Integrating Gradient Boosting and Artificial Intelligence Agent-Based Workflow Automation for Employee Attrition Prediction and Human Resource Meeting Scheduling

Vedant Keshavrao Deshmukh[1], Ajinkya Manoj Deshmukh[2], Kartik Nilesh Raghuwanshi[3], Anurag Sandeep Shrikhandkar[4], Dr Kalyani Hemant Deshmukh[5]

[1–4]B. Tech Final-Year Students, Department of Computer Science and Engineering
[5]Department of Computer Science and Engineering
[1–5]Prof. Ram Meghe Institute of Technology and Research, Badnera-Amravati, Maharashtra, India
[1]vedantdeshmukh975@gmail.com, [2]ajinkya71005@gmail.com, [3]raghuwanshikartik521@gmail.com,
[4]anuragshrikhandkar@gmail.com, [5]khdeshmukh@mitra.ac.in

***Abstract*—Models that predict employee attrition can point out people who are more likely to leave, yet a prediction by itself triggers no organisational response. This paper presents a prototype that connects a gradient boosting classifier, trained on the public IBM HR Analytics Employee Attrition and Performance dataset, to an n8n workflow that arranges human resource (HR) follow-up meetings through Google Calendar. A written decision rule maps the model score to four risk tiers, a seven-day cooldown stops meetings from being repeated, and an HR dashboard allows reviewers to confirm or snooze every recommendation before anything leaves the system. On a held-out test set of 294 employees, gradient boosting obtained an accuracy of 0.898, a precision of 0.743, a recall of 0.553, an F1-score of 0.634, an area under the receiver operating characteristic curve (ROC-AUC) of 0.864 and an area under the precision-recall curve (PR-AUC) of 0.682, whereas always predicting the majority class gives an accuracy of 0.840. In a 20-run test bench using synthetic records, the workflow created calendar events with Google Meet links, blocked all five duplicate triggers, and handled two edge cases, at a mean latency of 1.42 seconds. The test bench is small, the study does not measure any effect on attrition, and it depends on a single fictional dataset. Every prediction is therefore treated as a risk indicator and not as evidence that someone intends to resign.**

***Keywords*—Employee attrition prediction, gradient boosting, human resource analytics, workflow automation, n8n, meeting scheduling.**

I. INTRODUCTION

Employee attrition refers to employees leaving an organisation. The departure may be voluntary, when the employee decides to go, or involuntary, when the employer terminates the contract [1], [2]. Replacing a departed employee costs both time and accumulated organisational knowledge, so HR teams have good reason to identify, before the exit happens, those employees who face an elevated risk of leaving. Machine learning (ML) has been applied to this task using various data sources, and the public IBM HR Analytics Employee Attrition and Performance dataset has become a widely used benchmark [1]–[4].

Gradient boosting constructs a strong predictor by adding shallow decision trees sequentially, where each new tree is fitted to the residual errors of the ensemble built so far [5]. After encoding, it copes well with a mixture of numerical and categorical attributes, and it is a common baseline for tabular data. For instance, it served as one of the conventional baselines in a study that compared a Transformer -based network with other models on this dataset [2].

A system that only predicts has a limitation that is organisational rather than statistical. A risk score sitting in a table does not decide whom to contact, does not arrange a conversation, and does not record the outcome. Studies on this dataset usually report classification performance [1]–[3], and the step that turns a score into an HR action tends to remain outside the experiment. Workflow automation platforms such as n8n can link a model output to calendar services, which allows this step to be described and tested explicitly.

A prediction is a risk indicator derived from historical patterns; it is not proof that an employee plans to resign. Approaching an employee because of a score raises ethical and privacy concerns. For this reason, the workflow examined here includes an HR review stage in which each flagged recommendation can be confirmed or snoozed before any external action takes place.

Research problem. How can an attrition classifier be linked to a structured meeting-scheduling workflow that includes human review, and how can the prediction component and the scheduling component each be evaluated using measurable criteria?

Contributions. This paper (i) specifies an end-to-end pipeline that runs from a gradient boosting classifier, through a documented decision rule and an n8n workflow, to Google Calendar events; (ii) proposes an evaluation protocol that treats prediction quality and workflow execution separately and defines the denominator of every metric; (iii) sets out the privacy, fairness and human-oversight requirements of the design; and (iv) distinguishes what the implementation demonstrates from what is only proposed. The paper does not claim that gradient boosting, the dataset, n8n or Google Calendar is novel in itself, and it does not claim that the workflow lowers attrition.

The rest of the paper follows the IMRaD structure. Section II reviews related work. Section III describes the methodology, covering the problem statement, the proposed system, the dataset, the classifier, the decision rule, the n8n and Google Calendar integration, and the implementation. Section IV presents the results, Section V discusses them, and Section VI sets out the limitations, future work and conclusions.

II. LITERATURE REVIEW

A. Attrition Prediction on the IBM Dataset

The distributors of the dataset describe it as fictional data created by IBM data scientists [4]. One study compared four classifiers on it and found that a tuned Extra Trees classifier reached 93% accuracy, with monthly income, hourly rate, job level and age identified as the most influential factors [1]. That study describes oversampling as taking place before the 85:15 train-test split. Such an ordering allows synthetic samples derived from records similar to test data to enter the training set, which can inflate the held-out scores. The protocol proposed here therefore applies any resampling only within training folds. Another study compared a Transformer-based network for tabular data with gradient boosting, XGBoost, random forest and extremely randomised trees, using a stratified 85:15 split of 1,470 records of which 237 (16.1%) were leavers; the Transformer obtained higher ROC-AUC and PR-AUC values [2]. A third study applied machine learning, ensemble and deep learning methods to the same data and found that a feedforward network gave the best accuracy, recall and F1-score [3].

