Employee Attrition

Talent Retention Taskforce
Debanjan Nanda , Ayan Maity
nandadebanjan@gmail.com , ayanmaity813@gmail.com

May 1, 2024

Abstract

Employee attrition means when employees leave a company because of different reasons. This causes big problems for the company. In 2021, 57.3% of employees left their jobs. In this project we found some causes of Employee Attrition and made some frameworks to predicts employee attrition. We have used several Machine Learning techniques and got the best accuracy in Random Forest and XGBoost. We used Exploratory Data Analysis (EDA) to figure out why employees leave their jobs. Our project showed that how much money employees make each month, their hourly pay rate, their job level, and their age factors are the main reasons why they leave. Our approach may help the organizations to overcome employee attrition by improving the factors that cause attrition.

1 Introduction

1.1 What?

1. This project aims to predict whether employees will leave their jobs.
2. We'll gather data on various employee attributes such as age, job role, satisfaction levels, and tenure to train predictive models.
3. These models will utilize advanced machine learning techniques, including classification to make accurate predictions.
4. By analyzing historical data, we'll teach the models to identify patterns associated with turnover and tenure.
5. Ultimately, the goal is to develop reliable tools that assist companies in understanding and managing their workforce dynamics effectively.

1.2 Why?

1. Employee attrition projects are essential for companies to understand why employees are leaving their jobs.

2. Understanding the reasons behind employee departures allows companies to implement changes that encourage employee retention.

3. By reducing employee turnover, companies can save money that would otherwise be spent on hiring and training new employees.

4. Longer employee tenure leads to increased skill and expertise, benefiting the company's overall productivity and success.

5. Ultimately, decreasing turnover contributes to a more positive and stable work environment, improving morale and overall workplace satisfaction.

1.3 How?

1. The project will employ advanced machine learning techniques to analyze the collected data and develop predictive models.

2. classification models like logistic regression, decision trees, random forest can be utilized to predict employee attrition.

2 Literature review

Author	Paper	Data	Method	Result
Ali Raza, Kashif Munir, Mubarak Almutairi, Faizan Younas and Mian Muhammad Sadiq Fareed	Predicting Employee Attrition Using Machine Learning	IBM HR attrition dataset: https://shorturl.at/ciqu2	Four machine learning models—Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree Classifier (DTC), and Extra Trees Classifier (ETC)—were employed for prediction.	The study evaluated the models based on various metrics such as accuracy, precision, recall, F1 score, and ROC curve. The ETC model demonstrated superior performance across these metrics. This model achieved the highest accuracy score of 93%.
By Alao D. & Adeyemo A. B.	ANALYZING EMPLOYEE ATTRITION USING DECISION TREE ALGORITHMS	Employees in a Higher Institution in South-West Nigeria: https://iiste.org/Journals/index.php/CIS/article/view/10148	The authors employed classification techniques, specifically decision tree algorithms, to develop prediction models for employee attrition.	The performance of the developed models was assessed using various performance metrics such as True Positive Rate (TP Rate), False Positive Rate (FP Rate), Precision, F-Measure, and Receiver Operating Characteristic (ROC) curve. The decision tree model achieved the best accuracy level of almost 75%.

3 Proposed methodology

At first we have verified that there are no missing values in the dataset. Then we converted categorical variables like "Attrition," "Over18," and "OverTime" into numerical representations suitable for modeling.

We visualized the distribution of the target variable ("Attrition") using a count plot, showing the number of employees who stayed and left the company. Then we plotted histograms for numerical features like age, daily rate, and distance from home to understand their distributions and their relationship with attrition. Later we removed unnecessary columns like "EmployeeCount," "StandardHours," "Over18," and "EmployeeNumber," which likely wouldn't

contribute to the model's predictive power.

Figure 1: histrogram for numerical variables

Then we analyzed the relationship between attrition and various features such as age, daily rate, distance from home, job role, marital status, job involvement, and job level and visualized these relationships using histograms, count plots, and box plots to identify potential patterns or trends.

