실습 1: 언어 판별 프로그램

- 실습 목표
  - 외국어 글자를 읽어 들이고 어떤 언어인지 판별
  - 학습데이터 파일에 20개의 데이터, 8개는 테스트용

```
from sklearn import svm, metrics
import glob, os.path, re, json
# 텍스트를 읽어 들이고 출현 빈도 조사하기 --- (①)
Def check_freq(fname):
    name = os.path.basename(fname)
    lang = re.match(r ' ^[a-z]{2,} ' , name).group()
    with open(fname,  " r " , encoding= " utf-8 " ) as f:
        text = f.read()
    text = text.lower() # 소문자 변환
    # 숫자 세기 변수(cnt) 초기화하기
    cnt = [0 for n in range(0, 26)]
    code_a = ord( " a " )
    code_z = ord( " z " )
    # 알파벳 출현 횟수 구하기 --- (②)
    for ch in text:
        n = ord(ch)
        if code_a <= n <= code_z: # a~z 사이에 있을 때
            cnt[n - code_a] += 1
```

텍스트 파일을 읽기
알파벳의 출현 빈도를 조사

문자열의 처음부터 정규식과 매치되는지 조사 후 앞에 두 글자만 따서 저장한다.

a부터 z까지의 출현 빈도수 조사
*** 알파벳 이외의 글자는 무시**

실습 1: 언어 판별 프로그램


```
# 정규화하기 ---( ③)
    total = sum(cnt)
    freq = list(map(lambda n: n / total, cnt))
    return (freq, lang)

# 각 파일 처리하기
def load_files(path):
    freqs = []
    labels = []
    file_list = glob.glob(path)
    for fname in file_list:
        r = check_freq(fname)
        freqs.append(r[0]) # 각 기사의 freq추가
        labels.append(r[1]) # 각 기사의 lang추가
    return {"freqs":freqs, "labels":labels}
data = load_files("./lang/train/*.txt")
test = load_files("./lang/test/*.txt")
```

전체 카운팅된 문자의 개수를 각 문자의 개수로 나누어 리스트로 생성함

정규화 : 데이터를 일정한 규칙을 기반으로 변형해서 쉽게 사용할 수 있게 하는 것

실습 1: 언어 판별 프로그램


```
# 이후를 대비해서 JSON으로 결과 저장하기
with open("./lang/freq.json", "w", encoding="utf-8") as fp:
    json.dump([data, test], fp)
# 학습하기 --- (④)
clf = svm.SVC()
clf.fit(data[ " freqs " ], data[ " labels " ])
# 예측하기 --- (⑤)
predict = clf.predict(test[ " freqs " ])
# 결과 테스트하기 --- (⑥)
ac_score = metrics.accuracy_score(test["labels"], predict)
cl_report = metrics.classification_report(test["labels"],
predict)
print("정답률 =", ac_score)
print("리포트 =")
print(cl_report)
```

SVM(SVC)알고리즘을 학습

테스트 전용 데이터를 사용하여 예측한다.

예측한 결과를 기반으로 정답을 출력

실습2:데이터 마다 분포를 그래프로 확인


```
import matplotlib.pyplot as plt
import pandas as pd
import json
# 알파벳 출현 빈도 데이터 읽어 들이기 --- (①)
with open( " ./lang/freq.json " ,  " r " , encoding= " utf-8 " ) as fp:
    freq = json.load(fp)
# 언어마다 계산하기 --- (②)
lang_dic = {}
for I, lbl in enumerate(freq[0][ " labels " ]):
    fq = freq[0][ " freqs " ][i]
    if not (lbl in lang_dic):
        lang_dic[lbl] = fq
        continue
    for idx, v in enumerate(fq):
        lang_dic[lbl][idx] = (lang_dic[lbl][idx] + v) / 2
# Pandas의 DataFrame에 데이터 넣기 --- (③)
asclist = [[chr(n) for n in range(97,97+26)]]
df = pd.DataFrame(lang_dic, index=asclist)
# 그래프 그리기 --- (④)
plt.style.use('ggplot')
df.plot(kind="bar", subplots=True, ylim=(0,0.15))
plt.savefig("lang-plot.png")
```

이 전 텍스트 파일로 만든 빈도 데이터(JSON형식) 읽기

각 언어의 알파벳 출현 빈도를 집계

Pandas의 dataframe에 딕셔너리 자료형(dict) 데이터를 넣음