Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

Workforce Insights and Turnover Prediction Using Explainable AI

Lava Lavalith[1], Mahommed Farahan A[2], Anu L[3]
[1 2] Students and [3] Assistant Professor, Department of BCA
St. Francis College, Bengaluru, India
Email: daniel03anu2016@gmail.com

Abstract—Costly departures take place among employees, and the HR departments find out about it post facto. This paper proposes a workforce analytics system that features a prediction in the form of an individual employee attrition probability, label, risk band and a feature-level explanation of that decision. The proposed logistic regression model is calibrated using sigmoid calibration method and decision threshold of 0.25 selected based on validation data**.** One prediction service supplies the same output to all ten pages of the application, and an approximate Kernel SHAP explanation lists the features that raise and lower each estimate. Four classifiers were compared by cross-validated average precision. The comparison mixed weighting conditions, so it supports the model choice only for this application. On the IBM HR benchmark of 1,470 employees, the held-out test set gives a recall of 72.3% and a ROC-AUC of 0.805. Precision is 45.3%, and the model raises 41 false alarms. The benchmark is fictional and static. Calibration, explanation stability and fairness are not yet fully evaluated. The output is meant to support human review and not to decide employment outcomes.

Keywords—workforce analytics, employee turnover, probability calibration, explainable AI, SHAP, decision threshold, HR decision support

I. INTRODUCTION

Employee attrition disrupts teams and raises recruitment and training costs. Machine learning can flag employees who resemble past leavers, and a systematic review of AI in human resource management describes this predictive use [1]. A flag alone is hard to act on. A manager who sees a flagged employee asks why the model flagged that person and how far the score can be trusted.

Earlier attrition studies have tried AdaBoost, support vector machines and random forests [2], k-nearest neighbors [3] and fine-tuned large language models [4]. Most report a headline score. Such a score means little without the data split, the decision threshold and the metric behind it. The problem is sharper here because only 16.1% of the employees in our data carry an attrition label, so accuracy hides how poorly the minority class is found. Class weighting, resampling such as SMOTE [5] and threshold selection each address part of this problem. Raw classifier scores also need a calibration check before they are read as probabilities [6], [7]. SHAP [8], LIME [9] and counterfactual explanations [10], [11] make single predictions easier to inspect, although explanations have limits of their own [12], and simpler models deserve attention in high-stakes settings [13]. Bias in HR algorithms is a further reason to keep a person in the loop [14]–[16]. Section II compares these works in detail.

This paper describes an attrition prediction and decision-support system built as an undergraduate project. Its core is a calibrated logistic regression model. For each employee it gives a probability, a label and a risk band. A Kernel SHAP explanation on the same page shows which features pushed the probability up or down, so the user reads the number and the reasons together. Workforce analytics, a model comparison and two rule-based modules for retention cost and promotion readiness support this core.

The work addresses three research questions. *RQ1:* How well can attrition be flagged on a benchmark dataset at an operating point that favors recall? *RQ2:* Can each flag carry a calibrated probability and an explanation that a reviewer can read? *RQ3:* What do the results allow an HR team to conclude, and what do they leave open?

We separate the engineering contributions from the research findings. Two contributions are engineering work. The first is one prediction service that keeps probabilities, labels and risk bands consistent across ten application pages. The second is a prediction page that pairs every score with a local Kernel SHAP explanation and a what-if simulator. Two contributions are research findings. The third is a calibrated, recall-oriented model with a documented threshold and three risk bands. It is evaluated with imbalance-aware metrics, a trivial baseline and interval estimates. The fourth is a plain account of which components were evaluated and which were only built (Table IX). We claim no new prediction method and no new explanation technique.

Section II reviews related work and Section III states the problem. Section IV describes the system, Section V the data and Section VI the model. Section VII covers the prediction page and the SHAP explanations. Sections VIII and IX give the results and their discussion. Section X lists limitations and future work, and Section XI concludes.

II. RELATED WORK

A. Attrition Prediction and Class Imbalance

Classical classifiers dominate attrition research. Examples include k-nearest neighbors [3] and AdaBoost, support vector machines and random forests [2]. Large language models have also been tried [4]. The closest prior system is Retention Is All You Need [17]. It already combines an HR decision-support tool with SHAP, natural-language explanations and what-if analysis, so we do not present that combination as new. Our emphasis is different. We concentrate on calibrated probabilities, a threshold chosen on validation data, the same output on every page, and

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

an evaluation that reports false alarms and its own gaps. XGBoost [18] and wider studies of tabular learning [19]–[21] report that tree-based models are strong baselines and that results depend on data size and tuning. For that reason we compare tree models with a linear one. We did not run XGBoost or deep networks. For skewed classes the usual remedy is oversampling such as SMOTE [5], supported by toolkits such as imbalanced-learn [22]. We leave the data untouched and choose the decision threshold on validation data instead.

B. Calibration and Explainability

A classifier score need not be a probability. Guo et al. [6] show that post-hoc scaling, including a variant of Platt scaling, can calibrate predictions. Reference [7] surveys how calibration is assessed and improved. SHAP assigns each feature an additive contribution to one prediction [8], and its tree-based variant links local explanations to global understanding [23]. LIME explains a prediction with a local surrogate model instead [9]. Surveys and position papers set out goals and taxonomies for interpretability [24]–[26] and weigh the trade-offs of explaining black boxes [13], [26], [27]. The authors of [12] warn that Shapley-based importance has pitfalls. For that reason we describe SHAP output as association and not as cause. The what-if simulator draws on counterfactual explanations [10], [11].

