

~ 7/12 한 일

20230850 송준용

- 백준 문제 풀이 (파이썬)
- 파이썬 딥러닝 특강
- Node.js 스터디

백준 문제 풀이



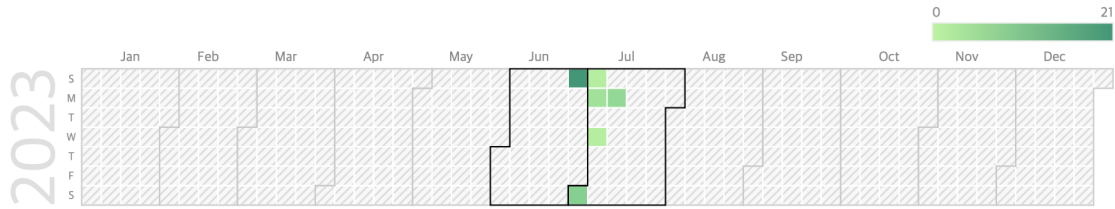
hello

정보언어

등수	117213
맞은 문제	41
맞았지만 만점을 받지 못한 문제	1
시도했지만 맞지 못한 문제	0
제출	65
맞았습니다	43
틀렸습니다	8
시간 초과	2
런타임 에러	6
컴파일 에러	6
학교/회사	성결대학교



2022-2023년

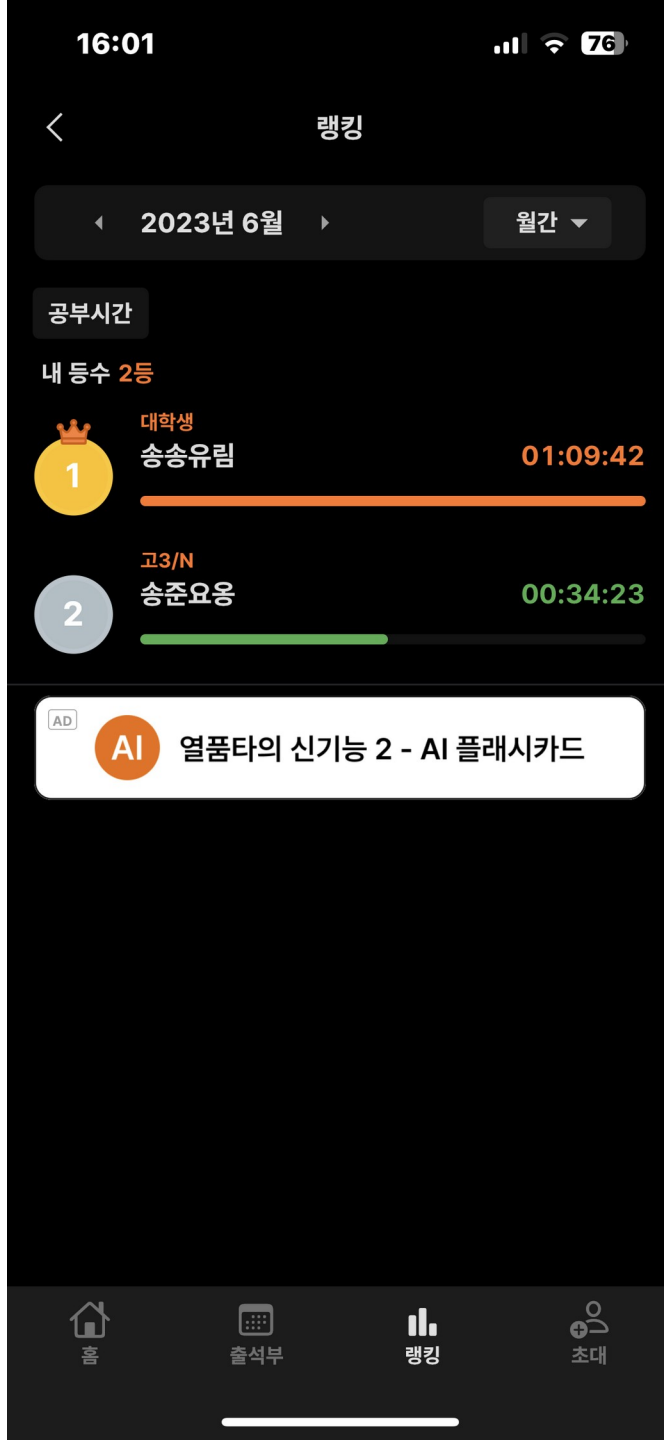


맞은 문제

1000 1001 1008 1057 1094 1110 1330 1463 1676 1789 1929 2217 2480 2525 2557 2588 2739 2751 2753 2839 2884 5347 9498 9655
10158 10171 10172 10430 10818 10869 10926 10950 10952 10998 11047 11382 13241 14681 17413 18108 25206

맞았지만 만점을 받지 못한 문제

16139



```

!pip install pyserial
!pip install numpy
!pip install scikit-learn
!pip install tensorflow

import serial
import time
import numpy as np
from datetime import datetime
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

attendance_list = []
attendance_data = []

def get_data_from_arduino():
    arduino = serial.Serial("COM3", 9600) # Change "COM3"
    time.sleep(2) # Wait for connection to establish
    while True:
        data = arduino.readline()
        data = data.decode()
        if "AttendanceData:" in data:
            _, hour, minute, rssi = data.split(" ")
            return int(hour), int(minute), int(rssi)

