

1 KUIS 1 PEMBELAJARAN MESIN

Nama : Elis Nurhidayati

NIM : 2241720035

Kelas / No.Urut : TI-3C / 08

Link Google Colab:

https://drive.google.com/file/d/1ud_C3vEgmmjinr1L5lWdAOTZS7xKYsyT/view?usp=shari

2 Intro

Pada kuis ini, Anda diminta untuk melakukan klasifikasi citra wajah dengan menggunakan dataset Labeled Face in Wild (LFW), Link dataset: <https://www.kaggle.com/datasets/jessicali9530/lfw-dataset>.

Spesifikasi pengerjaan kuis yang harus dipehuni adalah,

1. Setiap label dari dataset LFW harus berisikan minimal 10 gambar (5 poin)
2. Citra yang digunakan merupakan citra RGB / color (5 poin)
3. (Opsional) Lakukan proses pra pengolahan data jika diperlukan (ekstra 10 poin)
4. (Wajib) Gunakan fitur histogram untuk setiap channel citra (35 poin)
5. (Wajib) Gunakan algoritma SVM sebagai model dasar (35 poin)
6. (Wajib) Gunakan metric akurasi, dan *classification report* untuk melihat performa model (20 poin)

2.1 Challenge

Setelah menyelesaikan kuis berdasarkan spesifikasi wajib, Anda diperkenankan untuk

- Menggunakan fitur lain selain histogram (ekstra 20 poin)
- Menggunakan model lain selain SVM (ekstra 10 poin)
- Evaluasi dengan metric lain ataupun report dengan *confusion matrix* (ekstra 10 poin)

3 Boilerplate

Berikut merupakan boilerplate code yang dapat Anda gunakan sebagai acuan dasar pengerjaan kuis. Anda diperkenankan untuk **menambah** ataupun **mengurangi** bagian boilerplate yang disediakan.

4 Preparation

```
[ ]: # Load required library
# Import Required Library
# Import Required Library
!pip install opencv-python -q
!pip install scikit-learn -q
!pip install matplotlib -q
!pip install seaborn -q

import numpy as np
import pandas as pd
from sklearn.datasets import fetch_lfw_people
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, accuracy_score, \
    classification_report
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
```

5 Load Data

```
[ ]: # Load Dataset
# We will load dataset only for faces with contain min 10 images
# We also want to keep images in color; color=True
faces = fetch_lfw_people(min_faces_per_person=10, color=True) # No. 1 & 2

# Check Faces
print(f'Total Label: {len(faces.target_names)}')
print(f'Data dimension: {faces.images.shape}') # No. 2
print(f'Images flatten dimension: {faces.data.shape}')
```

Total Label: 158

Data dimension: (4324, 62, 47, 3)

Images flatten dimension: (4324, 8742)

```
[ ]: # Check by Plotted Images
# Generate random 15 images
rand_labels = [np.random.randint(0, len(faces.data)) for x in range(0,15)]

fig, ax = plt.subplots(3, 5)
fig.tight_layout()
for i, axi in enumerate(ax.flat):
```

```
axi.imshow(faces.images[rand_labels[i]], cmap='bone')
axi.set(xticks=[], yticks=[],
        xlabel=faces.target_names[faces.target[rand_labels[i]]])
```



6 Preprocessing

```
[ ]: """
    Pada bagian ini Anda diperbolehkan untuk melakukan proses pra pengolahan_
    ↳ data (preprocessing) sesuai dengan kebutuhan. Pra pengolahan data dapat_
    ↳ berupa,

    1. Standardisasi nilai fitur ataupun label
    2. Penyesuaian ukuran gambar
    3. Perubahan colorspace gambar
    4. dsb
    """
# Preprocessing: Resize images to 64x64 for consistency and normalize pixel_
↳ values
processed_images = []
```

```
# No. 3
for img in faces.images:
    # Resize image to (64, 64)
    resized_img = cv2.resize(img, (64, 64))
    # Normalize pixel values (0 to 1)
    normalized_img = resized_img / 255.0
    processed_images.append(normalized_img)

processed_images = np.array(processed_images)
```