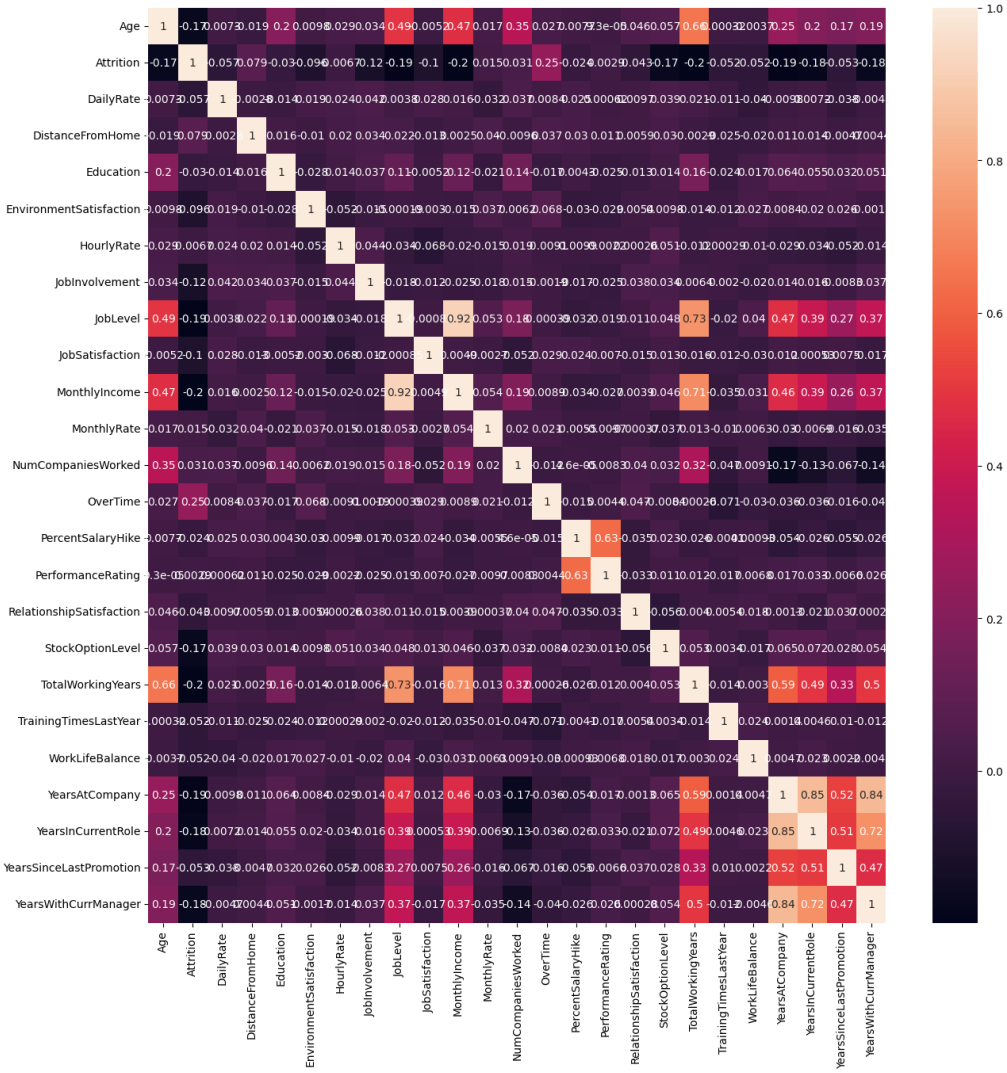

Figure 2: Correlation Heatmap

We have prepared the data for modeling by encoding categorical variables using one-hot encoding, scaling numerical features, and splitting the data into training and testing sets. We have selected categorical features ('BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole', 'MaritalStatus') and applied one-hot encoding to convert them into numerical representations suitable for modeling. The OneHotEncoder() from scikit-learn's preprocessing module is used for this purpose. The resulting encoded categorical features are concatenated into a Dataframe.

Since this is an imbalance dataset , we computed several classification models before balancing the dataset and after balancing the dataset. Then we did hyperparameter tuning for each model and got the best accuracy 90% in Xgboost and Random Forest Classifier.

Before balancing the dataset we have imported Logistic Regression from scikit-learn and trained

using the training data using the fit() method. After training, we've evaluated the model's performance on the test data using the score() method, which calculates the accuracy of the model. Then we computed the Confusion matrix to evaluate the performance of the classifier. It shows the count of true negatives, false positives, false negatives, and true positives. The confusion matrix is visualized using a heatmap to provide a clear representation of the model's performance in terms of correct and incorrect predictions. Then we generated Classification report which provides precision, recall, F1-score, and support for each class, along with averages (weighted and macro) across all classes. The output shows the accuracy of the logistic regression model, which is approximately 91.30% indicating that the logistic regression model performs reasonably well in predicting employee attrition. However, looking at the classification report, the precision, recall, and F1-score for the minority class (attrition = 1) are relatively lower compared to the majority class (attrition = 0). This suggests that the model may not perform well in identifying instances of attrition accurately, especially for the minority class.

Similarly we have executed these steps for other classification models like Decision Trees, Random Forest, Gradient Boosting, Xgboost, Exra Trees Classifier and Support Vector Machine. We have used SMOTE, Random Over-Sampling and SMOTETomek algorithms to balance the dataset. SMOTE generates synthetic samples for the minority class by interpolating between existing minority class samples. Random over-sampling randomly duplicates samples from the minority class until the class distribution is balanced. SMOTETomek is a combination of SMOTE and Tomek links. It first applies SMOTE to oversample the minority class and then removes Tomek links, which are pairs of samples (one from the majority class and one from the minority class) that are close to each other but of opposite classes.

After Balancing the class , again we have computed those models on the balanced dataset. Then we did hyperparameter tuning and got the best accuracy of 90% in Random Forest and Xgboost. We have used SMOTE, Random Over-Sampling and SMOTETomek algorithms to balance the dataset. SMOTE generates synthetic samples for the minority class by interpolating between existing minority class samples. Random over-sampling randomly duplicates samples from the minority class until the class distribution is balanced. SMOTETomek is a combination of SMOTE and Tomek links. It first applies SMOTE to oversample the minority class and then removes Tomek links, which are pairs of samples (one from the majority class and one from the minority class) that are close to each other but of opposite classes.

