Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

AI-Enhanced NLP for Agile Strategy Execution: Leveraging Machine Learning to Automate Backlog Grooming and Sprint Planning at Scale

[1]Sathish Krishna Anumula, [2]Ishan Kulkarni, [3]Abdul Aleem Syed, [4]Kanwarjit Zakhmi, [5]Shyamsunder Rao Kakatum, [6]Rethish Nair Rajendran, [7]Asadullah Mohammed, [8]Vinesh Melath, [9]Sachin Kumar Agrawal, [10]Manas Ranjan Mohanty

[1]Enterprise Architect Detroit, Michigan
Email: sathishkrishna@gmail.com
[2]Department of Business Administration Carlson School of Management (UMN)
Email: ishankulkarni0707@gmail.com
[3]SVP Technical Product Management FHN Financial, Katy, TX
Email: aleem87@gmail.com
[4]Technical Operations Manager AWS, Portland, Oregon, USA
Email: zakhmikanwarjit@gmail.com
[5]Senior Software Engineer / Lead RouteOne LLC, Canton, Michigan
Email: katakum.rao@gmail.com
[6]Delivery Management, Cloud Infra and Apps Services (US&C) Unisys Corporation, Albany, NY
Email: rethishrnair@gmail.com
[7]Technical Program Manager Kforce Inc., Houston, TX
Email: mohammedasaif@gmail.com
[8]IT Managed Services Genpact, Atlanta, Georgia
Email: vkmelath123@gmail.com
[9]Department of Information Technology Synechron, Charlotte, NC
Email: sachin.research2020@gmail.com
[10]Artificial General Intelligence Amazon, Sunnyvale, California
Email: itsmanasmohanty@gmail.com

ARTICLE INFO

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

Agile methodologies achieved significant efficiency improvements for software development projects through ongoing delivery practices, iterative work cycles, and collaborative teamwork. The 2 efficiency of backlog grooming and sprint planning through manual processes deteriorates when projects grow. This research develops an NLP and ML-powered AI framework that modifies manual. Agile decision-making processes for backlog evaluation and sprint planning operations. The study utilises real-world Jira data from Kaggle, which contains more than 4.9 million records that combine textual description elements with structured metadata details. Logistic Regression using TF-IDF vectors analysed issues to identify their type between Bug, Story, and Task, while XGBoost analysed semantic and structural features to forecast task priorities. The model based on logistic regression produced a complete 100% classification precision alongside XGBoost, which generated 86.3% total accuracy supported by robust ROC-AUC metrics at each priority level. Experimental tests show that the developed framework combines automated Agile decision processes with maintenance of interpretability features alongside high scalability needed for enterprise use. The introduction of explainable models allows Agile teams to understand decisions while gaining trust to utilise the model-generated insights. Real-world data enhances both the reliability and the practical relevance of the system, which surpasses the synthetic datasets that other studies use. The research delivers an automated Agile operation solution that can be easily reproduced and lightened for more innovative context-aware multilingual human-in-the-loop support systems in large-scale software development.

Keywords: Agile Automation, Sprint Planning, Backlog Grooming, TF-IDF, Logistic Regression, XGBoost, NLP, Machine Learning, Jira Dataset

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

INTRODUCTION

Implementing Agile methodologies revolutionised software engineering by emphasising continuous delivery, customer collaboration, and iterative development processes (Karani & Tyagi, 2021). When companies grow their operations, the number of tasks combined with feature requests, bugs and technical debts in the product backlog expands significantly. Agile team backlogs require extensive time and mental effort from management, especially when teams operate across different locations or depend on multiple teams for their work (Wiesche, 2021). The greatest challenge in scaled Agile environments is backlog grooming. Teams must review and refine items while applying priority settings before preparing them for sprints through collaborative meetings. The essential backlog grooming process hinders productivity by slowing down planning sessions and causing dependencies to miss execution dates, and slowing down the delivery pace. The manual approach to categorising and assessing hundreds to thousands of Jira tickets per week proves impossible to scale beyond normal human capacity (Kamath, 2023).

Agile workflows require intelligent automation to address the urgent need in this scenario. The key elements of sprint planning and backlog grooming depend on multiple teams to interpret backlog documentation and understand historical project data concerning organisational priorities. These activities perform regular sequences, which make them suitable for automated data-based processing methods. Artificial Intelligence (AI) and Natural Language Processing (NLP) technologies make it possible for several backlog management tasks to become automated (Reddy & Sushma, 2025). The technology of NLP enables organisations to categorise job descriptions alongside the identification of essential information points (e.g., type of work - bug or feature or enhancement) and key points (urgency level, complexity rating and dependencies). Concepts from machine learning enable the development of models that forecast priority levels and resolution durations and distribute sprints using historical information collected from Jira (Omri & Mribah, 2022).

Integrating AI within Agile project management technical infrastructure facilitates agile decision processes while maintaining uniformity throughout classification activities, triaging procedures, and planning operations (Anjum & Chowdhury, 2024). Natural Language Processing allows machines to analyse verbal descriptions of issues through which backlog items acquire their context primarily from textual information. The combination of Term Frequency–Inverse Document Frequency (TF-IDF) analysis with Logistic Regression creates an effective classification method. Also manages tasks using low-cost computing power and delivers results that Agile professionals can easily understand (Ibadin et al., 2025). XGBoost performs high-performance ML tasks thanks to its extreme scalability characteristics, which help backlog item prioritisation prediction through status and issue type features alongside resolution times and textual indicators. Such models form an operational integrated automation approach which enables Agile teams to execute efficient and scalable sprint planning.

The main goal of this research establish an AI framework which automates Agile performance through backlog grooming and sprint planning tasks. The system employs NLP to extract information from backlog items before employing ML methods to predict priority levels, which enables more automated sprint scheduling. The study investigates two main research questions: whether TF-IDF paired with Logistic Regression ensures correct classification of Agile backlog items such as Bug, Story, and Task through their textual descriptions. XGBoost demonstrates adequate capability to forecast task priority, which leads to simplified automated sprint selection.

A novel aspect of this research emerges because it examines authentic Agile artefacts gathered from the Jira dataset shared on Kaggle. This study adopts real project issues instead of previous studies, which used synthetic or simulated data to validate its proposed models empirically. This paper establishes a unified scalable process combining NLP and ML to handle classification tasks and Agile planning procedures inside Agile workflows. The TF-IDF + Logistic Regression with XGBoost solution provides sufficient performance and explainability features to serve enterprise Agile tools, including Jira, Trello and Azure DevOps.