7 Features Extraction

```
[ ]: """
    Pada bagian ini, lakukan proses ekstraksi fitur yang diminta, yaitu_
    histogram tiap channel.
    Terdapat banyak cara untuk melakukan ekstraksi fitur histogram.

    Hints:
    Salah satu cara untuk melakukan ekstraksi fitur histogram adalah,
    Anda dapat menggunakan fungsi np.histogram dari library numpy
    """
# No. 4
def extract_histogram_features(images):
    features = []
    for img in images:
        # Calculate histograms for each channel (R, G, B)
        r_hist = np.histogram(img[:, :, 0], bins=256, range=(0, 1))[0]
        g_hist = np.histogram(img[:, :, 1], bins=256, range=(0, 1))[0]
        b_hist = np.histogram(img[:, :, 2], bins=256, range=(0, 1))[0]
        # Concatenate histograms to form feature vector
        hist_features = np.concatenate((r_hist, g_hist, b_hist))
        features.append(hist_features)
    return np.array(features)

# Extract features from processed images
X = extract_histogram_features(processed_images)
y = faces.target
```

8 Build Model

```
[ ]: """
    Pada bagian ini lakukan proses pembuatan model,

    1. Pembuatan data training dan data testing
```

```

    2. Pembuatan objek model
    3. Proses fitting model
'''
# No. 5
# Split data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    ↳ random_state=42)

# Build SVM model
svm_model = SVC(kernel='linear', random_state=42)
# Fit model to training data
svm_model.fit(X_train, y_train)

```

```
[ ]: SVC(kernel='linear', random_state=42)
```

9 Evaluation

```

[ ]: '''
    Pada bagian ini, lakukan evaluasi terhadap data training dan data testing
    dengan menggunakan metric akurasi dan juga classfication report

    Hints:
    Anda dapat menggunakan referensi Jobseet 4 Percobaan 5
'''
# No. 6
from sklearn.metrics import classification_report

# Train evaluation
y_train_pred = svm_model.predict(X_train) # Predict using the trained model

print('Train Classification Report:')
print(classification_report(y_train, y_train_pred, zero_division=0))
print('Train Accuracy:', accuracy_score(y_train, y_train_pred)) # Tambahan_
    ↳ evaluasi akurasi untuk data training
print('.....')
# Test evaluation
y_test_pred = svm_model.predict(X_test) # Predict using the trained model

print('Test Classification Report:')
print(classification_report(y_test, y_test_pred, zero_division=0))
print('Test Accuracy:', accuracy_score(y_test, y_test_pred)) # Tambahan_
    ↳ evaluasi akurasi untuk data testing

```

Train Classification Report:

precision	recall	f1-score	support
-----------	--------	----------	---------

0	0.00	0.00	0.00	14
1	0.00	0.00	0.00	11
2	0.00	0.00	0.00	37
3	0.17	0.04	0.06	28
4	0.00	0.00	0.00	15
5	0.00	0.00	0.00	34
6	0.50	0.09	0.15	11
7	0.00	0.00	0.00	18
8	0.00	0.00	0.00	10
9	0.80	0.36	0.50	11
10	0.00	0.00	0.00	9
11	0.00	0.00	0.00	62
12	0.00	0.00	0.00	27
13	0.00	0.00	0.00	22
14	0.00	0.00	0.00	26
15	0.00	0.00	0.00	13
16	0.00	0.00	0.00	7
17	0.00	0.00	0.00	14
18	0.00	0.00	0.00	11
19	0.00	0.00	0.00	18
20	0.00	0.00	0.00	17
21	0.00	0.00	0.00	10
22	0.00	0.00	0.00	9
23	0.00	0.00	0.00	192
24	0.00	0.00	0.00	9
25	0.00	0.00	0.00	28
26	0.00	0.00	0.00	12
27	0.00	0.00	0.00	11
28	0.00	0.00	0.00	13
29	0.00	0.00	0.00	101
30	0.00	0.00	0.00	7
31	0.00	0.00	0.00	14
32	0.00	0.00	0.00	12
33	0.50	0.09	0.15	11
34	0.00	0.00	0.00	17
35	0.13	0.97	0.22	431
36	0.00	0.00	0.00	89
37	0.00	0.00	0.00	38
38	0.00	0.00	0.00	6
39	0.00	0.00	0.00	9
40	0.00	0.00	0.00	23
41	0.00	0.00	0.00	22
42	0.00	0.00	0.00	13
43	0.00	0.00	0.00	17
44	0.00	0.00	0.00	33
45	0.00	0.00	0.00	10
46	0.00	0.00	0.00	11
47	0.00	0.00	0.00	9

