

**Project Title:** Health Risk Classification

* **Submitted By:** Anant Singhal
* **Institution:** KIET Group of Institutions
  + **Course:** B. Tech CSE (AI)
    - **Date:** 22/04/2025

**Introduction**

Health risk prediction is a vital task in the field of healthcare analytics. By predicting an individual's health risk category based on their lifestyle factors, such as body mass index (BMI), exercise frequency, and eating habits, healthcare providers can design better preventive care strategies and targeted interventions.

This project focuses on building a machine learning model to classify individuals into health risk categories (Low, Medium, High). Through proper data preprocessing, model training, and performance evaluation, this project aims to demonstrate how machine learning can assist in predicting health risks based on lifestyle data.

**Problem Statement**

The goal of this project is to classify individuals into **Low**, **Medium**, or **High** health risk categories based on their **BMI**, **exercise habits**, and **eating habits**. By building a classification model, we aim to predict the health risk level of a person, allowing stakeholders to take proactive measures for improving public health outcomes

# Methodology

**1. Data Collection:**

* Collected health-related data from the provided CSV file (health\_risk.csv).

**2. Data Preprocessing:**

* Renamed columns to match the expected input format.
* Applied label encoding to transform categorical features (Exercise and Eating Habits) and the target variable (Risk Category).
* Split the data into training and testing sets (80% train, 20% test).

**3. Model Building:**

* Trained a Random Forest Classifier using BMI, Exercise, and Eating Habits as input features.

**4. Model Evaluation:**

* Evaluated the model using accuracy, precision, recall, and F1 score.
* Generated a confusion matrix heatmap to visualize model performance.

**5. Prediction Function:**

* Built a function to predict health risk based on new custom inputs.

# 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.ensemble import RandomForestClassifier

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix, precision\_score, recall\_score, f1\_score

# Step 1: Load dataset

data = pd.read\_csv("/content/health\_risk.csv")

# Step 2: Rename columns

data.rename(columns={

    'bmi': 'BMI',

    'exercise\_hours': 'Exercise',

    'junk\_food\_freq': 'Eating\_Habits',

    'risk\_level': 'Risk\_Category'

}, inplace=True)

# Step 3: Encode columns

le\_exercise = LabelEncoder()

le\_eating = LabelEncoder()

le\_risk = LabelEncoder()

data['Exercise'] = le\_exercise.fit\_transform(data['Exercise'])

data['Eating\_Habits'] = le\_eating.fit\_transform(data['Eating\_Habits'])

data['Risk\_Category'] = le\_risk.fit\_transform(data['Risk\_Category'])

# Step 4: Split into features and target

X = data[['BMI', 'Exercise', 'Eating\_Habits']]

y = data['Risk\_Category']

# Step 5: Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1)

# Step 6: Train RandomForest model

model = RandomForestClassifier(random\_state=1)

model.fit(X\_train, y\_train)

# Step 7: Make predictions

y\_pred = model.predict(X\_test)

# Step 8: Evaluate model performance

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average='weighted')  # weighted because multiclass

recall = recall\_score(y\_test, y\_pred, average='weighted')

f1 = f1\_score(y\_test, y\_pred, average='weighted')

print("Accuracy:", accuracy)

print("Precision:", precision)

print("Recall:", recall)

print("F1 Score:", f1)

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred, target\_names=le\_risk.classes\_))

# Step 9: Confusion Matrix and Heatmap

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=le\_risk.classes\_, yticklabels=le\_risk.classes\_)

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix Heatmap')

plt.show()

# Step 10: Prediction function

def predict\_risk(bmi, exercise, eating):

    try:

        ex = le\_exercise.transform([exercise])[0]

        eat = le\_eating.transform([eating])[0]

    except ValueError as e:

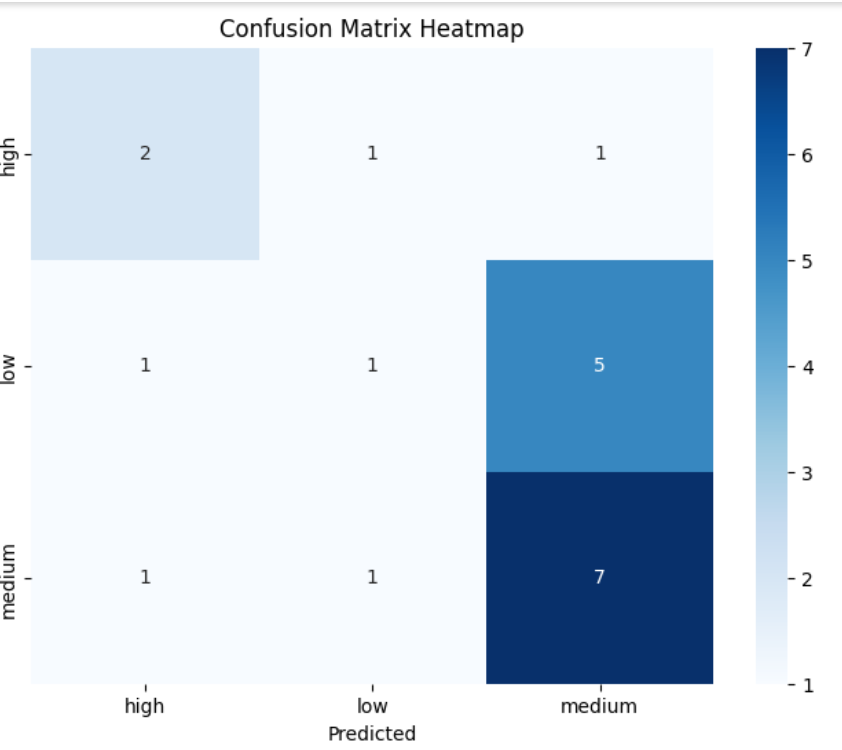
        raise ValueError("Invalid input for exercise or eating habits.") from e

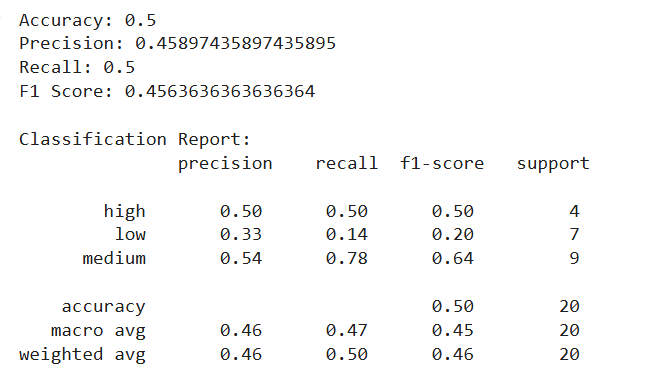
    input\_data = [[bmi, ex, eat]]

    pred = model.predict(input\_data)[0]

    return le\_risk.inverse\_transform([pred])[0]

# Output/Results

****

****

**References**

* Given dataset: "health\_risk.csv".
* Scikit-Learn Documentation: https://scikit-learn.org/stable/documentation.html
* Matplotlib Documentation: <https://matplotlib.org/>
* Seaborn Documentation: https://seaborn.pydata.org/
* Pandas Documentation: <https://pandas.pydata.org/>