



CS610 Applied Machine Learning

Group 6 Project Final Report

Topic: IBM Employee Attrition Analysis

Cao Fei

Yang Tianyi

Lam Yu Hay Gladwin

Wang Yizhi

Yang Jingyuan

March 2023

1. Introduction

Employee attrition is a significant concern for organizations worldwide, as it can lead to high costs incurred in hiring and retraining new staff, damage to reputation, and disruption of operations. Human Resource (HR) managers often rely on instincts to identify employees at risk of leaving, which might not be accurate. The primary goal of this project is to develop a machine learning (ML) model that can help HR users quantitatively identify employees who are likely to leave the company and the common reasons.

2 . Objective

- Apply classification models to predict employee churn and the probability they will leave
- Identify common reasons that drive employees to leave using factor analysis.
- For employees predicted to leave, predict when they will leave

3. Data Analytics Workflow

We first conducted exploratory data analysis. (See appendix 2). Next, we performed the following workflows to build our models and to analyze data.

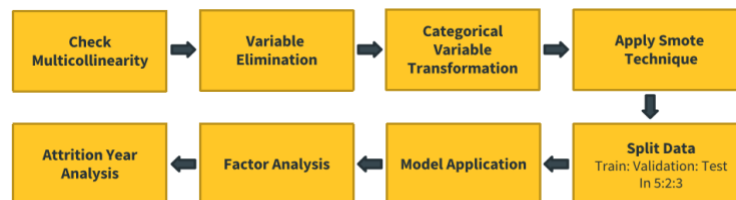


Figure 1: Data analytics workflow

4. Dataset and Manipulation

The dataset used is artificially created from Kaggle. (See appendix 1 for data dictionary) It contains 1,470 employees, with 35 columns of attributes related to demographics data, work experiences, work performances, and attrition status. The data is taken at a snapshot at a point in time. Each employee appears only once. Either they have left the company or are still in the company. We performed feature transformation, engineering and applied SMOTE to treat the imbalance class problem. We also performed model validation by splitting the dataset into training (50%), validation (20%), and test (30%).

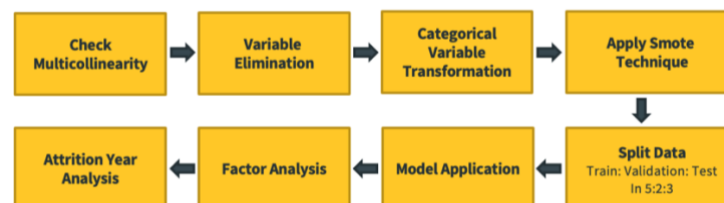


Figure 2: Data Manipulation workflow

5. Application of Machine Learning Models

We built various machine learning models to find out which was the best performer. These methods were shortlisted based on their ability to perform classification tasks, their interpretability, and their general performance in real-world applications.

5.1 Naive Bayes Classifier

Naïve Bayes is a classification algorithm based on Bayes' theorem that assumes independence between features, although it might not always be true in real-world datasets. We used the Gaussian Naïve Bayes algorithm that assumes the features are normally distributed. It is a good choice when working with continuous data. The conditional probabilities are calculated using the Gaussian probability density function:

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_{y,i}^2}} e^{\left(-\frac{(x_i - \mu_{y,i})^2}{2\sigma_{y,i}^2}\right)}$$

where, $\mu_{y,i}$ is the mean of the i -th feature for all samples in class y ; $\sigma_{y,i}$ is the standard deviation of the i -th feature for all samples in class y .

5.2 Logistic Regression

Logistic regression is a classification algorithm used to predict a binary outcome based on a set of independent variables. The sigmoid function is applied on the model coefficients, which represents the odds ratios of the input features.

$$p(1|\mathbf{x}_i) = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}_i}} \quad \text{logistic/sigmoid function} \quad \sigma(x) = \frac{1}{1 + e^{-x}}$$

Logistic regression is easy to implement, interpret, and very efficient to train. It makes no assumptions about distributions of classes in feature space. However, logistic regression may be prone to overfit if the number of observations is much less than the number of features.

