



MidTest

Minggu Ke-7



Faris Saifullah
3124640034

✓
0s

[17]

#no.1

import pandas as pd

dataset = pd.read_csv('titanic.csv')

(dataset.head())

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
[19] #no.2
data = dataset[['Sex', 'Age', 'Pclass', 'Fare', 'Survived']]
(data.head())
```

	Sex	Age	Pclass	Fare	Survived
0	male	22.0	3	7.2500	0
1	female	38.0	1	71.2833	1
2	female	26.0	3	7.9250	1
3	female	35.0	1	53.1000	1
4	male	35.0	3	8.0500	0

```
#no.3  
train_data = data[data['Age'].notna()][['Sex', 'Pclass', 'Fare', 'Survived']]  
(train_data.head())
```



	Sex	Pclass	Fare	Survived
0	0	3	7.2500	0
1	1	1	71.2833	1
2	1	3	7.9250	1
3	1	1	53.1000	1
4	0	3	8.0500	0



✓
0s


▶

#no.4
train_label = data[data['Age'].notna()]['Age']
(train_label.head())

⇅

	Age
0	22.0
1	38.0
2	26.0
3	35.0
4	35.0

```
#no.5  
test_data = data[data['Age'].isna()][['Sex', 'Pclass', 'Fare', 'Survived']]  
(test_data.head())
```



	Sex	Pclass	Fare	Survived
5	0	3	8.4583	0
17	0	2	13.0000	1
19	1	3	7.2250	1
26	0	3	7.2250	0
28	1	3	7.8792	1

```
[25] #no.6
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
train_data_normalized = scaler.fit_transform(train_data)
print("Normalized train_data:\n", train_data_normalized)
print("Min values of train_data:\n", scaler.data_min_)
print("Max values of train_data:\n", scaler.data_max_)

↵ Normalized train_data:
[[0.         1.         0.01415106 0.         ]
 [1.         0.         0.13913574 1.         ]
 [1.         1.         0.01546857 1.         ]
 ...
 [1.         0.         0.05855561 1.         ]
 [0.         0.         0.05855561 1.         ]
 [0.         1.         0.01512699 0.         ]]
Min values of train_data:
[0. 1. 0. 0.]
Max values of train_data:
[ 1.    3. 512.3292  1.]
```

```
#no.7
test_data_scaled = scaler.transform(test_data)
test_data_scaled_df = pd.DataFrame(test_data_scaled, columns=['Sex', 'Pclass', 'Fare', 'Survived'])
test_data_scaled_df.head()
```

	Sex	Pclass	Fare	Survived
0	0.0	1.0	0.016510	0.0
1	0.0	0.5	0.025374	1.0
2	1.0	1.0	0.014102	1.0
3	0.0	1.0	0.014102	0.0
4	1.0	1.0	0.015379	1.0




```
#no.8
from sklearn.neighbors import KNeighborsRegressor

knn_regressor = KNeighborsRegressor(n_neighbors=3)
knn_regressor.fit(train_data[['Sex', 'Pclass', 'Fare', 'Survived']], train_label)
class_result = knn_regressor.predict(test_data[['Sex', 'Pclass', 'Fare', 'Survived']])
print("Hasil prediksi usia untuk data yang hilang:", class_result)
```



```
Hasil prediksi usia untuk data yang hilang: [31.66666667 35.66666667 16.66666667 33.5      21.33333333 29.
26.97333333 20.      22.      29.      36.83333333 36.
20.      18.66666667 43.      38.5     10.66666667 29.
36.83333333 20.      36.83333333 36.83333333 29.      24.33333333
22.66666667 36.83333333 55.83333333 18.66666667 29.33333333 21.5
25.33333333 39.33333333 28.66666667 57.66666667 41.      39.33333333
41.33333333 42.      21.      55.83333333 20.      39.33333333
55.83333333 29.      33.33333333 32.16666667 25.83333333 21.
21.5     27.66666667 55.83333333 32.16666667 50.33333333 20.
35.66666667 57.66666667 38.5     37.      20.      29.33333333
25.      36.83333333 35.      39.33333333 18.66666667 43.66666667
29.      19.33333333 54.33333333 33.5     21.33333333 21.33333333
36.      16.66666667 20.      39.66666667 29.      26.
33.33333333 29.      31.      35.66666667 25.33333333 29.
21.5     55.83333333 19.33333333 17.66666667 22.33333333 36.83333333
45.      55.83333333 36.83333333 35.66666667 26.      21.5
40.      35.66666667 33.33333333 22.33333333 24.      38.83333333
32.16666667 42.66666667 36.83333333 33.      33.5     20.83333333
32.66666667 20.83333333 18.66666667 28.83333333 21.89     22.
32.66666667 55.83333333 36.83333333 25.33333333 20.83333333 20.
16.5     25.33333333 36.83333333 32.16666667 22.66666667 33.5
29.      16.66666667 26.33333333 21.      55.83333333 26.
39.      32.16666667 30.      29.33333333 29.      21.33333333
29.      24.66666667 45.      35.66666667 25.33333333 30.
18.      10.66666667 54.33333333 36.      18.      35.66666667
29.      29.      46.      28.83333333 47.66666667 33.
33.5     55.83333333 26.      33.      55.83333333 39.33333333
46.66666667 39.      32.66666667 28.66666667 25.      20.83333333
36.83333333 46.      39.33333333 39.      20.83333333 39.33333333
24.      29.      25.33333333]
```

