2021 2학기 컴퓨터공학전공 인공지능 수업 발표

경제적 생산, 사회적 지지 등에 따른 국가별 행복지수 분석

Happiness scored according to economic production, social support

전기전자통신컴퓨터공학부 컴퓨터공학전공
2019108257 김진웅

HAPPINESS SCORED ACCORDING TO ECONOMIC PRODUCTION, SOCIAL SUPPORT

CONTENTS

개요

발표의 배경

Dataset 소개

발표 배경

- 행복은 어디에서 오는것일까요

전 세계의 **행복지수**?
어떤 **기준**으로 측정하고
어떤 **결과**를 드러낼까?

행복이란 어디서 오는 걸까와 같은 질문을 되새기며
행복이란 단어에 대해 평소에 궁금함이 많았습니다.

이번 인공지능 발표를 위해 Kaggle에서 Dataset을 탐색하다가
전 세계의 국가들에 대해 여러 기준을 적용,
연도별 행복지수라는 것을 매긴 Dataset을 발견했습니다.

행복지수를 측정하는 기준으로 삼은
Column들과 드러나는 결과를 직접 분석하고 싶어
본 Dataset을 선정하게 되었습니다.

Dataset 소개

- 어떤 데이터셋을 사용했을까요

World Happiness Report

UN 자문기관인
SDSN(Sustainable Development Solution Network)이
전세계 155개국의 행복지수를
순위로 매긴 dataset으로, 2017년 3월 20일
유엔에서 열린 세계 행복의 날 행사에서 공개되었습니다.
취득한 데이터들은 Gallup World Poll에서 진행한
설문 데이터를 기반으로 마련되었습니다.

https://www.kaggle.com/unsdsn/world-happiness

Dataset 소개

- 어떤 데이터셋을 사용했을까요

자료 수집 방식

150여개의 국가에서 3000명의 사람들에게 설문조사를 실시하는데,
자신의 인생이 행복한지에 대해 설문을 진행합니다.
이때 설문에 응하는 사람들의 응답에 따라
인생 선택의 자유도, 정부에 대한 신뢰,
자신의 경제력, 사회나 가족의 지원,
기대수명, 관용이라는
6가지 기준에 각각 0~10점씩의 점수가 매겨집니다.
이때 각 국가마다 가장 최저치로 받은 6개의 분야 점수를 합산해
디스토피아 지수라는 지수를 따로 계산합니다.
해당 지수는 가치관의 차이를 반영하기 위함이며,
국가의 행복지수 측정에 직접적으로 관여하지 않습니다.

데이터 처리 및 시각화

모듈 import 및 사전 설정
데이터 읽어오기
데이터 시각화

모듈 import 및 사전 설정

- 사전에 필요한 모듈을 불러오고 기본적인 설정을 해두기

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

pandas: 데이터 분석 및 조작을 위한 라이브러리
numpy: 행렬 및 다차원 배열 처리를 위한 라이브러리
seaborn: 다양한 색상 테마와 통계용 차트 등의 기능을 추가한 시각화 패키지
matplotlib: 데이터를 차트나 플롯으로 그려주는 패키지

```
import scipy
```

scipy: numpy 라이브러리의 공학적 연산 처리를 위해 필요한 라이브러리

```
import folium
```

folium: 조정된 파이썬 데이터를 지도로 출력할 수 있도록 처리하는 라이브러리

```
import geopandas as gpd
```

geopandas: 지리공간 데이터 처리를 더 쉽게 만들어주는 라이브러리

```
# Setting the default style of the plots
sns.set_style('whitegrid')
sns.set_palette('Set2')

# My custom color palette
my_palette = ["#7A92FF", "#FF7AEF", "#B77AFF", "#A9FF7A", "#FFB27A", "#FF7A7A",
              "#7AFEFF", "#D57AFF", "#FFDF7A", "#D3FF7A"]
```

이후 나타낼 차트에 대해 색상 팔레트 미리 준비

데이터 읽어오기

- 데이터를 제대로 읽어오고, 사용하기 위해 설정하기

```
# Importing the 5 datasets
data_2015 = pd.read_csv("../input/world-happiness/2015.csv")
data_2016 = pd.read_csv("../input/world-happiness/2016.csv")
data_2017 = pd.read_csv("../input/world-happiness/2017.csv")
data_2018 = pd.read_csv("../input/world-happiness/2018.csv")
data_2019 = pd.read_csv("../input/world-happiness/2019.csv")
```

core 데이터를 불러오기

Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.4363	2.70201
Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204

```
data_2015 = data_2015[['Country', 'Happiness Rank', 'Happiness Score', 'Economy (GDP per Capita)', 'Family',
                       'Health (Life Expectancy)', 'Freedom', 'Generosity', 'Trust (Government Corruption)',
                       'Dystopia Residual']]
data_2016 = data_2016[['Country', 'Happiness Rank', 'Happiness Score', 'Economy (GDP per Capita)', 'Family',
                       'Health (Life Expectancy)', 'Freedom', 'Generosity', 'Trust (Government Corruption)',
                       'Dystopia Residual']]
data_2017 = data_2017[['Country', 'Happiness.Rank', 'Happiness.Score', 'Economy..GDP.per.Capita.', 'Family',
                       'Health..Life.Expectancy.', 'Freedom', 'Generosity', 'Trust..Government.Corruption.',
                       'Dystopia.Residual']]
```

