presented in Table 1. The Neural Network achieved the highest recall and F1-score, which indicates that the neural network has an excellent capability of identifying relevant textual patterns. Boosted Decision Trees exhibited high levels of recall, which was due to ensemble learning. Lower scores were obtained on the Averaged Perceptron, highlighting the limitations of trying to use simple linear models on complex enterprise text. Competitive accuracy and precision were realized with Logistic Regression with lower computational cost, which shows it to be appropriate in enterprise deployment, as confirmed by other studies.

Table 1: Overall performance comparison of machine learning models.

Model	Accuracy	Precision	Recall	F1-score
Averaged Perceptron	0.82	0.80	0.78	0.79
Boosted Decision Tree	0.87	0.85	0.88	0.86
Neural Network	0.88	0.86	0.90	0.88
Logistic Regression	0.86	0.87	0.85	0.86

Further evaluation of the stability and the generalization ability of Logistic Regression was conducted by conducting more experiments with other data partitions, besides the unseen data. Table 2 provides details of two training runs, which were done using different subsets of the data, and also when the data that had already been trained was evaluated. The results show a stable performance across several runs and a moderate decrease in recall and F1-score on untrained data, which is expected because of domain variability. These results support that Logistic Regression has stable and reliable performance under different data conditions, which supports that it can be used for enterprise deployment.

Table 2: Logistic Regression performance across multi-runs and untrained data.

Dataset Scenario	Accuracy	Precision	Recall	F1-Score
Run 1 (Baseline Split)	0.86	0.87	0.85	0.86
Run 2 (Alternative Split)	0.85	0.86	0.84	0.85
Untrained Data (Hold-out)	0.83	0.85	0.81	0.83

5.3. Precision–Recall Trade-off Analysis

Precision and recall are also important measurements in working enterprise text classification. A high precision reduces cases of false positives, whereas a high recall reduces cases of false negatives. It can be seen that the Neural Network produced better recall (see Table 1), whereas the highest precision was achieved by the Logistic Regression, which created a favorable trade-off in the context of enterprise, which supported the superiority of the Neural Network.

Table 3: Precision and Recall Comparison of Machine Learning Models.

Model	Precision	Recall
Averaged Perceptron	0.80	0.78
Boosted Decision Tree	0.85	0.88
Neural Network	0.86	0.90
Logistic Regression	0.87	0.85

5.4. Model Efficiency and Practical Considerations

Other than predictive performance, computational efficiency and interpretability are critical factors to consider when choosing machine learning models when applying them to the real world situation [30, 36, 38]. Models with lower training complexity and increased interpretability are typically fast to deploy, debug, and maintain, a luxury which is often particularly important in a production setting where resources are limited or fast execution is required. A qualitative comparison of the assessed models in terms of training complexity and interpretability is provided in Table 4 to compare the models. Such characteristics as low training complexity and high interpretability characterize the Averaged Perceptron and Logistic Regression, making them more appropriate in the contexts where model transparency and fast training are more important. On the other hand, Boosted Decision Trees and Neural Networks require much more computing resources to train since they have complex structures. Despite the moderate level of interpretability of Boosted Decision Trees, Neural Networks are opaque by nature and may hinder understanding of the decisions of the model and make it difficult to comply with regulatory requirements.

In practice, the choice of a model often involves a compromise between performance and efficiency. As an example, although Neural Networks have demonstrated the greatest predictive statistics (as seen in Table 1), their high computational requirements and low interpretability could limit their use in time and resource-scarce settings in the future [18, 30]. Logistic Regression, however, provides a favorable combination of accuracy, training efficiency, and interpretability, which makes it an attractive candidate to use in working systems in which interpretability and robustness take priority over all.

By and large, the assessment of model efficiency and interpretability, as well as predictive performance, enables practitioners to make effective decisions in accordance with both the technical and operational requirements.

Table 4: Model efficiency and interpretability comparison

Model	Training Complexity	Interpretability
Averaged Perceptron	Low	High
Boosted Decision Tree	High	Medium
Neural Network	High	Low
Logistic Regression	Low	High

5.5. Comparison of Run-Time Performance

Computational efficiency is another vital aspect in the case of text analysis systems of enterprise level, together with predictive performance. It can be impractical to use models or techniques that require too much training or inference time in cases of large data sets or real-time usage. In this regard, the average performance of each algorithm in terms of run-time was measured under the same experimental conditions.

Table 5: Comparison of run-time performance for evaluated algorithms.

Model	Training Time (s)	Inference Time (s)
Averaged Perceptron	5.5	1.9
Boosted Decision Tree	19.8	3.8
Neural Network	23.9	4.5
Logistic Regression	6.2	2.1

Table 5 shows that Averaged Perceptron and Logistic Regression models have much lower training and inference times compared to Boosted Decision Trees and Neural Networks. Though neural and ensemble-based models achieve high predictive performance, their computational cost is high and can hinder scaling when used in large enterprise settings due to challenges in boosting. The Logistic Regression model can strike a good balance as it provides a competitive classification, and significant run-time overheads are minimized by a large margin as well.

