

```
#!/usr/bin/env python
```

```
# coding: utf-8
```

```
# In[1]:
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import LabelEncoder
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.metrics import classification_report, confusion_matrix
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
# In[3]:
```

```
import pandas as pd
```

```
df=pd.read_csv("Mymensingh.csv")
```

```
df
```

```
# In[4]:
```

```
print(df.isnull().sum)
```

```
# In[5]:
```

```
df = df.drop(columns=['ID', 'Station', 'Year', 'Month', 'A_RAIN', 'T_RAN'])
```

```
# In[7]:
```

```
X = df.drop(columns=['RAN'])
y = df['RAN']
```

```
# In[9]:
```

```
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
```

```
imputer = SimpleImputer(strategy='median')
X_imputed = imputer.fit_transform(X)
```

```
# In[10]:
```

```
le = LabelEncoder()
y_encoded = le.fit_transform(y)
```

```
# In[11]:
```

```
X_train, X_test, y_train, y_test = train_test_split(X_imputed, y_encoded, test_size=0.2,
random_state=42)
```

```
# In[13]:
```

```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
```

```
# In[14]:
```

```
y_pred = knn.predict(X_test)
```

```
# In[15]:
```

```
print("Confusion Matrix:")
```

```
print(confusion_matrix(y_test, y_pred))
```

```
# In[17]:
```

```
plt.figure(figsize=(6, 5))
```

```
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d',
```

```
            xticklabels=le.classes_, yticklabels=le.classes_, cmap='Blues')
```

```
plt.xlabel("Predicted")
```

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

```
plt.title("KNN with Median Imputation")
```

```
plt.tight_layout()
```

```
plt.show()
```

```
# In[20]:
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
# Dummy classifier just to check it works
```

```
knn = KNeighborsClassifier(n_neighbors=5)
```

```
# In[21]:
```

```
knn.fit(X_train, y_train)
```

```
# In[22]:
```

```
print("\nClassification Report:")
```

```
print(classification_report(y_test, y_pred, target_names=le.classes_))
```

```
# In[ ]:
```