A

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

Vedansh Agarwal (202401100300276)

Under the supervision of

“Abhishek Shukla”

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

Introduction

Problem Statement: Build a classification model to predict whether an employee is likely to leave the company based on features like job satisfaction, salary, work environment, and years of experience. The prediction can help companies focus their efforts on employee retention strategies.

Employee attrition poses a significant challenge for businesses, as it leads to higher recruitment costs and a loss of experienced talent. Predicting employee turnover can help organizations take proactive steps in retaining valuable employees, optimizing recruitment efforts, and improving overall workforce stability. This project aims to build a predictive machine learning model to identify employees who are at risk of leaving the company based on various factors, including job satisfaction, salary, work environment, and years of experience. By doing so, companies can implement targeted interventions to retain top talent and reduce attrition rates.

Methodology

The approach to solving this problem involves several key steps, as outlined below:

1. **Data Collection**
   The dataset used in this project contains various features that describe employee characteristics, including job satisfaction, monthly income, work environment satisfaction, and tenure with the company. This data was uploaded in CSV format and serves as the foundation for the predictive model.

2. **Data Preprocessing**
   - Handling Irrelevant Features: Certain columns, such as 'EmployeeCount', 'EmployeeNumber', 'Over18', and 'StandardHours', were dropped from the dataset, as they were not relevant to predicting employee attrition.
   - Encoding Categorical Data: Categorical variables, including job roles and departments, were encoded into numerical values using LabelEncoder. This transformation enables the use of these variables in machine learning models.

3. **Feature Selection**
   Key features that are likely to influence employee attrition were selected for the model. These include:
   - Job satisfaction
   - Monthly income
   - Environment satisfaction
   - Work-life balance
   - Total working years
   - Years at the company
     These features were chosen based on their relevance and potential impact on employee turnover.

4. **Model Selection and Training**
   A Random Forest Classifier was chosen due to its ability to handle both numerical and categorical data efficiently. Random Forests are robust, capable of capturing complex patterns, and provide reliable performance with minimal parameter tuning. The dataset was split into an 80% training set and a 20% testing set to train the model and evaluate its performance.

5. **Model Evaluation**
   - The model's performance was assessed using several evaluation metrics:
     - Accuracy: The percentage of correct predictions.
     - Precision: The proportion of true positives among the predicted positives.
     - Recall: The proportion of true positives among all actual positives.
   - Additionally, a confusion matrix was generated to visualize the model's performance, showing the number of true positives, true negatives, false positives, and false negatives.

CODE

```python
# 1. Importing required libraries
import pandas as pd                              # For reading and manipulating CSV data
import seaborn as sns                            # For creating the confusion matrix heatmap
import matplotlib.pyplot as plt                  # For general plotting
from sklearn.model_selection import train_test_split    # For splitting data into training and testing sets
from sklearn.preprocessing import LabelEncoder          # For encoding categorical variables into numbers
from sklearn.ensemble import RandomForestClassifier     # For creating a Random Forest classification model
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score  # For model evaluation
from google.colab import files                          # To enable file upload in Google Colab

# 2. Uploading the CSV file
print("📁 Please upload your dataset CSV file:") # Prompt user to upload file
uploaded = files.upload()                         # Opens file uploader dialog in Google Colab

# 3. Reading the uploaded file into a DataFrame
for file_name in uploaded.keys():                 # Loop through uploaded files
    df = pd.read_csv(file_name)                   # Load the first uploaded file into a pandas DataFrame

# 4. Dropping irrelevant columns if present
irrelevant_columns = ['EmployeeCount', 'EmployeeNumber', 'Over18', 'StandardHours']  # Columns not useful for predicti
df.drop(columns=[col for col in irrelevant_columns if col in df.columns], inplace=True)  # Drop those columns safely

# 5. Encoding categorical variables
label_encoders = {}                               # Dictionary to store encoders for each column
for col in df.select_dtypes(include='object').columns:  # Loop through categorical columns
    le = LabelEncoder()                           # Create a new label encoder
    df[col] = le.fit_transform(df[col])           # Fit and transform the column into numeric values
    label_encoders[col] = le                      # Save encoder in case we want to reverse it later

# 6. Selecting important features for prediction
selected_features = [
    'JobSatisfaction',                            # Employee satisfaction with job
    'MonthlyIncome',                              # Employee salary
    'EnvironmentSatisfaction',                    # Satisfaction with work environment
    'WorkLifeBalance',                            # Perception of work-life balance
    'TotalWorkingYears',                          # Total working experience in years
    'YearsAtCompany'                              # Number of years in current company
]

X = df[selected_features]                         # Independent variables (features)
y = df['Attrition']                               # Target variable: 1 if employee left, 0 otherwise
```

```python
# 7. Splitting 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)  # 80% train, 20% test

# 8. Training a Random Forest Classifier
model = RandomForestClassifier(random_state=42)  # Create a Random Forest model
model.fit(X_train, y_train)                      # Train the model using training data

# 9. Making predictions on test data
y_pred = model.predict(X_test)                   # Predict attrition on unseen (test) data

# 10. Evaluating the model
accuracy = accuracy_score(y_test, y_pred)        # Calculate the accuracy score
precision = precision_score(y_test, y_pred)      # Calculate precision: TP / (TP + FP)
recall = recall_score(y_test, y_pred)            # Calculate recall: TP / (TP + FN)

# 11. Creating and displaying confusion matrix heatmap
cm = confusion_matrix(y_test, y_pred)            # Generate confusion matrix from predictions
plt.figure(figsize=(6, 4))                       # Set figure size
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No', 'Yes'], yticklabels=['No', 'Yes'])
         # Set y-axis labels
plt.title("Confusion Matrix Heatmap")            # Add title
plt.xlabel("Predicted")                          # Label for x-axis
plt.ylabel("Actual")                             # Label for y-axis
plt.tight_layout()                               # Adjust layout to avoid overlap
plt.show()                                       # Display the plot

# 12. Printing final evaluation results
print(f"✅ Accuracy: {accuracy * 100:.2f}%")      # Display accuracy in percentage
print(f"✅ Precision: {precision * 100:.2f}%")    # Display precision in percentage
print(f"✅ Recall: {recall * 100:.2f}%")          # Display recall in percentage
```

