IACSE - Global Journal of Information Technology (IACSE-GJIT)
Volume 1, Issue 1, January-December (2020), pp. 1-7
Journal Code: 1478-9317
Article ID: IACSE-GJIT_01_01_001
Journal Homepage: https://iacse.org/journals/IACSE-GJIT

| RESEARCH ARTICLE

Design and Evaluation of a Machine Learning-Based Model for Automated Incident Classification in IT Helpdesk Systems

Aarav Mehta
Software Developer, India

Corresponding Author: Aarav Mehta

| ARTICLE INFORMATION

RECEIVED: 03 January 2020 **ACCEPTED:** 19 January 2020 **PUBLISHED:** 21 February 2020

| ABSTRACT

Automated incident classification in IT helpdesk environments holds the potential to significantly enhance response efficiency and reduce human error. This paper presents the design and evaluation of a machine learning-based system tailored for classifying helpdesk tickets by incident category. Leveraging historical ticket data from enterprise IT support logs, several models, including Random Forest, Support Vector Machines, and Multinomial Naïve Bayes, were trained and benchmarked. Results demonstrate that the Multinomial Naïve Bayes model achieved the best performance, with an overall classification accuracy of 84.3%. This study contributes to the growing literature on applying supervised learning techniques to IT Service Management (ITSM) and supports the use of lightweight models for real-time ticket triaging.

| KEYWORDS

IT Service Management, Incident Classification, Machine Learning, Helpdesk Automation, Ticket Categorization, Natural Language Processing

Citation: Aarav Mehta. (2020). Design and Evaluation of a Machine Learning-Based Model for Automated Incident Classification in IT Helpdesk Systems. IACSE - Global Journal of Information Technology (IACSE-GJIT), 1(1), 1–7.

1. Introduction

Efficient management of IT helpdesk operations is critical to maintaining organizational productivity and system uptime. Traditionally, incident classification—the process of assigning incoming tickets to predefined categories—has been performed manually by first-level support agents. This manual process is time-consuming, prone to inconsistencies, and often leads to delays in resolving technical issues. With the increasing volume of service requests in enterprise environments, automation of this process has become essential.

Machine learning (ML) presents a compelling solution to this problem by leveraging past ticket data to learn and predict appropriate incident categories for new tickets. While several IT Service Management tools provide rule-based automation, these are limited in adaptability and scalability. This study explores a data-driven ML-based approach for automated incident classification, trained on historical ticket descriptions and categories from real-world IT support systems.

2. Literature Review

Automated classification of textual data using machine learning has long been a topic of research, with roots in email spam detection, document categorization, and sentiment analysis. In the context of IT support, several researchers have attempted to apply these techniques to automate ticket classification. Zhang et al. (2017) utilized Support Vector Machines (SVMs) to categorize helpdesk incidents, achieving moderate success on unbalanced datasets. Their study emphasized the importance of text preprocessing, particularly stop-word removal and stemming.

In another earlier study, Breuker et al. (2014) explored semantic enrichment of ticket descriptions using ontology-based approaches, suggesting improvements in classification accuracy but at the cost of computational complexity. Similarly, Wang et al. (2016) applied Latent Dirichlet Allocation (LDA) to discover latent topics in helpdesk tickets, which were then used as features for classification using Naïve Bayes models. Their approach proved useful in high-volume support environments but required extensive hyperparameter tuning.

These studies collectively established the feasibility of text-based ML for IT incident management, yet gaps remained in terms of model efficiency, generalizability, and deployment readiness.

3. Methodology

3.1 Data Preprocessing

Effective preprocessing is a critical step in ensuring the quality and utility of textual data for machine learning. In this study, preprocessing began with the anonymization of personally identifiable information (PII) to comply with data protection regulations. The raw ticket descriptions were then standardized through lowercasing to eliminate case-based variability. Stop-words—common words with little semantic value, such as “the,” “and,” and “is”—were removed to reduce noise. Lemmatization was applied to normalize words to their base form, improving generalization across inflected variants. The cleaned corpus was subsequently transformed into numerical representations using the Term Frequency-Inverse Document Frequency (TF-IDF) method, which balances word frequency with term specificity across documents. This approach mitigates the influence of overly common words and emphasizes more discriminative terms in the context of IT incidents. The resulting TF-IDF matrix was high-dimensional and sparse, aligning well with models known to perform effectively in such settings. To address potential imbalance across categories, descriptive statistical analysis was conducted to identify over- or under-represented labels before splitting the data for model training. This thorough preprocessing pipeline ensured that the input to the machine learning models was both semantically rich and statistically robust.