5.3 Random Forest

Random Forest is an ensemble learning method that constructs a multitude of decision trees. Each decision tree is constructed using a subset of the available features and a random sample of the data. The trees are constructed using a process called recursive binary splitting, where the data is recursively split into two subsets using a feature and a threshold value, such that the resulting subsets are as pure as possible. To decide which feature to split on, the decision tree algorithm uses a metric called information gain or entropy. The feature with the highest information gain is chosen as the split feature. The entropy and information gain are defined as:

$$H(S) = - \sum_{i=1}^N p_i \log_2(p_i) \quad IG(S, A) = H(S) - \sum_{i=1}^k \frac{|S_i|}{|S|} H(S_i)$$

Where, S is a set of examples, N is the number of classes, p_i is the probability of randomly selecting an example in S that belong to class i ; k is the number of possible values of the feature A and $|S_i|$ is the number of examples in subset S_i .

Additionally, we used hyperparameter tuning to optimize model performance by using grid search to find the best combination of the number of trees, maximum depth, maximum number of features used, minimum number of samples required to split a node and the minimum number of samples in a node.

5.4 Support Vector Machine (SVM)

SVM identifies a hyperplane that can best segregate n-dimensional space into classes so that the new data point can be classified into a class. It is capable of modeling complex relationships between input features and the output using kernel function. The kernel we chose is linear and C equals 0.1.

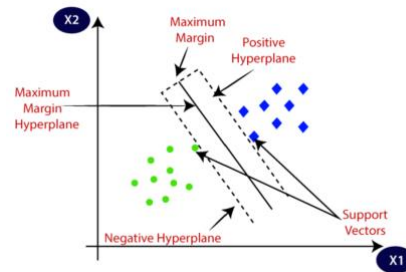


Figure 3: SVM Model

SVM provides good generalization performances even in high dimensional spaces. As for disadvantages, it is difficult to choose an optimal kernel and is slow to train relative to other models.

5.5 XGBoost

XGBoost's works by iteratively adding decision trees to an ensemble model, with each subsequent tree attempting to correct the errors of its predecessors. The model's final prediction is a weighted sum of the predictions of all the trees in the ensemble. It minimizes a regularized objective function that combines a loss function with a penalty term to prevent overfitting. The objective function is defined as:

$$L(y, \hat{y}) + \Omega(f) = \sum l(y_i, \hat{y}_i) + \Omega(f)$$

where $L(y, \hat{y})$ is the loss function, which measures the difference between the true labels y and the predicted labels \hat{y} , and $\Omega(f)$ is the penalty term, which discourages complex models. Penalty term is defined as:

$$\Omega(f) = \gamma \sum J + \frac{1}{2} \lambda \sum w^2$$

where γ and λ are hyperparameters that control the amount of shrinkage and regularization, respectively.

Gradient descent algorithm is used to optimize the objective function by iteratively updating the model's weights. Hyperparameter tuning was then done using sklearn's `gridsearchCV()` to find the best combination of maximum depth tree depth, learning rate, number of trees. XGBoost is a resilient and robust method that prevents and cubs over-fitting quite easily, and performs very well on medium, small, data with subgroups and structured datasets with not too many features. However, XGBoost does not perform so well on sparse and unstructured data.

5.6 Artificial Neural Network (ANN)

Neural networks are powerful and flexible models that can learn complex patterns in data. They automatically learn feature representations. However, neural networks can be computationally expensive and require large amounts of data to perform well. They may also be prone to overfitting and are less interpretable than simpler models. Using TensorFlow and Keras libraries, we built a multi-layer neural network with dense, batch normalization and dropout layers. For activation functions, we used ReLu and sigmoid functions.

