```
1 import numpy as np
2 from pandas import DataFrame
3 import pandas as pd
4 df = pd.read_csv('C:/Users/smcom/Desktop/sm_20/python/free/new2/datasets_527325_1205308_PatientI
5 df
```

	patient_id	global_num	sex	birth_year	age	country	province	city
0	1000000001	2.0	male	1964	50s	Korea	Seoul	Gangseo-gu
1	1000000002	5.0	male	1987	30s	Korea	Seoul	Jungnang-gu
2	1000000003	6.0	male	1964	50s	Korea	Seoul	Jongno-gu
3	1000000004	7.0	male	1991	20s	Korea	Seoul	Mapo-gu
4	1000000005	9.0	female	1992	20s	Korea	Seoul	Seongbuk-gu
...	...	...	...	...	...	...	...	...
3999	7000000010	NaN	female	NaN	20s	Korea	Jeju-do	Jeju-do
4000	7000000011	NaN	male	NaN	30s	Korea	Jeju-do	Jeju-do
4001	7000000012	NaN	female	NaN	20s	Korea	Jeju-do	Jeju-do
4002	7000000013	NaN	female	NaN	10s	China	Jeju-do	Jeju-do
4003	7000000014	NaN	female	NaN	30s	Korea	Jeju-do	Jeju-do

4004 rows × 18 columns

```
1
```

```
1 sel = df.loc[:,['sex','city', 'infection_case',
2                 'confirmed_date', 'released_date']]
3 sel2 = df.loc[:,['infection_case', 'birth_year']]
```

```
1 sel['diff_date'] = 100
2 sel2['diff_year'] = 100
```

```
1 sel.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4004 entries, 0 to 4003
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   sex             3674 non-null   object
```

```
sel['confirmed_date'] = sel['confirmed_date'].astype('datetime64[ns]')
sel['released_date'] = sel['released_date'].astype('datetime64[ns]')
```

```
 4   released_date   1508 non-null   object
```

```

sel['diff_date'] =   sel['released_date']-sel['confirmed_date']
sel2['diff_year'] = 100
sel2.dropna(inplace = True)
sel2['birth_year'].dropna()
sel2['birth_year'] = pd.to_numeric(sel2['birth_year'], errors='coerce')
```

```
sel2['birth_year'].count()
sel2
```

	infection_case	birth_year	diff_year
0	overseas inflow	1964.0	100
1	overseas inflow	1987.0	100
2	contact with patient	1964.0	100
3	overseas inflow	1991.0	100
4	contact with patient	1992.0	100
...	...	...	...
3990	etc	1998.0	100
3991	etc	1998.0	100
3992	etc	1972.0	100
3993	etc	1974.0	100
3996	overseas inflow	1996.0	100

2559 rows × 3 columns

```
qwe = sel2['birth_year'] <= 2020.0
asd = sel2['birth_year'] > 2020.0
is_qwe = sel2[qwe]
is_asd = sel2[asd]
is_qwe.diff_year = 2020.0 - is_qwe.birth_year
is_asd.diff_year = 2020.0 - is_asd.birth_year
#df1 = pd.merge(is_qwe , is_asd, left_on = "infection_case", right_index=True)
df1 = pd.concat([is_qwe,is_asd],axis = 1)
df1.columns = ["infection_case","birth_year","diff_year","sd","df","fg"]
df1.drop(["sd","df","fg"], axis='columns', inplace=True)
df1.head(5)
```

```
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩pandas₩core₩generic.py:5303: SettingWithCopyWarnin
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
  self[name] = value
```

	infection_case	birth_year	diff_year
0	overseas inflow	1964.0	56.0
1	overseas inflow	1987.0	33.0
2	contact with patient	1964.0	56.0
3	overseas inflow	1991.0	29.0
4	contact with patient	1992.0	28.0

```
df = pd.concat([sel,sel2,df1],axis = 1 , sort = False)
df.columns = ["sex","city","infection_case","confirmed_date","released_date","diff_date","xxx","
df.drop(["xxx","confirmed_date","released_date","sd","df","qw","birth_year"], axis='columns', in
df
```

	sex	city	infection_case	diff_date	diff_year
0	male	Gangseo-gu	overseas inflow	13 days	56.0
1	male	Jungnang-gu	overseas inflow	32 days	33.0
2	male	Jongno-gu	contact with patient	20 days	56.0
3	male	Mapo-gu	overseas inflow	16 days	29.0
4	female	Seongbuk-gu	contact with patient	24 days	28.0
...	...	...	...	...	...
3999	female	Jeju-do	overseas inflow	18 days	NaN
4000	male	Jeju-do	contact with patient	NaT	NaN
4001	female	Jeju-do	overseas inflow	32 days	NaN
4002	female	Jeju-do	overseas inflow	12 days	NaN
4003	female	Jeju-do	Itaewon Clubs	NaT	NaN

4004 rows × 5 columns

```
df.info()
#결측값 무시하고 진행
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4004 entries, 0 to 4003
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
```

```
df.sex = (df.sex == "female").astype(int)#10l 여성, 00l 남성
df['age'] = 100
df['age'] = np.where(df['diff_year'] >= 0, 0, df['age']) # 0~10
df['age'] = np.where(df['diff_year'] >= 10, 10, df['age']) # 10~20
df['age'] = np.where(df['diff_year'] >= 20 , 20, df['age']) # 20~30
df['age'] = np.where(df['diff_year'] >= 30 , 30, df['age']) # 30~40
df['age'] = np.where(df['diff_year'] >= 40 , 40, df['age']) # 40~all
df.diff_date = df.diff_date / np.timedelta64(1, 'D')
df['date'] = 100
df['date'] = np.where(df['diff_date'] >= 0 , 0, df['date']) # 0~10
df['date'] = np.where(df['diff_date'] >= 10 , 10, df['date']) # 0~10
df['date'] = np.where(df['diff_date'] >= 20 , 20, df['date']) # 10~20
df['date'] = np.where(df['diff_date'] >= 30 , 30, df['date'])  # 20~30
```

```
df = df[['sex','city','infection_case','date','age','diff_date','diff_year']]
dfs = df.dropna()
dfs
```

	sex	city	infection_case	date	age	diff_date	diff_year
0	0	Gangseo-gu	overseas inflow	10	40	13.0	56.0
1	0	Jungnang-gu	overseas inflow	30	30	32.0	33.0
2	0	Jongno-gu	contact with patient	20	40	20.0	56.0
3	0	Mapo-gu	overseas inflow	10	20	16.0	29.0
4	1	Seongbuk-gu	contact with patient	20	20	24.0	28.0
...	...	...	...	...	...	...	...
3990	0	Jeju-do	etc	30	20	32.0	22.0
3991	1	Jeju-do	etc	10	20	14.0	22.0
3992	1	etc	etc	10	40	13.0	48.0
3993	0	Jeju-do	etc	10	40	17.0	46.0
3996	1	Jeju-do	overseas inflow	0	20	9.0	24.0

679 rows × 7 columns

```
#dfs.to_csv('2890df.csv',mode='w')
```

```
aa = pd.read_csv('C:/Users/smcom/Desktop/sm_20/python/free/new2/results.csv',sep=',')
aa
```

	date	sex	age	infection_case_contact with patient	infection_case_etc	infection_case_Eu St. Mary's H
0	20	0	40	1	0	
1	10	1	40	1	0	
2	10	0	20	1	0	
3	20	0	20	0	0	
4	20	0	20	0	0	
...	...	...	...	...	...	
423	10	1	40	1	0	
424	30	1	40	1	0	
425	10	0	40	1	0	
426	20	0	20	0	1	

1 aa

	date	sex	age	infection_case_contact with patient	infection_case_etc	infection_case_Eu St. Mary's H
0	20	0	40	1	0	
1	10	1	40	1	0	
2	10	0	20	1	0	
3	20	0	20	0	0	
4	20	0	20	0	0	
...	...	...	...	...	...	
423	10	1	40	1	0	
424	30	1	40	1	0	
425	10	0	40	1	0	
426	20	0	20	0	1	
427	0	0	20	0	1	

428 rows × 20 columns

서포트 벡터 머신 활용

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.svm import SVC
4 from sklearn.preprocessing import StandardScaler
```

```
from sklearn.preprocessing import StandardScaler
aa.columns
```

```
Index(['date', 'sex', 'age', 'infection_case_contact with patient',
       'infection_case_etc', 'infection_case_Eunpyeong St. Mary's Hospital',
       'infection_case_Geochang Church', 'infection_case_Guro-gu Call Center',
       'infection_case_Gyeongsan Cham Joeun Community Center',
       'infection_case_Gyeongsan Jeil Silver Town',
       'infection_case_Gyeongsan Seorin Nursing Home',
       'infection_case_gym facility in Cheonan',
       'infection_case_Itaewon Clubs', 'infection_case_Milal Shelter',
       'infection_case_Ministry of Oceans and Fisheries',
       'infection_case_overseas inflow', 'infection_case_Pilgrimage to Israel',
       'infection_case_River of Grace Community Church',
       'infection_case_Seongdong-gu APT', 'infection_case_Shincheonji Church'],
      dtype='object')
```

```
x = aa.iloc[:,1:]
x.std = StandardScaler().fit_transform(x)
y = aa['date']
```

```
### Train & test data
x_train, x_test,y_train, y_test = train_test_split(x,y,test_size = 0.4)
```

```
### SVM
svc = SVC(kernel = 'rbf', C = 100, gamma= 0.01)
model = svc.fit(x_train,y_train)
```

```
### 예측
y_pred = model.predict(x_test)
y_pred
```

```
array([20,  0, 30,  0, 10, 20, 30, 20, 10, 30, 10, 10, 20, 10,  0, 10, 10,
       10, 30,  0, 20, 30, 10, 10, 20, 10, 10, 30, 30, 30, 10, 20, 20, 30,
       10, 30, 10, 10, 30, 10, 10,  0, 10,  0, 10, 30, 30, 10, 20, 30, 10,
       20, 20, 20, 30, 10, 10, 10, 10, 10, 30, 10, 10, 30, 10, 10, 10, 10,
       30, 30, 10, 20, 10, 10, 10, 10, 20, 10, 30, 10, 10, 10, 30, 30, 30,
       30,  0, 20, 10, 10, 10, 30, 10, 20, 10, 10, 10, 10, 10, 10, 10, 30,
       10, 10, 10, 10, 10, 30, 30, 10,  0, 10, 10, 30, 10, 30, 10, 10, 20,
       30, 10, 20, 10, 30, 30, 30, 20, 30, 30, 20, 20, 10, 10, 10, 30, 10,
       30, 30, 10, 10, 10, 10, 10, 30, 30, 30, 20, 20, 30, 10, 10, 30, 20,
       20, 30, 20, 10, 30, 10, 20, 10, 10, 30, 10, 10, 30, 10, 10, 30, 10,
       20, 30], dtype=int64)
```

```
### 교차표
pd.crosstab(y_test, y_pred)
```

```
col_0  0  10  20  30

date
```

```
1 from sklearn.metrics import classification_report
2 print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.25      0.18      0.21        11
          10       0.42      0.67      0.51        54
          20       0.46      0.22      0.30        58
          30       0.50      0.51      0.51        49

    accuracy                           0.44       172
   macro avg       0.41      0.40      0.38       172
weighted avg       0.45      0.44      0.42       172
```

```
1 ### acuracy
2 model.score(x_test,y_test)
```

```
0.4418604651162791
```

```
1 #alpha 조정 #매개변수 조정
2 from sklearn.model_selection import GridSearchCV
```

```
1 tuned_parameters = {'solver' : ['lbfgs'],
2                     'alpha' : [0.0001,0.001,0.01,0.1,1],
3                     'hidden_layer_sizes' : [(5,),(10,),(5,5),(10,10),(100,)]}
```

```
1 grid = GridSearchCV(MLPClassifier(),tuned_parameters)
2 %time grid.fit(x_train, y_train)
```

```
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
```

```
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩neural_network₩_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
```

```
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res, self.max_iter)
C:\Users\smcom\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:4
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
1 grid.best_params_
```

```
{'alpha': 0.01, 'hidden_layer_sizes': (100,), 'solver': 'lbfgs'}
```

#로지스틱 회귀분석

```
GridSearchCV(cv=None, error_score=nan,
```

```
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

```
                                        learning_rate='constant',
```

```
x = aa.iloc[:,1:]
y = aa['date']
y,y_levels = pd.factorize(y)
```

```
                                        random_state=None, shuffle=True,
```

```
y_levels
```

```
Int64Index([20, 10, 30, 0], dtype='int64')
```

```
### Train & test data
x_train, x_test,y_train, y_test = train_test_split(x,y,test_size = 0.4)
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
### logistic regression
logistic = LogisticRegression()
model = logistic.fit(x_train, y_train)
```

```
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩linear_model₩_logistic.py:940: Convergence
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

```
### 교차표
pd.crosstab(y_test, y_pred)
```

col_0	10	20	30
row_0			
0	19	33	14
1	12	20	15
2	15	15	18
3	3	4	4

```
### acuracy
model.score(x_test,y_test)
```

```
0.4186046511627907
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	66.0
1	0.00	0.00	0.00	47.0
2	0.00	0.00	0.00	48.0
3	0.00	0.00	0.00	11.0
10	0.00	0.00	0.00	0.0
20	0.00	0.00	0.00	0.0
30	0.00	0.00	0.00	0.0
accuracy			0.00	172.0
macro avg	0.00	0.00	0.00	172.0
weighted avg	0.00	0.00	0.00	172.0

```
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩metrics₩_classification.py:1272: Undefinec
  _warn_prf(average, modifier, msg_start, len(result))
C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩metrics₩_classification.py:1272: Undefinec
  _warn_prf(average, modifier, msg_start, len(result))
```

```
from sklearn.metrics import roc_curve
import matplotlib.pyplot as plt

```

```
### MLP
mlp=MLPClassifier(solver='lbfgs', hidden_layer_sizes=[10,10])
model=mlp.fit(x_train, y_train)
```

```
x = dfs[['age','date']]
y = dfs.sex
y,y_levels = pd.factorize(y)

```

```
x_train, x_test,y_train, y_test = train_test_split(x,y,test_size = 0.4)
```

```
svc = SVC(kernel = 'linear', C = 1)
model = svc.fit(x_train,y_train)
```

```
y_pred = model.predict(x_test)
y_pred
```

```
array([0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
       1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1.
```

```
### 교차표
pd.crosstab(y_test, y_pred)
```

col_0	0	1
row_0		
0	17	110
1	21	124

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.45      0.13      0.21       127
           1       0.53      0.86      0.65       145

    accuracy                           0.52       272
   macro avg       0.49      0.49      0.43       272
weighted avg       0.49      0.52      0.45       272
```

```
### acuracy
model.score(x_test,y_test)
```

```
0.5183823529411765
```

```
x = dfs[['sex','age']]
x.std = StandardScaler().fit_transform(x)
y = dfs['date']
```

```
### Train & test data
x_train, x_test,y_train, y_test = train_test_split(x,y,test_size = 0.4)
```

```
### SVM
svc = SVC(kernel = 'linear', C = 1)
model = svc.fit(x_train,y_train)
```

```
### 예측
y_pred = model.predict(x_test)
y_pred
```

```
array([10, 20, 20, 10, 20, 20, 10, 20, 10, 10, 20, 20, 10, 20, 20, 20, 20,
       20, 20, 20, 20, 10, 20, 20, 20, 20, 10, 20, 20, 20, 20, 20, 20, 10,
       10, 20, 10, 10, 20, 20, 10, 20, 20, 20, 20, 20, 20, 10, 20, 20, 20,
       10, 10, 10, 20, 10, 10, 20, 10, 20, 20, 10, 20, 20, 10, 20, 10, 20,
       20, 20, 20, 20, 10, 10, 10, 20, 10, 10, 20, 20, 20, 20, 20, 20, 20,
       20, 20, 10, 10, 20, 20, 20, 20, 10, 20, 20, 20, 20, 20, 20, 20, 20,
       20, 20, 20, 20, 20, 20, 20, 10, 20, 10, 20, 20, 20, 20, 20, 20, 10,
       20, 20, 20, 20, 20, 20, 10, 20, 20, 10, 10, 10, 20, 20, 20, 20, 10,
       10, 10, 10, 20, 20, 10, 20, 20, 10, 20, 20, 20, 20, 10, 10, 20, 20,
       20, 10, 20, 10, 10, 10, 20, 10, 10, 20, 20, 10, 20, 20, 20, 10, 20,
       10, 20, 10, 20, 10, 10, 10, 20, 20, 20, 20, 20, 20, 20, 20, 20, 10,
       20, 20, 20, 10, 20, 10, 20, 10, 20, 10, 10, 20, 20, 20, 20, 10, 20,
```

```
1 ### 교차표
2 pd.crosstab(y_test, y_pred)
```

col_0	10	20
date		
0	16	12
10	29	64
20	33	52
30	17	49

```
1 from sklearn.metrics import classification_report
2 print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        28
          10       0.31      0.31      0.31        93
          20       0.29      0.61      0.40        85
          30       0.00      0.00      0.00        66

    accuracy                           0.30       272
   macro avg       0.15      0.23      0.18       272
weighted avg       0.20      0.30      0.23       272

C:₩Users₩smcom₩anaconda3₩lib₩site-packages₩sklearn₩metrics₩_classification.py:1272: Undefinec
  _warn_prf(average, modifier, msg_start, len(result))
```

```
1 ### acuracy
2 model.score(x_test,y_test)
```

```
0.2977941176470588
```

```
1 x = dfs[['age','sex','date']]
2 x.std = StandardScaler().fit_transform(x)
3 y = dfs['city']
4 #나이랑 성별 완치기간으로 도시 예측하기
```

```
### Train & test data
x_train, x_test,y_train, y_test = train_test_split(x,y,test_size = 0.4)
```

```
### SVM
svc = SVC(kernel = 'linear', C = 1)
model = svc.fit(x_train,y_train)
```

```
### 예측
y_pred = model.predict(x_test)
y_pred
```

숨겨진 출력 표시

```
### 교차표
pd.crosstab(y_test, y_pred)
```

col_0 city	Cheonan-si	Cheongju-si	Gyeongsan-si	Nam-gu	Yeonsu-gu
Anseong-si	1	0	0	0	0
Anyang-si	1	0	0	0	0
Asan-si	3	0	0	0	0
Bucheon-si	2	0	0	0	0
Buk-gu	4	1	0	0	1
...	...	...	...	...	...
Yecheon-gun	0	0	1	0	0
Yeonsu-gu	3	0	2	0	0
Yeosu-si	1	0	0	0	0
Yongin-si	6	0	0	0	0
etc	2	1	0	0	0

62 rows × 5 columns

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
Anseong-si	0.00	0.00	0.00	1
Anyang-si	0.00	0.00	0.00	1
Asan-si	0.00	0.00	0.00	3
Bucheon-si	0.00	0.00	0.00	2
Buk-gu	0.00	0.00	0.00	6
Bupyeong-gu	0.00	0.00	0.00	5
Buyeo-gun	0.00	0.00	0.00	5
Changwon-si	0.00	0.00	0.00	9
Cheonan-si	0.17	0.82	0.29	44
Cheongju-si	0.00	0.00	0.00	2
Chilgok-gun	0.00	0.00	0.00	1
Chungju-si	0.00	0.00	0.00	4
Danyang-gun	0.00	0.00	0.00	1
Dong-gu	0.00	0.00	0.00	2
Dongdaemun-gu	0.00	0.00	0.00	2
Eumseong-gun	0.00	0.00	0.00	2
Eunpyeong-gu	0.00	0.00	0.00	1
Gangdong-gu	0.00	0.00	0.00	2
Gangnam-gu	0.00	0.00	0.00	3
Geochang-gun	0.00	0.00	0.00	8
Geoje-si	0.00	0.00	0.00	3
Gimcheon-si	0.00	0.00	0.00	6
Gimje-si	0.00	0.00	0.00	1
Goesan-gun	0.00	0.00	0.00	5
Goyang-si	0.00	0.00	0.00	2
Gunsan-si	0.00	0.00	0.00	1
Guro-gu	0.00	0.00	0.00	1
Gwacheon-si	0.00	0.00	0.00	1
Gwanak-gu	0.00	0.00	0.00	1
Gwangju-si	0.00	0.00	0.00	3
Gyeongsan-si	0.52	0.66	0.58	44
Gyeyang-gu	0.00	0.00	0.00	4
Hwasun-gun	0.00	0.00	0.00	1
Icheon-si	0.00	0.00	0.00	4
Jeju-do	0.00	0.00	0.00	3
Jeungpyeong-gun	0.00	0.00	0.00	1
Jongno-gu	0.00	0.00	0.00	5
Jung-gu	0.00	0.00	0.00	5
Michuhol-gu	0.00	0.00	0.00	5
Mokpo-si	0.00	0.00	0.00	3
Muan-gun	0.00	0.00	0.00	2
Nam-gu	0.00	0.00	0.00	8
Namdong-gu	0.00	0.00	0.00	4
Namyangju-si	0.00	0.00	0.00	2
Nonsan-si	0.00	0.00	0.00	1
Sangju-si	0.00	0.00	0.00	6
Seo-gu	0.00	0.00	0.00	4
Seocheon-gun	0.00	0.00	0.00	1
Seocho-gu	0.00	0.00	0.00	1
Seosan-si	0.00	0.00	0.00	6
Siheung-si	0.00	0.00	0.00	3
Songpa-gu	0.00	0.00	0.00	2
Suncheon-si	0.00	0.00	0.00	1
Suwon-si	0.00	0.00	0.00	3
Taean-gun	0.00	0.00	0.00	1
Ulju-gun	0.00	0.00	0.00	3

```
Wonju-si        0.00      0.00      0.00         5
Yecheon-gun     0.00      0.00      0.00         1
```

```
1 ### acuracy
2 model.score(x_test,y_test)
3 #실패
```

0.23897058823529413

```
macro avg       0.01      0.02      0.01       272
```

```
1 dfs.groupby(by = 'city',as_index = False).max()
```

	city	sex	infection_case	date	age	diff_date	diff_year
0	Anseong-si	1	contact with patient	10	40	11.0	68.0
1	Anyang-si	1	etc	0	30	8.0	35.0
2	Asan-si	1	overseas inflow	30	40	69.0	47.0
3	Bonghwa-gun	1	Bonghwa Pureun Nursing Home	30	40	38.0	65.0
4	Bucheon-si	0	overseas inflow	10	40	17.0	49.0
...	...	...	...	...	...	...	...
77	Yecheon-gun	1	etc	30	40	38.0	61.0
78	Yeonsu-gu	1	overseas inflow	30	40	72.0	67.0
79	Yeosu-si	1	overseas inflow	30	20	49.0	25.0
80	Yongin-si	1	etc	20	40	27.0	51.0
81	etc	1	overseas inflow	20	40	24.0	59.0

```
1 from sklearn.model_selection import train_test_split
2 from sklearn.linear_model import LogisticRegression
```

```
1 #### 설명변수(x)와 반응변수(y)
2 x = dfs.iloc[0:,0:3]
3 y = dfs.iloc[0:,4]
4 y,y_levels = pd.factorize(y)
5 #10이 내가 관심있는것의 확률 = 버지니카
6 # 시 데이터는 다음에 다루기로
```

```
1 y
```

```
array([0, 1, 0, 2, 2, 0, 2, 2, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2,
       0, 0, 2, 1, 0, 0, 0, 2, 2, 2, 2, 1, 0, 2, 0, 0, 1, 1, 1, 1, 2, 1,
       0, 0, 0, 1, 1, 2, 2, 0, 2, 0, 1, 0, 0, 2, 0, 0, 0, 0, 3, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 2, 0, 2, 0, 0, 1, 0, 1, 2, 0, 1, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 2, 0, 2, 2, 2, 0, 2, 0, 0, 1, 1, 1, 1, 0, 0, 2,
       2, 0, 0, 0, 0, 2, 2, 1, 0, 0, 1, 0, 0, 0, 3, 0, 0, 2, 1, 0, 0, 0,
       1, 0, 0, 0, 2, 1, 1, 0, 4, 2, 0, 2, 1, 2, 2, 2, 4, 2, 0, 2, 0, 2,
       0, 2, 0, 4, 2, 1, 2, 0, 2, 0, 2, 0, 0, 0, 0, 0, 1, 1, 2, 0, 1, 0,
       1, 1, 2, 0, 2, 0, 2, 2, 2, 4, 2, 2, 2, 2, 0, 0, 2, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 1, 1, 4, 2, 0, 0, 2, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 2,
       2, 1, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 1, 0, 1, 2, 0, 1, 0, 1, 2, 0,
       0, 2, 0, 0, 2, 0, 0, 1, 0, 2, 0, 1, 2, 2, 0, 0, 0, 0, 2, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 0, 1, 2, 1, 2, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 4, 4, 2, 2, 0, 0, 1, 0, 0, 0, 2, 2, 1, 0,
       1, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1,
       0, 0, 0, 0, 4, 1, 2, 0, 0, 0, 0, 2, 0, 4, 4, 0, 4, 4, 1, 0, 4, 3,
       0, 1, 0, 0, 4, 1, 3, 0, 0, 2, 1, 0, 4, 3, 0, 0, 0, 2, 1, 0, 1, 4,
       3, 2, 0, 0, 0, 1, 1, 1, 3, 0, 2, 0, 0, 2, 2, 2, 0, 0, 0, 0, 3, 1,
       0, 2, 0, 1, 0, 0, 0, 2, 0, 1, 0, 0, 0, 0, 1, 1, 1, 4, 0, 1, 0, 0,
       0, 2, 0, 0, 2, 1, 0, 0, 2, 4, 1, 0, 2, 1, 2, 0, 2, 2, 1, 0, 0, 0,
       0, 2, 0, 0, 2, 2, 0, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
1 y_levels
```

```
Int64Index([40, 30, 20, 0, 10], dtype='int64')
```

```
0  1  0  0  0  0  1  1  3  3  0  0  0  0  0  2  0  0  0  0  0  1
```

```
1 ### Train & test data
2 x_train, x_test,y_train, y_test = train_test_split(x,y,test_size = 0.4)
```

```
dtype=int64)
```

```
1 ### logistic regression
2 logistic = LogisticRegression()
3 model = logistic.fit(x_train, y_train)
```

숨겨진 출력 표시

```
1 ### 교차표
2 pd.crosstab(y_test, y_pred)
```

col_0	Cheonan-si	Cheongju-si	Gyeongsan-si	Nam-gu	Yeonsu-gu
row_0					
0	129	3	30	2	0
1	31	2	8	1	0
2	41	1	16	0	1
3	1	0	2	0	0
4	4	0	0	0	0

```
1
```

```
1
```

```
1
```