C. Fairness and Human Oversight

Surveys of bias in machine learning [14], criteria such as equality of opportunity [15] and work on algorithmic hiring [16] show that predictive systems can disadvantage particular groups. Practical research related to application of the explainable methods shows there are cases when explainability is used inappropriately [28]. It has already been established experimentally that transparency does not always ensure a good decision process [29]. That is why our outputs are suggestions and require human evaluation.

TABLE I
COMPARISON OF APPROACHES

Approach	Typical limitation	This work
Accuracy-focused classifiers [2], [3]	Weak on the minority class	PR-AUC selection, recall-oriented threshold
Uncalibrated scores	Probabilities not reliable	Sigmoid (Platt) calibration [6], [7]
Black-box predictors [4]	Hard to justify	SHAP [8], what-if [10], [11]
Model-only studies [1]–[4]	No integrated workflow	One service behind ten pages
Automated HR decisions	Fairness and trust risks [14]–[16]	Human review, stated limits

III. PROBLEM STATEMENT

Many organizations still decide on retention from delayed, descriptive reports. A manager sees who left, not who might leave. Existing attrition studies mostly report a classifier and a headline score [1]–[4]. This leaves three needs open: scores that behave like probabilities [6], [7], evidence for why an employee is flagged [8], [10], and clear limits on how the output may be used [14]–[16], [28].

Let x be the feature vector of an employee, with 30 raw input features, and let y be the binary attrition label (1 = left, 0 = stayed). The system estimates the calibrated probability

$$p(x) = P(y = 1 \mid x) \quad (1)$$

and assigns the label

$$\hat{y}(x) = 1 \text{ if } p(x) \geq \tau, \text{ otherwise } 0 \quad (2)$$

where $\tau = 0.25$ is the decision threshold chosen on validation data. Because the positive class is rare, τ sits below 0.5. The model is judged on recall, F1 and PR-AUC as well as accuracy. The target in (1) is the attrition label that ships with the benchmark. The records carry no observation dates, so the probability has no forecast horizon such as the next twelve months. The task is therefore classification of benchmark labels and not a validated forecast of when a real employee will leave. The probability is best read as how closely an employee resembles the labeled examples.

We use four terms in a fixed sense. Attrition is the benchmark label, and turnover is the wider workforce phenomenon it stands for. The prediction probability is the calibrated output p(x). The decision threshold is the cutoff of 0.25 that sets the label. A risk band is a display category (LOW, MEDIUM or HIGH) derived from the probability with separate cutoffs.

Four objectives follow from this setting. The first is to estimate attrition probability with a calibrated model that favors recall. The second is to explain each prediction with SHAP. The third is to let users test scenarios by editing permitted fields and re-scoring the record. The fourth is to add workforce analytics and rule-based modules that stay outside the learning model.

IV. SYSTEM OVERVIEW

A. Architecture

The system is a Python web application. A FastAPI backend served by Uvicorn exposes the API routes and mounts the Dash user interface through WSGI middleware, so both run in one process. The interface uses Dash, Dash Bootstrap Components and Plotly and has ten pages. Dash callbacks call Python service functions directly instead of sending internal HTTP requests, which keeps the pages responsive. Employee records sit in an SQLite database. It holds a table of the 1,470 employees, a table of stored predictions for existing employees and a table of 40 demonstration project rows. The project rows were generated with a fixed random seed and are not real project history. The REST API has routes for health checks, dashboard aggregates, employee lookup, prediction (manual and for existing employees), what-if scoring, model performance, reports, recommendations and the rule-based assistant. Fig. 1 shows the overall flow, and Table II lists the components that are used.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

TABLE II
TECHNOLOGY STACK

Component	Technology
Web API	FastAPI, Uvicorn
User interface	Dash, Dash Bootstrap Components, Plotly
Database	SQLite
Data and model	pandas, NumPy, scikit-learn [30], joblib
Explanations	SHAP (KernelExplainer) [8]
Reports	python-docx (Word report download)

B. Centralized Prediction Service

Every page takes its probabilities from one prediction service. The service loads the saved scikit-learn pipeline [30] once. It is the only source of probabilities, labels, risk bands and the threshold. This avoids a common dashboard failure, where different pages apply different cutoffs and show different answers for the same employee. A batch path scores all employees for the analytics pages. The single-employee path adds history logging and the SHAP explanation.

A request follows the same path on every page. The form values, or the stored record of an existing employee, go to the service as raw fields. The saved pipeline encodes and scales them and returns a calibrated probability. The service applies the threshold of 0.25 to set the label and the band cutoffs to set the risk level. For an existing employee the result is written to the history table. If an explanation is requested, the service runs Kernel SHAP against the same pipeline. The what-if path is identical, except that it starts from an existing record with a few permitted fields edited. Everything passes through one function. A change of threshold or model therefore takes effect on every page at once. A count on a summary page can also be traced back to the records behind it.

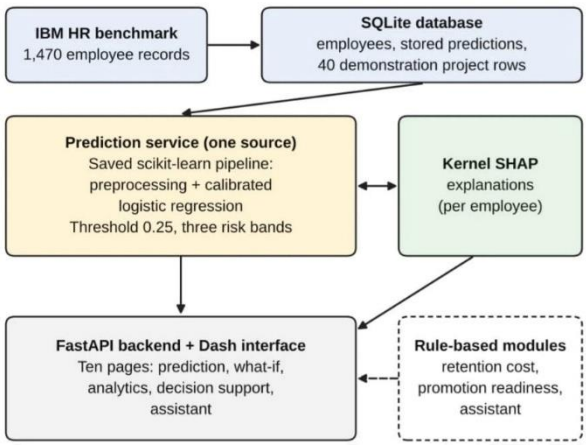


Fig. 1. System architecture. One prediction service connects calibrated scoring to the interface and to the SHAP explanations. The rule-based modules stay outside the learning model.