48	0.00	0.00	0.00	11
49	0.00	0.00	0.00	53
50	0.00	0.00	0.00	9
51	0.00	0.00	0.00	16
52	0.00	0.00	0.00	23
53	0.00	0.00	0.00	11
54	0.00	0.00	0.00	46
55	0.00	0.00	0.00	7
56	0.00	0.00	0.00	9
57	0.00	0.00	0.00	10
58	0.00	0.00	0.00	7
59	0.00	0.00	0.00	9
60	0.00	0.00	0.00	10
61	0.00	0.00	0.00	44
62	0.00	0.00	0.00	8
63	0.00	0.00	0.00	10
64	0.00	0.00	0.00	14
65	0.08	0.14	0.10	36
66	0.00	0.00	0.00	7
67	0.00	0.00	0.00	16
68	0.00	0.00	0.00	17
69	0.00	0.00	0.00	16
70	0.00	0.00	0.00	10
71	0.00	0.00	0.00	9
72	0.00	0.00	0.00	8
73	0.00	0.00	0.00	45
74	0.00	0.00	0.00	14
75	0.00	0.00	0.00	9
76	0.00	0.00	0.00	11
77	0.00	0.00	0.00	26
78	0.00	0.00	0.00	6
79	0.00	0.00	0.00	14
80	0.00	0.00	0.00	15
81	0.00	0.00	0.00	19
82	0.00	0.00	0.00	23
83	0.00	0.00	0.00	13
84	0.00	0.00	0.00	12
85	0.00	0.00	0.00	43
86	0.00	0.00	0.00	8
87	0.00	0.00	0.00	13
88	0.00	0.00	0.00	7
89	0.00	0.00	0.00	27
90	0.00	0.00	0.00	16
91	0.00	0.00	0.00	36
92	0.00	0.00	0.00	18
93	0.00	0.00	0.00	31
94	0.00	0.00	0.00	9
95	0.00	0.00	0.00	40

96	0.00	0.00	0.00	10
97	0.00	0.00	0.00	23
98	0.00	0.00	0.00	9
99	0.00	0.00	0.00	26
100	0.00	0.00	0.00	12
101	0.00	0.00	0.00	16
102	0.00	0.00	0.00	10
103	0.00	0.00	0.00	15
104	0.00	0.00	0.00	8
105	0.00	0.00	0.00	8
106	0.00	0.00	0.00	10
107	0.00	0.00	0.00	7
108	0.00	0.00	0.00	10
109	0.00	0.00	0.00	19
110	0.00	0.00	0.00	31
111	0.00	0.00	0.00	8
112	1.00	0.07	0.13	14
113	0.00	0.00	0.00	10
114	0.00	0.00	0.00	9
115	0.00	0.00	0.00	14
116	0.00	0.00	0.00	9
117	0.00	0.00	0.00	9
118	0.00	0.00	0.00	15
119	0.00	0.00	0.00	19
120	0.00	0.00	0.00	14
121	0.00	0.00	0.00	11
122	0.00	0.00	0.00	25
123	0.00	0.00	0.00	15
124	0.00	0.00	0.00	22
125	0.00	0.00	0.00	7
126	0.00	0.00	0.00	7
127	0.00	0.00	0.00	11
128	0.50	0.20	0.29	10
129	0.00	0.00	0.00	23
130	0.00	0.00	0.00	10
131	0.00	0.00	0.00	22
132	0.00	0.00	0.00	18
133	0.00	0.00	0.00	9
134	0.00	0.00	0.00	40
135	0.00	0.00	0.00	10
136	0.00	0.00	0.00	8
137	0.00	0.00	0.00	29
138	0.00	0.00	0.00	12
139	0.00	0.00	0.00	12
140	0.00	0.00	0.00	9
141	0.00	0.00	0.00	17
142	0.00	0.00	0.00	15
143	0.00	0.00	0.00	9

144	0.00	0.00	0.00	20
145	0.00	0.00	0.00	9
146	0.00	0.00	0.00	26
147	0.00	0.00	0.00	8
148	0.00	0.00	0.00	8
149	0.16	0.05	0.08	112
150	0.00	0.00	0.00	13
151	0.00	0.00	0.00	12
152	0.00	0.00	0.00	25
153	0.00	0.00	0.00	44
154	0.00	0.00	0.00	8
155	0.00	0.00	0.00	9
156	0.00	0.00	0.00	23
157	0.00	0.00	0.00	13
accuracy			0.13	3459
macro avg	0.02	0.01	0.01	3459
weighted avg	0.03	0.13	0.04	3459

