```
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, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix, classification_report

df = pd.read_csv(r"Life Expectancy Data.csv")
df.head()
```

```
       Country  Year      Status  Life expectancy   Adult Mortality  \
0  Afghanistan  2015  Developing              65.0            263.0
1  Afghanistan  2014  Developing              59.9            271.0
2  Afghanistan  2013  Developing              59.9            268.0
3  Afghanistan  2012  Developing              59.5            272.0
4  Afghanistan  2011  Developing              59.2            275.0

   infant deaths  Alcohol  percentage expenditure  Hepatitis B
Measles   ...  \
0             62     0.01               71.279624         65.0
1154  ...
1             64     0.01               73.523582         62.0
492  ...
2             66     0.01               73.219243         64.0
430  ...
3             69     0.01               78.184215         67.0
2787  ...
4             71     0.01                7.097109         68.0
3013  ...

   Polio  Total expenditure  Diphtheria    HIV/AIDS         GDP
Population  \
0    6.0               8.16         65.0        0.1  584.259210
33736494.0
1   58.0               8.18         62.0        0.1  612.696514
327582.0
2   62.0               8.13         64.0        0.1  631.744976
31731688.0
3   67.0               8.52         67.0        0.1  669.959000
3696958.0
4   68.0               7.87         68.0        0.1   63.537231
2978599.0
```

```
    thinness  1-19 years   thinness 5-9 years  \
0                   17.2                 17.3
1                   17.5                 17.5
2                   17.7                 17.7
3                   17.9                 18.0
4                   18.2                 18.2

   Income composition of resources  Schooling
0                            0.479       10.1
1                            0.476       10.0
2                            0.470        9.9
3                            0.463        9.8
4                            0.454        9.5

[5 rows x 22 columns]
```

```
# Instead of using inplace=True, assign back to the column directly
df['Year'] = df['Year'].fillna(df['Year'].median())
df['Schooling'] = df['Schooling'].fillna(df['Schooling'].mode()[0])

# Dropping the 'Status' column
df.drop('Status', axis=1, inplace=True)

label_enc = LabelEncoder()
df['Country'] = label_enc.fit_transform(df['Country'])
df['Schooling'] = label_enc.fit_transform(df['Schooling'])

print(df.columns)

X = df.drop('Life expectancy ', axis=1)

y = df['Country']
```

```
Index(['Country', 'Year', 'Life expectancy ', 'Adult Mortality',
       'infant deaths', 'Alcohol', 'percentage expenditure',
'Hepatitis B',
       'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total
expenditure',
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
       ' thinness  1-19 years', ' thinness 5-9 years',
       'Income composition of resources', 'Schooling'],
      dtype='object')
```

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

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
def evaluate_model(y_true, y_pred, model_name):
    accuracy = accuracy_score(y_true, y_pred)
    # Changed to use 'weighted' averaging for multiclass
    precision = precision_score(y_true, y_pred, average='weighted')
    recall = recall_score(y_true, y_pred, average='weighted')
    f1 = f1_score(y_true, y_pred, average='weighted')
    print(f'---{model_name}---')
    print(f'Accuracy: {accuracy:.2f}')
    print(f'Precision: {precision:.2f}')
    print(f'Recall: {recall:.2f}')
    print(f'F1 Score: {f1:.2f}')
    print(f'Confusion Matrix:\n{confusion_matrix(y_true, y_pred)}')
    print('-'*40)
```

1. Logistic Regression

```
from sklearn.impute import SimpleImputer
import warnings

warnings.filterwarnings('ignore')
imputer = SimpleImputer(strategy='mean')

X_train = imputer.fit_transform(X_train)
X_test = imputer.transform(X_test)

log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)
y_pred_lr = log_reg.predict(X_test)
evaluate_model(y_test, y_pred_lr, "Logistic Regression")
```

```
---Logistic Regression---
Accuracy: 0.76
Precision: 0.82
Recall: 0.76
F1 Score: 0.76
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 1 1]]
----------------------------------------
```

2. Naive Bayes

```
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)
```

```
y_pred_nb = nb_model.predict(X_test)
evaluate_model(y_test, y_pred_nb, "Naive Bayes")
```

```
---Naive Bayes---
Accuracy: 0.99
Precision: 0.99
Recall: 0.99
F1 Score: 0.99
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 3]]
----------------------------------------
```

3. K-Nearest Neighbors (KNN)

```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)
evaluate_model(y_test, y_pred_knn, "K-Nearest Neighbors")
```

```
---K-Nearest Neighbors---
Accuracy: 0.62
Precision: 0.69
Recall: 0.62
F1 Score: 0.62
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 1 0]]
----------------------------------------
```

4. Decision Tree

```
dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(X_train, y_train)
y_pred_dt = dt_model.predict(X_test)
evaluate_model(y_test, y_pred_dt, "Decision Tree")
```

```
---Decision Tree---
Accuracy: 0.98
Precision: 0.98
```

```
Recall: 0.98
F1 Score: 0.97
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 3]]
----------------------------------------
```

5. Random Forest

```
rf_model = RandomForestClassifier(random_state=42, n_estimators=100)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
evaluate_model(y_test, y_pred_rf, "Random Forest")
```

```
---Random Forest---
Accuracy: 0.99
Precision: 0.99
Recall: 0.99
F1 Score: 0.99
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 3]]
----------------------------------------
```

6. KMeans (Unsupervised Clustering) for analysis, not classification

```
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans.fit(X_train)
y_pred_kmeans = kmeans.predict(X_test)

y_pred_kmeans = np.where(y_pred_kmeans == 0, 1, 0)

evaluate_model(y_test, y_pred_kmeans, "KMeans Clustering (adjusted for
survival)")
```

```
---KMeans Clustering (adjusted for survival)---
Accuracy: 0.01
Precision: 0.00
Recall: 0.01
F1 Score: 0.00
```

```
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [2 2 0 ... 0 0 0]
 ...
 [4 0 0 ... 0 0 0]
 [1 0 0 ... 0 0 0]
 [3 0 0 ... 0 0 0]]
----------------------------------------
```

```
print("\n--- Classification Reports ---")
print("Logistic Regression:\n", classification_report(y_test,
y_pred_lr))
print("Naive Bayes:\n", classification_report(y_test, y_pred_nb))
print("K-Nearest Neighbors:\n", classification_report(y_test,
y_pred_knn))
print("Decision Tree:\n", classification_report(y_test, y_pred_dt))
print("Random Forest:\n", classification_report(y_test, y_pred_rf))
print("KMeans (Clustering):\n", classification_report(y_test,
y_pred_kmeans))
```

```
--- Classification Reports ---
Logistic Regression:
              precision    recall  f1-score   support

