Project Report

On

"EMPLOYEE ATTRITION PREDICTION"

Submitted
By

Mahima Bhushan

Net ID: mxb240023

Master of Science in Computer Science

Erik Jonsson School of Engineering & Computer Science

The University of Texas at Dallas

Academic Term: Fall 2025

Course Instructor: ***Rishabh Iyer***

ABSTRACT

The current trend we are seeing in the Software Industries is that there are a lot of employees resigning from their position for various reasons like better opportunities, pay, work culture etc. A company after training an employee and once he becomes skilled and valuable, if he decides to leave, this poses a huge loss for the company and the company faces many challenges to fill in that resource. This impacts the company directly in productivity, hiring cost and long-term growth. If the company can predict the attrition early, they can come up with retention strategies to engage the at-risk employees. Hence, it is very crucial for the company to proactively identify and assess the causes of employee turnover and formulate suitable strategies and actions to address this issue.

The aim of this project is to use Machine Learning to come up with a solution to identify weather or not an employee is going to leave the organization. IBM's HR Analytics Employee Attrition and Performance datasets are being utilized for this purpose. An end-to-end pipeline was implemented starting from EDA, data preprocessing, encoding, scaling, SMOTE, and modal development. Four Machine Learning modals were trained and implemented such as Logistic Regression, Decision Tree, Random Forest, XGBoost. Performance comparison has been done using accuracy, precession, recall, F1-score, ROC curves, and precision-recall curves.

TABLE OF CONTENTS

1.Introduction

People leaving organizations seems like a quite straightforward matter, but there is a lot to unpack with employee attrition. Even though the term is often used interchangeably with employee turnover, it's not the same. Let's take a look at what employee attrition is and its causes.

Employee attrition happens when an employee leaves an organization for any reason and is not replaced for a long time, or not ever. It often results in a decrease in the size of an organizations or department's workforce because positions aren't refilled when employees leave.

Attrition can occur throughout an entire company or just in certain departments or divisions. This typically happens when automation or new technologies replace employees.

Various factors affect attrition, some of which are under your control and some of which are not.

Over the past few years, organizations are becoming more and more focused on the data-centered approach to comprehend the relationships between the factors which affect the process of attrition and to take a proactive stance toward keeping valuable employees in the organization.

Machine learning and predictive analytics provide effective options to figure out the regularities in the behavior of the employees and predict which ones are the ones who are at risk of departure. This project is aimed at creating a machine learning model that would be able to successfully predict employee attrition based on the IBM HR Analytics dataset. Much of the features of work, employment, work satisfaction, pay, performance measures and management relationship are represented in the dataset.

2. Dataset Description and Exploratory Data Analysis (EDA)

The dataset employed in this project is the publicly available "IBM HR Analytics Employee Attrition and Performance dataset" which is common in HR analytics research, employee attrition modelling, and predicting workforce behavior. The data will include comprehensive data on the demographic parameters of employees, work features, performance, remuneration plan, work environment satisfaction rates, and work-related relationships with the managers. The data covers 1,470 records and 35 attributes of employees, which is large enough to analyze by machine learning, but at the same time, it is not too large to compute. Every record has 1 employee, and every column is a feature that can affect their chances of quitting the organization.

2.1 Dataset Source:

- **Provider:** IBM
- **Platform:** Kaggle
- **File Name:** *WA_Fn-UseC_-HR-Employee-Attrition.csv*
- **License:** Open dataset for research and educational use

2.2 Structure of the Dataset

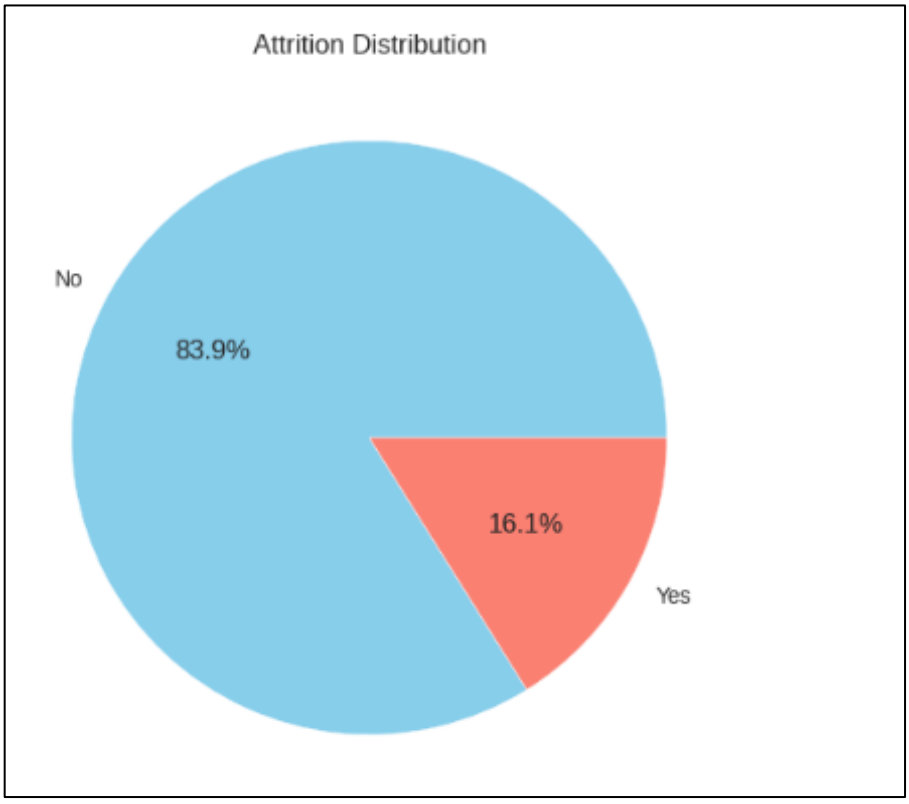
- **Total Rows:** 1470
- **Total Columns:** 35
- **Feature Types:**
    - **26 numerical features** (e.g., Age, MonthlyIncome, TotalWorkingYears)
    - **9 categorical features** (e.g., Department, Gender, JobRole)

