

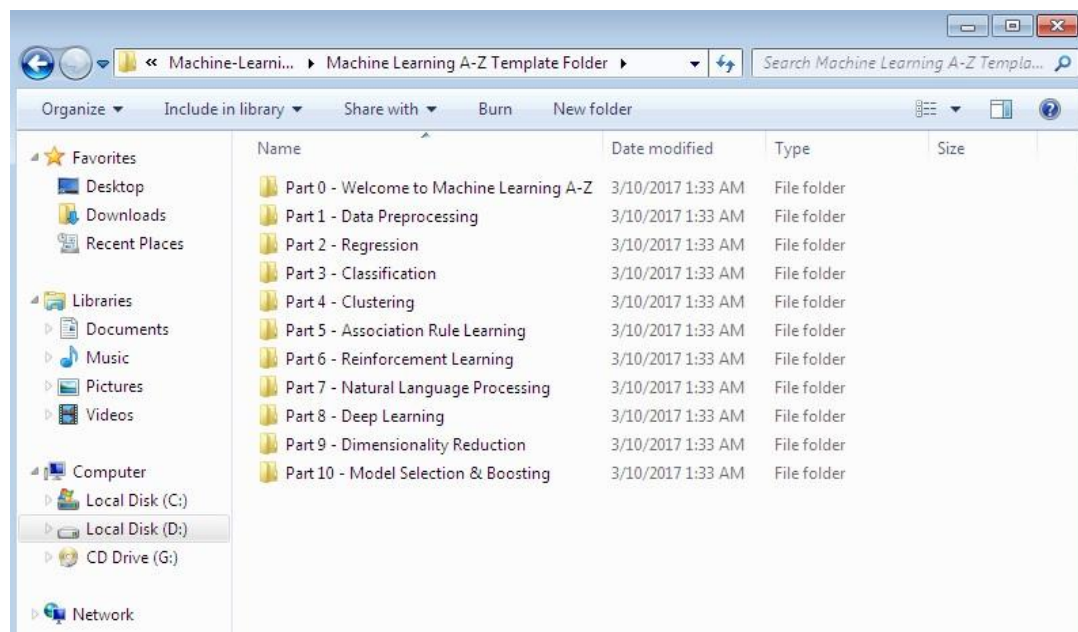
11S3207 - Pembelajaran Mesin

Week	:	04
Topic	:	Data Preprocessing
Due	:	End of session

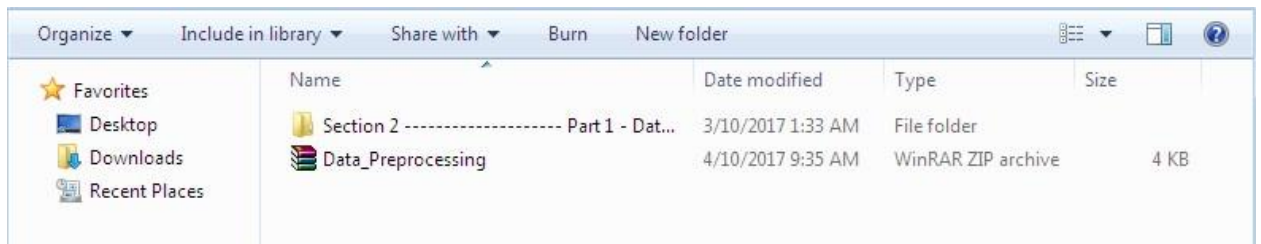
Salah satu bagian yang paling penting dalam pembuatan model Machine Learning adalah melakukan data preprocessing. Jika bagian ini tidak dilakukan, model machine learning tidak dapat berjalan dengan baik. Ketika ingin melakukan sebuah perjalanan, hal yang harus anda lakukan adalah mempersiapkan perjalanan anda, baik untuk pemesanan tiket penerbangan dan hotel, merencanakan itinerary dan lain- lain. Tahap persiapan ini dapat diandaikan sebagai tahap preprocessing sebelum model machine learning dibentuk. Salah satu aktivitas yang dilakukan pada tahapan ini adalah mempersiapkan Dataset yang akan digunakan untuk model machine learning. Berikut adalah tahapan yang dapat dilakukan untuk melakukan data preprocessing.

GET THE DATA SET

1. Download Data_Preprocessing.zip dan Machine-Learning-A-Z-Template-Folder.zip. Extract dan letakkan file Machine-Learning-A-Z-Template-Folder.zip pada lokasi dimana anda akan menyimpan file kerja anda. File ini anda gunakan sehingga setiap file kerja anda dapat disimpan secara lebih terstruktur.



2. Pindahkan dan extract Data_Preprocessing.zip kedalam folder D: \Machine-Learning-A-Z-Template-Folder\Machine Learning A-Z Template Folder\Part 1 - Data Preprocessing



3. Pindahkan semua data yang ada di folder Data_Preprocessing dan hapus folder yang telah anda extract sebelumnya.

Name	Date modified	Type	Size
Section 2 ----- Part 1 - Dat...	3/10/2017 1:33 AM	File folder	
categorical_data.py	9/12/2016 10:49 PM	PY File	1 KB
categorical_data.R	9/7/2016 7:07 AM	R File	1 KB
Data	9/7/2016 5:06 AM	Microsoft Excel C...	1 KB
data_preprocessing_template.py	9/3/2016 5:32 PM	PY File	1 KB
data_preprocessing_template.R	9/3/2016 5:37 PM	R File	1 KB
missing_data.py	9/7/2016 6:38 AM	PY File	1 KB
missing_data.R	9/7/2016 6:24 AM	R File	1 KB

4. Dataset yang akan digunakan menampilkan identitas dari pelanggan terdiri dari empat kolom, yaitu Country, Salary, Age, dan Purchased.

	A	B	C	D	E
1	Country	Age	Salary	Purchased	
2	France	44	72000	No	
3	Spain	27	48000	Yes	
4	Germany	30	54000	No	
5	Spain	38	61000	No	
6	Germany	40		Yes	
7	France	35	58000	Yes	
8	Spain		52000	No	
9	France	48	79000	Yes	
10	Germany	50	83000	No	
11	France	37	67000	Yes	
12					

5. Pada model machine learning, terdapat dua jenis variable yang berbeda, yaitu Independent Variable dan Dependent Variable. Independent Variable dapat digunakan untuk memprediksi Dependent Variable. Pada dataset diatas, Independent Variable adalah Country, Age, dan Salary, sementara Purchased adalah Dependent Variable, sehingga dengan menggunakan ketika independent variables yang ada, memungkinkan untuk memprediksi apakah pelanggan akan melakukan purchased terhadap produk.

IMPORTING THE LIBRARIES

Library adalah tool yang dapat digunakan untuk menyelesaikan suatu tugas khusus. Library dapat menerima input dari user dan menghasilkan output sesuai dengan yang diharapkan, sehingga setiap tugas dapat diselesaikan dengan lebih efisien. Pada praktikum kali ini, terdapat tiga libraries yang sangat dibutuhkan dalam data preprocessing, yaitu:

- numpy yang berisi tools matematika.
- matplotlib.pyplot yang dapat membantu membuat chart.
- pandas digunakan untuk import dan manage dataset.

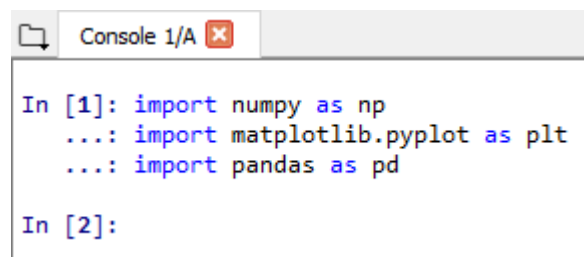
Berikut langkah- langkah yang dapat dilakukan untuk import dataset.

1. Buka editor Spyder yang telah diinstall sebelumnya, dan kemudian buat satu file baru dengan nama `data_preprocessing_template.py`, dan ketik code seperti gambar di bawah ini.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar  2 13:27:37 2020
4
5 @author: ranty
6 """
7
8 # Data Preprocessing
9 # Importing the Libraries
10
11 import numpy as np
12 import matplotlib.pyplot as plt
13 import pandas as pd
```

Line 11, 12, dan 13 membuat shortcut untuk masing- masing nama library yang akan digunakan.

2. Blok kode tersebut, klik kanan lalu pilih **Run Cell**, maka console akan menampilkan seperti gambar di bawah ini yang menunjukkan library telah berhasil diimport.



```
Console 1/A [X]

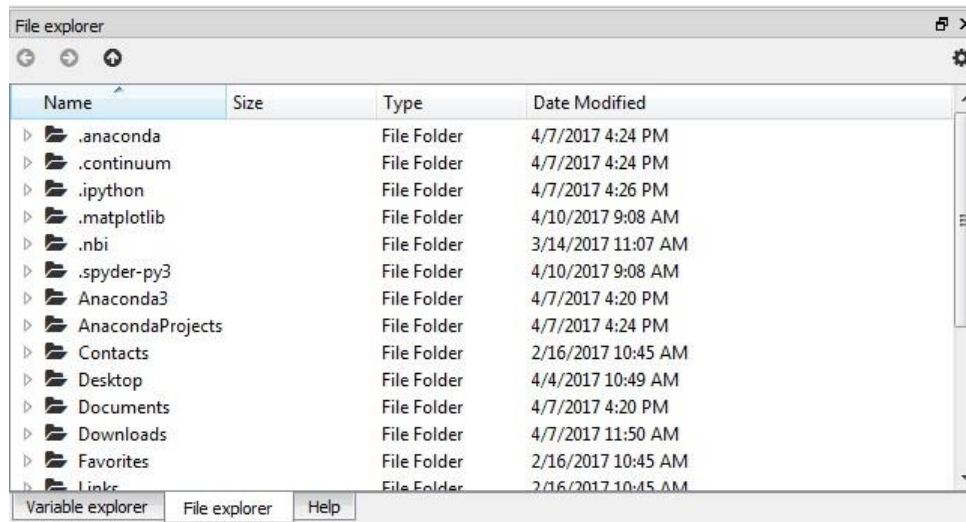
In [1]: import numpy as np
...: import matplotlib.pyplot as plt
...: import pandas as pd

In [2]:
```

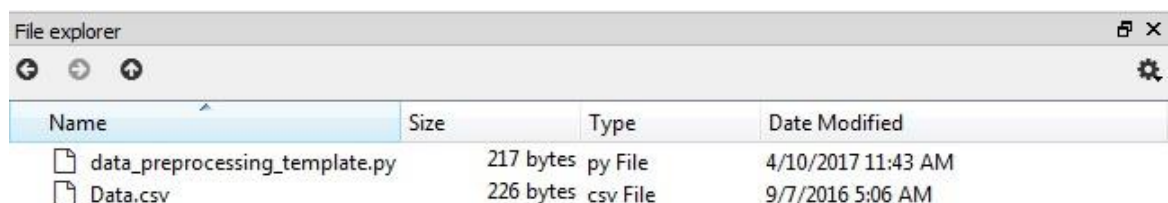
IMPORTING DATASET

Sebelum melakukan import dataset, langkah pertama yang harus dilakukan adalah menspesifikasikan working directory, dengan cara:

1. Klik File Explorer.



2. Tentukan working directory yang sudah ditentukan sebelumnya, yaitu pada folder MachineLearning-A-Z-Template-Folder, dan kemudian masuk ke folder Section 2 -----
---- Part 1 - Data Preprocessing -----. Anda dapat menentukan sendiri working directory selama ada dataset didalamnya. Dalam praktikum kali ini, anda dapat memindahkan file dataset Data.csv ke dalam folder yang sama dengan file data_preprocessing.py.



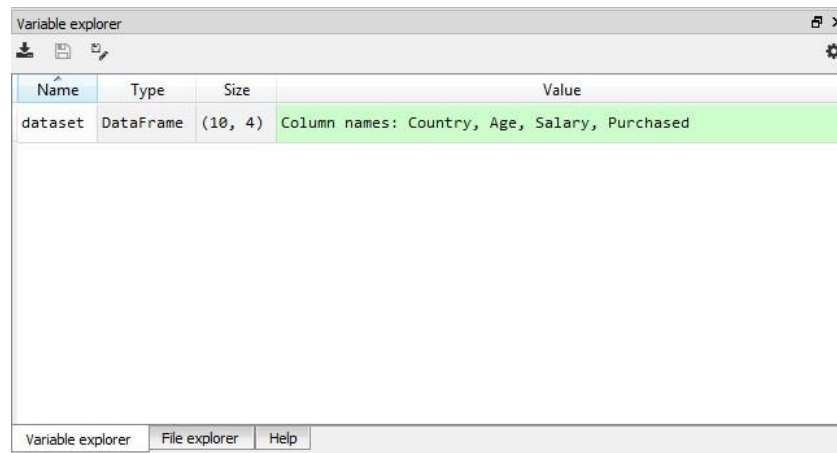
3. Selanjutnya adalah import dataset dengan menambahkan potongan kode, seperti gambar di bawah ini.

```
15 # Importing the Dataset
16 dataset= pd.read_csv('Data.csv')
```

4. Run Cell code tersebut, sehingga akan menampilkan output seperti gambar di bawah ini, yang menunjukkan dataset berhasil diimport.

```
In [5]: dataset= pd.read_csv('Data.csv')
```

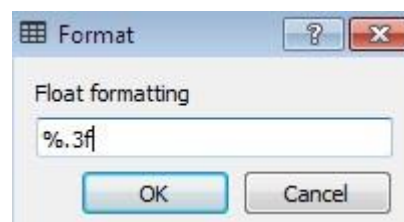
5. Setelah dataset berhasil diimport, pada Variable Explorer akan muncul dataset sebelumnya.



6. Ketika anda double click data tersebut, maka tampilan dataset akan menjadi seperti gambar di bawah ini.

Index	Country	Age	Salary	Purchased
0	France	44	7.2e+04	No
1	Spain	27	4.8e+04	Yes
2	Germany	30	5.4e+04	No
3	Spain	38	6.1e+04	No
4	Germany	40	nan	Yes
5	France	35	5.8e+04	Yes
6	Spain	nan	5.2e+04	No
7	France	48	7.9e+04	Yes
8	Germany	50	8.3e+04	No
9	France	37	6.7e+04	Yes

7. Kita dapat mengubah format dataset pada kolom Salary dengan cara klik Format, dan mengubah g menjadi f (float), dan klik OK, sehingga format angka pada Salary berubah.



8. Langkah selanjutnya adalah membuat matriks untuk ketiga independent variables dan membuat dependent variable factor dengan cara menambahkan lines code seperti gambar di bawah ini.

```
15 # Importing the Dataset
16 dataset= pd.read_csv('Data.csv')
17 X= dataset.iloc[:, :-1].values
```

Pada iloc terdapat dua variabel yang dipisahkan oleh koma. Titik dua (:) disebelah kiri menunjukkan baris dataset dan Titik dua (:) disebelah kanan menunjukkan kolom dataset, yang

berarti kita akan membuat matriks X untuk semua baris dan semua kolom kecuali kolom terakhir, yaitu Purchased. (**Purchased adalah Dependent Variable**).

9. Run Cell code tersebut, sehingga akan menghasilkan output seperti gambar di bawah ini.

```
In [6]: X= dataset.iloc[:, :-1].values
```

10. Ketik X pada console anda, sehingga akan menampilkan nilai dari matriks X.

```
In [7]: X
Out[7]:
array([[ 'France', 44.0, 72000.0],
       [ 'Spain', 27.0, 48000.0],
       [ 'Germany', 30.0, 54000.0],
       [ 'Spain', 38.0, 61000.0],
       [ 'Germany', 40.0, nan],
       [ 'France', 35.0, 58000.0],
       [ 'Spain', nan, 52000.0],
       [ 'France', 48.0, 79000.0],
       [ 'Germany', 50.0, 83000.0],
       [ 'France', 37.0, 67000.0]], dtype=object)
```

11. Kemudian kita akan membuat factor untuk dependent variables, dengan menambahkan kode seperti gambar di bawah ini.

```
15 # Importing the Dataset
16 dataset= pd.read_csv('Data.csv')
17 X= dataset.iloc[:, :-1].values
18 Y= dataset.iloc[:, 3].values
```

Dependent Variables berada pada kolom dengan indeks ketiga.

12. Run Cell code yang baru saja ditambahkan sehingga akan menampilkan output seperti gambar di bawah ini.

```
In [9]: Y
Out[9]:
array([ 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes'],
      dtype=object)
```

MISSING DATA

Masalah yang sering muncul pada model machine learning adalah adanya missing data pada dataset. Salah satu contoh missing data adalah pada file Data.csv ada dua data yang tidak terisi, yaitu Salary pada German dan Age pada Spain. Berikut adalah cara yang dapat digunakan untuk menyelesaikan masalah tersebut.

1. Tambahkan potongan kode seperti gambar di bawah ini.

```
20 # Missing Data
21 # Taking care of missing data
22 from sklearn.preprocessing import Imputer
```

Sklearn terdiri dari banyak library yang dapat digunakan untuk membuat machine learning model. Salah satu library sklearn adalah preprocessing dimana didalamnya terdapat Imputer class.

2. Tambahkan line of codes seperti gambar di bawah ini.

```
20 # Missing Data
21 # Taking care of missing data
22 from sklearn.preprocessing import Imputer
23 imputer= Imputer(missing_values= 'NaN', strategy='mean', axis=0)
24 imputer= imputer.fit(X[:, 1:3])
25 X[:, 1:3]= imputer.transform(X[:, 1:3])
```

Langkah selanjutnya adalah membuat objek imputer dari class Imputer. Untuk melihat parameter dari suatu class anda dapat menggunakan Ctrl + I. Untuk mengisi missing data (data NaN) anda dapat menggunakan fungsi transform yang akan mengembalikan nilai rata- rata dari kolom dimana terdapat sell yang bernilai NaN.

3. Run Cell pada kode tersebut dan ketik X pada console sehingga akan muncul seperti gambar di bawah ini.

```
In [12]: X
Out[12]:
array([[ 'France', 44.0, 72000.0],
       [ 'Spain', 27.0, 48000.0],
       [ 'Germany', 30.0, 54000.0],
       [ 'Spain', 38.0, 61000.0],
       [ 'Germany', 40.0, 63777.77777777778],
       [ 'France', 35.0, 58000.0],
       [ 'Spain', 38.77777777777778, 52000.0],
       [ 'France', 48.0, 79000.0],
       [ 'Germany', 50.0, 83000.0],
       [ 'France', 37.0, 67000.0]], dtype=object)
```

Gambar tersebut menunjukkan bahwa missing data yang tadinya berisi 'Nan', sudah diganti dengan nilai rata- rata. Jika console anda hanya menunjukkan beberapa array, anda dapat menambahkan code seperti gambar di bawah ini.

```
26 np.set_printoptions(threshold=1000)
```

TUGAS:

Ubahlah strategi yang digunakan untuk mengisi data NaN dengan median dan angka yang paling sering digunakan. (Print Screen dan lampirkan pada dokumen ini)

CATEGORICAL DATA

Pada dataset yang kita miliki terdapat dua categorical variable, yaitu Country (France, Spain, dan Germany), serta Purchased (Yes, dan No) yang mengandung beberapa kategori. Model machine learning didasarkan pada persamaan matematika, sehingga categorical variable harus diencode ke dalam bentuk angka (tidak dalam bentuk teks), sehingga persamaan matematika dapat dijalankan.

Berikut adalah langkah yang dapat dilakukan untuk encode categorical variable.

1. Tambahkan kode seperti gambar di bawah ini.

```
29 # Categorical Data
30 # Encoding Categorical Data
31 from sklearn.preprocessing import LabelEncoder
32 labelencoder_X= LabelEncoder()
33 labelencoder_X.fit_transform(X[:, 0])
```

Kali ini kita akan menggunakan kelas LabelEncoder dan membuat objek dengan nama labelencoder_x. Langkah selanjutnya adalah melakukan encode pada matriks X, namun hanya pada kolom yang pertama, yaitu Country.

2. Run Cell pada codes tersebut, sehingga akan menampilkan output seperti gambar di bawah ini, dimana Country telah diencode menjadi angka.

```
In [16]: from sklearn.preprocessing import LabelEncoder
...: labelencoder_X= LabelEncoder()
...: labelencoder_X.fit_transform(X[:, 0])
Out[16]: array([0, 2, 1, 2, 1, 0, 2, 0, 1, 0])
```

3. Langkah selanjutnya adalah mengganti nilai dari matriks X dengan angka yang sudah diencode (jika anda mengetikkan X pada console, data yang diprint adalah data yang belum di-encode), dengan cara menampung nilai encode tersebut kedalam matriks X pada kolom pertama.

```
34 X[:, 0]= labelencoder_X.fit_transform(X[:, 0])
```

4. Run Cell code tersebut sehingga akan menampilkan output seperti gambar di bawah ini.

```
In [17]: X[:, 0]= labelencoder_X.fit_transform(X[:, 0])

In [18]: X
Out[18]:
array([[0, 44.0, 72000.0],
       [2, 27.0, 48000.0],
       [1, 30.0, 54000.0],
       [2, 38.0, 61000.0],
       [1, 40.0, 63777.77777777778],
       [0, 35.0, 58000.0],
       [2, 38.77777777777778, 52000.0],
       [0, 48.0, 79000.0],
       [1, 50.0, 83000.0],
       [0, 37.0, 67000.0]], dtype=object)
```

TUGAS:

Isilah bagian tabel ini dengan nilai encode yang tergenerate.

No	Country	Encoded
1		
2		
3		

5. Ketika anda berhasil melakukan encode, terdapat permasalahan dimana persamaan pada model machine learning dapat membandingkan Spain lebih besar daripada Germany, dan Germany lebih besar daripada France, dimana persamaan ini merupakan persamaan yang salah. Untuk menyelesaikan masalah tersebut, anda membutuhkan Dummy Variables yang dapat digambarkan seperti gambar di bawah ini.

Country		France	Germany	Spain
France		1	0	0
Spain		0	0	1
Germany		0	1	0
Spain		0	0	1
Germany		0	1	0
France		1	0	0
Spain		0	0	1
France		1	0	0
Germany		0	1	0
France		1	0	0

6. Untuk mengaplikasikan solusi ini dapat dilakukan dengan menambahkan kelas dan codes seperti gambar dibawah ini.

```
31 from sklearn.preprocessing import LabelEncoder, OneHotEncoder
32 labelencoder_X= LabelEncoder()
33 labelencoder_X.fit_transform(X[:, 0])
34 X[:, 0]= labelencoder_X.fit_transform(X[:, 0])
35 onehotencoder= OneHotEncoder(categorical_features = [0])
36 X= onehotencoder.fit_transform(X).toarray()
```

TUGAS:

Mengapa `X=onehotencoder.fit_transform(X).toarray()` tidak membuat indeks kolom yang akan diencode seperti `X[:, 0]=labelencoder_X.fit_transform(X[:, 0])`?

7. Run Cell codes yang baru tersebut, sehingga akan menampilkan output seperti gambar di bawah ini.

```
In [22]: X
Out[22]:
array([[1.00000000e+00, 0.00000000e+00, 0.00000000e+00, 4.40000000e+01,
        7.20000000e+04],
       [0.00000000e+00, 0.00000000e+00, 1.00000000e+00, 2.70000000e+01,
        4.80000000e+04],
       [0.00000000e+00, 1.00000000e+00, 0.00000000e+00, 3.00000000e+01,
        5.40000000e+04],
       [0.00000000e+00, 0.00000000e+00, 1.00000000e+00, 3.80000000e+01,
        6.10000000e+04],
       [0.00000000e+00, 1.00000000e+00, 0.00000000e+00, 4.00000000e+01,
        6.37777778e+04],
       [1.00000000e+00, 0.00000000e+00, 0.00000000e+00, 3.50000000e+01,
        5.80000000e+04],
       [0.00000000e+00, 0.00000000e+00, 1.00000000e+00, 3.87777778e+01,
        5.20000000e+04],
       [1.00000000e+00, 0.00000000e+00, 0.00000000e+00, 4.80000000e+01,
        7.90000000e+04],
       [0.00000000e+00, 1.00000000e+00, 0.00000000e+00, 5.00000000e+01,
        8.30000000e+04],
       [1.00000000e+00, 0.00000000e+00, 0.00000000e+00, 3.70000000e+01,
        6.70000000e+04]])
```

8. Untuk melihat perubahan dengan lebih jelas, anda dapat melihat dari Variable Explorer, sehingga akan menampilkan seperti gambar di bawah ini.

X - NumPy array

	0	1	2	3	4
0	1.000	0.000	0.000	44.000	72000.000
1	0.000	0.000	1.000	27.000	48000.000
2	0.000	1.000	0.000	30.000	54000.000
3	0.000	0.000	1.000	38.000	61000.000
4	0.000	1.000	0.000	40.000	63777.778
5	1.000	0.000	0.000	35.000	58000.000
6	0.000	0.000	1.000	38.778	52000.000
7	1.000	0.000	0.000	48.000	79000.000
8	0.000	1.000	0.000	50.000	83000.000
9	1.000	0.000	0.000	37.000	67000.000

Format Resize ☒ Background color

Save and Close Close

TUGAS:

- Langkah selanjutnya adalah melakukan encode untuk Purchase dengan cara menambahkan kode seperti gambar di bawah ini.

```
38 labelencoder_Y= LabelEncoder()  
39 Y= labelencoder_Y.fit_transform(Y)
```

Mengapa kita hanya menggunakan labelEncoder pada Purchased?

- Run Cell kode tersebut, dan ketik Y pada console sehingga menampilkan output seperti gambar di bawah ini.

```
In [24]: Y  
Out[24]: array([0, 1, 0, 0, 1, 1, 0, 1, 0, 1])
```

Splitting The Dataset into Training Set and Test Set

Machine Learning berarti mesin yang belajar dari algoritma didasarkan pada dataset dan korelasi yang ada, sehingga dapat melakukan prediksi. Kita akan membangun machine learning pada dataset (training set) dan kemudian menguji performance model tersebut pada dataset yang lain (test set).

Untuk melakukan split, langkah yang dilakukan adalah sebagai berikut:

1. Ketik kode seperti gambar di bawah ini.

```
41 # Splitting The Dataset into Training Set and Test Set
42 from sklearn.model_selection import train_test_split
```

Import Library model selection dan import class train_test_split, kemudian buat training dan test set yang akan digunakan.

2. Langkah selanjutnya adalah membuat training set dan test set dengan menambah code seperti gambar di bawah ini.

```
41 # Splitting The Dataset into Training Set and Test Set
42 from sklearn.model_selection import train_test_split
43 X_train, X_test, Y_train, Y_test= train_test_split(X, Y, test_size=0.2, random_state=0 )
```

X_train dan Y_train digunakan sebagai training set dan X_test serta Y_test digunakan sebagai test set dimana besar dari test set adalah 20%.

3. Run Cell code tersebut dan lihat perubahan dari Variable Explorer yang akan menampilkan dataset dari x_train, y_train, x_test, dan y_test.

The screenshot shows four variable explorer windows in a Jupyter Notebook. The first window, 'X_test - NumPy array', displays a 2x5 grid of values. The second window, 'Y_test - NumPy array', displays a 2x1 grid of values. The third window, 'X_train - NumPy array', displays an 8x5 grid of values. The fourth window, 'Y_train - NumPy array', displays an 8x1 grid of values.

	0	1	2	3	4
0	0.000	1.000	0.000	30.000	54000.000
1	0.000	1.000	0.000	50.000	83000.000

	0
0	0
1	0

	0	1	2	3	4
0	0.000	1.000	0.000	40.000	63777.778
1	1.000	0.000	0.000	37.000	67000.000
2	0.000	0.000	1.000	27.000	48000.000
3	0.000	0.000	1.000	38.778	52000.000
4	1.000	0.000	0.000	48.000	79000.000
5	0.000	0.000	1.000	38.000	61000.000
6	1.000	0.000	0.000	44.000	72000.000
7	1.000	0.000	0.000	35.000	58000.000

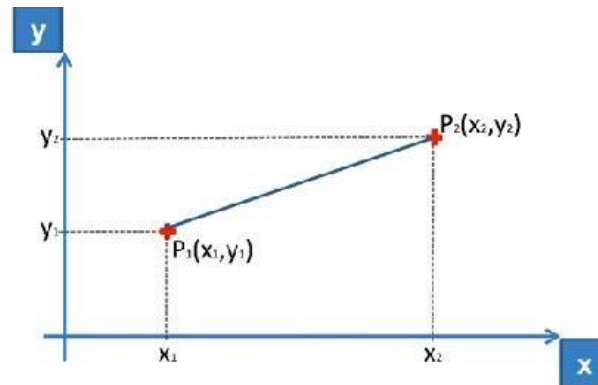
	0
0	1
1	1
2	1
3	0
4	1
5	0
6	0
7	1

TUGAS:

Ganti ukuran training set menjadi 0,8 dan screen capture tampilan pada tugas ini.

FEATURE SCALING

Dataset Data.crv yang kita import sebelumnya memiliki dua kolom yang berisi data numerik. Kolom Age memiliki data dengan rentang dari 27 hingga 50, sementara Salary memiliki rentang nilai antara 48000 hingga 83000. Banyak model machine learning bekerja didasarkan pada Euclidean Distance.



$$\text{Euclidean Distance between } P_1 \text{ and } P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Jika kita menggunakan rumus euclidean distance untuk Age dan Salary, data Salary akan mendominasi, sehingga data Age tidak akan terlihat. Contoh dengan Age misalnya $(48-27)^2=441$ dan Salary $(79000-48000)^2=961000000$, sehingga variable Salary akan mendominasi Age karena tidak berada pada skala yang sama. Dua rumus yang dapat digunakan untuk menyelesaikan masalah tersebut adalah Standardisation dan Normalisation.

Standardisation	Normalisation
$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation}(x)}$	$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$

Berikut adalah langkah yang dapat digunakan untuk scaling.

1. Tambahkan kode program seperti gambar di bawah ini.

```
47 # Feature Scaling
48 from sklearn.preprocessing import StandardScaler
49 sc_X= StandardScaler()
50 X_train= sc_X.fit_transform(X_train)
51 X_test= sc_X.transform(X_test)
```

2. Run Cell kode program tersebut, dan lihat pengaruhnya dari Variable Explorer.

X_test - NumPy array					
	0	1	2	3	4
0	-1.000	2.646	-0.775	-1.459	-0.902
1	-1.000	2.646	-0.775	1.985	2.140

X_train - NumPy array					
	0	1	2	3	4
0	-1.000	2.646	-0.775	0.263	0.124
1	1.000	-0.378	-0.775	-0.254	0.462
2	-1.000	-0.378	1.291	-1.975	-1.531
3	-1.000	-0.378	1.291	0.053	-1.111
4	1.000	-0.378	-0.775	1.641	1.720
5	-1.000	-0.378	1.291	-0.081	-0.168
6	1.000	-0.378	-0.775	0.952	0.986
7	1.000	-0.378	-0.775	-0.598	-0.482

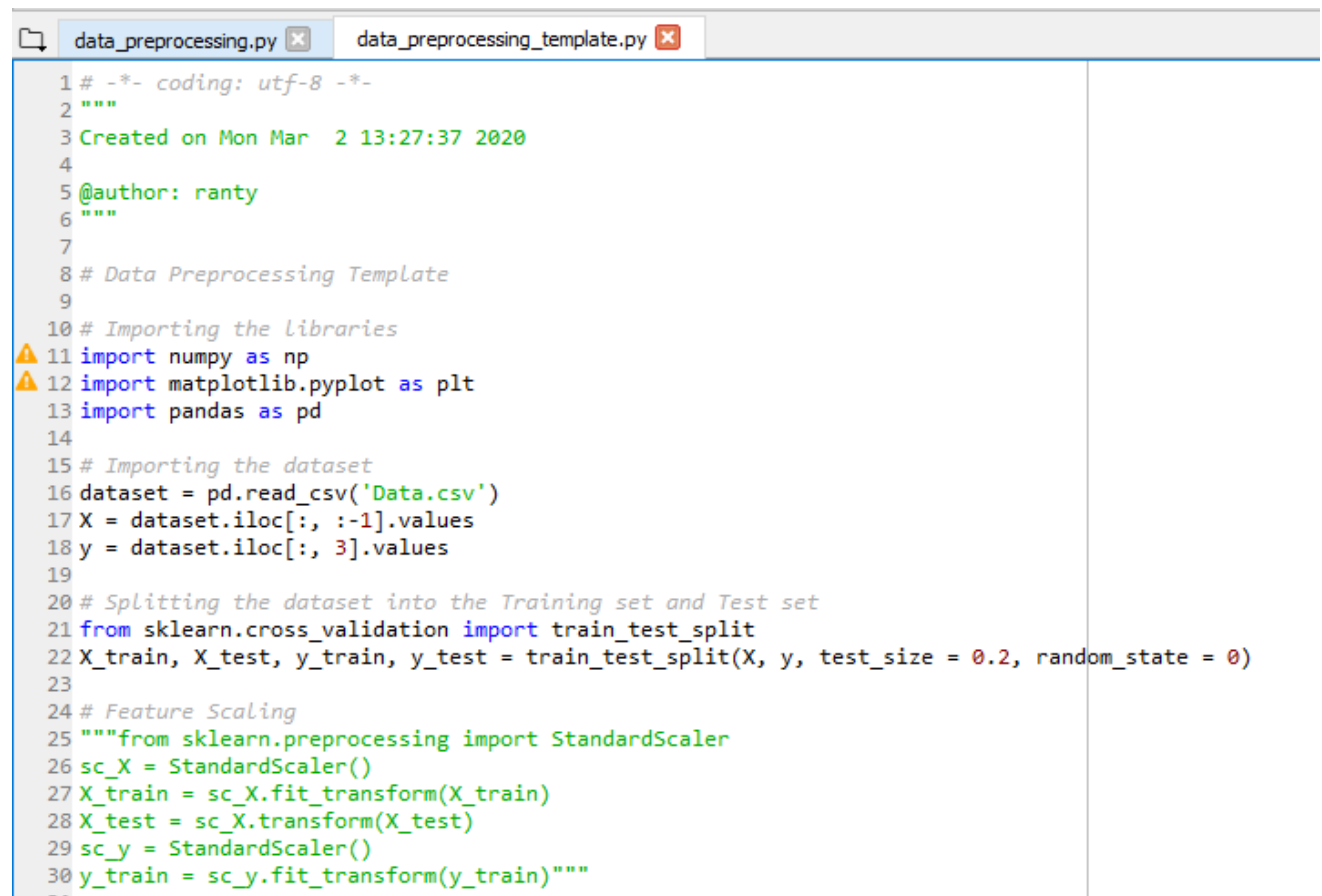
TUGAS:

Apakah kita perlu melakukan scaling terhadap dummy variables?

Apakah kita perlu melakukan scaling terhadap dataset Y_train dan Y_test?

DATA PREPROCESSING TEMPLATE

Template ini akan digunakan untuk tahap selanjutnya sehingga terdapat beberapa fungsi yang tidak disertakan dalam template seperti gambar di bawah ini.



```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Mar  2 13:27:37 2020
4
5 @author: ranty
6 """
7
8 # Data Preprocessing Template
9
10 # Importing the libraries
11 import numpy as np
12 import matplotlib.pyplot as plt
13 import pandas as pd
14
15 # Importing the dataset
16 dataset = pd.read_csv('Data.csv')
17 X = dataset.iloc[:, :-1].values
18 y = dataset.iloc[:, 3].values
19
20 # Splitting the dataset into the Training set and Test set
21 from sklearn.cross_validation import train_test_split
22 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
23
24 # Feature Scaling
25 """from sklearn.preprocessing import StandardScaler
26 sc_X = StandardScaler()
27 X_train = sc_X.fit_transform(X_train)
28 X_test = sc_X.transform(X_test)
29 sc_y = StandardScaler()
30 y_train = sc_y.fit_transform(y_train)"""
```

Template ini akan digunakan untuk setiap pembangunan model machine learning. Jika anda ingin menyimpan code sebelumnya, anda dapat menyimpan di file python lainnya.

Selamat mengerjakan dan have fun😊