Name Kaushal S Galav D15C 11 Aids lab Exp 5

```
# Step 1: Install necessary libraries (if not already installed)
!pip install pandas numpy scikit-learn

# Step 2: Import libraries
import pandas as pds np
import matplotlib.pyplot as plt
import sea
import numpy aborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

# Step 3: Load the dataset (Upload "Global Health Statistics.csv" to Colab
first)
from google.colab import files
uploaded = files.upload()

# Read the dataset
df = pd.read_csv("Global Health Statistics.csv")

# Step 4: Data Preprocessing
# Convert "Availability of Vaccines/Treatment" (Yes/No) into binary (1/0)
df["Availability of Vaccines/Treatment"] = df["Availability of
Vaccines/Treatment"].map({"Yes": 1, "No": 0})

# Selecting features for logistic regression (excluding non-numeric
columns)
selected_features = [
    "Prevalence Rate (%)", "Incidence Rate (%)", "Mortality Rate (%)",
    "Healthcare Access (%)", "Doctors per 1000", "Hospital Beds per 1000",
    "Average Treatment Cost (USD)", "Recovery Rate (%)",
    "Per Capita Income (USD)", "Education Index", "Urbanization Rate (%)"
]

X = df[selected_features]
y = df["Availability of Vaccines/Treatment"]  # Target variable
```

Name Kaushal S Galav D15C 11 Aids lab Exp 5

```
# Handling missing values (if any)
X = X.fillna(X.mean())

# Step 5: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Step 6: Feature Scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Step 7: Train Logistic Regression Model
model = LogisticRegression()
model.fit(X_train, y_train)

# Step 8: Make Predictions
y_pred = model.predict(X_test)

# Step 9: Evaluate Model Performance
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

# Print results
print(f"Model Accuracy: {accuracy:.2f}")
print("\nConfusion Matrix:")
print(conf_matrix)
print("\nClassification Report:")
print(classification_rep)

# Step 10: Visualizing Confusion Matrix
plt.figure(figsize=(5, 4))
sns.heatmap(conf_matrix, annot=True, cmap="Blues", fmt="d",
xticklabels=["No", "Yes"], yticklabels=["No", "Yes"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

Name Kaushal S Galav D15C 11 Aids lab Exp 5

```
Confusion Matrix:
[[46847 52889]
 [47392 52872]]

Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.47      0.48     99736
           1       0.50      0.53      0.51    100264

    accuracy                           0.50    200000
   macro avg       0.50      0.50      0.50    200000
weighted avg       0.50      0.50      0.50    200000
```