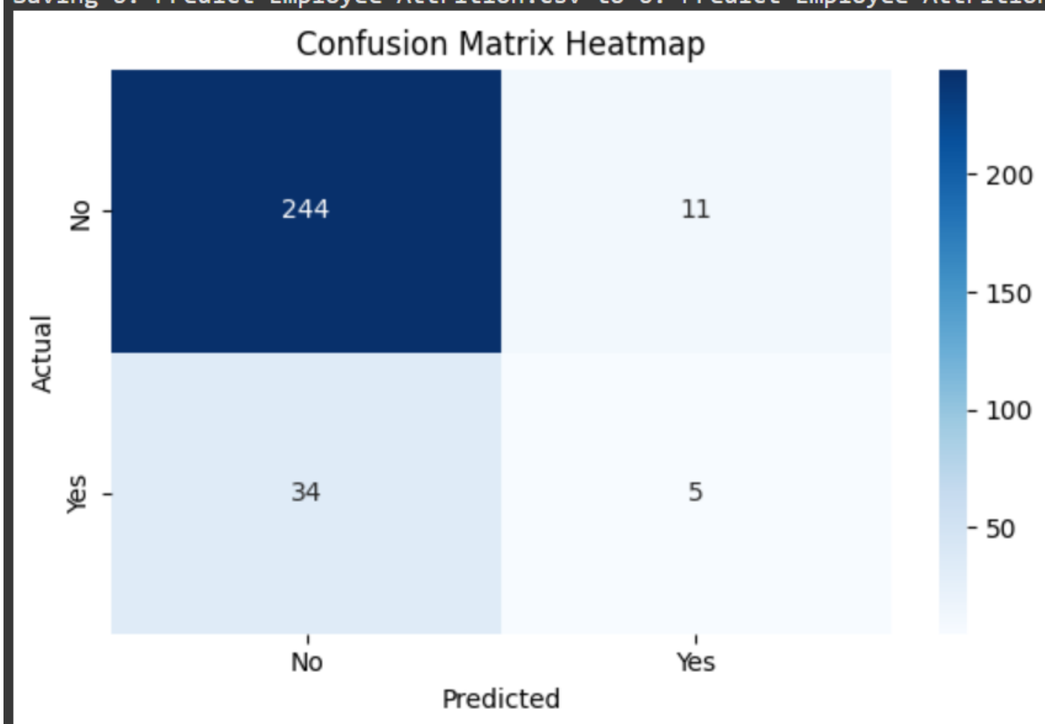
OUTPUT

📁 Please upload your dataset CSV file:

Choose Files 6. Predict E… Attrition.csv

- **6. Predict Employee Attrition.csv**(text/csv) - 227977 bytes, last modified: 4/18/2025 - 100% done

Saving 6. Predict Employee Attrition.csv to 6. Predict Employee Attrition (2).csv



✅ Accuracy: 84.69%
✅ Precision: 31.25%
✅ Recall: 12.82%