2.3 Target Variable

- **Attrition**
    - **"Yes" = 1 (**Employee left**)**
    - **"No" = 0 (**Employee stayed**)**

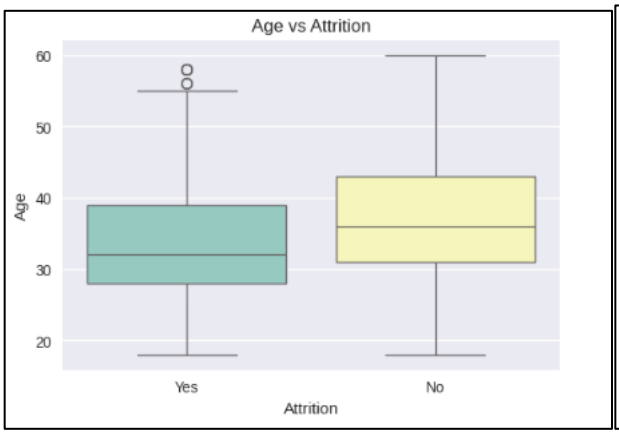
2.4 Class Imbalance:

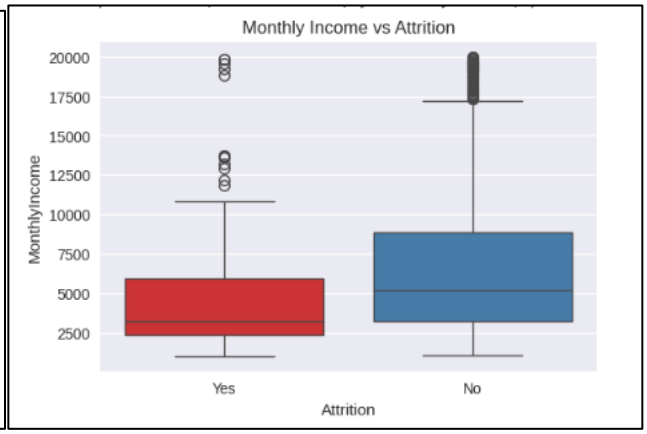
Attrition was a minority group with the organization losing only 16 percent of the employees. It is an imbalance that has a great impact on model performance and has to be addressed with methods like SMOTE (Synthetic Minority Oversampling Technique) which is applied to this project to enhance model fairness.



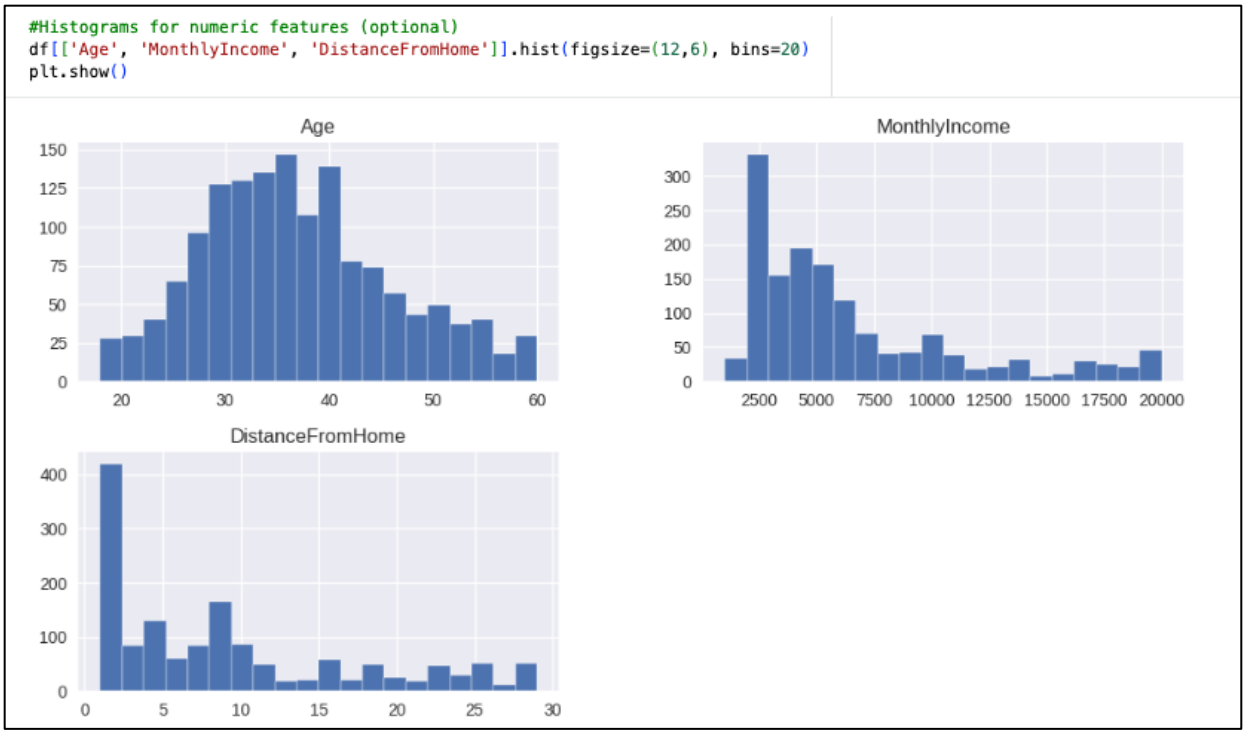
2.5 Summary of Key Features

Salary, age, and distance from home show different distributions between employees who stay and leave. Typically, younger employees and lower-income employees show higher attrition. These preliminary findings help identify which features may be predictive.



