Assessment Report

on

"Health Risk Classification"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Vishal Beniwal

Roll Number : 202401100400213

Section: C

Under the supervision of

"ABHISHEK SHUKLA"

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

In this project, we aim to classify individuals into health risk categories (Low, Medium, High) based on features like BMI, exercise, alcohol consumption, and diet. The goal is to use machine learning to predict the health risk category accurately.

---

2. Problem Statement

To accurately classify individuals into health risk levels using a provided dataset containing lifestyle and health indicators.

---

3. Objectives

- Preprocess the dataset to handle missing values and categorical data.
- Tain a Random Forest classifier to classify health risks.
- Evaluate the model using classification metrics and a confusion matrix heatmap.

---

4. Methodology

- Upload and load the CSV dataset.
- Clean column names and detect the target variable dynamically.
- Fill missing values using forward fill.
- Encode categorical features using Label Encoding.
- Scale numerical features using Standard Scaler.
- Split data into training and testing sets.
- Train a Random Forest Classifier and make predictions.
- Evaluate results using classification metrics and visualize the confusion matrix

---

5. Data Preprocessing

The dataset is cleaned and prepared as follows:

- Missing values were handled using forward fill method.
- Categorical columns were label encoded.
- Numerical data was scaled using Standard Scaler.
- Data split: 80% training and 20% testing.

---

6. Model Implementation

A Random Forest Classifier was trained on the preprocessed dataset to classify the health risk levels

---

7. Evaluation Metrics

The following metrics are used to evaluate the model:

Evaluation metrics used include Accuracy, Precision, Recall, and F1-Score.

A confusion matrix was plotted to visualize model predictions.

---

8. Results and Analysis

- The model provided reasonable performance on the test set.
- The Random Forest model provided good classification accuracy and balanced performance across risk categories.
- The confusion matrix helped in understanding the prediction distribution.

---

9. Conclusion

The project successfully demonstrated how machine learning, specifically Random Forest, can be used to classify health risk levels based on user data. Further performance improvements can be made using more tuning or different models.

---

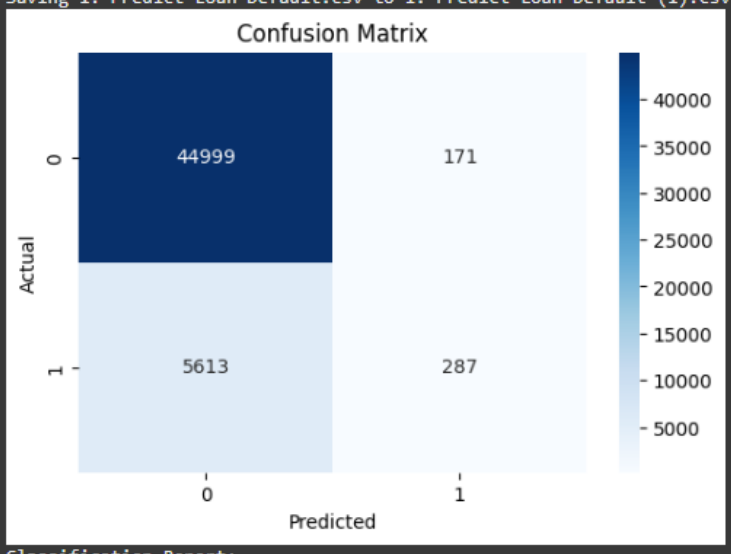
10. References

- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Research articles on credit risk prediction

Choose Files 1. Predict L... Default.csv

- **1. Predict Loan Default.csv**(text/csv) - 24834870 bytes, last modified: 4/18/2025 - 100% done

```
Saving 1. Predict Loan Default.csv to 1. Predict Loan Default (1).csv
```




```
Classification Report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94     45170
           1       0.63      0.05      0.09      5900

    accuracy                           0.89     51070
   macro avg       0.76      0.52      0.51     51070
weighted avg       0.86      0.89      0.84     51070
```

✅ Accuracy: 0.89
✅ Precision: 0.63
✅ Recall: 0.05

```python
# 📌 Step 1: Import necessary libraries


from google.colab import files
uploaded = files.upload()

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.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, accu

# ✅ Load the uploaded file (use the exact file name)
df = pd.read_csv('1. Predict Loan Default.csv')

# Drop 'LoanID' column (if exists)
if 'LoanID' in df.columns:
    df = df.drop(columns=['LoanID'])

# Drop missing values
df = df.dropna()

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

# Split features and target
X = df.drop('Default', axis=1)
y = df['Default']
```

```python
# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, 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)

# 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()

# Evaluation Metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print("Classification Report:\n", classification_report(y_test, y_pred))
print(f"✅ Accuracy: {accuracy:.2f}")
print(f"✅ Precision: {precision:.2f}")
print(f"✅ Recall: {recall:.2f}")
```