After Balancing the class , again we have computed those models on the balanced dataset. Then we did hyperparameter tuning and got the best accuracy of 90% in Random Forest and Xgboost.

4 Experimental result

4.1 Used Models:

Before balancing the dataset, the logistic regression model demonstrated a precision of 0.92 and a recall of 0.98, resulting in an F1 score of 0.81 for class 0. For class 1, the precision was 0.82, recall was 0.46, and F1 score was 0.59. The model achieved an accuracy of 91%.

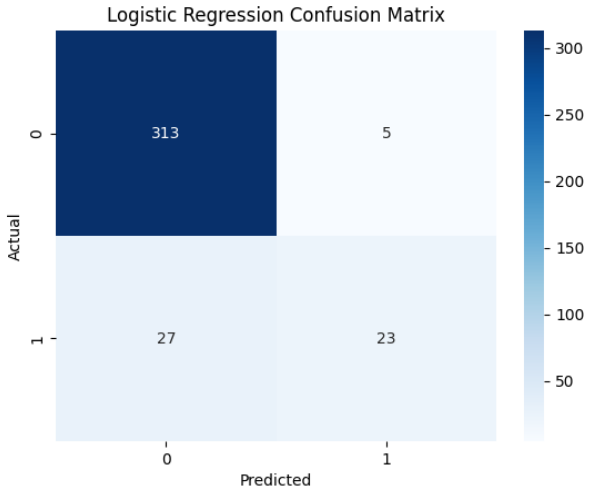

Figure 3: Logistic Regression

Decision Tree demonstrated a precision of 0.91 and a recall of 0.88, resulting in an F1 score of 0.89 for class 0. For class 1, the precision was 0.36, recall was 0.42, and F1 score was 0.39. The model achieved an accuracy of 82%.

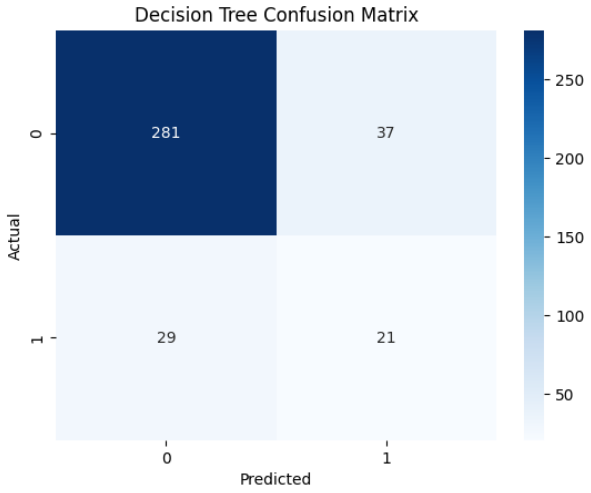

Figure 4: Decision Trees

Random Forest demonstrated a precision of 0.89 and a recall of 0.99, resulting in an F1 score of 0.94 for class 0. For class 1, the precision was 0.82, recall was 0.18, and F1 score was 0.30 . The model achieved an accuracy of 88%.

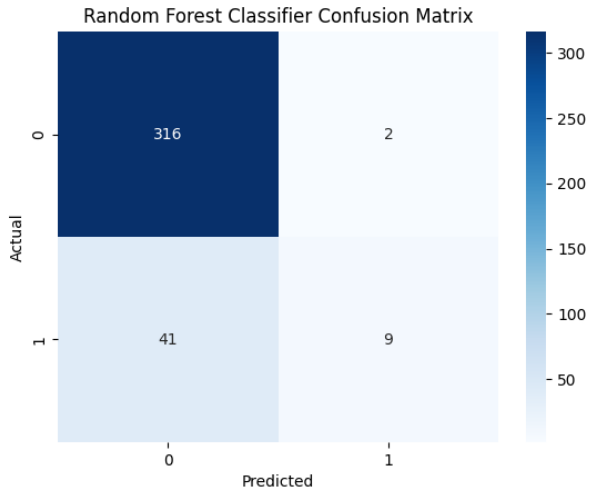

Figure 5: Random forest

Gradient Boosting demonstrated a precision of 0.91 and a recall of 0.97, resulting in an F1 score of 0.94 for class 0. For class 1, the precision was 0.72, recall was 0.42, and F1 score was 0.53. The model achieved an accuracy of 90%.