Train Accuracy: 0.12691529343740965

Test Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	5
1	0.00	0.00	0.00	1
2	0.00	0.00	0.00	2
3	0.00	0.00	0.00	7
4	0.00	0.00	0.00	6
5	0.00	0.00	0.00	2
6	1.00	0.25	0.40	4
7	0.00	0.00	0.00	2
8	0.00	0.00	0.00	1
9	0.00	0.00	0.00	1
10	0.00	0.00	0.00	4
11	0.00	0.00	0.00	15
12	0.00	0.00	0.00	15
13	0.00	0.00	0.00	2
14	0.00	0.00	0.00	3
15	0.00	0.00	0.00	4
16	0.00	0.00	0.00	3
17	0.00	0.00	0.00	1
18	0.00	0.00	0.00	3
19	0.00	0.00	0.00	3
20	0.00	0.00	0.00	2
21	0.00	0.00	0.00	1
22	0.00	0.00	0.00	4
23	0.00	0.00	0.00	44

24	0.00	0.00	0.00	2
25	0.00	0.00	0.00	3
26	0.00	0.00	0.00	2
27	0.00	0.00	0.00	3
28	0.00	0.00	0.00	2
29	0.00	0.00	0.00	20
30	0.00	0.00	0.00	6
32	0.00	0.00	0.00	6
33	0.00	0.00	0.00	2
34	0.00	0.00	0.00	5
35	0.11	0.95	0.20	99
36	0.00	0.00	0.00	20
37	0.00	0.00	0.00	6
38	0.00	0.00	0.00	6
39	0.00	0.00	0.00	4
40	0.00	0.00	0.00	3
41	0.00	0.00	0.00	8
42	0.00	0.00	0.00	3
43	0.00	0.00	0.00	5
44	0.00	0.00	0.00	6
45	0.00	0.00	0.00	2
46	0.00	0.00	0.00	3
47	0.00	0.00	0.00	3
48	0.00	0.00	0.00	4
49	0.00	0.00	0.00	18
50	0.00	0.00	0.00	1
51	0.00	0.00	0.00	4
52	0.00	0.00	0.00	5
53	0.00	0.00	0.00	2
54	0.00	0.00	0.00	6
55	0.00	0.00	0.00	3
56	0.00	0.00	0.00	5
57	0.00	0.00	0.00	1
58	0.00	0.00	0.00	3
59	0.00	0.00	0.00	1
60	0.00	0.00	0.00	7
61	0.00	0.00	0.00	11
62	0.00	0.00	0.00	2
63	0.00	0.00	0.00	2
64	0.00	0.00	0.00	7
65	0.00	0.00	0.00	6
66	0.00	0.00	0.00	5
67	0.00	0.00	0.00	5
68	0.00	0.00	0.00	7
69	0.00	0.00	0.00	4
70	0.00	0.00	0.00	1
71	0.00	0.00	0.00	4
72	0.00	0.00	0.00	3

73	0.00	0.00	0.00	8
74	0.00	0.00	0.00	3
75	0.00	0.00	0.00	10
76	0.00	0.00	0.00	6
77	0.00	0.00	0.00	5
78	0.00	0.00	0.00	5
79	0.00	0.00	0.00	3
80	0.00	0.00	0.00	4
81	0.00	0.00	0.00	4
82	0.00	0.00	0.00	5
83	0.00	0.00	0.00	6
84	0.00	0.00	0.00	3
85	0.00	0.00	0.00	17
86	0.00	0.00	0.00	4
87	0.00	0.00	0.00	1
88	0.00	0.00	0.00	4
89	0.00	0.00	0.00	5
90	0.00	0.00	0.00	2
91	0.00	0.00	0.00	5
92	0.00	0.00	0.00	4
93	0.00	0.00	0.00	10
94	0.00	0.00	0.00	4
95	0.00	0.00	0.00	8
96	0.00	0.00	0.00	4
97	0.00	0.00	0.00	6
98	0.00	0.00	0.00	2
99	0.00	0.00	0.00	7
100	0.00	0.00	0.00	3
101	0.00	0.00	0.00	4
102	0.00	0.00	0.00	2
103	0.00	0.00	0.00	3
104	0.00	0.00	0.00	3
105	0.00	0.00	0.00	2
106	0.00	0.00	0.00	5
107	0.00	0.00	0.00	3
108	0.00	0.00	0.00	5
109	0.00	0.00	0.00	3
110	0.00	0.00	0.00	6
111	0.00	0.00	0.00	3
112	0.00	0.00	0.00	5
113	0.00	0.00	0.00	5
114	0.00	0.00	0.00	1
115	0.00	0.00	0.00	6
116	0.00	0.00	0.00	2
117	0.00	0.00	0.00	1
118	0.00	0.00	0.00	3
119	0.00	0.00	0.00	3
120	0.00	0.00	0.00	1