           0       0.50      1.00      0.67         1
           1       0.40      1.00      0.57         2
           2       1.00      1.00      1.00         4
           3       1.00      0.67      0.80         3
           4       0.33      0.33      0.33         3
           5       0.67      0.67      0.67         3
           6       1.00      0.40      0.57         5
           7       1.00      1.00      1.00         2
           8       1.00      1.00      1.00         4
           9       0.50      1.00      0.67         2
          10       1.00      0.60      0.75         5
          11       1.00      1.00      1.00         5
          12       0.80      1.00      0.89         4
          13       0.83      1.00      0.91         5
          14       1.00      0.50      0.67         4
          15       1.00      0.33      0.50         3
          16       0.50      0.25      0.33         4
          17       0.67      0.67      0.67         3
          18       1.00      1.00      1.00         4
          19       1.00      0.75      0.86         4
          20       1.00      1.00      1.00         5
          21       0.00      0.00      0.00         0
          22       0.83      1.00      0.91         5
          23       0.00      0.00      0.00         1
```

24	1.00	0.50	0.67	2
25	0.80	0.57	0.67	7
26	0.50	1.00	0.67	4
27	1.00	0.75	0.86	4
28	1.00	0.50	0.67	4
29	1.00	0.67	0.80	3
30	0.60	0.60	0.60	5
31	0.60	1.00	0.75	3
32	0.50	0.40	0.44	5
33	0.75	0.75	0.75	4
34	1.00	1.00	1.00	6
35	0.00	0.00	0.00	1
36	0.80	0.80	0.80	5
37	0.50	0.50	0.50	2
39	0.00	0.00	0.00	0
40	0.67	1.00	0.80	4
41	0.50	1.00	0.67	1
42	0.80	1.00	0.89	4
43	0.80	1.00	0.89	4
44	1.00	0.80	0.89	5
45	0.50	1.00	0.67	2
46	1.00	1.00	1.00	2
47	0.17	0.50	0.25	2
48	1.00	1.00	1.00	4
50	1.00	1.00	1.00	2
51	0.00	0.00	0.00	1
52	0.67	0.80	0.73	5
53	0.00	0.00	0.00	2
54	1.00	0.33	0.50	3
55	1.00	1.00	1.00	3
56	0.25	0.20	0.22	5
57	1.00	1.00	1.00	2
58	1.00	1.00	1.00	4
59	0.14	1.00	0.25	1
60	0.80	0.80	0.80	5
61	1.00	1.00	1.00	2
62	1.00	1.00	1.00	1
63	1.00	0.50	0.67	2
64	0.00	0.00	0.00	3
65	0.50	0.40	0.44	5
66	0.75	0.50	0.60	6
67	1.00	1.00	1.00	4
68	1.00	0.75	0.86	4
69	1.00	0.20	0.33	5
70	1.00	1.00	1.00	5
71	0.67	0.50	0.57	4
72	0.57	1.00	0.73	4
73	0.33	1.00	0.50	2
74	0.00	0.00	0.00	0

75	0.00	0.00	0.00	1
76	1.00	1.00	1.00	4
77	1.00	0.83	0.91	6
78	1.00	1.00	1.00	7
79	1.00	0.67	0.80	3
80	1.00	1.00	1.00	2
81	0.33	1.00	0.50	1
82	1.00	0.60	0.75	5
83	1.00	1.00	1.00	1
84	0.80	1.00	0.89	4
85	1.00	0.67	0.80	6
86	1.00	1.00	1.00	1
87	1.00	0.71	0.83	7
88	1.00	1.00	1.00	1
89	0.50	0.50	0.50	2
90	1.00	0.50	0.67	4
91	0.57	0.67	0.62	6
92	1.00	0.25	0.40	4
93	0.50	1.00	0.67	2
94	0.00	0.00	0.00	1
95	0.60	0.60	0.60	5
96	0.50	0.67	0.57	3
97	0.60	1.00	0.75	3
98	1.00	1.00	1.00	4
99	0.60	0.75	0.67	4
100	0.00	0.00	0.00	1
101	1.00	1.00	1.00	2
102	1.00	1.00	1.00	7
103	0.00	0.00	0.00	1
104	0.33	1.00	0.50	1
106	0.33	0.50	0.40	2
107	1.00	1.00	1.00	2
108	1.00	1.00	1.00	2
109	0.75	1.00	0.86	3
111	0.50	1.00	0.67	2
112	1.00	0.25	0.40	4
113	0.50	1.00	0.67	1
114	0.83	0.83	0.83	6
115	0.60	1.00	0.75	3
116	1.00	1.00	1.00	4
118	1.00	1.00	1.00	2
119	1.00	0.60	0.75	5
120	1.00	0.67	0.80	3
121	1.00	0.25	0.40	4
122	0.25	0.33	0.29	3
123	1.00	1.00	1.00	3
125	1.00	1.00	1.00	5
126	1.00	1.00	1.00	2
127	1.00	0.75	0.86	4

129	0.50	1.00	0.67	1
130	0.67	1.00	0.80	2
131	0.33	0.50	0.40	2
132	0.33	0.50	0.40	2
133	0.75	1.00	0.86	3
134	1.00	0.50	0.67	2
135	0.67	0.67	0.67	3
136	1.00	1.00	1.00	4
137	0.71	0.83	0.77	6
138	0.67	0.50	0.57	4
139	0.40	0.50	0.44	4
140	1.00	1.00	1.00	1
141	0.00	0.00	0.00	1
143	0.33	0.50	0.40	2
144	1.00	0.75	0.86	4
145	0.75	0.75	0.75	4
146	0.00	0.00	0.00	1
147	0.50	0.33	0.40	3
148	1.00	1.00	1.00	2
149	1.00	0.33	0.50	3
150	1.00	0.33	0.50	3
151	0.00	0.00	0.00	0
152	0.33	0.50	0.40	2
153	1.00	1.00	1.00	1
154	1.00	0.75	0.86	4
155	1.00	1.00	1.00	5
156	0.57	0.80	0.67	5
157	1.00	1.00	1.00	4
158	1.00	1.00	1.00	4
159	1.00	1.00	1.00	3
160	0.50	1.00	0.67	2
161	1.00	1.00	1.00	2
162	1.00	0.67	0.80	3
163	1.00	0.80	0.89	5
164	0.67	1.00	0.80	2
165	1.00	1.00	1.00	2
166	1.00	0.80	0.89	5
167	1.00	1.00	1.00	3
168	1.00	0.50	0.67	2
169	1.00	0.67	0.80	6
170	0.80	0.80	0.80	5