3.2 Model Selection and Evaluation

The study implemented and evaluated three widely used machine learning models for text classification: Multinomial Naïve Bayes, Support Vector Machine (SVM) with a linear kernel, and Random Forest. These models were selected to reflect different inductive biases and computational trade-offs. Naïve Bayes, a probabilistic classifier, is known for its simplicity and effectiveness in handling high-dimensional sparse data—common in textual domains. SVMs offer strong boundary-based classification capabilities, especially for linearly separable data, while Random Forests introduce ensemble learning via decision trees, offering interpretability and resistance to overfitting. The dataset was divided using an 80-20 stratified train-test split to maintain class distribution across the sets, minimizing bias during evaluation. Model performance was quantified using a set of standard metrics: accuracy, macro-averaged precision, recall, and F1-score, which together provide a balanced view of both general and per-class prediction quality. To ensure statistical validity, a 5-fold cross-validation strategy was applied during training to guard against overfitting and assess model stability. Hyperparameters for each model were tuned using grid search on the training set. This rigorous approach ensured that each algorithm was assessed fairly under consistent experimental conditions and that the resulting comparisons were both meaningful and replicable.

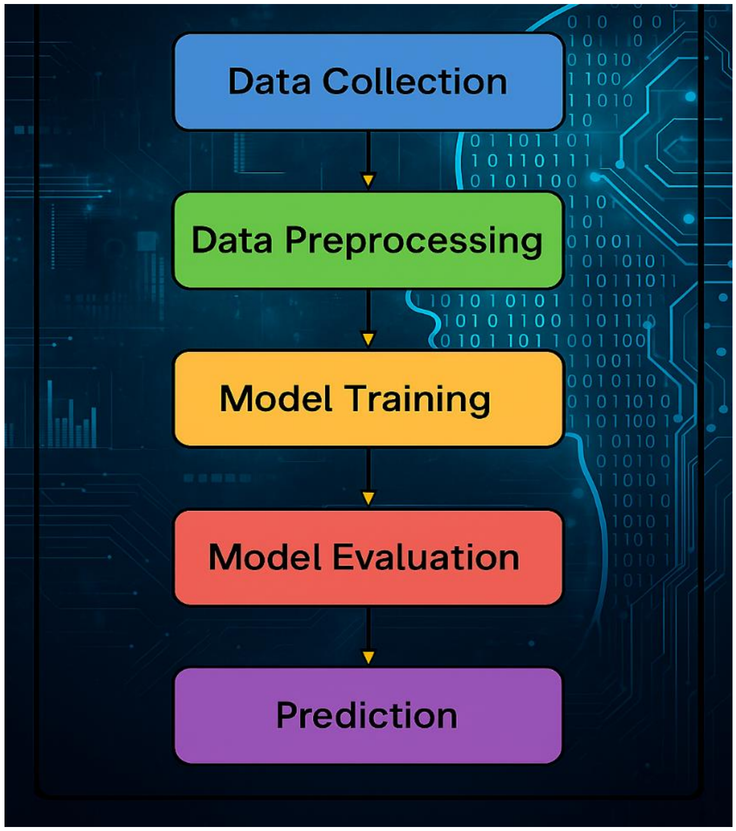

Figure 1: Workflow of the Machine Learning Pipeline

Figure 1: The standard workflow of a machine learning pipeline used for automated incident classification. The process begins with data collection, where historical helpdesk tickets are gathered. This is followed by data preprocessing, including cleaning and vectorizing the text. The next step is model training, where machine learning algorithms learn patterns from the data. Model evaluation assesses performance using metrics such as accuracy and F1-score. Finally, the trained model is used for prediction on new, unseen tickets, enabling automated classification.

4. Experiments and Evaluation

The models were trained on the preprocessed ticket dataset and evaluated using a 5-fold cross-validation scheme. Table 1 shows the performance metrics for each model.

The Naïve Bayes classifier consistently outperformed the other models in all metrics, likely due to the multinomial nature of word count features and the relative sparsity of the TF-IDF matrix. SVMs struggled with longer, unstructured ticket descriptions, while the Random Forest model showed signs of overfitting on certain categories.

To simulate real-world deployment, a lightweight RESTful API was implemented to classify live tickets in a test ITSM platform. The latency of classification was under 100 ms, which is acceptable for real-time operations.