ReLU Function: $f(x) = \max(0, x)$

Sigmoid

Function:

$$f(x) = \frac{1}{(1 + \exp(-x))}$$

A combination of L2 regularization and dropout was used to prevent overfitting. The model was then compiled using the 'adam' optimizer and the 'binary_crossentropy' loss function:

$$L(y, p) = -(y * \log(p) + (1 - y) * \log(1 - p))$$

where y is the true label and p is the predicted probability. L2 regularization adds a penalty term to the loss function, which is proportional to the squared magnitude of the model's weights.

$$L_2(w) = \lambda \cdot \sum w_i^2$$

where λ is the regularization strength and w_i are the model's weights. The model was trained for 50 epochs with a validation split of 0.2.

6 Evaluation Metrics

Evaluation metrics are used to evaluate the performance of each model.

F1 Score: It is the harmonic mean of precision and recall, providing a single metric that balances the trade-off between these two measures. It ranges from 0 (worst) to 1 (best):

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Recall: Recall, or sensitivity, is the proportion of true positive cases among the actual positive cases. It measures the ability of the model to identify all the positive instances correctly:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Accuracy: ratio of correct predictions to the total number of predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Confusion Matrix: shows the number of true positive, true negative, false positive, and false negative predictions made by the model. It provides a detailed view of the model's performance and helps identify areas for improvement.

Mean Squared Error (MSE): The average squared difference between the predicted and actual values, which can be used to evaluate the performance of classification models by comparing predicted probabilities with actual class labels:

$$MSE = \frac{1}{N} \cdot \sum (y_i - y'_i)^2$$

where N is the number of instances, y_i is the actual value, and y'_i is the predicted value.

R-squared (R²): A measure of how well the model's predictions fit the actual data. It ranges from 0 to 1, with higher values indicating better performance:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

where SS_{res} is the sum of squared residuals, and SS_{tot} is the total sum of squares.

ROC Curve: A graphical representation of the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity) at various threshold settings. It helps evaluate the performance of a classifier across a range of decision thresholds.

Area Under Curve (AUC): AUC measures the entire two-dimensional area underneath the ROC curve. It ranges from 0 to 1, with higher values indicating better performance. It summarizes the overall performance of a classifier across all possible decision thresholds.

7. Model Comparison and Chosen Model

Model Name	Accuracy Score	AUC	Precision	Recall	F-1 Score
Naïve Bayes	0.8378	0.9470	0.7934	0.9135	0.8492
Logistic Regression	0.9419	0.9752	0.9658	0.9162	0.9403
Random Forest	0.9270	0.9742	0.9675	0.8838	0.9237
SVM	0.8973	0.9497	0.9176	0.8730	0.8947
XGBoost	0.9176	0.9688	0.9612	0.8703	0.9135
Neural Network	0.9170	0.9588	0.9356	0.8934	0.9140

Figure 6: Model Comparison

Logistic regression (LR) model performed the best across all models. It has the highest accuracy score, AUC score, recall and F1-score. Although Random Forest has the best precision, LR is just slightly behind. As such, we choose the LR model to perform our employee attrition.

LR achieved an accuracy score of 0.9419 on the test set. This means that our model correctly classified 94.19% of employees as either leaving or not leaving the company based on their demographic and job-related factors. Its recall score of 0.9162 indicates that the model identified 91.62% of the true positive results correctly and the precision score of 0.9658 indicates that the model predicts 96.58% of the positive results correctly out of all the positive predictions.

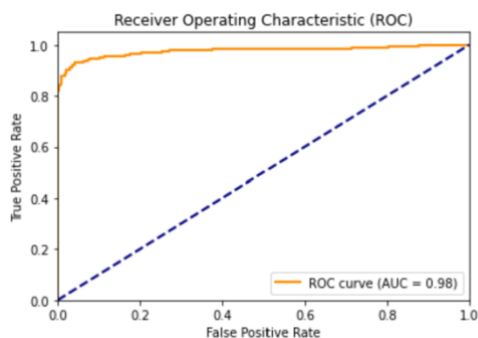


Figure 7: Confusion Matrix of Logistic Regression

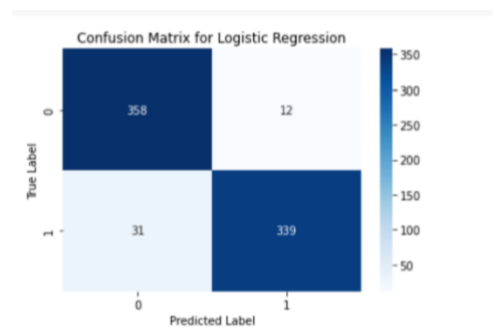


Figure 8: ROC Curve of Logistic Regression

The ROC curve starts at a high point on the y-axis, indicating that the model can identify true positive labels

