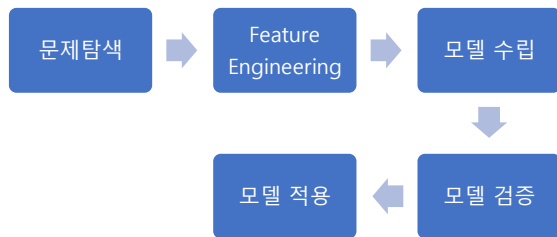


WEEK6 진짜로 해보는 실전 분석 프로젝트

두 가지 프로젝트 -> 1. 숫자 손글씨 인식 프로젝트

2. 와인 품질 측정 프로젝트

실전 프로젝트 단계 준비:



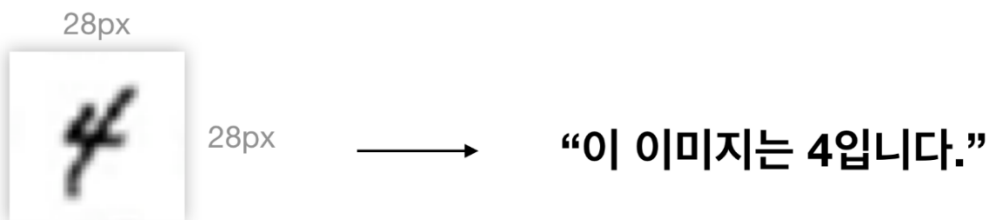
1. 숫자 손글씨 인식 프로젝트 설명(1)

OCR 숫자 인식기: 사진, 스캔 등 글자의 픽셀 이미지로부터 기계가 읽을 수 있는 문자로 변환하는 작업.

1) 문제 정의 (28 * 28 손글씨 숫자 이미지 입력받아 숫자 인식)

2) 가설 수립 (특징 데이터를 구성한 후 머신러닝을 통해 실제로 어떤 숫자인지 추측)

3) 목표 (28*28 사이즈의 숫자 손글씨 이미지로 부터 label값을 얻어내기)

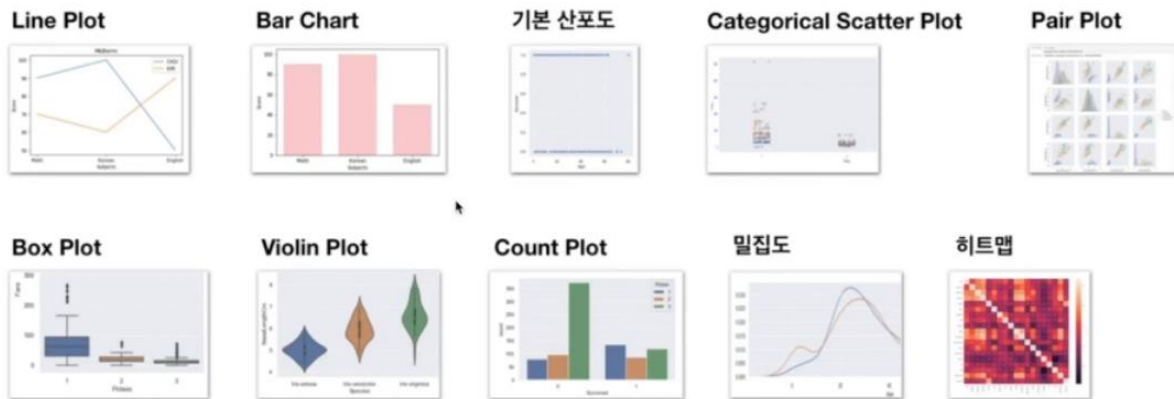


2. 숫자 손글씨 인식 프로젝트 설명(2)

- **EDA (탐색적 자료 분석)** : 특성에 숨겨진 패턴을 찾아내고 구조를 이해할 수 있도록 단서를 찾아 동분서주하는 방식의 분석.

→ 시각화 기법 등으로 더 깊은 구조를 파악하려는 방법

그동안 배운 시각화 기법



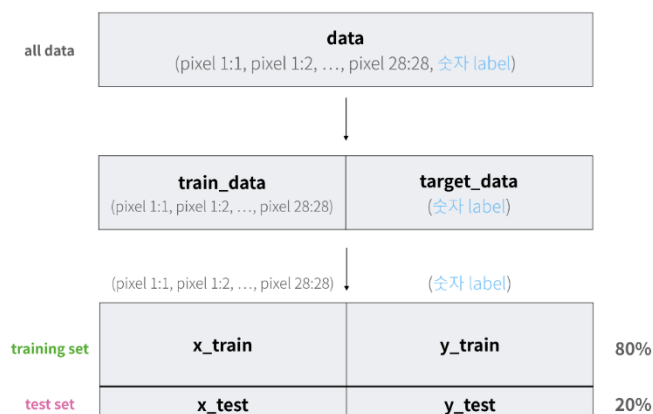
최소 요구 기준: 1) label(숫자 종류 0~9)별로 몇 개의 데이터가 있는지 시각화하기

2) 픽셀 특징이 가지고 있는 최솟값, 최댓값 파악하기 (**describe**같은 것 이용)

3) 이미지 살펴보기.

