Praktikum Data Preprocessing

Praktikum Klasifikasi dengan Studi Kasus Data Titanic

Ulima Inas Shabrina(2110181048)

Latihan 7 - #Assignment

- dataset ← titanic.csv
- train_label ← dataset fitur kelas (Survived)
- train_data ← dataset fitur selain kelas (Survived)
- test_data ← titanic_test.csv
- test_label ← titanic_testlabel.csv

- Lakukan klasifikasi test_data dan hitung error rationya
- Buatlah skenario tertentu pada pemilihan fitur sehingga error ratio dari klasifikasi test_data dapat sekecil mungkin
- Algoritma klasifikasi yang dipakai adalah k-NN atau Decision Tree
- Pembagi error ratio adalah jumlah keseluruhan data pada test_data

Mempersiapkan data

```
import pandas as pd
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier

dataset = pd.read_csv('titanic.csv')
test_dataset = pd.read_csv('titanic_test.csv')
test_label = pd.read_csv('titanic_testlabel.csv')
```

Menampilkan dataset dapat dilakukan dengan menggunakan method read_csv() pada library pandas

```
train_data = pd.DataFrame(dataset, columns = ['Age', 'Pclass', 'SibSp', 'Parch'])
train_data[['Age']] = train_data[['Age']].fillna(value = train_data[['Age']].median())

missing_index = train_data.isnull().any(1).to_numpy().nonzero()[0]
train_data = train_data.dropna()

train_label = pd.DataFrame(dataset, columns = ['Survived']).drop(missing_index)
```

```
test_data = pd.DataFrame(test_dataset, columns = ['Age', 'Pclass', 'SibSp', 'Parch'])

missing_index_test = test_data.isnull().any(1).to_numpy().nonzero()[0]
test_data = test_data.dropna()

test_label = pd.DataFrame(test_label, columns = ['Survived']).drop(missing_index_test)
```

```
dtc = DecisionTreeClassifier(max_depth=3)
dtc.fit(train_data, train_label)

class_result = dtc.predict(test_data)

accuration = dtc.score(test_data, test_label)
error = round((1-accuration)*100, 2)
accuration = round((accuration)*100, 2)

print('Accuration --> ', accuration, '%')
print('Error Ratio --> ', error, '%')
```

Percobaan menggunakan kolom Age, Pclass, SibSp, Parch

```
Accuration -->  62.05 %
Error Ratio -->  37.95 %
```

```
train_data = pd.DataFrame(dataset, columns = ['Sex', 'Age', 'Pclass', 'Fare', 'SibSp', 'Parch', 'Embarked']).replace(['m
train_data[['Age']] = train_data[['Age']].fillna(value = train_data[['Age']].median())

missing_index = train_data.isnull().any(1).to_numpy().nonzero()[0]
train_data = train_data.dropna()
train_data = train_data.replace(train_data.Embarked.unique(), [0,1,2])

train_label = pd.DataFrame(dataset, columns = ['Survived']).drop(missing_index)
```

```
test_data = pd.DataFrame(test_dataset, columns = ['Sex', 'Age', 'Pclass', 'Fare', 'SibSp', 'Parch', 'Embarked']).replace

missing_index_test = test_data.isnull().any(1).to_numpy().nonzero()[0]
test_data = test_data.dropna()
test_data = test_data.replace(test_data.Embarked.unique(), [0,1,2])

test_label = pd.DataFrame(test_label, columns = ['Survived']).drop(missing_index_test)
```

```
dtc = DecisionTreeClassifier(max_depth=3)
dtc.fit(train_data, train_label)

class_result = dtc.predict(test_data)

accuration = dtc.score(test_data, test_label)
error = round((1-accuration)*100, 2)
accuration = round((accuration)*100, 2)

print('Accuration --> ', accuration, '%')
print('Error Ratio --> ', error, '%')
```

```
Accuration -->  96.98 %
Error Ratio -->  3.02 %
```

Percobaan menggunakan kolom Sex, Age, Pclass, Fare, SibSp, Parch, Embarked

```
train_data = pd.DataFrame(dataset, columns = ['Sex', 'Pclass', 'SibSp', 'Parch']).replace(['male', 'female'], [0,1])

missing_index = train_data.isnull().any(1).to_numpy().nonzero()[0]
train_data = train_data.dropna()

train_label = pd.DataFrame(dataset, columns = ['Survived']).drop(missing_index)
```

```
test_data = pd.DataFrame(test_dataset, columns = ['Sex', 'Pclass', 'SibSp', 'Parch']).replace(['male', 'female'], [0,1])

missing_index_test = test_data.isnull().any(1).to_numpy().nonzero()[0]
test_data = test_data.dropna()

test_label = pd.DataFrame(test_label, columns = ['Survived']).drop(missing_index_test)
```

```
dtc = DecisionTreeClassifier(max_depth=3)
dtc.fit(train_data, train_label)

class_result = dtc.predict(test_data)

accuration = dtc.score(test_data, test_label)
error = round((1-accuration)*100, 2)
accuration = round((accuration)*100, 2)

print('Accuration --> ', accuration, '%')
print('Error Ratio --> ', error, '%')
```

Percobaan menggunakan kolom
Sex, Pclass, SibSp, Parch

```
Accuration -->  99.28 %
Error Ratio -->  0.72 %
```

Terimakasih