The headline figures from these studies cannot be compared directly, because the splits, resampling choices, tuning procedures and metrics all differ. They do indicate that the best model depends on the protocol. For this reason, the present paper reports gradient boosting under a single, fully documented protocol and does not rank it against published numbers.

B. Gradient Boosting

Friedman described boosting as stagewise gradient descent in function space and developed gradient boosting of regression trees for both regression and classification [5]. Mature implementations exist in popular libraries such as scikit-learn [6]. In the Transformer comparison mentioned above, gradient boosting obtained a ROC-AUC of 0.6929 and a PR-AUC of 0.3590 on one test split of 221 records that contained 36 leavers [2]. These values belong to that particular split and should not be treated as a benchmark for the present work.

C. Class Imbalance, Explainability and Fairness

When the positive class is a minority, as with the roughly one-in-six leavers noted above [2], precision-recall analysis can be more informative than ROC analysis alone [7]. Feature attribution methods such as SHapley Additive exPlanations (SHAP) can explain individual predictions [8], and surveys of bias in ML describe how choices about data and modelling can lead to unfair outcomes for particular groups [9]. These concerns carry more weight when a score leads to contact with a real person.

D. Workflow Automation and HR Decision Support

The Google Calendar node of n8n supports checking availability as well as creating, retrieving, updating and deleting events [10]. The Google Calendar application programming interface (API) can create an event and, if the conference data version parameter is set, can also request a Google Meet conference for that event [11]. Practitioner material that has not been peer reviewed shows that these elements are already being combined with attrition analysis. One community n8n template scores attrition risk with a language model and e-mails HR [12], and a public repository uses n8n to generate HR reports from an attrition analysis [13]. In the material we reviewed, neither source reports an evaluation of scheduling reliability.

E. Comparison and Research Gap

Table I summarises the sources discussed above. It rests on a limited, non-systematic web and database search carried out in October 2026. It therefore describes the sources we examined and does not show that no other work exists.

TABLE I. COMPARISON OF SELECTED RELATED WORK

Ref.	Data / setting	Method	Main finding	Limitation	Relevance here
[1]	IBM HR data: 1,470 records, 35 features	SVM, LR, DTC, tuned Extra Trees, SMOTE, 10-fold CV	Extra Trees reported 93% accuracy; income, hourly rate, job level and age were highlighted	Resampling described before the split; focus on accuracy; prediction only	Shows common protocol pitfalls to avoid
[2]	IBM HR data: stratified 85:15 split	Transformer-based network vs. GB, XGB, RF and ERT	Transformer obtained higher ROC-AUC and PR-AUC than the conventional baselines	Small test set (221 records, 36 leavers); no workflow	Reports GB as a baseline; supports the use of PR-AUC
[3]	IBM Watson HR dataset	ML, ensemble and deep learning	Feedforward network reported the best accuracy, recall and F1	Prediction only; protocol not comparable across studies	Context for model choice
[12]	Template: HR files	LLM risk scoring, threshold routing, e-mail alert	Automated alert with recommended actions	Not peer reviewed; no meeting scheduling; no reported evaluation	Practitioner precedent for n8n alerting
[13]	Employee CSV	Pandas analysis, n8n report generation	Automated HR report	Not peer reviewed; report only; no scheduling	Practitioner precedent for n8n in HR analytics

SVM: support vector machine; LR: logistic regression; DTC: decision tree classifier; SMOTE: synthetic minority oversampling technique; CV: cross-validation; GB: gradient boosting; XGB: XGBoost; RF: random forest; ERT: extremely randomised trees; CSV: comma-separated values; LLM: large language model.

Gap. Among the sources examined, peer-reviewed attrition studies report prediction performance, while practitioner n8n material reports alerts or reports. We found no source that uses defined metrics to evaluate the path from a classifier output, through human review, to a calendar event. This paper addresses that narrower gap. It does not claim to be the first work on attrition prediction or on HR workflow automation.

III. METHODOLOGY

A. Problem Statement, Objectives and Scope

1) Problem Statement

Consider an employee record x and a trained classifier that returns a score p(x) for the positive class. The task is to design and evaluate a workflow that decides whether p(x) justifies an HR follow-up recommendation, sends that recommendation through human review, and then creates a calendar meeting, while measuring prediction quality and scheduling reliability separately.

2) Objectives

The main objective is to evaluate an integrated pipeline, with human review, that runs from attrition prediction to meeting scheduling. The specific objectives are to (1) train and evaluate a gradient boosting classifier on the IBM dataset; (2) document the preprocessing, training, prediction and evaluation steps; (3) define the rule that converts a prediction into a workflow action; (4) design the integration between n8n and Google Calendar; (5) define measurable workflow metrics; and (6) discuss privacy, limitations, reproducibility and generalisability.

3) Research Questions

RQ1. How well does gradient boosting classify attrition on the chosen dataset under a documented split? RQ2. How does the workflow turn a reviewed, eligible prediction into a scheduling action? RQ3. Which operational failures, delays and duplicate-event risks occur during scheduling? RQ4. Which limitations affect the validity and practical use of the system?