121	0.00	0.00	0.00	2
122	0.00	0.00	0.00	5
123	0.00	0.00	0.00	2
124	0.00	0.00	0.00	5
125	0.00	0.00	0.00	4
126	0.00	0.00	0.00	3
127	0.00	0.00	0.00	7
128	1.00	0.25	0.40	4
129	0.00	0.00	0.00	9
130	0.00	0.00	0.00	2
131	0.00	0.00	0.00	4
132	0.00	0.00	0.00	5
133	0.00	0.00	0.00	4
134	0.00	0.00	0.00	12
135	0.00	0.00	0.00	1
136	0.00	0.00	0.00	3
137	0.00	0.00	0.00	4
138	0.00	0.00	0.00	5
139	0.00	0.00	0.00	3
140	0.00	0.00	0.00	2
141	0.00	0.00	0.00	6
142	0.00	0.00	0.00	4
143	0.00	0.00	0.00	1
144	0.00	0.00	0.00	5
145	0.00	0.00	0.00	1
146	0.00	0.00	0.00	7
147	0.00	0.00	0.00	8
148	0.00	0.00	0.00	2
149	0.00	0.00	0.00	32
150	0.00	0.00	0.00	3
151	0.00	0.00	0.00	5
152	0.00	0.00	0.00	7
153	0.00	0.00	0.00	5
154	0.00	0.00	0.00	2
155	0.00	0.00	0.00	4
156	0.00	0.00	0.00	1
157	0.00	0.00	0.00	1
accuracy			0.11	865
macro avg	0.01	0.01	0.01	865
weighted avg	0.02	0.11	0.03	865

Test Accuracy: 0.11098265895953757

10 Challenge

```
[ ]: from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import classification_report, accuracy_score, \
        confusion_matrix
    from skimage.feature import hog

    # 1. Menggunakan fitur lain selain histogram
    # Feature Extraction: Extract HOG features
    def extract_hog_features(images):
        hog_features = []
        for img in images:
            gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            features = hog(gray_img, pixels_per_cell=(8, 8), cells_per_block=(2, \
                ↪2), visualize=False)
            hog_features.append(features)
        return np.array(hog_features)

    X = extract_histogram_features(processed_images)
    X_hog = extract_hog_features(processed_images)

    # Mengkombinasikan fitur histogram and HOG
    X_combined = np.concatenate((X, X_hog), axis=1)
    y = faces.target

    # 2. Model selain SVM
    # 3. Report dengan Confusion Matrix
    # Train evaluation with Random Forest
    rf_model = RandomForestClassifier() # Inisialisasi model
    rf_model.fit(X_train, y_train)
    y_train_pred_rf = rf_model.predict(X_train)
    print("Train Classification Report (Random Forest):")
    print(classification_report(y_train, y_train_pred_rf, zero_division=0))
    print("Train Accuracy (Random Forest):", accuracy_score(y_train, \
        ↪y_train_pred_rf))
    print(".....")

    # Test evaluation with Random Forest
    y_test_pred_rf = rf_model.predict(X_test)
    print("Random Forest Test Classification Report:")
    print(classification_report(y_test, y_test_pred_rf, zero_division=0))
    print("Random Forest Test Accuracy:", accuracy_score(y_test, y_test_pred_rf))

    # Confusion Matrix for Random Forest
    cm = confusion_matrix(y_test, y_test_pred_rf)
    plt.figure(figsize=(10, 7))
```

```

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=faces_
    ↳target_names, yticklabels=faces.target_names) # Ensure faces.target_names_
    ↳is defined
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion Matrix')
plt.show()

