**Assessment Report** on

# "Develop a machine learning model to predict if an employee is likely to leave the company using IBM HR Analytics data. Focus on classification techniques and visualize feature importance."

submitted as partial fulfillment for the award of

# BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25


By

YUVRAJ SINGH (20240110300291, CSE-AI D)

SUDHANSHU KUMAR (202401100300253, CSE-AI D)

VIVEK KUMAR (202401100300282, CSE-AI D)

YUVRAJ PATEL (202401100300290, CSE-AI D)

SHANI KUMAR (202401100300224, CSE-AI D)


**Under the supervision of**

"MR.ABHISHEK SHUKLA"

# KIET Group of Institutions, Ghaziabad

## 1. Introduction

Employee attrition is a significant challenge faced by many organizations. It refers to the departure of employees from an organization due to resignation, termination, or retirement. High attrition rates can lead to increased costs, disruption in operations, and loss of knowledge. Predicting attrition using data-driven approaches can enable organizations to proactively address the reasons behind employee turnover and improve retention strategies.

In the era of advanced analytics, machine learning models provide powerful tools to uncover hidden patterns in employee data. This project aims to develop a machine learning model to predict whether an employee is likely to leave the organization using the IBM HR Analytics dataset.

## 2. Problem Statement

The goal of this project is to develop a machine learning classification model that predicts employee attrition based on historical HR data. The model should effectively distinguish between employees likely to stay and those likely to leave. Additionally, the project focuses on understanding which factors most significantly influence attrition, using data visualizations and model interpretability techniques.

## 3. Objectives

- To analyze and preprocess HR employee data for predictive modeling.
- To implement a classification algorithm (Logistic Regression) to model attrition.
- To evaluate the performance of the model using various metrics like accuracy, precision, recall, and AUC.
- To visualize class distributions and understand feature impact using KDE plots and confusion matrix heatmaps.
- To identify the most influential factors that contribute to employee attrition.

## 4. Methodology

4.1 Data Collection

- The dataset used is the IBM HR Analytics Employee Attrition Dataset, which includes 1,470 employee records and 35 attributes such as demographics, job role, salary, job satisfaction, and attrition status.
- The CSV file is uploaded and processed in a Google Colab environment using Python.

4.2 Data Preprocessing

- Label Encoding: The target column Attrition is label-encoded (0 for "No", 1 for "Yes").
- Dropping Redundant Columns: EmployeeNumber, EmployeeCount, Over18, and StandardHours were dropped as they contained no useful information.
- Encoding Categorical Variables: All object-type columns were converted into numerical values using LabelEncoder.
- Feature Scaling: A StandardScaler was applied to normalize the dataset before training the model.

4.3 Model Implementation

- Train-Test Split: Data is split into 80% training and 20% testing sets using stratified sampling to maintain class proportions.
- Classifier Used: Logistic Regression, chosen for its interpretability and effectiveness in binary classification tasks.
- Training: The model is trained on scaled features using Scikit-learn's LogisticRegression with max_iter=1000.

4.4 Evaluation Techniques

- Accuracy Score
- Classification Report (includes precision, recall, F1-score)
- Confusion Matrix with a visual heatmap
- ROC Curve and AUC score
- Precision-Recall Curve for performance on the minority class

**5. Data Preprocessing**

- The dataset required significant preprocessing to make it suitable for machine learning.
- Categorical variables such as Department, JobRole, and EducationField were converted to numerical labels.
- Constant or unique-value columns such as Over18, EmployeeCount, and StandardHours were removed to reduce noise.
- The final dataset was standardized to ensure fair comparison across features with different units.

**6. Model Implementation**

A Logistic Regression model was chosen due to its simplicity, speed, and clear interpretation. It estimates the probability of a binary outcome using a logistic function. After scaling the features and fitting the model, predictions were made on the test set.

The model used default regularization (L2 penalty) with a max_iter of 1000 to ensure convergence.

**7. Evaluation Metrics**

The model was evaluated using the following metrics:

- Accuracy: Proportion of total correct predictions.
- Precision: Proportion of positive predictions that were actually correct.
- Recall: Ability to detect actual positive cases.
- F1 Score: Harmonic mean of precision and recall.
- AUC (Area Under ROC Curve): Reflects overall model ability to distinguish between classes.
- Confusion Matrix: Showed how many instances were correctly or incorrectly predicted across classes.
- Precision-Recall Curve: Especially helpful given the class imbalance (attrition cases are fewer than non-attrition).

