MINI PROJECT 1 PSO

- Nama : Ricke Gita Arum Arifiani
- Nim : 1102204526
- Kelas: S2TE-BP1

```
import pandas as pd
from sklearn.model_selection import train_test_split, KFold, cross_val_score
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score, confusion_matrix
from tabulate import tabulate
```

```
dataset_cpa1 = pd.read_excel('CPA1.xlsx')
dataset_cpa2 = pd.read_excel('CPA2.xlsx')
dataset_cpa3 = pd.read_excel('CPA3.xlsx')
dataset_cpb1 = pd.read_excel('CPB1.xlsx')
dataset_cpb2 = pd.read_excel('CPB2.xlsx')
dataset_cpb3 = pd.read_excel('CPB3.xlsx')
dataset_cpb4 = pd.read_excel('CPB4.xlsx')
dataset_cpb5 = pd.read_excel('CPB5.xlsx')
dataset_cpb6 = pd.read_excel('CPB6.xlsx')
dataset_cpb7 = pd.read_excel('CPB7.xlsx')
```

```
def evaluate_model(X, y, classifier, cv=10):
    results = []

    # 10-fold Cross Validation
    kf = KFold(n_splits=cv, shuffle=True, random_state=42)
    scores = cross_val_score(classifier, X, y, cv=kf, scoring='f1_macro')
    y_pred = classifier.fit(X, y).predict(X)
    tn, fp, fn, tp = confusion_matrix(y, y_pred).ravel()
    sensitivity = tp / (tp + fn)
    specificity = tn / (tn + fp)
    accuracy = accuracy_score(y, y_pred)
    results.append(['10-fold Cross Validation',
                    f'{accuracy:.3f}',
                    f'{scores.mean():.3f} (+/- {scores.std():.3f})',
                    f'{sensitivity:.3f}',
                    f'{specificity:.3f}'])

    # 75% training, 25% testing
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)
    results.append(['75% training, 25% testing',
                    f'{accuracy_score(y_test, y_pred):.3f}',
                    f'{f1_score(y_test, y_pred, average="macro"):.3f}',
                    f'{recall_score(y_test, y_pred, average="macro"):.3f}',
                    f'{precision_score(y_test, y_pred, average="macro"):.3f}'])

    # 50% training, 25% validation, 25% testing
    X_train, X_tmp, y_train, y_tmp = train_test_split(X, y, test_size=0.5, random_state=42)
    X_val, X_test, y_val, y_test = train_test_split(X_tmp, y_tmp, test_size=0.5, random_state=42)
    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)
    results.append(['50% training, 25% validation, 25% testing',
                    f'{accuracy_score(y_test, y_pred):.3f}',
                    f'{f1_score(y_test, y_pred, average="macro"):.3f}',
                    f'{recall_score(y_test, y_pred, average="macro"):.3f}',
                    f'{precision_score(y_test, y_pred, average="macro"):.3f}'])

    return tabulate(results, headers=['Evaluation', 'Accuracy', 'F1-Score', 'Sensitivity', 'Specificity'], tablefmt='fancy_grid')
```

```
# Klasifikasi dataset CPA1
print("\nKlasifikasi dataset CPA1")
X_cpa1 = dataset_cpa1[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpa1 = dataset_cpa1['kelas']
print("\nSVM:")
print(evaluate_model(X_cpa1, y_cpa1, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpa1, y_cpa1, KNeighborsClassifier()))


# Klasifikasi dataset CPA2
print("\nKlasifikasi dataset CPA2")
X_cpa2 = dataset_cpa2[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpa2 = dataset_cpa2['kelas']
print("\nSVM:")
print(evaluate_model(X_cpa2, y_cpa2, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpa2, y_cpa2, KNeighborsClassifier()))

# Klasifikasi dataset CPA3
print("\nKlasifikasi dataset CPA3")
X_cpa3 = dataset_cpa3[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpa3 = dataset_cpa3['kelas']
print("\nSVM:")
print(evaluate_model(X_cpa3, y_cpa3, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpa3, y_cpa3, KNeighborsClassifier()))
```

```
Klasifikasi dataset CPA1

SVM:
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and
  _warn_prf(average, modifier, msg_start, len(result))
```