```

Train Classification Report (Random Forest):

	precision	recall	f1-score	support
0	1.00	0.07	0.13	14
1	1.00	0.09	0.17	11
2	0.20	0.03	0.05	37
3	1.00	0.25	0.40	28
4	0.00	0.00	0.00	15
5	0.83	0.15	0.25	34
6	1.00	0.27	0.43	11
7	1.00	0.11	0.20	18
8	0.00	0.00	0.00	10
9	1.00	0.45	0.62	11
10	0.00	0.00	0.00	9
11	1.00	0.10	0.18	62
12	0.00	0.00	0.00	27
13	0.67	0.09	0.16	22
14	0.40	0.08	0.13	26
15	1.00	0.08	0.14	13
16	0.00	0.00	0.00	7
17	0.50	0.07	0.12	14
18	0.00	0.00	0.00	11
19	1.00	0.11	0.20	18
20	1.00	0.06	0.11	17
21	0.20	0.10	0.13	10
22	0.00	0.00	0.00	9
23	0.40	0.14	0.21	192
24	0.00	0.00	0.00	9
25	0.00	0.00	0.00	28
26	1.00	0.17	0.29	12
27	0.62	0.45	0.53	11
28	1.00	0.08	0.14	13
29	1.00	0.06	0.11	101
30	1.00	0.29	0.44	7
31	0.75	0.21	0.33	14
32	1.00	0.08	0.15	12
33	1.00	0.36	0.53	11
34	0.25	0.06	0.10	17
35	0.14	0.95	0.25	431

36	0.39	0.17	0.24	89
37	1.00	0.08	0.15	38
38	0.00	0.00	0.00	6
39	0.00	0.00	0.00	9
40	0.40	0.09	0.14	23
41	0.50	0.18	0.27	22
42	0.00	0.00	0.00	13
43	1.00	0.06	0.11	17
44	0.53	0.24	0.33	33
45	0.00	0.00	0.00	10
46	1.00	0.09	0.17	11
47	1.00	0.22	0.36	9
48	0.00	0.00	0.00	11
49	0.42	0.09	0.15	53
50	0.00	0.00	0.00	9
51	0.50	0.06	0.11	16
52	1.00	0.17	0.30	23
53	1.00	0.09	0.17	11
54	0.33	0.17	0.23	46
55	1.00	0.14	0.25	7
56	1.00	0.22	0.36	9
57	1.00	0.20	0.33	10
58	1.00	0.14	0.25	7
59	0.33	0.11	0.17	9
60	0.00	0.00	0.00	10
61	0.28	0.16	0.20	44
62	0.00	0.00	0.00	8
63	1.00	0.20	0.33	10
64	0.50	0.07	0.12	14
65	0.88	0.19	0.32	36
66	0.00	0.00	0.00	7
67	0.00	0.00	0.00	16
68	0.33	0.06	0.10	17
69	1.00	0.06	0.12	16
70	0.00	0.00	0.00	10
71	1.00	0.11	0.20	9
72	0.00	0.00	0.00	8
73	1.00	0.04	0.09	45
74	0.00	0.00	0.00	14
75	0.00	0.00	0.00	9
76	0.00	0.00	0.00	11
77	0.50	0.04	0.07	26
78	1.00	0.17	0.29	6
79	0.50	0.14	0.22	14
80	1.00	0.07	0.12	15
81	1.00	0.05	0.10	19
82	1.00	0.17	0.30	23
83	1.00	0.08	0.14	13

