Health Risk Classification Report

A PROJECT REPORT

Submitted by

SAMVEG JAIN

(202401100300214)

in partial fulfillment for the award of the degree

of

Bachelor of Technology

KIET GROUP OF INSTITUTIONS GHAZIABAD

April 2025

Introduction :

With the growing emphasis on preventive healthcare, the ability to assess an individual's health risk based on lifestyle factors has become increasingly valuable. This report presents a machine learning-based approach to classify health risk levels—categorized as low, medium, or high—by analysing key personal health indicators such as Body Mass Index (BMI), exercise frequency, and junk food consumption habits.

The primary objective of this project is to develop a predictive model that can accurately determine a person's risk category using these measurable features. By leveraging historical data and applying a Naive Bayes classification algorithm, the model aims to identify patterns that correlate with varying levels of health risk. Such a tool can be instrumental in supporting early intervention strategies, raising awareness about the impact of lifestyle choices, and guiding individuals toward healthier behaviours.

This report outlines the methodology followed, the dataset used, model implementation, performance evaluation, and the insights drawn from the results.

Methodology :

The methodology for this project involves several key stages, from data acquisition and preprocessing to model development and evaluation. Each step is designed to ensure the accuracy and reliability of the predictive model.

1. Data Collection

The dataset used for this study consists of individual health records including three primary features:

- BMI (Body Mass Index): A numerical indicator of body fat based on height and weight.
- Exercise Hours: The average number of hours of physical activity per week.
- Junk Food Frequency: The number of times junk food is consumed per week. The target variable is Risk Level, categorized as *low*, *medium*, or *high*.

2. Data Preprocessing

The dataset was reviewed for completeness and consistency. Since all features were numerical and no missing values were detected, minimal preprocessing was required. Feature scaling was not necessary for the Naive Bayes classifier, as it assumes Gaussian distribution for continuous data.

3. Model Selection

A Naive Bayes classifier was selected for this classification task due to its simplicity, interpretability, and effectiveness in probabilistic classification problems. Specifically, the Gaussian Naive Bayes variant was used to accommodate the continuous nature of the input features.

4. Training and Testing

The dataset was split into training and testing subsets using an 80:20 ratio. The training set was used to fit the model, while the testing set was used to evaluate performance. This approach ensures that the model is tested on unseen data, providing a realistic estimate of its generalization ability.

5. Evaluation Matrix

The performance of the model was evaluated using the following matrix :

- Accuracy: The overall percentage of correctly classified instances.
- Confusion Matrix: A tabular representation of the predicted versus actual classifications.
- Classification Report: Includes precision, recall, and F1-score for each risk category.

Code :

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report,
accuracy_score, confusion_matrix

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

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

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Initialize the Naive Bayes classifier
model = GaussianNB()
```

```
# Train the model
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

Output Result

```
Accuracy: 0.4

Classification Report:
              precision    recall  f1-score   support

        high       0.50      0.20      0.29         5
         low       0.00      0.00      0.00         5
      medium       0.47      0.70      0.56        10

    accuracy                           0.40        20
   macro avg       0.32      0.30      0.28        20
weighted avg       0.36      0.40      0.35        20

Confusion Matrix:
 [[1 1 3]
 [0 0 5]
 [1 2 7]]
```

References and credits

1) Scikit-learn Documentation
   Scikit-learn: Machine Learning in Python
   https://scikit-learn.org/stable/documentation.html
   Used for implementing the Naive Bayes classifier and model evaluation techniques.
2) Pandas Documentation
   Pandas: Python Data Analysis Library
   https://pandas.pydata.org/docs/
   Utilized for data manipulation and preprocessing.
3) BMI Information
   World Health Organization (WHO): Body Mass Index (BMI) classification guidelines
   https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight

Credits

- Data preprocessing and modeling: Samveg Jain
- Supervision/Guidance : Mayank Lakhotia Sir