2024 Fall
20190741 김지수

인공지능과 마케팅 코딩과제1

: 당뇨 환자 분류

Preparing the Dataset

Import library

```
import pandas as pd
import numpy as np

#시각화
import matplotlib.pyplot as plt
import seaborn as sns

# 데이터 처리
from sklearn.preprocessing import MinMaxScaler
from imblearn.over_sampling import SMOTE # handling imbalanced data

# 머신러닝
from sklearn.model_selection import train_test_split,GridSearchCV, cross_val_score

# 분류모델
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

# 6. Metrics
from sklearn.metrics import accuracy_score, confusion_matrix, recall_score,
f1_score

# 7. Ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

Preparing the Dataset

Load Dataset

```
# 데이터 읽기
data = pd.read_csv("/content/data-Lab-2-5-diabet.csv", header = None,
names=['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin',
'BMI','DiabetesPedigreeFunction','Age','Outcome'])
# 헤더가 없는 엑셀 파일이라 header = None으로 표시하여 불러온다. 열 이름을 헷갈리지 않게 지정


# 데이터 앞부분 예시 표시
data.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	-0.294118	0.487437	0.180328	-0.292929	0.000000	0.001490	-0.531170	-0.033333	0
1	-0.882353	-0.145729	0.081967	-0.414141	0.000000	-0.207153	-0.766866	-0.666667	1
2	-0.058824	0.839196	0.049180	0.000000	0.000000	-0.305514	-0.492741	-0.633333	0
3	-0.882353	-0.105528	0.081967	-0.535354	-0.777778	-0.162444	-0.923997	0.000000	1
4	0.000000	0.376884	-0.344262	-0.292929	-0.602837	0.284650	0.887276	-0.600000	0

Split Dataset

```
# 데이터셋 분리 - 학습 데이터셋과 테스트 데이터셋
x = data.drop('Outcome',axis=1)
y = data['Outcome']
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_size=0.15, stratify=y)

# 재현성을 위해 랜덤 스테이트를 42로 설정하고, test 케이스 사이즈는 전체 데이터의 15%로 설정함
```

1. 85% 학습 데이터, 15% 테스트 데이터

2. 재현성을 위한 random state 설정

Preparing the Dataset

Smote 사용

```
y_train.value_counts()
x_train, y_train = smote.fit_resample(x_train,y_train)
```

	count
Outcome	
1	422
0	223

dtype: int64

1. 전체 데이터가 양성(1)이 422, 음성(0) 223개로 불균형

: SMOTE 방법을 사용하여 학습에 사용하는 데이터에서 선형적으로 늘려 음성과 양성을 같은 개수만큼 학습에 사용. 단순 복제하는 것이 아니라 기존 소수 클래스 데이터 포인트 사이에 새로운 데이터를 만들어서, 데이터를 확장하는 방식이다.

-> 데이터 셋의 개수가 800개로 매우 작아서 불균형하게 학습 데이터셋이 생성되어 실제 테스트에서 정확도가 낮아지는 문제를 방지하기 위해서 사용