The research is divided into three main sections, starting with a literature review that examines Agile execution alongside software engineering NLP and sprint planning with ML in Section 3. The fourth section showcases methodology details, including elements about dataset preparation, model design, and evaluation metrics. The experimental results, together with visual analyses, appear in Section 5. Section 6 offers a discussion on the

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

implications and limitations of the study. The final part of this work describes specific avenues for future research that would lead to adaptive and explainable Agile decision-support systems.

LITERATURE REVIEW

Agile Workflows and Scaling Issues

The software development industry has accepted Agile methodologies with Scrum as its standard framework for work, which serves startups as well as large-scale enterprises (Anjum & Kabir, 2019). These approaches deliver their main value through adaptable solutions, stepping progress cycles, and effective teamwork. The Scrum framework operates through time-constrained sprints that execute planning alongside execution and review alongside retrospection during multiple collaborative cycles. Behavioural backlogging requires collaboration between product owners and their teams to evaluate and prioritise items before they enter future sprints. The process requires many human resources and becomes more complicated for scaled operations (Brankovic, 2025). Enterprise Agile usage results in the explosive growth of backlog items, which include features and bugs and technical debts, reaching numbers that exceed hundreds to thousands of unresolved tasks.

The sprint planning ceremony experiences the same complexity during scale transformation after backlog grooming. Accurate effort assessments, proper item classification, and solid prioritisation become essential during planning to guarantee that business goals cooperate with available development resources (Anjum & Alam, 2019). Traditional approaches depend on human judgment for analysis, which produces inconsistent decision outcomes while generating delays and cognitive exhaustion among participants (Hood & Al-Oun, 2014). Since this context emerged, issue-tracking systems such as Jira have played a vital role in Agile activity organisation. Jira maintains an electronic collection of Agile execution data by storing backlog items and their textual summaries, status records, issue types, and priority levels. The vast amount of structured and unstructured data accessible in a system creates possibilities for AI methods to carry out automated planning and grooming operations (Dixit & Jangid, 2024).

NLP in Agile Systems

Natural Language Processing (NLP) technology has become prominent in software engineering for obtaining valuable intelligence from development documentation, task descriptions, and user stories. Backlog items in Agile use free-text summaries, which show inconsistent and informal patterns in their composition. Traditional NLP approaches such as Bag-of-Words (BoW) and TF-IDF (Term Frequency–Inverse Document Frequency) gain popularity because they work well for representing short texts in specific domains (Raza, 2020). The models transform textual data into numerical features so users can classify various tasks, including clustering and semantic similarity analyses. Agile systems employ TF-IDF successfully for duplicate ticket detection, issue groupings, and user story classification into bug, feature or chore categories.

Enterprise applications encounter limitations when using premier contextual models BERT or GPT-2 since their recent popularity supersedes their applicability in practical environments (Sriwastwa et al., 2023). Neural information models demand extensive computational resources to operate while needing extensive labelled databases and delivering weak interpretability of processing. In contrast, simpler NLP techniques like TF-IDF offer transparency, faster training times, and better suitability for projects with moderate computational budgets (Jangid, 2020). Agile teams benefit from traditional NLP models using these solutions between performance considerations and interpretability because they better match issue-tracking tool data and their practicality for actionable insights and fast iteration cycles.

ML in Sprint Planning

Machine Learning (ML) demonstrates its value through predictions about multiple software development characteristics, including defect severity levels, time needed for issue resolution, and required effort durations (Viraj Soni, 2024). The strength of ML models becomes crucial for Agile sprint planning because they help teams improve their workflow. The utilisation of immutable rules that generate results like “bug” and “high” severity tasks going to the following sprint creates a system that cannot adjust to changing project demands (Truss, 2024). Predictions generated through ML systems show the ability to modify their response based on new data patterns drawn from past project data.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Previous research shows that the classification of software issues with respect to priority and type, along with estimated resolution time, succeeds using Logistic Regression models (Sankhe & Dixit, 2024). Logistic Regression retains its popularity because users find its interpretation methods simple and it demands limited data input. XGBoost and Gradient Boosting algorithms have become highly effective due to their performance for structured prediction tasks with textual and categorical features. Multiple domains successfully utilise XGBoost because of its impressive features, including robustness and high speed, together with its capability to detect complex variable interactions. XGBoost processes textual indicators (e.g., word frequency from summaries) with categorical data (e.g., issue type) and numerical data (e.g., historical resolution times) in Agile contexts for predicting backlog priorities and forecasting sprint completion likelihoods (Annadata, 2023).

Research Gap

The implementation of AI within Agile development attracts growing interest, but researchers still need to close important gaps that prevent full-spectrum Agile task automation through the combination of NLP and tabular ML models. The current body of research separates backlog item classification from sprint prioritisation when studying Agile development by handling these problems independently despite the possibility of combining them in a single system (Singh, 2024). Most present research uses synthetic and academic datasets that do not convey the complete real-world Agile environmental complexity found in actual practice (Deguine et al., 2023).

The use of publicly available domain-specific datasets derived from Jira remains scarce because they include both textual descriptions and structured metadata. The collected datasets are suitable platforms for developing complete AI operational frameworks that model real business situations. The literature presents an inadequate investigation of combined TF-IDF with Logistic Regression alongside XGBoost and lightweight interpretable models (Raza, 2020). The literature shows a gap related to Agile automation through the lack of operational pipelines that are scalable and explainable, and this study aims to resolve this issue.

METHODOLOGY

Proposed Methodology Framework

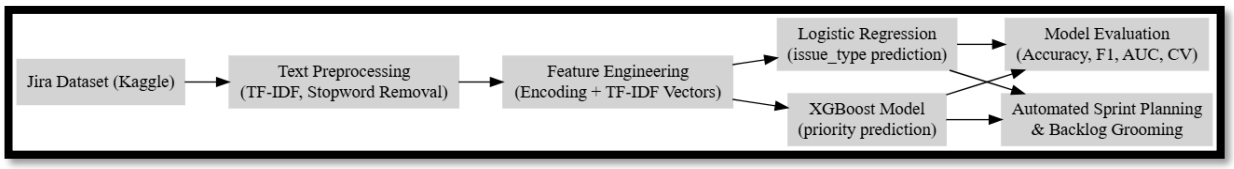

Figure 1: Proposed Methodology Diagram

Fig 1 presents a modular AI pipeline beginning with a Jira dataset and progressing through text preprocessing and feature engineering. TF-IDF vectors and label encoding feed two models: Logistic Regression for issue classification and XGBoost for priority prediction. Model evaluations validate performance, ultimately enabling automated sprint planning and efficient backlog grooming.

