Name : Mohamed Nawas Raza Mohamed

Student Number: 24148501

Link to Code :
https://colab.research.google.com/drive/1jGvqB9C_fpkhBG9OUzXOGv1hUi1a6G6d?usp=sharing

Blog

Problem Statement

Sudden heart rate variations can indicate severe health risks, including strokes, cardiac arrest, and long-term cardiovascular conditions. Accurate prediction of heart rate fluctuations can help in early detection and preventive healthcare interventions.

Objective

The primary objective of this study is to develop an accurate **time series forecasting model** to predict the next **20-minute heart rate readings** using machine learning techniques. The goal is to build a reliable model that can be utilized in real-world applications for patient monitoring.

Dataset Overview

- **5 CSV files** (*PiD...csv*): Additional time series data for exploratory data analysis and model training.
- **1 CSV file** (*PT_Train.csv*): The primary dataset used for training the final predictive model.

Each dataset contains **timestamped heart rate readings** captured using **medical sensors**. Proper preprocessing ensures data consistency, eliminates anomalies, and optimizes performance.

Implementation

Data Processing

The dataset underwent extensive preprocessing to ensure data quality:

- **Feature Selection**: Retained only `Timestamp (GMT)` and `Lifetouch Heart Rate` columns.
- **Handling Missing Values**: Applied **time-based interpolation** to fill gaps in sensor readings.
- **Outlier Detection & Removal**: Values exceeding **220 BPM** were capped to prevent distortions.
- **Time Series Alignment**: Standardized timestamps across datasets to maintain uniform intervals.
- **Log Transformation**: Applied **natural log transformation** to stabilize variance.

- **Differencing**: Ensured stationarity by applying **first-order differencing** before modeling.

Model Selection

Several models were explored, including **ARIMA, SARIMA, and LSTM-based approaches**. The **Auto-ARIMA** algorithm was used for hyperparameter tuning, optimizing the best (p, d, q) values by minimizing the **Akaike Information Criterion (AIC)**. Seasonal components were excluded as there was no periodic pattern in the dataset.

Evaluation & Diagnostics

The model's performance was evaluated using:

- **Augmented Dickey-Fuller (ADF) Test**: Confirmed stationarity after differencing.
- **Autocorrelation & Partial Autocorrelation (ACF/PACF) Plots**: Identified lag dependencies.
- **Residual Analysis**: Ensured residuals were normally distributed and uncorrelated.
- **Performance Metrics**:
  - **Root Mean Squared Error (RMSE)**: Quantified model accuracy.
  - **Mean Absolute Error (MAE)**: Evaluated deviation between actual and predicted values.
  - **Mean Absolute Percentage Error (MAPE)**: Assessed relative forecasting error.

Final Forecasting & Reconstruction

After training the best ARIMA model, the **next 20-minute predictions** were generated. To obtain actual heart rate values:

1. **Undo Differencing**: Cumulative summation of forecasted differences, adding the last known actual value.
2. **Reverse Log Transformation**: Applied **exponential transformation (antilog)** to restore original-scale predictions.
3. **Assign Time Index**: Created a **1-minute interval index** to structure the forecast output.

Submission

- The **final 20-minute predictions** were saved as a **CSV file** for submission.
- The **Python notebook** includes **comprehensive documentation, visualizations, justification of modeling choices, and performance analysis** to ensure clarity and reproducibility.
- Future improvements could involve incorporating **deep learning models (LSTMs, Transformers)** for enhanced accuracy and scalability.

```
Note: Due to the infussificent Please kindly check the code note book for in detailed report
```