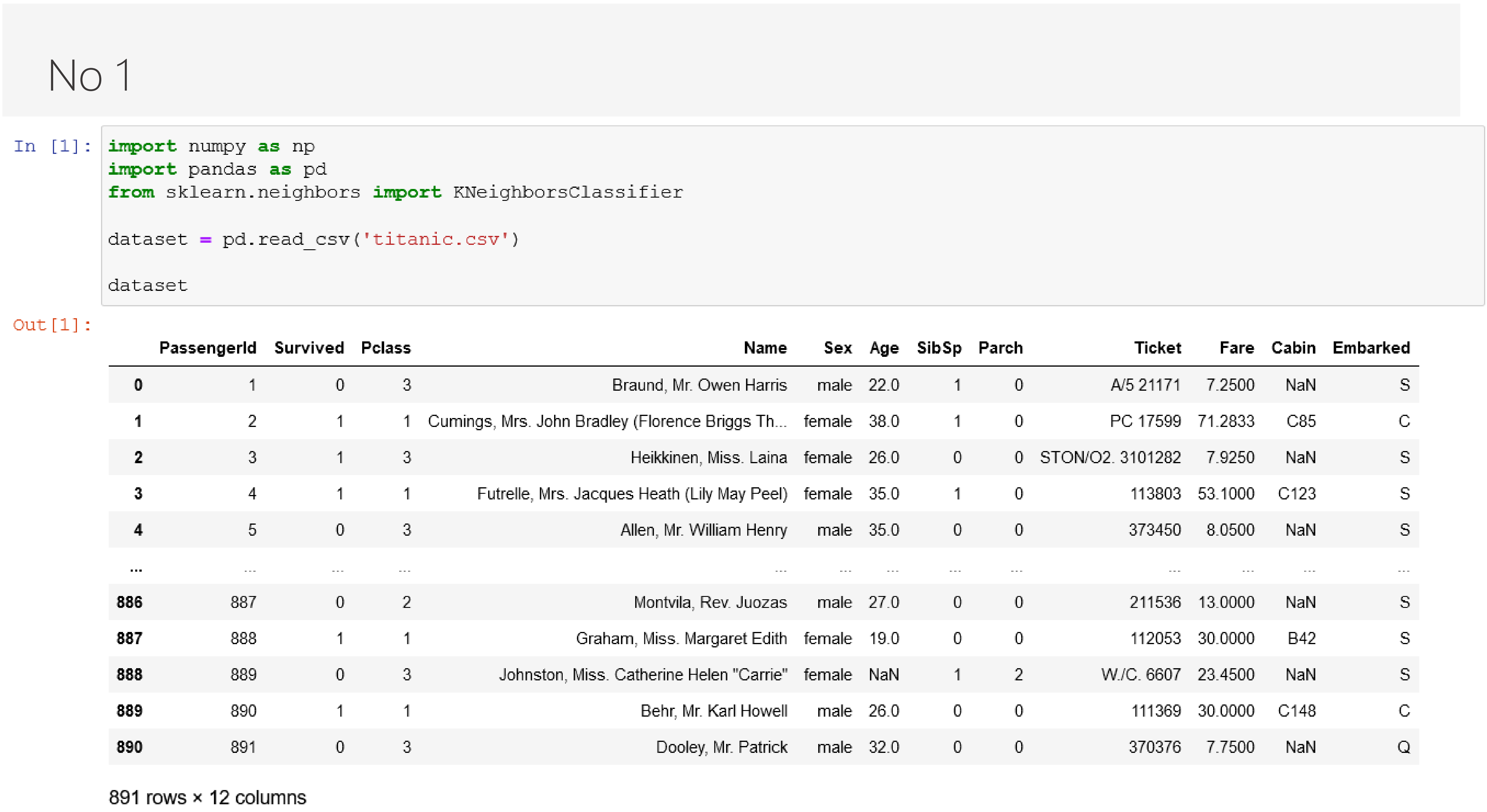
Praktikum Data Preprocessing

Classification

Ulima Inas Shabrina(2110181048)

