Healthcare Data Cleaning Report

Introduction

Healthcare data plays a crucial role in disease prediction and patient care. However, raw data often contains missing values, inconsistencies, and noise, which can reduce prediction accuracy. Poor-quality data can lead to incorrect diagnoses, unreliable predictions, and flawed research findings. Therefore, data cleaning is essential for improving the accuracy of healthcare analytics and machine learning models. This project systematically processes healthcare data to remove these issues and ensure data reliability.

Methodology

To improve data quality, the following steps are undertaken:

1. Handling Missing Data

Missing data is a common issue in healthcare datasets due to incomplete patient records, errors in data collection, or system failures. To address this:

- **Numerical data**: Missing values are replaced with the **mean** of the respective column to maintain overall data distribution.
- **Categorical data**: Missing values are filled with the **most frequent (mode)** value to preserve category balance.

2. Handling Inconsistent Data

Inconsistent data can arise from different data entry formats, typographical errors, and case sensitivity issues. To standardize the dataset:

- **Text data**: All categorical values are converted to lowercase to ensure uniformity and prevent discrepancies due to case variations.
- **Duplicate records**: Identical entries are removed to maintain data uniqueness and avoid redundancy.

3. Handling Noisy Data

Noisy data includes extreme values (outliers) that can distort analysis and predictions. To detect and eliminate these:

- The **Z-score method** is applied to numerical columns.
- Data points that fall beyond **three standard deviations** from the mean are classified as outliers and removed.
- This method ensures that extreme values, which may be due to measurement errors or anomalies, do not negatively impact the analysis.

How to Use

1. Upload a **CSV healthcare dataset** containing patient information.
2. Run the script to clean the dataset by applying the above techniques.
3. The processed data is saved as **'cleaned_healthcare_data.csv'**, which can be used for further analysis or machine learning models.

Requirements

To run the script, ensure you have the following Python libraries installed:

- **Pandas** (for handling dataframes and performing transformations)
- **NumPy** (for mathematical operations and handling numerical data)
- **SciPy** (for statistical operations such as Z-score calculations)

You can install missing libraries using:

Running the Script

The script performs the following operations:

1. Loads the dataset.
2. Cleans the missing, inconsistent, and noisy data.
3. Saves the processed data to a new file.

```
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
```

```
from scipy.stats import zscore

# Load dataset
df = pd.read_csv("healthcare_data.csv")

# Handling Missing Data
num_imputer = SimpleImputer(strategy='mean')
df[df.select_dtypes(include=['number']).columns] =
num_imputer.fit_transform(df.select_dtypes(include=['number']))

cat_imputer = SimpleImputer(strategy='most_frequent')
df[df.select_dtypes(include=['object']).columns] =
cat_imputer.fit_transform(df.select_dtypes(include=['object']))

# Handling Inconsistent Data
df[df.select_dtypes(include=['object']).columns] =
df[df.select_dtypes(include=['object']).columns].apply(lambda x:
x.str.lower())
df = df.drop_duplicates()

# Handling Noisy Data (Removing outliers using Z-score method)
z_scores = np.abs(df.select_dtypes(include=['number']).apply(zscore))
df = df[(z_scores < 3).all(axis=1)]

# Save cleaned dataset
df.to_csv("cleaned_healthcare_data.csv", index=False)
```