with a high rate even with a very low false positive rate. The confusion matrix shows that the model can classify a large proportion of true positive and true negative results, which means the model has the capability of distinguishing positive and negative cases.

8. Factor Analysis using Logistic Regression

Employees who worked overtime were less likely to leave than those who did not. This observation may suggest that employees who work longer hours are more committed to their current job and have less time to explore alternative employment opportunities. Employees in job levels 2 (medium seniority) and 4 (director) were less likely to leave the company than those in job levels 1 (junior) and 3 (senior).

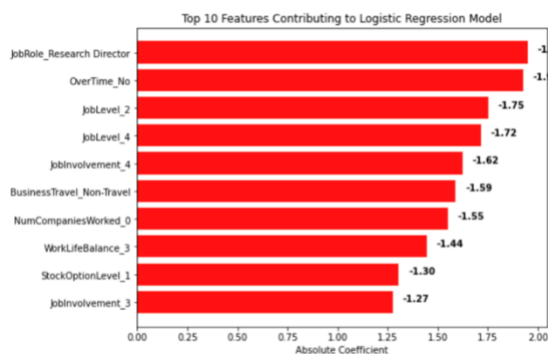


Figure 9: Coefficients of Top Ten Features

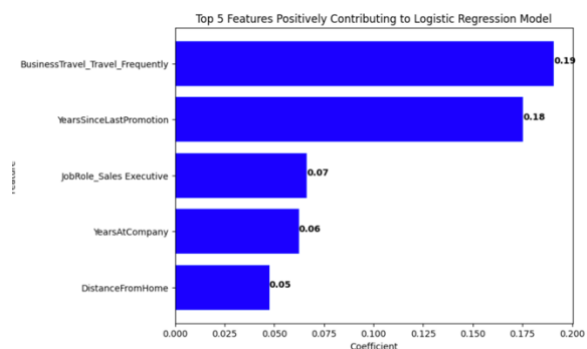


Figure 10: Coefficients of Top Five Features

Additionally, we explored variables that exhibited a positive correlation with employee attrition. The top five variables that positively contributed to our predictions were BusinessTravel_Travel_Frequently, YearsSinceLastPromotion, JobRole_Sales Executive, YearsAtCompany, and DistanceFromHome.

YearsSinceLastPromotion and DistanceFromHome stood out as noteworthy contributors to our model's predictions. Employees who feel that they are not progressing professionally or have limited opportunities for growth may seek out other job opportunities. Employees who have longer commutes may experience higher levels of job dissatisfaction due to factors such as increased stress, reduced leisure time, and decreased work-life balance from travelling. These findings are only a few of many possible contributors to employee attrition. We should consider all factors holistically.

9. Attrition Year Analysis



1:

Figure 11: Workflow of Attrition Year Analysis

With the employees predicted to leave, we predict if they will leave within the next 6 months. Due to lack of time variables, we are unable to perform time series analysis. Instead, we filtered out employees who attrited in our dataset and built a regression model with the “years at company” as the target variable. Originally, “years at company” variable refers to how long an employee has worked in the company. By filtering out employees who attributed, the “years at company” variable will now mean how long the

employee has stayed in the company before they left. With the predicted “years at company” values, we can work out the difference between the predicted years at company and current years at company. If the absolute difference is equal and below 0.5, the employee is flagged to leave within the next 6 months. We built the model using Adaboost regressor. In Adaboost, many weak learners are built. Each subsequent learner is trained to improve on the prediction errors caused by previous weak learners.