Assignment# - Klasifikasi dengan k-NN

1. dataset ← titanic.csv
2. test_dataset ← titanic_test.csv
3. train_data ← ambil dataset kolom fitur (Age, Fare). Hilangkan baris data yang terdapat missing values (catat posisi data yang hilang → pos_missing_train)
4. test_data ← ambil test_dataset kolom fitur (Age, Fare). Hilangkan baris data yang terdapat missing values (catat posisi data yang hilang → pos_missing_test)
5. train_label ← ambil dataset kolom kelas (Survived), yang bukan pos_missing_train
6. test_label ← titanic_testlabel.csv, yang bukan pos_missing_test
7. train_data ← lakukan normalisasi pada train_data dengan Min-Max 0-1 (catat nilai min dan max setiap atribut)
8. test_data ← lakukan normalisasi pada test_data dengan Min-Max 0-1 (dengan nilai min dan max setiap atribut pada Langkah 7)
9. class_result ← Lakukan klasifikasi test_data terhadap train_data dengan k-NN (k=1..10)
10. Bandingkan hasil klasifikasi class_result dengan test_label. Jika tidak sama berarti error. Berapakah error ratio-nya untuk masing-masing k?



No 1

In [1]:

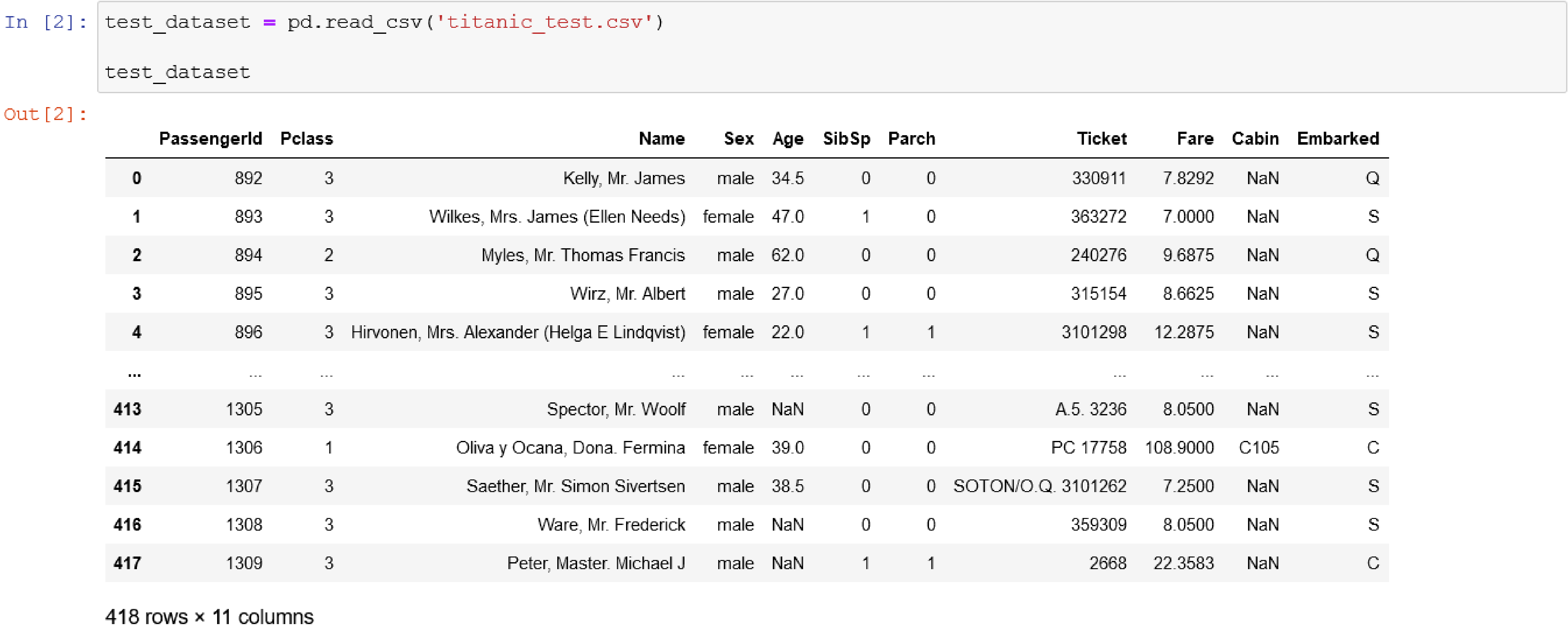
```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier

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

dataset
```

