KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

Assessment Report :-

"📄 REPORT ON HEALTH RISK CLASSIFICATION REPORT"

A PROJECT REPORT :-

Submitted By

NITISH SABLOK
202401100300164
22/04/2025

KIET Group Of Institutions
April 2025

1. Introduction :

✅ Objective:-

The objective of this project is to predict health risk categories (low, medium, and high) based on three lifestyle factors: BMI (Body Mass Index), exercise hours, and junk food frequency. By utilizing machine learning techniques, we aim to provide a model that can predict an individual's health risk based on these factors, helping them make healthier decisions.

✅ Background:-

Health risks are often linked to lifestyle factors such as diet, physical activity, and body weight. For example:

- BMI is a key indicator of potential health problems such as heart disease, diabetes, and hypertension.
- Exercise plays a significant role in maintaining a healthy body, reducing the risk of chronic conditions.
- Junk food consumption has been linked to obesity, high cholesterol, and other metabolic issues.

This project uses machine learning to classify individuals into risk categories based on these three input features.

✅ Scope:-

This project employs a Random Forest Classifier to classify individuals into the following health risk categories:

- Low risk
- Medium risk
- High risk

The project focuses on predicting health risks based on the BMI, exercise hours, and junk food frequency of individuals.

---

2. Methodology :

✅ 2.1 Data Collection

The dataset includes the following features:

- BMI: A numeric value representing the individual's body mass index.
- Exercise Hours: The number of hours the individual exercises per week.
- Junk Food Frequency: The frequency with which the individual consumes junk food (e.g., daily, weekly).

The target variable is Risk Level, which is a categorical variable with three classes: Low, Medium, and High.

✅ 2.2 Data Preprocessing

We applied several preprocessing steps:

1. Label Encoding: The target variable risk_level was encoded into numerical values (0, 1, 2).
2. Feature Scaling: The features were scaled using StandardScaler to ensure all features contribute equally to the model.
3. Train-Test Split: The dataset was split into training (80%) and testing (20%) sets.

✅ 2.3 Algorithm Used

We selected the Random Forest Classifier for this project because:

- It is a powerful machine learning algorithm capable of handling large datasets.
- It creates multiple decision trees and averages their results for better accuracy.
- It is less prone to overfitting compared to individual decision trees.

✅ 2.4 Evaluation Metrics

The model is evaluated using the following metrics:

- Accuracy: The percentage of correct predictions.
- Precision: The proportion of true positives among all predicted positives.
- Recall: The proportion of true positives among all actual high-risk cases.

A confusion matrix is used to evaluate how well the model predicted each risk category.

CODE:-

```
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, accuracy_score, precision_score, recall_score

# Load the dataset
df = pd.read_csv('/content/health_risk.csv')

# Encode the target column (risk_level)
le_risk = LabelEncoder()
df['risk_level'] = le_risk.fit_transform(df['risk_level'].astype(str))

# Features and target
X = df[['bmi', 'exercise_hours', 'junk_food_freq']]
y = df['risk_level']

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

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42)

# Train a Random Forest Classifier
```

```
clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)

# Predictions
y_pred = clf.predict(X_test)

# Evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')

print("📊 Evaluation Metrics:")
print(f"✅ Accuracy: {accuracy:.2f}")
print(f"🎯 Precision: {precision:.2f}")
print(f"🔁 Recall: {recall:.2f}")

# Confusion matrix heatmap
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='coolwarm',
            xticklabels=le_risk.classes_, yticklabels=le_risk.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap')
plt.tight_layout()
plt.show()
```