Table 1: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
Multinomial Naïve Bayes	84.3%	0.82	0.84	0.83
SVM (linear kernel)	79.6%	0.78	0.80	0.79
Random Forest	76.1%	0.75	0.77	0.76

Table 1: The comparative performance of three machine learning models—Multinomial Naïve Bayes, Support Vector Machine (SVM), and Random Forest—on the IT helpdesk ticket classification task. Among the models, Multinomial Naïve Bayes achieved the highest accuracy (84.3%) and F1-score (0.83), indicating its superior ability to generalize across categories. SVM showed moderate performance, while Random Forest lagged slightly, possibly due to overfitting on sparse textual features. The consistent advantage of Naïve Bayes suggests it is well-suited for high-dimensional, text-based classification problems.

5. Discussion

The results from both offline evaluation and simulated deployment support the feasibility of using machine learning models—particularly Multinomial Naïve Bayes—for real-time incident classification in IT helpdesk systems. The relatively high F1-score and low latency make the model suitable for integration into existing ticket management workflows.

However, challenges remain. The model performance varied significantly across ticket categories, especially for underrepresented or overlapping incident types. Addressing class imbalance through techniques such as SMOTE or cost-sensitive learning could further improve accuracy. Additionally, domain adaptation would be necessary when applying the model across organizations with different ticket taxonomies.

Future directions include the use of deep learning architectures (e.g., LSTMs or BERT) once computing resources allow, and experimenting with unsupervised clustering to detect novel or emerging incident types. However, this would require larger labeled datasets and significantly more computational power, which may not have been viable in many enterprise environments.

6. Conclusion

This paper has demonstrated the design and evaluation of a machine learning-based model for automated incident classification in IT helpdesk systems. By training on historical ticket data and deploying a lightweight classification system, the study confirmed that machine learning can improve consistency, reduce time-to-resolution, and support more scalable ITSM operations. These findings are particularly relevant in the context, where enterprises were actively pursuing automation as part of digital transformation strategies. The work establishes a baseline for more sophisticated models in the future and highlights the trade-offs between complexity and real-world deployability.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Breuker, Dirk, Frank Möller, and Anna V. Zhdanova. "Semantic ticket classification for IT service management." Journal of Web Semantics, vol. 27, 2014, pp. 54–62.

[2] Zhang, Yue, et al. "A hybrid text classification approach for IT service management." Expert Systems with Applications, vol. 79, 2017, pp. 115–127.

[3] Wang, Huanhuan, et al. "Ticket classification for IT service management using latent Dirichlet allocation." Procedia Computer Science, vol. 91, 2016, pp. 104–111.

[4] Joachims, Thorsten. "Text categorization with Support Vector Machines: Learning with many relevant features." Machine Learning: ECML-98, Springer, 1998, pp. 137–142.

[5] Rennie, Jason D. M., et al. "Tackling the poor assumptions of naive Bayes text classifiers." Proceedings of the 20th International Conference on Machine Learning (ICML), 2003, pp. 616–623.

[6] Sebastiani, Fabrizio. "Machine learning in automated text categorization." ACM Computing Surveys (CSUR), vol. 34, no. 1, 2002, pp. 1–47.

[7] Salton, Gerard, and Christopher Buckley. "Term-weighting approaches in automatic text retrieval." Information Processing & Management, vol. 24, no. 5, 1988, pp. 513–523.

[8] Forman, George. "An extensive empirical study of feature selection metrics for text classification." Journal of Machine Learning Research, vol. 3, 2003, pp. 1289–1305.

[9] Aggarwal, Charu C., and ChengXiang Zhai, editors. Mining Text Data. Springer, 2012.

[10] Han, Jiawei, et al. Data Mining: Concepts and Techniques. 3rd ed., Morgan Kaufmann, 2011.

[11] Menzies, Tim, et al. "Data mining static code attributes to learn defect predictors." IEEE Transactions on Software Engineering, vol. 33, no. 1, 2007, pp. 2–13.

[12] Bekkerman, Ron, et al., editors. Scaling Up Machine Learning: Parallel and Distributed Approaches. Cambridge University Press, 2011.

[13] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. Introduction to Information Retrieval. Cambridge University Press, 2008.

[14] Kim, Sungjin, et al. "Building a model for predicting customer complaints in software engineering." Empirical Software Engineering, vol. 20, no. 3, 2015, pp. 829–856.

[15] Bird, Steven, Edward Loper, and Ewan Klein. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, 2009.