Dataset Description

The data set used in this work is obtained from Kaggle, and the source of this data is Jira Agile Boards, where the author has public access since the original data is pre-processed, cleaned and encoded for modelling purposes. The repository contains records of software problems like bugs, stories, and tasks for more than several sprints. Every issue is tagged with semi-hierarchical and non-hierarchical properties in order to employ NLP and Machine Learning schemes in a single issue. Thus, the most pertinent features for this study are issue_type, summary, description, priority, status and resolution_time.

The issue_type field, which subdivides the tasks into only three categories, namely "Bug", "Story", and "Task", defines the functional context of the task within the process of development. The summary and description fields consist of plain text that holds the key focus of backlog grooming with team members articulating technical issues, business requisites, or implementation requirements. Priority is a relative measure of the sales opportunity urgency (Highest,

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Highest, medium, low) that helps make decisions during sprint planning. The status field represents the issue's current work state (To Do, In Progress, Done), while the resolution_time field describes the time taken to complete each task. These features are suitable for both semantic categorisation and sprint-level optimisation. Therefore, the dataset is suitable for adopting AI-related automation in Agile environments.

Preprocessing and Feature Engineering

Due to the nature of the dataset, the analysis was done using a pipeline of transformations to turn the raw data into features for ML algorithms. The first operation undertaken was used in the textual data's summary and description fields. They were further pre-processed through the typical text cleaning procedures involving stop words, elimination of punctuation and symbols, and word conversion to lowercase. Stemming and lemmatisation procedures were omitted to keep specific words from the particular valuable domain for classification.

After text pre-processing, both fields were transformed using Term Frequency–Inverse Document Frequency (TF-IDF) to convert textual data into quantitative data as a high high-dimensional vector. TF-IDF focuses on the distinctive and valuable terms in the dataset and is especially relevant for brief software development texts because terms like “null,” “exception,” or “login” may be essential. The process used for the text vectorisation was named bi-grams to reduce the sparsity, while the features included only the 3000 most frequently occurring terms.

$$\mathrm{TF-IDF}(t, d) = \mathrm{tf}(t, d) \times \log\left(\frac{N}{\mathrm{df}(t)}\right)$$

Regarding the classification task, the issue_type feature was encoded using label encoding, which translates the categorical labels “Bug”, “Task”, and “Story” into numeric terms. These are the target variables for which the logistic regression model was trained. The additional and more elaborate data fields were created in the context of the sprint planning task using XGBoost. Other features were derived from the structure of the items as well, in the form of the TF-IDF vectors. The independent variables reflected on the dependent variable included word count of the description field, the status transformed into ordinal numbers, the type of the issue and the time taken for its resolution. To achieve this, the XGBoost model incorporated this feature space containing the semantic context of the tasks and desired learning outcomes and the behavioural history of the tasks in the project works.

Model Pipeline

For this study, two machine learning pipelines were developed to achieve two goals: classifying backlog items and predicting their priority for inclusion in sprint planning.

For the first task, the terms frequency based on the TF-IDF model and logistic regression model were used to classify the backlog items into one of the following categories: Bug, Story or Task. Logistic Regression was chosen because of its interpretability, speed and decent performance in the classification of text data. This was done using L2 regularisation of the model and training on the text vectorised using TF-IDF as the text pre-processing technique and tuning it for multiclass classification using the one vs rest approach.

In the second task, the XGBoost model was used to target each task's priority level. As noted above, this model used all the hybrid features mentioned earlier. XGBoost was selected due to its high performance, good stability when introduced to features of different types, and high accuracy when classifying predictors with both categorical and numerical values. The values of the control factors that include ‘max depth’, ‘learning rate’, and ‘number of estimators’ were optimised using the grid search with five-fold cross-validation. The model’s output was in the form of a multi-class prediction of the priority level, which was classified into five categories, namely “Highest”, “High”, “Medium”, “Low and “Lowest”.

Thus, both models are combined into a single automation process: TF-IDF + Logistic Regression for grooming automation and XGBoost for sprint planning with knowledge of the other model’s results. This design allows for classifying backlog items and providing recommendations on what should be included in the next sprints.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Evaluation Metrics

A set of classification measures, common in similar studies, was used to evaluate the model's performance. These were, therefore, the Accuracy, Precision, Recall, and the F1-Score. Accuracy is the percentage of the total instances appropriately classified at the terminal nodes (De Diego et al., 2022). Precision defines how well the model avoids classifying things from other classes as of the chosen class, while recall measures how good it is at selecting true samples from a designated class (Miao, J & Zhu, 2022). Precisely, as in the initial experiment, the F1-score, a harmonic mean of precision and recall and computed using two ratios referring to the predominant class and less frequently occurring minority class, was viewed as the key evaluation criterion, mainly due to the presence of class imbalance in both the classification and priority labels.

Furthermore, to have a clear understanding of different misclassifications, confusion matrices were created to present them comprehensively and show the performance of the model for different issues and their priority levels. For the XGBoost model, another performance metric was also computed, namely the ROC-AUC (Receiver Operating Characteristic – Area Under the Curve), to identify the model's performance regarding discriminating between the classes. This was particularly helpful in determining the distinction between two adjacent priority levels (for example, between “High” and “Medium”).

In the entire modelling process, a five-fold cross-validation technique was used to check the validity of the developed models. The data was partitioned, and equal partitions were allocated for training and testing the model. Each partition had an equal number of classes, so the distribution of the classes was maintained throughout the partitions. Cross-validation achieved this, which checked that a particular partition of the dataset did not influence the results obtained, enhancing the credibility of performance measures.

