Growth Hackers 16기 DS Quest

목차

[유의사항 및 제출 양식 안내]

※ DS 퀘스트는 [Programming], [Data Analysis] 그리고 [전략 도출] 세 영역으로 구성되어 있으며, [Programming] 영역은 Pandas & Python과 알고리즘 문항, [Data Analysis]는 EDA, 모델링 문항으로 구성되어 있습니다.

※ 코딩이 처음이신 경우, Codecademy 의 기초 Python 강좌 혹은 Jump to Python(박응용 著)을 통해 Python을 학습하신 후 문제를 해결하실 수 있습니다.

(Jump to Python : https://wikidocs.net/book/1)

※ 퀘스트 관련 문의사항은 recruit@ghsnu.com 또는 카카오톡 플러스친구 bit.ly/GH16th_ASK로 보내주시기 바랍니다.

※ '지원서'에 안내된 구글 폼(bit.ly/GH16th_INFO)도 꼭 제출 부탁드립니다.

```
김OO-GH16기-DS-지원서.zip
├── 김OO-GH16기-DS-지원서.pdf
├── 김OO-GH16기-DS-퀘스트풀이.pdf
├── Programming/
│   ├── quest-1/
│   │   └── solution.ipynb
│   └── quest-2/
│       └── sol.ipynb
└── DataAnalysis/
    ├── analysis-solution.xlsx
    ├── analysis-solution.ipynb
```

※ 최종 Quest 파일은 지원서 파일과 함께 압축하여 단일한 .zip 파일로 제출해주시길 바랍니다. 파일명은 '[이름]- GH16기-DS-지원서.zip'입니다. (e.g. 김철수-GH16기-DS-지원서.zip)

※ 압축 파일은 '지원서 파일', '퀘스트 풀이 파일', '[Programming] 영역 소스코드 파일', '[Data Analysis] 영역 풀이 파일'을 포함해야 합니다. 세부 디렉토리 구조는 위 예시를 참고 바랍니다.

※ 퀘스트 풀이 파일은 [Programming], [Data Analysis] 그리고 [전략 도출] 각 영역의 분량 제한을 지켜 하나의 pdf 파일로 변환 후 제출 바랍니다. 분량 제한을 초과할 시 채점에 불이익이 있을 수 있습니다.

※ 퀘스트 풀이 파일의 파일명은 '[이름]-GH16기-DS-퀘스트풀이.pdf'입니다.

(ex. 김철수-GH16기-DS-퀘스트풀이.pdf)

※ [Programming] 영역 풀이 시, 기타 라이브러리 사용 및 보조함수, 타 모듈 작성이 가능합니다. 타 모듈 및 라이브러리 사용 시, solution.ipynb 또는 sol.ipynb 안에서 import 해주시길 바랍니다.

※ [Data Analysis] 영역 풀이 파일들(.xlsx, .ipynb 등)은 DataAnalysis 폴더에 모두 넣어주세요.

[Programming]

※ 본 영역은 Python 기초 소양을 측정하기 위한 것이므로 Python을 활용하여 해결해주시길 바랍니다.

※ 풀이코드는 .ipynb파일로 제출 부탁드립니다.

※ 프로그래밍 영역에 대한 pdf 답안은 코드 자체가 아닌 코드의 구조와 로직을 설명해주시길 바랍니다. pdf 답안의 분량은 quest-1, quest-2를 합쳐서 2페이지 이내로 제출해 주세요. 제출하시는 소스 코드의 분량 제한은 없습니다.

[quest – 1 / Pandas & Python]

본 문항은 Growth Hackers에서 자주 쓰이는 Pandas에 대한 기초적 이해도 증진을 목적으로 합니다. 제공된 데이터와 pandas 라이브러리를 활용하여, 아래 문항들을 답변해주시길 바랍니다. (solution.ipynb 파일에 작업하여 그대로 제출 부탁드립니다.)