각각의 csv파일에서 column의 이름들이 다르게 적혀있어 모든 column들을 통일시켜 데이터를 읽어올 수 있도록 하기 위해 각 csv파일의 column들을 불러오는 코드

데이터 읽어오기

- 데이터를 제대로 읽어오고, 사용하기 위해 설정하기

```
# Tables do not have the same column names, so we need to fix that
new_names = ['Country', 'Happiness Rank', 'Happiness Score', 'Economy (GDP per Capita)', 'Family',
                        'Health (Life Expectancy)', 'Freedom', 'Generosity', 'Trust (Government Corruption)',
                        'Dystopia Residual']
```

```
# Merge the data together
data = pd.concat([data_2015, data_2016, data_2017, data_2018, data_2019], axis=0)
data.head(5)
```

불러온 csv파일들의 Table들이 가진 Column들의 이름을 하나로 통일하기 위해 new_names 라는 이름의 Column list를 작성하고 모든 파일을 하나로 merge 시킨다.

데이터 읽어오기

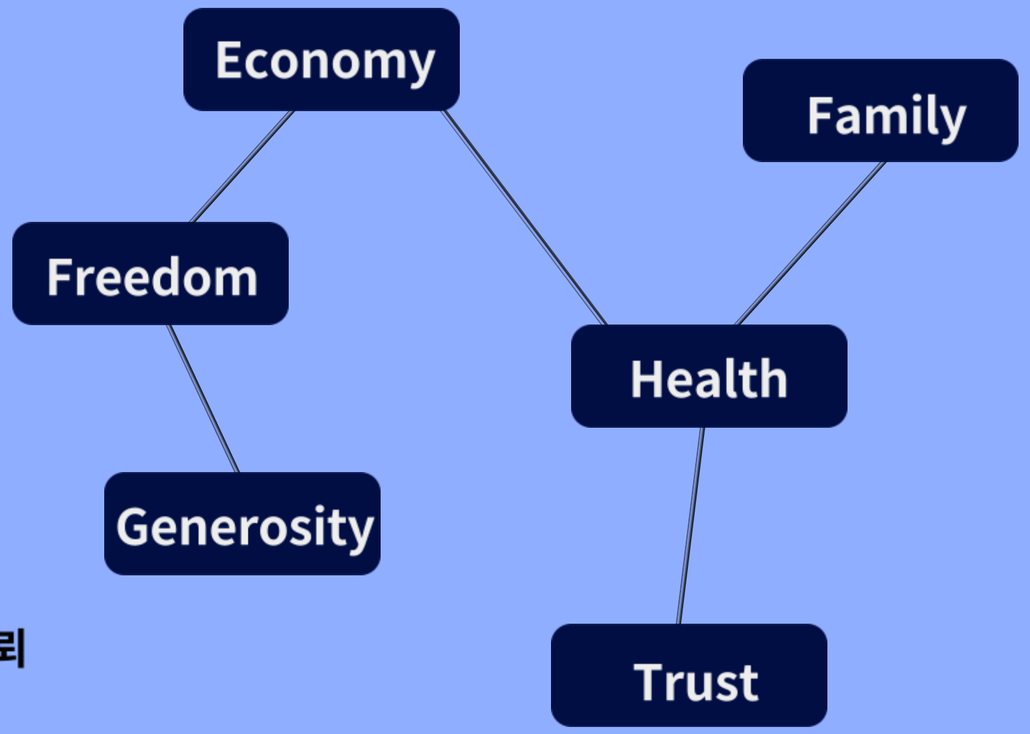
- 데이터를 제대로 읽어오고, 사용하기 위해 설정하기

모든 Column은 다음과 같다.

Country = 국가명
Happiness Rank = 해당 국가의 행복지수 순위
Happiness Score = 해당 국가의 행복지수

Economy (GDP per Capita) = 1인당 국내총생산량
Family = 가족 혹은 사회의 지원
Health (Life Expectancy) = 기대수명
Freedom = 인생 선택의 자유도
Generosity = 관용
Trust (Government Corruption) = 정부에 대한 신뢰