4) Scope and Assumptions

The scope is a prototype evaluated on one fictional public dataset, using test employees or synthetic calendar entries; real employee data are outside the scope. The study assumes that HR reviewers exist and are able to confirm or snooze each recommendation, that the employee identifier in the dataset can be mapped to a test calendar identity, and that a score is a risk indicator and not a calibrated probability unless calibration is evaluated.

B. Proposed System

The system comprises ten functional components, shown in Fig. 1: (1) dataset input; (2) preprocessing; (3) the gradient boosting model; (4) the prediction output; (5) a decision rule with risk tiers and a cooldown; (6) an HR review stage; (7) an n8n workflow; (8) Google Calendar event creation and invitations; (9) Google Meet link generation; and (10) logging, error handling and retries. Components 1 to 5 are supported by the model experiment in Sec. IV-A. Components 7 to 9 are supported by the n8n workflow canvas (Fig. 3), the received Google Calendar invitation with a Meet link (Fig. 6) and the test bench in Sec. IV-B. The cooldown, the HR review stage (component 6) and the retry logic (component 10) are implemented in the application layer, namely the back end and a Python alert agent, and not as nodes of the n8n workflow. Input schema validation is a proposed extension.

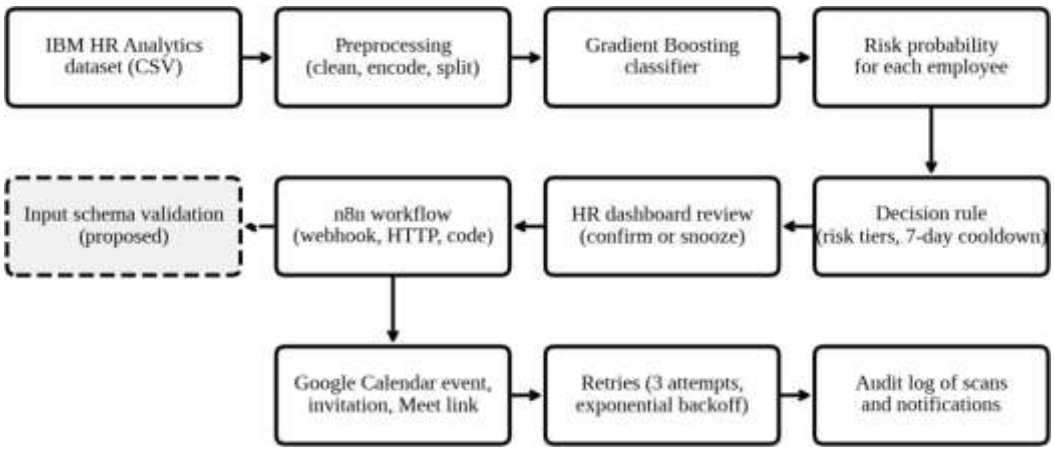


Fig. 1. System architecture

C. Dataset Description

The IBM HR Analytics Employee Attrition and Performance dataset is publicly available on Kaggle and is presented as fictional data created by IBM data scientists [4]. Other studies describe it as containing 1,470 records with 35 attributes, of which 237 (16.1%) carry the label "Attrition = Yes" [1], [2]. The authors' own run is consistent with this: 1,233 records (83.9%) are labelled "No" and 237 (16.1%) are labelled "Yes" (Fig. 2). The target is the binary attribute "Attrition", where "Yes" is the positive class (the employee left) and "No" is the negative class (the employee stayed). Since the data are fictional, the results describe behaviour on this benchmark and do not establish performance in a real organisation.

TABLE II. DATASET AND SPLIT SUMMARY

Item	Value	Source of value
Records	1,470	Class counts in Fig. 2
Attributes	35 / 30	35 raw attributes as reported in [1]; 30 model inputs after removing four columns and the target
Positive class (attrition = yes), count and share	237 (16.1%)	Authors' class distribution, Fig. 2
Training / test records	1,176 / 294	Stratified 80/20 split on the target
Positive records in training / test	190 / 47	The stratified split preserves the 16.1% share
Random seed and stratification	42; stratified on attrition	Training pipeline configuration

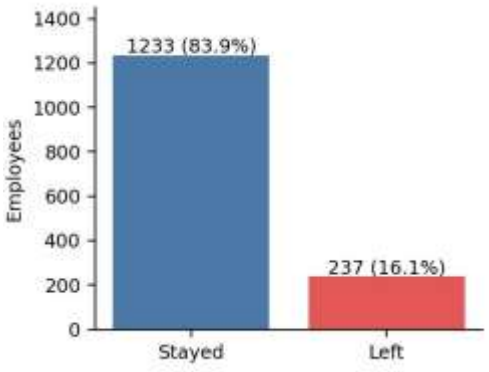


Fig. 2. Target-class distribution of the dataset

D. Data Preprocessing

Four columns are dropped before modelling: EmployeeCount (constant 1), StandardHours (constant 80), Over18 (constant "Y") and EmployeeNumber (an identifier). The remaining 30 input features consist of 23 numerical attributes, which are median-imputed and standardised to zero mean and unit variance, and seven categorical attributes (BusinessTravel, Department, EducationField, Gender, JobRole, MaritalStatus and OverTime), which are mode-imputed and one-hot encoded, with unseen categories ignored. All transformations are combined in a single scikit-learn pipeline that is applied identically to training and test records. The split is stratified on the target and uses a fixed random seed of 42. No resampling is performed, so no synthetic records can enter the test set.