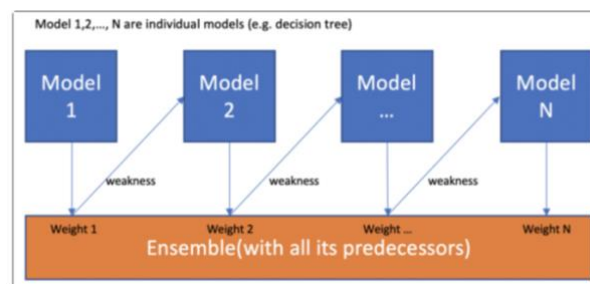


Figure 12: AdaBoost Process (NumPy Ninja. (2021, May 10))

We then performed hyperparameter tuning on the learning rate, loss function and number of trees using `GridSearchCV()` from the sklearn library. We arrive on the following result and optimal hyperparameters:

	Optimal Hyperparameters	R2	RMSE
AdaBoost Regressor	learning rate = 0.25, loss function = exponential, n_estimators = 80	0.879	1.741

Our model scored 87.9% R^2 and 1.741 RMSE on test set.

9.1. Segmentation Analysis

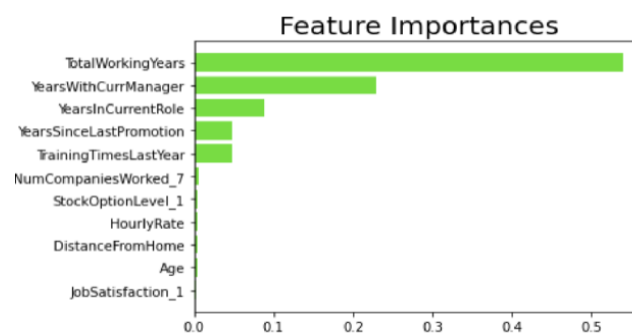


Figure13: Top 10 feature importance of Attrition Year analysis

Among those employees predicted to leave, we analyze the difference between the segment of employees predicted to leave within the next 6 months and those beyond the 6 months period. From the feature importance analysis, total working years is the most important feature followed by years with current manager, years in current role, years since last promotion, number of trainings received in past year etc. We also performed cluster analysis.

Class	Total Working Years	Years with current manager	Years in current role	Years since last promotion	Number of trainings last year	Num of Companies Worked (7)	Stock Option Level_1	Hourly Rate	Distance from home	Age	Job satisfaction
0	7.708	2.424	2.46	1.733	2.378	0.014	0.069	65.485	10.507	33.045	0.097
1	6.784	2.763	2.76	1.456	2.15	0.003	0.073	63.815	9.819	32.108	0.164

Employees who are likely to leave within the next 6 months have shorter total working years' experience,

spend longer time with their current manager, and in their current role. They have less training, worked in a lesser number of companies before with lower hourly rate and age. This is in spite of having higher stock options, job satisfaction and working closer to home. (See appendix 4 for parallel coordinates plot)

10. Model Deployment

To help human resource (HR) department easily use the model, the model can be deployed into a pipeline. HR business users will just need to upload an employee list and the model will output a list of employees predicted to leave and whether they will leave within the next 6 months. This can be seen by the workflow below:

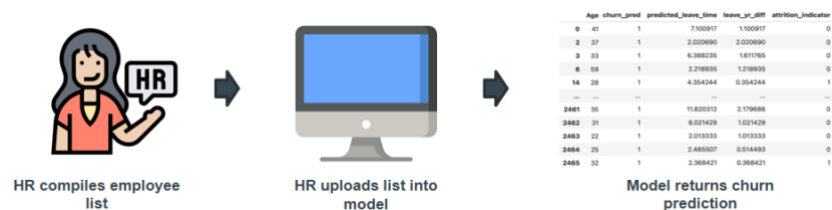


Figure 14: Model deployment pipeline for HR business user

11. Conclusion and Future Work

We propose the below as future work to further improve our model based on existing issues.

