로지스틱 회귀를 이용한 질병 발생 예측

로지스틱 회귀 : 선형 회귀 방식을 분류에 적용한 알고리즘으로

주로 이진 분류에 사용된다

필요한 모듈을 불러온후 분석에 필요한 데이터를 불러온다

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings(action='ignore')
df = pd.read_csv("val.csv")
df.head(5)
```

```
Bad key "text.kerning_factor" on line 4 in
C:₩Users₩MINWOO₩anaconda3₩lib₩site-packages₩matplotlib₩mpl-data₩stylelib₩_classic_
test_patch.mplstyle.
You probably need to get an updated matplotlibrc file from
https://github.com/matplotlib/matplotlib/blob/v3.1.3/matplotlibrc.template
or from the matplotlib source distribution
```

Out[1]:

	고혈압	이상지질혈증	뇌졸증	심근경색증	협심증	관절염	골관절염	류마티스관절염	골다공증	폐결핵	...	총콜레스테롤	HDL콜레스테롤	중성지방	헤모글러빈	헤마토크리트	혈중요소질소	혈중크레아티닌	백혈구
0	1	1	1	0	0	0	0	0	0	0	...	111.0	43.0	83.0	12.9	38.7	27.0	1.48	8.27
1	0	0	0	0	0	0	0	0	0	0	...	224.0	36.0	157.0	14.3	42.6	22.0	0.95	6.89
2	1	0	0	0	0	1	1	0	0	0	...	165.0	44.0	101.0	14.3	42.3	15.0	0.57	5.67
3	0	0	0	0	0	0	0	0	0	0	...	197.0	51.0	95.0	13.5	39.9	20.0	0.70	8.25
4	0	0	0	0	0	0	0	0	0	0	...	160.0	32.0	229.0	10.3	30.2	23.0	0.98	5.49

5 rows × 61 columns

In [2]:

```
list(df)
```

Out[2]:

```
['고혈압',
 '이상 지질 혈증',
 '뇌졸증',
 '심근경색증',
 '협심증',
 '관절염',
 '골관절염',
 '류마티스 관절염',
 '골다공증',
 '폐결핵',
 '천식',
 '당뇨병',
 '갑상선 질환',
 '위암',
 '간암',
 '유방암',
 '자궁경부암',
 '폐암',
 '갑상선암',
 '우울증',
 '아토피 피부염',
 '알레르기 비염',
 '백내장',
 '녹내장',
 'B형간염',
 'C형간염',
 '신장',
 '체중 ',
 '허리둘레',
 '체질량 지수',
 '1주일간 걷기 일수',
 '하루 앉아서 보내는 시간(평균)',
 '우울감 여부',
 '주중 평균 수면 시간',
 '주말하루 평균 수면 시간',
 '식사량 감소',
 '불안, 우울',
 '통증 불편',
 '일상활동',
 '맥박의 규칙성',
 '15초 맥박수',
 '1차 수축기 혈압',
 '1차 이완기 혈압',
 '2차 수축기 혈압',
 '2차 이완기 혈압',
 '3차 수축기 혈압',
 '3차 이완기 혈압',
 '최종 수축기 혈압',
 '최종 이완기 혈압',
 '공복 혈당',
 '당화혈 색소',
 '총콜레스테롤',
 'HDL콜레스테롤',
 '중성지방',
 '헤모글러빈',
 '헤마토크리트',
 '혈중요소질소',
 '혈중크레아티닌',
 '백혈구',
```

```
 '적혈구',
 '혈소판']
```

고혈압~C형간염까지의 변수들은 환자가 질병의 걸렸는지 안걸렸는지를 나타낸다

질병에 걸리면 1, 걸리지 않으면 0이 입력되어 있다.

신장~혈소판까지의 변수들은 환자 개인 특성의 데이터이다.

분석하기에 앞서 결측치를 찾아내서 제거한다

In [3]:

```
df=df.dropna()
df.isnull().sum()
```

Out[3]:

```
고혈압          0
이상 지질 혈증     0
뇌졸증          0
심근경색증        0
협심증          0
            ..
혈중요소질소       0
혈중크레아티닌      0
백혈구          0
적혈구          0
혈소판          0
Length: 61, dtype: int64
```

고혈압~C형간염까지의 변수들에 대하여 회귀 분석을 진행해야 되기 때문에 필요한 작업들을 함수화한다

로지스틱 회귀분석에서는 데이터 전처리가 필요하여 DataScaler이라는 전처리 함수를 생성하였다.

StandardScaler( )로 평균이 0, 분산 1로 데이터 분포를 변환한다

In [4]:

```
from sklearn.preprocessing import StandardScaler
def DataScaler(df_X):
    scaler = StandardScaler()
    data_scaled = scaler.fit_transform(df_X)
    return data_scaled
```

학습 데이터와 테스트 데이터를 나누는 함수를 생성하였다.

학습 데이터를 80%, 테스트 데이터를 20%로 하였다

In [5]:

```
from sklearn.model_selection import train_test_split
def DataSplit(df_X,df_Y):
    X_train , X_test, Y_train , Y_test = train_test_split(data_scaled, df_Y, test_size=0.2, rand
om_state=0)
    return X_train , X_test, Y_train , Y_test
```

독립변수와 종속변수 즉, X와 Y를 분류한다

각종 질병 진단 여부를 종속 변수로, 나머지 변수들은 독립변수로 둔다.

각각의 질병 발생 예측 분석의 정확도를 알아본다.

그 후 환자 특성에 대한 변수들의 상관계수를 따로 추출하여 csv 파일로 저장한다.

In [6]:

```
from sklearn.metrics import accuracy_score, roc_auc_score
import statsmodels.api as sm
for i in range(26):
    df_Y = df.iloc[:, i]
    df_X = df.drop(df_Y.name,axis=1)
    data_scaled = DataScaler(df_X)
    X_train , X_test, Y_train , Y_test = DataSplit(df_X,df_Y)
    lr_clf = LogisticRegression()
    lr_clf.fit(X_train, Y_train)
    lr_preds = lr_clf.predict(X_test)
    print(df_Y.name,'의 정확도: {:0.3f}'.format(accuracy_score(Y_test, lr_preds)))
    df2 = pd.DataFrame(lr_clf.coef_)
    df2.columns=[df_X]
    df2 = df2.iloc[:,25:]
    df2.index=['{}의 상관계수'.format(df_Y.name)]
    df2.to_csv("test{}.csv".format(i))
```

```
고혈압 의 정확도: 0.861
이상 지질 혈증 의 정확도: 0.840
뇌졸증 의 정확도: 0.973
심근경색증 의 정확도: 0.991
협심증 의 정확도: 0.982
관절염 의 정확도: 1.000
골관절염 의 정확도: 0.998
류마티스 관절염 의 정확도: 0.993
골다공증 의 정확도: 0.930
폐결핵 의 정확도: 0.967
천식 의 정확도: 0.976
당뇨병 의 정확도: 0.952
갑상선 질환 의 정확도: 0.970
위암 의 정확도: 0.996
간암 의 정확도: 0.999
유방암 의 정확도: 0.997
자궁경부암 의 정확도: 0.989
폐암 의 정확도: 0.998
갑상선암 의 정확도: 1.000
우울증 의 정확도: 0.994
아토피 피부염 의 정확도: 0.963
알레르기 비염 의 정확도: 0.954
백내장 의 정확도: 0.840
녹내장 의 정확도: 0.961
B형간염 의 정확도: 0.977
C형간염 의 정확도: 0.982
```

각 상관계수들의 상위 30%만 색깔을 다르게 하였다.

그 중 가장 많이 중복된 변수 중 상위 10개만 선택하여 새로운 환자 개인 특성에 대한 데이터를 만들었다.

선택한 변수들은 아래와 같다.

In [77]:

```
dfNew = pd.read_csv("value_1.csv")
dfNew=dfNew.dropna()
dfNew.iloc[:,26:].head(20)
dfNew.iloc[:,26:].columns
```

Out[77]:

```
Index(['허리둘레', '통증 불편', '주중 평균 수면 시간', '체질량 지수', '3차 수축기
혈압', '헤마토크리트',
       '헤모글러빈', '2차 이완기 혈압', '불안, 우울', '일상활동'],
      dtype='object')
```

선택한 변수들을 독립변수로 설정하여 질병 발생 가능성에 대한 회귀 예측을 하였다.

In [8]:

```
for i in range(26):
    df_Y = dfNew.iloc[:, i]
    df_X = dfNew.iloc[:,26:]
    data_scaled = DataScaler(df_X)
    X_train , X_test, Y_train , Y_test = DataSplit(df_X,df_Y)
    lr_clf = LogisticRegression()
    lr_clf.fit(X_train, Y_train)
    lr_preds = lr_clf.predict(X_test)
    print(df_Y.name,'의 정확도: {:0.3f}'.format(accuracy_score(Y_test, lr_preds)))
```

```
고혈압 의 정확도: 0.812
이상 지질 혈증 의 정확도: 0.809
뇌졸증 의 정확도: 0.976
심근경색증 의 정확도: 0.991
협심증 의 정확도: 0.979
관절염 의 정확도: 0.880
골관절염 의 정확도: 0.888
류마티스 관절염 의 정확도: 0.984
골다공증 의 정확도: 0.922
폐결핵 의 정확도: 0.962
천식 의 정확도: 0.969
당뇨병 의 정확도: 0.911
갑상선 질환 의 정확도: 0.968
위암 의 정확도: 0.992
간암 의 정확도: 1.000
유방암 의 정확도: 0.992
자궁경부암 의 정확도: 0.990
폐암 의 정확도: 0.998
갑상선암 의 정확도: 0.998
우울증 의 정확도: 0.985
아토피 피부염 의 정확도: 0.958
알레르기 비염 의 정확도: 0.964
백내장 의 정확도: 0.860
녹내장 의 정확도: 0.946
B형간염 의 정확도: 0.977
C형간염 의 정확도: 0.982
```

대부분 90% 이상의 정확도를 가지는 예측 모델이 생성되었다.

그 후 환자 가상의 데이터를 생성 후 위 예측모델에 적용하였다.

예측을 위하여 환자 개인 특성에 대한 값 들을 전처리 하였다.

In [78]:

```
df_New = pd.read_csv("현재 환자.csv")
New_data_scaled = DataScaler(df_New.iloc[:,5:])
New_data_scaled = pd.DataFrame(data=New_data_scaled,columns=df_New.iloc[:,5:].columns)
New_data_scaled = pd.merge(df_New.iloc[:,:5],New_data_scaled,how="outer",left_index = True, rig
ht_index = True)
New_data_scaled.head(5)
```

Out[78]:

	Patient ID	Patient Name	Name of disease	Sex	Age	허리둘레	통증 불편	주중 평균 수면 시간	체질량 지수	3차 수축기 혈압
0	1	강성훈	NaN	0	5	-1.553195	-2.236068	1.633066	0.222121	0.608430
1	2	권아현	NaN	1	26	-0.090352	0.000000	0.165141	-0.248845	0.446182
2	3	김가영	NaN	1	34	0.055932	0.000000	0.532123	-0.511661	-1.014051
3	4	김동현	NaN	0	12	-0.658279	0.000000	0.776777	-0.938184	-0.202810
4	5	김정우	NaN	0	21	-0.357106	0.000000	-0.935802	-1.312383	0.121686

위 로지스틱 회귀를 이용하여 한 명의 환자의 질병 발생 가능성을 예측하였다.

예시로 '석민규'환자의 질병 발생 가능성을 예측하였다.

In [74]:

```
for i in range(26):
    df_Y = dfNew.iloc[:, i]
    df_X = dfNew.iloc[:,26:]
    data_scaled = DataScaler(df_X)
    X_train , X_test, Y_train , Y_test = DataSplit(df_X,df_Y)
    lr_clf = LogisticRegression()
    lr_clf.fit(X_train, Y_train)
    df_New_X = New_data_scaled.loc[New_data_scaled["Patient Name"] == '석민규']
    df_New_X = df_New_X.iloc[:,5:]
    print("{}의 발생 가능성".format(df_Y.name),lr_clf.predict(df_New_X))
```

```
고혈압의 발생 가능성 [0]
이상 지질 혈증의 발생 가능성 [1]
뇌졸증의 발생 가능성 [0]
심근경색증의 발생 가능성 [0]
협심증의 발생 가능성 [0]
관절염의 발생 가능성 [1]
골관절염의 발생 가능성 [0]
류마티스 관절염의 발생 가능성 [0]
골다공증의 발생 가능성 [0]
폐결핵의 발생 가능성 [0]
천식의 발생 가능성 [0]
당뇨병의 발생 가능성 [0]
갑상선 질환의 발생 가능성 [0]
위암의 발생 가능성 [0]
간암의 발생 가능성 [0]
유방암의 발생 가능성 [0]
자궁경부암의 발생 가능성 [0]
폐암의 발생 가능성 [0]
갑상선암의 발생 가능성 [0]
우울증의 발생 가능성 [0]
아토피 피부염의 발생 가능성 [0]
알레르기 비염의 발생 가능성 [0]
백내장의 발생 가능성 [0]
녹내장의 발생 가능성 [0]
B형간염의 발생 가능성 [0]
C형간염의 발생 가능성 [0]
```

숫자가 0인 경우 질병이 발생하지 않을 확률이 더 큰 것이고,

숫자가 1인 경우 질병이 발생할 확률이 더 큰 것이다.

각종 질병의 발생할 확률을 알아보았다.

In [75]:

```
for i in range(26):
    df_Y = dfNew.iloc[:, i]
    df_X = dfNew.iloc[:,26:]
    data_scaled = DataScaler(df_X)
    X_train , X_test, Y_train , Y_test = DataSplit(df_X,df_Y)
    lr_clf = LogisticRegression()
    lr_clf.fit(X_train, Y_train)
    df_New_X = New_data_scaled.loc[New_data_scaled["Patient Name"] == '석민규']
    df_New_X = df_New_X.iloc[:,5:]
    print("{}이 발생할 확률".format(df_Y.name))
    print("{0:.2f}".format(lr_clf.predict_proba(df_New_X)[0,1]*100),"%")
```

```
고혈압이 발생할 확률
34.65 %
이상 지질 혈증이 발생할 확률
57.50 %
뇌졸증이 발생할 확률
2.12 %
심근경색증이 발생할 확률
4.93 %
협심증이 발생할 확률
2.76 %
관절염이 발생할 확률
56.53 %
골관절염이 발생할 확률
42.68 %
류마티스 관절염이 발생할 확률
15.53 %
골다공증이 발생할 확률
13.69 %
폐결핵이 발생할 확률
5.92 %
천식이 발생할 확률
18.61 %
당뇨병이 발생할 확률
23.87 %
갑상선 질환이 발생할 확률
9.00 %
위암이 발생할 확률
0.84 %
간암이 발생할 확률
0.33 %
유방암이 발생할 확률
0.79 %
자궁경부암이 발생할 확률
0.36 %
폐암이 발생할 확률
0.06 %
갑상선암이 발생할 확률
0.28 %
우울증이 발생할 확률
1.14 %
아토피 피부염이 발생할 확률
14.17 %
알레르기 비염이 발생할 확률
13.11 %
백내장이 발생할 확률
30.54 %
녹내장이 발생할 확률
27.80 %
B형간염이 발생할 확률
9.46 %
C형간염이 발생할 확률
4.92 %
```

일주일 후의 환자의 데이터를 불러와 '석민규' 환자의 질병 발생 확률을 다시 예측한다.

In [76]:

```
df_New = pd.read_csv("현재 환자_일주일후.csv")
New_data_scaled = DataScaler(df_New.iloc[:,5:])
New_data_scaled = pd.DataFrame(data=New_data_scaled,columns=df_New.iloc[:,5:].columns)
New_data_scaled = pd.merge(df_New.iloc[:,:5],New_data_scaled,how="outer",left_index = True, rig
ht_index = True)
for i in range(26):
    df_Y = dfNew.iloc[:, i]
    df_X = dfNew.iloc[:,26:]
    data_scaled = DataScaler(df_X)
    X_train , X_test, Y_train , Y_test = DataSplit(df_X,df_Y)
    lr_clf = LogisticRegression()
    lr_clf.fit(X_train, Y_train)
    df_New_X = New_data_scaled.loc[New_data_scaled["Patient Name"] == '석민규']
    df_New_X = df_New_X.iloc[:,5:]
    print("{}이 발생할 확률".format(df_Y.name))
    print("{0:.2f}".format(lr_clf.predict_proba(df_New_X)[0,1]*100),"%")
```

고혈압이 발생할 확률
25.52 %
이상 지질 혈증이 발생할 확률
49.56 %
뇌졸증이 발생할 확률
1.58 %
심근경색증이 발생할 확률
3.53 %
협심증이 발생할 확률
1.94 %
관절염이 발생할 확률
51.37 %
골관절염이 발생할 확률
36.83 %
류마티스 관절염이 발생할 확률
15.76 %
골다공증이 발생할 확률
13.11 %
폐결핵이 발생할 확률
5.97 %
천식이 발생할 확률
17.26 %
당뇨병이 발생할 확률
16.30 %
갑상선 질환이 발생할 확률
8.82 %
위암이 발생할 확률
0.74 %
간암이 발생할 확률
0.21 %
유방암이 발생할 확률
0.67 %
자궁경부암이 발생할 확률
0.35 %
폐암이 발생할 확률
0.05 %
갑상선암이 발생할 확률
0.19 %
우울증이 발생할 확률
1.07 %
아토피 피부염이 발생할 확률
12.84 %
알레르기 비염이 발생할 확률
14.17 %
백내장이 발생할 확률
32.02 %
녹내장이 발생할 확률
24.66 %
B형간염이 발생할 확률
7.99 %
C형간염이 발생할 확률
4.20 %