C. Application Pages

The interface has ten pages in four groups (Table III). Overview shows historical workforce metrics next to current risk summaries. Employee Insights offers searchable employee data alongside predictions. Reports produces workforce summaries and a downloadable Word report built with python-docx. Historical and predicted figures stay distinct in these views. A historical attrition count comes from the observed benchmark labels. The number of flagged employees depends on the saved model and the chosen cutoff. The two can therefore differ without either being wrong.

TABLE III
APPLICATION PAGES BY GROUP

Group	Pages
Prediction	Turnover Prediction, What-If Simulator
Analytics	Overview, Employee Insights, Model Performance, Reports
Decision support	Recommendations, Retention Cost Estimator, Promotion Readiness
Assistant	HR Analytics Assistant (rule-based)

D. Supporting Modules

Three modules run on explicit rules and are kept apart from the learning model. The Recommendations page returns rule-based observations and candidate staffing actions. The Retention Cost Estimator approximates the cost exposure of one departure as

$$C_i = 12 \times m_i \times k \qquad (3)$$

where m_i is monthly income and k is a replacement-cost factor It defaults to 1.5 and ranges from 0.5 to 2.0. This is an assumption not the cost. The outcome of this calculation will be forecasting by annual income. This is neither organizational cost nor forecasted one.The Promotion Readiness score runs from 0 to 100. It is built by rule from time since promotion, performance, time in the current role, training and pay relative to peers, and it has not been validated against promotion outcomes. The HR Analytics Assistant is a rule-based question-answering module. It answers from computed workforce statistics and does not use a language model or generative AI. No Since there are no supporting results found for these modules, they have been listed in this report as a help tool.

E. Implementation Details

Module Preprocess & Module Calibrated Model are artefacts from a package containing the version details. The Version Details from Model performance Page: Version 2.0.The cache of employee rows and predictions lives in process memory, so a change to the database or the model needs a restart or an explicit cache invalidation. The loaded data, the saved model and its metadata must all refer to the same version, and cached outputs must not survive a change to one of them. The application has no authentication layer and is meant for a trusted local environment. Because it handles employee-level information, access control must be added before any deployment that exposes such data.

V. DATASET AND PREPROCESSING

A. DATASETS AND CLEANING

The IBM HR Analytics Employee Attrition & Performance dataset is used in this study. This static dataset has been widely used in employee attrition research [4], [17]. One study describes it as synthetic [17]; therefore, it does not represent data from a live organization. Our results therefore describe associations inside this benchmark and not findings about a real

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

workforce. The original file has 35 columns (34 attributes and the attrition label), and its counts of 237 leavers and 1,233 stayers match those reported in [17]. Three constant columns (EmployeeCount, Over18 and StandardHours) were dropped, which leaves 32. The copy we inspected had 1,470 records and no missing cells, duplicate rows or duplicate employee identifiers. The employee number is kept only as a record identifier. The attrition column is the outcome. Neither is used as an input.

B. Features and Preprocessing

That leaves 30 raw input features in five groups. Personal details are age, gender, marital status, education and education field. Job context covers department, job role, job level, business travel, overtime and distance from home. Pay covers monthly income, the daily, hourly and monthly rates, salary hike and stock option level. Satisfaction and engagement cover environment, job and relationship satisfaction, job involvement, work-life balance and performance rating. Tenure covers total working years, years at the company, in the current role, since the last promotion and with the current manager, the number of companies worked at and training times last year. By type they are two binary fields (Gender, OverTime), five nominal fields (BusinessTravel, Department, EducationField, JobRole, MaritalStatus) and 23 numeric fields. Binary fields are one-hot encoded with one category dropped. Nominal fields are one-hot encoded with unknown categories ignored. Numeric fields are standardized. All steps sit inside the saved scikit-learn pipeline [30]. The prediction code passes raw fields, so training and serving use identical preprocessing and no page builds its own encoded row. Ignoring unseen categories avoids an encoding failure. It says nothing about whether a new category is well covered by the training data.

C. Class Imbalance and Data Split

Only 16.1% of employees left (Table IV). A model that always predicts stayed would be right 83.9% of the time on the full data. We do not oversample with SMOTE [5]. Imbalance is handled with balanced sample weights during the initial fitting and with threshold selection on validation data. The data were split into stratified training, validation and test partitions of 60%, 20% and 20%, with a fixed random seed of 42. Candidate fitting, cross-validation and sigmoid calibration use the training partition. Validation probabilities guide the threshold and the calibration report. The test partition is held back for the final evaluation, so the threshold was never tuned on test errors. The software environment is Python 3.12 with scikit-learn [30].

TABLE IV
DATASET SUMMARY

Item	Value
Employees (rows)	1,470
Columns after cleaning	32 (35 in the original file)
Model input features	30 (2 binary, 5 nominal, 23 numeric)
Stayed (No)	1,233 (83.9%)
Left (Yes)	237 (16.1%)
Training / validation / test	882 / 294 / 294

VI. PREDICTION MODEL

A. Models Under Consideration and Selection

In this comparison we will use four different machine learning models trained on our preprocessed dataset: logistic regression, decision tree, random forest and histogram-based gradient boosting (HistGradientBoosting). The choice consists of one interpretable linear algorithm and three highly efficient tree-based models – standard benchmarks on tabular data [19], [20]. Each of these models is examined via 5-fold stratified cross validation on the training split. Average precision, reported as PR-AUC, was the main metric. Validation recall and F1 were secondary checks. Table V gives the values as the Model Performance page shows them.