E. Gradient Boosting Classifier

Gradient boosting builds an additive model $F_M(x)$ out of weak learners [5]. The process begins with a constant initial model (1). In every iteration, the negative gradient of the loss with respect to the current predictions is computed (2), a regression tree h_m is fitted to these pseudo-residuals, and the ensemble is

updated using a learning rate ν (3). For binary classification, the loss is the logistic loss, and the logistic function converts the model output into a score (Eqs. (4), (5)).

$$F_0(x) = argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma) \quad (1)$$

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F=F_{m-1}} \quad (2)$$

$$F_m(x) = F_{m-1}(x) + \nu\gamma_m h_m(x) \quad (3)$$

$$L(y, F) = -[y\ log\ \sigma(F) + (1-y)\ log(1-\sigma(F))] \quad (4)$$

$$p(x) = \sigma(F_M(x)) = \frac{1}{1+e^{-F_M(x)}} \quad (5)$$

Here, N is the number of training records, $y_i \in \{0, 1\}$ is the label, M is the number of trees and σ is the logistic function. Table III lists the library and the main hyperparameters of the evaluated model. All other settings keep the library defaults, including the log-loss objective, the Friedman mean-squared-error split criterion, a minimum of two samples to split a node, one sample per leaf and a tolerance of 0.0001.

TABLE III. MODEL CONFIGURATION

Item	Value
Library and version	scikit-learn 1.6.1 (Python 3.10)
Number of trees (M)	200
Learning rate (ν)	0.05
Maximum tree depth	3
Subsample fraction	1.0 (all training records)
Tuning method and search space	Grid search with 5-fold cross-validation over trees {100, 200}, learning rate {0.01, 0.05, 0.1} and depth {3, 5}
Class-imbalance handling	None beyond the stratified split; no resampling or class weights
Random seed	42

F. Model Evaluation

Accuracy alone can be misleading in attrition prediction. When roughly one record in six is positive [2], a model that answers "no" for everyone is right most of the time yet detects no leavers. The evaluation therefore reports precision, recall, F1-score, the confusion matrix, ROC-AUC and PR-AUC [7], all computed on the held-out test set only. Three classifiers were compared: logistic regression, a random forest with 100 trees, and gradient boosting. All three used the same 80/20 split, the same random seed of 42 and the same preprocessing pipeline. Precision, recall and F1-score for the positive class are defined in Eqs. (6) to (8), where TP, TN, FP and FN denote the numbers of true positives, true negatives, false positives and false negatives, respectively.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1 = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (8)$$

G. Prediction-to-Workflow Decision Rule

The workflow acts on a score and not only on a class label. For an employee i with score p_i and eligibility indicator e_i (for example, an active employee with no open follow-up), a recommendation a_i is raised according to Eq. (9), where τ is the decision threshold.

$$a_i = 1\ if\ p_i \geq \tau\ and\ e_i = 1;\ otherwise\ a_i = 0 \quad (9)$$

The threshold and the eligibility conditions are fixed as follows. A score below 0.40 indicates low risk (standard review cycle). A score from 0.40 up to, but not including, 0.65 indicates medium risk (department watchlist). A score from 0.65 up to, but not including, 0.80 indicates high risk (flag on the HR dashboard and a notification to the manager). A score of 0.80 or higher indicates critical risk and starts the scheduling workflow, so τ = 0.80 in Eq. (9). An employee is eligible only if no meeting was created during the previous seven days, unless the risk level has risen from high to critical. For the classification metrics in Sec. IV, a score of 0.50 or higher counts as a positive prediction. A boosted-tree score is a ranking signal and should not be described as a calibrated probability unless calibration has been evaluated. A recommendation (a_i = 1) is passed to an HR reviewer, who decides whether any contact is appropriate. The prediction does not show that the employee intends to leave.

H. n8n Workflow Integration

In the implemented workflow, a request reaches n8n through a webhook node that accepts Hypertext Transfer Protocol (HTTP) POST requests (Fig. 3). The nodes, in order, are a webhook, an HTTP request node (POST to a Google API endpoint), a code node (JavaScript), a second HTTP request node (POST to a Google API endpoint) and a respond-to-webhook node that returns the result to the caller. Every node shows a successful execution in the screenshot, and the webhook node holds pinned test data. The payload contains the employee identifier, name, e-mail address, department, manager e-mail address, the attrition probability, the risk level (low, medium, high or critical), a list of top reasons, a suggested action and an International Organisation for Standardisation (ISO) 8601 timestamp. The canvas contains no condition node or error branch, because the tier, cooldown and review logic run in the application before the webhook is called. The webhook request is sent by the application back end, which dispatches it asynchronously when an employee submits feedback, and by a Python alert agent that also performs a scheduled nightly scan. Each dispatch makes up to three attempts with exponential backoff. Figure 4 shows the sequence of messages.