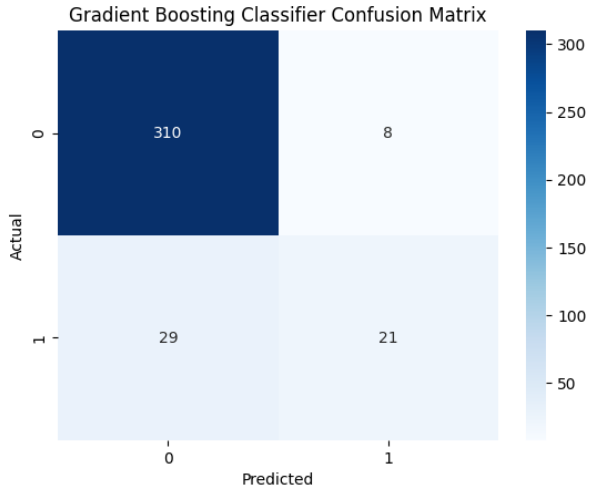

Figure 6: Gradient Boosting

XGBoost demonstrated a precision of 0.92 and a recall of 0.95, resulting in an F1 score of 0.93 for class 0. For class 1, the precision was 0.59, recall was 0.44, and F1 score was 0.51. The model achieved an accuracy of 88%.

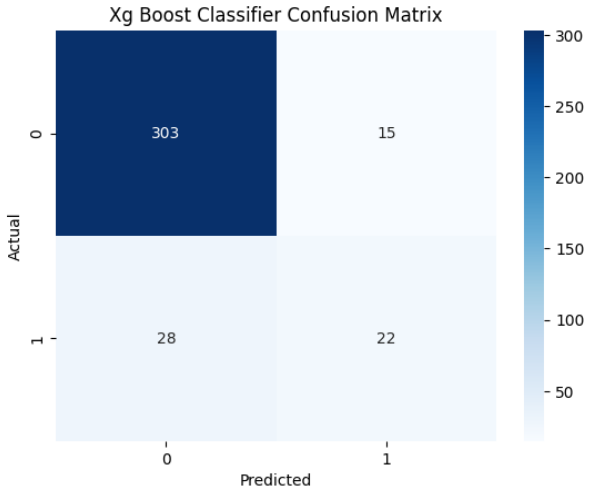

Figure 7: Xgboost

Extra tree classifier demonstrated a precision of 0.89 and a recall of 0.99, resulting in an F1 score of 0.94 for class 0. For class 1, the precision was 0.73, recall was 0.22, and F1 score was 0.34. The model achieved an accuracy of 88%.

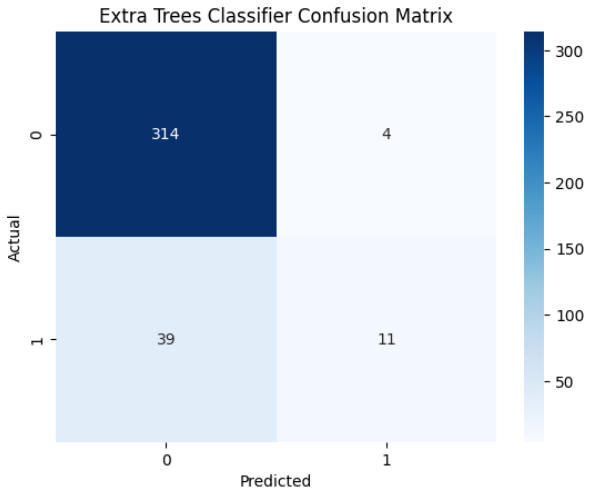

Figure 8: Extra Trees Classifier

Support Vector Machine demonstrated a precision of 0.90 and a recall of 0.99, resulting in an F1 score of 0.94 for class 0. For class 1, the precision was 0.87, recall was 0.26, and F1 score was 0.40. The model achieved an accuracy of 89%.

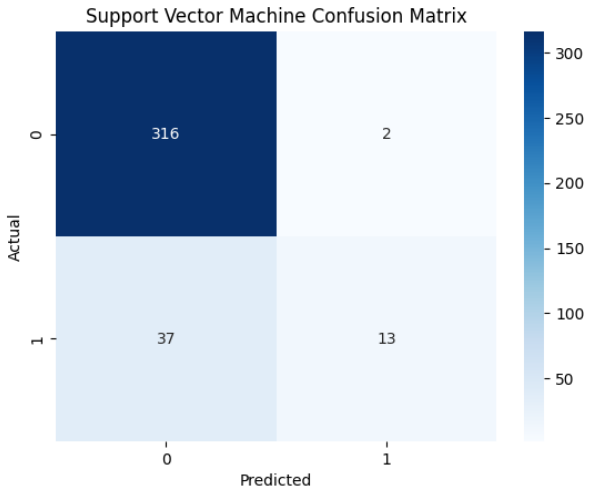

Figure 9: SVM

After balancing the dataset, the logistic regression model demonstrated a precision of 0.95 and a recall of 0.76, resulting in an F1 score of 0.84 for class 0. For class 1, the precision was 0.33, recall was 0.76, and F1 score was 0.46. The model achieved an accuracy of 76%.