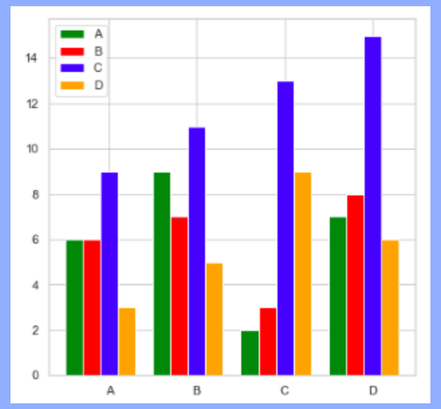
Dystopia Residual = 디스토피아 지수



데이터 시각화

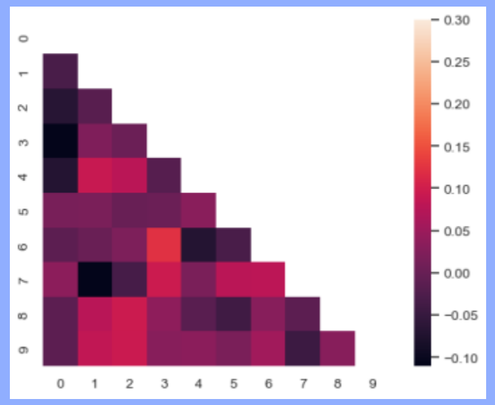
- 결과를 시각화할 패키지에 대해서

matplotlib



matpliot 라이브러리 사용
막대그래프를 통한 직접적인 비교

seaborn



seaborn 라이브러리 사용
상관관계를 나타내는 히트맵 출력

Folium

folium 라이브러리를 사용
특정 데이터를 담은 지도 출력

결과 분석

- 2015, 2019년 가장 행복지수가 높은 나라 top 10
- 2015, 2019년 가장 행복지수가 낮은 나라 top 10
- 행복지수에 가장 큰 영향을 준 요인
- 행복을 주는 요인에 대한 인식의 변화
- 지도로 보는 전세계의 2019년 행복 지수

결과 분석 - 2015, 2019년 가장 행복지수가 높은 나라 top 10


```
country_score_avg = data[data['Year']==2019].groupby(by = ['Country or region'])['Score'].mean
().reset_index()
table = country_score_avg.sort_values(by = 'Score', ascending = False).head(10)

table
```

Year에 2015년부터 2019년 사이의 값을 입력

ascending을 false로 두고 head()에 10을 넣어 상위 10개의 국가를 담은 테이블을 불러옴.

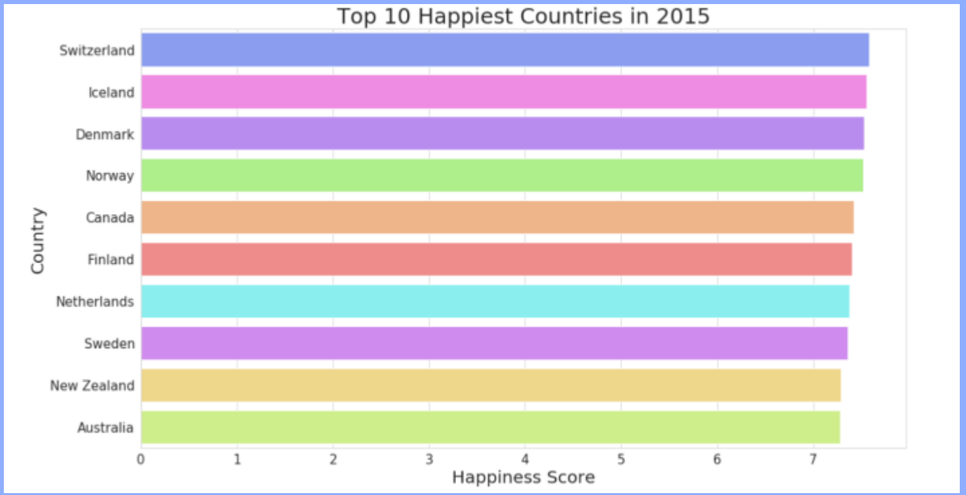
	Country or region	Score
43	Finland	7.769
36	Denmark	7.600
105	Norway	7.554
57	Iceland	7.494
98	Netherlands	7.488
133	Switzerland	7.480
132	Sweden	7.343
99	New Zealand	7.307
23	Canada	7.278
6	Austria	7.246

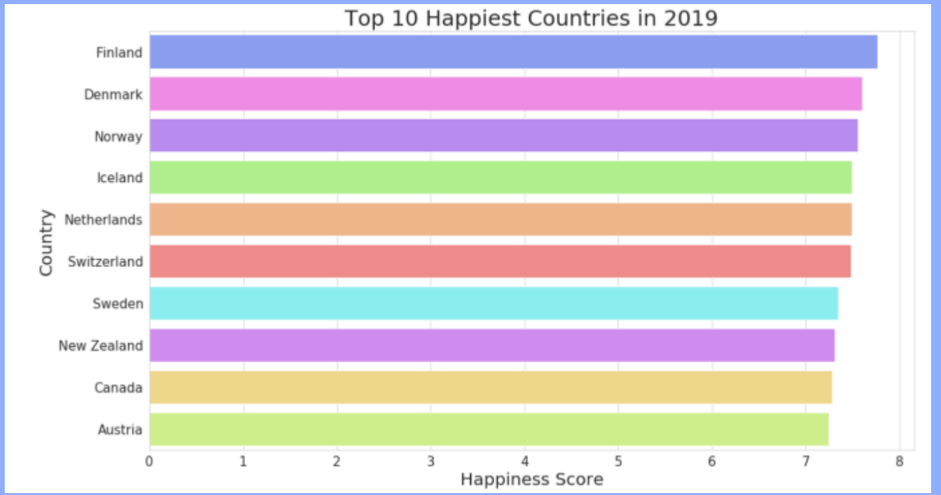
```
plt.figure(figsize = (16, 9))
sns.barplot(y = table['Country or region'], x = table['Score'], palette = my_palette)

plt.title("Top 10 Happiest Countries in 2019", fontsize = 25)
plt.xlabel("Happiness Score", fontsize = 20)
plt.ylabel("Country", fontsize = 20)
plt.xticks(fontsize = 15)
plt.yticks(fontsize = 15);
```

matplotlib에서 테이블 값을 받고
이를 이전에 설정한 my_palette 색상에 맞춤

x축의 이름을 plt.xlabel("")안에, 폰트 사이즈를 fontsize로 설정
y축의 이름을 plt.ylabel("")안에, 폰트 사이즈를 fontsize안에 설정
x축 변수들의 폰트 사이즈를 15로 하도록 plt.xticks(fontsize = 15)로,
y축 변수들의 폰트 사이즈를 15로 하도록 plt.yticks(fontsize = 15)로 설정

