

Python으로 머신러닝 입문



세션장: 구은아, 김은기



EDA

EDA란?

Exploratory Data Analysis

탐색적 데이터 분석

EDA에서 확인해야 하는 것

1

데이터 크기
및
변수 확인

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

#	Column	Non-Null Count	Dtype
0	PassengerId	891 non-null	int64
1	Survived	891 non-null	int64
2	Pclass	891 non-null	int64
3	Name	891 non-null	object
4	Sex	891 non-null	object
5	Age	714 non-null	float64
6	SibSp	891 non-null	int64
7	Parch	891 non-null	int64
8	Ticket	891 non-null	object
9	Fare	891 non-null	float64
10	Cabin	204 non-null	object
11	Embarked	889 non-null	object

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

EDA에서 확인해야 하는 것

2

결측값,
이상치 등
확인

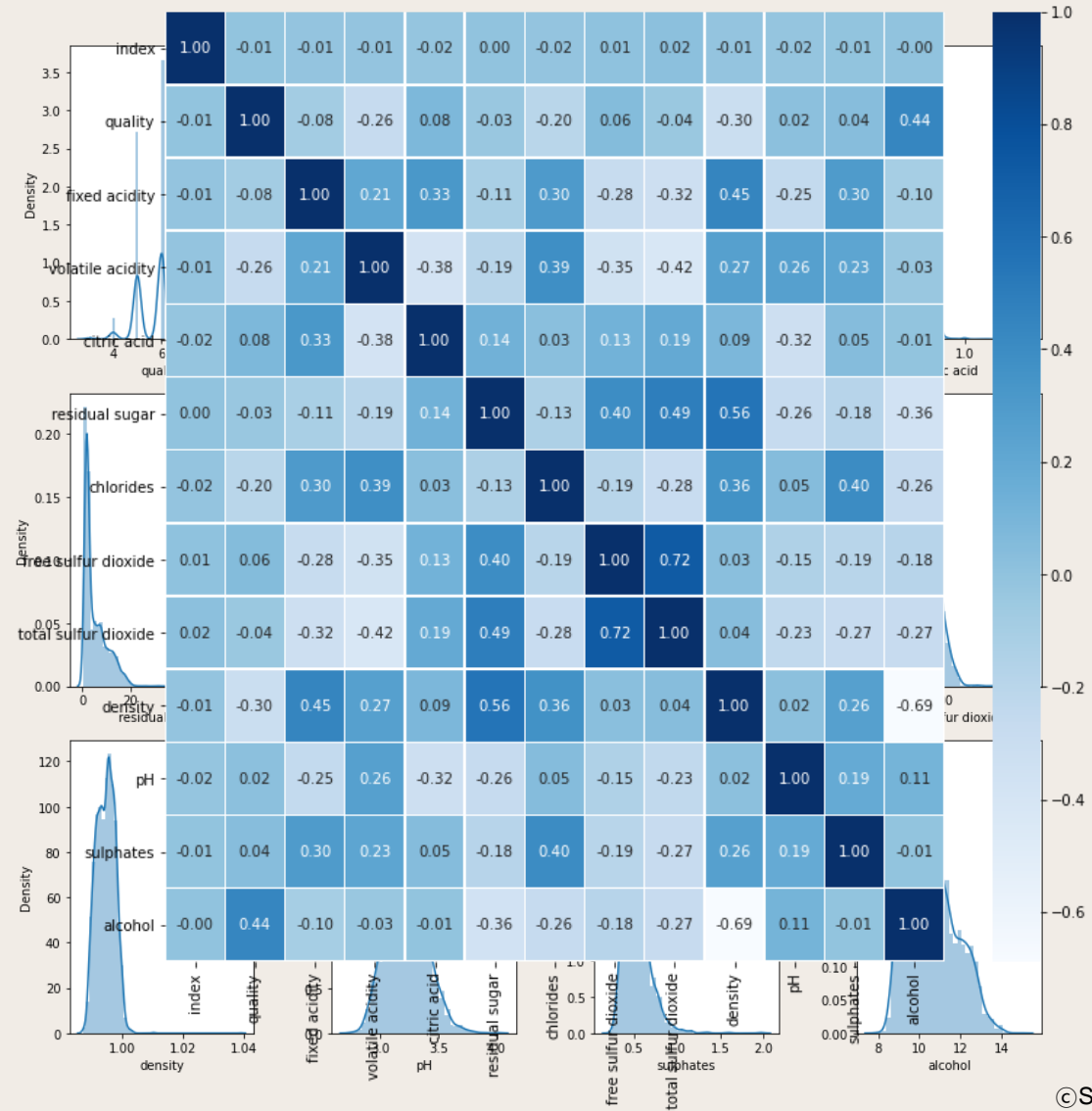
```
df.isnull().sum()
```

```
] PassengerId    0  
   Survived      0  
   Pclass       0  
   Name         0  
   Sex          0  
   Age         177  
   SibSp        0  
   Parch        0  
   Ticket       0  
   Fare         0  
   Cabin       687  
   Embarked     2  
   FamilySize   0  
dtype: int64
```