84	0.00	0.00	0.00	12
85	0.00	0.00	0.00	43
86	0.00	0.00	0.00	8
87	1.00	0.08	0.14	13
88	0.00	0.00	0.00	7
89	1.00	0.11	0.20	27
90	0.00	0.00	0.00	16
91	0.75	0.08	0.15	36
92	0.00	0.00	0.00	18
93	1.00	0.06	0.12	31
94	0.67	0.22	0.33	9
95	0.45	0.12	0.20	40
96	0.67	0.20	0.31	10
97	1.00	0.04	0.08	23
98	0.00	0.00	0.00	9
99	1.00	0.04	0.07	26
100	0.00	0.00	0.00	12
101	0.67	0.12	0.21	16
102	1.00	0.10	0.18	10
103	0.00	0.00	0.00	15
104	1.00	0.12	0.22	8
105	1.00	0.12	0.22	8
106	0.20	0.30	0.24	10
107	0.00	0.00	0.00	7
108	1.00	0.20	0.33	10
109	1.00	0.05	0.10	19
110	1.00	0.03	0.06	31
111	0.25	0.12	0.17	8
112	0.33	0.36	0.34	14
113	0.50	0.10	0.17	10
114	0.00	0.00	0.00	9
115	0.50	0.07	0.12	14
116	1.00	0.44	0.62	9
117	0.00	0.00	0.00	9
118	0.00	0.00	0.00	15
119	1.00	0.21	0.35	19
120	1.00	0.29	0.44	14
121	0.00	0.00	0.00	11
122	0.83	0.20	0.32	25
123	0.00	0.00	0.00	15
124	1.00	0.09	0.17	22
125	0.00	0.00	0.00	7
126	1.00	0.14	0.25	7
127	1.00	0.09	0.17	11
128	1.00	0.40	0.57	10
129	1.00	0.17	0.30	23
130	1.00	0.10	0.18	10
131	1.00	0.05	0.09	22

132	1.00	0.17	0.29	18
133	0.12	0.11	0.12	9
134	1.00	0.05	0.10	40
135	1.00	0.10	0.18	10
136	1.00	0.12	0.22	8
137	0.67	0.07	0.12	29
138	0.00	0.00	0.00	12
139	1.00	0.08	0.15	12
140	1.00	0.11	0.20	9
141	1.00	0.06	0.11	17
142	0.00	0.00	0.00	15
143	0.20	0.11	0.14	9
144	1.00	0.20	0.33	20
145	0.00	0.00	0.00	9
146	0.33	0.04	0.07	26
147	0.00	0.00	0.00	8
148	0.00	0.00	0.00	8
149	0.72	0.16	0.26	112
150	0.00	0.00	0.00	13
151	0.00	0.00	0.00	12
152	1.00	0.12	0.21	25
153	0.92	0.25	0.39	44
154	1.00	0.12	0.22	8
155	0.00	0.00	0.00	9
156	0.12	0.13	0.12	23
157	1.00	0.15	0.27	13
accuracy				0.21 3459
macro avg		0.55 0.10	0.15	3459
weighted avg		0.54 0.21	0.18	3459

Train Accuracy (Random Forest): 0.2101763515466898

Random Forest Test Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	5
1	0.00	0.00	0.00	1
2	0.00	0.00	0.00	2
3	0.00	0.00	0.00	7
4	0.00	0.00	0.00	6
5	0.00	0.00	0.00	2
6	0.00	0.00	0.00	4
7	0.00	0.00	0.00	2
8	0.00	0.00	0.00	1
9	0.00	0.00	0.00	1
10	0.00	0.00	0.00	4
11	0.00	0.00	0.00	15

12	0.00	0.00	0.00	15
13	0.00	0.00	0.00	2
14	0.00	0.00	0.00	3
15	0.00	0.00	0.00	4
16	0.00	0.00	0.00	3
17	0.00	0.00	0.00	1
18	0.00	0.00	0.00	3
19	0.00	0.00	0.00	3
20	0.00	0.00	0.00	2
21	0.00	0.00	0.00	1
22	0.00	0.00	0.00	4
23	0.00	0.00	0.00	44
24	0.00	0.00	0.00	2
25	0.00	0.00	0.00	3
26	0.00	0.00	0.00	2
27	0.00	0.00	0.00	3
28	0.00	0.00	0.00	2
29	0.00	0.00	0.00	20
30	0.00	0.00	0.00	6
31	0.00	0.00	0.00	0
32	0.00	0.00	0.00	6
33	0.00	0.00	0.00	2
34	0.00	0.00	0.00	5
35	0.12	0.88	0.21	99
36	0.09	0.05	0.06	20
37	0.00	0.00	0.00	6
38	0.00	0.00	0.00	6
39	0.00	0.00	0.00	4
40	0.00	0.00	0.00	3
41	0.00	0.00	0.00	8
42	0.00	0.00	0.00	3
43	0.00	0.00	0.00	5
44	0.00	0.00	0.00	6
45	0.00	0.00	0.00	2
46	0.00	0.00	0.00	3
47	0.00	0.00	0.00	3
48	0.00	0.00	0.00	4
49	0.00	0.00	0.00	18
50	0.00	0.00	0.00	1
51	0.00	0.00	0.00	4
52	0.00	0.00	0.00	5
53	0.00	0.00	0.00	2
54	0.00	0.00	0.00	6
55	0.00	0.00	0.00	3
56	0.00	0.00	0.00	5
57	0.00	0.00	0.00	1
58	0.00	0.00	0.00	3
59	0.00	0.00	0.00	1