결과 분석 - 2015, 2019년 가장 행복지수가 높은 나라 top 10





2015년에는 북유럽 국가인 스위스, 아이슬란드, 덴마크,
노르웨이, 캐나다, 핀란드, 네덜란드, 스웨덴, 뉴질랜드, 호주 순으로 있었고,
10위권 내의 국가 중에 호주의 자리를 오스트리아가 가져간 것을 제외하면
다른 국가들은 10위 이내의 높은 행복지수를 보여준다.
즉, 4~5년의 기간동안 높은 행복지수를 유지하고 있다.

결과 분석 - 2015, 2019년 가장 행복지수가 낮은 나라 top 10


```
table2 = country_score_avg.sort_values(by = 'Score', ascending = True).head(10)

table2
```

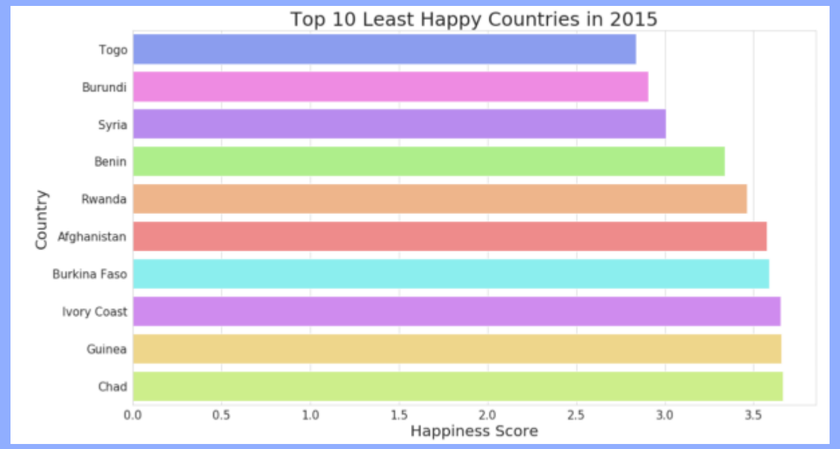
	Country or region	Score
128	South Sudan	2.853
24	Central African Republic	3.083
0	Afghanistan	3.203
137	Tanzania	3.231
117	Rwanda	3.334
153	Yemen	3.380
83	Malawi	3.410
134	Syria	3.462
16	Botswana	3.488
53	Haiti	3.597

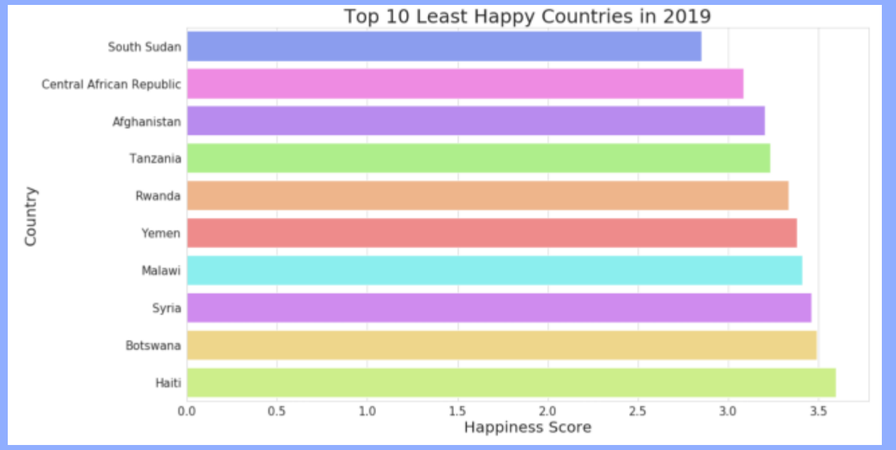
```
plt.figure(figsize = (16, 9))
sns.barplot(y = table2['Country or region'], x = table2['Score'], palette = my_palette)

plt.title("Top 10 Least Happy Countries in 2019", fontsize = 25)
plt.xlabel("Happiness Score", fontsize = 20)
plt.ylabel("Country", fontsize = 20)
plt.xticks(fontsize = 15)
plt.yticks(fontsize = 15);
```

이전과 달리
ascending을 false로 두고 head()에 10을 넣어 상위 10개의 국가를 담은 테이블2 를 불러옴.

x축의 이름을 plt.xlabel("")안에, 폰트 사이즈를 fontsize로 설정
y축의 이름을 plt.ylabel("")안에, 폰트 사이즈를 fontsize안에 설정
x축 변수들의 폰트 사이즈를 plt.xticks(fontsize = 15)로,
y축 변수들의 폰트 사이즈를 plt.yticks(fontsize = 15)로 설정

결과 분석 - 2015, 2019년 가장 행복지수가 낮은 나라 top 10





상위 10개의 국가와는 전혀 다르게도, 4년만에 대부분의 국가의 이름이 변경되었다.
빠른 경제성장으로 인해 해당 하위 10위권을 탈출하는 국가도 있는 반면,
내전과 같이 국가 전체를 뒤흔드는 사건으로 인해 최악에 가까운
행복지수로 떨어지는 국가들로 그 원인을 예측할 수 있다.
그 예시로 2015년 경부터 진행된 내전이 진행될수록 행복지수가 떨어지고 있는
남수단과 아프가니스탄, 중앙아프리카 공화국을 들 수 있다.