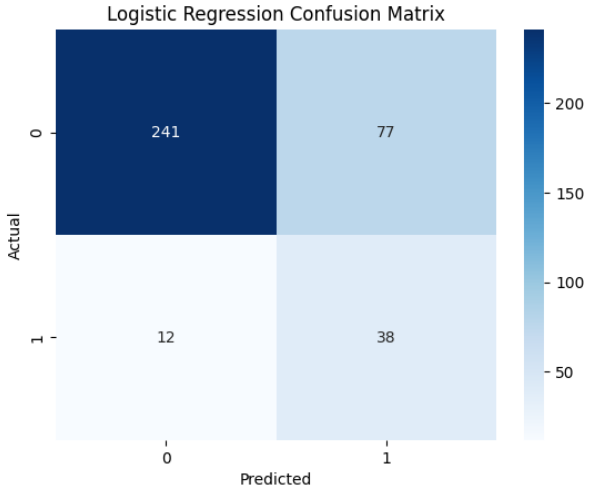

Figure 10: Logistic Regression

Decision Tree demonstrated a precision of 0.89 and a recall of 0.81, resulting in an F1 score of 0.85 for class 0. For class 1, the precision was 0.24, recall was 0.38, and F1 score was 0.29. The model achieved an accuracy of 75%.

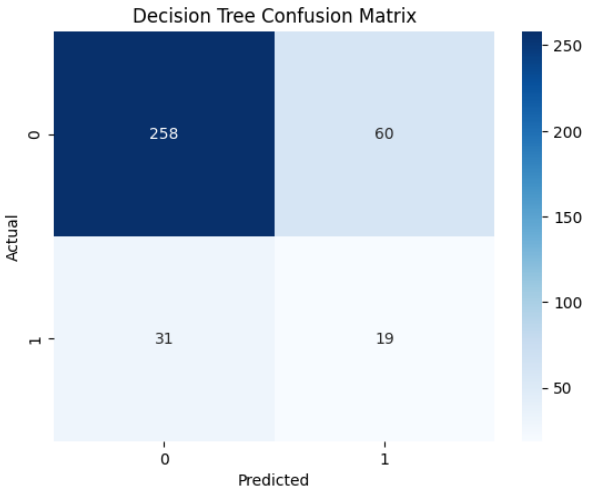

Figure 11: Decision Trees

Random Forest demonstrated a precision of 0.91 and a recall of 0.97, resulting in an F1 score of 0.94 for class 0. For class 1, the precision was 0.69, recall was 0.36, and F1 score was 0.47 . The model achieved an accuracy of 89%.

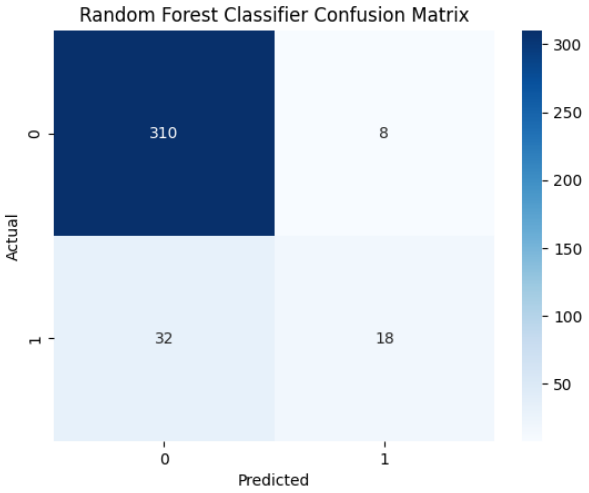

Figure 12: Random Forest

Gradient Boosting demonstrated a precision of 0.92 and a recall of 0.96, resulting in an F1 score of 0.94 for class 0. For class 1, the precision was 0.66, recall was 0.50, and F1 score was 0.57. The model achieved an accuracy of 90%.

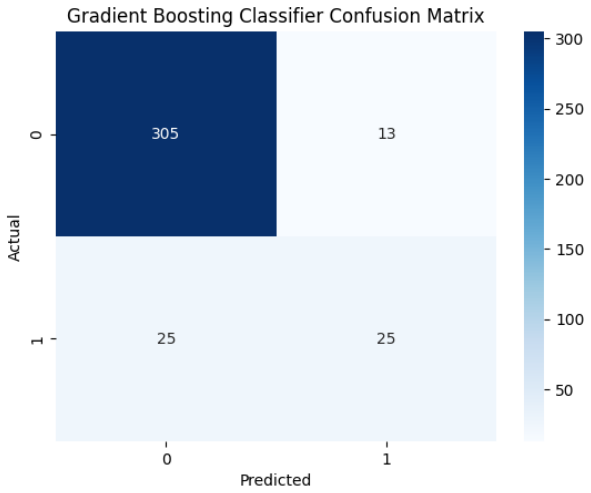

Figure 13: Gradient Boosting

XGBoost demonstrated a precision of 0.92 and a recall of 0.94, resulting in an F1 score of 0.93 for class 0. For class 1, the precision was 0.57, recall was 0.48, and F1 score was 0.52. The model achieved an accuracy of 88%.