171	0.80	0.80	0.80	5
172	1.00	1.00	1.00	4
173	0.75	0.75	0.75	4
174	1.00	1.00	1.00	3
175	0.75	1.00	0.86	3
176	1.00	0.50	0.67	2
177	0.75	0.75	0.75	4
178	0.00	0.00	0.00	1

179	1.00	0.83	0.91	6
181	0.75	1.00	0.86	3
182	1.00	1.00	1.00	1
183	1.00	0.67	0.80	3
184	1.00	1.00	1.00	3
185	1.00	1.00	1.00	2
186	0.60	1.00	0.75	3
187	1.00	1.00	1.00	2
188	1.00	0.75	0.86	4
189	1.00	1.00	1.00	2
190	1.00	1.00	1.00	4
191	0.50	1.00	0.67	1
192	1.00	0.33	0.50	3
accuracy			0.76	588
macro avg	0.74	0.73	0.70	588
weighted avg	0.82	0.76	0.76	588

Naive Bayes:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
1	1.00	1.00	1.00	2
2	1.00	1.00	1.00	4
3	1.00	1.00	1.00	3
4	1.00	1.00	1.00	3
5	0.60	1.00	0.75	3
6	1.00	0.60	0.75	5
7	1.00	1.00	1.00	2
8	1.00	1.00	1.00	4
9	1.00	1.00	1.00	2
10	1.00	1.00	1.00	5
11	1.00	1.00	1.00	5
12	1.00	1.00	1.00	4
13	1.00	1.00	1.00	5
14	1.00	1.00	1.00	4
15	1.00	1.00	1.00	3
16	1.00	1.00	1.00	4
17	1.00	1.00	1.00	3
18	1.00	1.00	1.00	4
19	1.00	1.00	1.00	4
20	1.00	1.00	1.00	5
22	1.00	1.00	1.00	5
23	1.00	1.00	1.00	1
24	1.00	1.00	1.00	2
25	1.00	1.00	1.00	7
26	1.00	1.00	1.00	4
27	1.00	1.00	1.00	4
28	1.00	1.00	1.00	4
29	1.00	1.00	1.00	3

30	1.00	1.00	1.00	5
31	1.00	1.00	1.00	3
32	1.00	1.00	1.00	5
33	1.00	1.00	1.00	4
34	1.00	1.00	1.00	6
35	1.00	1.00	1.00	1
36	1.00	1.00	1.00	5
37	1.00	1.00	1.00	2
40	1.00	1.00	1.00	4
41	1.00	1.00	1.00	1
42	1.00	1.00	1.00	4
43	1.00	1.00	1.00	4
44	1.00	1.00	1.00	5
45	1.00	1.00	1.00	2
46	1.00	1.00	1.00	2
47	1.00	1.00	1.00	2
48	1.00	1.00	1.00	4
50	1.00	1.00	1.00	2
51	1.00	1.00	1.00	1
52	1.00	1.00	1.00	5
53	1.00	1.00	1.00	2
54	1.00	1.00	1.00	3
55	1.00	1.00	1.00	3
56	1.00	1.00	1.00	5
57	1.00	1.00	1.00	2
58	0.80	1.00	0.89	4
59	0.00	0.00	0.00	1
60	1.00	1.00	1.00	5
61	1.00	1.00	1.00	2
62	1.00	1.00	1.00	1
63	1.00	1.00	1.00	2
64	1.00	1.00	1.00	3
65	1.00	1.00	1.00	5
66	1.00	1.00	1.00	6
67	1.00	1.00	1.00	4
68	1.00	1.00	1.00	4
69	1.00	1.00	1.00	5
70	1.00	1.00	1.00	5
71	1.00	1.00	1.00	4
72	1.00	1.00	1.00	4
73	1.00	1.00	1.00	2
75	1.00	1.00	1.00	1
76	1.00	1.00	1.00	4
77	1.00	1.00	1.00	6
78	1.00	1.00	1.00	7
79	1.00	1.00	1.00	3
80	1.00	1.00	1.00	2
81	1.00	1.00	1.00	1
82	1.00	1.00	1.00	5

83	1.00	1.00	1.00	1
84	1.00	1.00	1.00	4
85	1.00	1.00	1.00	6
86	1.00	1.00	1.00	1
87	1.00	1.00	1.00	7
88	1.00	1.00	1.00	1
89	1.00	1.00	1.00	2
90	1.00	1.00	1.00	4
91	1.00	1.00	1.00	6
92	1.00	1.00	1.00	4
93	1.00	1.00	1.00	2
94	1.00	1.00	1.00	1
95	0.83	1.00	0.91	5
96	1.00	0.67	0.80	3
97	0.75	1.00	0.86	3
98	1.00	0.75	0.86	4
99	1.00	1.00	1.00	4
100	1.00	1.00	1.00	1
101	1.00	1.00	1.00	2
102	1.00	1.00	1.00	7
103	1.00	1.00	1.00	1
104	1.00	1.00	1.00	1
106	1.00	1.00	1.00	2
107	1.00	1.00	1.00	2
108	1.00	1.00	1.00	2
109	1.00	1.00	1.00	3
111	1.00	1.00	1.00	2
112	1.00	1.00	1.00	4
113	1.00	1.00	1.00	1
114	1.00	1.00	1.00	6
115	1.00	1.00	1.00	3
116	1.00	1.00	1.00	4
118	1.00	1.00	1.00	2
119	1.00	1.00	1.00	5
120	1.00	1.00	1.00	3
121	1.00	1.00	1.00	4
122	1.00	1.00	1.00	3
123	1.00	1.00	1.00	3
125	1.00	1.00	1.00	5
126	1.00	1.00	1.00	2
127	1.00	1.00	1.00	4
129	1.00	1.00	1.00	1
130	1.00	1.00	1.00	2
131	1.00	1.00	1.00	2
132	1.00	1.00	1.00	2
133	1.00	1.00	1.00	3
134	1.00	1.00	1.00	2
135	1.00	1.00	1.00	3
136	1.00	1.00	1.00	4

137	1.00	1.00	1.00	6
138	1.00	1.00	1.00	4
139	1.00	1.00	1.00	4
140	1.00	1.00	1.00	1
141	1.00	1.00	1.00	1
143	1.00	1.00	1.00	2
144	1.00	1.00	1.00	4
145	1.00	1.00	1.00	4
146	0.00	0.00	0.00	1
147	0.75	1.00	0.86	3
148	1.00	1.00	1.00	2
149	1.00	1.00	1.00	3
150	1.00	1.00	1.00	3
152	1.00	1.00	1.00	2
153	1.00	1.00	1.00	1
154	1.00	1.00	1.00	4
155	1.00	1.00	1.00	5
156	1.00	1.00	1.00	5
157	1.00	1.00	1.00	4
158	1.00	1.00	1.00	4
159	1.00	1.00	1.00	3
160	1.00	1.00	1.00	2
161	1.00	1.00	1.00	2
162	1.00	1.00	1.00	3
163	1.00	1.00	1.00	5
164	1.00	1.00	1.00	2
165	1.00	1.00	1.00	2
166	1.00	1.00	1.00	5
167	1.00	1.00	1.00	3
168	1.00	1.00	1.00	2
169	1.00	1.00	1.00	6
170	1.00	1.00	1.00	5
171	1.00	1.00	1.00	5
172	1.00	1.00	1.00	4
173	1.00	1.00	1.00	4
174	1.00	1.00	1.00	3
175	1.00	1.00	1.00	3
176	1.00	1.00	1.00	2
177	1.00	1.00	1.00	4
178	0.00	0.00	0.00	1
179	0.86	1.00	0.92	6
181	1.00	1.00	1.00	3
182	1.00	1.00	1.00	1
183	1.00	1.00	1.00	3
184	1.00	1.00	1.00	3
185	1.00	1.00	1.00	2
186	1.00	1.00	1.00	3
187	1.00	1.00	1.00	2
188	1.00	1.00	1.00	4