결과 분석 - 행복지수에 가장 큰 영향을 준 요인


```
c1 = scipy.stats.pearsonr(data['Score'], data['GDP per capita'])
c2 = scipy.stats.pearsonr(data['Score'], data['Social support'])
c3 = scipy.stats.pearsonr(data['Score'], data['Healthy life expectancy'])
c4 = scipy.stats.pearsonr(data['Score'], data['Freedom to make life choices'])
c5 = scipy.stats.pearsonr(data['Score'], data['Generosity'])
c6 = scipy.stats.pearsonr(data['Score'], data['Perceptions of corruption'])

print('Happiness Score + GDP: pearson = ', round(c1[0],2), '   pvalue = ', round(c1[1],4))
print('Happiness Score + Family: pearson = ', round(c2[0],2), '   pvalue = ', round(c2[1],4))
print('Happiness Score + Health: pearson = ', round(c3[0],2), '   pvalue = ', round(c3[1],4))
print('Happiness Score + Freedom: pearson = ', round(c4[0],2), '   pvalue = ', round(c4[1],4))
print('Happiness Score + Generosity: pearson = ', round(c5[0],2), '   pvalue = ', round(c5[1],4))
print('Happiness Score + Trust: pearson = ', round(c6[0],2), '   pvalue = ', round(c6[1],4))
```

```
corr = data.corr()

# Generate a mask for the upper triangle
mask = np.triu(np.ones_like(corr, dtype=np.bool))

# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(16, 9))

# Generate a custom diverging colormap
cmap = sns.diverging_palette(0, 25, as_cmap=True, s = 90, l = 45, n = 5)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})

plt.title('What influences our happiness?', fontsize = 25)
plt.xticks(fontsize = 15)
plt.yticks(fontsize = 15);
```

scipy라이브러리의 scipy.stats.pearsonr을 이용해
각각의 기준이 얼마나 행복지수에 있어
영향을 보이는지 data['score'] 형식으로 반환

