Proyek Ujian Akhir Soft Computing
Artificial Neural Network

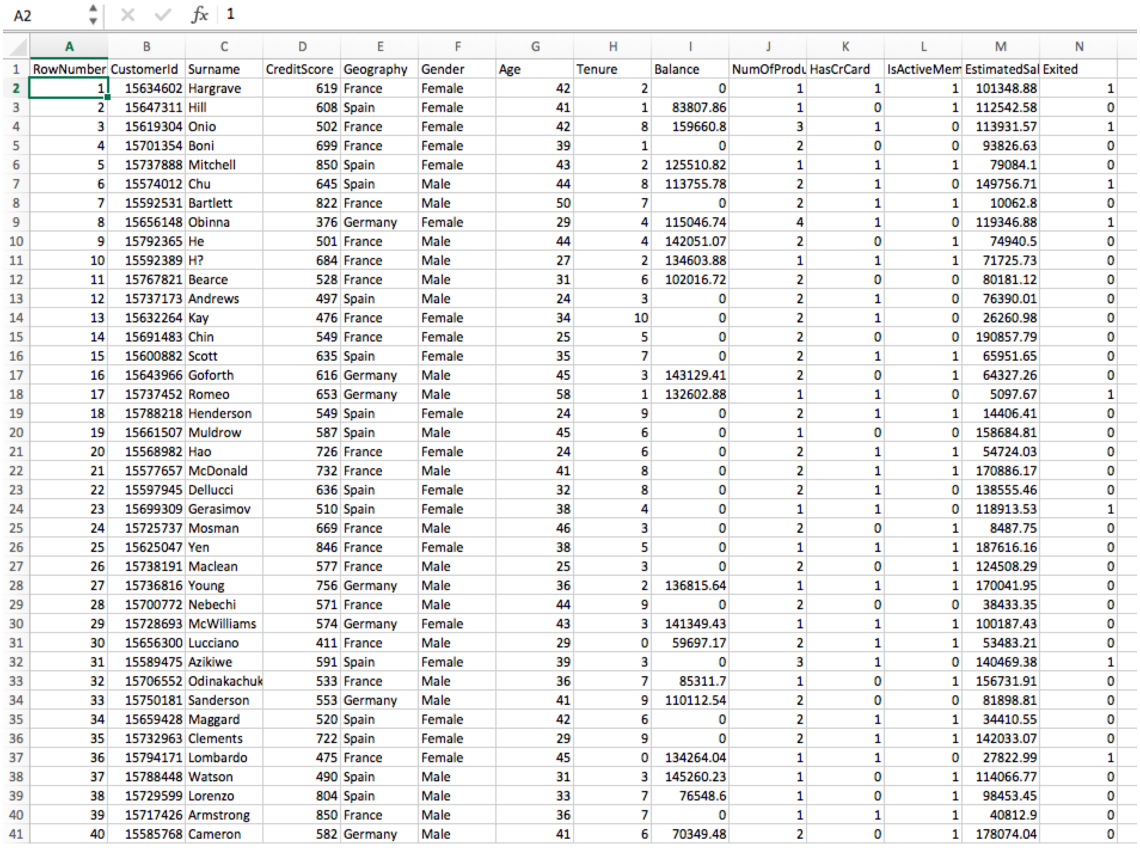
Disusun Guna Memenuhi Tugas Mata Kuliah Soft Computing yang Diampu oleh Ida Bagus Kresna Sudiatmika, S.Kom. M.T.

Oleh:

Reza Pahlevi
1601020015
Teknik Informatika

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER PRIMAKARA
PROGRAM STUDI INFORMATIKA
2018

Dalam UAS kali ini, kita akan mengerjakan sebuah program pelatihan data menggunakan Python. Beberapa package yang harus diimpor terlebih dahulu ialah numpy, matplotlib, panda, tensorflow, dan keras. Package tersebut dibutuhkan agar program bisa berjalan. Berikut ini adalah dataset yang berformat .csv untuk digunakan sebagai data latih:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProdu	HasCrCard	IsActiveMem	EstimatedSal	Exited
2	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.88	1
3	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
4	3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	1
5	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
6	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.1	0
7	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1
8	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
9	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
10	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.5	0
11	10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73	0
12	11	15767821	Bearce	528	France	Male	31	6	102016.72	2	0	0	80181.12	0
13	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0
14	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	0
15	14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.79	0
16	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.65	0
17	16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	0	1	64327.26	0
18	17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1	1	0	5097.67	1
19	18	15788218	Henderson	549	Spain	Female	24	9	0	2	1	1	14406.41	0
20	19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.81	0
21	20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.03	0
22	21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.17	0
23	22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.46	0
24	23	15699309	Gerasimov	510	Spain	Female	38	4	0	1	1	0	118913.53	1
25	24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0
26	25	15625047	Yen	846	France	Female	38	5	0	1	1	1	187616.16	0
27	26	15738191	Maclean	577	France	Male	25	3	0	2	0	1	124508.29	0
28	27	15736816	Young	756	Germany	Male	36	2	136815.64	1	1	1	170041.95	0
29	28	15700772	Nebechi	571	France	Male	44	9	0	2	0	0	38433.35	0
30	29	15728693	McWilliams	574	Germany	Female	43	3	141349.43	1	1	1	100187.43	0
31	30	15656300	Lucciano	411	France	Male	29	0	59697.17	2	1	1	53483.21	0
32	31	15589475	Azikiwe	591	Spain	Female	39	3	0	3	1	0	140469.38	1
33	32	15706552	Odinakachuk	533	France	Male	36	7	85311.7	1	0	1	156731.91	0
34	33	15750181	Sanderson	553	Germany	Male	41	9	110112.54	2	0	0	81898.81	0
35	34	15659428	Maggard	520	Spain	Female	42	6	0	2	1	1	34410.55	0
36	35	15732963	Clements	722	Spain	Female	29	9	0	2	1	1	142033.07	0
37	36	15794171	Lombardo	475	France	Female	45	0	134264.04	1	1	0	27822.99	1
38	37	15788448	Watson	490	Spain	Male	31	3	145260.23	1	0	1	114066.77	0
39	38	15729599	Lorenzo	804	Spain	Male	33	7	76548.6	1	0	1	98453.45	0
40	39	15717426	Armstrong	850	France	Male	36	7	0	1	1	1	40812.9	0
41	40	15585768	Cameron	582	Germany	Male	41	6	70349.48	2	0	1	178074.04	0

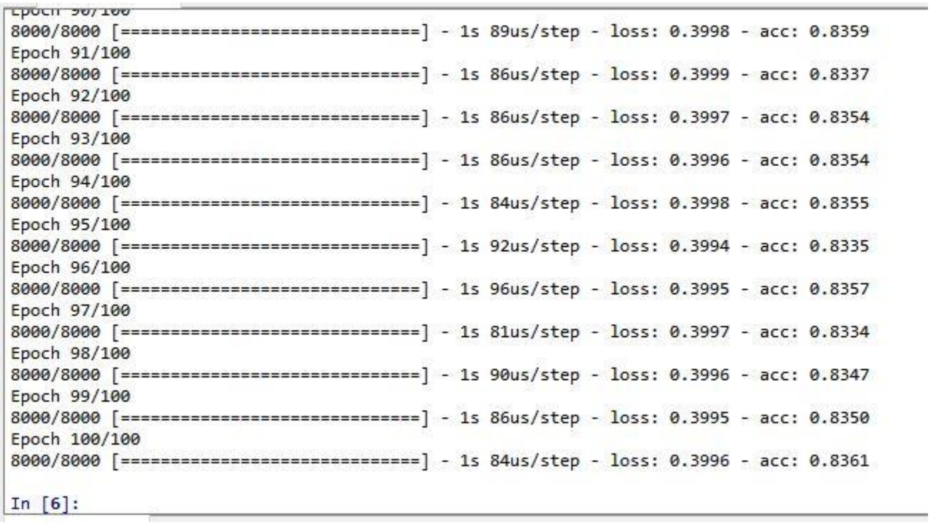
Kemudian, untuk sintaks programnya bisa dilihat dibawah ini:

```
# Data preprocessing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#----------------Data Preprocessing-------------------

# Import dataset
dataset = pd.read_csv('Churn_Modelling.csv')
```

```
#data dari kolom ke-3 sampai ke-12
X = dataset.iloc[:, 3:13].values

#jumlah data (1 or 0)
y = dataset.iloc[:, 13].values

# mengubah data kategorikal ke data numerik
# karena ANN hanya bisa bekerja pada data numerik
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
X[:, 1] = labelencoder_X_1.fit_transform(X[:, 1])

# Setelah ini, kita akan lihat kolom countries berubah menjadi angka
labelencoder_X_2 = LabelEncoder()
X[:, 2] = labelencoder_X_2.fit_transform(X[:, 2])

onehotencoder = OneHotEncoder(categorical_features = [1])
X = onehotencoder.fit_transform(X).toarray()

#Menghilangkan variabel dummy
X = X[:, 1:]

# Membagi dataset menjadi data latih dan data uji coba
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

#  ------------------Pembuatan ANN-------------------------
# Import Keras
import keras
from keras.models import Sequential
from keras.layers import Dense

#Mendefinisikan ANN
classifier = Sequential()

# menambah input layer and hidden layer no.1
# 6 output nodes, Relu activation function and 11 input nodes
# Output node ditentukan dari jumlah input nodes+1/2
# pastikan bobotnya diberikan nomor acak dengan nomor kecil yang
mendekati nol

classifier.add(Dense(output_dim = 6, init = 'uniform', activation =
'relu', input_dim = 11))

# Menambahkan hidden layer kedua untuk mencapai deep neural network
classifier.add(Dense(output_dim = 6, init = 'uniform', activation =
'relu'))
```

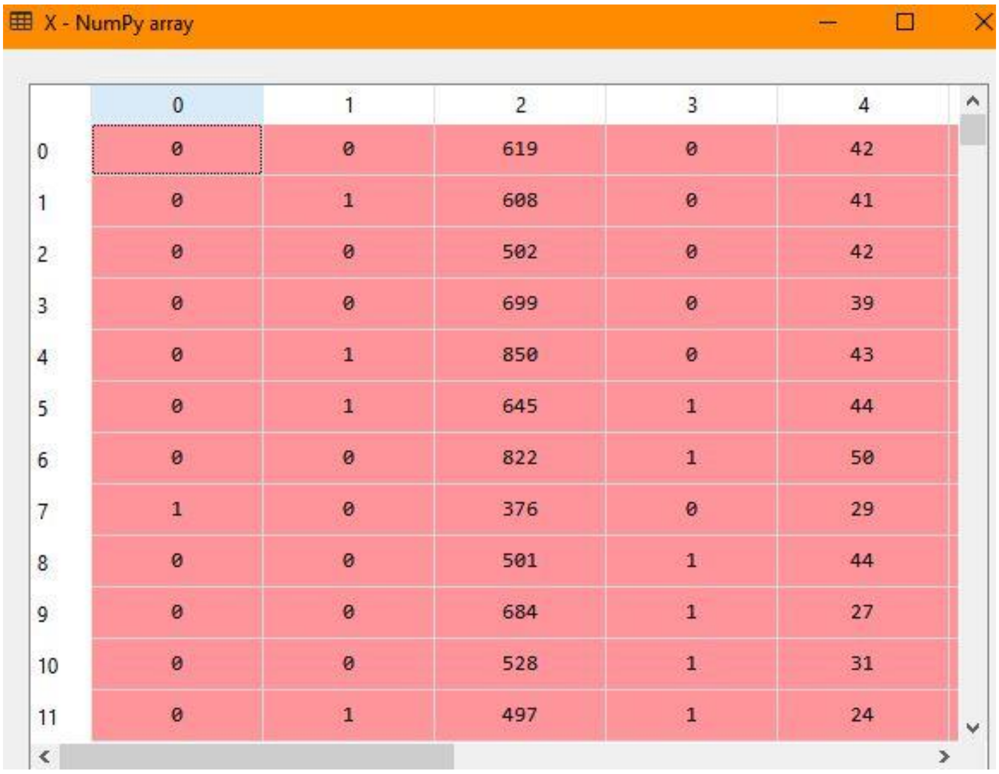
```
# Menambahkan Output layer
classifier.add(Dense(output_dim = 1, init = 'uniform', activation =
'sigmoid'))


# Mengeksekusi NN
# binary_crossentropy loss function digunakan jika output biner
tersebut ada
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy',
metrics = ['accuracy'])


classifier.fit(X_train, y_train, batch_size = 10, nb_epoch = 100)

# Menyesuaikan classifier ke data latih
# Kita akan membuat classifier di sini

# Memprediksi hasil data uji coba
y_pred = classifier.predict(X_test)

# Create a treshold to predict a true or false for leaving the
# the bank.
y_pred = (y_pred > 0.5)

# Membuat Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

##  1545 + 136 prediksi benar and 230 + 50 prediksi salah

# akurasi komputasi 1545 + 136 / 2000 prediksi == 0.8405 % akurat.
```

Apabila di-run, maka hasilnya akan seperti ini:



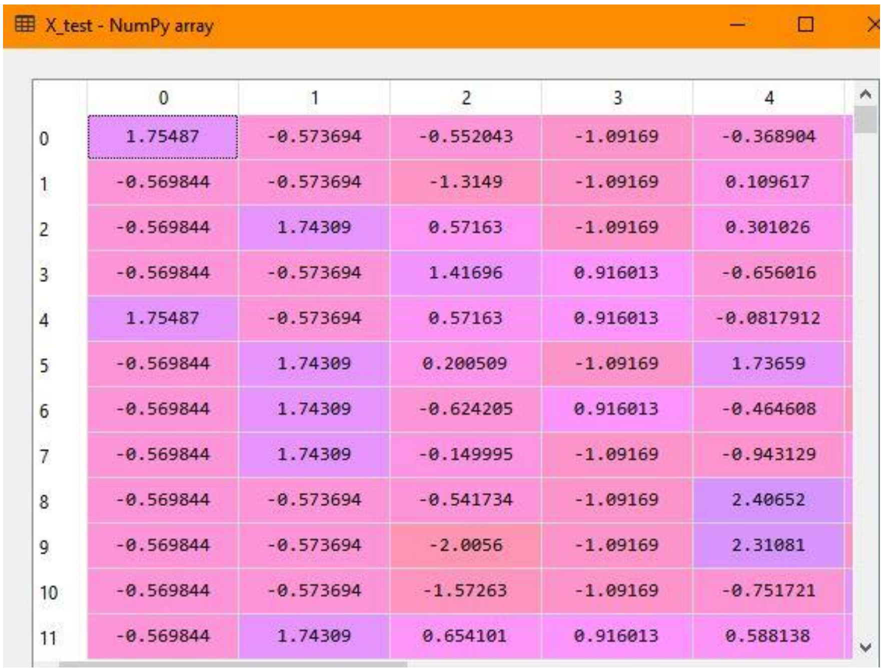
```
Epoch 90/100
8000/8000 [==============================] - 1s 89us/step - loss: 0.3998 - acc: 0.8359
Epoch 91/100
8000/8000 [==============================] - 1s 86us/step - loss: 0.3999 - acc: 0.8337
Epoch 92/100
8000/8000 [==============================] - 1s 86us/step - loss: 0.3997 - acc: 0.8354
Epoch 93/100
8000/8000 [==============================] - 1s 86us/step - loss: 0.3996 - acc: 0.8354
Epoch 94/100
8000/8000 [==============================] - 1s 84us/step - loss: 0.3998 - acc: 0.8355
Epoch 95/100
8000/8000 [==============================] - 1s 92us/step - loss: 0.3994 - acc: 0.8335
Epoch 96/100
8000/8000 [==============================] - 1s 96us/step - loss: 0.3995 - acc: 0.8357
Epoch 97/100
8000/8000 [==============================] - 1s 81us/step - loss: 0.3997 - acc: 0.8334
Epoch 98/100
8000/8000 [==============================] - 1s 90us/step - loss: 0.3996 - acc: 0.8347
Epoch 99/100
8000/8000 [==============================] - 1s 86us/step - loss: 0.3995 - acc: 0.8350
Epoch 100/100
8000/8000 [==============================] - 1s 84us/step - loss: 0.3996 - acc: 0.8361

In [6]:
```

Dari data X :



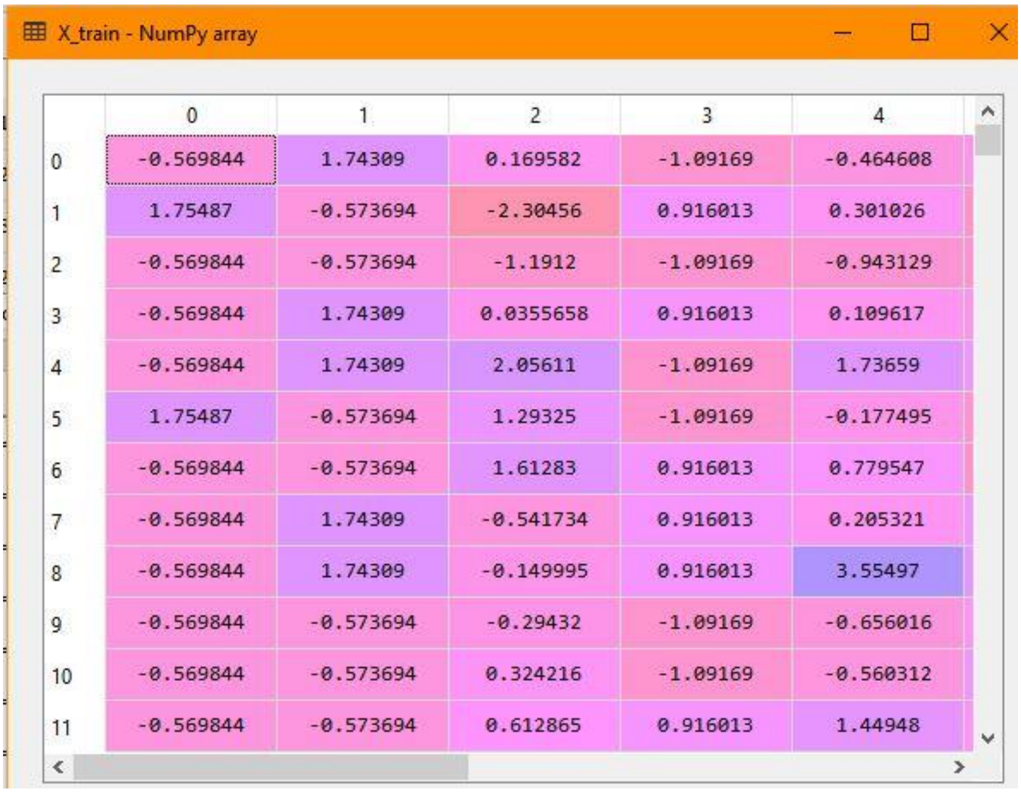
	0	1	2	3	4
0	0	0	619	0	42
1	0	1	608	0	41
2	0	0	502	0	42
3	0	0	699	0	39
4	0	1	850	0	43
5	0	1	645	1	44
6	0	0	822	1	50
7	1	0	376	0	29
8	0	0	501	1	44
9	0	0	684	1	27
10	0	0	528	1	31
11	0	1	497	1	24

Dari data X_test :



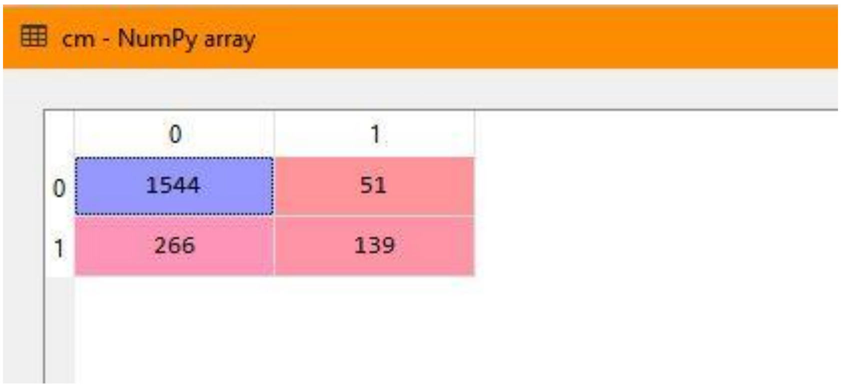
	0	1	2	3	4
0	1.75487	-0.573694	-0.552043	-1.09169	-0.368904
1	-0.569844	-0.573694	-1.3149	-1.09169	0.109617
2	-0.569844	1.74309	0.57163	-1.09169	0.301026
3	-0.569844	-0.573694	1.41696	0.916013	-0.656016
4	1.75487	-0.573694	0.57163	0.916013	-0.0817912
5	-0.569844	1.74309	0.200509	-1.09169	1.73659
6	-0.569844	1.74309	-0.624205	0.916013	-0.464608
7	-0.569844	1.74309	-0.149995	-1.09169	-0.943129
8	-0.569844	-0.573694	-0.541734	-1.09169	2.40652
9	-0.569844	-0.573694	-2.0056	-1.09169	2.31081
10	-0.569844	-0.573694	-1.57263	-1.09169	-0.751721
11	-0.569844	1.74309	0.654101	0.916013	0.588138

Dari data X_train :



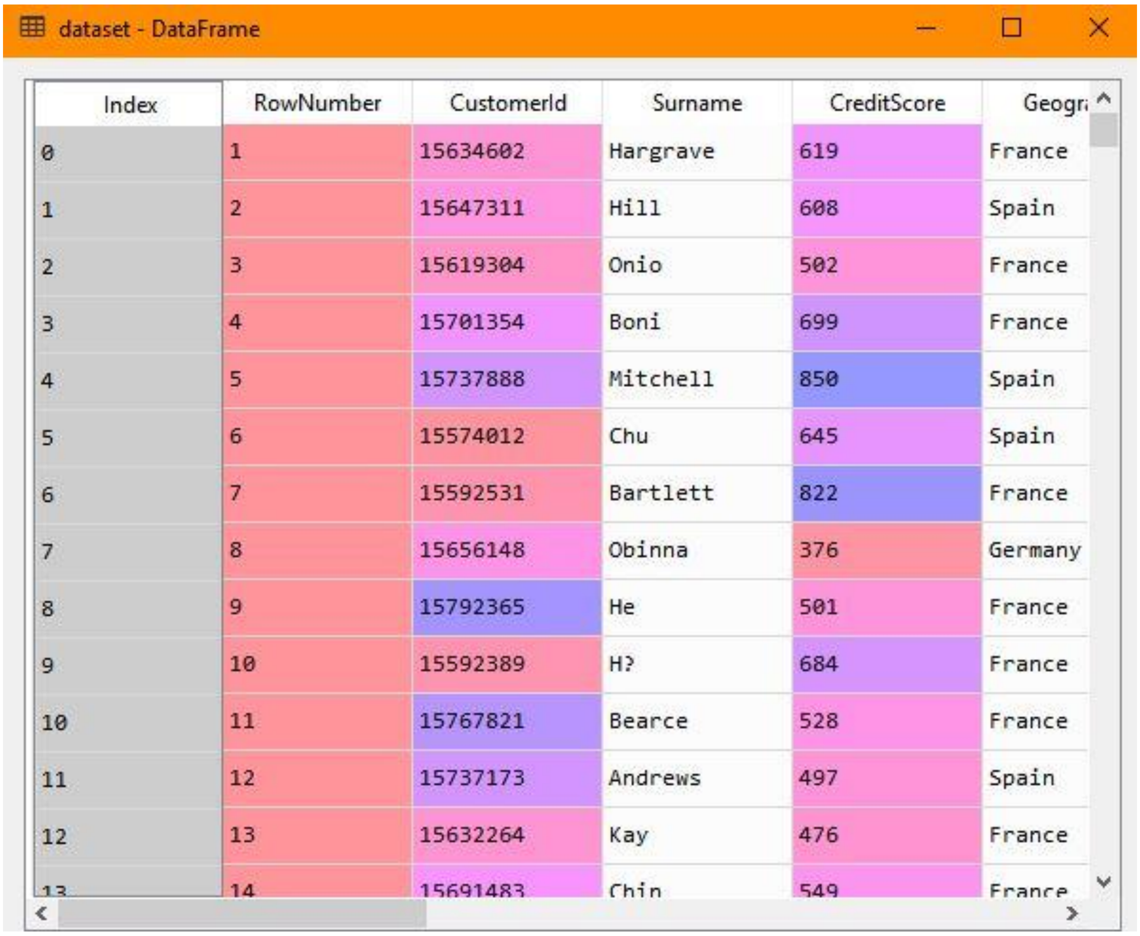
	0	1	2	3	4
0	-0.569844	1.74309	0.169582	-1.09169	-0.464608
1	1.75487	-0.573694	-2.30456	0.916013	0.301026
2	-0.569844	-0.573694	-1.1912	-1.09169	-0.943129
3	-0.569844	1.74309	0.0355658	0.916013	0.109617
4	-0.569844	1.74309	2.05611	-1.09169	1.73659
5	1.75487	-0.573694	1.29325	-1.09169	-0.177495
6	-0.569844	-0.573694	1.61283	0.916013	0.779547
7	-0.569844	1.74309	-0.541734	0.916013	0.205321
8	-0.569844	1.74309	-0.149995	0.916013	3.55497
9	-0.569844	-0.573694	-0.29432	-1.09169	-0.656016
10	-0.569844	-0.573694	0.324216	-1.09169	-0.560312
11	-0.569844	-0.573694	0.612865	0.916013	1.44948

Dari data cm :



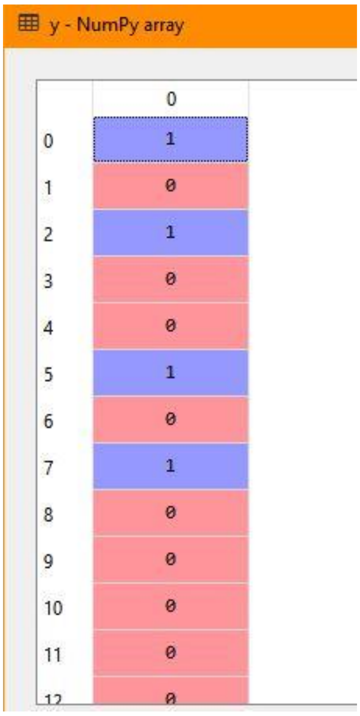
	0	1
0	1544	51
1	266	139

Dari data dataset :



Index	RowNumber	CustomerId	Surname	CreditScore	Geogr
0	1	15634602	Hargrave	619	France
1	2	15647311	Hill	608	Spain
2	3	15619304	Onio	502	France
3	4	15701354	Boni	699	France
4	5	15737888	Mitchell	850	Spain
5	6	15574012	Chu	645	Spain
6	7	15592531	Bartlett	822	France
7	8	15656148	Obinna	376	Germany
8	9	15792365	He	501	France
9	10	15592389	H?	684	France
10	11	15767821	Bearce	528	France
11	12	15737173	Andrews	497	Spain
12	13	15632264	Kay	476	France
13	14	15691483	Chin	549	France

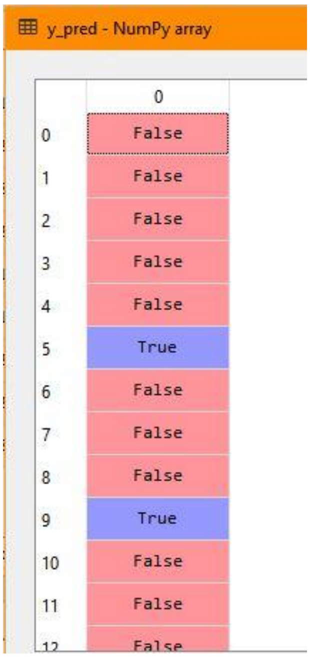
Dari data y :



y - NumPy array

	0
0	1
1	0
2	1
3	0
4	0
5	1
6	0
7	1
8	0
9	0
10	0
11	0
12	0

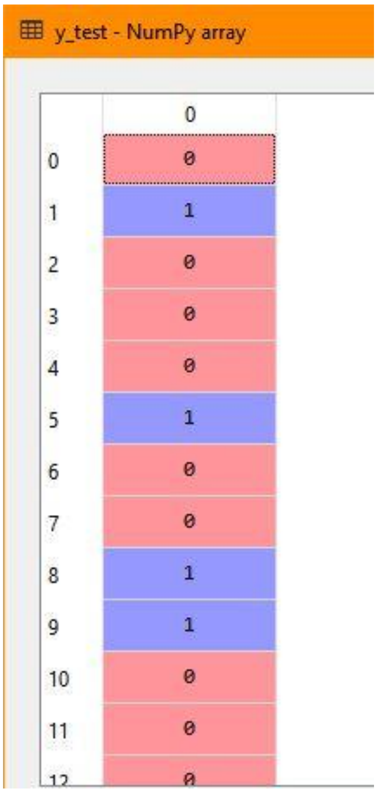
Dari data y_pred :



y_pred - NumPy array

	0
0	False
1	False
2	False
3	False
4	False
5	True
6	False
7	False
8	False
9	True
10	False
11	False
12	False

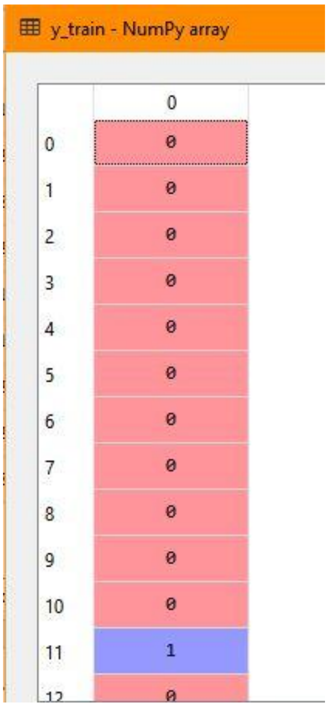
Dari data y_test :



y_test - NumPy array

	0
0	0
1	1
2	0
3	0
4	0
5	1
6	0
7	0
8	1
9	1
10	0
11	0
12	0

Dari data y_train :



y_train - NumPy array

	0
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	1
12	0