```
         189       1.00      1.00      1.00         2
         190       1.00      1.00      1.00         4
         191       1.00      1.00      1.00         1
         192       1.00      1.00      1.00         3

    accuracy                           0.99       588
   macro avg       0.98      0.98      0.98       588
weighted avg       0.99      0.99      0.99       588

K-Nearest Neighbors:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       0.33      1.00      0.50         2
           2       0.57      1.00      0.73         4
           3       1.00      0.67      0.80         3
           4       0.67      0.67      0.67         3
           5       1.00      0.67      0.80         3
           6       1.00      0.40      0.57         5
           7       0.50      0.50      0.50         2
           8       0.57      1.00      0.73         4
           9       0.50      0.50      0.50         2
          10       0.40      0.40      0.40         5
          11       1.00      0.60      0.75         5
          12       0.60      0.75      0.67         4
          13       0.83      1.00      0.91         5
          14       1.00      0.50      0.67         4
          15       0.22      0.67      0.33         3
          16       0.50      0.25      0.33         4
          17       0.33      0.67      0.44         3
          18       1.00      1.00      1.00         4
          19       1.00      0.75      0.86         4
          20       0.25      0.20      0.22         5
          22       0.80      0.80      0.80         5
          23       0.00      0.00      0.00         1
          24       0.25      0.50      0.33         2
          25       0.25      0.14      0.18         7
          26       0.43      0.75      0.55         4
          27       1.00      0.75      0.86         4
          28       1.00      0.25      0.40         4
          29       0.20      0.33      0.25         3
          30       0.50      0.20      0.29         5
          31       0.50      0.67      0.57         3
          32       0.29      0.40      0.33         5
          33       1.00      0.75      0.86         4
          34       1.00      1.00      1.00         6
          35       0.00      0.00      0.00         1
          36       0.40      0.40      0.40         5
          37       0.25      0.50      0.33         2
          39       0.00      0.00      0.00         0
```

40	0.43	0.75	0.55	4
41	1.00	1.00	1.00	1
42	0.33	0.25	0.29	4
43	1.00	0.75	0.86	4
44	0.57	0.80	0.67	5
45	0.29	1.00	0.44	2
46	1.00	0.50	0.67	2
47	0.00	0.00	0.00	2
48	1.00	1.00	1.00	4
49	0.00	0.00	0.00	0
50	0.33	0.50	0.40	2
51	0.00	0.00	0.00	1
52	0.60	0.60	0.60	5
53	0.00	0.00	0.00	2
54	1.00	0.33	0.50	3
55	0.67	0.67	0.67	3
56	1.00	0.80	0.89	5
57	0.33	1.00	0.50	2
58	0.75	0.75	0.75	4
59	0.00	0.00	0.00	1
60	0.60	0.60	0.60	5
61	1.00	1.00	1.00	2
62	0.25	1.00	0.40	1
63	1.00	1.00	1.00	2
64	0.50	0.33	0.40	3
65	0.60	0.60	0.60	5
66	1.00	0.33	0.50	6
67	0.80	1.00	0.89	4
68	0.50	0.75	0.60	4
69	0.67	0.40	0.50	5
70	0.50	0.80	0.62	5
71	0.67	0.50	0.57	4
72	0.67	1.00	0.80	4
73	0.25	0.50	0.33	2
75	0.25	1.00	0.40	1
76	1.00	1.00	1.00	4
77	1.00	0.33	0.50	6
78	1.00	1.00	1.00	7
79	0.50	0.67	0.57	3
80	0.00	0.00	0.00	2
81	0.50	1.00	0.67	1
82	0.33	0.20	0.25	5
83	0.17	1.00	0.29	1
84	0.60	0.75	0.67	4
85	1.00	0.67	0.80	6
86	1.00	1.00	1.00	1
87	1.00	0.57	0.73	7
88	0.00	0.00	0.00	1
89	0.50	1.00	0.67	2

90	0.75	0.75	0.75	4
91	0.50	0.50	0.50	6
92	1.00	0.50	0.67	4
93	0.50	1.00	0.67	2
94	0.00	0.00	0.00	1
95	0.67	0.40	0.50	5
96	0.67	0.67	0.67	3
97	0.67	0.67	0.67	3
98	0.75	0.75	0.75	4
99	0.50	0.75	0.60	4
100	0.00	0.00	0.00	1
101	1.00	0.50	0.67	2
102	1.00	1.00	1.00	7
103	0.00	0.00	0.00	1
104	0.00	0.00	0.00	1
106	0.33	0.50	0.40	2
107	0.67	1.00	0.80	2
108	0.00	0.00	0.00	2
109	1.00	0.67	0.80	3
111	0.50	0.50	0.50	2
112	0.67	0.50	0.57	4
113	1.00	1.00	1.00	1
114	0.60	0.50	0.55	6
115	0.50	0.67	0.57	3
116	0.60	0.75	0.67	4
118	1.00	1.00	1.00	2
119	0.33	0.20	0.25	5
120	0.00	0.00	0.00	3
121	0.50	0.25	0.33	4
122	0.50	0.33	0.40	3
123	1.00	1.00	1.00	3
125	0.67	0.40	0.50	5
126	1.00	1.00	1.00	2
127	1.00	0.75	0.86	4
129	0.50	1.00	0.67	1
130	1.00	1.00	1.00	2
131	0.00	0.00	0.00	2
132	1.00	0.50	0.67	2
133	0.50	0.33	0.40	3
134	1.00	1.00	1.00	2
135	0.33	0.33	0.33	3
136	1.00	0.50	0.67	4
137	1.00	0.83	0.91	6
138	1.00	0.75	0.86	4
139	0.75	0.75	0.75	4
140	1.00	1.00	1.00	1
141	0.00	0.00	0.00	1
143	0.33	0.50	0.40	2
144	1.00	0.25	0.40	4
145	0.67	1.00	0.80	4

146	0.00	0.00	0.00	1
147	0.50	0.67	0.57	3
148	1.00	1.00	1.00	2
149	1.00	0.67	0.80	3
150	0.75	1.00	0.86	3
151	0.00	0.00	0.00	0
152	0.50	0.50	0.50	2
153	1.00	1.00	1.00	1
154	0.75	0.75	0.75	4
155	0.75	0.60	0.67	5
156	0.67	0.40	0.50	5
157	0.80	1.00	0.89	4
158	1.00	0.50	0.67	4
159	1.00	1.00	1.00	3
160	1.00	0.50	0.67	2
161	1.00	1.00	1.00	2
162	1.00	0.67	0.80	3
163	1.00	0.40	0.57	5
164	0.00	0.00	0.00	2
165	1.00	0.50	0.67	2
166	1.00	0.60	0.75	5
167	1.00	0.67	0.80	3
168	1.00	1.00	1.00	2
169	1.00	1.00	1.00	6
170	0.67	0.80	0.73	5
171	1.00	0.60	0.75	5
172	0.75	0.75	0.75	4
173	0.00	0.00	0.00	4
174	1.00	0.33	0.50	3
175	1.00	0.67	0.80	3
176	0.50	0.50	0.50	2
177	0.75	0.75	0.75	4
178	0.00	0.00	0.00	1
179	0.67	0.33	0.44	6
180	0.00	0.00	0.00	0
181	0.60	1.00	0.75	3
182	1.00	1.00	1.00	1
183	0.00	0.00	0.00	3
184	1.00	1.00	1.00	3
185	1.00	1.00	1.00	2
186	0.60	1.00	0.75	3
187	1.00	1.00	1.00	2
188	1.00	0.25	0.40	4
189	1.00	1.00	1.00	2
190	1.00	1.00	1.00	4
191	0.33	1.00	0.50	1
192	0.00	0.00	0.00	3
accuracy			0.62	588

