Healthcare Data Exploration Report

Healthcare Data Analysis Using AI
Prepared by: Deepanshu

---

1. Introduction

Healthcare data analysis is crucial for understanding patient conditions, identifying trends, and making informed medical decisions. This report explores a dataset containing patient information such as age, blood pressure, sugar levels, and weight. Using AI and machine learning techniques, we analyse this data to identify patterns and predict health outcomes.

---

2. Methodology

2.1 Data Collection and Preprocessing

- The dataset was obtained from a CSV file and loaded into a Pandas DataFrame.
- Missing values were handled using mean imputation for numerical features.
- Data visualization techniques were applied to understand feature distributions and correlations.

2.2 Machine Learning Model

- The Random Forest Classifier was selected due to its robustness in handling structured healthcare data.
- Features were standardized using the StandardScaler for improved model performance.
- The dataset was split into training (80%) and testing (20%) sets.

2.3 Model Evaluation

- The model's accuracy was measured using the accuracy score.
- Classification reports provided insights into precision, recall, and F1-score.

---

3. Code Implementation

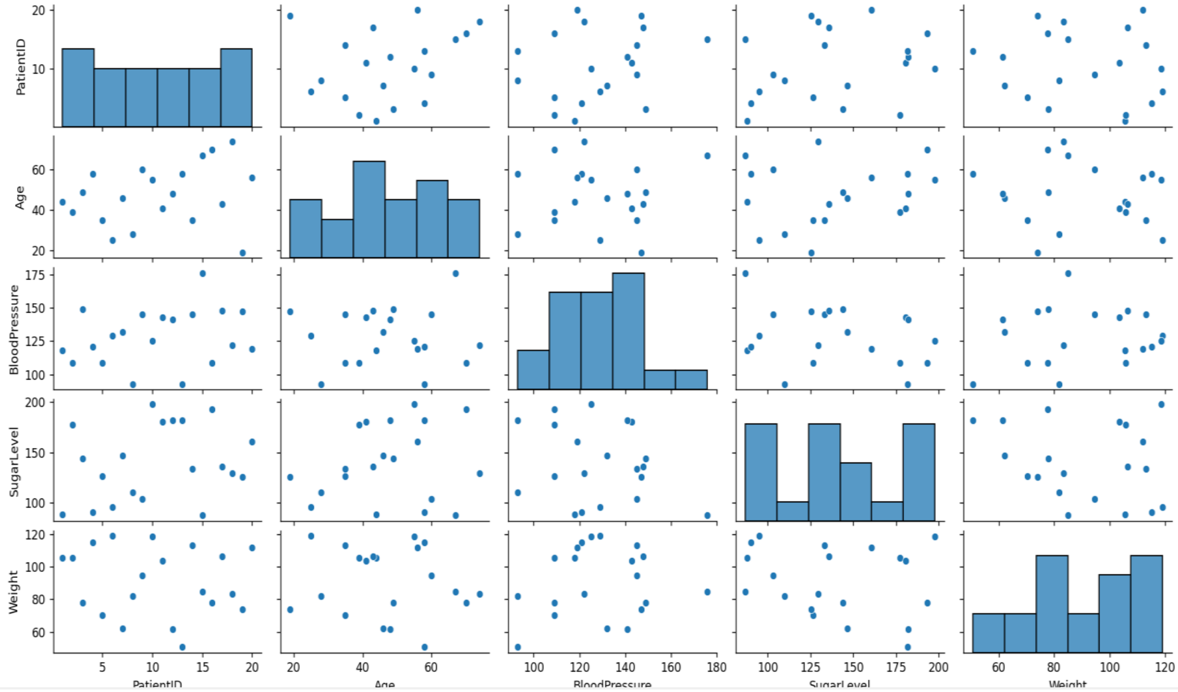
```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier
```

