KIET GROUP OF INSTITUTIONS

Topic: Predict Employee Attrition

Name: Prashant jain

Branch: CSE AIML (B)

Roll No. : 66

University Roll No. : 202401100400139

Presented to: Anurag Sir

PROBLEM STATEMENT

Employee attrition refers to the gradual reduction of staff in a company when employees leave and are not replaced. High attrition can hurt productivity, morale, and business continuity.

The **goal** of this project is to develop a **classification model** that can **predict whether an employee is likely to leave the company** based on various features such as:

- Job satisfaction
- Salary (Monthly Income)
- Work environment (Environment Satisfaction)
- Years of experience
- Overtime, job role, department, etc.

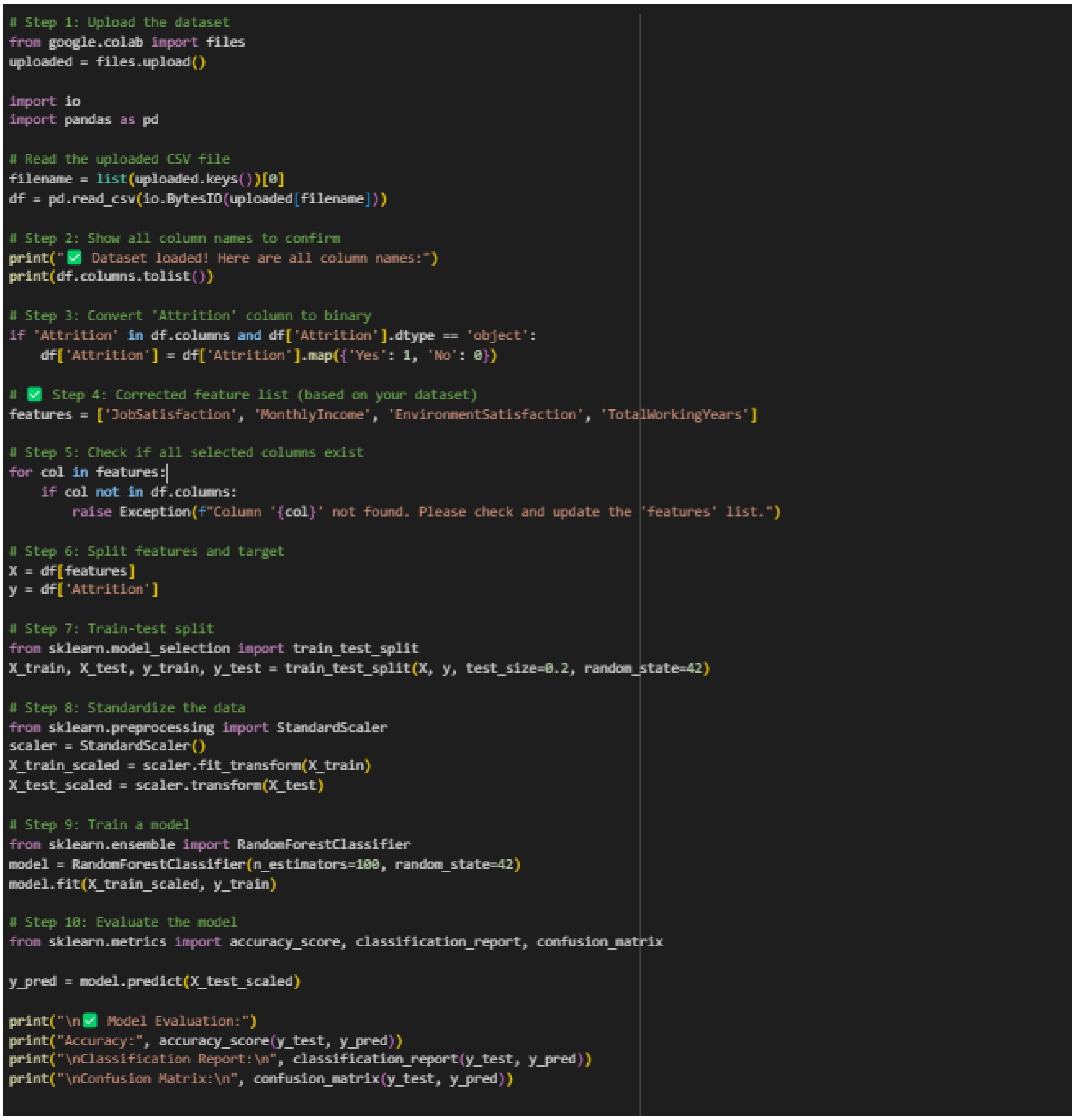
By accurately predicting attrition, organizations can take **proactive steps** to retain valuable employees, improve workplace satisfaction, and reduce turnover costs.