Logistic regression had the highest cross-validated PR-AUC (0.645). It also had the best validation recall (0.708) and F1 (0.496), so it was kept for the saved pipeline. The tree-based models trail it on every metric, and the decision tree has the lowest PR-AUC (0.284). Its coefficients can be inspected, which fits the argument for interpretable models in high-stakes settings [13]. One caveat applies to the whole table. Balanced sample weights were used when the initial candidates were fitted but not in the cross-validation call or the later calibration refit. The comparison therefore mixes weighting conditions. It supports the choice of a classifier for this application, but no general ranking of the model families follows from it. We do not claim that logistic regression is better than the other families in general.

TABLE V
MODEL COMPARISON ON TRAINING AND VALIDATION DATA

Model	CV PR-AUC	Validation recall	Validation F1
Logistic Regression (selected)	**0.645**	**0.708**	**0.496**
HistGradientBoosting	0.602	0.375	0.468
Random Forest	0.575	0.312	0.405
Decision Tree	0.284	0.417	0.377

B. Probability Calibration

Classifier scores are not automatically probabilities [6], [7]. We fitted sigmoid (Platt) calibration to the selected model. It maps the raw score $f(x)$ to a probability by

$$p(x) = 1 / (1 + \exp(A \times f(x) + B)) \quad (4)$$

using five-fold CalibratedClassifierCV on the training partition only. The Brier score is the mean squared difference between the predicted probability and the binary outcome. We report it on validation data in Section VIII. The refit did not keep the original sample weights, which matters when the scores are read. A reliability diagram would be the natural check of calibration itself, and we have not produced one. In this paper the word calibrated means that the scores went through sigmoid calibration. It does not mean that good calibration has been shown.

The Brier score and recall measure different things. The Brier score measures the distance between a predicted probability and the observed label. Recall counts the positives found after thresholding. A model can change its probability estimates without changing every label, and a threshold can change labels

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

without touching the probabilities. So we report both kinds of measure.

C. Decision Threshold and Risk Bands

The operating threshold was chosen on validation data. We swept thresholds from 0.15 to 0.50 on the validation probabilities and took the one with the best F1. That value was 0.25. An employee with $p(x)$ of at least 0.25 is labeled likely to leave, and everyone else is labeled likely to stay. The value suits this validation split. No measured cost of false alarms and no limit on how many cases a manager can review went into it. A lower threshold generally sends more records to review, and a higher one can leave more leavers undetected.

The application also shows a three-level risk band whose cutoffs differ from the classification rule (Table VI). An employee can therefore be labeled likely to leave and still sit in the MEDIUM band. The threshold answers whether a case should be reviewed. The band conveys how strong the signal is. For example, a probability of 0.30 falls in the MEDIUM band and crosses the threshold, and a probability of 0.55 falls in the HIGH band and gets the same label. One further cutoff, 0.70, is used only for an interface count of high-risk employees. That count is not the size of the HIGH band. Showing the number beside the label and the band helps users read cases near a boundary without treating a small numeric difference as a change in the employee's intent.

TABLE VI
PROBABILITY CUTOFFS AND THEIR ROLES

Probability p	Risk band	Label or use
$p < 0.25$	LOW	Likely to stay
$0.25 \le p < 0.50$	MEDIUM	Likely to leave
$p \ge 0.50$	HIGH	Likely to leave
$p \ge 0.70$	None	Interface count of high-risk employees only

D. Evaluation Metrics

Because the positive class is rare, we report precision, recall and F1 for the positive class alongside accuracy,

$$\text{Precision} = TP / (TP + FP) \quad (5)$$

$$\text{Recall} = TP / (TP + FN) \quad (6)$$

$$F1 = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \quad (7)$$

where TP, FP and FN count true positives, false positives and false negatives at the chosen threshold. ROC-AUC and PR-AUC do not depend on the threshold. PR-AUC is computed as average precision. Its baseline is the positive prevalence and not 0.5, which makes it the more informative of the two when only about 16% of the records are positive. We treat recall as the main metric because the system is meant to surface employees for review. Precision tells us how many unnecessary reviews that costs.

VII. PREDICTION PAGE AND SHAP EXPLANATIONS

A. The Prediction Page

The Turnover Prediction page scores an employee and returns the calibrated probability, the label and the risk band. Fig. 2 shows the result for employee 31. The page shows the employee ID, the label, the probability with a ring gauge, the risk level, the model version (2.0) and the threshold used (0.25). A bar repeats the probability. Here the model gives 14.60%, which is below the threshold of 0.25. The page therefore shows the label LIKELY TO STAY with a LOW risk level. The model version and threshold are printed on the page so that a screenshot can be traced back to the configuration that produced it.

Below the result, a panel titled Why This Prediction? splits the strongest contributing features into factors that increase the risk and factors that reduce it (Fig. 3). For employee 31 it lists four factors on the first side and five on the second, each with a signed value and a bar. A Prediction Summary restates the result in plain words and names the top two factors on each side. It adds that these factors influenced the estimate and do not prove why someone will leave.

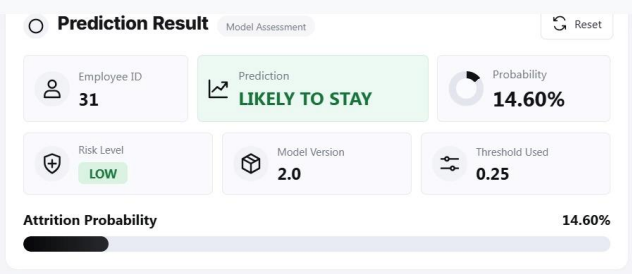


Fig. 2. Prediction result for employee 31 on the Turnover Prediction page: label, probability, risk level, model version and threshold used.