```
   macro avg       0.62      0.60      0.58       588
weighted avg       0.69      0.62      0.62       588

Decision Tree:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         2
           2       1.00      1.00      1.00         4
           3       1.00      1.00      1.00         3
           4       1.00      1.00      1.00         3
           5       1.00      1.00      1.00         3
           6       1.00      1.00      1.00         5
           7       1.00      1.00      1.00         2
           8       1.00      1.00      1.00         4
           9       1.00      1.00      1.00         2
          10       1.00      0.80      0.89         5
          11       1.00      1.00      1.00         5
          12       1.00      1.00      1.00         4
          13       0.83      1.00      0.91         5
          14       0.75      0.75      0.75         4
          15       1.00      1.00      1.00         3
          16       1.00      1.00      1.00         4
          17       1.00      1.00      1.00         3
          18       1.00      1.00      1.00         4
          19       1.00      1.00      1.00         4
          20       1.00      1.00      1.00         5
          22       1.00      1.00      1.00         5
          23       1.00      1.00      1.00         1
          24       1.00      1.00      1.00         2
          25       0.88      1.00      0.93         7
          26       0.67      1.00      0.80         4
          27       1.00      1.00      1.00         4
          28       1.00      0.50      0.67         4
          29       1.00      0.67      0.80         3
          30       1.00      1.00      1.00         5
          31       1.00      1.00      1.00         3
          32       1.00      1.00      1.00         5
          33       1.00      1.00      1.00         4
          34       1.00      1.00      1.00         6
          35       1.00      1.00      1.00         1
          36       1.00      1.00      1.00         5
          37       1.00      1.00      1.00         2
          40       1.00      1.00      1.00         4
          41       1.00      1.00      1.00         1
          42       1.00      1.00      1.00         4
          43       1.00      1.00      1.00         4
          44       1.00      1.00      1.00         5
          45       1.00      1.00      1.00         2
```

46	1.00	1.00	1.00	2
47	1.00	1.00	1.00	2
48	1.00	1.00	1.00	4
50	1.00	1.00	1.00	2
51	1.00	1.00	1.00	1
52	1.00	1.00	1.00	5
53	1.00	1.00	1.00	2
54	1.00	1.00	1.00	3
55	1.00	1.00	1.00	3
56	1.00	1.00	1.00	5
57	1.00	1.00	1.00	2
58	1.00	1.00	1.00	4
59	1.00	1.00	1.00	1
60	1.00	1.00	1.00	5
61	1.00	1.00	1.00	2
62	1.00	1.00	1.00	1
63	1.00	1.00	1.00	2
64	1.00	1.00	1.00	3
65	0.71	1.00	0.83	5
66	1.00	1.00	1.00	6
67	1.00	1.00	1.00	4
68	1.00	1.00	1.00	4
69	1.00	0.60	0.75	5
70	1.00	1.00	1.00	5
71	1.00	1.00	1.00	4
72	1.00	1.00	1.00	4
73	1.00	1.00	1.00	2
75	1.00	1.00	1.00	1
76	1.00	1.00	1.00	4
77	1.00	1.00	1.00	6
78	1.00	1.00	1.00	7
79	1.00	1.00	1.00	3
80	1.00	1.00	1.00	2
81	1.00	1.00	1.00	1
82	1.00	1.00	1.00	5
83	1.00	1.00	1.00	1
84	1.00	1.00	1.00	4
85	1.00	1.00	1.00	6
86	1.00	1.00	1.00	1
87	1.00	1.00	1.00	7
88	1.00	1.00	1.00	1
89	1.00	1.00	1.00	2
90	1.00	1.00	1.00	4
91	1.00	1.00	1.00	6
92	1.00	1.00	1.00	4
93	1.00	1.00	1.00	2
94	1.00	1.00	1.00	1
95	1.00	1.00	1.00	5
96	1.00	0.67	0.80	3

97	0.75	1.00	0.86	3
98	1.00	1.00	1.00	4
99	1.00	1.00	1.00	4
100	1.00	1.00	1.00	1
101	1.00	1.00	1.00	2
102	1.00	1.00	1.00	7
103	1.00	1.00	1.00	1
104	1.00	1.00	1.00	1
106	1.00	1.00	1.00	2
107	1.00	1.00	1.00	2
108	1.00	1.00	1.00	2
109	1.00	1.00	1.00	3
111	1.00	1.00	1.00	2
112	1.00	1.00	1.00	4
113	1.00	1.00	1.00	1
114	1.00	0.67	0.80	6
115	1.00	1.00	1.00	3
116	1.00	1.00	1.00	4
118	1.00	1.00	1.00	2
119	1.00	1.00	1.00	5
120	1.00	1.00	1.00	3
121	1.00	1.00	1.00	4
122	0.60	1.00	0.75	3
123	1.00	1.00	1.00	3
125	1.00	1.00	1.00	5
126	1.00	1.00	1.00	2
127	1.00	1.00	1.00	4
129	1.00	1.00	1.00	1
130	1.00	1.00	1.00	2
131	1.00	1.00	1.00	2
132	1.00	1.00	1.00	2
133	1.00	1.00	1.00	3
134	1.00	1.00	1.00	2
135	1.00	1.00	1.00	3
136	0.80	1.00	0.89	4
137	1.00	1.00	1.00	6
138	1.00	1.00	1.00	4
139	1.00	1.00	1.00	4
140	1.00	1.00	1.00	1
141	1.00	1.00	1.00	1
143	0.50	0.50	0.50	2
144	1.00	1.00	1.00	4
145	1.00	1.00	1.00	4
146	0.00	0.00	0.00	1
147	1.00	1.00	1.00	3
148	1.00	1.00	1.00	2
149	1.00	1.00	1.00	3
150	1.00	1.00	1.00	3
152	1.00	1.00	1.00	2