<데이터 설명>

※ 아래의 데이터들은 실제 데이터에 일부 변형을 가한 형태로 실제 데이터와 차이가 있을 수 있습니다.

cheese_details_url_country_datetime.csv

: cheese.com 웹사이트에 존재하는 각 페이지의 url, 지역, 무게 등의 정보를 담은 데이터

칼럼명	칼럼 데이터 타입	칼럼 내용
url	Object	Cheese.com 웹사이트에서 해당 치즈의 페이지
Country	Object	치즈의 생산 국가
Region	Object	치즈의 생산 지역
weight	Int64	치즈의 무게
datetime	Object	웹사이트 등록 일자

cheese_details_0_500.csv, cheese_details_500_1187

: cheese.com 웹사이트에 존재하는 치즈의 특성에 대한 데이터

칼럼명	칼럼 데이터 타입	칼럼 내용

url	Object	Cheese.com 웹사이트에서 해당 치즈의 페이지
Type	Object	치즈의 유형
Fat_content	Object	치즈의 지방 함량
Milk	Object	치즈 생산에 사용된 우유 종류
Calcium_content	Object	치즈의 칼슘 함량
Texture	Object	치즈의 질감
color	Object	치즈의 색깔

<문제>

문제 0

cheese_details_url_country_datetime.csv, cheese_details_0_500.csv, cheese_details_500_1187.csv를 pd.read_csv를 활용해 각각 data_1, data_2, data_3 데이터 프레임으로 내려 받으세요. (데이터 설명에 존재하지 않는 열은 drop 해주세요.)

문제 1「테이블 합치기」

문제 1-0

data_1의 datetime 열의 데이터 타입을 pd.to_datetime을 활용해 timestamp로 변경하고 해당 datetime 열의 달(month)를 담은 month 열을 추가하세요. 이 때, dt.month를 활용하세요.

(예시)

	datetime	→	month
row 1	2023-01-07		1
row 2	2023-08-12		8
row 3	2023-12-24		12

문제 1-1

data_2와 data_3를 행을 기준으로 합한 problem_1_1_df 데이터 프레임을 생성하세요. 이 때, ignore_index()를 활용하여 새로운 인덱스를 만드세요.

문제 1-2

문제 1-1에서 생성된 problem_1_1_df와 data_1을 url을 기준으로 inner join한 problem_1_2_df 데이터 프레임을 생성하세요.

문제 2「결측치 처리」

문제 2-1

문제 1-2에서 생성된 problem_1_2_df 데이터 프레임에 대해 isnull(), mean()을 활용하여 problem_1_2_df의 각 열의 이름을 index로 갖고, 값으로 해당 열의 결측치 비율을 갖는 시리즈 객체를 missing_ratio라는 이름으로 생성하세요.

문제 2-2

문제 2-1에서 생성된 missing_ratio를 활용하여 결측치 비율이 60%를 넘는 열의 이름을 담은 list를 생성하세요. 이를 활용하여 problem_1_2_df에서 결측치 비율이 60%를 넘어서는 열을 제거한 problem_2_df 데이터 프레임을 생성하세요.

문제 3「함수 생성 및 적용」

문제 3-1

input으로 문자열 또는 null이 주어질 때, input이 null인 경우 0, input이 문자열인 경우 해당 문자열을 쉼표(,)를 기준으로 분리하여 리스트를 생성하고 해당 리스트의 길이를 반환하는 count_mixed 함수를

생성하세요. (hint : split 함수를 사용해보세요.)

(input, output 예시)

(input : null / output : 0) (input : cow, goat / output : 2) (input : cow / output : 1)

문제 3-2

문제 2에서 생성된 problem_2_df 데이터 프레임에 count_mixed 함수를 apply를 통해 milk, country열에 적용하고 각각의 값을 mixed, country_mixed라는 열로 저장한 problem_3_df 데이터 프레임을 생성하세요

(예시)

	milk	country	mixed	country_mixed
row 1		Italy	0	1
row 2	Sheep	France, USA	1	2
row 3	Cow, Goat	France	2	1

문제 4

문제 3에서 생성된 problem_3_df 데이터 프레임에서 country_mixed열과 mixed열이 모두 1인 행들만 포함한 problem_4_df 데이터 프레임을 생성하세요.

문제 5

문제 4에서 생성된 problem_4_df 데이터 프레임의 color 열에 대해 yellow가 포함된 행들의 color 열에 yellow를 배정한 problem_5_df 데이터 프레임을 생성하세요.