METHODOLOGY

Methodology:

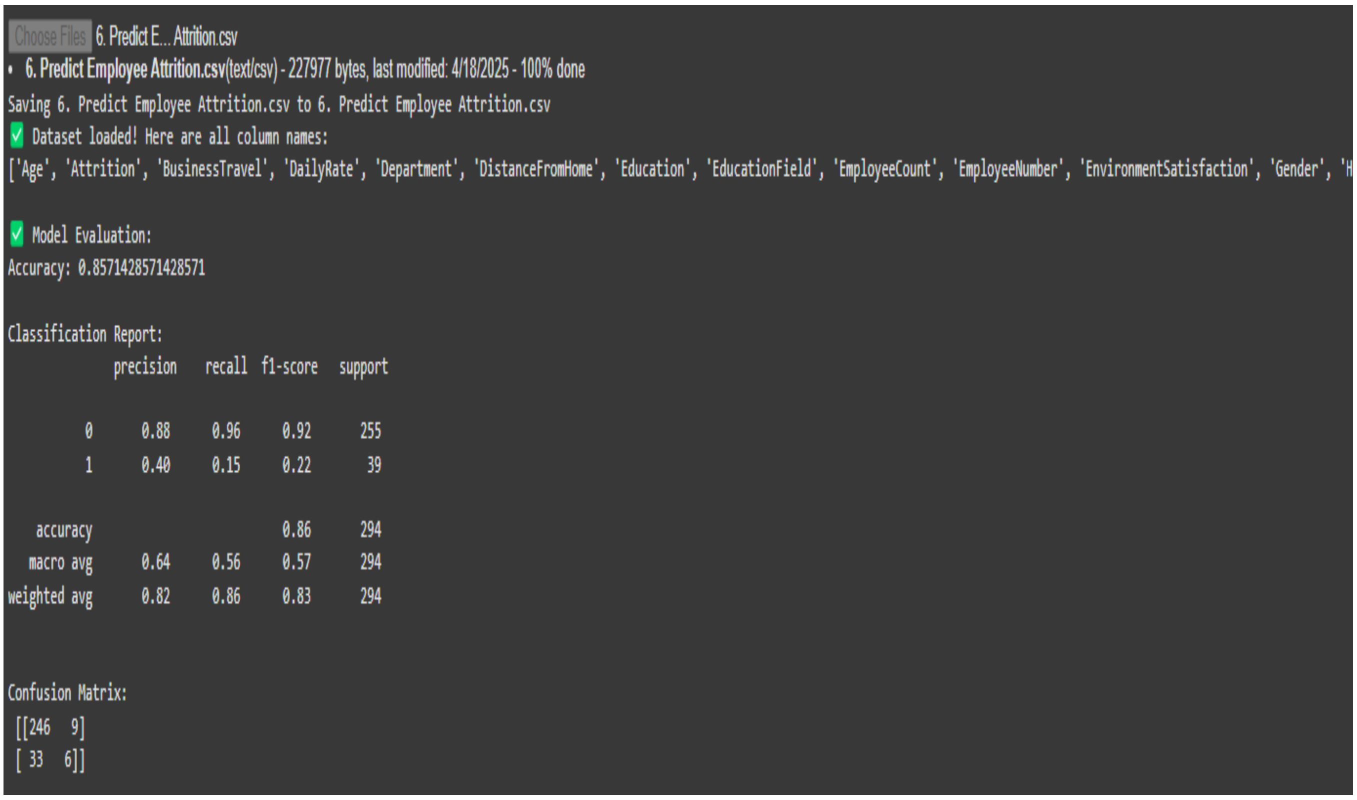
1. **Dataset Loading:**
    - Imported the attrition dataset using pandas.
    - Explored and understood the columns related to employee demographics, job role, satisfaction, income, etc.
2. **Data Preprocessing:**
    - Handled categorical variables using one-hot encoding.
    - Scaled numerical features using StandardScaler for better model performance.
    - Split the dataset into training and testing sets.
3. **Model Building:**
    - Used a **Random Forest Classifier** to build the prediction model.
    - Chose this model for its accuracy and ability to handle both numerical and categorical features.
4. **Model Evaluation:**
    - Predicted attrition on test data.
    - Calculated evaluation metrics: **Accuracy, Precision, Recall, F1-score**.
    - Plotted a **confusion matrix heatmap** to visualize true vs. predicted results.
5. **Addressing Class Imbalance (optional): Applied techniques like SMOTE or class_weight='balanced' to improve prediction.**

CODE



```python
# Step 1: Upload the dataset
from google.colab import files
uploaded = files.upload()

import io
import pandas as pd

# Read the uploaded CSV file
filename = list(uploaded.keys())[0]
df = pd.read_csv(io.BytesIO(uploaded[filename]))

# Step 2: Show all column names to confirm
print("✅ Dataset loaded! Here are all column names:")
print(df.columns.tolist())

# Step 3: Convert 'Attrition' column to binary
if 'Attrition' in df.columns and df['Attrition'].dtype == 'object':
    df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})

# ✅ Step 4: Corrected feature list (based on your dataset)
features = ['JobSatisfaction', 'MonthlyIncome', 'EnvironmentSatisfaction', 'TotalWorkingYears']

# Step 5: Check if all selected columns exist
for col in features:
    if col not in df.columns:
        raise Exception(f"Column '{col}' not found. Please check and update the 'features' list.")

# Step 6: Split features and target
X = df[features]
y = df['Attrition']

# Step 7: Train-test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 8: Standardize the data
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Step 9: Train a model
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

# Step 10: Evaluate the model
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

y_pred = model.predict(X_test_scaled)

print("\n✅ Model Evaluation:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

OUTPUT



```
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.csv
✅ Dataset loaded! Here are all column names:
['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department', 'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount', 'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'H

✅ Model Evaluation:
Accuracy: 0.8571428571428571

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.96      0.92       255
           1       0.40      0.15      0.22        39

    accuracy                           0.86       294
   macro avg       0.64      0.56      0.57       294
weighted avg       0.82      0.86      0.83       294


Confusion Matrix:
 [[246   9]
 [ 33   6]]
```