4) 데이터셋 구성

데이터셋 구성 최소 요구사항



● 모델링

- 분류 문제? 회귀 문제?
- 분류에 효과적인 모델 찾아보기
- 모델 RUNNING 해보기

● 최종검증

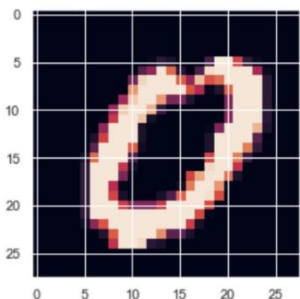
- 모델의 score 구하기. 최고 성능 모델 선택
- test set 이용해 선택한 모델의 최종 성능 평가
- 결론 도출 : 000모델을 통해 000의 정확도로 숫자 손글씨를 OCR할 수 있다.

```
In [15]: # random으로 픽하기
import random
for i in range(4):
    n = random.randrange(0, len(x_test))

    img = np.reshape(x_test.iloc[n].values, [28, 28])
    plt.imshow(img)
    plt.show()

    result = forest.predict([x_test.iloc[n].values])[0]
    print("인식된 숫자는", result, "입니다.")
```

executed in 560ms, finished 11:44:38 2019-03-14



인식된 숫자는 0 입니다.

작업자마다 변수 이름이 다를 수 있는 부분

1. 문제를 이해합니다.
이미지의 각 픽셀을 특징 데이터로 취급하여 어떤 숫자인지 인식하는 문제입니다.
2. EDA 및 Feature Engineering을 실시합니다.
어떤 식으로 이미지의 픽셀을 특징 데이터로 구성하였는지 파악합니다.
3. 가설 검증 계획을 수립합니다.
classification을 다루는 머신러닝 모델링
4. 데이터셋을 구성합니다.
상황과 목적에 따라 적절한 train set, (validation set), test set을 구성합니다.
5. 모델링하고 학습합니다.
DecisionTreeClassifier, RandomForestClassifier, ...
6. 모델을 평가하고 검증합니다.
7. 최종 결론을 도출합니다.
000모델을 통해 000의 정확도로 숫자 손글씨를 OCR할 수 있다.

<숫자 손글씨 인식 프로젝트 과정 총정리>

3. 와인 품질 측정 프로젝트 설명(1)

dataset: 산성도, 도수, 설탕 잔량 등 11가지 입력 정보와 그에 대응되는 퀄리티 값을 가진 1,599개 데이터

- 1) 문제정의 (와인의 화학 측정 데이터로부터 기존에는 미각, 후각으로 측정하던 와인의 품질을 추정)
- 2) 가설수립 (화학데이터로 특징 데이터를 구성 후 머신러닝을 통해 미각 측정 없이 와인의 품질을 추정 가능하다)
- 3) 목표 (화학 특징의 데이터를 입력받아 0~10 사이의 숫자로 와인 품질을 추정하기)

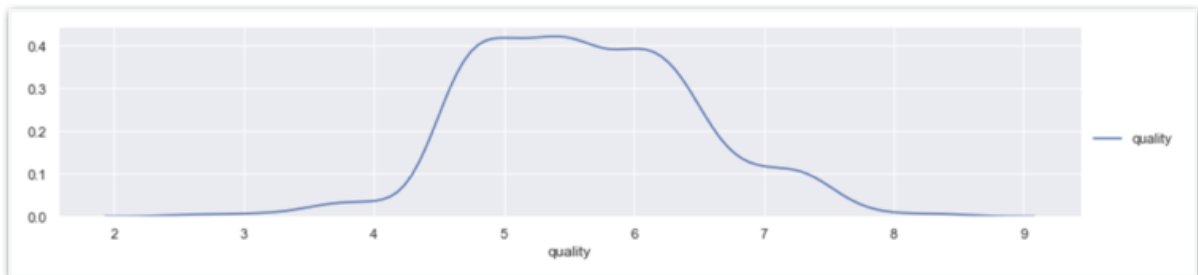
volatile acidity	citric acid	residual sugar
5.99.000000	1599.000000	1599.000
.527821	0.270976	2.538806

→ “이 와인의 품질은 5.31입니다.”