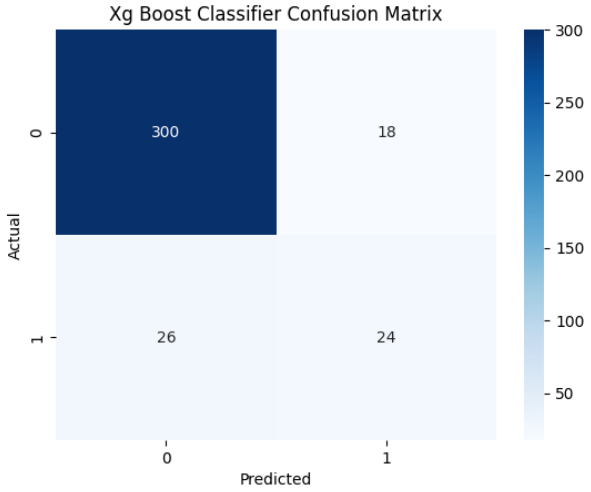

Figure 14: Xgboost

Extra tree classifier demonstrated a precision of 0.91 and a recall of 0.96, resulting in an F1 score of 0.94 for class 0. For class 1, the precision was 0.64, recall was 0.42, and F1 score was 0.51. The model achieved an accuracy of 89%.

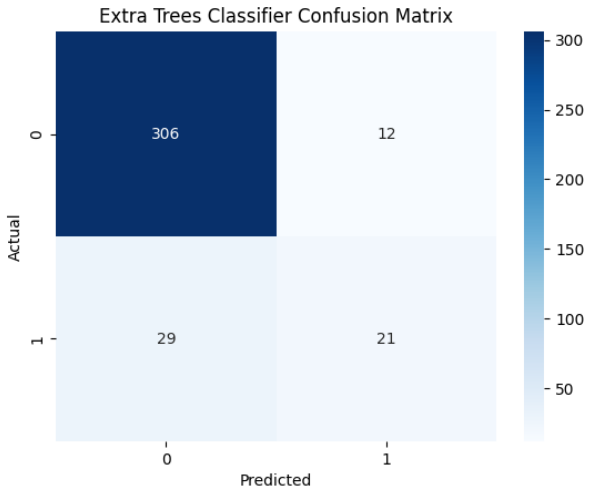

Figure 15: Extra Trees Classifier

Support Vector Machine demonstrated a precision of 0.92and a recall of 0.89, resulting in an F1 score of 0.91 for class 0. For class 1, the precision was 0.43, recall was 0.54, and F1 score was 0.48. The model achieved an accuracy of 84%.

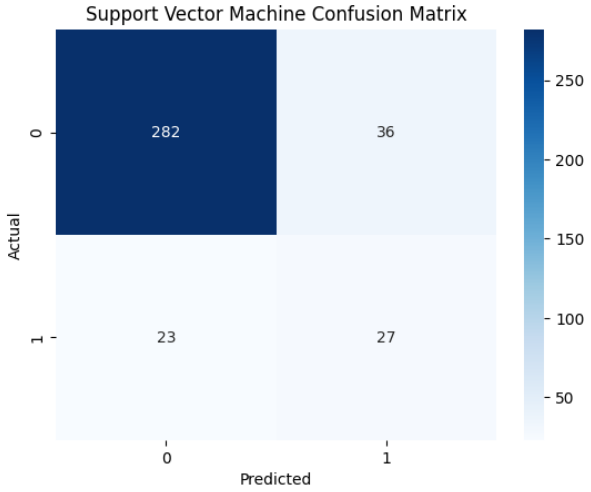

Figure 16: SVM

After Hyperparameter tuning, the logistic regression model demonstrated a precision of 0.95 and a recall of 0.76, resulting in an F1 score of 0.84 for class 0. For class 1, the precision was 0.33, recall was 0.76, and F1 score was 0.48. The model achieved an accuracy of 76%.

Decision Tree demonstrated a precision of 0.89 and a recall of 0.93, resulting in an F1 score of 0.91 for class 0. For class 1, the precision was 0.41, recall was 0.30, and F1 score was 0.34. The model achieved an accuracy of 85%.

