

SOURCE CODE

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

# Step 1: Simulated Patient Data
data = {
    'age': [25, 45, 52, 36, 23, 57, 62, 41, 29, 33],
    'bmi': [22.0, 28.5, 31.0, 26.4, 21.5, 35.2, 33.1, 29.4, 24.8, 27.3],
    'blood_pressure': [120, 140, 150, 130, 110, 160, 155, 135, 115, 125],
    'glucose': [85, 145, 160, 120, 90, 170, 165, 140, 100, 110],
    'disease': [0, 1, 1, 0, 0, 1, 1, 1, 0, 0] # 1 = disease present, 0 = no disease
}

df = pd.DataFrame(data)

# Step 2: Preprocessing
X = df.drop('disease', axis=1)
y = df['disease']
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42)
```

Step 3: Model Training

model = LogisticRegression()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

Step 4: Evaluation

print("Accuracy:", accuracy_score(y_test, y_pred))

print("Classification Report:\n", classification_report(y_test, y_pred))

Step 5: Prediction Function

def predict_disease(age, bmi, bp, glucose):

input_data = scaler.transform([[age, bmi, bp, glucose]])

prediction = model.predict(input_data)

return "Disease Detected" if prediction[0] == 1 else "No Disease Detected"

print(predict_disease(50, 30.0, 145, 155))

Step 6: Visualizations

Bar Graph - Average metrics by disease class

avg_metrics = df.groupby('disease')[['age', 'bmi', 'blood_pressure', 'glucose']].mean()

avg_metrics.plot(kind='bar')

plt.title("Average Health Metrics by Disease Status")

plt.ylabel("Average Value")

plt.xticks([0, 1], ['No Disease', 'Disease'], rotation=0)

plt.tight_layout()

```
plt.show()
```

```
# Pie Chart - Disease distribution
```

```
df['disease'].value_counts().plot(kind='pie', autopct='%1.1f%%', labels=['No  
Disease', 'Disease'], startangle=90)
```

```
plt.title("Disease Distribution")
```

```
plt.ylabel("")
```

```
plt.show()
```

```
# Histogram - Glucose levels
```

```
plt.hist(df['glucose'], bins=5, color='skyblue', edgecolor='black')
```

```
plt.title("Glucose Level Distribution")
```

```
plt.xlabel("Glucose")
```

```
plt.ylabel("Frequency")
```

```
plt.grid(True)
```

```
plt.show()
```

```
# Scatter Plot - Age vs Glucose
```

```
sns.scatterplot(x='age', y='glucose', hue='disease', palette={0: 'green', 1: 'red'},  
data=df)
```

```
plt.title("Age vs Glucose Scatter Plot")
```

```
plt.xlabel("Age")
```

```
plt.ylabel("Glucose")
```

```
plt.legend(title='Disease')
```

```
plt.grid(True)
```

```
plt.show()
```