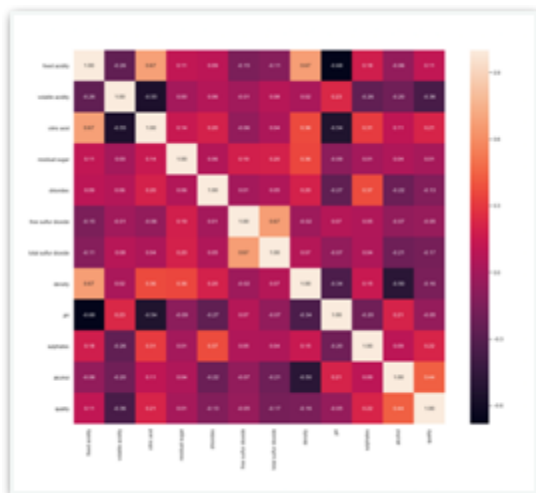
4. 와인 품질 측정 프로젝트 설명(2)

최소 요구 기준:

1) 와인 품질의 밀집도 파악하기



2) 특징 상관계수 히트맵



- 모델링 및 최종 결론 등은 앞의 숫자 인식 프로젝트와 비슷함

결론 : 000모델을 통해 000의 정확도로 와인의 품질을 추정할 수 있다.

5. 숫자 손글씨 인식 프로젝트 실전 코드

```
#1. 데이터 불러오기
import numpy as np
import pandas as pd
df = pd.read_csv('data/digit.csv')
df.head()

#2. EDA & Feature Engineering
import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
sns.set()

df.describe()

#label 에 따른 빈도수 살펴보기
sns.catplot(data=df, x='label', kind='count')

#이미지 살펴보기
numbers = df.drop(['label'], axis=1)

nth = 0 # 0 ~ 9999 까지 바꾸면서 확인
img = np.reshape(numbers.iloc[nth].values, [28, 28])
plt.imshow(img)
plt.show()

#3. Dataset 구성하기
train_data = df.drop('label', axis=1)
target_data = df['label']

print(train_data.shape, target_data.shape)

#train data 와 test data 나누기
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(train_data, target_data, test_size=0.2)

print(train_data.shape, x_train.shape, x_test.shape)

#4. 모델링과 학습
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=100)

#train 데이터 학습
forest.fit(x_train, y_train)

#정확도 확인
print('training set accuracy:', forest.score(x_train, y_train))

#5. 모델 검증
print('test set accuracy:', forest.score(x_test, y_test))
```

```

        #실제 예측 결과물 살펴보기
# random 으로 픽하기
import random

for i in range(4):
    n = random.randrange(0, len(x_test))

    img = np.reshape(x_test.iloc[n].values, [28, 28])
    plt.imshow(img)
    plt.show()

    result = forest.predict([x_test.iloc[n].values])[0]
    print("인식된 숫자는", result, "입니다.")

```

6. 와인 품질 측정 프로젝트 실전 코드

```

#1. 먼저 데이터 불러오기
#11 개의 특징데이터 개수 1599 개 출력데이터 quality 와인 품질

import pandas as pd
import numpy as np
df = pd.read_csv('data/wine.csv')
df.head()

df.describe()

#2. EDA & Feature Engineering
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set()

        #밀집도 확인해보기. 보아뱀모양
facet = sns.FacetGrid(df, aspect = 4)
facet.map(sns.kdeplot, 'quality')
facet.add_legend()
plt.show()

        #히트맵
plt.figure(figsize=(20, 15))
sns.heatmap(df.corr(), annot=True, fmt='.2f', square=True)
plt.show()

#3. Dataset 구성하기
input_data = df.drop(['quality'], axis=1)
target_data = df['quality']

```

```

print(input_data.shape, target_data.shape)

#train data 와 test 데이터 나누기
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(input_data, target_data, test_size=0.2)
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)

#4. 모델링 & 학습
#랜덤포레스트
from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor(n_estimators=100)

forest.fit(x_train, y_train)
print('training set accuracy: ', forest.score(x_train, y_train))

#5. 모델 검증
print('test set accuracy: ', forest.score(x_test, y_test))

#실제 값과 내 예측값 비교
y_predict = forest.predict(x_test)
comparision = pd.DataFrame(y_test)
comparision['내 예측'] = y_predict

comparision

```