

PREDICTIVE MODEL - CODED IN PYTHON

AYUSH CHALISHAJAR
2023-2024 CYSF

This code implements a matching learning algorithm model for analyzing health indicator data of multiple patients. The algorithm takes as input various health indicators such as heart rate, oxygen saturation, blood pressure, cholesterol levels, and glucose levels for each patient. It then compares these values against predefined acceptable ranges for each health indicator. If any values fall outside the acceptable ranges, the algorithm flags the patient as high risk and identifies the specific health indicator(s) with anomalies.

The model presented in the code is a simplified version of what might be considered a basic anomaly detection system for health indicators. While it provides a functional demonstration of how such a system could work, there are several key differences between this simplified model and a real-world anomaly detection algorithm used in healthcare:

1. **Feature Engineering and Selection:** In a real-world scenario, extensive feature engineering and selection would be necessary to determine which health indicators are most relevant for anomaly detection. This process involves analyzing large datasets to identify meaningful patterns and correlations between various health indicators and health outcomes. Advanced techniques such as dimensionality reduction and feature selection algorithms may be employed to extract the most informative features from the data.
2. **Statistical Modeling:** Real anomaly detection algorithms often employ sophisticated statistical models to characterize the distribution of normal and abnormal data points for each health indicator. These models may include Gaussian distributions, kernel density estimation, or other probabilistic techniques to quantify the likelihood of observing a given data point under normal conditions. Deviations from expected distributions can then be flagged as anomalies.
3. **Dynamic Thresholding:** In practice, the acceptable ranges for health indicators may vary depending on factors such as age, gender, medical history, and comorbidities. Real anomaly detection systems often incorporate dynamic thresholding techniques that adaptively adjust the acceptable ranges based on individual patient characteristics and contextual information. This allows for more personalized anomaly detection and reduces the likelihood of false alarms.

In any case, the model created demonstrates the potential practicality of using Machine Learning in Oncology. Such technologies can offer a valuable tool for healthcare professionals to analyze health indicator data, detect anomalies, and identify patients at risk, ultimately leading to improved patient outcomes and quality of care.

MODEL CODE: <https://replit.com/join/ngyauxhjxf-ayushchalishaja>

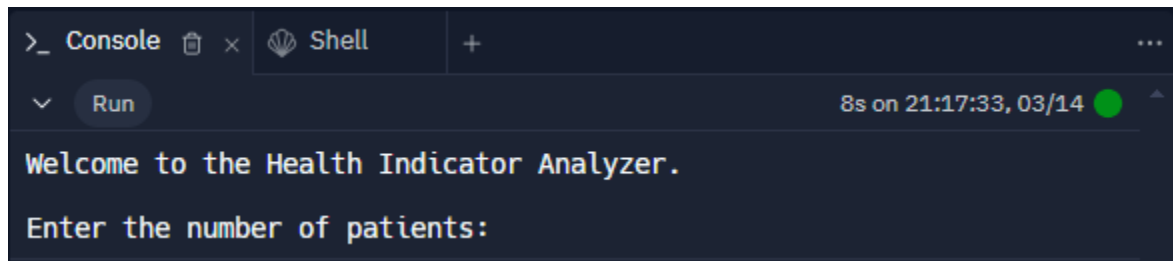
```
main.py x +
main.py > ... Format
1 import numpy as np
2
3 # Define acceptable ranges for health indicators
4 acceptable_ranges = {
5     'heart_rate': (50, 110),
6     'oxygen_saturation': (92, 100),
7     'systolic_bp': (90, 140),
8     'diastolic_bp': (60, 90),
9     'total_cholesterol': (125, 200),
10    'ldl_cholesterol': (0, 130),
11    'hdl_cholesterol': (40, 60),
12    'glucose': (70, 140)
13 }
14
15 # Function to input patient data
16 def input_patient_data(num_patients):
17     """
18     Prompt the user to input data for each patient.
19     """
20     print("\nPlease input data for each patient.")
21     patient_data = []
22     for i in range(num_patients):
23         print(f"\nPatient {i+1}:")
24         try:
25             data = input("Enter Heart Rate (bpm) Oxygen Saturation (%) Systolic Blood Pressure (mmHg) Diastolic
26             Blood Pressure (mmHg) Total Cholesterol (mg/dL) LDL Cholesterol (mg/dL) HDL Cholesterol (mg/dL) Glucose (mg/dL)
27             (separated by spaces): ").split()
28             if len(data) != len(acceptable_ranges):
29                 raise ValueError("Invalid number of values entered. Please provide data for all health
30                 indicators.")
31             data = [float(d) for d in data]
32             except ValueError as e:
33                 print(f"Error: {e}. Please try again.")
34                 return input_patient_data(num_patients)
35             patient_data.append(data)
36     return np.array(patient_data)
```

```
main.py x +
main.py > ...
35 # Detect anomalies based on acceptable ranges
36 def detect_anomalies(tests):
37     """
38     Detect anomalies in patient data based on acceptable ranges.
39     """
40     anomalies = []
41     for patient_idx, patient_tests in enumerate(tests):
42         is_high_risk = False
43         for indicator_idx, (indicator, (min_val, max_val)) in enumerate(acceptable_ranges.items()):
44             if not min_val <= patient_tests[indicator_idx] <= max_val:
45                 anomalies.append((patient_idx, indicator_idx))
46                 is_high_risk = True
47         if is_high_risk:
48             print(f"\nPatient {patient_idx + 1} is flagged as high risk:")
49             for anomaly in anomalies:
50                 if anomaly[0] == patient_idx:
51                     print(f"Indicator: {list(acceptable_ranges.keys())[anomaly[1]]}")
52             print()
53     return anomalies
54
55 # Main function
56 def main():
57     """
58     Main function to execute the program.
59     """
60     print("Welcome to the Health Indicator Analyzer.")
61     try:
62         num_patients = int(input("\nEnter the number of patients: "))
63         if num_patients <= 0:
64             raise ValueError("Number of patients must be a positive integer.")
65     except ValueError as e:
66         print(f"Error: {e}. Please enter a valid number of patients.")
67         return main()
68
69     # Input all patient data
70     patient_data = input_patient_data(num_patients)
71
```

```
main.py x +
main.py > ...
72 # Detect anomalies in patient data
73 anomalies = detect_anomalies(patient_data)
74
75 if len(anomalies) == 0:
76     print("No anomalies detected.")
77 else:
78     print("Anomalies detected.")
79
80 if __name__ == "__main__":
81     main()
82
```