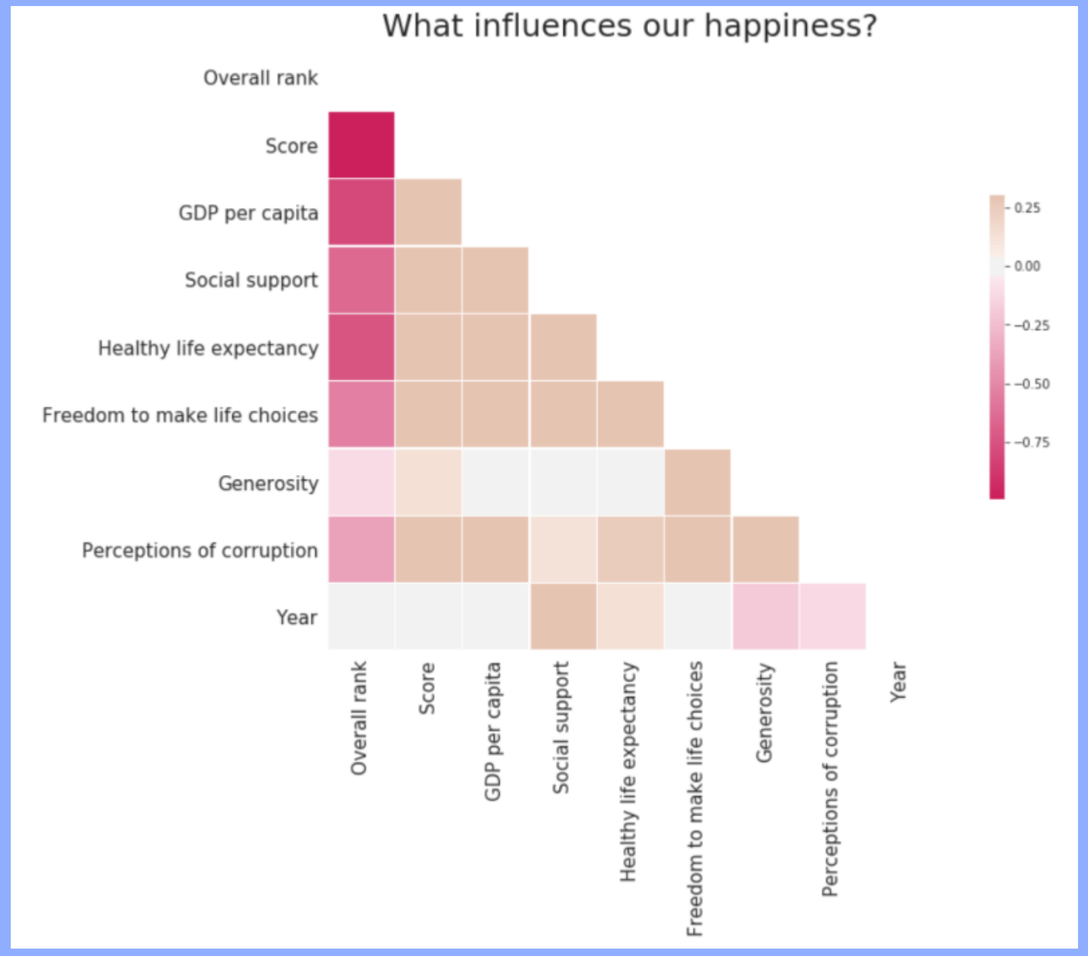
이후 print 구문으로 각 기준들의 상관계수를 출력

그 다음 data의 상관계수를 구하기 위해
data.corr()를 사용

이후 seaborn 패키지의 sns.heatmap()을 이용해
히트맵을 제작

matplotlib를 이용해 해당 히트맵의
제목 내용, 제목 크기, x축 변수들의 폰트 사이즈,
y축 변수들의 폰트사이즈를 함께 출력

결과 분석 - 행복지수에 가장 큰 영향을 준 요인




```
Happiness Score + GDP: pearson =  0.79    pvalue =  0.0
Happiness Score + Family: pearson =  0.65    pvalue =  0.0
Happiness Score + Health: pearson =  0.74    pvalue =  0.0
Happiness Score + Freedom: pearson =  0.55    pvalue =  0.0
Happiness Score + Generosity: pearson =  0.14    pvalue =  0.0001
Happiness Score + Trust: pearson =  0.4    pvalue =  0.0
```

상단의 수치를 출력한 모습과 같이, 경제적 성장, 혹은
개인의 총생산량이 0.79라는 가장 높은 영향력을 보여준다.

그 뒤로 개인의 건강, 가족, 행동의 자유,
정부에 대한 신뢰가 있었고,
관용이 가장 행복지수에 대한
영향요인 중 작은 값을 보이고 있다.

결과 분석 - 행복을 주는 요인에 대한 인식의 변화


```
# First we group the data by year and average the factors
grouped = data.groupby(by = 'Year')[['Score', 'GDP per capita',
       'Social support', 'Healthy life expectancy',
       'Freedom to make life choices', 'Generosity',
       'Perceptions of corruption']].mean().reset_index()

# Now we reconstruct the df by using melt() function
grouped = pd.melt(frame = grouped, id_vars='Year', value_vars=['Score', 'GDP per capita',
       'Social support', 'Healthy life expectancy',
       'Freedom to make life choices', 'Generosity',
       'Perceptions of corruption'], var_name='Factor', value_name='Avg_value')
```

data.groupby(by = 'Year') 형식을 이용해
2015년부터 2019년까지의
모든 데이터들을 년도별 그룹으로 묶음

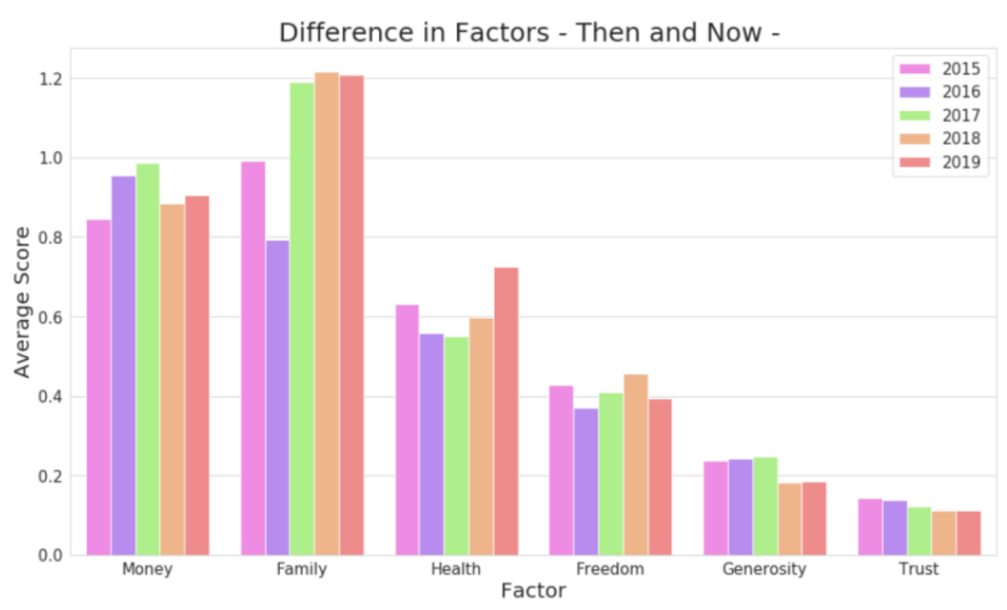
pandas 라이브러리의
melt() 함수를 이용해
그룹화한 년도별 그룹 내에서
평균 값인 Avg_value를 획득

```
plt.figure(figsize = (16, 9))

ax = sns.barplot(x = grouped[grouped['Factor'] != 'Score']['Factor'], y = grouped['Avg_value'],
            palette = my_palette[1:], hue = grouped['Year'])

plt.title("Difference in Factors - Then and Now - ", fontsize = 25)
plt.xlabel("Factor", fontsize = 20)
plt.ylabel("Average Score", fontsize = 20)
plt.xticks(fontsize = 15)
plt.yticks(fontsize = 15)
plt.legend(fontsize = 15)

ax.set_xticklabels(['Money','Family', 'Health', 'Freedom', 'Generosity', 'Trust']);
```

seaborn 라이브러리의 barplot을 이용해
Year로 그룹화한 각 변수마다의 연도별 평균값을
Avg_value를 통해 받고,

행복지수 산정 기준 6개를 기준으로
한 기준당 2015년부터 2019년까지의 평균값이
barplot으로 묶여서 나오도록 출력

결과 분석 - 행복을 주는 요인에 대한 인식의 변화



경제 생산량은 초반과 달리 그 이상은
중요요인으로 꼽히지 않게 되었고,
2017년부터 경제 생산량보다
가족, 본인의 건강이라는 두 기준이
행복지수의 요인으로 주목받을 정도로
크게 상승하고 있다.