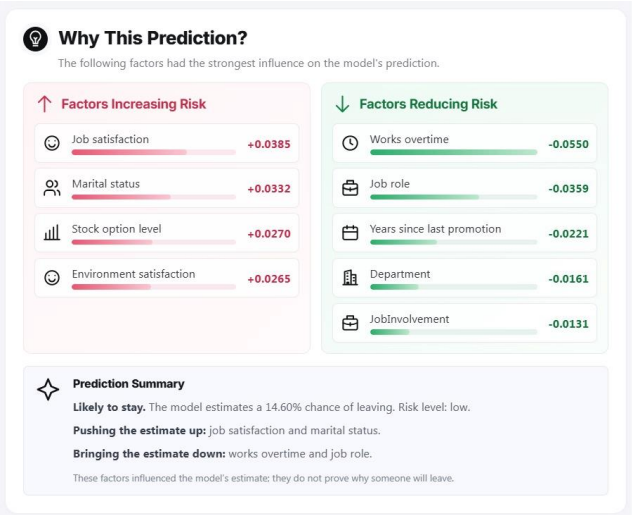


Fig. 3. Kernel SHAP explanation for employee 31. The signed values are contributions in probability units, listed as factors that increase and factors that reduce the estimated risk.

Behind the page, the single-employee path calls the prediction service, which returns the probability, the label, the risk band and the threshold. It then requests a Kernel SHAP explanation. Predictions for existing employees are written to the history table, and manual predictions are not. A saved entry records a scoring event and not an attrition outcome observed later. Repeating a prediction on an unchanged benchmark record therefore adds no evidence about what happens over time.

Manual predictions need a clear warning. The form exposes only 12 of the 30 model inputs and copies the other 18 from a baseline employee record. A manual result is therefore conditional on those hidden values. The manual API route requires all 30 inputs.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

B. Kernel SHAP

Explanations use SHAP [8], which writes a prediction as a base value plus additive feature contributions,

$$f(x) = \varphi_0 + \Sigma\, \varphi_i(x) \qquad (8)$$

where φ_0 is the base value and φ_i is the contribution of feature *i*. We use model-agnostic Kernel SHAP over the saved calibrated pipeline, because tree-specific explainers [23] do not suit a logistic regression pipeline. Each explanation returns raw feature labels, values and estimated contributions, sorted by impact and split into factors that raise the modeled output and factors that lower it. The explainer works on the calibrated probability, so the contributions are in probability units. A value of +0.0385 is about 3.9 percentage points. The result still depends on the background data and the explainer configuration. It is one view of the model and not a fixed fact about the employee.

Kernel SHAP estimates contributions by sampling subsets of features, filling in the missing ones from background data and fitting a weighted linear model to the resulting predictions [8]. It needs only the prediction function, so it can sit on top of the saved pipeline together with its preprocessing. The cost grows with the number of features and samples, which is one reason the global view uses a sample of employees. Because SHAP values are additive, the contributions plus the base value should reproduce the displayed probability. This would give a quick consistency check. We have not tested it systematically (Table IX).

A linear model may seem to make SHAP unnecessary, since its coefficients can be read directly. In our pipeline that reading is awkward. The coefficients act on standardized and one-hot encoded columns, so a single raw field such as job role is spread over several coefficients. The calibration step also bends the scale between the raw score and the probability that the user sees. Kernel SHAP works on the final probability and reports one contribution per raw field, which is the unit an HR reviewer thinks in. We chose it over LIME, which fits a new local surrogate around each record and depends on how that neighborhood is sampled [9]. Kernel SHAP also samples, but it rests on the Shapley value definition [8]. That gives the additivity used for the consistency check and a clear reading of the base value. The coefficients themselves stay available for inspection.

C. Reading the Explanation

An explanation is read by starting at the background value and following the signed contributions to the final prediction. In Fig. 3 the left list holds the factors that raise the modeled risk for employee 31: job satisfaction (+0.0385), marital status (+0.0332), stock option level (+0.0270) and environment satisfaction (+0.0265). The right list holds the factors that lower it: works overtime (−0.0550), job role (−0.0359), years since last promotion (−0.0221), department (−0.0161) and job involvement (−0.0131). The four positive values sum to +0.125 and the five negative values to −0.142. The displayed factors therefore leave a net push of about −0.017, and the base value and the remaining features account for the rest of the 14.60%. The page does not print the base value or the feature values, so a reader cannot check the sum from the panel alone. Contributions of 0.01 to 0.05 look small next to the threshold of 0.25. Next to a score of 14.60% they are large. They matter most for employees whose probability sits close to the cutoff, where one factor can decide whether the label flips.

A positive contribution means that a feature raises this model output relative to the background. The employee is not necessarily unhappy or about to leave. Nor does the value measure how real-world risk would change if that feature alone were altered. Shapley-based importance has known pitfalls [12], and explanations do not guarantee better decisions from the people who read them [28], [29]. One employee example shows how the interface works, but it cannot establish explanation quality. The application labels explanations as associations and presents them as input to human review.

D. Global View and Categorical Features

Global explanations are computed on a deterministic sample of up to 200 employees with a fixed random seed, which keeps the computation affordable. The result is cached against the modification times of the data and model files, so it refreshes when either file changes. A feature that stands out across the sample need not dominate every employee's explanation.

Kernel SHAP perturbs numeric inputs, but several of our features are categorical. The explanation code maps each category to a numeric code, lets SHAP perturb the codes and maps the perturbed values back to the nearest known category. This is a weakness of the method. Categories such as department or job role have no natural numeric distance. The numeric codes are only a computational stand-in, and the order of the codes can change the result. The fact that the model is logistic regression does not by itself validate this procedure. For this reason the explanations are labeled as approximate. Use them to inspect one prediction, not to rank independent causes exactly. We have not yet tested their stability across background samples and category encodings. A possible alternative is to explain the transformed one-hot columns directly and group their contributions by raw field. We list both steps as future work. Showing an explanation in the interface does not validate it, and we do not offer these explanations as validated evidence.