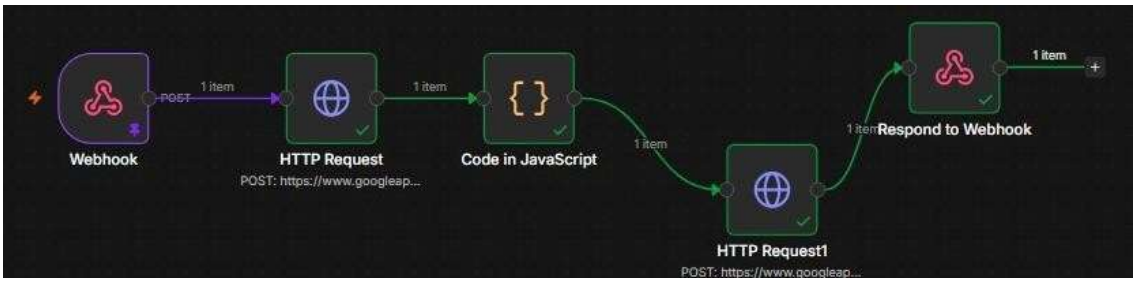


Fig. 3. Implemented n8n workflow canvas

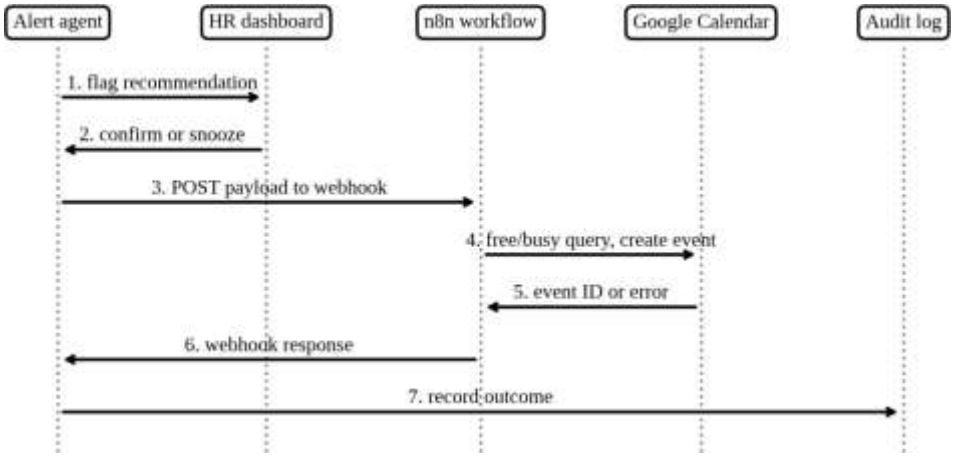


Fig. 4. Message sequence from a flagged recommendation to a calendar event and an audit record

I. Google Calendar and Meeting Scheduling

The scheduling stage calls Google services directly from two n8n HTTP Request nodes instead of using the built-in n8n Google Calendar node, which offers an availability check and operations to create, get, update and delete events [10]. The first node posts to the calendar free/busy endpoint to obtain the availability of HR and the manager. The Code node then picks the earliest free 30-minute window between 10:00 and 17:00 on the next business day and formats the times as Request for Comments (RFC) 3339 timestamps. The second node posts to the events endpoint of the primary calendar with the conference data version set to 1 and requests a Google Meet conference of type hangoutsMeet [11]. The invitation that was received (Fig. 6) shows an event titled “Career & Wellbeing Check-in”, starting at 10:00 India Standard Time on 12 October 2026, with an organiser, guests and a Google Meet link. The title of the invitation is neutral and shows no risk score. However, the event description built by the Code node includes the top reasons from the payload, which Sec. III-J treats as a privacy concern. Duplicate prevention, retries and audit logging are implemented in the application layer (Sec. III-H).

J. Privacy, Security and Responsible Use

Employee-level risk scores are sensitive. The design therefore uses role-based access control with signed tokens, so that predictions are visible only to authorised HR users and direct managers and are hidden from employees. It also uses Open Authorisation (OAuth) 2.0 for calendar access, an audit log that records scans, model inferences, webhook triggers and notifications, and an HR review stage in which flagged recommendations can be confirmed or snoozed before any external action. Two issues need attention before any real deployment. First, the webhook payload contains more than the scheduling step requires (name, e-mail address, department, manager e-mail address, probability and top reasons), and the event description includes the top reasons, so invitees could see the risk drivers. Both the payload and the description should be minimised. Second, no retention limit for logs has been defined. Models can also be biased. False positives expose employees to unwarranted attention, while false negatives leave at-risk employees unnoticed [9]. A fairness assessment across groups is left for future work. No compliance with any regulation or security standard is claimed, because none has been assessed.

Ethics statement. All experiments used the publicly available, anonymised IBM HR Analytics Employee Attrition benchmark dataset. All company names, employee titles and e-mail addresses that appear in application interfaces, webhook payloads and calendar invitations are synthetic placeholders created for experimental validation and are not real. No real personally identifiable data were collected, processed or disclosed.

K. Implementation Details

Table IV lists the components together with the status that the available evidence supports.