```

```

def scan_beacon():
    hour, minute, rssi = get_data_from_arduino()
    attendance_data.append([hour, minute, rssi])

    attendance_result = model.predict_classes(scaler.transform(np.array([[hour, minute, rssi]])))[0]
    if attendance_result == 1:
        print("참석 중")
    else:
        print("결석 중")
    attendance_list.append(attendance_result)

time_interval = 5 * 60 # 5분
total_duration = 60 * 60 # 1시간
start_time = time.time()

# 아두이노로부터 데이터를 수신하는 동안 5분 간격으로 스캔 실행
while time.time() - start_time < total_duration:
    scan_beacon()
    time.sleep(time_interval)

attendance_data = np.array(attendance_data)

# 샘플 참석 여부 (1: 참석, 0: 결석)
attended = np.random.randint(0, 2, size=len(attendance_data))

# 학습 데이터와 테스트 데이터로 나누기

```

```

데이터 정규화
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

모델 생성 및 학습
model = Sequential([
    Dense(32, activation='relu', input_shape=(3,)),
    Dense(64, activation='relu'),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))

훈련된 모델의 정확도 출력
accuracy = model.evaluate(X_test, y_test)[1]
print(f"모델 정확도: {accuracy:.4f}")

참석률 계산
def calculate_attendance_percentage(attendance_list):
    total_entries = len(attendance_list)
    attended_count = sum(attendance_list)
    attendance_percentage = (attended_count / total_entries) * 100
    return attendance_percentage

이 완료된 후 참석률 계산
attendance_percentage = calculate_attendance_percentage(attendance_list)
print(f"수업 참석률: {attendance_percentage:.2f}%")

```

파이썬 딥러닝 특강

```

import cv2
import easyocr

# EasyOCR을 사용하여 이미지에서 텍스트 추출
def extract_text(image_path):
    reader = easyocr.Reader(['en'], gpu=False)
    image = cv2.imread(image_path)
    result = reader.readtext(image)
    return '\n'.join([res[-2] for res in result])

# 예제 이미지 파일 경로
image_paths = [
    "image1_A+.jpg", "image2_A+.jpg", "image3_A+.jpg",
    "image1_A.jpg", "image2_A.jpg", "image3_A.jpg",
    "image4_B+.jpg", "image5_B+.jpg", "image6_B+.jpg",
    "image7_B.jpg", "image8_B.jpg", "image9_B.jpg",
    "image10_C+.jpg", "image11_C+.jpg", "image12_C+.jpg",
    "image13_C.jpg", "image14_C.jpg", "image15_C.jpg"
]

text_data = [extract_text(image_path) for image_path in image_paths]

```

```

    'A+', 'A+', 'A+',
    'A', 'A', 'A',
    'B+', 'B+', 'B+',
    'B', 'B', 'B',
    'C+', 'C+', 'C+',
    'C', 'C', 'C'
])

# EasyOCR 출력은 딥러닝 모델을 활용하므로 추가 전처리 과정이 필요하지 않습니다.

# 예제와 마찬가지로 학점 분류를 수행하기 위해 머신러닝 알고리즘 적용
X_train, X_test, y_train, y_test = train_test_split(text_data, y, test_size=0.2, random_state=42)
vectorizer = TfidfVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)

clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))

# 새로운 이미지 파일에서 텍스트 추출
new_image_path = "new_image.jpg"
new_text = extract_text(new_image_path)

# 텍스트 데이터 전처리 및 예측
new_X = vectorizer.transform([new_text])
predicted_grade = clf.predict(new_X)
print("예측된 학점:", predicted_grade[0])

```

Node.js

- > 🧑 JS_Study week5
- > 🧑 JS_Study week4
- > 🧑 JS_Study week 3
- > 🧑 JS_Study week 2
- > 📄 준용이의 파이썬 일기장
- > 🕒 JS_Study week 1
- > 🦁 CSS Part3
- > 📄 시작하기



JS_Study week5

1. 이터레이터

이터레이션(Iteration)이란 반복 처리를 뜻한다.

특정 데이터 안의 요소들을 연속적으로 처리하는 행위를 말한다.

```
let arr = [1, 2, 3, 4, 5]
for (let i = 0; i < arr.length; i++){
  console.log(arr[i]);
}
```

반복 가능한 객체인 배열은 이터러블 객체라고 부른다(iterate + able)

기존의 반복문인 `for`, `forEach`, `for-in` 은 문제점이 하나 있다.

기존의 반복문인 `for`, `forEach`, `for-in` 에는 몇 가지 문제점이 있다.

가독성: 모든 반복문에서 인덱스(i)와 배열 길이(arr.length)를 사용해야 하기 때문에 코드가 복잡해지고 가독성이 떨어질 수 있다.

Commit



django-practice Public ☆ Star ▼

Python Updated 28 minutes ago

Clang Public ☆ Star ▼

C Updated 46 minutes ago

pythonselfstudy Public ☆ Star ▼

Updated 47 minutes ago

JSStudy Public ☆ Star ▼

JavaScript Updated 2 weeks ago

할 일

- 백준 문제 풀이 (스터디 + 파이썬 공부)
- 인공지능 코드 (현재 하고 있는 연구 활동)
- 일주일에 커밋 3 번
- Node.js 스터디 끝나면 리액트 스터디 시작