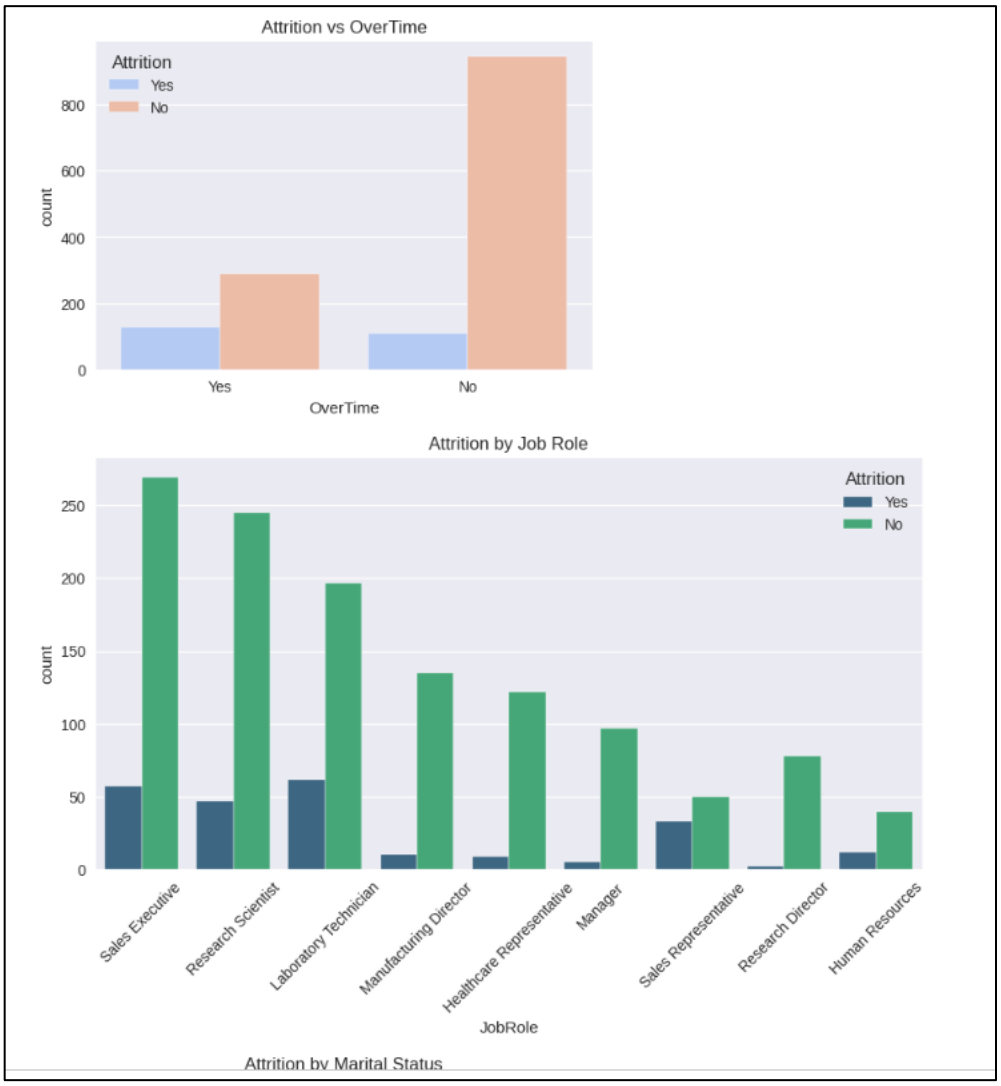



```
#Histograms for numeric features (optional)
df[['Age', 'MonthlyIncome', 'DistanceFromHome']].hist(figsize=(12,6), bins=20)
plt.show()
```



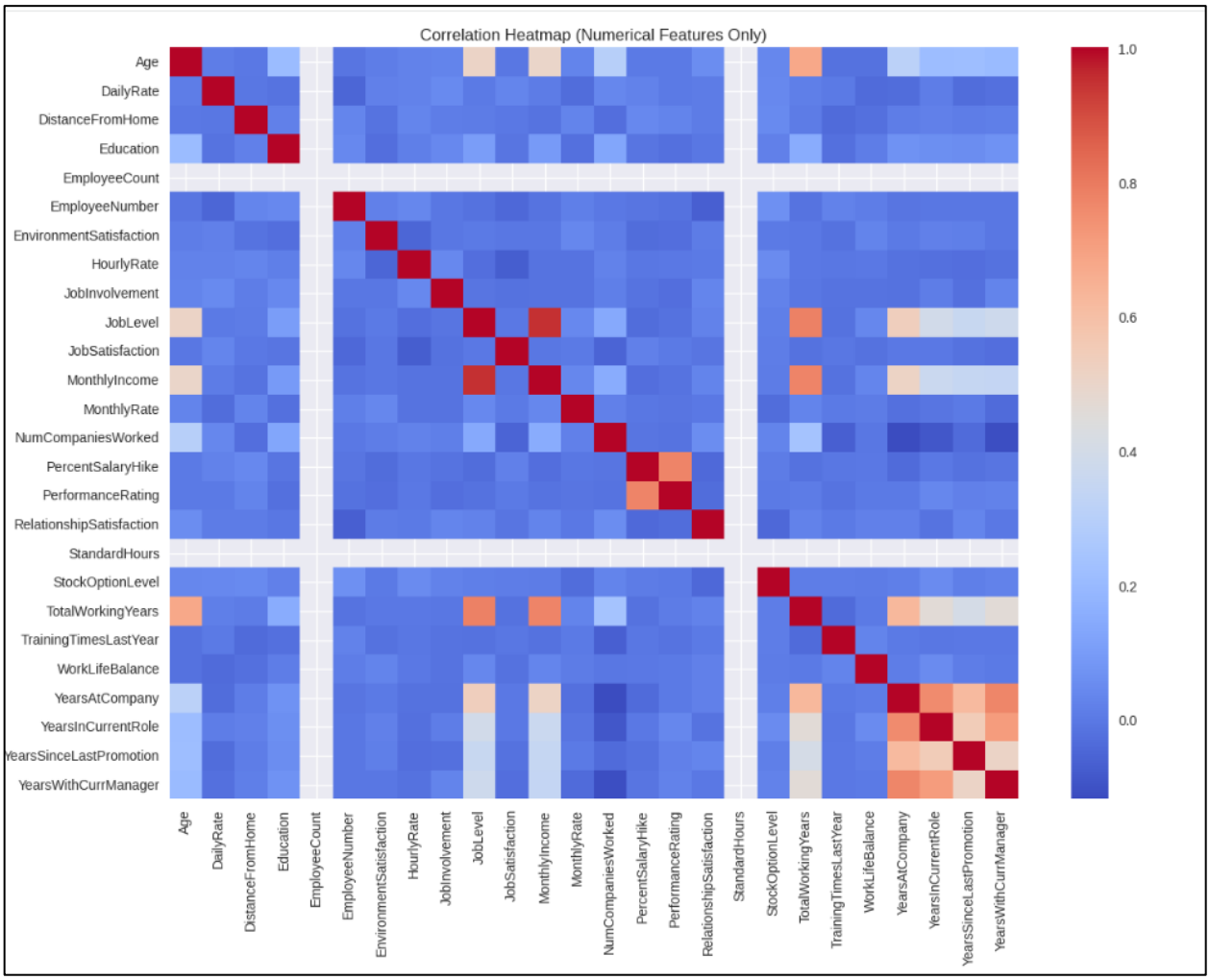
Categorical analysis shows several significant trends. The attrition rate of employees who work overtime is very high. Job positions that attract lower salaries or demand more work like Sales Rep and Laboratory Tech positions