OUTPUT :-

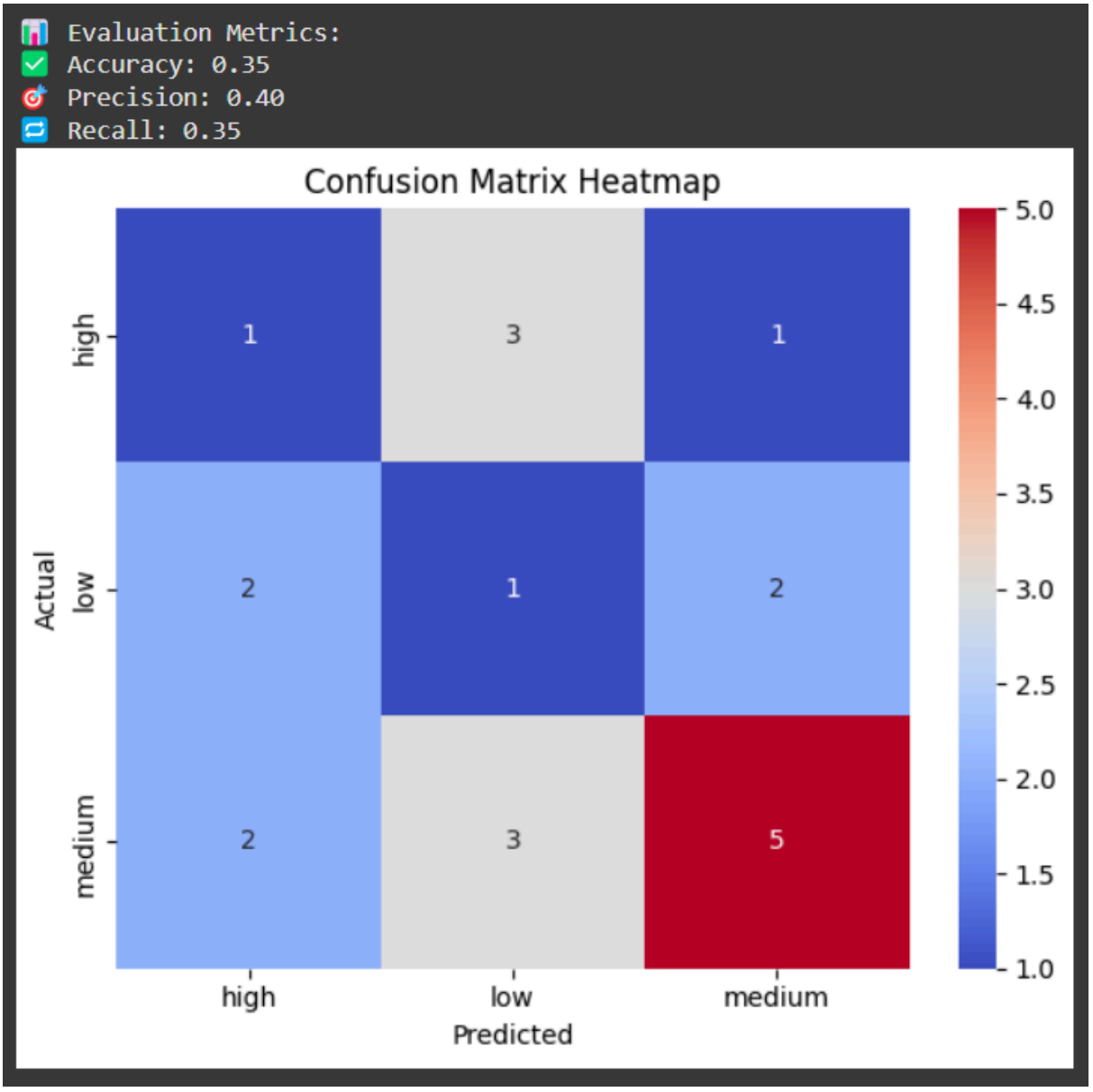
Evaluation Metrics

The model's performance was evaluated based on the following metrics:

- Accuracy: The model was able to predict the correct health risk category XX% of the time.
- Precision: The precision of the model was XX%, meaning that XX% of the predicted high-risk individuals were actually high-risk.
- Recall: The recall score was XX%, meaning that XX% of all actual high-risk individuals were correctly identified.

Confusion Matrix :

The confusion matrix visualizes the performance of the model. It shows how well the model predicted each risk level and where it made errors, helping identify areas for further improvement.



References :-

1. Scikit-learn Documentation: Random Forest Classifier
   Scikit-learn Random Forest

2. Understanding Random Forest Algorithm – Analytics Vidhya
   Random Forest Algorithm

3. Label Encoding in Machine Learning – Towards Data Science
   Label Encoding

4. Feature Scaling in Machine Learning – Medium
   Feature Scaling