EDA에서 확인해야 하는 것

3

개별 변수
및
변수 간 관계
관측



타이타닉 생존자 예측

평가지표 파이썬 실습

파이썬 패키지

정확도

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# 원본 데이터를 재로딩, 데이터 가공, 학습데이터/테스트 데이터 분할.
titanic_df = pd.read_csv('./titanic_train.csv')
y_titanic_df = titanic_df['Survived']
X_titanic_df = titanic_df.drop('Survived', axis=1)
X_titanic_df = transform_features(X_titanic_df)
X_train, X_test, y_train, y_test = train_test_split(X_titanic_df, y_titanic_df, \
                                                    test_size=0.2, random_state=0)

# 위에서 생성한 Dummy Classifier를 이용하여 학습/예측/평가 수행.
myclf = MyDummyClassifier()
myclf.fit(X_train, y_train)

mypredictions = myclf.predict(X_test)
print('Dummy Classifier의 정확도는: {0:.4f}'.format(accuracy_score(y_test, mypredictions)))
```

파이썬 패키지

교차행렬

```
from sklearn.metrics import confusion_matrix
```

```
# 앞절의 예측 결과인 fakepred와 실제 결과인 y_test의 Confusion Matrix출력  
confusion_matrix(y_test , fakepred)
```

정밀도 재현율

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

```
def get_clf_eval(y_test , pred):  
    confusion = confusion_matrix( y_test, pred)  
    accuracy = accuracy_score(y_test , pred)  
    precision = precision_score(y_test , pred)  
    recall = recall_score(y_test , pred)  
    print('오차 행렬')  
    print(confusion)  
    print('정확도: {0:.4f}, 정밀도: {1:.4f}, 재현율: {2:.4f}'.format(accuracy , precision ,recall))
```

파이썬 패키지

F1스코어

```
from sklearn.metrics import f1_score
f1 = f1_score(y_test, pred)
print('F1 스코어: {0:.4f}'.format(f1))
```

ROC커브