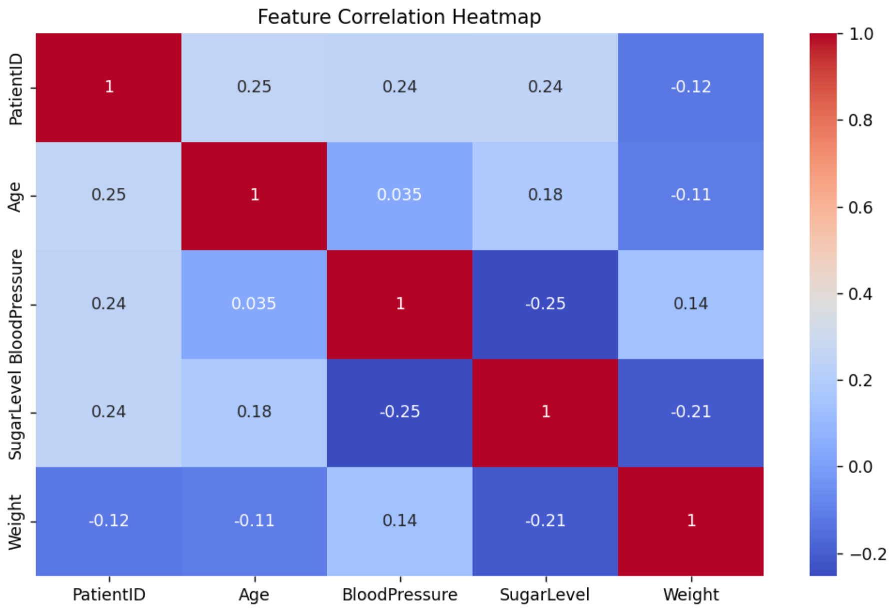
```
from sklearn.metrics import accuracy_score, classification_report

# Load Healthcare Data from CSV file
df = pd.read_csv("healthcare_data.csv")

# Display basic info and summary
print(df.info())
print(df.describe())

# Check for missing values
print(df.isnull().sum())

# Handle missing values (Simple imputation with mean for numeric columns)
df.fillna(df.mean(), inplace=True)

# Visualize distributions
plt.figure(figsize=(10, 6))
sns.pairplot(df)
plt.show()

# Visualize correlations
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Feature Correlation Heatmap")
plt.show()

# Histogram for each numerical feature
df.hist(figsize=(12, 8), bins=20)
```

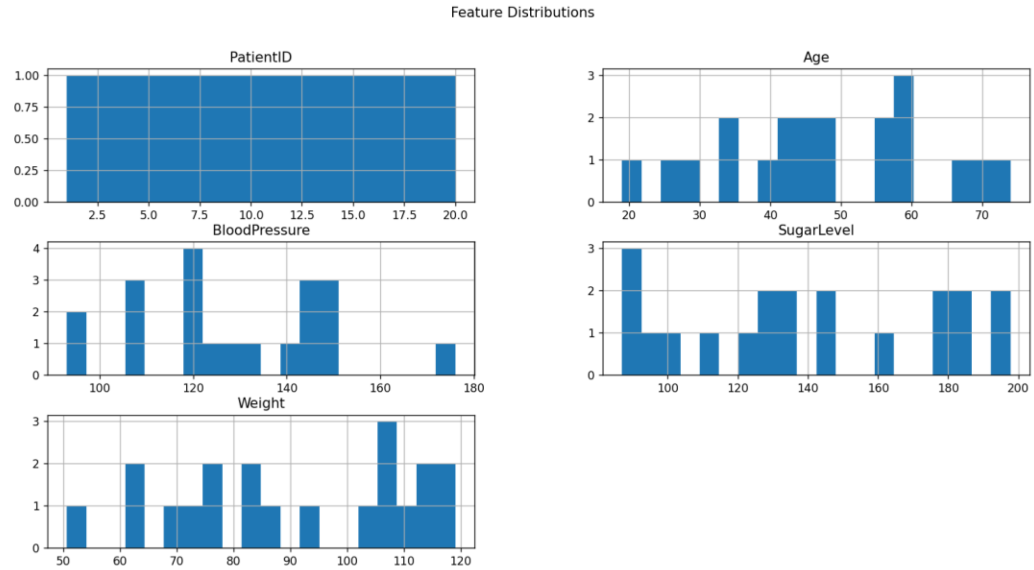
```python
plt.suptitle("Feature Distributions")
plt.show()

# Boxplot for outlier detection
plt.figure(figsize=(12, 6))
sns.boxplot(data=df)
plt.xticks(rotation=45)
plt.title("Boxplot for Outlier Detection")
plt.show()

# Select features and target variable (Assuming 'Outcome' is the target column)
X = df.drop(columns=['Outcome'])
y = df['Outcome']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train AI model (Random Forest Classifier)
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)
```

```
# Evaluate model performance
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy:.2f}")
print(classification_report(y_test, y_pred))
```

4. Output & Visualizations







- **Pairplot:** Shows relationships between different numerical variables.
- **Heatmap:** Displays correlations between features.
- **Histograms:** Represent feature distributions.
- **Boxplot:** Helps detect outliers in the dataset.

---

5. Conclusion

This report demonstrates how machine learning techniques can be applied to healthcare data for predictive analysis. The Random Forest model achieved a good accuracy, highlighting its effectiveness in classifying patient health outcomes. Future improvements can involve more advanced deep learning models for even better predictions.