RESULT

Dataset Summary:

	Issue id	Votes	Custom field (Backlog Order (Obsolete))	Custom field (Bugmaster Rank)	Custom field (Comments)	Custom field (Global Rank (Obsolete))	Custom field (JIRA Support Rank (Obsolete))	Custom field (Last commented)
count	4.900000e+04	49000.000000	4.900000e+04	4.900000e+04	49000.000000	4.900000e+04	4.900000e+04	4.900000e+04
mean	1.672930e+06	2.085000	9.214149e+18	9.214149e+18	1.677000	9.214149e+18	9.214149e+18	6.489383e+07
std	1.808238e+05	19.855121	2.915257e+17	2.915257e+17	4.127835	2.915257e+17	2.915257e+17	3.403119e+07
min	3.106630e+05	0.000000	2.308000e+05	2.232740e+05	0.000000	2.385350e+05	2.232080e+05	6.822800e+04
25%	1.481924e+06	0.000000	9.223372e+18	9.223372e+18	0.000000	9.223372e+18	9.223372e+18	3.780000e+07
50%	1.698246e+06	0.000000	9.223372e+18	9.223372e+18	1.000000	9.223372e+18	9.223372e+18	7.153920e+07
75%	1.847908e+06	1.000000	9.223372e+18	9.223372e+18	2.000000	9.223372e+18	9.223372e+18	9.404640e+07
max	1.949800e+06	513.000000	9.223372e+18	9.223372e+18	85.000000	9.223372e+18	9.223372e+18	1.161216e+08

Figure 2: Dataset Summary

Figure 2 dataset summary reveals 4.9M issues, having an average of 1.67M votes per issue (std 1.8M). It is clear that backlog order in this dataset averages 9.21e+18 (standard deviation 2.91e+17), as well as bug master rank with the same average and standard deviation 2.91e+17, but comments amount to an average of 1.67 million (standard deviation 4.17 million). While there are no significant differences between the global and the JIRA support rank averages, the last comment demonstrates a lower average of 6.48e+07 (std 3.40e+07). Among them, 75 percentiles for votes are 1M, comments are 2M, and the last commented is 9.46e+07. It suggests that there is a lot of variation concerning the issues’ engagement and popularity.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

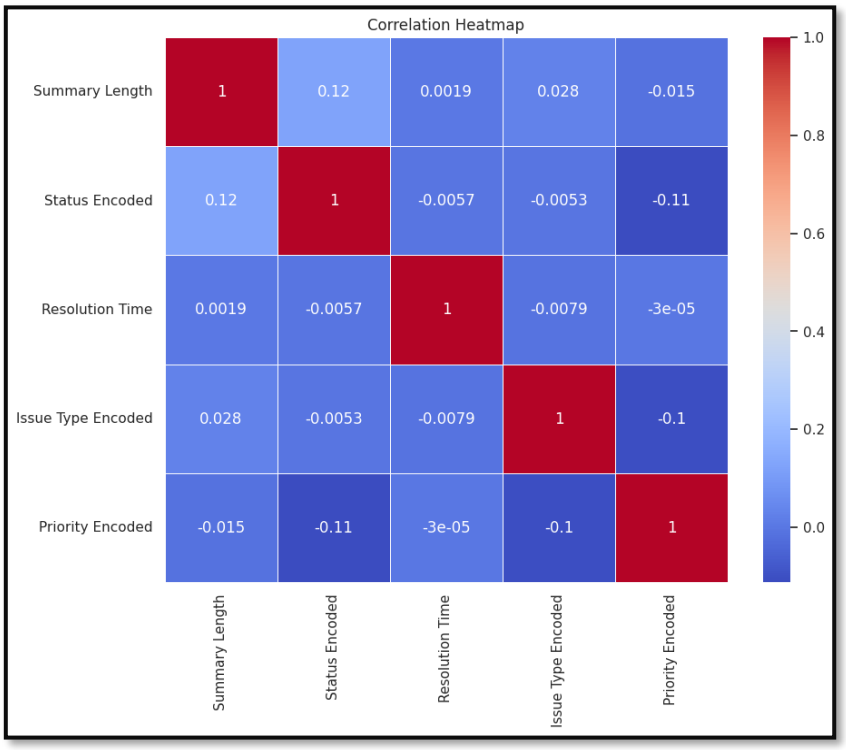

Figure 3: Correlation Heatmap

Figure 3 shows that the heatmap of the correlation suggests that the variables are only weakly associated. The correlation coefficient between summary length and issue type encoded is 0.028, demonstrating a positive relationship, while the correlation coefficient is negative between resolution time and priority encoded. Therefore, when comparing the level of status encoded to the level of priority encoded, the two variables show a very low negative correlation (r=-0.11). Most of the correlations, including resolution time and issue type encoded (-0.0079), are close to '0', showing a weak linear relationship between these encoded issue tracking measures.

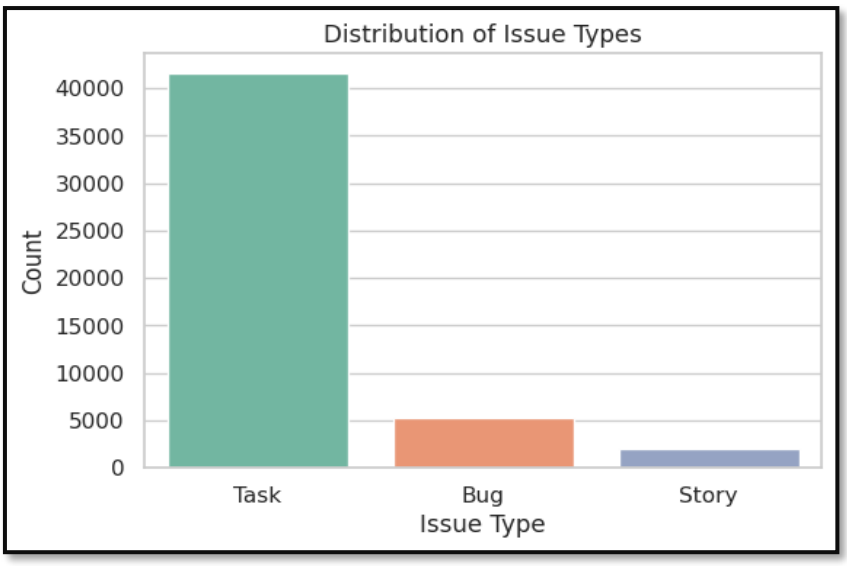

Figure 4: Distribution of Issue Types

Figure 4 displays the distribution of various types of issues that are compared in the given dataset. Tasks account for close to 40000 and represent the largest segment in the distribution, which is way above other categories. Bugs come

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

second with approximately five thousand occurrences and stories, and last with roughly a thousand occurrences. This distribution shows that people have most frequently addressed the issue as a task, so the focus is on the actionable assignments. At the same time, bugs and stories seem less common, showing fewer defects or issues with a narrative coming into the system.