60	0.00	0.00	0.00	7
61	0.00	0.00	0.00	11
62	0.00	0.00	0.00	2
63	0.00	0.00	0.00	2
64	0.00	0.00	0.00	7
65	0.00	0.00	0.00	6
66	0.00	0.00	0.00	5
67	0.00	0.00	0.00	5
68	0.00	0.00	0.00	7
69	0.00	0.00	0.00	4
70	0.00	0.00	0.00	1
71	0.00	0.00	0.00	4
72	0.00	0.00	0.00	3
73	0.00	0.00	0.00	8
74	0.00	0.00	0.00	3
75	0.00	0.00	0.00	10
76	0.00	0.00	0.00	6
77	0.00	0.00	0.00	5
78	0.00	0.00	0.00	5
79	0.00	0.00	0.00	3
80	0.00	0.00	0.00	4
81	0.00	0.00	0.00	4
82	0.00	0.00	0.00	5
83	0.00	0.00	0.00	6
84	0.00	0.00	0.00	3
85	0.00	0.00	0.00	17
86	0.00	0.00	0.00	4
87	0.00	0.00	0.00	1
88	0.00	0.00	0.00	4
89	0.00	0.00	0.00	5
90	0.00	0.00	0.00	2
91	0.00	0.00	0.00	5
92	0.00	0.00	0.00	4
93	0.00	0.00	0.00	10
94	0.00	0.00	0.00	4
95	0.00	0.00	0.00	8
96	0.00	0.00	0.00	4
97	0.00	0.00	0.00	6
98	0.00	0.00	0.00	2
99	0.00	0.00	0.00	7
100	0.00	0.00	0.00	3
101	0.00	0.00	0.00	4
102	0.00	0.00	0.00	2
103	0.00	0.00	0.00	3
104	0.00	0.00	0.00	3
105	0.00	0.00	0.00	2
106	0.00	0.00	0.00	5
107	0.00	0.00	0.00	3

108	0.00	0.00	0.00	5
109	0.00	0.00	0.00	3
110	0.00	0.00	0.00	6
111	0.00	0.00	0.00	3
112	0.00	0.00	0.00	5
113	0.00	0.00	0.00	5
114	0.00	0.00	0.00	1
115	0.00	0.00	0.00	6
116	0.00	0.00	0.00	2
117	0.00	0.00	0.00	1
118	0.00	0.00	0.00	3
119	0.00	0.00	0.00	3
120	0.00	0.00	0.00	1
121	0.00	0.00	0.00	2
122	0.00	0.00	0.00	5
123	0.00	0.00	0.00	2
124	0.00	0.00	0.00	5
125	0.00	0.00	0.00	4
126	0.00	0.00	0.00	3
127	0.00	0.00	0.00	7
128	0.33	0.25	0.29	4
129	0.00	0.00	0.00	9
130	0.00	0.00	0.00	2
131	0.00	0.00	0.00	4
132	0.00	0.00	0.00	5
133	0.00	0.00	0.00	4
134	0.00	0.00	0.00	12
135	0.00	0.00	0.00	1
136	0.00	0.00	0.00	3
137	0.00	0.00	0.00	4
138	0.00	0.00	0.00	5
139	0.00	0.00	0.00	3
140	0.00	0.00	0.00	2
141	0.00	0.00	0.00	6
142	0.00	0.00	0.00	4
143	0.00	0.00	0.00	1
144	0.00	0.00	0.00	5
145	0.00	0.00	0.00	1
146	0.00	0.00	0.00	7
147	0.00	0.00	0.00	8
148	0.00	0.00	0.00	2
149	0.22	0.06	0.10	32
150	0.00	0.00	0.00	3
151	0.00	0.00	0.00	5
152	0.00	0.00	0.00	7
153	0.00	0.00	0.00	5
154	0.00	0.00	0.00	2
155	0.00	0.00	0.00	4

Random Forest Test Accuracy: 0.10520231213872833