TABLE IV. IMPLEMENTATION COMPONENTS AND STATUS

Component	Technology	Purpose	Implementation status
Data and preprocessing	Python 3.10, pandas 2.2.3, NumPy 2.2.3, SciPy 1.15.3, scikit-learn 1.6.1	Load, clean, encode, split	Implemented as one pipeline (Sec. III-D)
Classifiers	Gradient boosting, with logistic regression and random forest as baselines; scikit-learn 1.6.1; FastAPI and Uvicorn service; joblib artefact	Predict attrition risk	Implemented; metrics in Table V
Decision rule	Risk tiers and threshold rule (Eq. (9)); seven-day cooldown	Threshold and eligibility	Implemented; $\tau = 0.80$ for scheduling
Prediction-to-n8n interface	HTTP POST to an n8n webhook node	Deliver eligible predictions	Webhook implemented (Fig. 3); sent by the application back end and the Python alert agent
Workflow engine	n8n (5 nodes including a Code node)	Orchestrate scheduling	Implemented; evaluated in Sec. IV-B
Calendar and invitations	Google API through HTTP Request nodes [11]	Create events, invite attendees	Implemented; invitation received (Fig. 6)
Google Meet link	Calendar API conference data [11]	Provide a video link	Implemented; link present in the invitation
HR review stage	HR dashboard (confirm or snooze); signed-token role checks	Human review before contact	Implemented in the application, not as an n8n node
Duplicate checks, retries, logging	Seven-day cooldown; three attempts with exponential backoff; audit log table	Reliability and audit	Implemented in the application; tested in Sec. IV-B
Deployment environment	n8n instance (hosting not specified)	Runtime	Not documented in this version

IV. RESULTS

The ML results come from the authors' experiment on a held-out test set. The workflow results come from a 20-run test bench that used synthetic records and test calendars. They therefore describe behaviour on a small sample and are not estimates of reliability.

A. Machine Learning Results

Three classifiers were trained and evaluated on a single held-out test set containing 294 records, 47 of which are positive (Table II). Table V gives the metrics at the standard threshold of 0.50, Fig. 5 compares them, and Table VI shows the confusion matrix of gradient boosting.

TABLE V. HELD-OUT TEST-SET METRICS (N = 294, 47 POSITIVES)

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Gradient boosting	0.898	0.743	0.553	0.634	0.864
Logistic regression	0.871	0.636	0.447	0.525	0.812
Random forest	0.881	0.688	0.468	0.557	0.829
Majority class (always no)	0.840	—	0.000	0.000	—

PR-AUC is more informative than ROC-AUC when the positive class is a minority [7]. It was 0.682 for gradient boosting, 0.584 for logistic regression and 0.621 for random forest, compared with a positive-class prevalence of 0.160. Table VI shows the confusion matrix of gradient boosting at the 0.50 threshold.

TABLE VI. CONFUSION MATRIX OF GRADIENT BOOSTING ON THE TEST SET AT THRESHOLD 0.50

	Predicted: No	Predicted: Yes
Actual: No	TN = 238	FP = 9
Actual: Yes	FN = 21	TP = 26

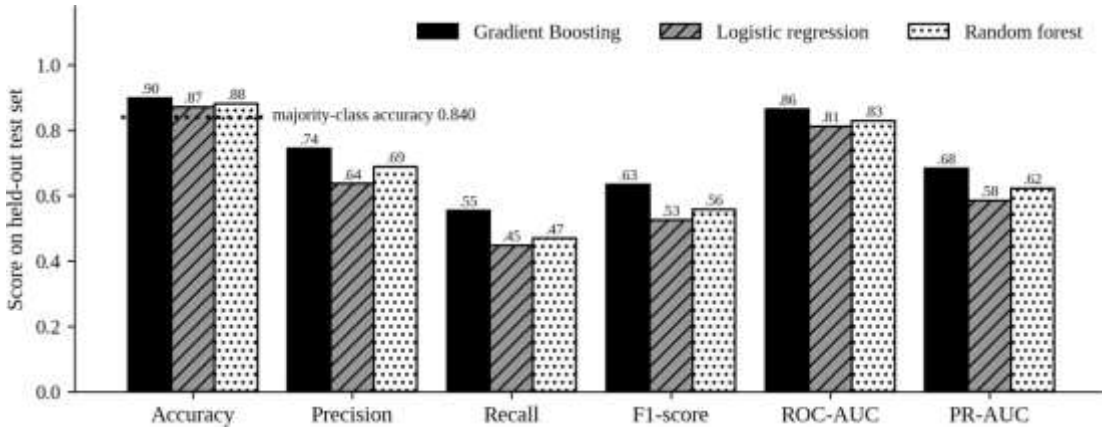


Fig. 5. Test-set metrics of the three classifiers

Gradient boosting obtained the highest score on every reported metric, with a ROC-AUC of 0.864 against 0.829 for random forest and 0.812 for logistic regression. These differences come from a single split with 47 positive records, and neither a significance test nor a cross-validated confidence interval was computed. They therefore do not establish that gradient boosting is better in general. Its accuracy of 0.898 is 5.8 percentage points above the majority-class baseline of 0.840. A recall of 0.553 means that 26 of the 47 leavers were identified and 21 were missed. In total, 35 employees (11.9% of the test set) were flagged, and 26 of them (74.3%) were genuine leavers. In an HR screening setting, the missed leavers are the main error. Lowering the 0.50 threshold would trade precision for recall, but this was not evaluated. A cross-validated comparison with confidence intervals remains future work.