이때 re모듈의 compile을 활용하세요.

(예시)

	color	→	color
row 1	pale yellow		yellow
row 2	green		green
row 3	yellow		yellow

문제 6

문제 5에서 생성된 problem_5_df 데이터 프레임에 value_counts()를 활용해 국가명을 인덱스로 데이터 프레임에서 해당 국가에 해당하는 행의 수를 값으로 담은 시리즈 객체를 생성하세요.

그 다음, 해당 시리즈 객체를 to_dict()를 활용하여 dictionary형태로 바꾼 country_dict를 생성하세요.

이때, country_dict에서 value값을 기준으로 내림차순 정렬해주세요.

[quest – 2 / Algorithm]

그핵이는 장마철에 대비하기 위해 들판에 빗물 저류조*를 설치하고 있습니다. 때마침 저류조의 성능을 시험하기에 충분한 양의 비가 내리고 있습니다. 저류조의 성능은 저류조에 각각 몇 칸의 물이 고이는지 판단하는 방식으로 이루어집니다.

아래의 설명을 참조하여 저류조를 표현한 list가 input으로 주어졌을 때 저류조에 고이는 물의 칸수를 파악하는 함수 your_algorithm()을 sol.ipynb에 작성해주시길 바랍니다.

*빗물 저류조: 호우로 발생하는 많은 양의 빗물을 지하에 간단히 저장하여 홍수를 예방함과 동시에 수자원부족 시 저류된 빗물을 활용할 수 있도록 한 구조물

<유의사항>

1) 퀘스트 파일에 주어진 Test Dataset을 활용하여 1000개의 input에 대한 정답을 여부를 파악할 수 있습니다.
2) 실제 채점에서는 주어진 Test Dataset 이외의 데이터를 추가로 활용하여 채점이 이루어집니다.
3) 퀘스트 파일에 주어진 길이가 10,000,000인 리스트를 활용해 본인이 작성한 함수의 효율성을 파악할 수 있습니다.
4) 실제 채점에서는 input에 대해 정답을 맞추는 것뿐만 아니라 해당 알고리즘의 효율성 또한 채점 대상이 됩니다.

<설명>

1) 저류조는 길이가 8~10000000이고 0~4의 값을 갖는 list로 주어집니다.

2) list에 존재하는 0은 구멍으로, 바닥과 연결되어 있으며 이로 인해 바닥과 연결된 모든 물은 바닥에 흡수되어 사라집니다.

3) 저류조의 양 끝은 막혀 있지 않습니다. (즉, 양쪽 바깥으로 물이 흘러내릴 수 있습니다.)

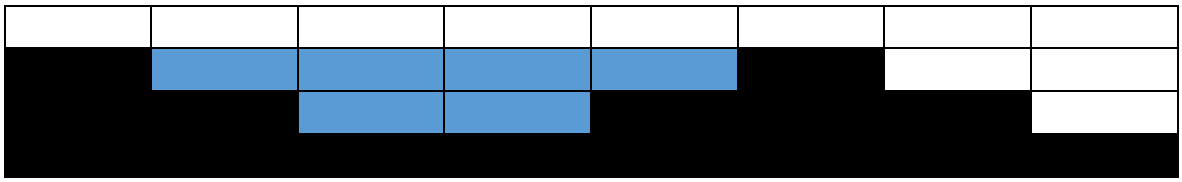
4) 저류조에 비가 내리면 아래와 같이 물이 고이게 됩니다.

5) 저류조를 표현한 list가 input으로 주어졌을 때 해당 구조물에 물이 고이는 칸수를 output으로 하는 함수를 만들어주세요.

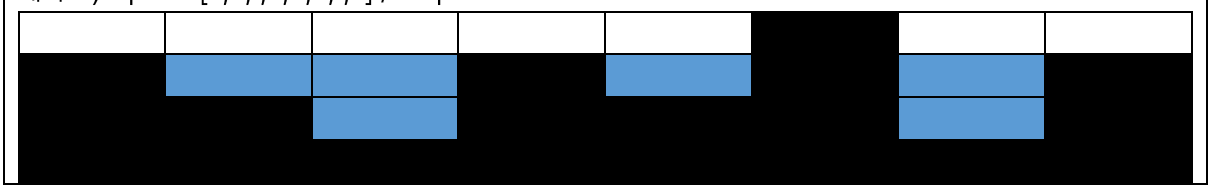
<그림 예시>

예시 1) input = [3,2,1,0,2,3,2,1] / output = 0

예시 2) input = [3,2,1,1,2,3,2,1] / output = 6



예시 3) input = [3,2,1,3,2,4,1,3] / output = 6



[Data Analysis]

※ 본 영역은 지원자의 데이터 분석 능력을 평가하기 위해 출제되었습니다. 분석을 위한 논리를 중심적으로 평가하므로, 도구와 방식은 평가의 대상이 되지 않으며, 별다른 제한을 두지 않습니다.

※ 풀이에 사용하시는 .ipynb, .py, .r, .xlsx 등 소스 코드 및 풀이 도구의 제한은 없습니다.

※ 제공된 데이터 셋을 바탕으로 풀이해 주시고, 외부 데이터를 접목하지는 말아 주시기 바랍니다.

※ 답안의 형태 및 분량은 다음과 같이 총 5 페이지 이내로 제한해주시길 바랍니다.

- EDA 문항 풀이 최대 2 페이지 (시각화 및 코드 포함)
- 모델링 문항 풀이 최대 3 페이지 (시각화 및 코드 포함)

※ 본 문제에서 등장하는 맥락, 상황은 오직 데이터 문제 출제의 편의만을 위해 가정한 것으로, 실제와는 전혀 무관한 완전한 허구이며, 그 어떠한 가치판단도 포함하고 있지 않음을 밝힙니다.

[GH Delivery 데이터 분석]

GH Delivery는 모회사 GH팡의 신생 로지스틱스 자회사입니다. 현재 GH Delivery는 모회사 GH팡의 직매입 상품의 배송을 담당하고 있습니다. 그핵이는 배달시간이 지연될 것 같은 주문들을 미리 파악하여 배달 시간을 최적화하고, 이를 통해 고객 만족도 및 전반적인 Order_Rating을 높이고자 합니다. 이를 위해서 그핵이는 배달 시간을 예측하는 모델을 만들고자 합니다. 주어진 데이터 셋을 활용하여 분류 모델을 생성해주세요.

<유의사항>

1) 데이터 처리, 가공, 조합, 시각화 방식과 관련 프레임워크 사용은 자유입니다. 하지만 제공된 데이터 셋 이외의 외부 데이터는 도입하지 말아 주시기 바랍니다.

2) 데이터 셋에 위도, 경도 데이터가 주어졌지만 해당 지역은 가상지역이라고 가정합니다.

<데이터 설명>

GH_Delivery.csv

: 배달시간, 배달 시 교통상황 등의 정보가 포함되어 있습니다.

칼럼명	설명
Order_ID	배달의 ID (데이터 테이블에서 key로 사용됨)
Agent_Age	배달원의 나이, 현재 고용된 배달원들의 나이는 20~60세
Order_Rating	고객이 해당 배달에 준 평점(1~5점 사이의 값)
Store_Latitude	물건을 보내는 가게(배송 출발지)의 위도
Store_Longitude	물건을 보내는 가게(배송 출발지)의 경도
Drop_Latitude	물건을 받는 고객(배송 도착지)의 위도
Drop_Longitude	물건을 받는 고객(배송 도착지)의 경도
Order_Date	주문 일자
Order_Time	주문 시간
Pickup_Time	배달원이 가게에서 배달물품을 픽업한 시간
Weather	날씨
Traffic	교통상황
Vehicle	배달에 이용된 이동수단
Area	배달 지역
Delivery_Time	배달에 걸린 시간
Category	배달 품목의 카테고리

[EDA]

EDA(Exploratory Data Analysis, 탐색적 데이터 분석)는 데이터의 이상치, 결측치 등을 적절히 처리하고, 데이터의 특징을 올바르게 이해하여 인사이트를 찾아내는 작업을 의미합니다. GH Delivery는 배달에 대한 고객 만족도를 핵심 성과 지표(Key Performance Indicator)로 삼고 있기 때문에, Order_Rating과 Delivery_Time에 관심이 많습니다.

<문제>

문제 0

주어진 데이터에서 결측치, 이상치 또는 정합성이 맞지 않는 데이터가 있을 수 있습니다. 만약 있다면 적절히 삭제 혹은 대체하고, 해당 처리 방식에 대한 이유를 설명해주세요.

그리고 처리가 완료된 데이터 프레임을 processed_df라는 이름으로 저장해주세요.

문제 1

Category에 따른 Delivery_Time의 분포를 파악하고자 합니다. 적절한 방식을 활용하여 시각화를 진행하고 도출한 인사이트를 간략하게 설명해주세요.

문제 2

Delivery Time과 Order_Rating의 관계를 파악하고자 합니다. 적절한 방식을 활용해 시각화를 진행하고 도출한 인사이트를 간략하게 설명해주세요.

[Modeling]

이상치, 결측치 및 정합성에 맞지 않는 데이터를 적절히 처리한 processed_df를 활용하여 아래의 문제들을 해결해주세요.

<문제>

문제 1

배달 시간을 의미하는 Delivery_Time 열을 활용하여 배달이 느린 경우 1, 느리지 않은 경우 0을 갖는 Delayed라는 열을 processed_df 데이터 프레임에 추가하고, modeling_df라는 이름으로 저장하세요.

(hint : eda 문항 1번을 참고하여 배달 시간의 느리고 빠름을 결정 해보세요.)

문제 2-1

문제 1의 결과 생성된 modeling_df 데이터 프레임의 모든 행을 활용하여 DecisionTreeClassification을 사용한 분류 모델을 만들어주세요.

이때, 범주형 열에 대해서 적절한 encoding을 진행하세요.

문제 2-2

평가지표로 accuracy, precision, recall의 값을 출력하세요. 또한 출력한 평가지표들의 값을 통해 해당 모델을 평가하세요. 만약 모델에 문제가 있다면 왜 그런 문제가 발생했다고 생각하는지 설명해주세요.

참고로 그핵이는 배달 시간이 상대적으로 오래 걸리는 배달을 파악하는 것을 중요하게 생각하고 있습니다. 이를 참고하여 모델을 평가해주세요.

문제 2-3

데이터 셋 및 하이퍼 파라미터를 변형하여 모델의 성능을 개선시키고 사용한 방법에 대해 설명해주세요.

(모델 성능의 개선과 더불어 다양한 시도 또한 채점에 반영됩니다.)

문제 2-4

문제 2-3의 결과 생성된 모델에서 중요한 feature가 무엇인지 파악하고 이를 통해 배달 시간이 길어지는 원인은 무엇인지, 그리고 이를 해결하기 위한 방법은 무엇이 있을지 제시해주세요.

<유의사항>

1) 모델링을 위해 EDA에서 진행한 전처리 외 추가적인 전처리가 필요하다고 판단되는 경우 진행해 주시면 됩니다.
2) DecisionTreeClassification 모델 생성이 가능한 모든 종류의 툴, 언어, 환경을 허용합니다. (엑셀, R, java, C언어 등)
   주어진 예시 코드는 파이썬입니다.

```python
#모듈 임포트
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import accuracy_score, classification_report
from sklearn import tree

#train, test 데이터 분리

mymodel = DecisionTreeClassifier()

#accuracy, precision, recall 평가
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

report = classification_report(y_test, y_pred, target_names=['1','0'])
print(report)
```

[전략 도출]

※ 본 영역은 지원자의 문제 해결 능력을 평가하기 위해 출제되었습니다.

※ 답안의 분량은 공백포함 500자 이내 입니다.

전략 도출

현존하는 기업 또는 업종 중 관심 있는 곳의 문제를 정의한 후, 해당 문제를 해결하기 위한 전략을 한 가지 서술해 주세요. (공백포함 500자 이내)

- 기업/업종, 전략은 반드시 각각 1가지만 제시해 주세요, 2개 이상 제시하더라도 처음에 쓴 한 가지만 반영됩니다.
- 작성하신 전략은 1) 현실성, 2) 창의성을 중점으로 채점할 예정입니다.
- 면접 시에 이 답변과 관련하여 추가적인 질문이 있을 예정입니다.