Healthcare Data Cleaning

Prepared by: Aditya Shukla

Date: 11-03-2025

Dataset:
healthcare_data(1).csv

Introduction

This report analyses employee salary data to uncover trends, relationships, and insights. The dataset includes employee information such as EmployeeID, Age, Department, Experience, and Salary. The analysis aims to:

1. Understand the distribution of salaries.
2. Compare salaries across departments.
3. Examine the relationship between salary, age, and experience. 4. Identify trends and provide actionable insights.

Methodology

The analysis was conducted using Python and its libraries:

- Pandas: For data manipulation and analysis.
- NumPy: For numerical computations.
- Matplotlib and Seaborn: For data visualization. The steps followed include:

1. Importing necessary libraries.
2. Loading the dataset.
3. Exploring the dataset (data types, missing values, etc.).
4. Visualizing salary distribution and relationships between variables.
5. Summarizing findings

CODE

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('/content/healthcare_data
(1).csv')
print(df.head()) # Display the first few rows
of the DataFrame
```

```
# Data Cleaning
# Remove duplicates
df = df.drop_duplicates()


# Handle missing values
df = df.dropna()

## Remove outliers using IQR
for col in ['Age', 'BloodPressure',
'SugarLevel', 'Weight']:
    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)]
```

```
# Age distribution
plt.figure(figsize=(8, 6))
sns.histplot(df['Age'], kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()

# Blood Pressure distribution
plt.figure(figsize=(8, 6))
sns.histplot(df['BloodPressure'], kde=True)
plt.title('Blood Pressure Distribution')
plt.xlabel('Blood Pressure')
plt.ylabel('Frequency')
plt.show()
```

```
# Sugar Level distribution
plt.figure(figsize=(8, 6))
sns.histplot(df['SugarLevel'], kde=True)
plt.title('Sugar Level Distribution')
plt.xlabel('Sugar Level')
plt.ylabel('Frequency')
plt.show()
```

```
# Weight distribution
plt.figure(figsize=(8, 6))
sns.histplot(df['Weight'], kde=True)
plt.title('Weight Distribution')
plt.xlabel('Weight')
plt.ylabel('Frequency')
plt.show()
```

```
# Scatter plot of Blood Pressure vs Sugar Level
plt.figure(figsize=(8, 6))
sns.scatterplot(x='BloodPressure',
y='SugarLevel', data=df)
plt.title('Blood Pressure vs. Sugar Level')
plt.xlabel('Blood Pressure')
plt.ylabel('Sugar Level')
plt.show()


# Select features for clustering (Blood
Pressure and Sugar Level)
X = df[['BloodPressure', 'SugarLevel']]

# Apply KMeans clustering with 3 clusters
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X)

# Visualize the clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x='BloodPressure',
y='SugarLevel', hue='Cluster', data=df,
palette='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0],
kmeans.cluster_centers_[:, 1], s=300, c='red',
label='Centroids')
plt.title('Patient Segmentation with KMeans
Clustering')
plt.xlabel('Blood Pressure')
plt.ylabel('Sugar Level')
plt.legend()
plt.show()
```

```
# Another visualization with customized colors
plt.figure(figsize=(10, 6))
sns.scatterplot(x='BloodPressure',
y='SugarLevel', hue='Cluster', data=df,
palette=['green', 'blue', 'orange'], s=100)
plt.scatter(kmeans.cluster_centers_[:, 0],
kmeans.cluster_centers_[:, 1], s=300, c='red',
label='Centroids')
plt.title('Patient Segmentation with KMeans
Clustering (Colored Groups)')
plt.xlabel('Blood Pressure')
plt.ylabel('Sugar Level')
plt.legend()
plt.show()
```