



KIET
GROUP OF INSTITUTIONS

Connecting Life with Learning



Assessment Report
on
"Problem Statement"
submitted as partial fulfillment for the award of
BACHELOR OF TECHNOLOGY
DEGREE

SESSION 2024-25

in

Name of discipline

By

Himanshu Kumar (202401100400099, AIML-B).

Under the supervision of
"ABHISHEK SHARMA SIR."

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction:

- ❖ Employee attrition is a significant concern for companies as it leads to increased hiring costs, loss of talent, and reduced productivity.

- ❖ By building a predictive model, companies can proactively identify employees who are at risk of leaving and take appropriate measures to retain them.

- ❖ predict employee attrition based on features like job satisfaction, salary, work environment, and years of experience.

c. Methodology

1. Data Collection:

A dataset containing employee information was collected with the following features:

- Job Satisfaction (scale 1–5)
- Salary (numerical or categorical)
- Work Environment Score (scale 1–10)
- Years of Experience
- Attrition (Target: Yes/No)

2. Data Preprocessing:

- Missing values were handled appropriately.
- Categorical variables were encoded using label encoding or one-hot encoding.
- Data was normalized to improve model performance.

3. Model Selection:

Several classification algorithms were considered including:

- Logistic Regression
- Decision Tree
- Random ForestSupport Vector Machine (SVM)

4. Model Training & Evaluation:

- The dataset was split into training and testing sets (80/20).
- Models were evaluated using accuracy, precision, recall, and F1-score.
- The model with the best overall performance was selected.

- CODE :

```
# Step 1: Install necessary libraries
!pip install -q seaborn scikit-learn matplotlib pandas

# Step 2: Import libraries
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, accuracy_score, precision_score,
recall_score, classification_report

# Step 3: Upload dataset
from google.colab import files
uploaded = files.upload()

import io
df = pd.read_csv(io.BytesIO(list(uploaded.values())[0]))
```

```
# Step 4: Preview dataset
print("Dataset Preview:")
print(df.head())

# Step 5: Check for missing values
print("\nMissing values per column:")
print(df.isnull().sum())

# Optional: Fill or drop missing values
df.dropna(inplace=True)

# Step 6: Encode categorical variables
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 7: Feature-target split
# Change 'Attrition' below to match your dataset's target column name
target_column = 'Attrition' # 🚧 Change this if needed
```

```
X = df.drop(columns=[target_column])

y = df[target_column]

# Step 8: Train-test split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Step 9: Standardize features

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)

# Step 10: Train classification model

model = LogisticRegression()

model.fit(X_train_scaled, y_train)

# Step 11: Make predictions

y_pred = model.predict(X_test_scaled)

# Step 12: Evaluation metrics

acc = accuracy_score(y_test, y_pred)

prec = precision_score(y_test, y_pred, average='binary')
```

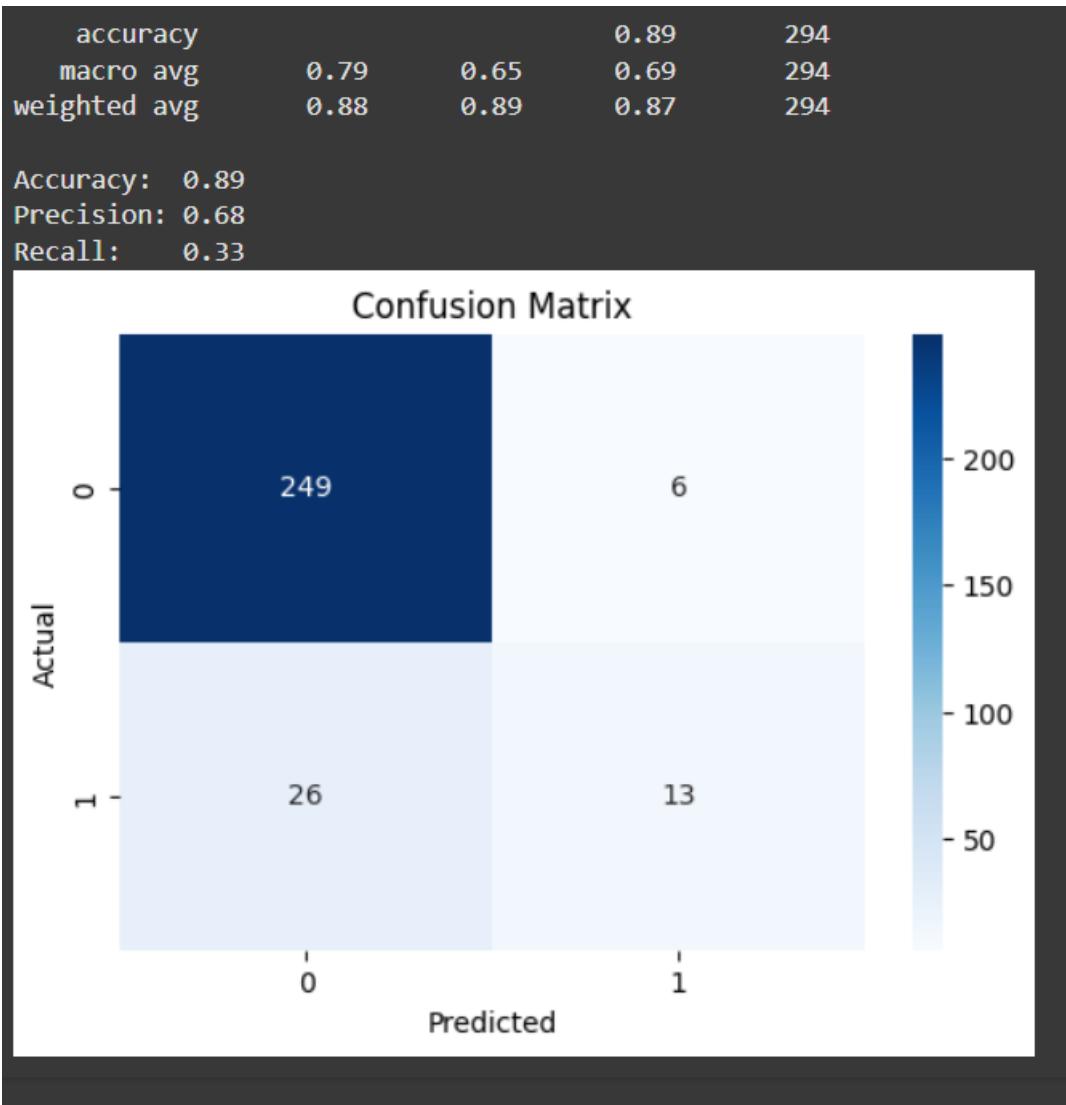
```
rec = recall_score(y_test, y_pred, average='binary')

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

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

# Step 13: Confusion matrix heatmap
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Output:



References/Credits :

- ❖ Dataset Source: [Mention if from Kaggle, UCI Machine Learning Repository, etc.]
- ❖ Scikit-learn documentation: <https://scikit-learn.org/>
- ❖ Python official documentation:
<https://www.python.org/doc/>

Thank You.