5.6. Justification of Model Selection

The justification of the choice of the best-suited model, Logistic Regression, is presented in Table 6 based on the constraints of the enterprise. Unlike strictly accuracy-based selection strategies, this study takes into account the multi-criteria viewpoint that reflects the concerns of deployment in the real world. The models that are in use in an enterprise environment should consider predictive efficacy and computational efficiency in moderation with interpretability and operational risk. Logistic Regression has the advantage of competitive accuracy and a constant F1-Score yet maintains a high precision, which is critical in avoiding false positive classification in ultimate business operations. Its recall is marginally lower than that of neural and ensemble-based models, but at the cost of the added computational cost of more complex models, this is sufficient in the intended application setting.

Operationally, the Logistic Regression has low training and inference times, which make it update models quickly and provide real-time or near- real-time prediction implementation of the model update. In addition, it has a transparent decision-making process that allows practitioners to interpret feature contributions, a requirement to debug, comply, and win the trust of stakeholders. These features, combined, reduce the risk of deployment and maintenance expenses and make the Logistic Regression a feasible and trustworthy option for an enterprise, a scalable text analysis system.

Table 6: Justification for Selecting Logistic Regression.

Criterion	Logistic Regression
Accuracy	Comparative
Precision	High
Recall	Adequate
F1-Score	Stable
Training Time	Low
Inference	Low
Interpretability	High
Deployment Risk	Low

A single analysis of the measures of classification performance and computational efficiency reveals some characteristic trade-offs between the considered algorithms. Two-Class Neural Network and Boosted Decision Tree models returned high values of recall, which confirms a high ability to identify relevant textual patterns. However, these performance improvements were also characterized by a significantly longer training and inference time, which may limit their applicability in large and/or time-sensitive enterprise applications. The Averaged Perceptron and Logistic Regression models had significantly lesser computational overheads. Although the Averaged Perceptron was able to execute quickly and was simpler to generate algorithms, it had a relatively worse performance of accuracy and F1 -score, which reduced its overall effectiveness with respect to classification of complicated text-related tasks in high level of complexity classification tasks. In comparison, Logistic Regression achieved competitive accuracy and precision at low run-time cost. In the case of joint consideration of predictive effectiveness and operational efficiency, Logistic Regression becomes the most appropriate algorithm of text analysis oriented towards enterprises. Its moderate performance, interpretability, and productive execution make it especially suitable for applications in real-world systems of large-scale text classification systems.

6. CONCLUSION

It was a comparative study within the framework of systematic comparative analysis of two-class supervised machine- learning algorithms, including Averaged Perceptron, Logistic Regression, Boosted Decision Trees, and Neural Network regarding the enterprise-scale text classification and analysis. The paper has compared the predictive performance based on accuracy, precision, recall, and F1-score and area under the ROC curve, and presented, at the same time, pragmatic deployment considerations, which include computational efficiency, interpretability, and ORC. With the help of the multi-criteria assessment model, the model choice will allow the current study to ensure the consistency of the model choice with the real enterprise constraints (not only with the performance benchmarks). The experimental outcomes reveal that though Neural networks as well as boosted decision trees have higher chances of recall and high discriminatory power, they are more costly to train and incur higher overhead inference. This form of computation has a very high probability of diminishing scalability and responsiveness to operations in production applications. The Averaged Perceptron, on the other hand, has the disadvantage of being simple and fast, but has a low representational ability of complex textual data, which causes less overall classification performance. All the evaluation dimensions identified the Logistic Regression model as the most balanced and reliable. It obtained competitive accuracy and F1 -score, reached the highest precision of the models evaluated, had stable AUC performance, and has always had low runtime costs. Importantly, its open probabilistic model enables interpretability of the contribution of features, which is a requirement of explainability, regulatory acceptability, and trust of stakeholders in enterprise systems. Therefore, Logistic Regression is especially the ideal one in large-scale, production-intensive text-classification pipelines in which robustness, maintainability, and operational risk are major factors of consideration.

In addition to the empirical observations, this research supports the fact that alignment of algorithmic selection and organizational and operational goals should be taken seriously. Although complex models might provide marginal performance improvements, in many enterprise applications, the added cost of computation, reduced interpretability, and higher maintenance costs are not worth the price of the added marginal gains of these models, which are small. The findings highlight that the well-established linear models are also very competitive when considered as a whole instead of looking at them based on the predictive measures. Further studies will build on this inquiry in a number of directions. To be more precise, the review framework will be utilized with multi-class and multi-label text-classification tasks, which are more representative of the enterprise application case scenarios in the real world. Second, more expressive feature representations, including contextual word embeddings and domain-adaptive representations, will be investigated to maintain efficiency and improve model expressiveness at the same time. Third, the combination of interpretable linear models with representation-learning methods will be explored to resolve the trade-off between performance and transparency in an intermediate space of hybrid strategies. Lastly, future research will entail studies of robustness, fairness, and concept drift to gain an improved insight into long-term model behavior in dynamic enterprise settings. In general, this work indicates that ethical assessment and feasible limitations are necessary to make machine-learning implementation successful.

Although it is comparatively simple, Logistic Regression is still an effective and reliable foundation of enterprise text analytics and can be considered a strong background for future methodological improvements.