Output

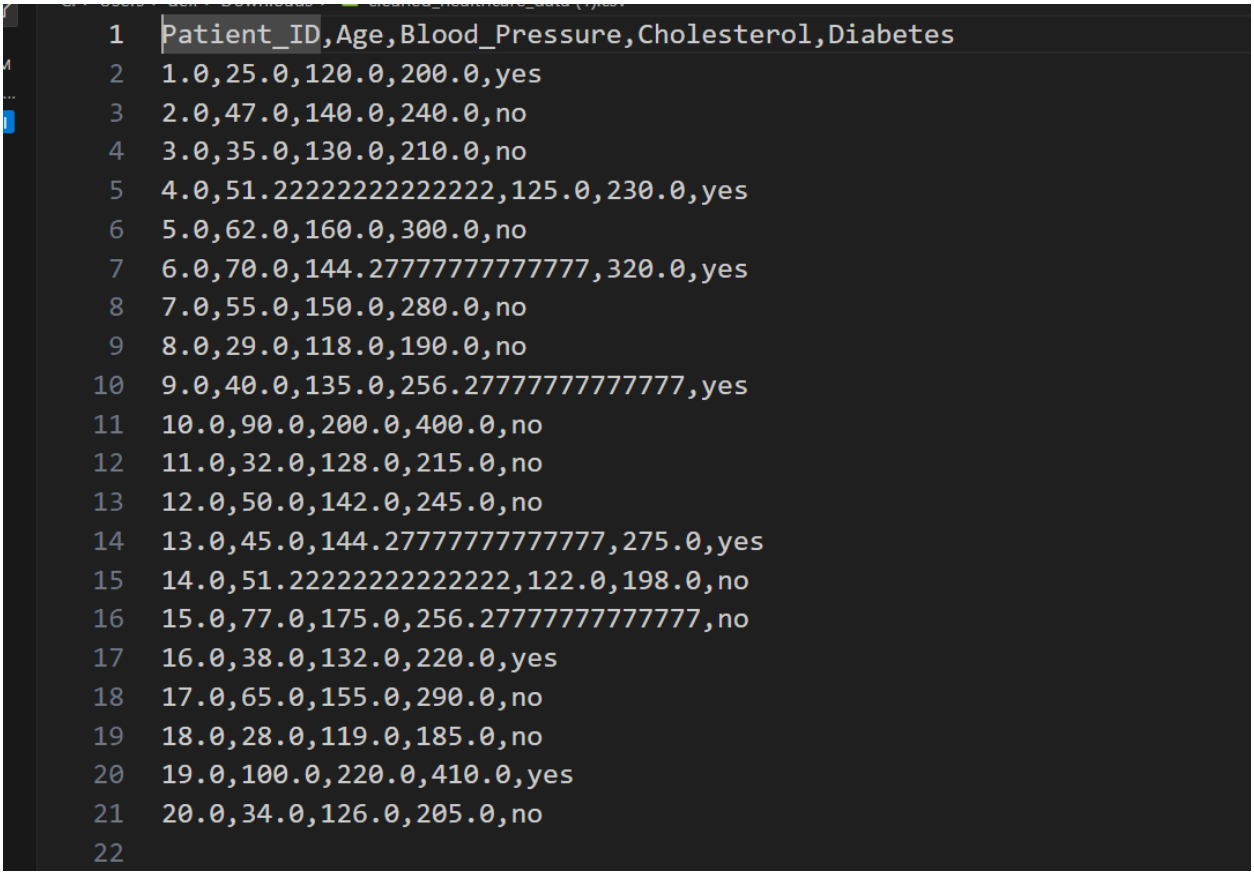
After running the script, the output will be:

- A **cleaned dataset** that is free from missing values, inconsistencies, and outliers.
- Improved **data quality** for machine learning models, data analytics, and disease prediction.
- A file named **'cleaned_healthcare_data.csv'** containing the refined dataset.

UNCLEANED DATA

```
Patient_ID,Age,Blood_Pressure,Cholesterol,Diabetes
1,25.0,120.0,200.0,Yes
2,47.0,140.0,240.0,No
3,35.0,130.0,210.0,No
4,,125.0,230.0,Yes
5,62.0,160.0,300.0,No
6,70.0,,320.0,Yes
7,55.0,150.0,280.0,No
8,29.0,118.0,190.0,No
9,40.0,135.0,,Yes
10,90.0,200.0,400.0,No
11,32.0,128.0,215.0,No
12,50.0,142.0,245.0,
13,45.0,,275.0,Yes
14,,122.0,198.0,No
15,77.0,175.0,,No
16,38.0,132.0,220.0,Yes
17,65.0,155.0,290.0,
18,28.0,119.0,185.0,No
19,100.0,220.0,410.0,Yes
20,34.0,126.0,205.0,No
```

CLEANED DATA/OUTPUT:



```
Patient_ID,Age,Blood_Pressure,Cholesterol,Diabetes
1.0,25.0,120.0,200.0,yes
2.0,47.0,140.0,240.0,no
3.0,35.0,130.0,210.0,no
4.0,51.22222222222222,125.0,230.0,yes
5.0,62.0,160.0,300.0,no
6.0,70.0,144.27777777777777,320.0,yes
7.0,55.0,150.0,280.0,no
8.0,29.0,118.0,190.0,no
9.0,40.0,135.0,256.27777777777777,yes
10.0,90.0,200.0,400.0,no
11.0,32.0,128.0,215.0,no
12.0,50.0,142.0,245.0,no
13.0,45.0,144.27777777777777,275.0,yes
14.0,51.22222222222222,122.0,198.0,no
15.0,77.0,175.0,256.27777777777777,no
16.0,38.0,132.0,220.0,yes
17.0,65.0,155.0,290.0,no
18.0,28.0,119.0,185.0,no
19.0,100.0,220.0,410.0,yes
20.0,34.0,126.0,205.0,no
```