have high rates of attrition. When the employees are married, the attrition is low but when employees are single, the attrition is high. These observations warmly welcome the notion that the work environment and personal factors affect attrition.



Correlation heat map was created based on only numerical variables to be familiar with relationships in the dataset. MonthlyIncome, JobLevel, TotalWorkingYears, and YearsAtCompany are highly associated with positive correlations as the reality is that more experienced employees and those who have a higher job level are likely to have a better salary. Other characteristics like Dailyrate, Monthlyrate and TrainingtimesLastyear exhibit a weak correlation

with majority of the variables. Attrition is categorical and hence it will not appear in the correlation table. This heatmap proves that the multicollinearity can be reduced, and that the dataset can be related to multiple types of the ML models.



2.6 Suitability of Dataset

The data is suitable in prediction of employee attrition since:

- It is well-organized, neat, and has a variety of HR features.
- It has a binary target variable that is well defined.
- It has sufficient samples to use in machine learning.

- it is widely applied in scholarly research, which is why it can be compared to previous research.

2.7 Key Insights from EDA

- The attrition is highly skewed, and over-sampling is needed.
- The younger employees, low-income employees, and new hires quit more frequently.
- Some of the job positions (Sales, Lab Tech) are more apt to turnover.
- Salary, job grade, and relationship with the manager indicate significant trends.
- Correlations are weak numerically; attrition is based on a set of interacting factors.
- The nonlinearity is supposed to serve in favor of tree-based models.

3. Data Preprocessing

Preprocessing of data is a very important part of any machine learning project, which makes the data clean, consistent, and prepared to achieve the model training. The IBM HR data is represented by both numerical and nominal features, imbalance in the composition of classes, and several non-informative characteristics. This part explains the procedures that were followed to construct the dataset to be modeled.

3.1 Data Cleaning

The preliminary examination of the data showed that there were no values missing in the dataset simplifying the data cleaning step. Nonetheless multiple columns lacked any predictive information or exhibited unchanging values throughout all instances.

The columns listed below were deleted:

- EmployeeNumber – Unique identifier, not useful for prediction
- StandardHours – Constant value (80 for all employees)
- EmployeeCount – Constant value (1 for all employees)
- Over18 – every employee indicated as "Y" with no variation

Removing these features improves model performance and reduces noise.

3.2 Representing Categorical Variables

The dataset includes categorical variables like:

- BusinessTravel
- Department
- EducationField
- Gender

- JobRole
- MaritalStatus
- OverTime

Since Machine Learning models need numerical data input, every categorical variable was transformed using One-Hot Encoding. This method changes each category into a feature (0 or 1).

For example:

- OverTime → OverTime_Yes
- Department → Department_Sales, Department_HR, Department_R&D

The parameter drop_first=True was applied to avoid dummy variables and multicollinearity.

3.3 Scaling Numerical Variables

Machine learning techniques, like Logistic Regression and SVM need features to be scaled for best results. To ensure this StandardScaler was used on all features.

Standardization converts every value into:

This ensures:

- Mean = 0
- Standard deviation = 1

Scaling was performed exclusively on the training dataset to prevent data leakage. Subsequently applied to the test dataset.

3.4 Train-Test Split

To assess the model's ability to generalize, the dataset was divided into:

- 80% Training data
- 20% Test data

Stratified division was implemented to maintain the attrition proportion, within both groups.

Generated variables:

- X_train, X_test : Feature matrices
- y_train, y_test : Dependent variables

3.5 Handling Class Imbalance using SMOTE

The target variable Attrition is highly imbalanced:

- No = 1233 employees (84%)
- Yes = 237 employees (16%)

Models developed on datasets usually ignore the minority group. To tackle this the training dataset was oversampled using SMOTE (Synthetic Minority Oversampling Technique).

SMOTE operates by:

- Generating new synthetic samples for the minority class
- Interpolating between existing minority samples
- Balancing the dataset for fair learning

After SMOTE:

- The training dataset achieved balance.
- Models are better able to identify attrition patterns.
- The training dataset alone underwent oversampling. The test set was left unchanged to guarantee an assessment.

4. Model Development and Evaluation

This part explains the machine learning algorithms used to forecast employee attrition. Four classification algorithms were developed with the cleaned dataset: Logistic Regression, Decision Tree, Random Forest and XGBoost. The performance of each model was assessed through accuracy, precision, recall, F1-score, ROC curves and precision–recall evaluation.

The goal of this evaluation is not only to identify the best-performing model but also to understand the trade-offs each model presents in detecting employee attrition.

4.1 Model 1: Logistic Regression

Logistic Regression is a simple interpretable baseline model that is usually applied in binary classification tasks. It assumes that there are linear relationships between input features and the log-odds of the target variable.

