Assessment Report

on

“Predict Employee Attrition”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Sarthak sharma

Roll Number : 202401100400166 ,

section: c

Under the supervision of

“ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

May, 2025

📊 Employee Attrition Prediction Report

📚 Introduction

Employee attrition, or turnover, refers to employees leaving a company. Predicting whether an employee will leave the organization is crucial for businesses to plan retention strategies and improve employee satisfaction. This report details a machine learning approach to predict employee attrition using various features such as job satisfaction, salary, work environment, and experience.

The dataset used for this project includes historical data about employees and whether they stayed with the company (target variable: `Attrition`). The goal is to build a predictive model using a **Random Forest Classifier** to accurately forecast employee attrition.

🛠️ Methodology

1. Data Preprocessing

Data preprocessing is crucial to ensure that the dataset is clean and suitable for machine learning. The following steps were performed:

- **Handling Missing Data**: Any missing data or irrelevant columns were removed.
- **Encoding Categorical Variables**: Label encoding was used for categorical variables to convert them into numeric values.
- **Feature Selection**: Columns like `EmployeeNumber`, `EmployeeCount`, `Over18`, and `StandardHours` were dropped as they were deemed non-informative.

2. Model Selection

A **Random Forest Classifier** was chosen for this classification task due to its ability to handle large datasets and its robustness to overfitting. It works by creating multiple decision trees and using them to predict the target.

3. Model Training & Evaluation

The dataset was split into training (80%) and testing (20%) sets. The model was trained on the training set, and its performance was evaluated on the testing set using metrics such as **accuracy**, **precision**, and **recall**. A confusion matrix was also used to visualize the model’s performance.

💻 Code

The following is the code used in this project:

python

CopyEdit

```
# 📦 Step 1: Import Required Libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score

# 📁 Step 2: Upload Dataset File
from google.colab import files
uploaded = files.upload()  # Upload '6. Predict Employee
Attrition.csv'

# 📂 Step 3: Load the Dataset
df = pd.read_csv("6. Predict Employee Attrition.csv")
```