E. What-If Simulator

The What-If Simulator lets a user change permitted fields of an existing employee and re-score the same record with the same model, in the spirit of counterfactual explanations [10], [11]. It shows how sensitive the score is to an edit. Whether the edit would stop anyone from leaving is a separate question, because the model has seen no interventions and learned only from labeled benchmark records.

F. Using Explanations in Practice

We expect users to start from a workforce summary and open an individual record when they need a closer look. There they read the probability and the band first, then scan the two factor lists, and may try a permitted scenario. Each step answers a different question. The probability summarizes the model output. The explanation shows how features are associated with it. The scenario shows how sensitive the score is to edited inputs. None of the three captures what an employee intends, and none shows that a given retention action will work.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

TABLE VII
INTERPRETATION OF DECISION-SUPPORT OUTPUTS

Output	Interpretation
Attrition probability	Calibrated model score.
SHAP contribution	Association, not a cause.
What-if result	Re-scoring; no intervention effect.
Retention cost	Annual income times an assumed factor.
Promotion readiness	Rule-based; separate from the model.

Some contributing features can be discussed with an employee, such as job satisfaction or the role itself. Others describe the person and not the job. Marital status is the obvious example, and it appears among the contributors for employee 31 in Fig. 3. These are legitimate columns of the benchmark, but a reviewer should not act on them.The need in these additional criteria of bias is another reason why the fairness audit of Section X is required. Table VII contains an interpretation for all metrics used by this audit.

VIII. EXPERIMENTAL RESULTS

A. Test metrics. Results are reported on the held-out test partition of 294 employees. This partition was not used for model comparison or threshold selection. It holds 47 employees who left and 247 who stayed. The metrics use the calibrated model at the threshold of 0.25 and come from the Model Performance page. Fig. 4 shows them as proportions. Accuracy is 81.6%, precision 45.3%, recall 72.3% and F1 55.7%. ROC-AUC is 0.805 and PR-AUC is 0.621.



Fig. 4. Held-out test metrics on 294 unseen test records at a threshold of 0.25, as shown on the Model Performance page.

The ROC curve in Fig. 5 rises quickly. It reaches a true positive rate of about 0.72 at a false positive rate near 0.12, stays flat to about 0.3 and ends above the diagonal of a random classifier. The precision-recall curve in Fig. 6 starts near 1.0 and falls to about 0.54 just before a recall of 0.72. It then drops to about 0.31 and ends at the base rate of about 0.16. The operating point at the threshold of 0.25 comes from Table VIII and is not marked on the plots. Its true positive rate is 34/47 = 0.723 and its false positive rate is 41/247 = 0.166. That point sits on the flat section of the ROC curve, which matches the plot. In the same way, the precision-recall point of 0.723 recall and 0.453 precision sits on the vertical drop at a recall of 0.72, between about 0.54 and 0.31.

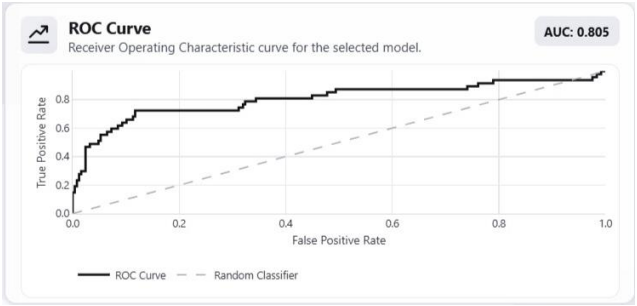


Fig. 5. ROC curve of the selected model on the held-out test partition (AUC 0.805), with the random-classifier diagonal.

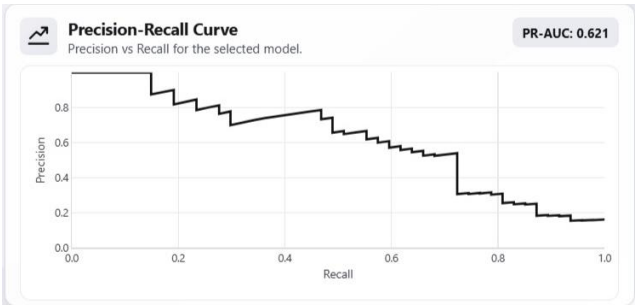


Fig. 6. Precision-recall curve of the selected model on the held-out test partition (PR-AUC 0.621).

B. Confusion Matrix and Review Workload

Table VIII gives the confusion matrix. The model found 34 of the 47 employees who left (recall 72.3%) and missed 13. It also flagged 41 employees who stayed, so only 34 of its 75 positive flags were correct (precision 45.3%). Recall ends up above precision on purpose, because the threshold was tuned for F1 on a rare class. At this threshold 75 of the 294 test records are flagged, about 25.5% of the partition. The false-positive rate is about 16.6% (41 of 247). The miss rate among leavers is about 27.7% (13 of 47). These figures describe this test partition and should not be carried over unchanged to another workforce.

TABLE VIII
CONFUSION MATRIX ON THE HELD-OUT TEST PARTITION

Actual \ Predicted	Stayed	Left	Total
Stayed	206	41	247
Left	13	34	47
Total	219	75	294

C. Accuracy Against a Trivial Baseline

An always-stayed classifier would reach 247/294 = 84.0% test accuracy. That is higher than the model's 81.6%. Yet it would find none of the leavers, while the model finds 34 of 47. Accuracy alone would hide the trade-off between detected leavers and false alarms. The test PR-AUC of 0.621 should be read against the positive prevalence of 47/294, roughly 0.160. PR-AUC summarizes the whole precision-recall relationship. The precision of 45.3% belongs to a single operating point. The two numbers answer different questions.

