Health Risk Classification Using Machine Learning

Assessment Models Title Page

Project Name: Health Risk Classification Using Machine Learning

Submitted By: ABHISHEK KUMAR

Roll No: 202401100300007

Date: 22/04/2025

Instructor: MR> BIKKI KUMAR

---

1. Introduction

Health risk classification is a crucial task in preventive healthcare analytics. Using patient-related data, we aim to categorize individuals into various risk levels (e.g., Low, Medium, High) based on key features such as age, weight, smoking habits, or other indicators. This classification can help in prioritizing medical attention and improving health outcomes.

In this project, we use a machine learning model to perform classification based on provided health data. We evaluate the model using key metrics like accuracy, precision, recall, and visualize the results using a confusion matrix heatmap.

---

2. Methodology

The approach involves the following steps:

1. **Data Preprocessing:** Load and inspect the dataset, separate features and labels.
2. **Data Splitting:** Split the dataset into training and test sets (80-20 ratio).
3. **Model Training:** Use a Random Forest Classifier to learn patterns from the training data.

4. **Prediction:** Predict the health risk levels for the test data.
5. **Evaluation:** Calculate performance metrics like accuracy, precision, and recall.

6. **Visualization:** Generate a heatmap of the confusion matrix to visualize model performance.

---

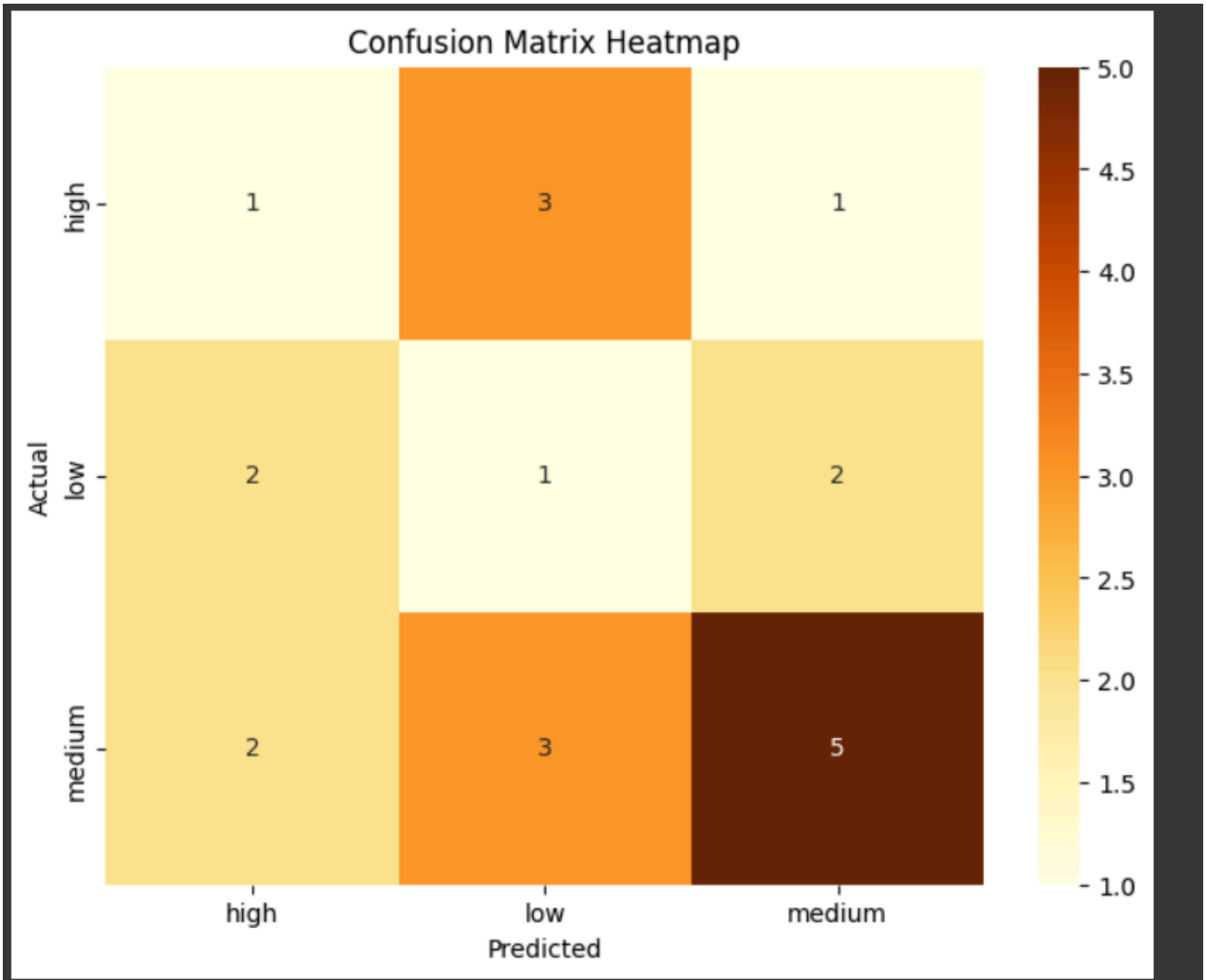
3. Code Implementation

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score

# 1. Load the dataset
df = pd.read_csv('/mnt/data/health_risk.csv')

# 2. Basic preprocessing (assuming target is 'RiskLevel')
X = df.drop('risk_level', axis=1)
y = df['risk_level']

# Optional: Convert categorical variables if any
# X = pd.get_dummies(X)

# 3. Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# 4. Train a classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# 5. Make predictions
y_pred = model.predict(X_test)

# 6. Confusion matrix and heatmap
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='YlOrBr', xticklabels=model.classes_,
yticklabels=model.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap')
plt.show()

# 7. 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(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
```

---

4. Output and Result



5. References / Credits

Dataset Source: https://drive.google.com/file/d/1CGMrCE_5FyqZjKHgp6nPgxJg7PpxST2n/view?usp=drive_link

Scikit-learn Documentation: https://scikit-learn.org/stable/

Seaborn Documentation: https://seaborn.pydata.org/
Libraries Used: pandas as pd, seaborn as sns, matplotlib.pyplot as plt

Image Credit: Wikipedia Commons (used for illustrative purposes)