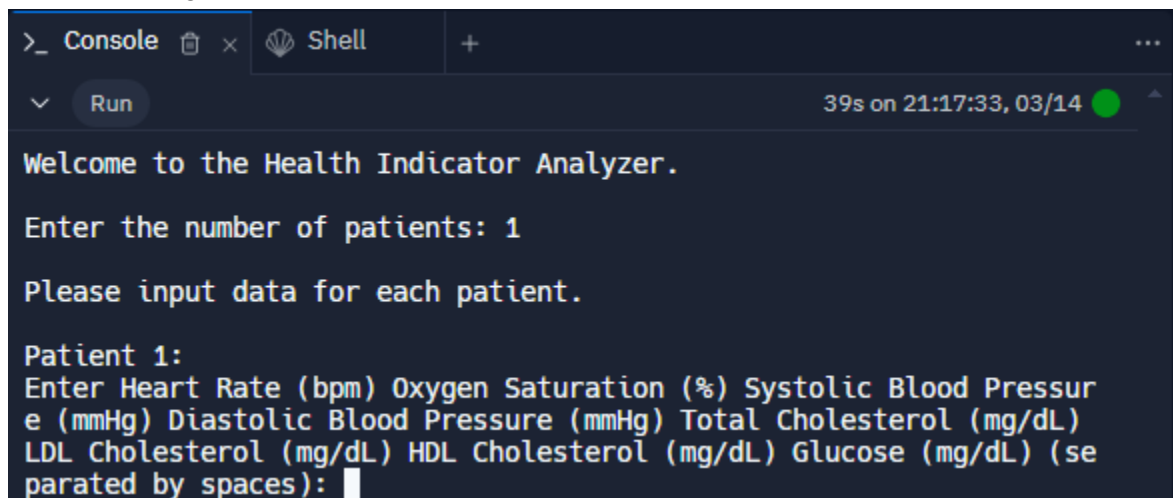
MODEL CONSOLE:

Original Prompt -



```
>_ Console [trash] [x] Shell + ...
  Run 8s on 21:17:33, 03/14
Welcome to the Health Indicator Analyzer.
Enter the number of patients:
```

After Indicating the Number of Patients -



```
>_ Console [trash] [x] Shell + ...
  Run 39s on 21:17:33, 03/14
Welcome to the Health Indicator Analyzer.
Enter the number of patients: 1
Please input data for each patient.

Patient 1:
Enter Heart Rate (bpm) Oxygen Saturation (%) Systolic Blood Pressure (mmHg) Diastolic Blood Pressure (mmHg) Total Cholesterol (mg/dL) LDL Cholesterol (mg/dL) HDL Cholesterol (mg/dL) Glucose (mg/dL) (separated by spaces):
```