```
from sklearn.metrics import roc_curve
```

```
# 레이블 값이 1일때의 예측 확률을 추출
```

```
pred_proba_class1 = lr_clf.predict_proba(X_test)[: , 1]
```

```
fprs , tprs , thresholds = roc_curve(y_test, pred_proba_class1)
```

```
# 반환된 임계값 배열 로우가 47건이므로 샘플도 10건만 추출하되, 임계값을 5 Step으로 추출.
```

```
thr_index = np.arange(0, thresholds.shape[0], 5)
```

```
print('샘플 추출을 위한 임계값 배열의 index 10개:', thr_index)
```

```
print('샘플용 10개의 임계값: ', np.round(thresholds[thr_index], 2))
```

```
# 5 step 단위로 추출된 임계값에 따른 FPR, TPR 값
```

```
print('샘플 임계값별 FPR: ', np.round(fprs[thr_index], 3))
```

```
print('샘플 임계값별 TPR: ', np.round(tprs[thr_index], 3))
```

파이썬 패키지

AUC

```
from sklearn.metrics import roc_auc_score
```

```
pred = lr_clf.predict(X_test)  
roc_score = roc_auc_score(y_test, pred)  
print('ROC AUC 값: {:.4f}'.format(roc_score))
```

과제 안내

과제 제출 방법



Korea University Computer Club

고려대학교 컴퓨터동아리 공식 Github Organization 입니다

Seoul, Korea <https://kucc.co.kr>

1. kucc git에 접속
<https://github.com/kucc>

Overview Repositories 41 Packages People 95 Teams 4 Projects

Pinned

Archive

Public

KUCC 세션 혹은 스터디의 기록물을 남기는 아카이브

Repositories

Find a repository...

Type

Language

Sort

New

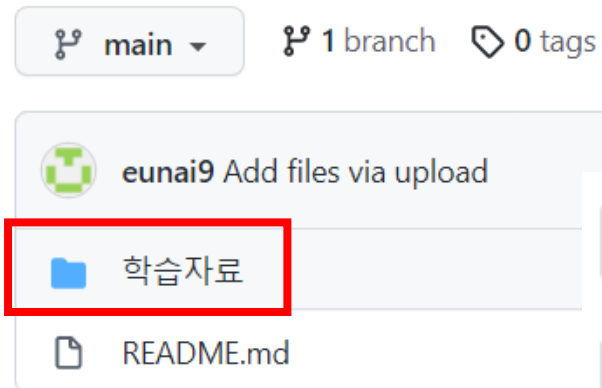
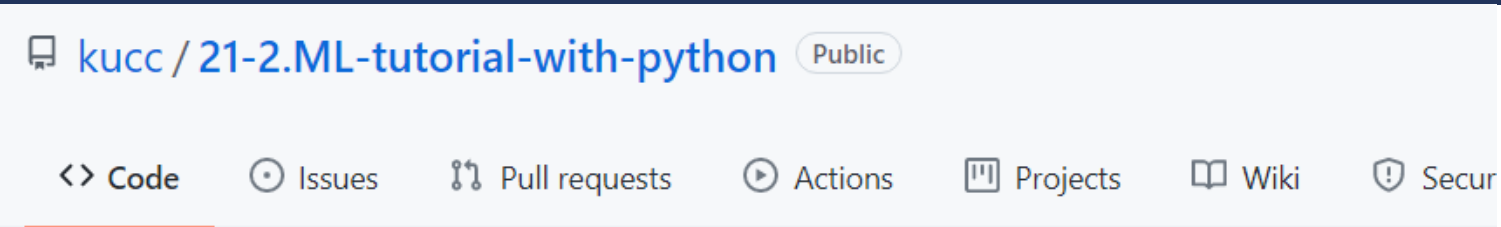
21-2.ML-tutorial-with-python

21-2 파이썬으로 머신러닝 입문 세션

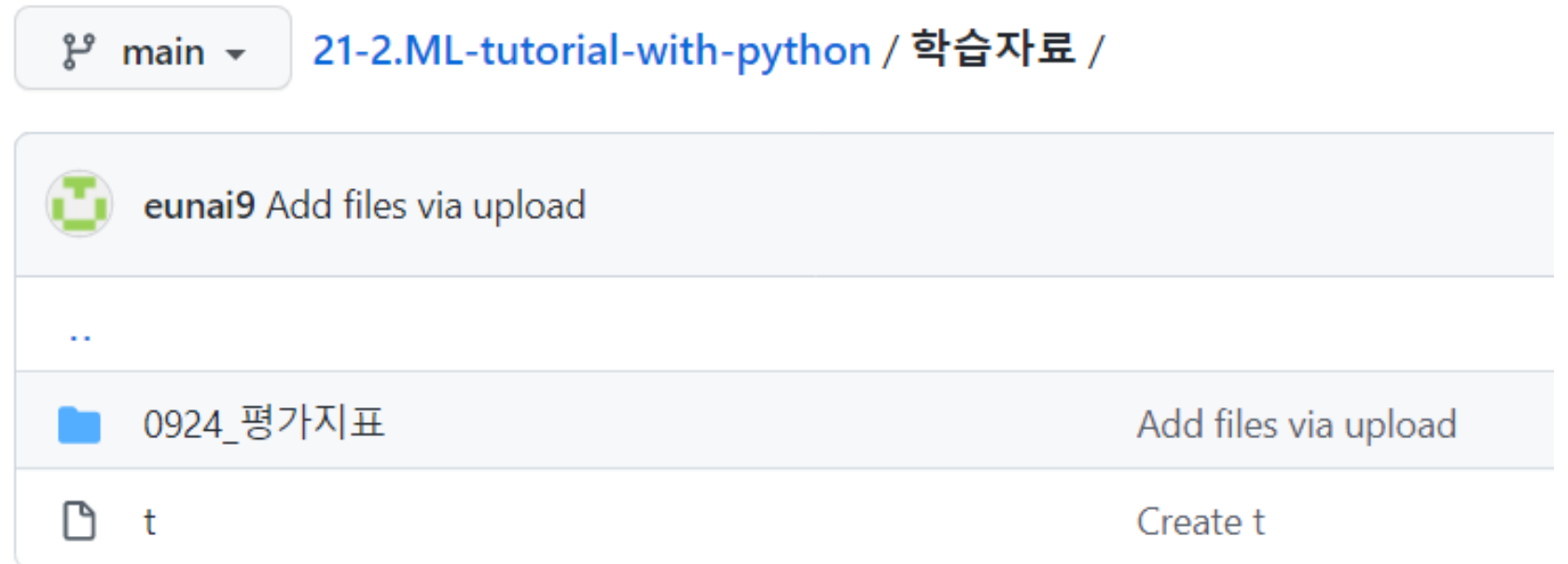
2. 왼쪽과 같은 이름을 가진 레포를 찾아 클릭
<https://github.com/kucc/21-2.ML-tutorial-with-python>

Jupyter Notebook 0 0 0 0 Updated 20 seconds ago

과제 제출 방법



학습자료 폴더에는 앞으로 학습자료 폴더에 수업에서 사용된 ppt, 실습파일, 과제가 올라올 예정입니다.



과제 제출 방법

The screenshot illustrates the steps to create a new file and commit it on GitHub. Red boxes and numbers highlight the key actions:

- 1**: Clicking the **Add file** button in the top right corner, which opens a dropdown menu with the option **Create new file**.
- 2**: Clicking the **Create new file** option, which leads to the file creation page. The path `21-2.ML-tutorial-with-python / 000 / t` is shown in the breadcrumb, with the folder `000` and the file name `t` highlighted.
- 3**: Clicking the **Commit new file** button in the bottom right corner of the commit dialog.

The commit dialog also shows the file content: `this is a folder of 000`, which is also highlighted with a red box. The dialog includes options to commit directly to the `main` branch or create a new branch.

과제 제출 방법



main ▾

21-2.ML-tutorial-with-python / 학습자료 / 0924_평가지표 /



eunai9 Add files via upload

..



0924_과제.ipynb

다운받기!

Add files via upload



diabetes.csv

Add files via upload



t

Create t

과제: 피마 인디언 당뇨병 예측

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

diabetes_data = pd.read_csv('diabetes.csv')
diabetes_data.head()
```

```
# 피쳐 데이터 세트 X, 레이블 데이터 세트 y를 추출.
# 맨 끝이 Outcome 컬럼으로 레이블 값임. 컬럼 위치 -1을 이용해 추출
X = diabetes_data.iloc[:, :-1]
y = diabetes_data.iloc[:, -1]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 156, stratify=y)
```

```
# 로지스틱 회귀로 학습, 예측 및 평가 수행.
lr_clf = LogisticRegression()
lr_clf.fit(X_train, y_train)
pred = lr_clf.predict(X_test)
```

diabetes.csv 데이터를 로지스틱 회귀로 예측한 샘플이 0924_과제.ipynb 파일에 있습니다.

- 1) 이를 이용하여 diabetes 데이터를 각자 자유롭게 EDA 해보고, 필요한 경우 전처리를 하여 다시 예측해봅시다.
- 2) 그리고 예측한 데이터를 이용해 정확도, 교차행렬, 정밀도, 재현율, F1스코어, ROC curve, AUC를 구해봅시다.

수고하셨습니다!
과제 열심히 하시고 다음 주에 보어요~