B. Workflow Results

The scheduling pipeline is evaluated independently of the classifier. The test bench consisted of 20 runs using synthetic employee records and test calendars. Eighteen runs were normal dispatches that completed successfully, and two were edge cases that the system handled (an invalid address and a busy slot). Separately, five duplicate triggers were tested, together with three busy-slot cases in total. All five n8n nodes completed in the documented run (Fig. 3), and the invitation with a Google Meet link was received (Fig. 6). Table VII defines each metric with its denominator and reports the observed values. Equations (10) and (11) give the two rates that use the number of test cases N_{test} and the number of created events $N_{created}$.

$$Success\ rate = \frac{N_{success}}{N_{test}} \quad (10)$$

$$Duplicate - event\ rate = \frac{N_{dup}}{N_{created}} \quad (11)$$

TABLE VII. WORKFLOW METRICS, DEFINITIONS, DENOMINATORS AND OBSERVED VALUES

Metric	Definition	Denominator	Value
Test cases run	Webhook dispatches submitted to the workflow	—	20 (synthetic records)
Event-creation success	Cases that ended with a calendar event with the expected attendees and time	All test cases	18 normal dispatches created an event; 2 edge cases were handled
Meet-link success	Created events that contain a conference link	Created events	Present in all created events (100%)
Scheduling latency	Time from workflow trigger to event creation	Successful cases	1.42 s (mean)
Duplicate-event rate	Events created more than once for the same recommendation	Created events	0 duplicates; 5 of 5 duplicate triggers blocked by the cooldown
Failure and recovery	Injected failures (unavailable slot, API error) and the observed handling	Injected failures	3 of 3 busy slots rolled to the next free slot; invalid address handled gracefully
API or integration errors	Count and type of error responses	All API calls	2 handled edge cases in 20 runs (invalid address, busy slot)

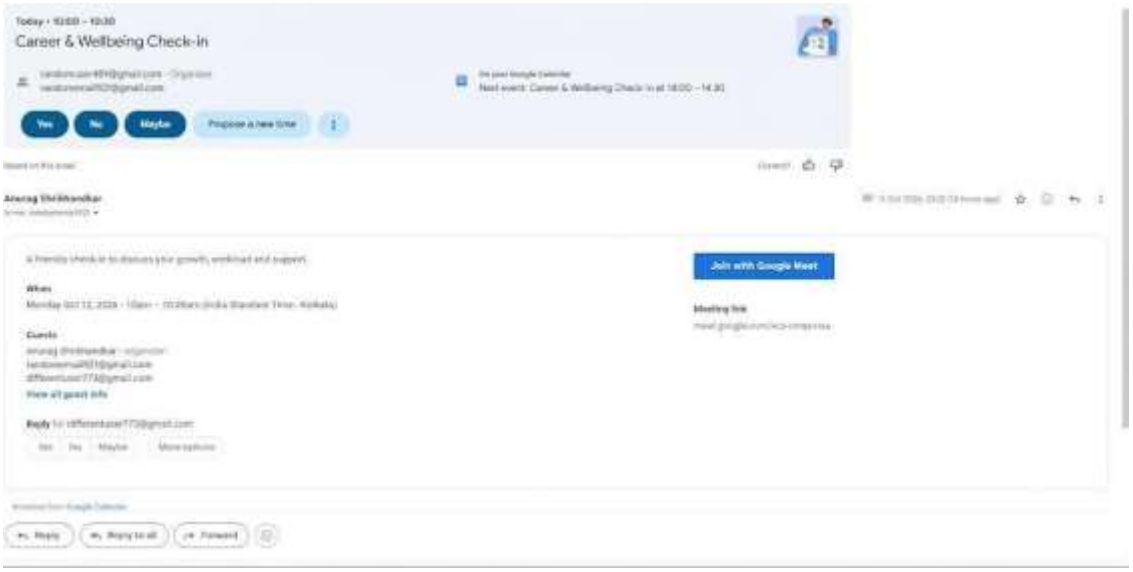


Fig. 6. Google Calendar invitation received from the workflow

V. DISCUSSION

A. Interpretation

On this benchmark and split, the ML results show that gradient boosting separates leavers from stayers with a ROC-AUC of 0.864 and detects just over half of the leavers (recall 0.553) at the 0.50 threshold. The test bench shows that a webhook request produced a calendar event with a Meet link in a mean of 1.42 seconds, that the cooldown blocked all five duplicate triggers, and that busy slots were moved to the next free slot. It does not demonstrate reliability at scale: with no observed failures, the one-sided 95% lower confidence bound on a success rate is only about 0.85 for 18 dispatches and about 0.55 for five duplicate tests. The results also do not show that meetings reduce attrition or that the system works on real employee data. RQ1 is answered for a single split. RQ2 is answered functionally by the test bench. RQ3 is answered only for the failure types that were tested, namely duplicates, busy slots and an invalid address; delays under load and exhausted retries were not tested. RQ4 is addressed in Sec. VI-A.

B. Research Gap and Contribution

The related work in Sec. II shows that attrition studies concentrate on prediction, whereas practitioner n8n material concentrates on alerts or reports [1]–[3], [12], [13]. The integration studied here joins the two, adding a review stage, a cooldown and separate metrics for each side. The test bench indicates that the integration path functions on a small set of synthetic runs, and the ML results show a useful ranking signal (ROC-AUC 0.864) with moderate recall (0.553) at a threshold of 0.50. These two findings interact. Scheduling starts only at a score of 0.80 or higher, which is stricter than the 0.50 used for the headline metrics. Fewer employees therefore reach the scheduling stage, and the recall at that operating point cannot exceed 0.553. The tier cutoffs thus directly determine HR workload and missed risk, and future work should report precision and recall at each cutoff. The prediction provides a ranking signal and the workflow provides the surrounding process, but neither validates the other.