Evaluation	Accuracy	F1-Score	Sensitivity	Specificity
10-fold Cross Validation	0.525	0.351 (+/- 0.097)	0.89	0.16
75% training, 25% testing	0.46	0.315	0.5	0.23
50% training, 25% validation, 25% testing	0.42	0.296	0.5	0.21

```
K-NN:
```

Evaluation	Accuracy	F1-Score	Sensitivity	Specificity
10-fold Cross Validation	0.825	0.732 (+/- 0.093)	0.78	0.87
75% training, 25% testing	0.76	0.758	0.758	0.758
50% training, 25% validation, 25% testing	0.62	0.616	0.62	0.617

```
Klasifikasi dataset CPA2

SVM:
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and
  _warn_prf(average, modifier, msg_start, len(result))
```

Evaluation	Accuracy	F1-Score	Sensitivity	Specificity
10-fold Cross Validation	0.66	0.402 (+/- 0.145)	0.8	0.52
75% training, 25% testing	0.46	0.315	0.5	0.23
50% training, 25% validation, 25% testing	0.42	0.296	0.5	0.21

```
K-NN:
```

Evaluation	Accuracy	F1-Score	Sensitivity	Specificity
10-fold Cross Validation	0.795	0.728 (+/- 0.141)	0.72	0.87
75% training, 25% testing	0.82	0.819	0.82	0.819

50% training, 25% validation, 25% testing	0.66	0.660	0.681	0.683

```
# Klasifikasi dataset CPB1
print("\nKlasifikasi dataset CPB1")
X_cpb1 = dataset_cpb1[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpb1 = dataset_cpb1['kelas']
print("\nSVM:")
print(evaluate_model(X_cpb1, y_cpb1, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpb1, y_cpb1, KNeighborsClassifier()))

# Klasifikasi dataset CPB2
print("\nKlasifikasi dataset CPB2")
X_cpb2 = dataset_cpb2[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpb2 = dataset_cpb2['kelas']
print("\nSVM:")
print(evaluate_model(X_cpb2, y_cpb2, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpb2, y_cpb2, KNeighborsClassifier()))

# Klasifikasi dataset CPB3
print("\nKlasifikasi dataset CPB3")
X_cpb3 = dataset_cpb3[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpb3 = dataset_cpb3['kelas']
print("\nSVM:")
print(evaluate_model(X_cpb3, y_cpb3, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpb3, y_cpb3, KNeighborsClassifier()))

# Klasifikasi dataset CPB4
print("\nKlasifikasi dataset CPB4")
X_cpb4 = dataset_cpb4[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpb4 = dataset_cpb4['kelas']
print("\nSVM:")
print(evaluate_model(X_cpb4, y_cpb4, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpb4, y_cpb4, KNeighborsClassifier()))

# Klasifikasi dataset CPB5
print("\nKlasifikasi dataset CPB5")
X_cpb5 = dataset_cpb5[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpb5 = dataset_cpb5['kelas']
print("\nSVM:")
print(evaluate_model(X_cpb5, y_cpb5, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpb5, y_cpb5, KNeighborsClassifier()))

# Klasifikasi dataset CPB6
print("\nKlasifikasi dataset CPB6")
X_cpb6 = dataset_cpb6[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpb6 = dataset_cpb6['kelas']
print("\nSVM:")
print(evaluate_model(X_cpb6, y_cpb6, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpb6, y_cpb6, KNeighborsClassifier()))

# Klasifikasi dataset CPB7
print("\nKlasifikasi dataset CPB7")
X_cpb7 = dataset_cpb7[['powLv1', 'powLv2', 'powLv3', 'powLv4', 'powLv5']]
y_cpb7 = dataset_cpb7['kelas']
print("\nSVM:")
print(evaluate_model(X_cpb7, y_cpb7, SVC()))
print("\nK-NN:")
print(evaluate_model(X_cpb7, y_cpb7, KNeighborsClassifier()))
```

ore	Sensitivity	Specificity
(+/- 0.159)	0.99	0.28

	0.5	0.23
	0.5	0.21

ication.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero

ication.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero

ore	Sensitivity	Specificity
(+/- 0.051)	0.89	0.89
	0.879	0.879
	0.808	0.8

ore	Sensitivity	Specificity
(+/- 0.157)	1	0.28
	0.5	0.23
	0.5	0.21

ication.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero

ication.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero

ore	Sensitivity	Specificity
(+/- 0.058)	0.93	0.87
	0.882	0.88
	0.821	0.821