```
         153       1.00      1.00      1.00         1
         154       1.00      1.00      1.00         4
         155       1.00      1.00      1.00         5
         156       1.00      1.00      1.00         5
         157       1.00      1.00      1.00         4
         158       1.00      1.00      1.00         4
         159       1.00      1.00      1.00         3
         160       1.00      1.00      1.00         2
         161       1.00      1.00      1.00         2
         162       1.00      1.00      1.00         3
         163       1.00      0.80      0.89         5
         164       1.00      1.00      1.00         2
         165       1.00      1.00      1.00         2
         166       1.00      1.00      1.00         5
         167       1.00      1.00      1.00         3
         168       1.00      1.00      1.00         2
         169       1.00      1.00      1.00         6
         170       1.00      1.00      1.00         5
         171       0.83      1.00      0.91         5
         172       1.00      1.00      1.00         4
         173       1.00      1.00      1.00         4
         174       0.75      1.00      0.86         3
         175       1.00      1.00      1.00         3
         176       1.00      1.00      1.00         2
         177       1.00      1.00      1.00         4
         178       0.00      0.00      0.00         1
         179       1.00      1.00      1.00         6
         181       1.00      1.00      1.00         3
         182       1.00      1.00      1.00         1
         183       1.00      1.00      1.00         3
         184       1.00      1.00      1.00         3
         185       1.00      1.00      1.00         2
         186       1.00      1.00      1.00         3
         187       1.00      1.00      1.00         2
         188       1.00      1.00      1.00         4
         189       1.00      1.00      1.00         2
         190       1.00      1.00      1.00         4
         191       1.00      1.00      1.00         1
         192       1.00      1.00      1.00         3

    accuracy                           0.98       588
   macro avg       0.97      0.97      0.97       588
weighted avg       0.98      0.98      0.97       588

Random Forest:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         2
           2       1.00      1.00      1.00         4
```

3	1.00	1.00	1.00	3
4	1.00	1.00	1.00	3
5	1.00	1.00	1.00	3
6	1.00	1.00	1.00	5
7	1.00	1.00	1.00	2
8	1.00	1.00	1.00	4
9	1.00	1.00	1.00	2
10	1.00	1.00	1.00	5
11	1.00	1.00	1.00	5
12	1.00	1.00	1.00	4
13	1.00	1.00	1.00	5
14	1.00	1.00	1.00	4
15	1.00	1.00	1.00	3
16	1.00	1.00	1.00	4
17	1.00	1.00	1.00	3
18	1.00	1.00	1.00	4
19	1.00	1.00	1.00	4
20	1.00	1.00	1.00	5
22	1.00	1.00	1.00	5
23	1.00	1.00	1.00	1
24	1.00	1.00	1.00	2
25	1.00	1.00	1.00	7
26	1.00	1.00	1.00	4
27	1.00	1.00	1.00	4
28	1.00	1.00	1.00	4
29	1.00	1.00	1.00	3
30	1.00	1.00	1.00	5
31	1.00	1.00	1.00	3
32	1.00	1.00	1.00	5
33	1.00	1.00	1.00	4
34	1.00	1.00	1.00	6
35	1.00	1.00	1.00	1
36	1.00	1.00	1.00	5
37	1.00	1.00	1.00	2
40	1.00	1.00	1.00	4
41	1.00	1.00	1.00	1
42	1.00	1.00	1.00	4
43	1.00	1.00	1.00	4
44	1.00	1.00	1.00	5
45	1.00	1.00	1.00	2
46	1.00	1.00	1.00	2
47	1.00	1.00	1.00	2
48	1.00	1.00	1.00	4
50	1.00	1.00	1.00	2
51	1.00	1.00	1.00	1
52	1.00	1.00	1.00	5
53	1.00	1.00	1.00	2
54	1.00	1.00	1.00	3
55	1.00	1.00	1.00	3

56	1.00	1.00	1.00	5
57	1.00	1.00	1.00	2
58	1.00	1.00	1.00	4
59	1.00	1.00	1.00	1
60	1.00	0.80	0.89	5
61	1.00	1.00	1.00	2
62	1.00	1.00	1.00	1
63	1.00	1.00	1.00	2
64	1.00	1.00	1.00	3
65	1.00	1.00	1.00	5
66	1.00	1.00	1.00	6
67	1.00	1.00	1.00	4
68	1.00	1.00	1.00	4
69	1.00	1.00	1.00	5
70	1.00	1.00	1.00	5
71	1.00	1.00	1.00	4
72	1.00	1.00	1.00	4
73	1.00	1.00	1.00	2
75	1.00	1.00	1.00	1
76	1.00	1.00	1.00	4
77	1.00	1.00	1.00	6
78	1.00	1.00	1.00	7
79	1.00	1.00	1.00	3
80	1.00	1.00	1.00	2
81	1.00	1.00	1.00	1
82	0.83	1.00	0.91	5
83	1.00	1.00	1.00	1
84	1.00	1.00	1.00	4
85	1.00	1.00	1.00	6
86	1.00	1.00	1.00	1
87	1.00	1.00	1.00	7
88	1.00	1.00	1.00	1
89	1.00	1.00	1.00	2
90	1.00	1.00	1.00	4
91	1.00	1.00	1.00	6
92	1.00	1.00	1.00	4
93	1.00	1.00	1.00	2
94	1.00	1.00	1.00	1
95	1.00	1.00	1.00	5
96	1.00	1.00	1.00	3
97	1.00	1.00	1.00	3
98	1.00	1.00	1.00	4
99	1.00	1.00	1.00	4
100	1.00	1.00	1.00	1
101	1.00	1.00	1.00	2
102	1.00	1.00	1.00	7
103	1.00	1.00	1.00	1
104	1.00	1.00	1.00	1
106	1.00	1.00	1.00	2

107	1.00	1.00	1.00	2
108	1.00	1.00	1.00	2
109	1.00	1.00	1.00	3
111	1.00	1.00	1.00	2
112	1.00	1.00	1.00	4
113	1.00	1.00	1.00	1
114	1.00	1.00	1.00	6
115	1.00	1.00	1.00	3
116	1.00	1.00	1.00	4
118	1.00	1.00	1.00	2
119	1.00	1.00	1.00	5
120	1.00	1.00	1.00	3
121	1.00	1.00	1.00	4
122	1.00	1.00	1.00	3
123	1.00	1.00	1.00	3
125	1.00	1.00	1.00	5
126	1.00	1.00	1.00	2
127	1.00	1.00	1.00	4
129	1.00	1.00	1.00	1
130	1.00	1.00	1.00	2
131	1.00	1.00	1.00	2
132	1.00	1.00	1.00	2
133	1.00	1.00	1.00	3
134	1.00	1.00	1.00	2
135	1.00	1.00	1.00	3
136	1.00	1.00	1.00	4
137	1.00	1.00	1.00	6
138	1.00	1.00	1.00	4
139	1.00	1.00	1.00	4
140	1.00	1.00	1.00	1
141	1.00	1.00	1.00	1
143	0.67	1.00	0.80	2
144	1.00	1.00	1.00	4
145	1.00	1.00	1.00	4
146	0.00	0.00	0.00	1
147	1.00	1.00	1.00	3
148	1.00	1.00	1.00	2
149	1.00	1.00	1.00	3
150	1.00	1.00	1.00	3
152	1.00	1.00	1.00	2
153	1.00	1.00	1.00	1
154	1.00	1.00	1.00	4
155	1.00	1.00	1.00	5
156	1.00	1.00	1.00	5
157	1.00	1.00	1.00	4
158	1.00	1.00	1.00	4
159	1.00	1.00	1.00	3
160	1.00	1.00	1.00	2
161	1.00	1.00	1.00	2