Several assumptions remain. The dataset is fictional. The eligibility rule and threshold are design choices that influence how many employees are flagged. HR review is a procedural control whose effectiveness has not been tested here. The present evidence supports only claims about how the pipeline functions on test data; it does not support claims about retention or employee satisfaction.

VI. CONCLUSION

A. Limitations and Future Work

The main limitations are as follows. The study relies on one public, fictional dataset whose age, provenance and representativeness are limited, so generalisation to real organisations is limited. Class imbalance is handled only through a stratified split, without resampling or class weights, and 21 of the 47 test leavers are missed at the 0.50 threshold. No performance is reported at the 0.80 scheduling cutoff. There is no formal explainability analysis such as SHAP [8], and the method that produces the top reasons in the payload has not been evaluated. Evaluation relies on one split with 47 positive test records. The test bench consists of 20 runs on synthetic data, with no load testing and no failure injection for retries. The invitation description carries risk drivers. There is no external validation. Employee-level scoring carries privacy and ethical risks [9]. The system depends on external APIs and on workflow reliability. Finally, there is no evidence that scheduling meetings reduces attrition.

Future work includes validation on appropriately authorised organisational data; comparison with other classifiers under the same protocol, using cross-

validation with confidence intervals; threshold tuning and class-imbalance treatment inside training folds, with precision and recall reported at each risk-tier cutoff; calibration and explainability analysis; fairness evaluation across groups; minimisation of the payload and the invitation description; load testing and failure injection for the workflow and its retries; and a controlled study of whether the workflow helps HR staff. The items listed here are not part of the implemented system.

B. Concluding Remarks

This paper described a pipeline in which a gradient boosting classifier trained on the IBM HR Analytics dataset feeds an n8n workflow that supports HR follow-up meetings through Google Calendar. The pipeline includes risk tiers, a seven-day cooldown and an HR review stage, and the paper defined separate evaluation criteria for the prediction and scheduling components. The related work suggests that earlier studies emphasise prediction metrics and that practitioner workflows lack a reported evaluation of scheduling. On a held-out set of 294 employees, gradient boosting reached a ROC-AUC of 0.864, a PR-AUC of 0.682 and an accuracy of 0.898, against a majority-class baseline of 0.840, with a recall of 0.553 at the 0.50 threshold. A 20-run test bench on synthetic records produced calendar events with Meet links in a mean of 1.42 seconds and blocked all five duplicate triggers. The system treats every score as a risk indicator, depends on one fictional dataset, and makes no claim about reduced attrition. Its value lies in a reproducible and evaluable link between prediction and action, subject to the limitations described above.

ACKNOWLEDGEMENT

The authors thank Dr Kalyani H. Deshmukh and the Department of Computer Science and Engineering for their guidance and support. Parts of the manuscript text were drafted with the assistance of an artificial intelligence (AI) language model (Claude, Anthropic) and were reviewed and edited by the authors, who take responsibility for the content.

REFERENCES

[1] A. Raza, K. Munir, M. Almutairi, F. Younas, and M. M. S. Fareed, “Predicting employee attrition using machine learning approaches,” *Appl. Sci.*, vol. 12, no. 13, Art. no. 6424, 2022, doi: 10.3390/app12136424.

[2] W. Li, “A transformer-based deep learning framework to predict employee attrition,” *PeerJ Comput. Sci.*, 2023, doi: 10.7717/peerj-cs.1570.

[3] M. Nandal, V. Grover, D. Sahu, and M. Dogra, “Employee attrition: Analysis of data-driven models,” *EAI Endorsed Trans. Internet Things*, vol. 10, no. 1, 2024, doi: 10.4108/eetiot.4762.

[4] “IBM HR analytics employee attrition & performance,” Kaggle dataset (contributor: pavansubhasht). [Online]. Available: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset (accessed Oct. 11, 2026).

[5] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.

[6] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[7] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” *PLoS ONE*, vol. 10, no. 3, Art. no. e0118432, 2015, doi: 10.1371/journal.pone.0118432.

[8] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4765–4774.

[9] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–35, 2021, doi: 10.1145/3457607.

[10] n8n, “Google Calendar node,” n8n Documentation. [Online]. Available: https://docs.n8n.io/integrations/builtin/app-nodes/n8n-nodes-base.googlecalendar/ (accessed Oct. 11, 2026).

[11] Google, “Events: insert,” Google Calendar API v3 Reference. [Online]. Available: https://developers.google.com/calendar/v3/reference/events/insert (accessed Oct. 11, 2026).

[12] “Employee attrition risk detection & HR alerts using Azure OpenAI GPT-4o-mini & Gmail,” n8n workflow template 9309. [Online]. Available: https://n8n.io/workflows/9309-employee-attrition-risk-detection-and-hr-alerts-using-azure-openai-gpt-4o-mini-and-gmail/ (accessed Oct. 11, 2026).

[13] shwetamauryak, “HR-AI-Attrition-Workflow: HR employee attrition analysis using Python, Pandas, n8n automation, and AI-style reporting,” GitHub repository. [Online]. Available: https://github.com/shwetamauryak/HR-AI-Attrition-Workflow (accessed Oct. 11, 2026).