The confusion matrix gives further views. Of the 219 employees labeled as staying, 206 (94.1%) did stay. A negative label is therefore more reliable than a positive one, although it is still no guarantee. Specificity is 83.4% (206 of 247). Balanced accuracy, the mean of recall and specificity, is 77.9%, against 50% for the always-stayed classifier. The model needs about 2.2 flags for

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

every true leaver it finds. Its precision is roughly 2.8 times the base rate of 16.0%. This does not make the model ready for operational use. It shows only that the model finds more leavers than a trivial rule, at the cost of false alarms.

D. Calibration

The validation Brier scores are 0.1500 for the initially weighted model, 0.0961 after the calibrated refit and 0.1366 for a constant base-rate predictor. These are validation figures and are separate from the held-out metrics in Fig. 4. The sample weighting also changed at the refit, so the drop from 0.1500 to 0.0961 cannot be credited to sigmoid calibration alone. On validation data the saved model scores better than the base-rate predictor. The gap is $0.1366 - 0.0961 = 0.0405$. Only about 16% of the records are positive, so a predictor that always answers with the base rate already scores 0.1366, and that is the fair point of comparison. That is weaker than showing good calibration across the full probability range or across groups. A reliability diagram and a matched comparison of calibrated and uncalibrated models are still missing [6], [7].

E. Sampling Uncertainty

The test set holds only 47 leavers, so every rate computed on them is uncertain. Using the counts in Table VIII, a 95% Wilson interval for the recall of 34/47 runs from about 58% to 83%. For the precision of 34/75 it runs from about 35% to 57%. The headline values of 72.3% and 45.3% are therefore best read as rough levels and not as exact properties of the model. Repeated splits or cross-validated test estimates would narrow these ranges.

IX. DISCUSSION

A. Why a Simple Model

The cross-validated PR-AUC values in Table V explain why we implemented logistic regression. They do not give a general ranking, because the weighting protocol still has to be made consistent. Studies of tabular learning show that results depend on data size and configuration [19], [20]. A model with inspectable coefficients also suits a system whose main output is an explanation [13].

B. Recall over Accuracy

The model catches 34 of 47 leavers but raises 41 false alarms, and its accuracy falls below the always-stayed baseline. Whether that trade is acceptable depends on intervention costs, manager capacity and policy, none of which we measured. False alarms use up manager time and may create unfair suspicion. A flagged employee is a reason to start a conversation and not evidence of intent to leave. The 13 missed leavers show the opposite risk. A record below the threshold cannot be assumed to carry no retention concerns. This agrees with guidance that predictive HR tools need human oversight [14]–[16], [28], [29].

C. Comparison with Other Studies

Scores in the literature cannot be compared directly with ours. Study [4], for example, reports weighted-average precision, recall and F1 after oversampling the training data with SMOTE, and a fine-tuned language model reaches an F1 of 0.92 in that setting. Our figures describe the positive class at one threshold on an untouched test partition. Weighted averages are dominated by the majority class and look higher on imbalanced data. We do not rank systems by headline numbers across papers.

D. Calibration, Trust and Rules

The calibrated refit lowers the validation Brier score, which supports probability-oriented evaluation [6], [7]. Calibration across subgroups or other workforces remains untested. A benchmark probability is not an independently validated chance that a real employee will leave. Interpretations should help clarify the meaning of each of the metrics used. Mistakes in the model and issues with the data should be preserved as such. Speaking of the retention cost and promotion readiness metrics, there are some assumptions made about them (the replacement factor and the score weights). This is why the first two metrics are separated from the attrition metric. The attrition component needs a reproducible predictive evaluation on suitable outcome data. The cost component needs defensible replacement-cost assumptions. The promotion component needs an independent assessment of its criteria and intended use. A strong result in one component cannot validate the others.

E. Sensitive Features and Fairness

The benchmark includes Gender, Age and MaritalStatus among its 30 input features, so the model can use them and SHAP can surface them, as it does for marital status in Fig. 3. A feature contribution is not evidence of discrimination, but it is a signal that group-level errors deserve a look. An equality-of-opportunity check [15] would compare recall across groups. A high false-alarm rate in one group would mean more unwarranted scrutiny for those employees [14], [16]. We have not run such a check. Subgroup recall, false-positive rate and group size would be the first measures to report. Until then, a prediction is a prompt to review the whole situation and not a ranking of people.

F. Responsible Use

We suggest three rules for any pilot of a tool like this. The output should only be used to prioritize conversations and never as the sole basis of an adverse decision. The explanation and the caveats should always be shown together with the score. Every scoring event should be logged so that reviewers can audit how the tool was used. The prediction history table already supports the last rule for existing employees. Reviewers should also know that a manual prediction from the short form is a quick check and not a full assessment of a person.

G. Threats to Validity

Internal validity is limited by the weighting inconsistency in the model comparison and by the use of a single stratified split. The reported metrics carry split-to-split variation on top of the sampling interval above. External validity is limited by the benchmark itself, which is fictional, static and small. Nothing here shows how the model would perform on a real workforce with a different mix of roles and a different attrition rate. Construct validity is limited because the label has no time horizon. We cannot say what kind of departure the model flags or how soon it would happen.

H. Answers to the Research Questions

RQ1. On the held-out test partition the model finds 34 of 47 leavers (recall 72.3%) at a precision of 45.3% and a ROC-AUC of 0.805. It does so with 41 false alarms. This holds for one

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

fictional benchmark and one split. *RQ2.* Each flag carries a probability and a SHAP explanation. On validation data the probabilities score better in Brier terms than a However, the predictor is not well-calibrated. The explanations are approximate themselves and their robustness has not yet been verified. RQ3. The findings show that output can be used to filter out data for human checking on this benchmark dataset. This finding does not validate real world predictions, notions of fairness, or efficacy of retention action.