```
#no.9
if len(class_result) == data['Age'].isna().sum():
    data.loc[data['Age'].isna(), 'Age'] = class_result
    print("\nData dengan nilai Age yang sudah diisi:")
    print(data.head())
else:
    print(f"Error: Jumlah prediksi ({len(class_result)}) tidak cocok dengan jumlah nilai Age yang hilang ({data['Age'].isna().sum()})")
```

Data dengan nilai Age yang sudah diisi:



	Sex	Age	Pclass	Fare	Survived
0	0	22.0	3	7.2500	0
1	1	38.0	1	71.2833	1
2	1	26.0	3	7.9250	1
3	1	35.0	1	53.1000	1
4	0	35.0	3	8.0500	0

```
#no.10
import pandas as pd
test_dataset = pd.read_csv('titanic_test.csv')
test_dataset.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S	



```
#no.11
train_data = pd.read_csv('titanic_test.csv')
train_data = train_data[['Sex', 'Age', 'Pclass', 'Fare']]
train_data.head()
```

	Sex	Age	Pclass	Fare	
0	male	34.5	3	7.8292	
1	female	47.0	3	7.0000	
2	male	62.0	2	9.6875	
3	male	27.0	3	8.6625	
4	female	22.0	3	12.2875	

```
#no.12  
train_label = data['Survived']  
data.head()
```



	Sex	Age	Pclass	Fare	Survived
0	0	22.0	3	7.2500	0
1	1	38.0	1	71.2833	1
2	1	26.0	3	7.9250	1
3	1	35.0	1	53.1000	1
4	0	35.0	3	8.0500	0

```
#no.13  
test_data = test_dataset[['Sex', 'Age', 'Pclass', 'Fare']].dropna()  
test_data.head()
```

	Sex	Age	Pclass	Fare
0	male	34.5	3	7.8292
1	female	47.0	3	7.0000
2	male	62.0	2	9.6875
3	male	27.0	3	8.6625
4	female	22.0	3	12.2875



```
#no.14  
test_label = pd.read_csv("titanic_testlabel.csv")  
test_label.head()
```

	PassengerId	Survived	
0	892	0	
1	893	1	
2	894	0	
3	895	0	
4	896	1	

```
#no.15
train_data = pd.read_csv('titanic.csv')
train_data = train_data[['Sex', 'Age', 'Pclass', 'Fare']]
le = LabelEncoder()
train_data['Sex'] = le.fit_transform(train_data['Sex'])
train_data['Age'].fillna(train_data['Age'].mean(), inplace=True)
scaler = MinMaxScaler()
train_data = scaler.fit_transform(train_data)
min_values = scaler.data_min_
max_values = scaler.data_max_
train_data = pd.DataFrame(train_data, columns=['Sex', 'Age', 'Pclass', 'Fare'])
train_data.head()
print("Normalized train_data:\n", train_data)
print("Min values of train_data:\n", min_values)
print("Max values of train_data:\n", max_values)
```

```
Normalized train_data:
   Sex  Age  Pclass  Fare
0  1.0  0.271174    1.0  0.014151
1  0.0  0.472229    0.0  0.139136
2  0.0  0.321438    1.0  0.015469
3  0.0  0.434531    0.0  0.103644
4  1.0  0.434531    1.0  0.015713
..  ...    ...    ...    ...
886 1.0  0.334004    0.5  0.025374
887 0.0  0.233476    0.0  0.058556
888 0.0  0.367921    1.0  0.045771
889 1.0  0.321438    0.0  0.058556
890 1.0  0.396833    1.0  0.015127
```