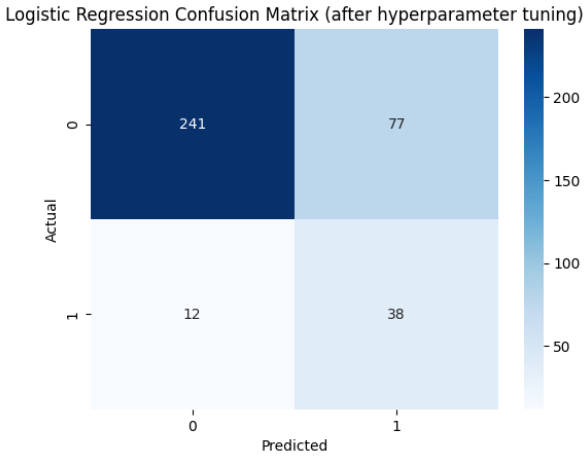

Figure 17: Logistic Regression

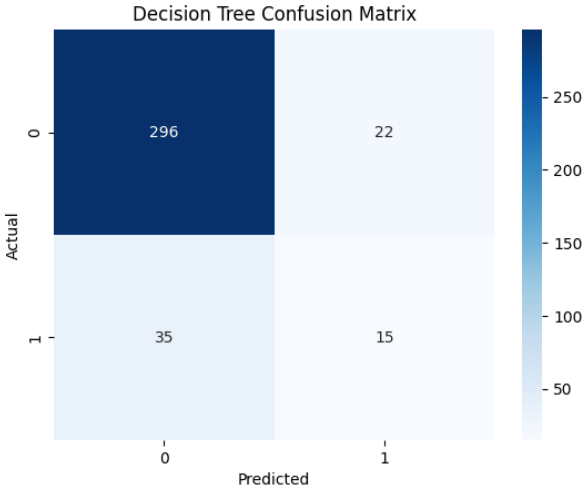

Figure 18: Decision Trees

Random Forest demonstrated a precision of 0.91 and a recall of 0.98, resulting in an F1 score of 0.94 for class 0. For class 1, the precision was 0.75, recall was 0.42, and F1 score was 0.54 . The model achieved an accuracy of 90%.
Gradient Boosting demonstrated a precision of 0.94 and a recall of 0.96, resulting in an F1 score of 0.95 for class 0. For class 1, the precision was 0.70, recall was 0.60, and F1 score was 0.65. The model achieved an accuracy of 88%.
XGBoost demonstrated a precision of 0.92 and a recall of 0.96, resulting in an F1 score of 0.94 for class 0. For class 1, the precision was 0.64, recall was 0.50, and F1 score was 0.56. The model achieved an accuracy of 90%.

Extra tree classifier demonstrated a precision of 0.91 and a recall of 0.96, resulting in an F1 score of 0.93 for class 0. For class 1, the precision was 0.61, recall was 0.38, and F1 score was 0.47. The model achieved an accuracy of 88%.

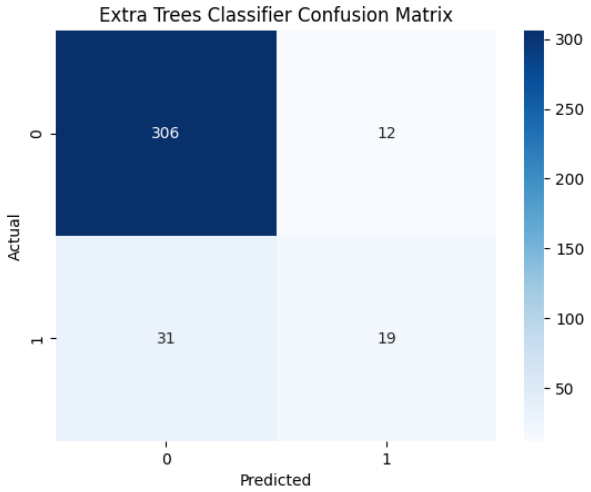

Figure 19: Extra Trees Classifier

Support Vector Machine demonstrated a precision of 0.87and a recall of 0.99, resulting in an F1 score of 0.92 for class 0. For class 1, the precision was 0.43, recall was 0.06, and F1 score was 0.11. The model achieved an accuracy of 86%.

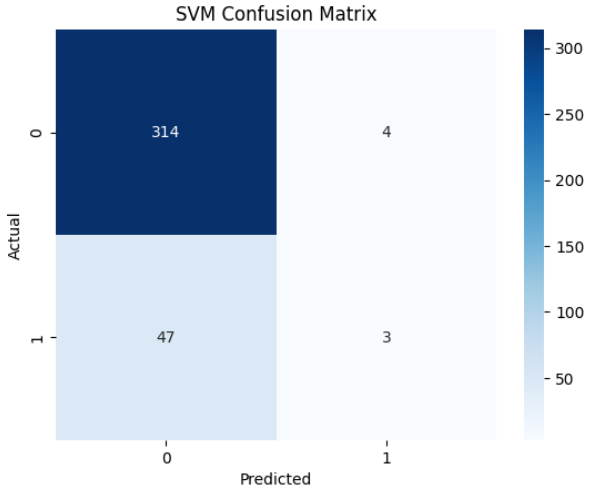

Figure 20: SVM

4.2 Dataset:

We have used IBM HR Analytics dataset for our project work. The link for the dataset is given below:`https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset`

4.3 Performance Metric

4.3.1 Accuracy

Accuracy measures the proportion of correctly classified instances among all instances, calculated by dividing the number of correct classifications by the total instances.

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

4.3.2 Precision

Precision quantifies the number of correctly predicted positive cases out of all instances predicted as positive. It is calculated by dividing the number of true positives by the sum of true positives and false positives.

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

4.3.3 Recall

Recall measures the proportion of correctly predicted positive cases out of all actual positive cases. It is calculated by dividing the number of true positives by the sum of true positives and false negatives.

