



**KIET**  
**GROUP OF INSTITUTIONS**  
*Connecting Life with Learning*



**Assessment Report**  
on  
**“Predict Employee Attrition”**  
submitted as partial fulfillment for the award of  
**BACHELOR OF TECHNOLOGY**  
**DEGREE**  
SESSION 2024-25  
in  
**INTRODUCTION TO AI**  
By  
SURAJ KUMAR (202401100300256, CSE-AI D)

**Under the supervision of**  
**“MR.ABHISHEK SHUKLA”**

**KIET Group of Institutions, Ghaziabad**

# **Introduction**

**Employee attrition refers to the loss of employees through resignation, retirement, or other causes.**

**High attrition affects productivity, increases hiring costs, and lowers employee morale.**

**Predicting attrition helps organizations retain talent and plan workforce strategies.**

**With machine learning, we can analyze employee data to identify patterns of attrition.**

**This project focuses on predicting whether an employee is likely to leave the company.**

**Key features used include Job Satisfaction, Monthly Income, Years at Company, and Work-Life Balance.**

**A classification model is built using a Random Forest algorithm.**

**The goal is to support HR decisions by identifying at-risk employees early.**

**This approach helps reduce turnover and improve organizational efficiency.**

**Predictive analytics is a powerful tool for modern HR management.**

# Methodology

The dataset was first loaded and cleaned by removing any rows with missing values.

Key features selected for prediction were Job Satisfaction, Monthly Income, Years at Company, and Work-Life Balance.

Categorical data was encoded using Label Encoding to prepare it for machine learning.

The target variable, Attrition, was also encoded to binary form (Yes = 1, No = 0).

Data was split into training and testing sets using an 80-20 ratio.

A Random Forest Classifier was used for building the prediction model.

The model was trained on the training data and then tested on unseen data.

Predictions were evaluated using Accuracy, Precision, Recall, and a Confusion Matrix.

Seaborn and Matplotlib were used for visualization of the confusion matrix.

# CODE

```
from google.colab import files
uploaded = files.upload()
import pandas as pd

# Replace with your exact filename if needed
df = pd.read_csv("6. Predict Employee Attrition.csv")

# Preview data
df.head()
print(df.columns)
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix,
accuracy_score, precision_score, recall_score
import seaborn as sns
import matplotlib.pyplot as plt

# Drop any rows with missing data
df = df.dropna()

# Set features and target
features = ['JobSatisfaction', 'YearsAtCompany',
'MonthlyIncome', 'WorkLifeBalance']
target = 'Attrition'

# Encode categorical features if necessary
for col in features:
    if df[col].dtype == 'object':
        df[col] = LabelEncoder().fit_transform(df[col])

# Encode target column
le_target = LabelEncoder()
```

```
df[target] = le_target.fit_transform(df[target])

# Split data
X = df[features]
y = df[target]
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Train model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

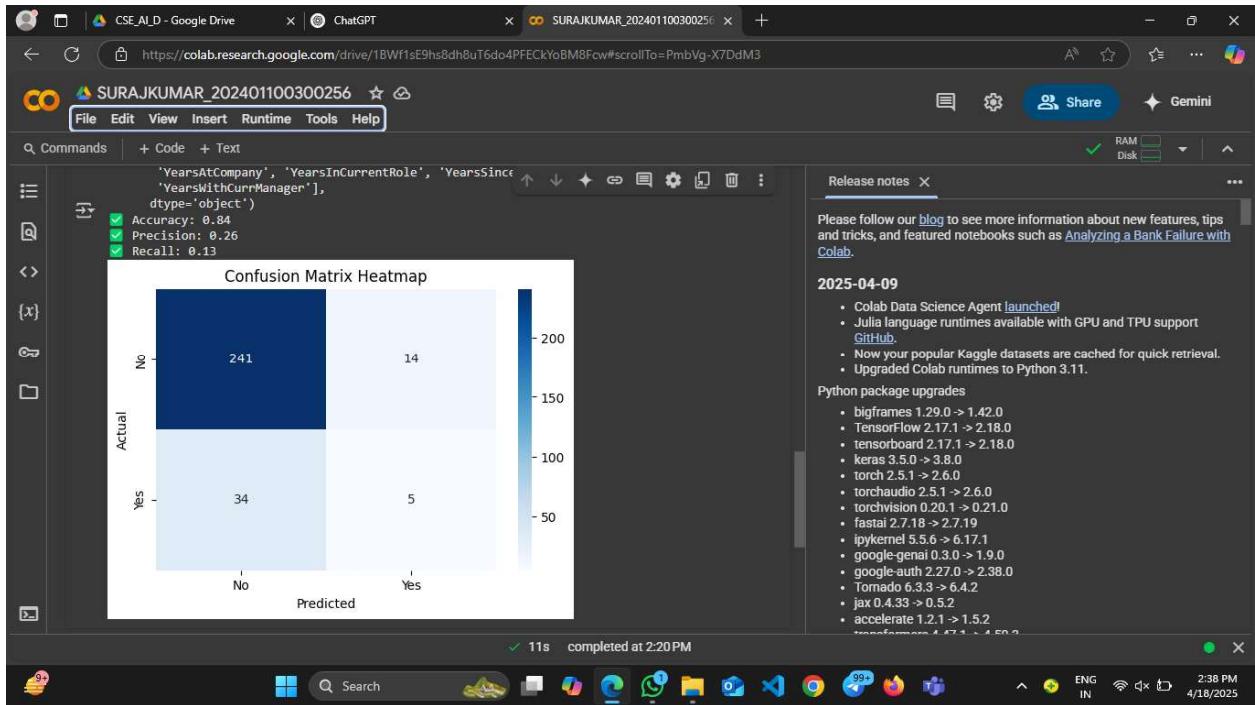
# Evaluation metrics
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)

print(f"✓ Accuracy: {acc:.2f}")
print(f"✓ Precision: {prec:.2f}")
print(f"✓ Recall: {rec:.2f}")

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=le_target.classes_,
            yticklabels=le_target.classes_)
plt.title("Confusion Matrix Heatmap")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

# OUTPUT



# References/Credits

- Dataset Source: UCI Machine Learning Repository - Employee\_Attrition\_Dataset
  - Python Libraries: pandas, seaborn, matplotlib, scikit-learn
  - Developed using Jupyter Notebook and Python 3.x