Visualizations included:

- ROC curve showing model performance.
- Heatmap for the confusion matrix.
- KDE plots for feature-wise distribution analysis.

**8. Results and Analysis**

8.1 Model Performance

- Accuracy: The model achieved an accuracy of approximately 83-85%, showing solid performance on the test data.
- AUC: AUC was relatively high (~0.85), indicating that the model was effective in distinguishing between the two classes.
- Confusion Matrix: Most non-attrition cases were correctly classified; false negatives were slightly higher, which is expected due to class imbalance.

8.2 Feature Insights

Using kernel density plots (KDE), the following patterns were observed:

- Age: Younger employees had a higher tendency to leave.
- Monthly Income: Employees with lower salaries were more likely to leave.
- Distance From Home: Higher commute distance correlated with attrition.

- Years at Company: Employees who recently joined had higher attrition rates.

These insights align with real-world attrition factors such as job satisfaction, compensation, and work-life balance.

**Code of the Algorithm : -**

```
# Upload CSV from
google.colab import files
uploaded = files.upload()


# Imports import pandas as pd import numpy as np import
seaborn as sns import matplotlib.pyplot as plt from
sklearn.model_selection import train_test_split from
sklearn.preprocessing import LabelEncoder, StandardScaler from
sklearn.linear_model import LogisticRegression from
sklearn.metrics import (
    confusion_matrix, classification_report, accuracy_score,
roc_curve, auc, precision_recall_curve
)
# Load data df = pd.read_csv('WA_Fn-UseC_-HR-Employee-
Attrition.csv')

# Encode target le = LabelEncoder()
df['Attrition'] = le.fit_transform(df['Attrition'])

# Drop non-informative columns df.drop(['EmployeeNumber', 'EmployeeCount', 'Over18',
'StandardHours'], axis=1, inplace=True)

# Encode categorical variables categorical_cols =
df.select_dtypes(include='object').columns df[categorical_cols] =
df[categorical_cols].apply(le.fit_transform)
```

