

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Step 1: Load the data
df = pd.read_csv("/content/Cleaned-Data.csv")

# Step 3: Combine severity columns into one label
severity_columns = ['Severity_Mild', 'Severity_Moderate', 'Severity_None', 'Severity_Severe']
df['Severity'] = np.argmax(df[severity_columns].values, axis=1)

# Optional: map numeric values to severity names
severity_map = {0: 'Mild', 1: 'Moderate', 2: 'None', 3: 'Severe'}
df['Severity_Label'] = df['Severity'].map(severity_map)

# Step 4: Drop unused columns
df = df.drop(columns=severity_columns + ['Country'])

# Step 5: Split the data
X = df.drop(columns=['Severity', 'Severity_Label'])
y = df['Severity']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 6: Train Naive Bayes model
model = GaussianNB()
model.fit(X_train, y_train)

# Step 7: Make predictions
y_pred = model.predict(X_test)

# # Step 8: Evaluate the model
print("\nModel Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=['Mild', 'Moderate', 'None', 'Sev

# Step 9: Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Mild', 'Moderate', 'None', 'Severe'], yticklabels=['Mild', 'Moderate', 'None', 'Severe'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

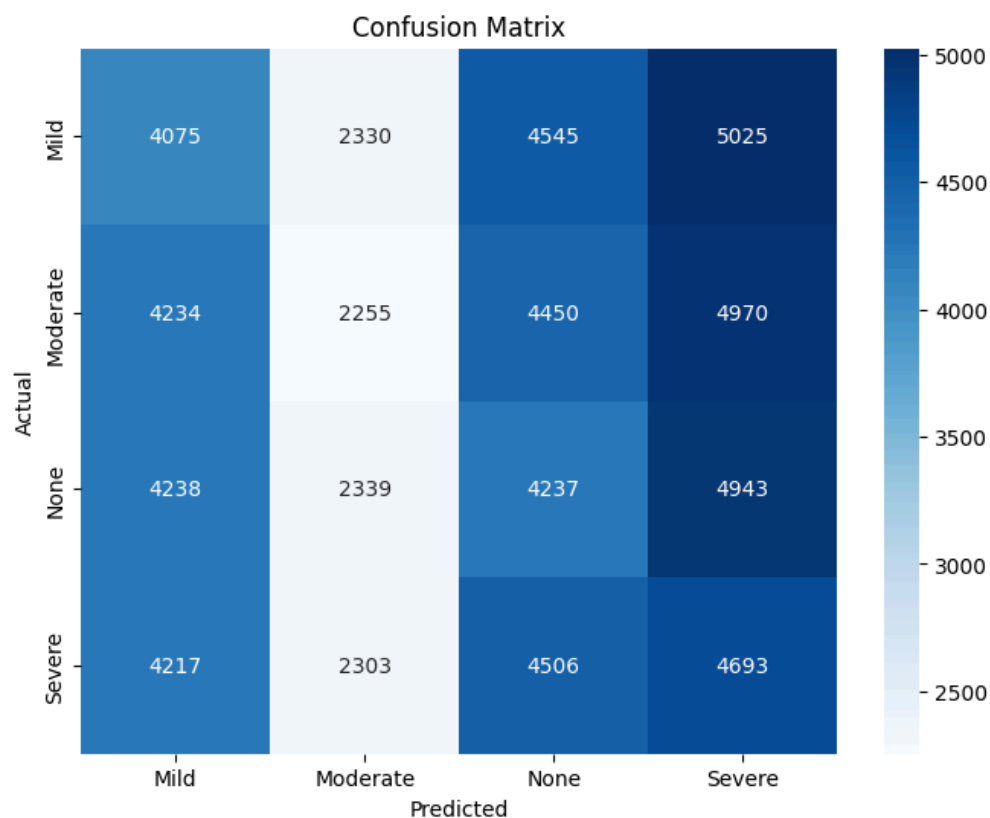
```



Model Accuracy: 0.24084595959595959

Classification Report:

	precision	recall	f1-score	support
Mild	0.24	0.26	0.25	15975
Moderate	0.24	0.14	0.18	15909
None	0.24	0.27	0.25	15757
Severe	0.24	0.30	0.27	15719
accuracy			0.24	63360
macro avg	0.24	0.24	0.24	63360
weighted avg	0.24	0.24	0.24	63360



# Two example patients

# Example 1: No symptoms (expected prediction: None)