X. LIMITATIONS AND FUTURE WORK

A. Limitations

Data. The benchmark is static and fictional and has no date fields, so the system can neither predict time to leave nor forecast attrition trends. *Partial inputs.* The prediction form exposes only 12 of the 30 model inputs and copies the other 18 from a baseline record. *Evaluation consistency.* Balanced sample weights are applied when the initial models are fitted but not in the cross-validation call or the calibration step, so Table V is not fully like-for-like. The accuracy, precision, recall, F1 and AUC figures come from saved metadata and are not regenerated when a page is served.

TABLE IX
IMPLEMENTED VERSUS EVALUATED COMPONENTS

Component	Status	Evidence so far
Attrition model	Implemented, evaluated	Held-out test metrics, one split
Calibration	Implemented, partly evaluated	Validation Brier score; no reliability diagram
SHAP explanation	Implemented, not evaluated	No fidelity, stability or user test
What-if simulator	Implemented, not evaluated	Re-scoring only; no outcome data
Cost, promotion, assistant	Rule-based, not validated	Assumptions stated, no outcome data

Table IX separates what has been measured from what has only been built. Explanations. Kernel SHAP with categorical encoding is an approximation, and its output shows association and not cause [12]. Interface counts. The prediction page counts high-risk employees at a probability of 0.70, while the HIGH band starts at 0.50, so the two numbers differ by design. Fairness and security. We have not audited the model for bias across groups, which matters for any HR use [14], [15]. The application has no authentication layer. Demonstration data. The 40 project rows are generated examples and not real history.

B. Future Work

Some steps are needed before the results can count as a valid experiment. We plan to fix the weighting inconsistency and rerun the model comparison under one protocol with the same preprocessing, weighting and cross-validation. We plan to compare calibrated and uncalibrated versions under matched conditions and to add a reliability diagram. For the explanations we plan to check that SHAP contributions sum to the calibrated probability within a stated tolerance and to test their stability across background samples and category encodings. We also plan to report subgroup recall and false-positive rates, and to repeat the evaluation on several splits. The threshold should again be chosen on validation data, with the test set kept for reporting.

Other steps concern deployment and wider use. These include validation on real organizational data with dates, a move toward time-to-event models, role-based access control and a user study with HR practitioners. The study should test whether the explanations improve decisions [28], [29], whether people can tell a feature contribution from a causal effect, and whether they can tell a risk band from a forecast horizon.

XI. CONCLUSION

This paper described a system that puts a calibrated attrition probability and a SHAP explanation on one page, supported by a shared prediction service. On the held-out test set of a fictional benchmark, the model finds 34 of 47 leavers, with a recall of 72.3% and a ROC-AUC of 0.805. It also flags 41 employees who stayed, and its accuracy sits below that of an always-stayed rule, so recall, precision and PR-AUC describe it better. The explanation shows which features push the modeled probability up or down. It should be read as an approximate association and not as a cause. Calibration, explanation stability and fairness remain to be tested, and the benchmark does not support claims of real-world forecasting accuracy or of retention effectiveness. Before any organizational use the system needs a consistent model-comparison protocol, external validation, access control and a fairness audit. Until then its outputs should be treated as evidence for careful human review.

REFERENCES

[1] S. Nosratabadi, R. K. Zahed, V. V. Ponkratov, and E. V. Kostyrin, "Artificial intelligence models and employee lifecycle management: A systematic literature review," 2022, arXiv:2209.07335.

[2] M. Karimi and K. S. Viliyani, "Employee turnover analysis using machine learning algorithms," 2024, arXiv:2402.03905.

[3] R. Yedida, R. Reddy, R. Vahi, R. Jana, A. GV, and D. Kulkarni, "Employee attrition prediction," 2018, arXiv:1806.10480.

[4] X. Ma, W. Liu, C. Zhao, and L. R. Tukhvatulina, "Can large language model predict employee attrition?" 2024, arXiv:2411.01353.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[6] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," 2017, arXiv:1706.04599.

[7] T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: A survey on how to assess and improve predicted class probabilities," 2021, arXiv:2112.10327.

[8] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017, arXiv:1705.07874.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," 2016, arXiv:1602.04938.

[10] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," 2017, arXiv:1711.00399.

[11] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," 2019, arXiv:1905.07697.

[12] I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. Friedler, "Problems with Shapley-value-based explanations as feature importance measures," 2020, arXiv:2002.11097.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

[13] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," 2018, arXiv:1811.10154.

[14] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," 2019, arXiv:1908.09635.

[15] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," 2016, arXiv:1610.02413.

[16] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, "Mitigating bias in algorithmic hiring: Evaluating claims and practices," 2019, arXiv:1906.09208.

[17] K. Mohiuddin et al., "Retention is all you need," 2023, arXiv:2304.03103.

[18] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," 2016, arXiv:1603.02754.

[19] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on tabular data?" 2022, arXiv:2207.08815.

[20] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," 2021, arXiv:2106.03253.

[21] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," 2021, arXiv:2110.01889.

[22] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," 2016, arXiv:1609.06570.

[23] S. M. Lundberg et al., "Explainable AI for trees: From local explanations to global understanding," 2019, arXiv:1905.04610.

[24] A. Barredo Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," 2019, arXiv:1910.10045.

[25] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, arXiv:1702.08608.

[26] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," 2018, arXiv:1802.01933.

[27] Z. C. Lipton, "The mythos of model interpretability," 2016, arXiv:1606.03490.

[28] U. Bhatt et al., "Explainable machine learning in deployment," 2019, arXiv:1909.06342.

[29] F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, J. W. Vaughan, and H. Wallach, "Manipulating and measuring model interpretability," 2018, arXiv:1802.07810.

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

(This work is licensed under a Creative Commons Attribution 4.0 International License.)