Issues	Potential Solution
Optimizing Machine learning model	<ul style="list-style-type: none"> Use stacking to combines different models with different weightage to improve performance and stability
Overfit prevention	<ul style="list-style-type: none"> Apply regularization techniques to penalize model complexity to reduce overfitting for better performance
Small and imbalanced dataset	<ul style="list-style-type: none"> Collect more data, especially on minority class Using SMOTE technique to treat imbalance class might not be able to capture the actual distribution of the variables, which may impact model performance on real world data
Biased data sources	<ul style="list-style-type: none"> Collecting data anonymously as Hawthorne effect might cause bias data collection of 'environment satisfaction' and 'job satisfaction' data. Resignation could be voluntarily or involuntarily. Collecting this additional data field will allow us to analyse the segments separately to avoid Simpson's paradox.
Missing time data	<ul style="list-style-type: none"> Collecting datetime data of each resignation will allow us to perform time series analysis on which month of the year has the most resignation

In conclusion, our model provides a solution for HR users to predict which employees would leave and when they would leave. This allows HR users to take timely preventive measures to retain employees. Our factor analysis from our model can also help HR to identify areas and company policies to improve on to improve attrition. Despite so, HR users should use this data ethically. They need to ensure there is robust employee data governance in place in order to maintain employees' trust in how their data is being used.

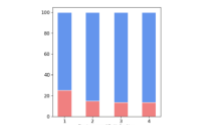
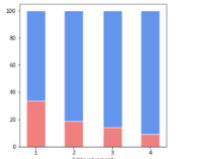
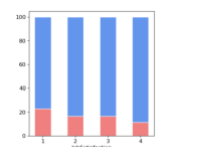
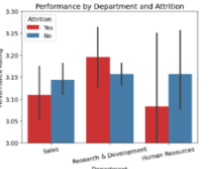
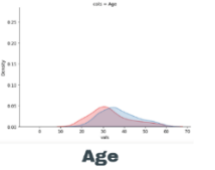

11. Appendix

Appendix 1: Dataset Data Dictionary

Name	Description
AGE	Age of employee
Attrition	Employee has left the company or not (0=no, 1=yes)
Business Travel	Categorical Data: No Travel, Travel Frequently, Travel Rarely
Daily Rate	Numerical Value for Salary
Department	Categorical Data. The department that employees affiliated with.
Distance from Home	Numerical Data. The distance from work to home.
Education	Education Level (1 = Below College, 2 = College, 3 = Bachelor, 4 = Master, 5 = Doctor)
Education Field	Field of Education
Employee Count	Employee Count
Employee Number	Employee Number
Environment Satisfaction	Work Environment Satisfaction Level (1 = Low, 2 = Medium, 3 = High, 4 = Very High)
Gender	Gender of employee
Hourly Rate	Hourly Salary
Job Involvement	Job Involvement Level (1 = Low, 2 = Medium, 3 = High, 4 = Very High)
Job Level	Responsibility level within the company (1 = New, 2 = Junior, 3 = Senior, 4 = Director, 5 = Manager)
Job Role	Position within the company
Job Satisfaction	Job Satisfaction Level (1 = Low, 2 = Medium, 3 = High, 4 = Very High)
Marital Status	Marital status of the employee
Monthly Income	Net pay per month
Monthly Rate	Gross pay per month
Number of Companies Worked	Number of Companies worked at
Over18	Whether the employee is above 18 years of age or not
Overtime	Whether overtime or not
Percent Salary Hike	Percent salary hike for last year
Performance Rating	Performance rating for last year (1 = Low, 2 = Good, 3 = Excellent, 4 = Outstanding)
Relationship Satisfaction	Relationship satisfaction level (1 = Low, 2 = Medium, 3 = High, 4 = Very High)
Standard Hours	Standard hours of work for the employee
Stock Option Level	Level of the stock option grant contracted at the time of employment (0 = None, 1 = Low, 2 = High, 3 = Very High)
Total Working Years	Total number of years the employee has worked so far
Training Times Last Year	Number of times training was conducted for this employee last year
Work Life Balance	Work life balance level (1 = Bad, 2 = Good, 3 = Better, 4 = Best)
Years at Company	Total number of years spent at the company by the employee
Years in Current Role	Number of years in current role
Years since Last Promotion	Number of years since last promotion