```
patient_no_disease = {
    'Fever': 0,
    'Tiredness': 0,
    'Dry-Cough': 0,
    'Difficulty-in-Breathing': 0,
    'Sore-Throat': 0,
    'None_Sympton': 1,
    'Pains': 0,
    'Nasal-Congestion': 0,
    'Runny-Nose': 0,
    'Diarrhea': 0,
    'None_Experiencing': 1,
    'Age_0-9': 0,
    'Age_10-19': 0,
    'Age_20-24': 0,
    'Age_25-59': 1,
    'Age_60+': 0,
    'Gender_Female': 1,
    'Gender_Male': 0,
    'Gender_Transgender': 0,
    'Contact_Dont-Know': 0,
    'Contact_No': 1,
    'Contact_Yes': 0
}
```

```
# Example 2: Many symptoms (expected prediction: Mild/Moderate/Severe)
patient_disease = {
    'Fever': 1,
    'Tiredness': 1,
    'Dry-Cough': 1,
    'Difficulty-in-Breathing': 1,
    'Sore-Throat': 1,
    'None_Sympton': 0,
    'Pains': 1,
    'Nasal-Congestion': 1,
    'Runny-Nose': 1,
    'Diarrhea': 1,
    'None_Experiencing': 0,
    'Age_0-9': 0,
    'Age_10-19': 0,
    'Age_20-24': 1,
    'Age_25-59': 0,
    'Age_60+': 0,
    'Gender_Female': 0,
    'Gender_Male': 1,
    'Gender_Transgender': 0,
    'Contact_Dont-Know': 0,
    'Contact_No': 0,
    'Contact_Yes': 1
}

# Convert to DataFrames
patient_df = pd.DataFrame([patient_no_disease, patient_disease])

# Predict
predictions = model.predict(patient_df)
predicted_labels = [severity_map[p] for p in predictions]

# Show Results
print("\n--- Prediction Results ---")
print("Patient 1 (No symptoms):", predicted_labels[0]) # Expected: 'None'
print("Patient 2 (With symptoms):", predicted_labels[1]) # Expected: 'Mild/Moderate/Severe'
```



```
--- Prediction Results ---
Patient 1 (No symptoms): None
Patient 2 (With symptoms): Mild
```

```
# Another set of two example patients

# Example 3: Mild symptoms (expected prediction: Mild or Moderate)
patient_example3 = {
    'Fever': 1,
    'Tiredness': 1,
    'Dry-Cough': 0,
    'Difficulty-in-Breathing': 0,
    'Sore-Throat': 1,
    'None_Sympton': 0,
    'Pains': 1,
    'Nasal-Congestion': 0,
    'Runny-Nose': 1,
    'Diarrhea': 0,
    'None_Experiencing': 0,
    'Age_0-9': 0,
    'Age_10-19': 0,
    'Age_20-24': 0,
    'Age_25-59': 1,
    'Age_60+': 0,
    'Gender_Female': 0,
    'Gender_Male': 1,
    'Gender_Transgender': 0,
    'Contact_Dont-Know': 1,
    'Contact_No': 0,
    'Contact_Yes': 0
}
```

```


# Example 4: Severe case (all symptoms + age 60+ + contact with positive case)
patient_example4 = {
    'Fever': 1,
    'Tiredness': 1,
    'Dry-Cough': 1,
    'Difficulty-in-Breathing': 1,
    'Sore-Throat': 1,
    'None_Sympton': 0,
    'Pains': 1,
    'Nasal-Congestion': 1,
    'Runny-Nose': 1,
    'Diarrhea': 1,
    'None_Experiencing': 0,
    'Age_0-9': 0,
    'Age_10-19': 0,
    'Age_20-24': 0,
    'Age_25-59': 0,
    'Age_60+': 1,
    'Gender_Female': 1,
    'Gender_Male': 0,
    'Gender_Transgender': 0,
    'Contact_Dont-Know': 0,
    'Contact_No': 0,
    'Contact_Yes': 1
}

# Convert to DataFrame
patient_df_2 = pd.DataFrame([patient_example3, patient_example4])

# Predict
predictions_2 = model.predict(patient_df_2)
predicted_labels_2 = [severity_map[p] for p in predictions_2]

# Show Results
print("\n--- More Predictions ---")
print("Patient 3 (Mild symptoms):", predicted_labels_2[0]) # Expected: Mild/Moderate
print("Patient 4 (Severe symptoms):", predicted_labels_2[1]) # Expected: Severe

```



```

--- More Predictions ---
Patient 3 (Mild symptoms): None
Patient 4 (Severe symptoms): Mild

```