Introduction to Big Data

Lecture 11. Classification

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import numpy as np
import pandas as pd
import seaborn as sns
```

Classification with Personal Loan Data

- Experience
- Income
- Famliy
- CCAvg: Average monthly card spent
- Education: Education level (1: undergrad; 2, Graduate; 3; Advance)
- Mortgage
- Personal Loan: Personal Loan (1:Yes, 0:No)
- Securities account: Securities (1:Yes, 0:No)
- CD account: CD account (1:Yes, 0:No)
- Online: Online account (1:Yes, 0:No)
- CreidtCard: Credit Card (1:Yes, 0:No)

```
PerLoan = pd.read_csv("[Directory]/personalLoan.csv")
PerLoan.head()
```

```
PerLoan.shape
```

```
PerLoan.columns
```

```
PerLoan.rename(columns={'Personal Loan':'PersonalLoan'}, inplace=True)
```

```
PerLoan.columns
```

```
PerLoan.describe()
```

```
# check missing values
PerLoan.isnull().any()
```

```
PerLoan.count()
```

```
PL_X = PerLoan[['Age','CCAvg','Income','Education']]
PL_Y = PerLoan['PersonalLoan']
```

Logit Regression with statsmodels

```
from statsmodels.formula.api import logit

statsLogitModel = logit('PersonalLoan ~ Age + CCAvg + Income + Education',data=PerLoan)
statsLogitModel
```

```
statsLogitModel_res = statsLogitModel.fit()
```

```
print(statsLogitModel_res.summary())
```

```
statsLogitModel_res.params
```

```
np.exp(statsLogitModel_res.params)
```

Logit Regression with sklearn

Whole sample

```
from sklearn.linear_model import LogisticRegression

LogitModel0 = LogisticRegression()
```

```
LogitModel0_res = LogitModel0.fit(PL_X, PL_Y)
LogitModel0_res
```

```
LogitModel0_res.coef_
```

```
LogitModel0_res.intercept_
```

Test and train sample

```
from sklearn.model_selection import train_test_split

PL_X_train, PL_X_test, PL_Y_train, PL_Y_test = train_test_split(PL_X, PL_Y, test_size=0.3,
                                                                random_state=0)
```

```
# Practice of Random Sampling
PL_X_train1, PL_X_test1, PL_Y_train1, PL_Y_test1 = train_test_split(PL_X, PL_Y,
                                                                    test_size=0.3)
PL_X_train1.head()
```

```
PL_X_train1, PL_X_test1, PL_Y_train1, PL_Y_test1 = train_test_split(PL_X, PL_Y,
                                                                    test_size=0.3,
                                                                    random_state=1)
PL_X_train1.head()
```

```
PL_X_train1, PL_X_test1, PL_Y_train1, PL_Y_test1 = train_test_split(PL_X, PL_Y,
                                                                    test_size=0.3,
                                                                    random_state=1)
PL_X_train1.head()
```

```
PL_X_train
```

```
PL_X_test
```

```
PL_Y_train
```

```
PL_Y_test
```

```
from sklearn.linear_model import LogisticRegression

LogitModel = LogisticRegression()
```

```
LogitModel.fit(PL_X_train, PL_Y_train)
```

```
LogitModel.coef_
```

```
LogitModel.intercept_
```

▾ Validation

```
PL_Y_pred = LogitModel.predict(PL_X)
```

```
PL_Y_train_pred = LogitModel.predict(PL_X_train)
```

```
PL_Y_test_pred = LogitModel.predict(PL_X_test)
```

```
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score, confusion_matrix
```

▾ Accuracy Score

```
accuracy_score(PL_Y_test, PL_Y_test_pred)
```

▾ Recall Score

```
recall_score(PL_Y_test, PL_Y_test_pred)
```

▾ Precision Score

```
precision_score(PL_Y_test, PL_Y_test_pred)
```

▾ Specificity

```
tn, fp, fn, tp = confusion_matrix(PL_Y_test, PL_Y_test_pred).ravel()
specificity = tn / (tn+fp)
specificity
```

✓ 0초 오전 9:36에 완료됨