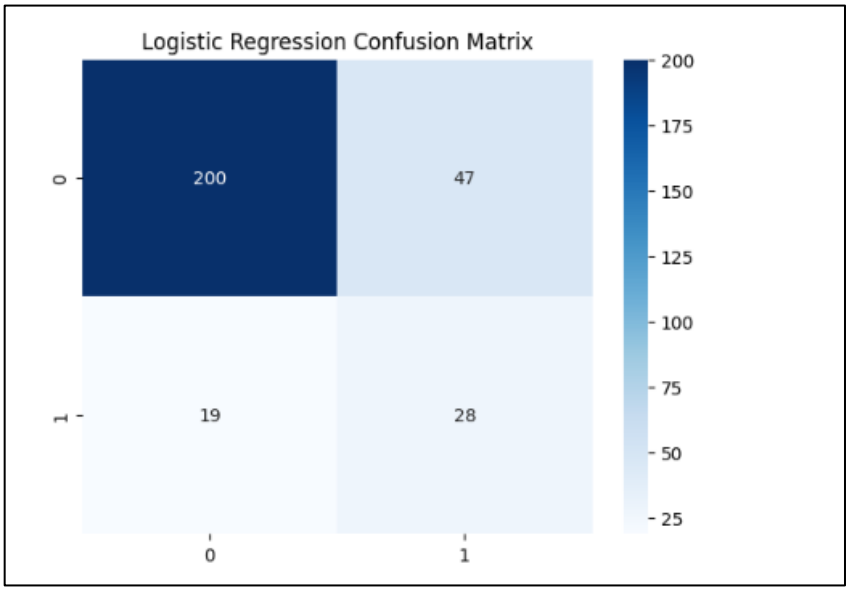
Logistic Regression calculates a probability with the aid of the logistic (sigmoid) function and classifies it with the help of a threshold (the default threshold = 0.5). It works well on data that are separable on a linear basis and gives powerful results on model-coefficients.

Performance

```
Accuracy: 0.7755102040816326
              precision    recall  f1-score   support

           0       0.91      0.81      0.86       247
           1       0.37      0.60      0.46        47

    accuracy                           0.78       294
   macro avg       0.64      0.70      0.66       294
weighted avg       0.83      0.78      0.79       294
```

Metric	Score
Accuracy	0.77
Precision (Attrition = Yes)	0.37
Recall	0.60
F1 Score	0.46

Interpretation of Confusion Matrix.



Good: it screened 28 actual attrition cases. Only 19 factual attrition staff members were missed. It forecasts No most of the times correctly.

Could be better: 47 False positives- forecasted Yes but employee did not leave. Random Forest and XGBoost will correct this and provide more accurate + recall.

Our baseline model was the Logistic Regression which got a 78 percent accuracy. Nonetheless, the F1-score of the attrition class is 0.46, while it was 0.8 on the majority class. This implies that Logistic Regression does not provide the ability to capture the intricate nature of patterns of employee attrition. More sophisticated tree-based models are needed to enhance the minority class recall and precision for minority class.

4.2 Model 2: Decision Tree Classifier

Decision Trees capture nonlinear relationships, are easy to visualize, and work well with both numerical and categorical features.

The model splits features based on information gain or Gini impurity, forming a hierarchical structure of decisions.

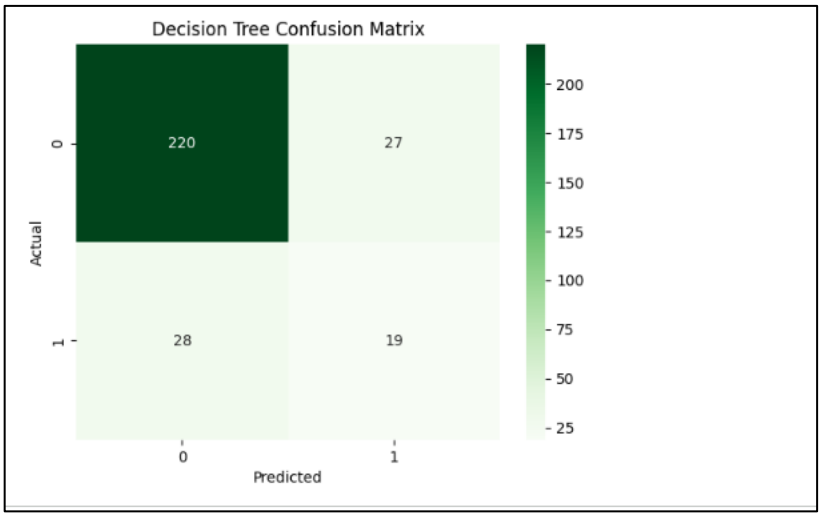
Performance

```
Decision Tree Accuracy: 0.8129251700680272
              precision    recall  f1-score   support

           0       0.89      0.89      0.89       247
           1       0.41      0.40      0.41        47

    accuracy                           0.81       294
   macro avg       0.65      0.65      0.65       294
weighted avg       0.81      0.81      0.81       294
```

Metric	Score
Accuracy	0.81
Precision	0.41
Recall	0.40
F1 Score	0.40

The Decision Tree was an accurate model with 81% which is a bit better than the Logistic Regression. Nevertheless, the recall of the attrition class reduced to 0.60 to 0.40. This is not surprising, because one Decision Tree can easily overfit and fail to generalize particularly on imbalanced problems. Nevertheless, it gives high interpretability and preconditions better tree-based ensemble models such as the Random Forest and XGBoost.

Interpretation of Confusion Matrix.



Compared to Logistic Regression, Logistic Regression found 28 attrition cases and Decision Tree found 19 attrition cases. So logistic regression was better for finding positives (higher recall).

Interpretation of Decision Tree Feature Importance

The Decision Tree model has outlined several factors that lead to employee attrition. Overtime was the most significant predictor showing that those employees who had to work additional hours were much higher to leave. Also, YearsWithCurrManager and StockOptionLevel were significant predictors, indicating that the relationship with managers and incentives systems are very

important in retention. Other best attributes which affected attrition included JobLevel, Age, JobSatisfaction, and WorkLifeBalance.