사용한 모델

1. 전통적인 로지스틱 회귀모형

2. SVM(서포트 벡터 머신)
분류 알고리즘 중에 하나로 분류율이 좋은 알고리즘. 특히 이진 데이터(binary data) 분류에 특화되어 있음

3. 랜덤 포레스트 모형
여러 개의 의사결정 나무(Decision Tree)를 결합하여 예측을 수행하는 앙상블 학습 방법

4. XG boost
그래디언트 부스팅(이전 모델의 오류를 보완하는 방식으로 모델을 계속 추가하여 예측 성능을 향상)을 기반으로 한 고성능 앙상블 학습 방법

Train the model

모델 학습

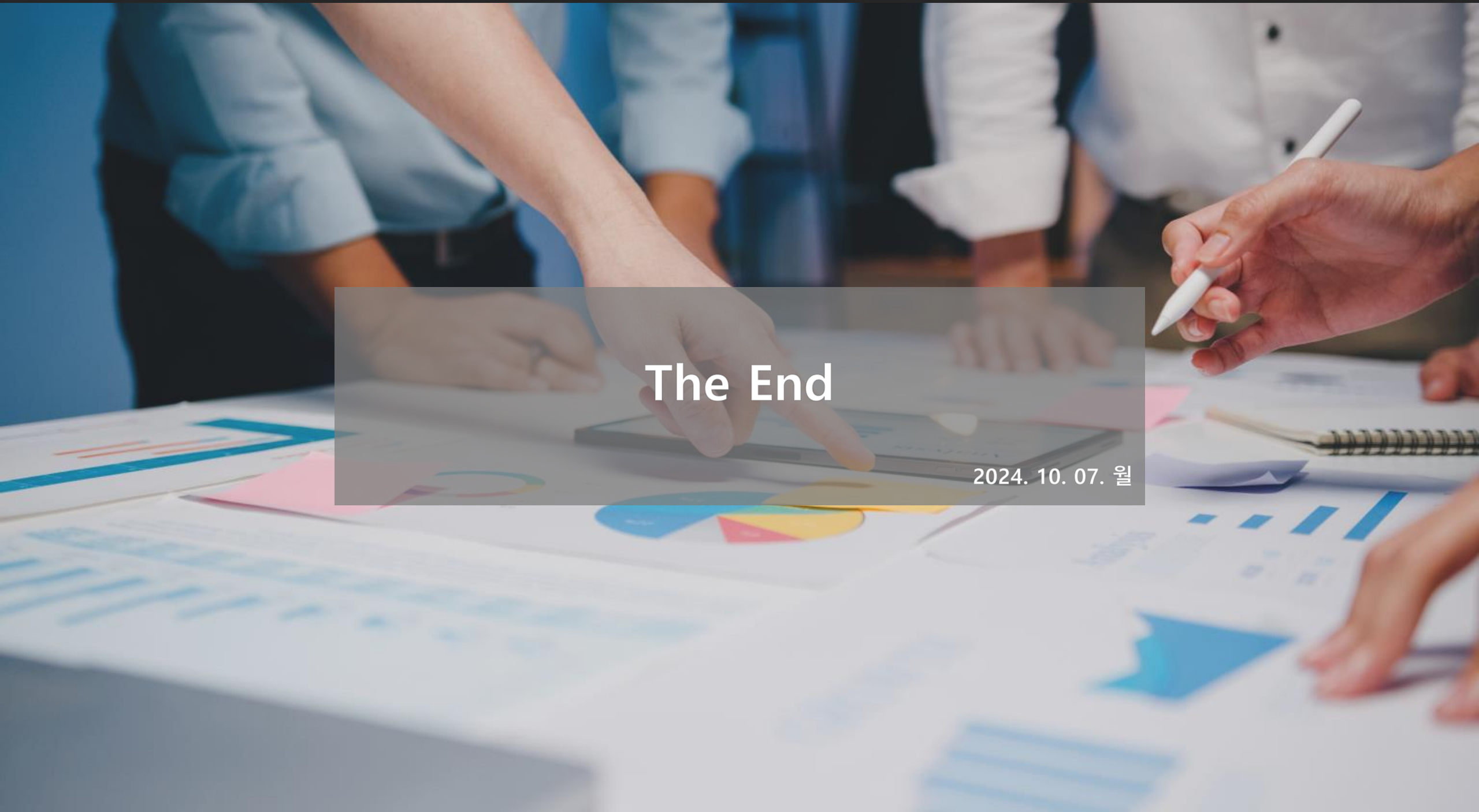
```
alg = ['LogisticRegression','SVC','RandomForestClassifier','XGBoostClaassifier']
acc = []
rec= []
F1 = []

def evaluate(model):
  model.fit(x_train,y_train)   #모델 피팅
  pre = model.predict(x_test)  #모델 예측값 생성
  accuracy = accuracy_score(pre,y_test) #모델 정확도
  recall = recall_score(pre,y_test)
#Recall (모델 재현율): 실제로 양성인 것들 중에서 모델이 양성으로 예측한 비율
(의학적 진단에서 양성인 사람을 양성으로 진단하는 것이 중요)

  acc.append(accuracy)
  rec.append(recall)

  sns.heatmap(confusion_matrix(pre,y_test),annot=True)
  print(model)
  print('Accuracy : ',accuracy,'Recall : ',recall,)
```

Evaluate the Dataset

모델 평가

```
evaluate(model_LR)
evaluate(model_SVM)
evaluate(model_RFC)
evaluate(model_XGB)
final_result = pd.DataFrame({"Algorithm":alg ,'Accuarcy':acc,"recall":rec})
```

	Algorithm	Accuarcy	recall
0	LogisticRegression	0.754386	0.828571
1	SVC	0.763158	0.861538
2	RandomForestClassifier	0.710526	0.773333
3	XGBoostClaassifier	0.736842	0.805556



The End

2024. 10. 07. 월