Out[1]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

No 2



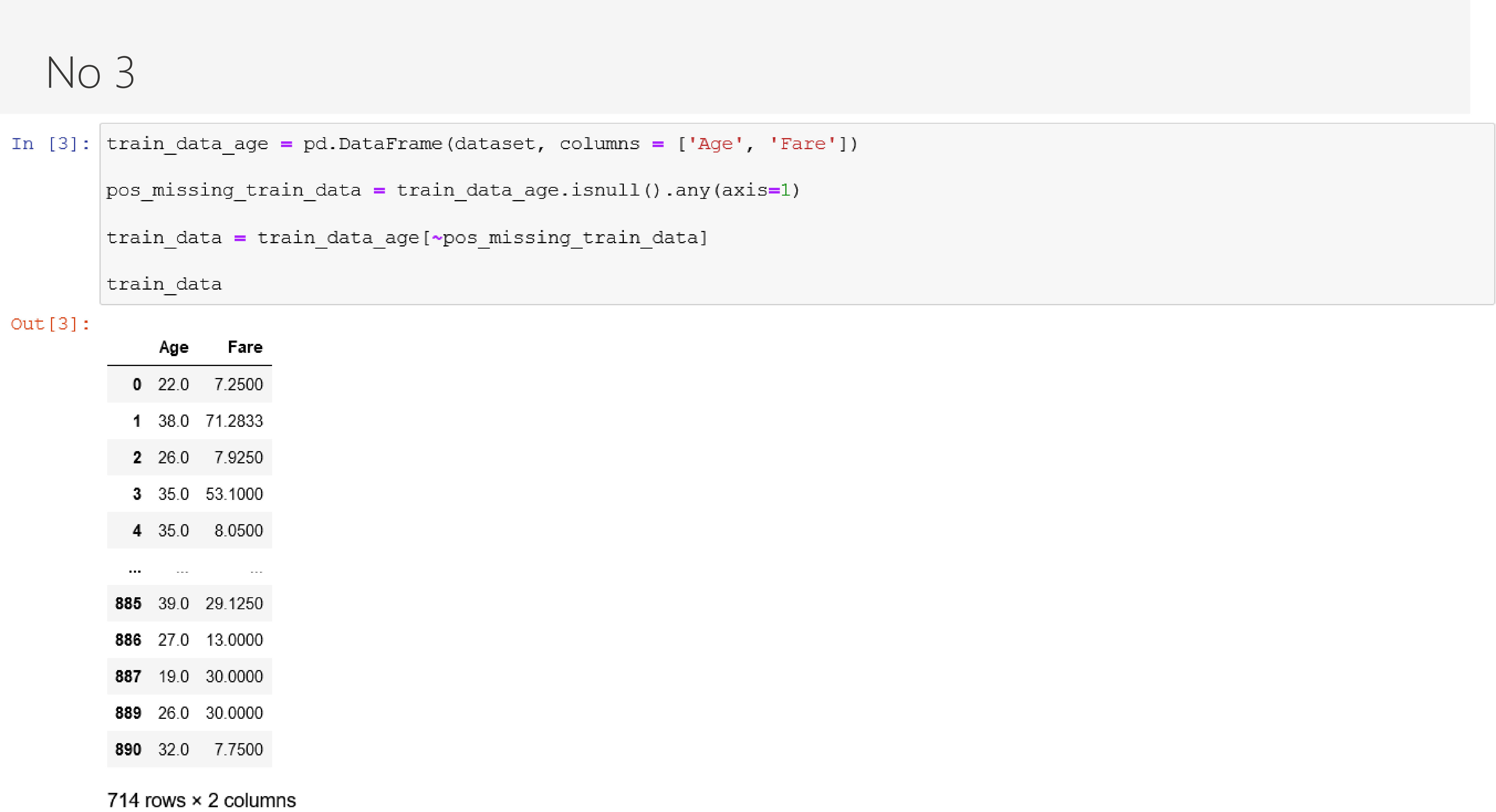
In [2]:

```
test_dataset = pd.read_csv('titanic_test.csv')

test_dataset
```

Out[2]:

	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
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