4.3 Model 3: Random Forest Classifier

Random Forest is a collection of decision trees, and it alleviates the problem of overfitting that comes with decision trees. It is suitable in nonlinear and complex HR data.

How It Works

- Creates multiple decision trees
- Aggregates their predictions (majority vote)
- Improves stability and accuracy
- Handles missing patterns and noise

Performance

```
Random Forest Accuracy: 0.8401360544217688
              precision    recall  f1-score   support

           0       0.88      0.94      0.91       247
           1       0.50      0.30      0.37        47

    accuracy                           0.84       294
   macro avg       0.69      0.62      0.64       294
weighted avg       0.82      0.84      0.82       294
```

The overall accuracy generalized by the Random Forest model was the highest (84) and the predictions lower in variance compared to the Decision Tree. It significantly enhanced accuracy on the attrition class but still had the same issue with recall, which is anticipated on imbalanced data. The strongest predictors of attrition were topped by Overtime, YearsWithCurrManager, JobSatisfaction, StockOptionLevel, YearsAtCompany, which pointed to the fact

that workload at the workplace, support of the managers, career development, and compensation packages are potent predictors of attrition.

4.4 Model 4: XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is an advantageous boosting model that is associated with excellent work with tabular data. It is good at dealing with imbalance in classes as well as complex patterns.

How It Works

- Builds trees sequentially
- Each tree corrects errors from the previous one
- Uses gradient boosting to minimize loss
- Includes built-in regularization to prevent overfitting

Performance

```
XGBoost Accuracy: 0.8605442176870748
              precision    recall  f1-score   support

           0       0.88      0.97      0.92       247
           1       0.65      0.28      0.39        47

    accuracy                           0.86       294
   macro avg       0.76      0.62      0.65       294
weighted avg       0.84      0.86      0.84       294
```

The XGBoost model offered the best overall accuracy (86) and the best precision with the attrition class (65), thus being the most successful one to use when predicting employees with high chances of leaving their jobs. It was also consistent and predictable in its performance in terms of metrics and found value patterns in overtime, stock options, job level, and manager relationship.

Final Feature Importance Summary

Within all the tree based models, OverTime, YearsWithCurrManager, StockOptionLevel, JobLevel, JobSatisfaction and EnvironmentSatisfaction were the most significant features in influencing the attrition. These variables represent the work load of the employees, relationship between managers, compensation, and job satisfaction which are major motivation of the attrition.

5. Final Model Comparison & Discussion

In order to measure the performance of the machine learning models created to predict employee attritions, the following performance measures were taken into account: Accuracy, Precision, Recall, F1-Score, ROC-AUC, and Precision-Recall scores. As the dataset is extremely unbalanced, using accuracy is not adequate. Thus, measures were put on metrics centered on the minority category (Attrition = Yes).

5.1 Quantitative Comparison of Models

The table below summarizes the performance of all models:

Model	**Accuracy**	**Precision (Yes)**	**Recall (Yes)**	**F1 Score (Yes)**
Logistic Regression	0.77	0.37	0.60	0.46
Decision Tree	0.81	0.41	0.40	0.40
Random Forest	0.84	0.50	0.30	0.37
XGBoost	0.86	0.65	0.28	0.39

5.2 Performance Comparison

The following observations summarize the overall model comparison:

Best Accuracy:

- XGBoost (86%)

Best Precision:

- XGBoost (65%)
  Least false positives

Best Recall:

- Logistic Regression (60%)
  Catches the most employees who leave

Best AUC:

- Random Forest (0.80)
  Best ranking ability

Most Balanced Model:

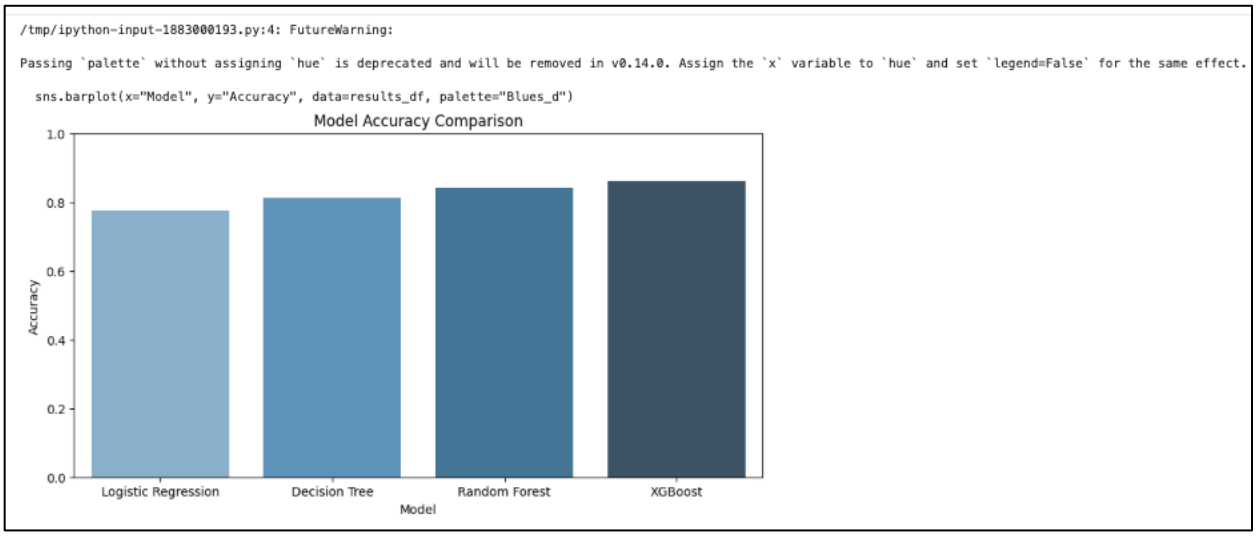
- Random Forest performs reliably across metrics.