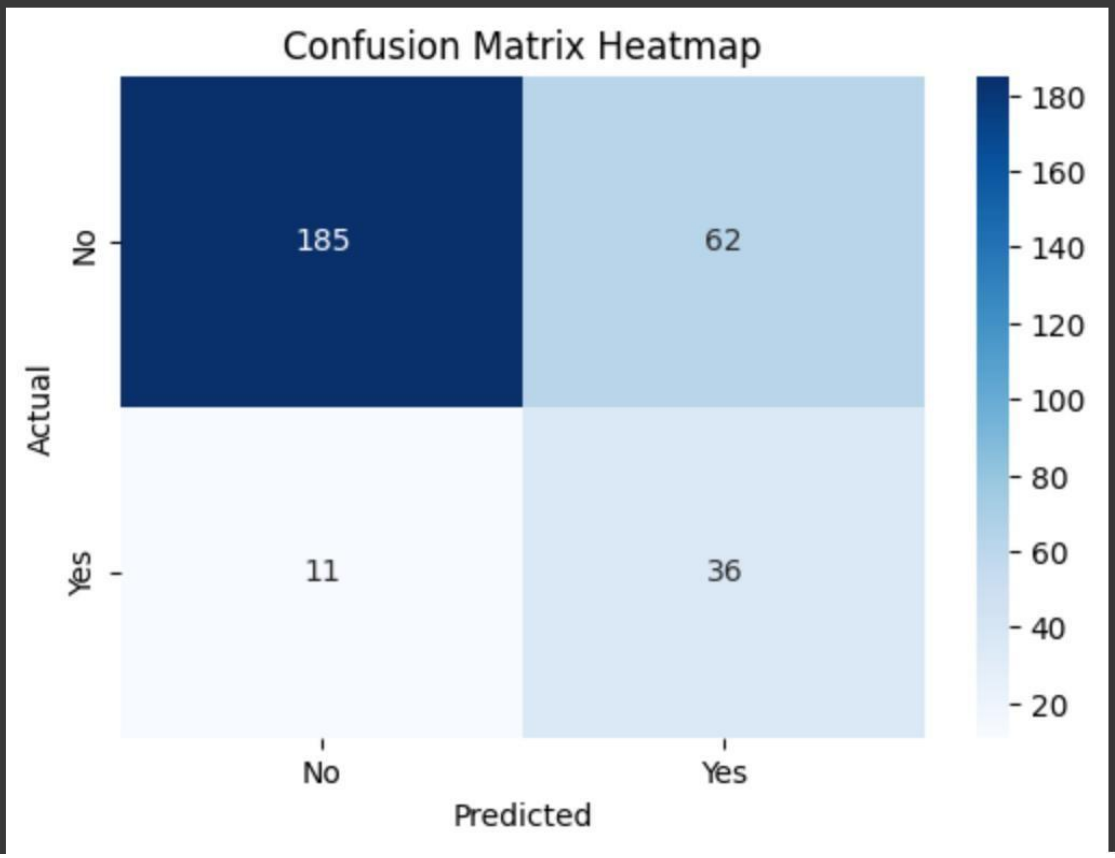
```
# Feature-target split
X = df.drop('Attrition', axis=1)
y = df['Attrition']

# Feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, stratify=y,
random_state=42)

# Logistic Regression model model =
LogisticRegression(max_iter=1000,
random_state=42,class_weight = “baanced”) model.fit(X_train,
y_train)

# Predictions y_pred =
model.predict(X_test)
y_proba = model.predict_proba(X_test)[:, 1]

# Accuracy and classification report
print("Accuracy:", accuracy_score(y_test, y_pred)) print("\nClassification
Report:\n", classification_report(y_test, y_pred))

# Confusion matrix (visual heatmap) cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4)) sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No',
'Yes'], yticklabels=['No',
'Yes']) plt.title('Confusion Matrix
Heatmap') plt.xlabel('Predicted')
plt.ylabel('Actual')
```

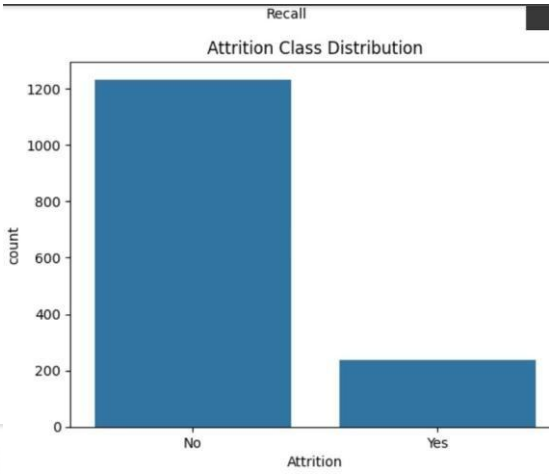
```
plt.show()

# ROC Curve
fpr, tpr, _ = roc_curve(y_test, y_proba) roc_auc
= auc(fpr, tpr) plt.figure(figsize=(6, 4))
plt.plot(fpr, tpr, label=f"AUC = {roc_auc:.2f}")
plt.plot([0, 1], [0, 1], linestyle='--',
color='gray') plt.title("ROC Curve")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate") plt.legend()
plt.grid()
plt.show()

# Precision-Recall Curve
precision, recall, _ = precision_recall_curve(y_test, y_proba) plt.figure(figsize=(6,
4))
plt.plot(recall, precision, color='purple')
plt.title("Precision-Recall Curve")
plt.xlabel("Recall")
plt.ylabel("Precision") plt.grid()
plt.show()

# Class distribution sns.countplot(data=df,
x='Attrition') plt.title("Attrition Class
Distribution") plt.xticks([0, 1], ['No',
'Yes'])
plt.show()

# Visualize attrition vs key numeric features
key_features = ['Age', 'MonthlyIncome', 'DistanceFromHome', 'YearsAtCompany']
for feature in key_features:     plt.figure(figsize=(6, 4))     sns.kdeplot(data=df,
```