418 rows × 11 columns



No 3

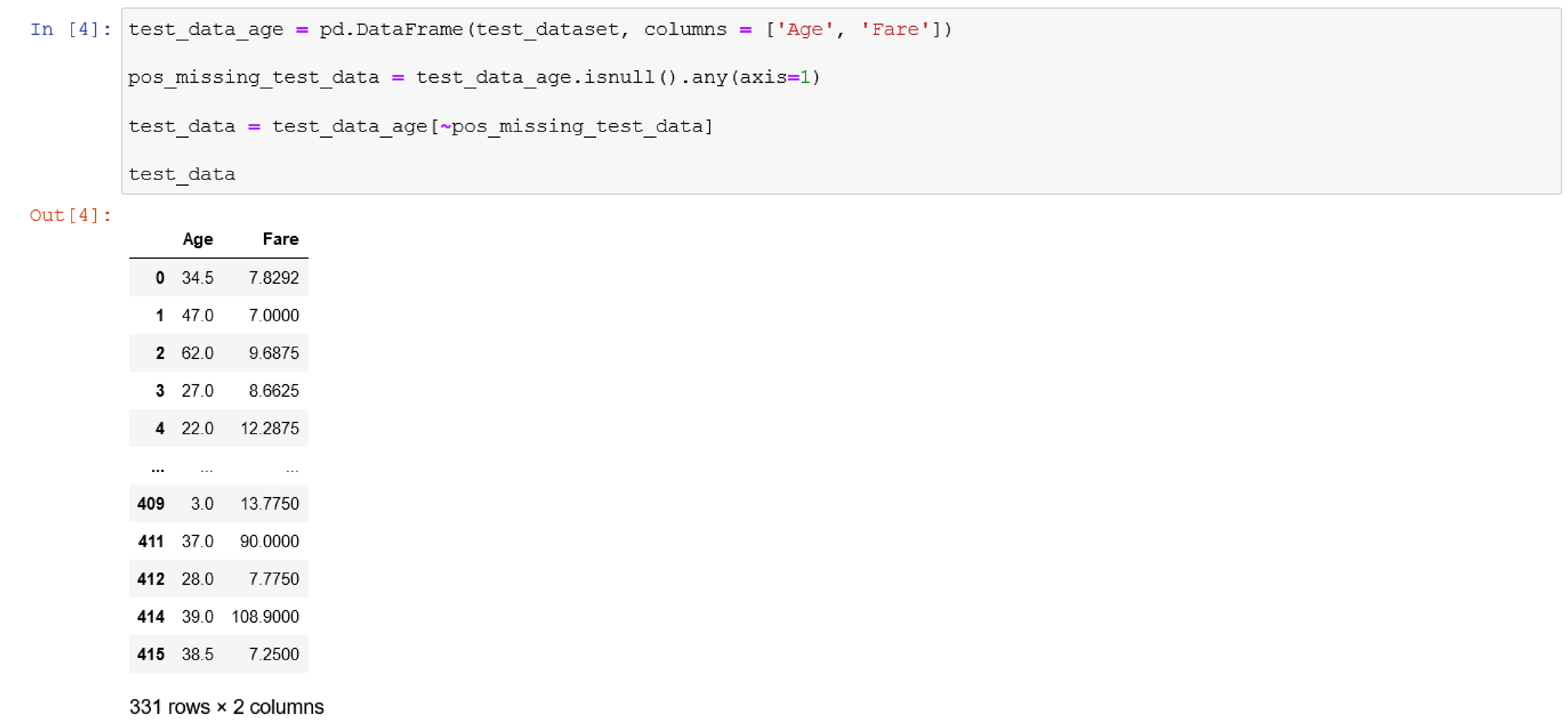
In [3]:

```
train_data_age = pd.DataFrame(dataset, columns = ['Age', 'Fare'])

pos_missing_train_data = train_data_age.isnull().any(axis=1)

train_data = train_data_age[~pos_missing_train_data]

train_data
```

Out[3]:

```
      Age     Fare
  0  22.0   7.2500
  1  38.0  71.2833
  2  26.0   7.9250
  3  35.0  53.1000
  4  35.0   8.0500
...   ...      ...
885  39.0  29.1250
886  27.0  13.0000
887  19.0  30.0000
889  26.0  30.0000
890  32.0   7.7500

714 rows × 2 columns
```

No 4



In [4]:

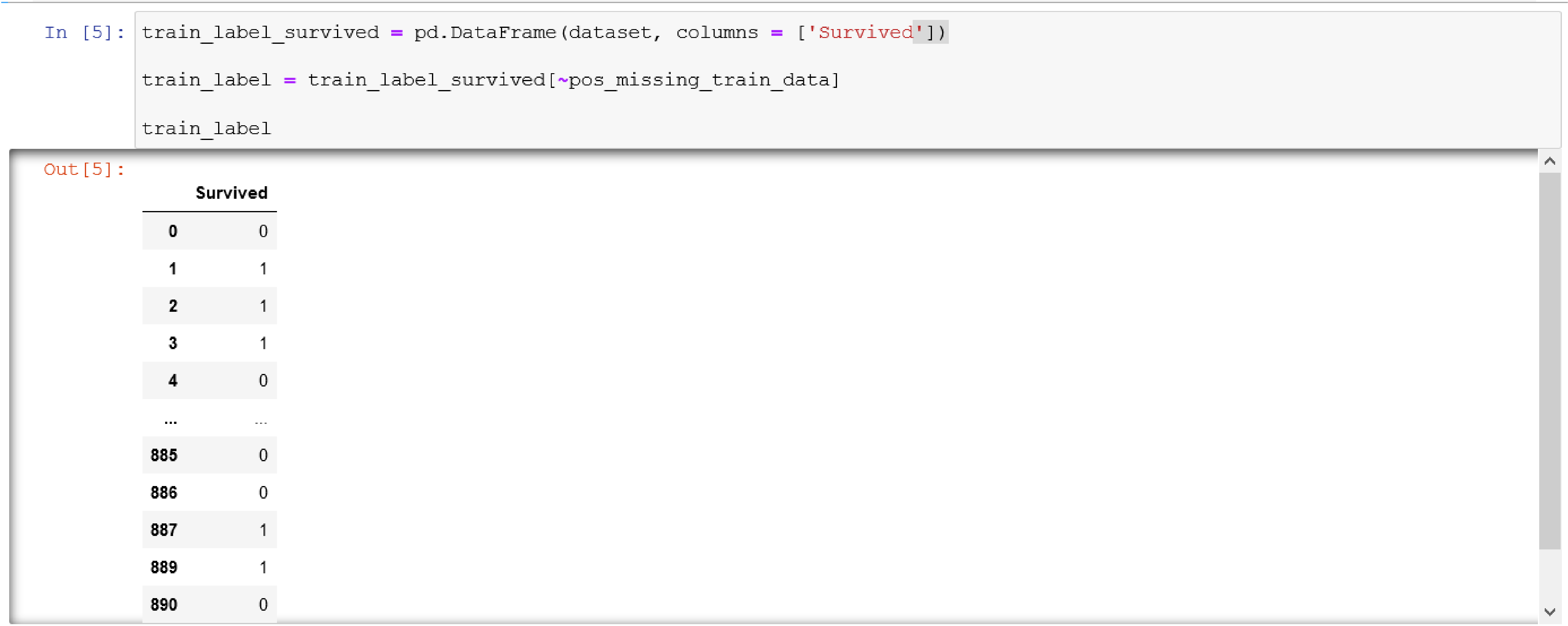
```
test_data_age = pd.DataFrame(test_dataset, columns = ['Age', 'Fare'])

pos_missing_test_data = test_data_age.isnull().any(axis=1)

test_data = test_data_age[~pos_missing_test_data]

test_data
```

Out[4]:

	Age	Fare
0	34.5	7.8292
1	47.0	7.0000
2	62.0	9.6875
3	27.0	8.6625
4	22.0	12.2875
...	...	...
409	3.0	13.7750
411	37.0	90.0000
412	28.0	7.7750
414	39.0	108.9000
415	38.5	7.2500

331 rows × 2 columns

No 5



In [5]:

```
train_label_survived = pd.DataFrame(dataset, columns = ['Survived'])

train_label = train_label_survived[~pos_missing_train_data]

train_label
```

Out[5]:

	Survived
0	0
1	1
2	1
3	1
4	0
...	...
885	0
886	0
887	1
889	1
890	0

No 6

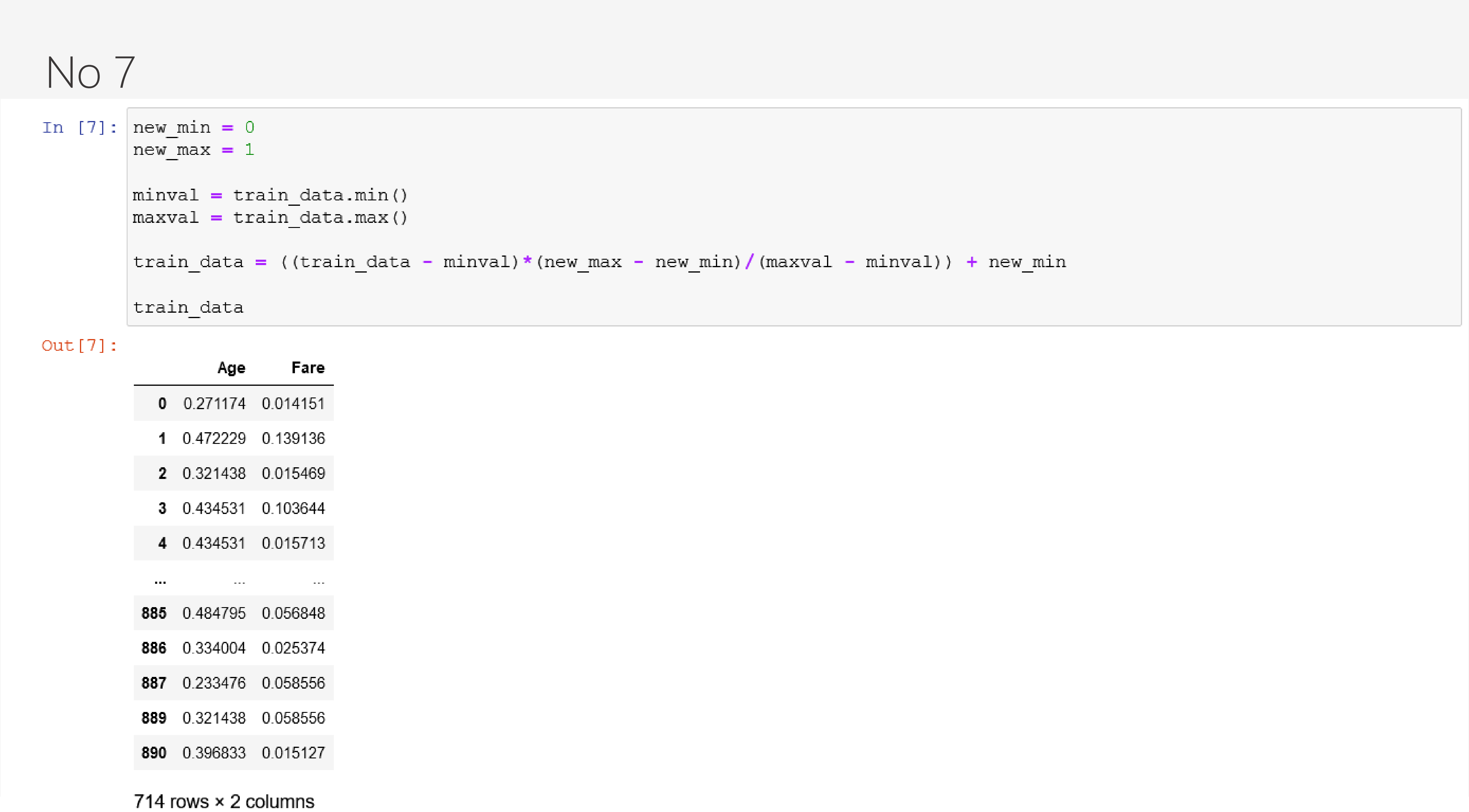
In [6]:

```
test_label = pd.read_csv('titanic_testlabel.csv')

test_label_survived = pd.DataFrame(test_label, columns = ['Survived'])

test_label2 = test_label_survived[~pos_missing_test_data]

test_label2
```

Out[6]:

	Survived
0	0
1	1
2	0
3	0
4	1
...	...
409	1
411	1
412	1
414	1
415	0

331 rows × 1 columns



No 7

In [7]:

```
new_min = 0
new_max = 1

minval = train_data.min()
maxval = train_data.max()

train_data = ((train_data - minval)*(new_max - new_min)/(maxval - minval)) + new_min

train_data
```

Out[7]:

	Age	Fare
0	0.271174	0.014151
1	0.472229	0.139136
2	0.321438	0.015469
3	0.434531	0.103644
4	0.434531	0.015713
...	...	...
885	0.484795	0.056848
886	0.334004	0.025374
887	0.233476	0.058556
889	0.321438	0.058556
890	0.396833	0.015127

714 rows × 2 columns

No 8

In [8]:

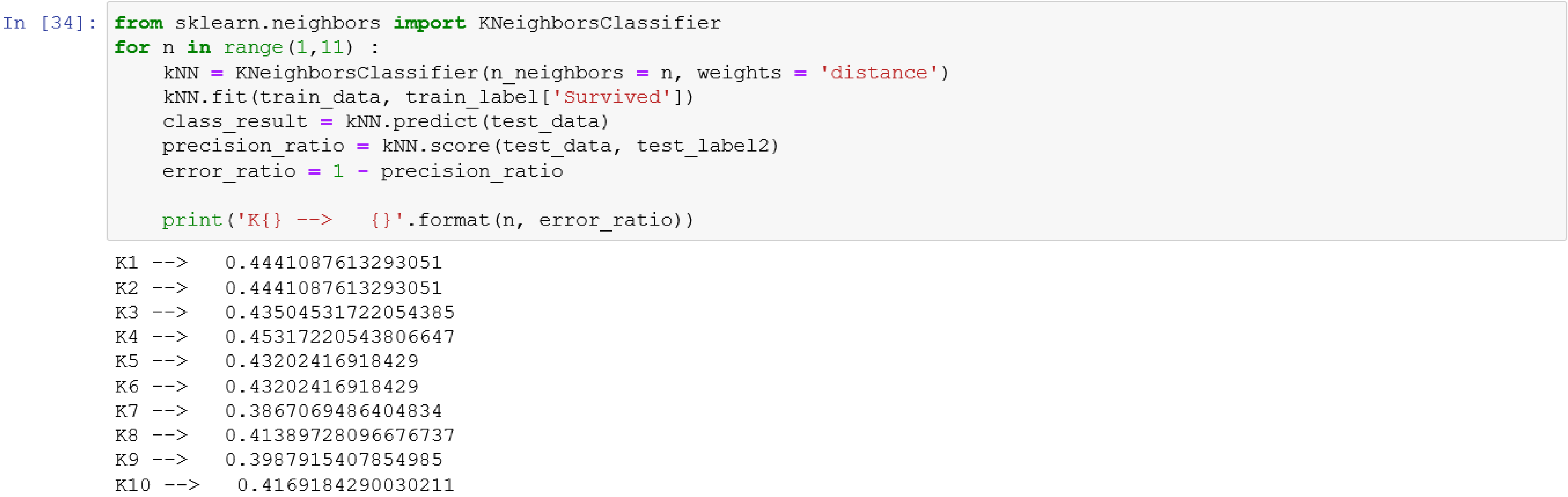
```
new_min = 0
new_max = 1

minval = test_data.min()
maxval = test_data.max()

test_data = ((test_data - minval)*(new_max - new_min)/(maxval - minval)) + new_min

test_data
```

Out[8]:

	Age	Fare
0	0.452723	0.015282
1	0.617566	0.013663
2	0.815377	0.018909
3	0.353818	0.016908
4	0.287881	0.023984
...	...	...
409	0.037320	0.026887
411	0.485692	0.175668
412	0.367005	0.015176
414	0.512066	0.212559
415	0.505473	0.014151

331 rows × 2 columns

No 9 & 10



```
In [34]: from sklearn.neighbors import KNeighborsClassifier
         for n in range(1,11) :
             kNN = KNeighborsClassifier(n_neighbors = n, weights = 'distance')
             kNN.fit(train_data, train_label['Survived'])
             class_result = kNN.predict(test_data)
             precision_ratio = kNN.score(test_data, test_label2)
             error_ratio = 1 - precision_ratio

             print('K{} -->   {}'.format(n, error_ratio))
```

```
K1 -->   0.4441087613293051
K2 -->   0.4441087613293051
K3 -->   0.43504531722054385
K4 -->   0.45317220543806647
K5 -->   0.43202416918429
K6 -->   0.43202416918429
K7 -->   0.3867069486404834
K8 -->   0.41389728096676737
K9 -->   0.3987915407854985
K10 -->   0.4169184290030211
```

Terimakasih