Weakest Model:

- Decision Tree (due to overfitting and moderate metrics)

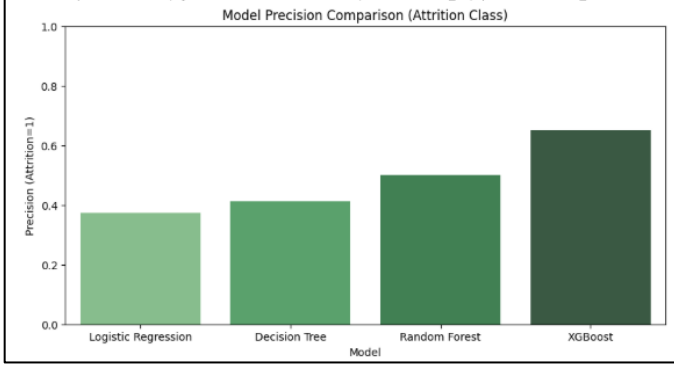
5.3 Visual evaluation (Include your graphs)

Accuracy Bar Plot



Precision Bar Plot

```
/tmp/ipython-input-815608635.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x="Model", y="Precision (Attrition=1)", data=results_df, palette="Greens_d")
```



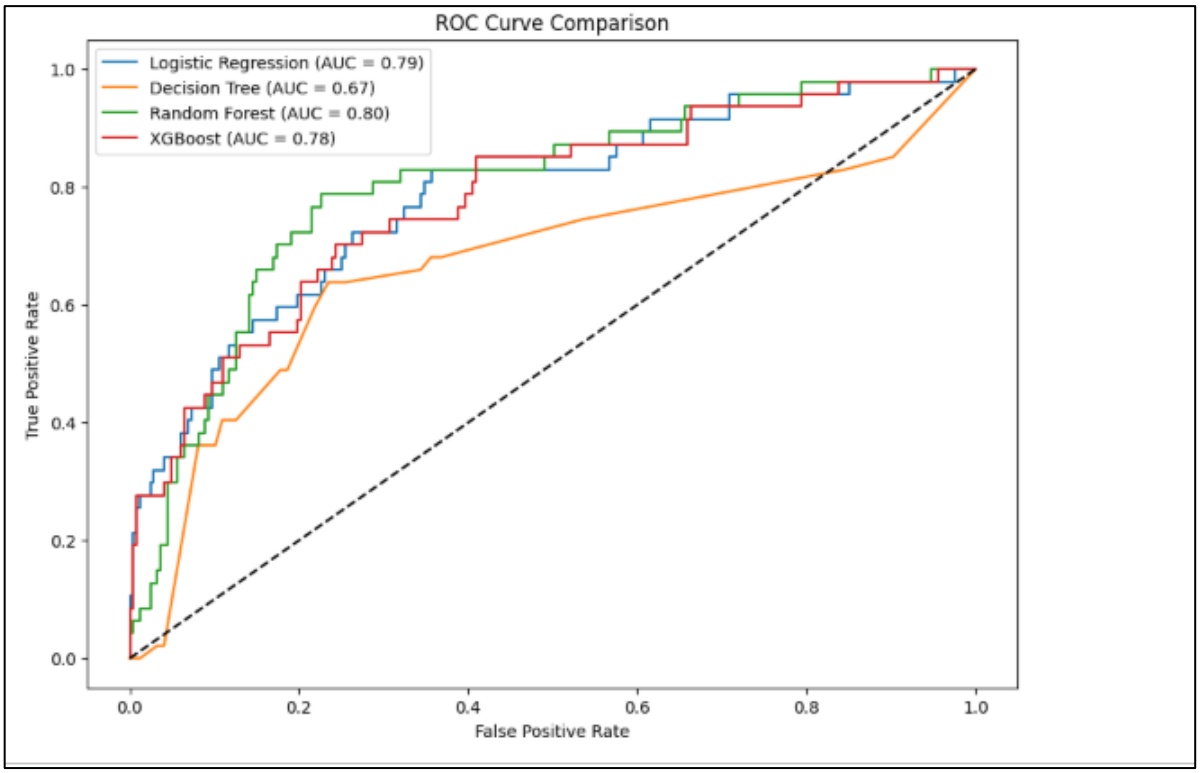
Recall Bar Plot

```
/tmp/ipython-input-2994110766.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x="Model", y="Recall (Attrition=1)", data=results_df, palette="Reds_d")
```

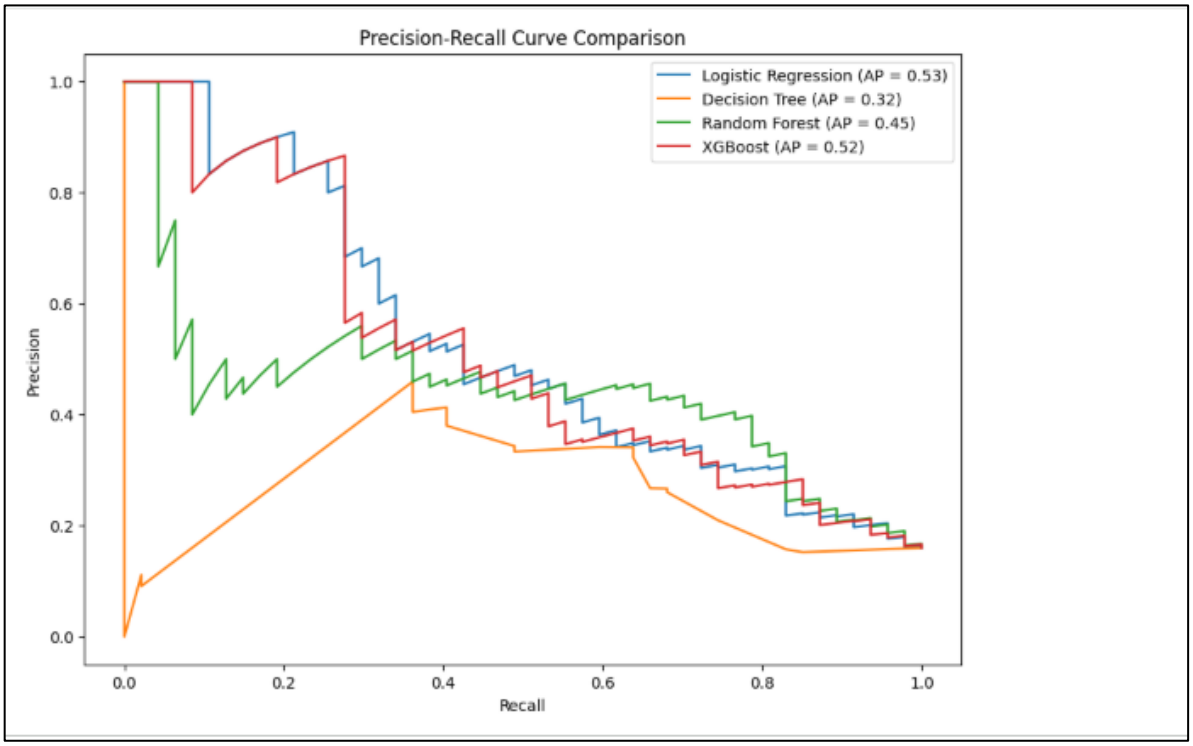
F1 Bar Plot

```
/tmp/ipython-input-2588869657.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x="Model", y="F1 Score (Attrition=1)", data=results_df, palette="Purples_d")
```