FUNDING

No external or institutional funding is received for this work.

CONFLICTS OF INTEREST

There are no conflicts of interest to disclose.

ACKNOWLEDGMENT

The authors are thankful to the Advanced Computing and Research Laboratory, Department of Computer Application, Integral University, Lucknow, for providing the necessary support to carry out the research work. The manuscript communication number issued by Integral University is IU/R&D/2026-MCN0004369.

REFERENCES

[1] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," Mining Text Data, pp. 163–222, 2012.
[2] F. Sebastiani, "Machine learning in automated text categorization," ACM Comput- ing Surveys, vol. 34, no. 1, pp. 1–47, 2002.
[3] J. E. A. Devlin, "Bert: Pre-training of deep bidirectional transformers," NAACL, 2019.
[4] T. E. A. Brown, "Language models are few-shot learners," NeurIPS, 2020.
[5] Y. Yang and X. Liu, "A re-examination of text categorization methods," SIGIR, pp. 42–49, 1999.
[6] A. Ng and M. Jordan, "On discriminative vs generative classifiers," NIPS, pp. 841–848, 2002.
[7] G. Forman, "An extensive empirical study of feature selection," Journal of Machine Learning Research, vol. 3, pp. 1289–1305, 2003.
[8] D. E. A. Lewis, "Rcv1: A new benchmark collection," Journal of Machine Learning Research, vol. 5, pp. 361–397, 2004.
[9] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988.
[10] C. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge University Press, 2008.
[11] F. Rosenblatt, "The perceptron: A probabilistic model," Psychological Review, vol. 65, no. 6, pp. 386–408, 1958.
[12] Y. Freund and R. Schapire, "Large margin classification using the perceptron algorithm," Machine Learning, vol. 37, pp. 277–296, 1999.
[13] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning," Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119–139, 1997.
[14] R. Schapire, "Boosting: Foundations and algorithms," MIT Press, 2013.
[15] Z.-H. Zhou, "Ensemble methods," Foundations and Trends in ML, 2012.
[16] C. Bishop, "Pattern recognition and machine learning," Springer, 2006.
[17] Y. Kim, "Convolutional neural networks for sentence classification," EMNLP, 2014.
[18] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," MIT Press, 2016.
[19] S. E. A. Wang, "A survey on text classification," IEEE Access, vol. 7, pp. 137 160– 137 173, 2018.
[20] K. E. A. Kowsari, "Text classification algorithms: A survey," Information, vol. 10, no. 4, 2019.
[21] S. Irram and M. Suaib, "Trends on machine learning techniques used in medical disease detection," Journal of Dynamics and Control, vol. 9, no. 4, pp. 13–22, 2025, a review of ML methods for disease diagnosis and classification.
[22] R. Bano, N. Hashmi, and U. Fatima, "An AI-based framework for early cancer detection using machine learning techniques," International Journal of Advanced Research, Ideas and Innovations in Technology, vol. 11, no. 5, pp. 221–230, 2025.
[23] S. Wasim and J. Ahmad, "A robust and lightweight intrusion detection system using machine learning for 5 G networks," International Journal of Applied Mathematics, vol. 38, no. 2s, pp. 267–278, 2025.
[24] J. Ahmad, S. Kumar et al., "Light saed: A robust, lightweight, and culturally adaptable cross-modal transformer for sarcasm-aware classification," Journal of Neonatal Surgery, vol. 14, no. 14s, p. 111, 2025.
[25] S. Irram and M. Suaib, "Overview of machine learning techniques for predictive analytics in healthcare," International Journal of Scientific Research and Reviews, vol. 13, no. 3, pp. 45–58, 2025.

[26] C. Shearer, "The crisp-dm model: The new blueprint for data mining," Journal of Data Warehousing, vol. 5, no. 4, pp. 13–22, 2000.
[27] S. Malmasi and M. Dras, "Language identification using classifier ensembles," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2077–2082.
[28] M. F. Porter, "An algorithm for suffix stripping," Program, vol. 14, no. 3, pp. 130–137, 1980.
[29] D. W. Hosmer and S. Lemeshow, Applied Logistic Regression. Wiley, 2013.
[30] M. Kuhn and K. Johnson, Applied Predictive Modeling. Springer, 2013.
[31] T. Mitchell, "Machine learning," McGraw-Hill, 1997.
[32] Microsoft, "Azure machine learning documentation," 2023 https://learn.microsoft.com/azure/machine-learning/.
[33] R. Tibshirani, "Regression shrinkage and selection via the lasso," Journal of the Royal Statistical Society, vol. 58, no. 1, pp. 267–288, 1996.
[34] A. Y. Ng, "Feature selection, l1 vs. l2 regularization," ICML, 2004.
[35] S. Menard, Applied Logistic Regression Analysis. Sage, 2002.
[36] J. Nocedal and S. Wright, Numerical Optimization. Springer, 2006.
[37] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," Radiology, vol. 143, no. 1, pp. 29–36, 1982.
[38] D. E. A. Hand, "Principles of data mining," MIT Press, 2001.