***TITTLE***

***Healthcare Data Cleaning Improving Disease Prediction Accuracy by Handling Missing, Inconsistent, and Noisy Patient Data***

***Name:-***Utkrisht Verma;

Roll No:-58

Abstract***:***

***Healthcare data is often plagued by missing values, inconsistencies, and noise, which can adversely affect disease prediction accuracy. This report outlines a systematic approach to preprocessing such data to enhance its quality and reliability for predictive modeling. Techniques such as handling missing values, standardizing inconsistent entries, removing outliers, and scaling numerical data are applied to a sample dataset. improvements in data quality, paving the way***

***for more accurate predictions and better healthcare outcomes.***

***Introduction:***

***Data is the backbone of modern healthcare research and predictive analytics. However, raw healthcare data collected from diverse sources often contains irregularities, such as:***

* ***Missing Values: Partial or incomplete records.***
* ***Inconsistent Data: Variability in how information is recorded (e.g., "M" vs "Male").***
* ***Noisy Data: Outliers or erroneous entries.***

***Such issues reduce the effectiveness of machine learning models. Cleaning and preprocessing data is essential for ensuring accurate disease prediction. This report discusses techniques to clean data while working with a sample dataset.***

***Problem Statement:***

***The goal is to clean healthcare data to improve disease prediction accuracy by:***

1. ***Handling missing values in numerical and categorical data.***
2. ***Correcting inconsistent data entries.***
3. ***Detecting and removing noisy data (e.g., outliers).***
4. ***Standardizing numerical data for predictive modeling.***

***Methodology:***

***The data cleaning process followed these steps:***

1. ***Handling Missing Values:***
   * ***Numerical columns were imputed using the mean.***
   * ***Categorical columns were imputed using the most frequent value.***
2. ***Standardizing Inconsistent Data:***
   * ***Categorical variables (e.g., Gender) were standardized (e.g., "M" and "Male" unified to "Male").***
3. ***Noise Removal:***
   * ***The Interquartile Range (IQR) method was used to detect and remove outliers.***
4. ***Scaling Numerical Data:***
   * ***StandardScaler was applied to normalize numerical features for consistency in machine learning models.***

***A Python implementation of this process was created, with sample data provided to demonstrate the steps.***

***Results:***

***The data cleaning process successfully transformed raw, inconsistent, and incomplete healthcare data into a cleaner dataset ready for analysis. Key improvements include:***

* ***Filling in missing values to avoid data loss.***
* ***Correcting and unifying inconsistent entries for categorical variables.***
* ***Removing outliers to enhance data reliability.***
* ***Standardizing numerical features, enabling better performance of machine learning algorithms.***

***The cleaned data was saved as a CSV file named cleaned\_healthcare\_data.csv for future use.***

***Conclusion:***

***Healthcare data cleaning is a critical step toward improving the accuracy of disease prediction models. By addressing missing values, inconsistencies, and noise, the quality and usability of the data are significantly enhanced. The systematic approach presented in this report can be applied to real-world healthcare datasets, leading to improved predictive insights and better healthcare outcomes.***

***Recommendations:***

1. ***Use domain knowledge to determine acceptable ranges for outliers.***
2. ***Regularly validate cleaning techniques on real-world data.***
3. ***Incorporate visualization techniques to further identify patterns or anomalies in the data.***

***Future Work:***

***The methods described here can be further enhanced by incorporating advanced techniques, such as:***

* ***Machine learning-based imputation methods for missing data.***
* ***Advanced outlier detection techniques like DBSCAN or Isolation Forest.***
* ***Real-time data cleaning pipelines for streaming healthcare data.***

Code:-

import pandas as pd

import numpy as np

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import StandardScaler

# Sample healthcare data

data = {

    "Patient\_ID": [101, 102, 103, 104, 105],

    "Age": [25, 40, np.nan, 35, 29],

    "Gender": ["M", "F", "Male", "F", np.nan],

    "Blood\_Pressure": [120, 140, 150, np.nan, 130],

    "Cholesterol\_Level": [200, 240, 300, 150, np.nan],

    "Disease\_Present": [0, 1, 1, 0, 1],  # 0: No, 1: Yes

}

# Create DataFrame

df = pd.DataFrame(data)

# Display original dataset

print("Original Dataset:\n", df, "\n")

# Handle missing values

# Replace missing numerical values with the mean

imputer = SimpleImputer(strategy="mean")

numerical\_columns = df.select\_dtypes(include=[np.number]).columns

df[numerical\_columns] = imputer.fit\_transform(df[numerical\_columns])

# Replace missing categorical values with the most frequent value

imputer\_cat = SimpleImputer(strategy="most\_frequent")

categorical\_columns = df.select\_dtypes(include=["object"]).columns

df[categorical\_columns] = imputer\_cat.fit\_transform(df[categorical\_columns])

# Standardize inconsistent categorical data

df["Gender"] = df["Gender"].replace({"M": "Male", "F": "Female"})

# Remove noisy data

# Example: Removing extreme outliers in numerical columns using IQR

for col in numerical\_columns:

    q1 = df[col].quantile(0.25)

    q3 = df[col].quantile(0.75)

    iqr = q3 - q1

    lower\_bound = q1 - 1.5 \* iqr

    upper\_bound = q3 + 1.5 \* iqr

    df = df[(df[col] >= lower\_bound) & (df[col] <= upper\_bound)]

# Standardize numerical columns for scaling

scaler = StandardScaler()

df[numerical\_columns] = scaler.fit\_transform(df[numerical\_columns])

# Save cleaned dataset as CSV file

csv\_filename = "cleaned\_healthcare\_data.csv"

df.to\_csv(csv\_filename, index=False)

print(f"Cleaned Dataset:\n{df}\n")

print(f"Cleaned data saved to '{csv\_filename}'.")

***Screenshots:-***

***A screenshot of a computer

Description automatically generated***