돈보다 본인의 건강과 가족을 중요시하는 경향이
전보다 늘고있다고 분석할 수 있다.

결과 분석 - 지도로 보는 전세계의 2019년 행복 지수


```
#json file with the world map
import matplotlib.pyplot as plt
import geopandas as gpd

country_geo = gpd.read_file('../input/worldcountries/world-countries.json')

#import another CSV file that contains country codes
country_codes = pd.read_csv('../input/iso-country-codes-global/wikipedia-iso-country-codes.csv')
country_codes.rename(columns = {'English short name lower case' : 'Country or region'}, inplace
= True)

#Merge the 2 files together to create the data to display on the map
data_to_plot = pd.merge(left= country_codes[['Alpha-3 code', 'Country or region']],
                        right= country_score_avg[['Score', 'Country or region']],
                        how='inner', on = ['Country or region'])
data_to_plot.drop(labels = 'Country or region', axis = 1, inplace = True)
```

```
my_map = folium.Map(location=[10, 6], zoom_start=1.49)

my_map.choropleth(geo_data=country_geo, data=data_to_plot,
                  name='choropleth',
                  columns=['Alpha-3 code', 'Score'],
                  key_on='feature.id',
                  fill_color='BuPu', fill_opacity=0.5, line_opacity=0.2,
                  nan_fill_color='white',
                  legend_name='Average Happiness Indicator')

my_map.save('data_to_plot.html')

from IPython.display import HTML
HTML('<iframe src=data_to_plot.html width=850 height=500></iframe>')
```

