Data Cleansing and Transformation Report

Team Member's Details

Group Name: Data Explorer

Name: Mohammad Tohin Bapari

Email: tohin@gmx.de

Country: Germany

University: Bergische Universität Wuppertal

Specialization: Data Science

Problem Description

The task involves data cleansing and transformation of a healthcare dataset. The objective is to handle missing values using various imputation techniques and to identify and handle outliers.

Github Repository Link

https://github.com/iamtohin/Data-Analytst-Internship-at-Data-Glacier/tree/main/Week%209

Explained Summary with Code Snapshots

The task focuses on data cleansing and transformation. Initially, the dataset is loaded and its structure is explored. Missing values are identified and handled using mean and mode imputation techniques for numerical and categorical columns respectively. Outliers are identified using the IQR and Z-score methods and are handled appropriately.

Key steps include:

1. Importing necessary libraries
2. Loading the dataset

3. Identifying missing values and handling them using imputation

4. Identifying and handling outliers using IQR and Z-score methods

```
# Importing Libraries

import pandas as pd

import numpy as np

from sklearn.impute import SimpleImputer

from scipy import stats


# Load the dataset

file_path = 'Healthcare_dataset.xlsx'

data = pd.read_excel(file_path,'Dataset')


# Display the first few rows of the dataset to understand its structure

data.head()


# Dataset Information

data.info(5)


# Identify missing values

missing_values = data.isnull().sum()


# Handle missing values using mean/median/mode imputation

imputer_mean = SimpleImputer(strategy='mean')

imputer_mode = SimpleImputer(strategy='most_frequent')


# Numerical columns
```

```
num_cols = data.select_dtypes(include=['float64', 'int64']).columns

data[num_cols] = imputer_mean.fit_transform(data[num_cols])


# Categorical columns

cat_cols = data.select_dtypes(include=['object']).columns

data[cat_cols] = imputer_mode.fit_transform(data[cat_cols])


# Verify if all missing values are handled

missing_values_after_imputation = data.isnull().sum()

print(missing_values_after_imputation)


# Handling outliers using IQR method

Q1 = data[num_cols].quantile(0.25)

Q3 = data[num_cols].quantile(0.75)

IQR = Q3 - Q1


# Removing outliers

data = data[~((data[num_cols] < (Q1 - 1.5 * IQR)) |(data[num_cols] > (Q3 + 1.5 * IQR))).any(axis=1)]


# Handling outliers using Z-score method

z_scores = np.abs(stats.zscore(data[num_cols]))

data = data[(z_scores < 3).all(axis=1)]


# Verify the data after handling outliers

print(data.shape)
```