```
          162       1.00      1.00      1.00         3
          163       1.00      1.00      1.00         5
          164       1.00      1.00      1.00         2
          165       1.00      1.00      1.00         2
          166       1.00      1.00      1.00         5
          167       1.00      1.00      1.00         3
          168       1.00      1.00      1.00         2
          169       1.00      1.00      1.00         6
          170       1.00      1.00      1.00         5
          171       1.00      1.00      1.00         5
          172       1.00      1.00      1.00         4
          173       0.80      1.00      0.89         4
          174       1.00      1.00      1.00         3
          175       1.00      1.00      1.00         3
          176       1.00      1.00      1.00         2
          177       1.00      1.00      1.00         4
          178       0.00      0.00      0.00         1
          179       1.00      1.00      1.00         6
          181       1.00      1.00      1.00         3
          182       1.00      1.00      1.00         1
          183       1.00      1.00      1.00         3
          184       1.00      1.00      1.00         3
          185       1.00      1.00      1.00         2
          186       1.00      1.00      1.00         3
          187       1.00      1.00      1.00         2
          188       1.00      1.00      1.00         4
          189       1.00      1.00      1.00         2
          190       1.00      1.00      1.00         4
          191       1.00      1.00      1.00         1
          192       1.00      1.00      1.00         3

    accuracy                           0.99       588
   macro avg       0.99      0.99      0.99       588
weighted avg       0.99      0.99      0.99       588

KMeans (Clustering):
              precision    recall  f1-score   support

           0       0.00      1.00      0.01         1
           1       0.01      1.00      0.01         2
           2       0.00      0.00      0.00         4
           3       0.00      0.00      0.00         3
           4       0.00      0.00      0.00         3
           5       0.00      0.00      0.00         3
           6       0.00      0.00      0.00         5
           7       0.00      0.00      0.00         2
           8       0.00      0.00      0.00         4
           9       0.00      0.00      0.00         2
          10       0.00      0.00      0.00         5
          11       0.00      0.00      0.00         5
```

12	0.00	0.00	0.00	4
13	0.00	0.00	0.00	5
14	0.00	0.00	0.00	4
15	0.00	0.00	0.00	3
16	0.00	0.00	0.00	4
17	0.00	0.00	0.00	3
18	0.00	0.00	0.00	4
19	0.00	0.00	0.00	4
20	0.00	0.00	0.00	5
22	0.00	0.00	0.00	5
23	0.00	0.00	0.00	1
24	0.00	0.00	0.00	2
25	0.00	0.00	0.00	7
26	0.00	0.00	0.00	4
27	0.00	0.00	0.00	4
28	0.00	0.00	0.00	4
29	0.00	0.00	0.00	3
30	0.00	0.00	0.00	5
31	0.00	0.00	0.00	3
32	0.00	0.00	0.00	5
33	0.00	0.00	0.00	4
34	0.00	0.00	0.00	6
35	0.00	0.00	0.00	1
36	0.00	0.00	0.00	5
37	0.00	0.00	0.00	2
40	0.00	0.00	0.00	4
41	0.00	0.00	0.00	1
42	0.00	0.00	0.00	4
43	0.00	0.00	0.00	4
44	0.00	0.00	0.00	5
45	0.00	0.00	0.00	2
46	0.00	0.00	0.00	2
47	0.00	0.00	0.00	2
48	0.00	0.00	0.00	4
50	0.00	0.00	0.00	2
51	0.00	0.00	0.00	1
52	0.00	0.00	0.00	5
53	0.00	0.00	0.00	2
54	0.00	0.00	0.00	3
55	0.00	0.00	0.00	3
56	0.00	0.00	0.00	5
57	0.00	0.00	0.00	2
58	0.00	0.00	0.00	4
59	0.00	0.00	0.00	1
60	0.00	0.00	0.00	5
61	0.00	0.00	0.00	2
62	0.00	0.00	0.00	1
63	0.00	0.00	0.00	2
64	0.00	0.00	0.00	3

65	0.00	0.00	0.00	5
66	0.00	0.00	0.00	6
67	0.00	0.00	0.00	4
68	0.00	0.00	0.00	4
69	0.00	0.00	0.00	5
70	0.00	0.00	0.00	5
71	0.00	0.00	0.00	4
72	0.00	0.00	0.00	4
73	0.00	0.00	0.00	2
75	0.00	0.00	0.00	1
76	0.00	0.00	0.00	4
77	0.00	0.00	0.00	6
78	0.00	0.00	0.00	7
79	0.00	0.00	0.00	3
80	0.00	0.00	0.00	2
81	0.00	0.00	0.00	1
82	0.00	0.00	0.00	5
83	0.00	0.00	0.00	1
84	0.00	0.00	0.00	4
85	0.00	0.00	0.00	6
86	0.00	0.00	0.00	1
87	0.00	0.00	0.00	7
88	0.00	0.00	0.00	1
89	0.00	0.00	0.00	2
90	0.00	0.00	0.00	4
91	0.00	0.00	0.00	6
92	0.00	0.00	0.00	4
93	0.00	0.00	0.00	2
94	0.00	0.00	0.00	1
95	0.00	0.00	0.00	5
96	0.00	0.00	0.00	3
97	0.00	0.00	0.00	3
98	0.00	0.00	0.00	4
99	0.00	0.00	0.00	4
100	0.00	0.00	0.00	1
101	0.00	0.00	0.00	2
102	0.00	0.00	0.00	7
103	0.00	0.00	0.00	1
104	0.00	0.00	0.00	1
106	0.00	0.00	0.00	2
107	0.00	0.00	0.00	2
108	0.00	0.00	0.00	2
109	0.00	0.00	0.00	3
111	0.00	0.00	0.00	2
112	0.00	0.00	0.00	4
113	0.00	0.00	0.00	1
114	0.00	0.00	0.00	6
115	0.00	0.00	0.00	3
116	0.00	0.00	0.00	4
118	0.00	0.00	0.00	2

119	0.00	0.00	0.00	5
120	0.00	0.00	0.00	3
121	0.00	0.00	0.00	4
122	0.00	0.00	0.00	3
123	0.00	0.00	0.00	3
125	0.00	0.00	0.00	5
126	0.00	0.00	0.00	2
127	0.00	0.00	0.00	4
129	0.00	0.00	0.00	1
130	0.00	0.00	0.00	2
131	0.00	0.00	0.00	2
132	0.00	0.00	0.00	2
133	0.00	0.00	0.00	3
134	0.00	0.00	0.00	2
135	0.00	0.00	0.00	3
136	0.00	0.00	0.00	4
137	0.00	0.00	0.00	6
138	0.00	0.00	0.00	4
139	0.00	0.00	0.00	4
140	0.00	0.00	0.00	1
141	0.00	0.00	0.00	1
143	0.00	0.00	0.00	2
144	0.00	0.00	0.00	4
145	0.00	0.00	0.00	4
146	0.00	0.00	0.00	1
147	0.00	0.00	0.00	3
148	0.00	0.00	0.00	2
149	0.00	0.00	0.00	3
150	0.00	0.00	0.00	3
152	0.00	0.00	0.00	2
153	0.00	0.00	0.00	1
154	0.00	0.00	0.00	4
155	0.00	0.00	0.00	5
156	0.00	0.00	0.00	5
157	0.00	0.00	0.00	4
158	0.00	0.00	0.00	4
159	0.00	0.00	0.00	3
160	0.00	0.00	0.00	2
161	0.00	0.00	0.00	2
162	0.00	0.00	0.00	3
163	0.00	0.00	0.00	5
164	0.00	0.00	0.00	2
165	0.00	0.00	0.00	2
166	0.00	0.00	0.00	5
167	0.00	0.00	0.00	3
168	0.00	0.00	0.00	2
169	0.00	0.00	0.00	6
170	0.00	0.00	0.00	5
171	0.00	0.00	0.00	5

