heartrate

July 31, 2024

1 Anomaly Detection in Heart Rate Data

Importing the important libraries for the project.

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest
```

Specifies the path to the CSV file containing the heartrate data.Loads the heart rate data from the CSV file into a pandas DataFrame called `heart_rate_data`

[3]:
```
# Load the heart rate data from the CSV file
file_path = 'heartrate.csv'
heart_rate_data = pd.read_csv(file_path)
```

Converts the `datetime` column in the DataFrame to a datetime object using the specified format.

[4]:
```
# Convert 'datetime' column to datetime object
heart_rate_data['datetime'] = pd.to_datetime(heart_rate_data['datetime'],␣
↪format='%d/%m/%y %H:%M')
```

Sets the `datetime` column as the index of the DataFrame, modifying the DataFrame in place.

[5]:
```
# Set the datetime as the index
heart_rate_data.set_index('datetime', inplace=True)
```

Drops any rows in the DataFrame that contain missing values, modifying the DataFrame in place.

[6]:
```
# Check for missing values and handle them (here, we drop rows with missing␣
↪values)
heart_rate_data.dropna(inplace=True)
```

Creates an instance of the Isolation Forest algorithm with a contamination parameter set to 0.01 (indicating that 0.02% of the data is expected to be anomalies) and a fixed random state for reproducibility.

[25]:
```
# Define the model
isolation_forest = IsolationForest(contamination=0.002, random_state=42)
```

Fits the Isolation Forest model to the **heartrate** column of the DataFrame and predicts anomalies, storing the results in a new column called **anomaly**. Anomalies are labeled as -1.

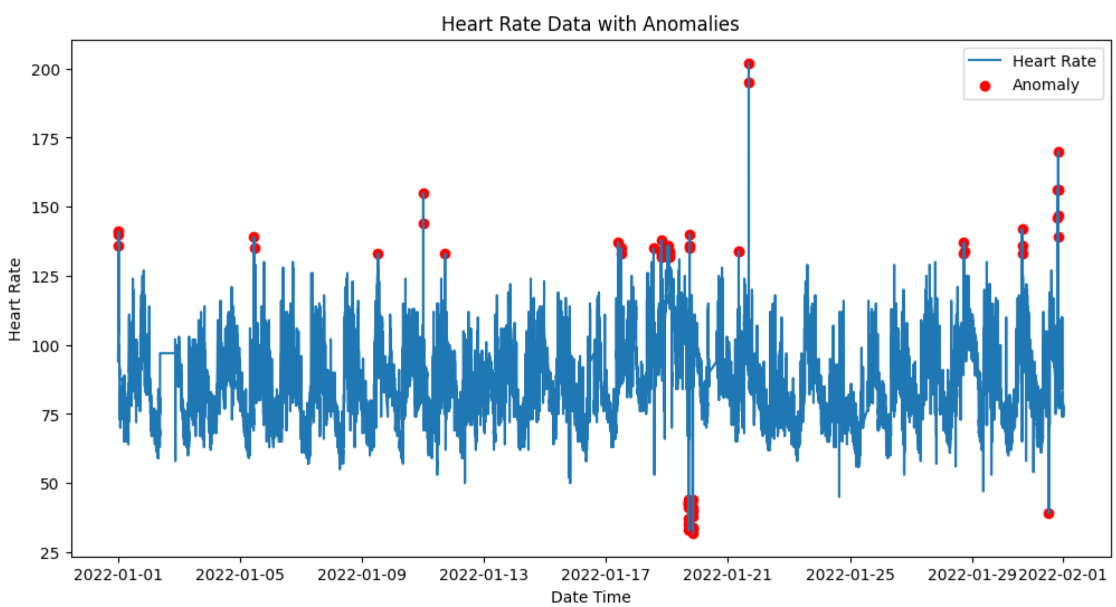
[26]:
```
# Fit the model
heart_rate_data['anomaly'] = isolation_forest.
    ↪fit_predict(heart_rate_data[['heartrate']])
```

Filters the DataFrame to include only the rows where the **anomaly** column is -1, indicating an anomaly.

[27]:
```
# Anomalies are labeled as -1 in the anomaly column
anomalies = heart_rate_data[heart_rate_data['anomaly'] == -1]
```

Plotting the graph

[28]:
```
# Plot the heart rate data with anomalies
plt.figure(figsize=(12, 6))
plt.plot(heart_rate_data.index, heart_rate_data['heartrate'], label='Heart␣
    ↪Rate')
plt.scatter(anomalies.index, anomalies['heartrate'], color='red',␣
    ↪label='Anomaly', marker='o')
plt.xlabel('Date Time')
plt.ylabel('Heart Rate')
plt.title('Heart Rate Data with Anomalies')
plt.legend()
plt.show()
```



[29]:
```
# Display the anomalies
print(anomalies[['heartrate']])
```

```
                     heartrate
datetime
2022-01-01 00:23:00      140.0
2022-01-01 00:24:00      141.0
2022-01-01 00:25:00      136.0
2022-01-05 10:58:00      139.0
2022-01-05 10:59:00      135.0
…                          …
2022-01-31 19:51:00      156.0
2022-01-31 19:52:00      147.0
2022-01-31 19:58:00      170.0
2022-01-31 19:59:00      156.0
2022-01-31 20:00:00      139.0

[69 rows x 1 columns]
```

[30]:
```
# Find the maximum outlier
max_outlier = anomalies['heartrate'].idxmax()
max_outlier_value = anomalies.loc[max_outlier]

# Print the maximum outlier
print(f"Maximum Outlier:\n{max_outlier_value}")
```

```
Maximum Outlier:
heartrate    202.0
anomaly       -1.0
Name: 2022-01-21 16:22:00, dtype: float64
```

1.1 Summary

1. **Overall Trend**: The heart rate data shows fluctuations over time, which is typical for such physiological data.

2. **Detected Anomalies**: The anomalies detected by the Isolation Forest algorithm are marked with red dots. These points indicate heart rate values that significantly deviate from the norm based on the overall distribution of the data.

1.1.1 Explanation of Anomalies:

- **Isolation Forest Algorithm**: This algorithm works by isolating observations. It randomly selects a feature and splits the data points based on randomly chosen split values between the maximum and minimum values of the selected feature. The algorithm is particularly effective for detecting outliers in high-dimensional datasets.
- **Contamination Parameter**: Set to 0.002, meaning we expect about 0.2% of the data to be anomalies. This parameter can be adjusted based on domain knowledge or further statistical analysis. I have used 0.2, 0.1, 0.05, 0.01 but 0.02 suites the best.

1.1.2 Observations from the Chart:

- **Regular Variations**: The majority of heart rate values fall within a consistent range with regular variations.
- **Outliers**: The detected anomalies (red dots) represent moments when the heart rate significantly deviates from the typical range.
- **Maximum Outlier**: Heartrate 202.0
- **Possible Reason**: These could be due to various factors such as physical activity, stress, measurement errors, or health conditions.
- **Consistency**: Most of the heart rate data appears consistent and within a typical physiological range, indicating no widespread issues or irregularities.

1.1.3 Steps to take:

1. **Investigate Anomalies**: Look into the specific times and conditions when these anomalies occurred to understand potential causes (e.g., physical activity, stress, or sensor errors).
2. **Further Analysis**: Perform additional statistical tests or use other anomaly detection techniques to confirm the findings.
3. **Domain Expertise**: Consult with medical professionals to interpret the significance of the anomalies in the context of overall health.