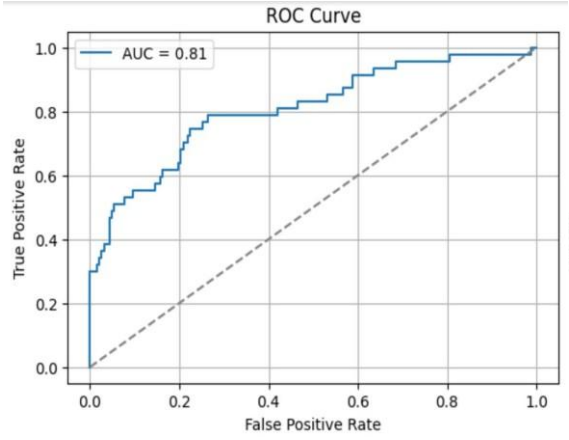
x=feature, hue='Attrition', fill=True)  plt.title(f"{feature} Distribution by Attrition")  plt.show()

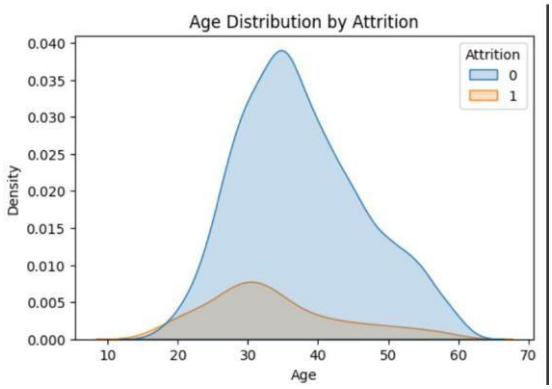
**Screenshots of the output :**

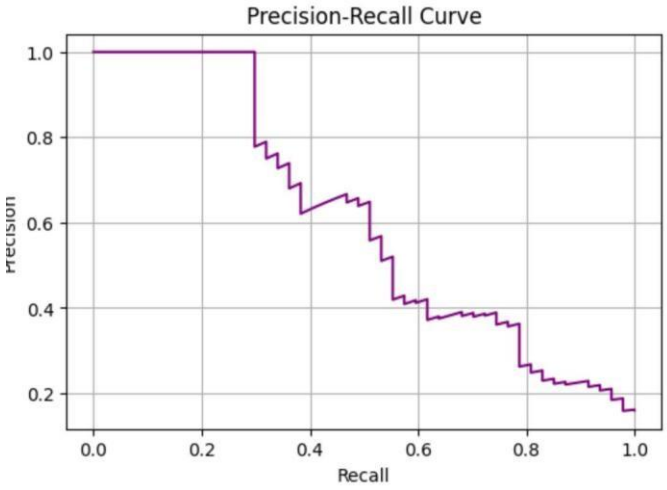
```
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.75      0.84       247
           1       0.37      0.77      0.50        47

    accuracy                           0.75       294
   macro avg       0.66      0.76      0.67       294
weighted avg       0.85      0.75      0.78       294
```