Years with Current Manager	Number of years under current manager
----------------------------	---------------------------------------

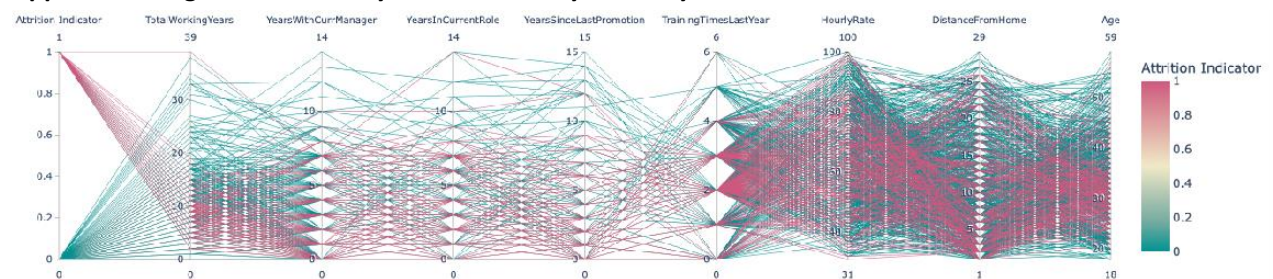
Appendix 2: Exploratory Data Analysis of Dataset

 <p>Environment Satisfaction</p>	<ul style="list-style-type: none"> Employees who are less satisfied with the environment tend to leave the company.
 <p>Job Involvement</p>	<ul style="list-style-type: none"> Employees who are less involved in the job tend to leave the company.
 <p>Job Satisfaction</p>	<ul style="list-style-type: none"> Employees who are less satisfied with their jobs tend to leave the company.
 <p>By Department</p>	<ul style="list-style-type: none"> R&D departments who have high performance ratings tend to leave the company compared to other departments. Maybe these people are vulnerable to poaching by other companies.
 <p>Age</p>	<ul style="list-style-type: none"> Younger employees tend to leave the company compared to older employees.
 <p>Monthly Income</p>	<ul style="list-style-type: none"> Employees who earn less income tend to leave the company.

Appendix 3: XG Boost results

The XGBoost model was initialized with hyperparameters `n_estimators=100`, `max_depth=3`, `learning_rate=0.1`, and `random_state=1234`. The initial model achieved a promising 92.16% accuracy in correctly classifying employees as staying or leaving based on the given independent variables. To improve performance, we conducted hyperparameter tuning through grid search, finding that an updated `max_depth` of 5 improved the model's performance. However, the updated model with tuned hyperparameters showed slightly lower precision and accuracy scores compared to the original model with default hyperparameters. Ultimately, we decided to use the original model with default hyperparameters for our specific use case, considering the trade-offs between precision, recall, and accuracy and the context and application of our model. It's crucial to consider the specific use case and application context when selecting and tuning a model's hyperparameters.

Appendix 4. Segmentation analysis for Attrition year Analysis



12. References

Javatpoint. Support Vector Machine Algorithm. Retrieved April 4, 2023, from [Support Vector Machine \(SVM\) Algorithm - Javatpoint](#)

NumPy Ninja. (2021, May 10). Understanding AdaBoost. NumPy Ninja.
<https://www.numpyninja.com/post/understanding-adaboost>

Praisidio. "Employee Turnover Rates by Industry, Location, & Role in 2022." 2023,
<https://www.praisidio.com/turnover-rates>. Accessed 2 Mar. 2023

Plotly. (n.d.). Parallel coordinates plot. Retrieved April 4, 2023, from <https://plotly.com/python/parallel-coordinates-plot/>

Smith, John, et al. "IBM Employee Attrition Dataset." Kaggle, Kaggle, 2018,
<https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset>