Data Analytics Week 12 Assignment

202011431 산업공학과 차승현

Classification

Analysis Procedure

1) Data 및 Module Import

In [55]:
```
from sklearn.model_selection import train_test_split
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
```

In [56]:
```
df = pd.read_csv(r'data_week12.csv')
df
```

Out[56]:

	Number of times pregnant.	Plasma glucose concentration a 2 hours in an oral glucose tolerance test.	Diastolic blood pressure (mm Hg).	Triceps skinfold thickness (mm).	2-Hour serum insulin (mu U/ml).	Body mass index (weight in kg/(height in m)^2).	Diabetes pedigree function.	Age (years).	Class variable (0 or 1).
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [57]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                                                                    Non-Null Count  Dtype
---  ------                                                                    --------------  -----
 0   Number of times pregnant.                                                 768 non-null    int64
 1   Plasma glucose concentration a 2 hours in an oral glucose tolerance test. 768 non-null    int64
 2   Diastolic blood pressure (mm Hg).                                         768 non-null    int64
 3   Triceps skinfold thickness (mm).                                          768 non-null    int64
 4   2-Hour serum insulin (mu U/ml).                                           768 non-null    int64
 5   Body mass index (weight in kg/(height in m)^2).                           768 non-null    float64
 6   Diabetes pedigree function.                                               768 non-null    float64
 7   Age (years).                                                              768 non-null    int64
 8   Class variable (0 or 1).                                                  768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [58]:
```
df.isnull().sum()
```

Out[58]:
```
Number of times pregnant.                                                 0
Plasma glucose concentration a 2 hours in an oral glucose tolerance test. 0
Diastolic blood pressure (mm Hg).                                         0
Triceps skinfold thickness (mm).                                          0
2-Hour serum insulin (mu U/ml).                                           0
Body mass index (weight in kg/(height in m)^2).                           0
Diabetes pedigree function.                                               0
Age (years).                                                              0
Class variable (0 or 1).                                                  0
dtype: int64
```

데이터를 불러와 데이터의 결측치의 개수, 정보를 확인하였고 문제는 없었다.
데이터는 총 8개의 feature(X)와 하나의 label(Y)로 이루어져 있다.
Train, test데이터로 split하고 데이터를 스케일링 처리하였다.

In [59]:
```
features = df.iloc[:, :-1]
features
```

Out [59]:

	Number of times pregnant.	Plasma glucose concentration a 2 hours in an oral glucose tolerance test.	Diastolic blood pressure (mm Hg).	Triceps skinfold thickness (mm).	2-Hour serum insulin (mu U/ml).	Body mass index (weight in kg/(height in m)^2).	Diabetes pedigree function.	Age (years).
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33
...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63
764	2	122	70	27	0	36.8	0.340	27
765	5	121	72	23	112	26.2	0.245	30
766	1	126	60	0	0	30.1	0.349	47
767	1	93	70	31	0	30.4	0.315	23

768 rows × 8 columns

In [60]:
```
classes = df.iloc[:, -1]
classes
```

Out [60]:
```
0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
766    1
767    0
Name: Class variable (0 or 1)  Length: 768, dtype: int64
```

In [61]:
```
X_train, X_test, y_train, y_test = train_test_split(features, classes, test_size = 0.2, random_state=42)
```

In [62]:
```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)
```

Train과 test의 비율을 0.2로 설정하였다.

1. Naïve Bayes Classifier

```
In [65]: nb = GaussianNB()
         fitted = nb.fit(X_train, y_train)
         y_pred = fitted.predict(X_train)

In [68]: confusion_matrix(y_train, y_pred)

Out[68]: array([[338,  63],
                [ 89, 124]], dtype=int64)

In [71]: a = accuracy_score(y_train, y_pred)
         print("accuracy: ", a)
         p = precision_score(y_train, y_pred)
         print("precision: ", p)
         r = recall_score(y_train, y_pred)
         print("recall: ", r)
         f1 = f1_score(y_train, y_pred)
         print("f1-score: ",f1)

         accuracy:  0.752442996742671
         precision:  0.6631016042780749
         recall:  0.5821596244131455
         f1-score:  0.62
```

Confusion matrix에 의해 오차행렬은 위와 같이 나타났고, 정확도는 75.24%로 확인되었다.

2. Logistic Regression

Logistic Regression

```
In [72]: lr = LogisticRegression()
         fitted2 = lr.fit(X_train, y_train)
         y_pred2 = fitted2.predict(X_train)

In [73]: confusion_matrix(y_train, y_pred2)

Out[73]: array([[354,  47],
                [ 94, 119]], dtype=int64)

In [74]: a = accuracy_score(y_train, y_pred2)
         print("accuracy: ", a)
         p = precision_score(y_train, y_pred2)
         print("precision: ", p)
         r = recall_score(y_train, y_pred2)
         print("recall: ", r)
         f1 = f1_score(y_train, y_pred2)
         print("f1-score: ",f1)

         accuracy:  0.7703583061889251
         precision:  0.7168674698795181
         recall:  0.5586854460093896
         f1-score:  0.6279683377308708
```

Confusion matrix에 의해 오차행렬은 위와 같이 나타났고, 정확도는 77.036%로 확인되었다.