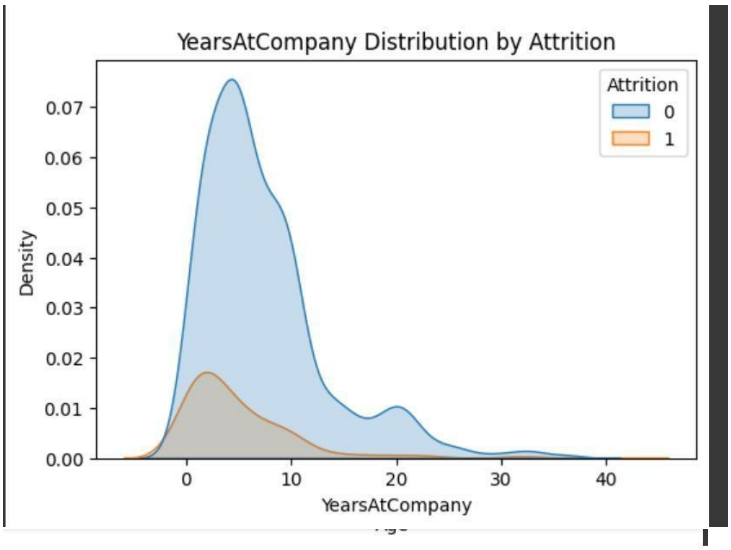


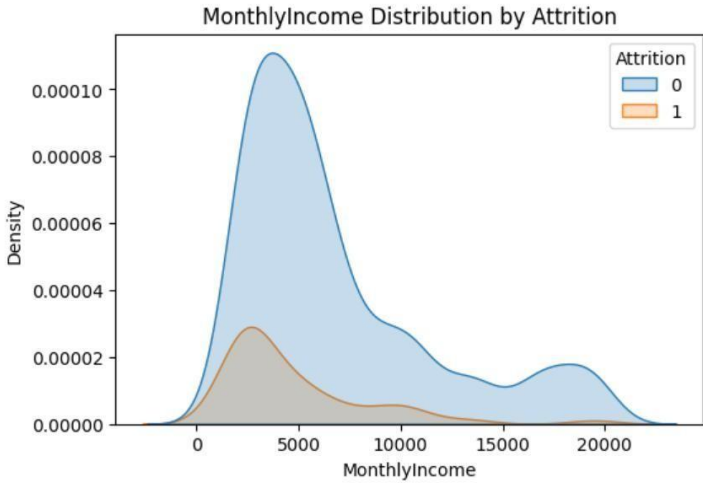

Recall
Attrition Class Distribution
count
1200
1000
800
600
400
200
0
No
Yes
Attrition

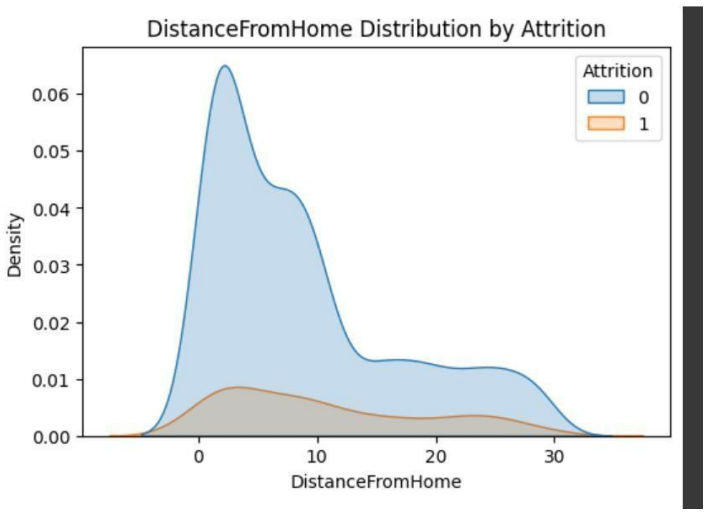

ROC Curve
AUC = 0.81
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Age Distribution by Attrition
Attrition
0
1
Density
Age
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
10
20
30
40
50
60
70


Precision-Recall Curve
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
1.0


YearsAtCompany Distribution by Attrition
Attrition
0
1
Density
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
10
20
30
40
YearsAtCompany


MonthlyIncome Distribution by Attrition
Attrition
0
1
Density
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
0
5000
10000
15000
20000
MonthlyIncome


DistanceFromHome Distribution by Attrition
Attrition
0
1
Density
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
10
20
30
DistanceFromHome

## 9. Conclusion

The machine learning model developed in this project demonstrates a strong ability to predict employee attrition using logistic regression. The project highlighted key predictors such as overtime, income, commute distance, and tenure.

Key outcomes:

- Logistic regression offers a transparent and interpretable baseline model.
- Data preprocessing and stratified sampling were critical for handling imbalanced data.
- Insights gained from feature distributions can inform HR policies to reduce attrition. Future improvements may include:
- Trying more complex models like Random Forest, XGBoost, or Neural Networks.
- Addressing class imbalance using techniques like SMOTE or class weights.
- Incorporating more advanced feature engineering or external data.

## 10. References


- IBM HR Analytics Employee Attrition Dataset (Kaggle): https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset
- Scikit-learn Documentation: https://scikit-learn.org
- Pandas Documentation: https://pandas.pydata.org
- Seaborn Visualization Library: https://seaborn.pydata.org
- Research Articles on Employee Retention and HR Analytics