```
          172       0.00      0.00      0.00         4
          173       0.00      0.00      0.00         4
          174       0.00      0.00      0.00         3
          175       0.00      0.00      0.00         3
          176       0.00      0.00      0.00         2
          177       0.00      0.00      0.00         4
          178       0.00      0.00      0.00         1
          179       0.00      0.00      0.00         6
          181       0.00      0.00      0.00         3
          182       0.00      0.00      0.00         1
          183       0.00      0.00      0.00         3
          184       0.00      0.00      0.00         3
          185       0.00      0.00      0.00         2
          186       0.00      0.00      0.00         3
          187       0.00      0.00      0.00         2
          188       0.00      0.00      0.00         4
          189       0.00      0.00      0.00         2
          190       0.00      0.00      0.00         4
          191       0.00      0.00      0.00         1
          192       0.00      0.00      0.00         3

     accuracy                           0.01       588
    macro avg       0.00      0.01      0.00       588
 weighted avg       0.00      0.01      0.00       588
```

Evaluate the performance using classification metrics.

```python
def evaluate_model(y_true, y_pred, model_name):
    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred, average='weighted')
    recall = recall_score(y_true, y_pred, average='weighted')
    f1 = f1_score(y_true, y_pred, average='weighted')

    print(f'---{model_name}---')
    print(f'Accuracy: {accuracy:.2f}')
    print(f'Precision: {precision:.2f}')
    print(f'Recall: {recall:.2f}')
    print(f'F1 Score: {f1:.2f}')
    print(f'Confusion Matrix:\n{confusion_matrix(y_true, y_pred)}')
    print('-'*40)

# 1. Logistic Regression
y_pred_lr = log_reg.predict(X_test)
evaluate_model(y_test, y_pred_lr, "Logistic Regression")
```

```python
# 2. Naive Bayes
y_pred_nb = nb_model.predict(X_test)
evaluate_model(y_test, y_pred_nb, "Naive Bayes")

# 3. K-Nearest Neighbors (KNN)
y_pred_knn = knn.predict(X_test)
evaluate_model(y_test, y_pred_knn, "K-Nearest Neighbors")

# 4. Decision Tree
y_pred_dt = dt_model.predict(X_test)
evaluate_model(y_test, y_pred_dt, "Decision Tree")

# 5. Random Forest
y_pred_rf = rf_model.predict(X_test)
evaluate_model(y_test, y_pred_rf, "Random Forest")

# 6. KMeans (Clustering)
y_pred_kmeans = np.where(kmeans.predict(X_test) == 0, 1, 0)
evaluate_model(y_test, y_pred_kmeans, "KMeans Clustering")
```

```
---Logistic Regression---
Accuracy: 0.76
Precision: 0.82
Recall: 0.76
F1 Score: 0.76
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 1 1]]
----------------------------------------
---Naive Bayes---
Accuracy: 0.99
Precision: 0.99
Recall: 0.99
F1 Score: 0.99
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 3]]
----------------------------------------
---K-Nearest Neighbors---
Accuracy: 0.62
```

```
Precision: 0.69
Recall: 0.62
F1 Score: 0.62
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 1 0]]
----------------------------------------
---Decision Tree---
Accuracy: 0.98
Precision: 0.98
Recall: 0.98
F1 Score: 0.97
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 3]]
----------------------------------------
---Random Forest---
Accuracy: 0.99
Precision: 0.99
Recall: 0.99
F1 Score: 0.99
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 4 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 3]]
----------------------------------------
---KMeans Clustering---
Accuracy: 0.01
Precision: 0.00
Recall: 0.01
F1 Score: 0.00
Confusion Matrix:
[[1 0 0 ... 0 0 0]
 [0 2 0 ... 0 0 0]
 [2 2 0 ... 0 0 0]
```

```
 ...
 [4 0 0 ... 0 0 0]
 [1 0 0 ... 0 0 0]
 [3 0 0 ... 0 0 0]]
----------------------------------------
```

Compare the performance of the entire classification algorithm (in table format having recall, precision, F1-score, accuracy))

```
model_performance = {
    "Model": [],
    "Accuracy": [],
    "Precision": [],
    "Recall": [],
    "F1-Score": []
}

def evaluate_and_store_metrics(y_true, y_pred, model_name):
    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred, average='weighted')
    recall = recall_score(y_true, y_pred, average='weighted')
    f1 = f1_score(y_true, y_pred, average='weighted')


    model_performance["Model"].append(model_name)
    model_performance["Accuracy"].append(accuracy)
    model_performance["Precision"].append(precision)
    model_performance["Recall"].append(recall)
    model_performance["F1-Score"].append(f1)


# 1. Logistic Regression
y_pred_lr = log_reg.predict(X_test)
evaluate_and_store_metrics(y_test, y_pred_lr, "Logistic Regression")

# 2. Naive Bayes
y_pred_nb = nb_model.predict(X_test)
evaluate_and_store_metrics(y_test, y_pred_nb, "Naive Bayes")

# 3. K-Nearest Neighbors (KNN)
y_pred_knn = knn.predict(X_test)
evaluate_and_store_metrics(y_test, y_pred_knn, "K-Nearest Neighbors")

# 4. Decision Tree
y_pred_dt = dt_model.predict(X_test)
```

```
evaluate_and_store_metrics(y_test, y_pred_dt, "Decision Tree")

# 5. Random Forest
y_pred_rf = rf_model.predict(X_test)
evaluate_and_store_metrics(y_test, y_pred_rf, "Random Forest")

# 6. KMeans (Clustering)
y_pred_kmeans = np.where(kmeans.predict(X_test) == 0, 1, 0)
evaluate_and_store_metrics(y_test, y_pred_kmeans, "KMeans Clustering")

performance_df = pd.DataFrame(model_performance)
performance_df
```

	Model	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression	0.755102	0.815873	0.755102	0.755959
1	Naive Bayes	0.988095	0.986071	0.988095	0.985733
2	K-Nearest Neighbors	0.615646	0.690063	0.615646	0.619155
3	Decision Tree	0.976190	0.978225	0.976190	0.974226
4	Random Forest	0.994898	0.992687	0.994898	0.993445
5	KMeans Clustering	0.005102	0.000027	0.005102	0.000054

Conclusion of this Mini Project :

In this project, we applied various classification algorithms and evaluated their performance using accuracy, precision, recall, and F1-score. Random Forest performed best due to its ensemble nature, while simpler models like Logistic Regression and Naive Bayes were faster but less accurate. The choice of the best model depends on the dataset's complexity and the need for either performance or interpretability.

```
```