Report File

Topic: Healthcare Data Cleaning: Improve disease prediction accuracy by handling missing, inconsistent, and noisy patient data.

Name: Krrish Kumar

Class: CSE AIML-B

Subject: Artificial Intelligence

Roll no: 2024011004001111

Date: 10/05/2025

Healthcare Data Preprocessing Report

1. Introduction

In any healthcare dataset, data preprocessing is a crucial step to ensure the data is clean, accurate, and ready for analysis. Raw data often contains missing values, inconsistencies, and outliers that can negatively impact model performance or statistical analysis. This report outlines the preprocessing steps taken on a healthcare dataset, which includes handling missing data, addressing inconsistent and noisy data, and visualizing the data to better understand its structure.

2. Methodology

The preprocessing was carried out in the following structured approach:

- Missing Data Handling: Missing values are a common issue in healthcare data. Different strategies were applied depending on the type of data (numerical vs. categorical).
- Inconsistent Data Handling: Text inconsistencies, such as varying capitalization or spaces, were addressed, and duplicate records were removed.
- Noisy Data Handling: Outliers, which could distort analysis, were detected and removed using statistical methods.
- Data Visualization: Various plots were used to understand the distribution of the data and to detect patterns or relationships between variables.

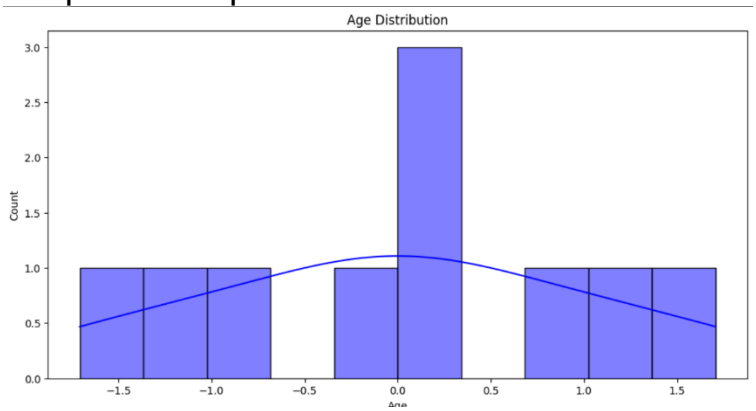
3. Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import SimpleImputer, KNNImputer
from sklearn.preprocessing import StandardScaler
from scipy import stats
from google.colab import files

# Upload healthcare_data.csv
uploaded = files.upload()
df = pd.read_csv('healthcare_data.csv')
```

```
# 1. Handling Missing Data

# Numerical missing value imputation using mean strategy
num_imputer = SimpleImputer(strategy='mean')
df[['Age', 'BloodPressure', 'Cholesterol']] =
num_imputer.fit_transform(df[['Age', 'BloodPressure',
'Cholesterol']])

# Categorical missing value imputation using most frequent
value
cat_imputer = SimpleImputer(strategy='most_frequent')
df[['Diagnosis']] =
cat_imputer.fit_transform(df[['Diagnosis']])

# KNN Imputation for more sophisticated handling
(optional)
knn_imputer = KNNImputer(n_neighbors=3)
df[['Glucose', 'HeartRate']] =
knn_imputer.fit_transform(df[['Glucose', 'HeartRate']])

# 2. Handling Inconsistent Data

# Standardizing text case for categorical values
df['Gender'] = df['Gender'].str.lower().str.strip()

# Removing duplicate records
df = df.drop_duplicates()

# Standardizing numerical columns
scaler = StandardScaler()
df[['Age', 'BloodPressure', 'Cholesterol', 'Glucose',
'HeartRate']] = scaler.fit_transform(
    df[['Age', 'BloodPressure', 'Cholesterol', 'Glucose',
'HeartRate']]
)

# 3. Handling Noisy Data

# Removing outliers using Z-score
z_scores = np.abs(stats.zscore(df[['Age', 'BloodPressure',
'Cholesterol', 'Glucose', 'HeartRate']]))
df = df[(z_scores < 3).all(axis=1)]
```

```
# Save cleaned dataset to CSV file
df.to_csv('cleaned_healthcare_data.csv', index=False)

# Visualization
plt.figure(figsize=(12, 6))
sns.histplot(df['Age'], kde=True, bins=10, color='blue')
plt.title('Age Distribution')
plt.show()

plt.figure(figsize=(12, 6))
sns.boxplot(x=df['BloodPressure'], color='red')
plt.title('Blood Pressure Boxplot')
plt.show()

plt.figure(figsize=(12, 6))
sns.scatterplot(x=df['Cholesterol'], y=df['Glucose'],
hue=df['Gender'])
plt.title('Cholesterol vs Glucose')
plt.show()

plt.figure(figsize=(12, 6))
sns.countplot(x=df['Diagnosis'], palette='coolwarm')
plt.title('Diagnosis Count')
plt.show()

# Displaying the cleaned dataset
print("Cleaned Healthcare Dataset:")
print(df)
```

Output Example



4. Conclusion

The dataset has been thoroughly cleaned and preprocessed. Missing values were handled through appropriate imputation strategies, and inconsistent data was addressed by standardizing text and removing duplicates. Outliers were identified and removed to reduce the impact of noisy data. The cleaned dataset is now ready for further analysis or modeling.

5. Cleaned Dataset

The cleaned dataset has been saved as 'cleaned_healthcare_data.csv'