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

4.3.4 F1 Score

The F1 score combines both precision and recall into a single metric by taking their harmonic mean. It provides a balance between precision and recall.

$$\text{F1 Score} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

4.4 Time Complexity

- Logistic Regression:
  - Training: $O(n_{\text{features}} \times n_{\text{samples}} \times n_{\text{iterations}})$
  - Prediction: $O(n_{\text{features}} \times n_{\text{samples}})$
- Support Vector Machine (SVM):
  - Training: $O(n_{\text{samples}}^2 \times n_{\text{features}})$ to $O(n_{\text{samples}}^3 \times n_{\text{features}})$
  - Prediction: $O(n_{\text{support vectors}} \times n_{\text{features}})$
- Decision Tree:
  - Training: $O(n_{\text{samples}} \times n_{\text{features}} \times \log(n_{\text{samples}}))$ to $O(n_{\text{samples}} \times n_{\text{features}}^2)$
  - Prediction: $O(\log(n_{\text{samples}}))$
- Random Forest and Extra Trees Classifier:
  - Training: $O(n_{\text{trees}} \times n_{\text{samples}} \times n_{\text{features}} \times \log(n_{\text{samples}}))$
  - Prediction: $O(n_{\text{trees}} \times \log(n_{\text{samples}}))$
- XGBoost and Gradient Boosting:
  - Training: $O(n_{\text{trees}} \times n_{\text{samples}} \times n_{\text{features}})$
  - Prediction: $O(n_{\text{trees}} \times n_{\text{features}})$

We are giving the results in the following table :

	Class-0			**Class-1**			**Accuracy**
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
Logistic Regression	0.92	0.98	0.95	0.82	0.46	0.59	91
Decision Tree	0.91	0.88	0.89	0.36	0.42	0.39	82
Random Forest	0.89	0.99	0.94	0.82	0.18	0.30	88
Gradient Boosting	0.91	0.97	0.94	0.72	0.42	0.53	90
XGBoost	0.92	0.95	0.93	0.59	0.44	0.51	88
Extra Trees Classifier	0.89	0.99	0.94	0.73	0.22	0.34	88
SVM	0.90	0.99	0.94	0.87	0.26	0.40	89

Table 1: Before Balancing the classes

	Class-0			Class-1			Accuracy
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
Logistic Regression	0.95	0.76	0.84	0.33	0.76	0.46	76
Decision Tree	0.89	0.81	0.85	0.24	0.38	0.29	75
Random Forest	0.91	0.97	0.94	0.69	0.36	0.47	89
Gradient Boosting	0.92	0.96	0.94	0.66	0.50	0.57	90
XGBoost	0.92	0.94	0.93	0.57	0.48	0.52	88
Extra Trees Classifier	0.91	0.96	0.94	0.64	0.42	0.51	89
SVM	0.92	0.89	0.91	0.43	0.54	0.48	84

Table 2: After Balancing the classes

	Class-0			Class-1			Accuracy
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
Logistic Regression	0.95	0.76	0.84	0.33	0.76	0.46	76
Decision Tree	0.89	0.93	0.91	0.41	0.30	0.34	85
Random Forest	0.91	0.98	0.94	0.75	0.42	0.54	90
Gradient Boosting	0.94	0.96	0.95	0.70	0.60	0.65	88
XGBoost	0.92	0.96	0.94	0.64	0.50	0.56	90
Extra Trees Classifier	0.91	0.96	0.93	0.61	0.38	0.47	88
SVM	0.87	0.99	0.92	0.43	0.06	0.11	86

Table 3: After Hyperparameter Tuning

5 Summary

The employee attrition project delves into understanding the underlying causes of employee turnover within an organization. It involves analyzing various factors such as job satisfaction, salary, work environment, career growth opportunities, and employee demographics. By leveraging data analytics and machine learning methodologies, patterns and trends in employee attrition are identified, providing valuable insights for HR professionals and management. These insights help in developing proactive strategies to mitigate attrition risks and enhance employee retention.

Through predictive modeling techniques, the project aims to forecast which employees are most likely to leave the organization, allowing HR teams to intervene with targeted interventions such as personalized training, career development programs, or adjustments to compensation and benefits. Additionally, the project may uncover systemic issues within the organization that contribute to high turnover rates, enabling management to address underlying concerns and foster a more conducive work environment.

Ultimately, the goal of the employee attrition project is to empower organizations with actionable insights to reduce turnover, retain top talent, and foster a culture of employee engagement and satisfaction, thereby improving overall organizational performance and success.

References

[1] https://www.apollotechnical.com/employee-retention-statistics/.

[2] Dr B Merceline Anitha. A study on employee attrition and retention with reference to evron impex. https://www.researchgate.net/publication/370471266_A_STUDY_ON_EMPLOYEE_ATTRITION_AND_RETENTION_WITH_REFERENCE_TO_EVRON_IMPEX.

[3] Alao D. Adeyemo A. B. Analyzing employee attrition using decision tree algorithms. https://core.ac.uk/download/pdf/234697248.pdf.

[4] Ali Raza, Kashif Munir, Mubarak Almutairi, Faizan Younas, and Mian Muhammad Sadiq Fareed. Title of the article. *Applied Sciences*, 12(13):6424.