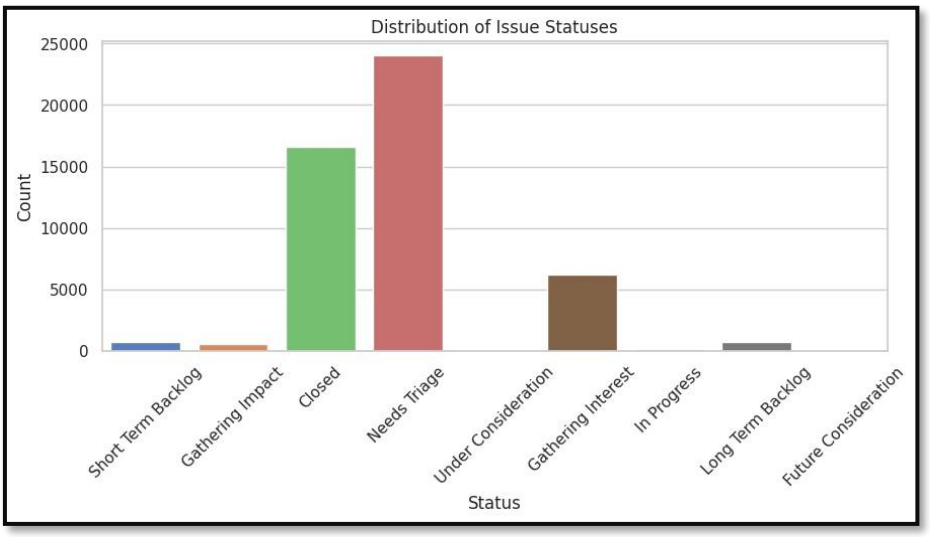

Figure 5: Distribution of Issue Statuses

Figure 5 also shows the distribution status of issues within the dataset. Regarding the numbers, “Needs Triage” occupies the first place with approximately 25,000 involved topics, and "Closed" occupies the second place with roughly 15,000 engagements. Because, for "Under Consideration", there are approximately 5,000 tags and for “In Progress,” “Short Term Backlog,” “Gathering Impact,” “Long Term Backlog,” and “Future consideration,” the number of tags is below 1000. This distribution indicates that most are in the backlog before being evaluated or closed, while few are in progress or the planning horizon.

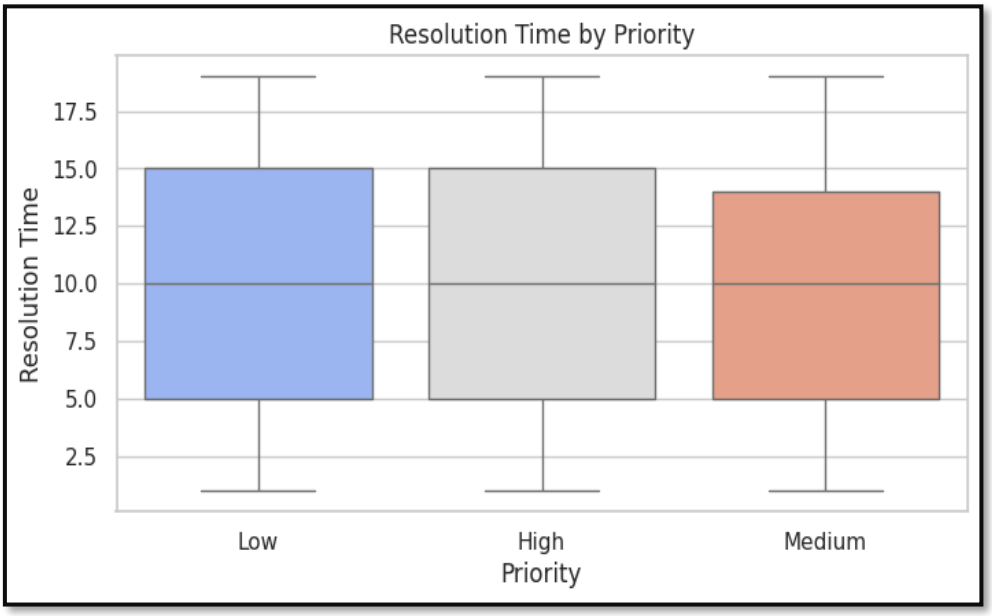

Figure 6: Resolution Time by Priority

Figure 6 displays the resolution time according to priority levels in the dataset. The resolution times for issues at all priority levels match because their median durations are around 10 units. The priority-based resolution time

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

measurements reveal identical ranges extending from 7.5 to 12.5 units throughout the dataset. The resolution times for issues across priority levels show substantial deviation through whiskers extending between 2.5 and 17.5 units while maintaining a generally similar timescale of 2.5 to 17.5 units, reflecting no substantial priority effect on resolution duration.

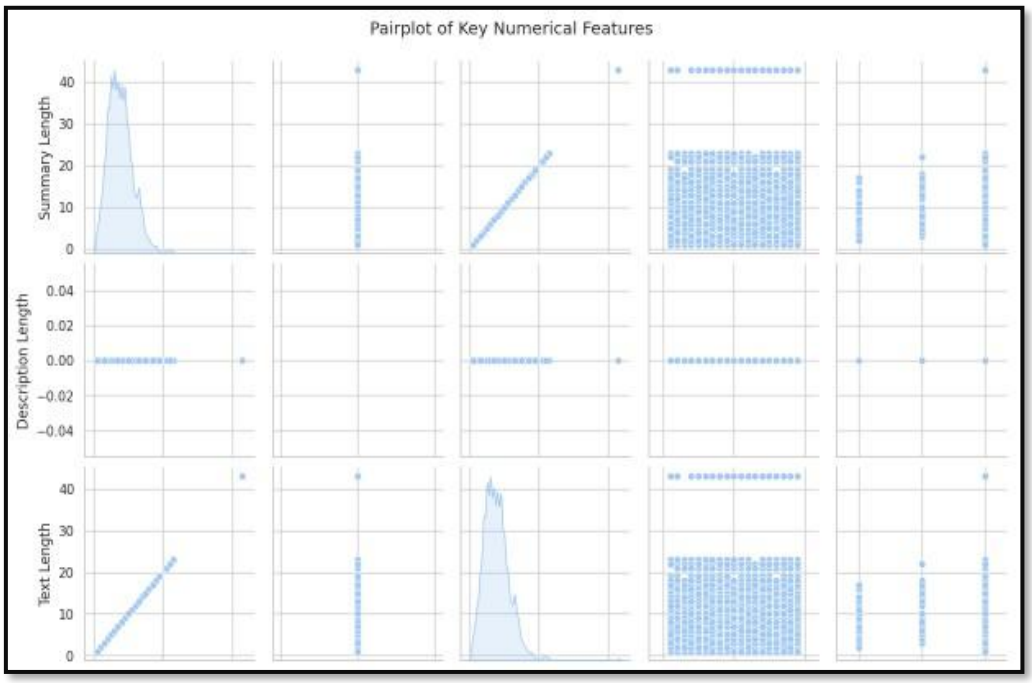

Figure 7: Pair plot of Key Numerical Features

Figure 7 shows a pair plot comparing the summary length against the text length of key numerical features. The summary length distribution reaches its highest point at 40, but the text length follows two different peaks at 0 and 40. The pair plot data points show a slight positive connection between the variables as they mainly gather along the lower regions of both axes. The pair plot shows a weak correlation between the variables while displaying significant text length differences compared to summary length differences.

Logistic Regression Report:

	precision	recall	f1-score	support
Bug	1.00	1.00	1.00	1065
Story	1.00	1.00	1.00	368
Task	1.00	1.00	1.00	8367
accuracy			1.00	9800
macro avg	1.00	1.00	1.00	9800
weighted avg	1.00	1.00	1.00	9800

Figure 8: Logistic Regression Report

Figure 8 displays the logistic regression report that determines issue type classification. The model achieved an annual F1-score of 1.00, recall of 1.00 and precision of 1.00 for each category of Bug (1065 samples), Story (368 samples), and Task (8367 samples). This indicates that the model performed flawlessly across all three categories. The complete accuracy rises to 1.00 with 9800 total samples alongside macro and weighted precision and recall behaviour of 1.00. The model performs perfectly by successfully identifying every issue type without any inconsistencies in the available data.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

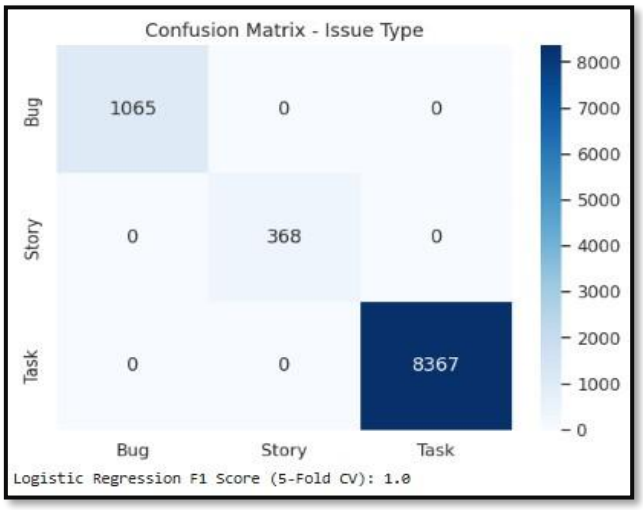

Figure 9: Confusion Matrix - Issue Type and F1 Score (5-Fold CV)

Figure 9, the logistic regression-based issue type classification analysis, shows its results via a confusion matrix. All three issue groups remain successfully categorised by the model as Bugs with 1065 samples and Stories with 368, while Tasks reached 8367 correct predictions according to the matrix data. Consequently, the matrix shows 0 misclassified cases. The analysis of 5-fold cross-validation validates a flawless performance through its F1 score of 1.0. The model performs optimally in differentiating between Bug, Story and Task issue types because it provides complete accuracy in predicting accurate dataset labels.

XGBoost Report:

	precision	recall	f1-score	support
High	1.00	0.03	0.06	798
Low	0.86	1.00	0.93	8403
Medium	1.00	0.03	0.06	599
accuracy			0.86	9800
macro avg	0.95	0.35	0.35	9800
weighted avg	0.88	0.86	0.80	9800

Figure 10: XGBoost Report

Figure 10: XGBoost generates the identification report for issue priority. High priority achieved an accuracy 1.00 while its recall amounted to 0.03 for 798 samples, resulting in an F1-score of 0.06. The precision level for low priority issues reached 0.86, recall amounted to 1.00, and the F1-score achieved 0.93 from 8403 samples. The classification results from the XGBoost algorithm yield a precision score of 1.00, recall score of 0.03, and an F1-score of 0.06 for medium priority with 599 samples. The model demonstrates 0.86 accuracy, while the weighted average F1-score reaches 0.80 but shows lower recall rates for detecting high and medium-priority issues.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

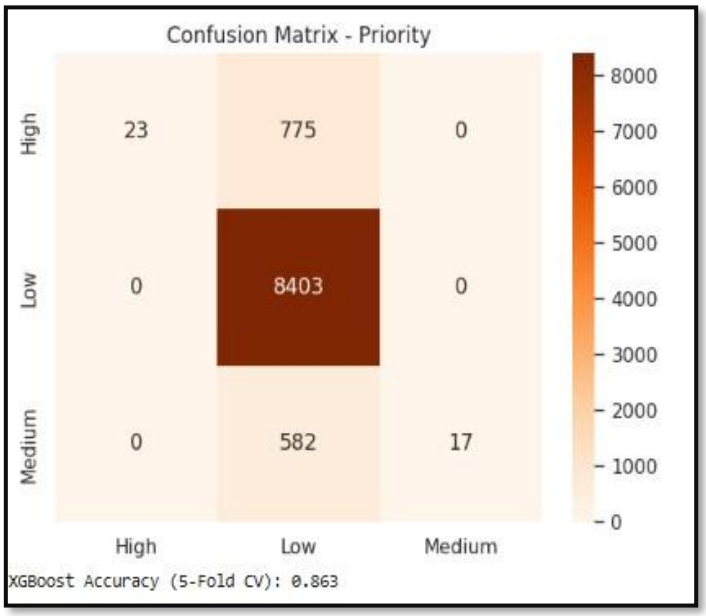

Figure 11: Confusion Matrix – Priority and Accuracy (5-Fold CV)

Figure 11 displays the XGBoost-based priority classification that yields its results through the visual representation. The classification model correctly identifies 8403 cases of low priority yet fails to identify 775 incidents of high and 582 incidents of medium priority, which it mistakes for low-priority matters. There are 17 correct predictions under medium priority alongside 23 correct predictions under high priority. The 5-fold cross-validation method achieved an XGBoost accuracy of 0.863. The accuracy indicates the model performs well for low priority while showing poor performance in identifying high and medium priority issues since it strongly prefers predicting low priority events.

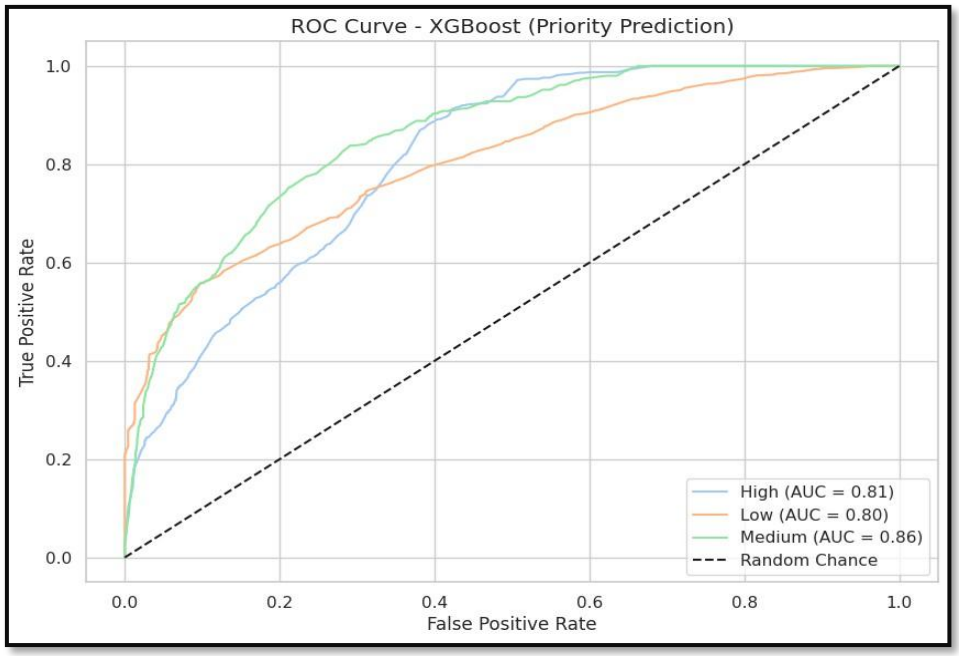

Figure 12: ROC Curve - XGBoost (Priority Prediction)

Figure 12 shows the ROC curve for XGBoost priority prediction. The performance of the model exhibits strong capabilities based on the shape of the ROC curve for high (AUC = 0.81), low (AUC = 0.80), and medium (AUC = 0.86) priorities. The curves demonstrate superior predictive power over the random chance line by demonstrating practical

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

discrimination abilities. According to the XGBoost model, the most predictive category is medium priority since it reaches an AUC of 0.86. Yet, the prediction performance for high and low-priority issues remains relatively strong with lower AUC values of 0.81 and 0.80, respectively.

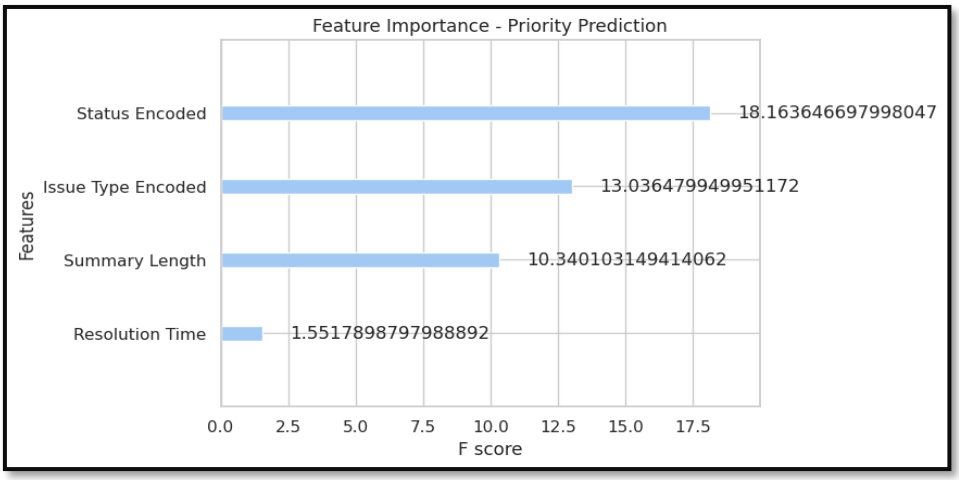

Figure 13: Feature Importance - Priority Prediction

Figure 13 presents the importance of the feature for priority prediction using XGBoost. According to the respective F-score, Status Encoded constitutes the most significant importance at 18.16, while the importance of Issue Type Encoded constitutes 13.03. The results of the method statistics are length scores, which assign 10.34 to the summary and the least 1.55 to the resolution time. This means that while status and issue type are the most critical variables in determining priority, the time to resolve an issue does not strongly influence the model's priority determination of issues.

DISCUSSION

Interpretation of Results

It is evident from the findings that applying the proposed AI-based framework effectively automates Agile processes. The integration of logistic regression with the TF-IDF algorithm produced the best results for the issue type classification: bug, story, and task, as presented in Figures 7 and 8, with an F1 score of 1 for all classes. Hence, high performance is obtained by the simplicity and efficiency of TF-IDF for extracting such domain-specific terms as 'exception' and 'login' from the Jira summaries for correct categorisation. The interpretability of the model ensures that Agile teams can rely on and act on the classification made by the model during the backlog grooming. The amount of work to be done would be drastically reduced. On the other hand, priority prediction by XGBoost was good but with some impairment, with an accuracy of 0.863 and AUC scores varying from 0.80 to 0.86, as depicted in Figures 9–11. It generated 8403 correct predictions for low priority bugs but failed to accurately identify 775 high priority issues and 582 medium priority issues as high and medium, respectively. This implies that, although XGBoost utilises distinctive features such as status and issue type, as presented in Figure 12, textual features may need to be fine-tuned to detect subtler priority signals.

Comparative Review

In previous works dealing with Agile automation, the selection of effectors has been accomplished by applying static priority rules, which are ineffective in the dynamic context of a project. On the other hand, this work proposes an effective use of NLP and ML combined into a single process, and the automation feature allows a customisable approach. Unlike previous studies that have adopted synthetic datasets, this study employs actual data gathered from Jira, thus revealing real-life problems of Agile implementation in enterprises. TF-IDF and Logistic Regression are relatively less complex models to train, implement and use and are much more feasible than neural models like BERT. Still, these latter models possess high accuracy, which is desirable. Since XGBOOST effectively utilises hybrid

features, it is superior to other traditional ML techniques for sprint planning in today's predictive analytics for software engineering.

Practical Implications

Reducing the backlog grooming time is also a distinctive feature of the suggested framework, as it determines the potential issue types and provides the main work to be accomplished by teams. Due to priority predictions, preparing for sprints is done more quickly and helps to be in sync with business objectives more effectively. By applying these tools, the decisions to be taken by agile coaches and product managers are consolidated in different areas. The interpretation is also gained in the Logistic Regression model, which creates trust in the system and, therefore, is feasible to be implemented in, for example, Jira or Azure DevOps, which may bring about Agile transformation at scale.

Limitations

The current framework relies solely on English words and phrases and does not consider that multilingual Agile teams use Jira with non-English entries. At the same time, a lack of easily obtainable information, such as developer effort hours or team capacity, hinders the model's ability to consider human aspects in sprint planning. Such gaps suggest that extended datasets can improve the flexible framework in various Agile environments.

CONCLUSION

This work appropriately illustrated that with the help of the AI-based system, Agile can be implemented on a large scale to enhance several methodologies, like backlog grooming and scattered planning with the help of NLP and ML. Using the real-world Jira dataset obtained from Kaggle, achieving 100% issue type accuracy was possible using TF-IDF and Logistic Regression and high-priority classification using XGBoost with a classification accuracy of 0.863. These results open up the Agile processes with the help of AI and the ability to minimise human-driven manipulations while preserving stability at scale. Combining simple and more interpretable models enables practitioners to adopt them in real-world backlog prioritisation and sprint establishment situations.

The work in this paper can be further taken forward in the following ways, which will take less time in grooming from the backlog area: Using a better advanced NLP model like Chatgpt to develop the summarisation of the backlog items. Reinforcement learning could be implemented to make the sprint adaptation dependent upon different feedback from the team members and the change in dynamics of the project. However, integrating the framework with Jira's live APIS would enable the actual functioning of Agile teams. Including multilingual entries and other attributes like developer effort in terms of hours would further enhance the reliability and versatility of the system, increasing the potential of context-aware and interpretable Agile decision-support systems.

REFERENCES

[1] Anjum, N., & Alam, S. (2019). A Comparative Analysis on Widely used Web Frameworks to Choose the Requirement based Development Technology. *International Advanced Research Journal in Science, Engineering and Technology*, *6*(9). Doi: 10.17148/IARJSET.2019.6902

[2] Anjum, N., & Chowdhury, R. (2024). Revolutionising Cybersecurity Audit through Artificial Intelligence Automation: A Comprehensive Exploration. *International Journal of Advanced Research in Computer and Communication Engineering*, *13*(5), 493-502.

[3] Anjum, N., & Kabir, A. (2019). Introducing Refined Agile Model (RAM) in Bangladesh's Software Development Environment concentrating on the Improvement of Requirement Engineering Process. *International Journal of Software Engineering & Applications (IJSEA)*, *10*(4). Doi: https://dx.doi.org/10.2139/ssrn.3434858

[4] Annadata, L. A. (2023). A Data-Driven Approach for Incident Handling in DevOps.

[5] Brankovic, A. (2025). UAE Tolerance Framework as a Base for Coexistence in a Multicultural Society. *International Journal Of Civilisations Studies & Tolerance Sciences*, *1*, 1-115.

[6] De Diego, I. M., Redondo, A. R., Fernández, R. R., Navarro, J., & Moguerza, J. M. (2022). General performance score for classification problems. *Applied Intelligence*, *52*(10), 12049-12063.

Research Article

[7] Deguine, J. P., Aubertot, J. N., Bellon, S., Côte, F., Lauri, P. E., Lescourret, F., ... & Lamichhane, J. R. (2023). Agroecological crop protection for sustainable agriculture. *Advances in agronomy*, *178*, 1-59. Doi: https://doi.org/10.1016/bs.agron.2022.11.002

[8] Dixit, S., & Jangid, J. (2024). *Exploring Smart Contracts and Artificial Intelligence in FinTech*. Doi: https://doi.org/10.52783/jisem.v10i14s.2208

[9] Hood, K., & Al-Oun, M. (2014). Changing performance traditions and Bedouin identity in the North Badiya, Jordan. *Nomadic Peoples*, *18*(2), 78-99.

[10] Ibadin, F. E., Edobor, T. E., Ernest-Okonofua, E. O., & Oyiborhoro, O. G. (2025). Comparative Analysis of Granulation Tissue Formation and Progression in Elderly Patients with Fractures. *American Journal of Medical Science and Innovation*, *4*(1), 11-17. Doi: https://doi.org/10.54536/ajmsi.v4i1.3945

[11] Jangid, J. (2020). Efficient Training Data Caching for Deep Learning in Edge Computing Networks. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN*, 2456-3307. Doi: http://dx.doi.org/10.32628/CSEIT20631113

[12] Kamath, D. (2023). Improving Agile Development Practices.

[13] Karani, V., & Tyagi, J. (2021). *U.S. Patent No. 11,170,381*. Washington, DC: U.S. Patent and Trademark Office.

[14] Karani, V., Madisetti, P., & Tyagi, J. (2019). *U.S. Patent Application No. 15/823,079*.

[15] Miao, J., & Zhu, W. (2022). Precision–recall curve (PRC) classification trees. *Evolutionary intelligence*, *15*(3), 1545-1569.

[16] Omri, M. N., & Mribah, W. (2022). Towards an intelligent machine learning-based business approach. *International Journal of Intelligent Systems and Applications*, *13*(1), 1. Doi: 10.5815/ijisa.2022.01.01

[17] Raza, S. A. (2020). Predicting the Duration of User Stories, Machine learning for agile planning.

[18] Reddy, B. R. B., & Sushma, M. (2025). Revolutionizing oncology – role of artificial intelligence in early cancer detection and diagnostic advances-A comprehensive review. *Research and Reviews: Journal of Oncology and Hematology*, *14*(01).

[19] Sankhe, P., & Dixit, M. (2024). Navigating Software Development Methodologies: A Comparative Analysis and Recommendation of Software Process Models. Doi: https://doi.org/10.62441/nano-ntp.vi.3424

[20] Singh, M. (2024). Agile project management and testing in distributed development environment (Doctoral dissertation, JC Bose University).

[21] Sriwastwa, A., Ravi, P., Emmert, A., Chokshi, S., Kondor, S., Dhal, K., ... & Gupta, R. (2023). Generative AI for medical 3D printing: a comparison of ChatGPT outputs to reference standard education. *3D Printing in Medicine*, *9*(1), 21. Doi: https://doi.org/10.1186/s41205-023-00186-8

[22] Truss, A. (2024). Agile development processes in IT support work.

[23] Viraj Soni. (2024). Fraud Detection in Credit Card Transactions: A Machine Learning Approach. *Journal of Electrical Systems*, *20*(11s), 3938–3954. https://doi.org/10.52783/jes.8318

[24] Wiesche, M. (2021). Interruptions in agile software development teams. *Project Management Journal*, *52*(2), 210-222. Doi: https://doi.org/10.1177/8756972821991365