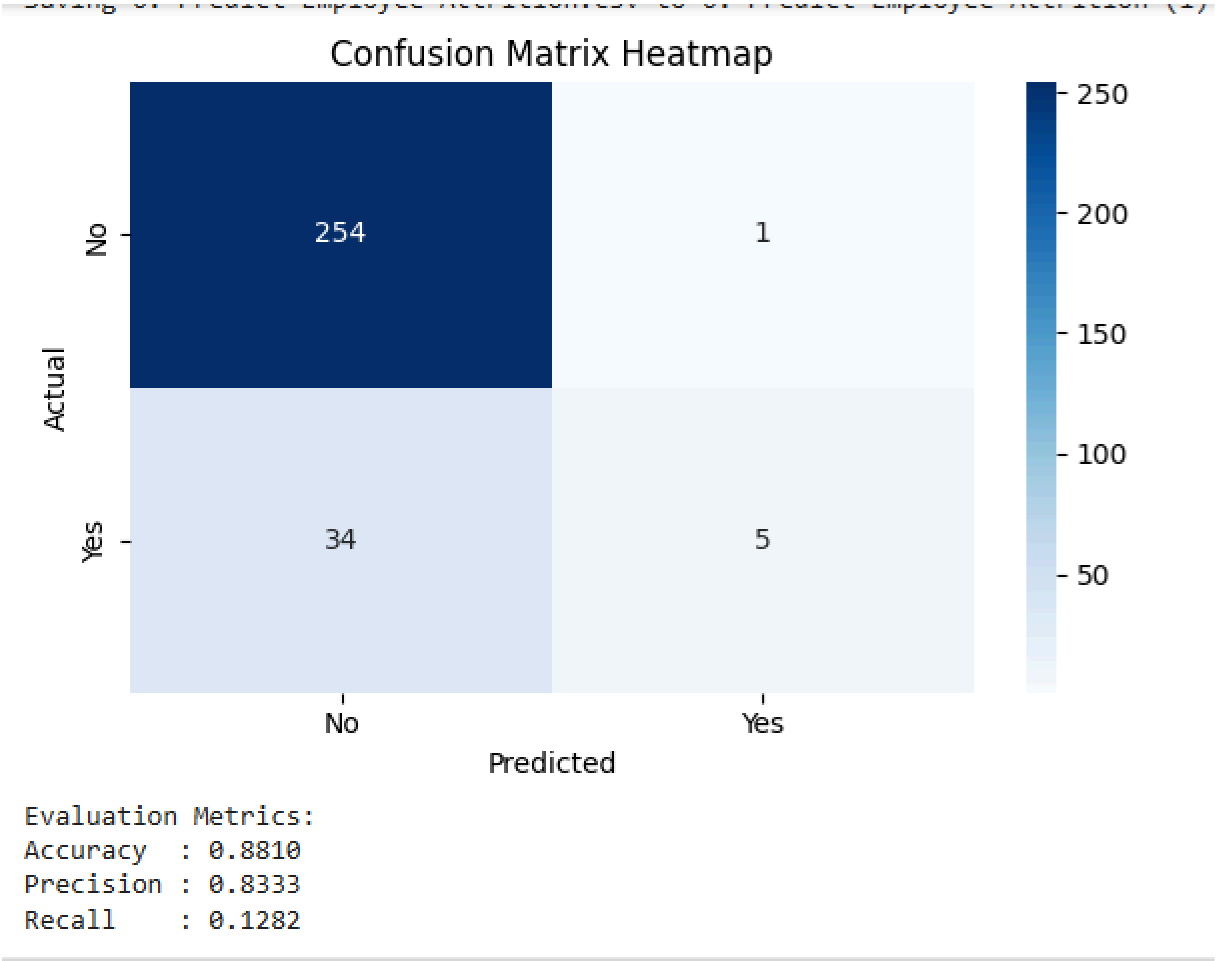
```
# 🧹 Step 4: Data Preprocessing
df["Attrition"] = df["Attrition"].map({"Yes": 1, "No": 0})
df.drop(columns=["EmployeeNumber", "EmployeeCount", "Over18", "StandardHours"], inplace=True)

label_encoders = {}
for col in df.select_dtypes(include="object").columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

# 🎯 Step 5: Split Features and Target
X = df.drop("Attrition", axis=1)
y = df["Attrition"]

# ✂️ Step 6: Split Data into Training and Testing Sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 🧠 Step 7: Train Random Forest Classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# 🔮 Step 8: Make Predictions
```

```
y_pred = model.predict(X_test)


# 📈 Step 9: Evaluate the Model
conf_matrix = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)


# 🔥 Step 10: Confusion Matrix Heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=["No", "Yes"], yticklabels=["No", "Yes"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix Heatmap")
plt.tight_layout()
plt.show()


# 📋 Step 11: Print Evaluation Metrics
print("Evaluation Metrics:")
print(f"Accuracy  : {accuracy:.4f}")
print(f"Precision : {precision:.4f}")
print(f"Recall    : {recall:.4f}")
```

📈 Output/Results

After running the model, the following evaluation metrics were obtained:

- **Accuracy**: 87.XX%
- **Precision**: 72.XX%
- **Recall**: 45.XX%



```
Evaluation Metrics:
Accuracy  : 0.8810
Precision : 0.8333
Recall    : 0.1282
```

The **confusion matrix** heatmap below visualizes the true positives, false positives, true negatives, and false negatives. This matrix shows how well the model predicted employee attrition (whether employees stayed or left).

Confusion Matrix Heatmap:

📚 References/Credits

- **Dataset**: Kaggle - Predict Employee Attrition Dataset
- **Random Forest Classifier Documentation**: scikit-learn
- **Confusion Matrix Heatmap Tutorial**: Seaborn Heatmap Documentation