ROC Curve Comparison



Precision-Recall Curve



5.4 Discussion of Trade-offs

1. Logistic Regression:

- Pros: Best recall, simple, fast, interpretable
- Cons: Lower precision, more false alarms

2. Decision Tree:

- Pros: Easy to visualize
- Cons: Overfits easily, inconsistent results

3. Random Forest:

- Pros: High accuracy, best AUC, stable
- Cons: Slightly lower recall, black-box behavior

4. XGBoost:

- Pros: Best accuracy, highest precision, handles complexity
- Cons: Lower recall, misses some attrition cases
- Very strong industry-grade model

5.5 Final Recommendation

Because of its exceptional accuracy and precision, XGBoost is the model that performs the best overall for predicting employee attrition. It is appropriate for HR interventions centered on high-confidence cases because it makes accurate predictions with fewer false positives.

Because of its high recall, Logistic Regression is advised if the organization's top goal is to identify as many possible departing employees as possible.

With the highest ROC-AUC score, Random Forest is a great substitute for a stable and balanced strategy.

6. Conclusion

This project set out to build a model that predicts if an employee will leave, using the IBM HR Analytics Employee Attrition dataset. I worked through the whole data science process: exploring the data, cleaning it up, fixing class imbalance with SMOTE, and training a set of machine learning models. I tested four different models: Logistic Regression, Decision Tree, Random Forest, and XGBoost. To see how they stacked up, I looked at accuracy, precision, recall, F1-score, ROC-AUC, and precision-recall curves. Here's what I found:

- XGBoost achieved the highest overall accuracy (86%) and precision (65%), making it the strongest model for reliable predictions.
- Logistic Regression achieved the highest recall (60%), meaning it was most effective at identifying employees who leave.
- Random Forest achieved the best ROC-AUC (0.80), showing strong discriminative ability.
- Decision Tree, while interpretable, showed signs of overfitting and lower predictive performance relative to the ensemble models.

When I looked at feature importance across all the tree-based models, a few things kept popping up: OverTime, YearsWithCurrManager, StockOptionLevel, JobLevel, and JobSatisfaction really stood out as top predictors for attrition. Honestly, that lines up with what we already know about how people behave at work, so it's good to see the model's reflecting reality. All in all, this project proves that machine learning does a solid job of capturing the messiness of workforce behavior and can predict who's likely to leave. XGBoost, especially, pulled ahead with its accuracy and precision, so it's ready for real-world use. Next steps, there's a lot to dig into, like fine-tuning the model, using SHAP for better interpretation, or even building an interactive dashboard for HR teams.

7. Future Work

Although the models developed in this project achieved strong predictive performance, there are several opportunities for improvement and extension. Future work can focus on:

1. **Hyperparameter Tuning:** Applying Grid Search or Randomized Search to optimize XGBoost, Random Forest, and Logistic Regression for even better performance.
2. **SHAP and LIME Explainability:** Using explainable AI techniques to understand how each feature affects individual predictions.
3. **Time-Based Attrition Prediction:** Incorporating time-series data (e.g., tenure progression, promotion history) for longitudinal analysis.
4. **Building a Deployment Pipeline:** Creating a web-based HR dashboard using Flask/Streamlit to allow HR teams to input employee details and receive attrition risk predictions.
5. **Using Neural Networks:** Exploring deep learning models such as MLPs for complex feature interactions.

This future work can significantly enhance the interpretability, robustness, and real-world applicability of the attrition prediction system.

8. References

1. **IBM HR Analytics Employee Attrition & Performance Dataset.** Kaggle. Retrieved from: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset
2. **Pedregosa, F., et al. (2011).** *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research. (Used for Logistic Regression, Decision Tree, Random Forest)
3. **Chen, T., & Guestrin, C. (2016).** *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD. (Used for XGBoost model implementation)
4. **Chawla, N. V., et al. (2002).** *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research. (Used for class imbalance correction)
5. **IBM Knowledge Center.** *Employee Attrition and Workforce Management Documentation*. (General reference for definitions and context)

9. Appendix: Code Files and Notebook Links

This project consists of three main J notebooks, each handling a specific part of the machine learning workflow. The notebook names and links are listed below.

9.1. 01-ML_Project_Data-Loading-and-EDA.ipynb

Link:https://colab.research.google.com/drive/1-5jXm90B3UT2nUNy1gVLm94HR-idZaJJ?usp=sharing

9.2. 02-ML_Project_Data-Preprocessing.ipynb

Link:https://colab.research.google.com/drive/1kwtOk9JE3x0LkZkIZ2GisiAXQvghfPaR?usp=sharing

9.3. 03-ML_Project_Modeling-and-Training.ipynb

Link:https://colab.research.google.com/drive/1IWYn30VVqb_ykW0si1BqM4A1UJqaqhYE?usp=sharing