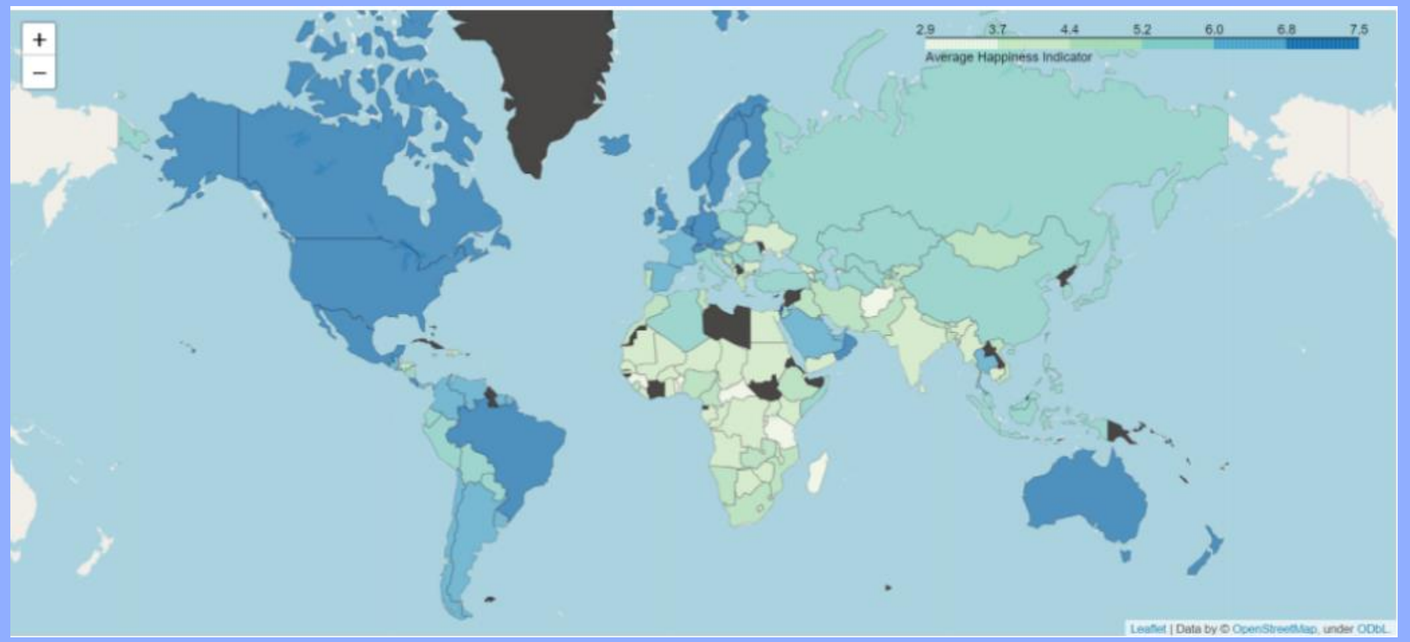
geopandas의 파일 읽어오기 기능으로
세계지도 json파일을 읽기

국가별 iso 표준 국가코드가 있는 csv파일을 읽기

folium.Map을 사용해 folium 패키지를 생성
데이터를 담은 folium map 파일을
html 파일로 저장

HTML을 임포트해 해당 html파일을 iframe태그를 사용,
창을 새로 띄우지 않고 화면에 바로 출력

결과 분석 - 지도로 보는 전세계의 2019년 행복 지수



색깔이 어두울수록 행복지수가 높은 곳이다.
한국을 비롯한 대부분의 아시아 지역은 중위권에 머물고 있음을
녹색으로 보여주고 있다. 그에 반해 내전이나 빈곤, 정치적 문제가 자주 발생하는
팔레스타인 지역과 남아프리카 지역은 색깔이 많이 밝아 행복지수가 낮은 곳임을
지도를 보고 알 수 있다.

분석결과를 바탕으로 예측 해보기

미래 예측 - 어떤 기준이 앞으로의 행복에 더 큰 영향을 줄까?


```
# Importing the libraries
from sklearn.model_selection import train_test_split # for data validation

# Models
from sklearn.linear_model import LinearRegression, BayesianRidge, LassoLars
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
import xgboost as xgb
from xgboost import XGBRegressor

# Metrics and Grid Search
from sklearn import model_selection, metrics
from sklearn.model_selection import GridSearchCV
```

LinearRegression, BayesianRidge, LassoLars, RandomForestRegressor, DecisionTreeRegressor, XGBRegressor와 같이 6개 모델의 절대 에러확률을 검출하기 위해 data_model을 생성한다.

```
# Creating the table
data_model = data.groupby(by= 'Country or region')['Score', 'GDP per capita',
       'Social support', 'Healthy life expectancy',
       'Freedom to make life choices', 'Generosity',
       'Perceptions of corruption'].mean().reset_index()

# Creating the dependent and independent variables
y = data_model['Score']
X = data_model[['GDP per capita',
       'Social support', 'Healthy life expectancy',
       'Freedom to make life choices', 'Generosity',
       'Perceptions of corruption']]

# Splitting the data to avoid under/overfitting
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
```

train_test_split으로 X_train, X_test, y_train, y_test 와 같은 테스트데이터와 학습 데이터를 분리 시킨다.

미래 예측 - 어떤 기준이 앞으로의 행복에 더 큰 영향을 줄까?


```
# Creating a predefined function to test the models
def modelfit(model):
    model.fit(X_train, y_train)
    preds = model.predict(X_test)
    mae = metrics.mean_absolute_error(y_test, preds)
    print('MAE:', round(mae,4))
```

modelfit()이란 이름의 Predefined function을 이용하여
각 모델 6개의 학습 성능을 평가한다.
이때 mean_absolute_error, 즉 에러가 일어날 정도를
MAE라 정하고, 각 모델들의 학습 결과를 MAE로 출력하도록 한다.
즉, MAE가 낮을 수록 신뢰할 수 있는 학습 모델인 것이다.

```
# Linear Regression

lm = LinearRegression(n_jobs = 10000)
modelfit(lm)
```

```
MAE: 0.3769
```

```
# XGBoost
xg = XGBRegressor(learning_rate=0.1, n_estimators=5000)
modelfit(xg)
```

```
[12:41:00] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprec
ated in favor of reg:squarederror.
MAE: 0.401
```

각 모델들을 modelfit()함수에 대입하여
MAE 값을 출력하게 한다.

```
# Random Forest Regressor

rf = RandomForestRegressor(n_jobs = 1000)
modelfit(rf)
```

```
MAE: 0.3986
```

```
# Decision Tree
dt = DecisionTreeRegressor()
modelfit(dt)
```

```
MAE: 0.5293
```

```
# Bayesian Linear Model
br = BayesianRidge(n_iter=1000, tol = 0.5)
modelfit(br)
```

```
MAE: 0.3723
```

```
# Lasso Lars
ls = LassoLars()
modelfit(ls)
```

```
MAE: 0.9779
```

미래 예측 - 어떤 기준이 앞으로의 행복에 더 큰 영향을 줄까?

MAE

LinearRegression : 0.3769

BayesianRidge : 0.3723

LassoLars : 0.9779

RandomForestRegressor : 0.3986

DecisionTreeRegressor : 0.5293

XGBRegressor : 0.401

MAE가 가장 작은 수인 0.3723을 보이는
BayesianRidge 모델을 사용하기로 한다.
해당 모델을 final_model에 집어넣고
학습 데이터를 집어 넣어 학습 시킨다.

```
final_model = BayesianRidge(n_iter = 10, tol = 0.1, alpha_2 = 0.1)
final_model.fit(X_train, y_train)
```

```
BayesianRidge(alpha_1=1e-06, alpha_2=0.1, compute_score=False, copy_X=True,
              fit_intercept=True, lambda_1=1e-06, lambda_2=1e-06, n_iter=10,
              normalize=False, tol=0.1, verbose=False)
```

Permutation Importance를 이용해
예측 모델로 선정한 BayesianRidge모델을
estimator에 집어넣는다.
이후 show_weight를 이용해
결과를 columns.tolist()로 표현한다.

```
# How important is each variable into predicting the overall Happiness Score?

import eli5
from eli5.sklearn import PermutationImportance

perm = PermutationImportance(estimator=final_model, random_state=1)
perm.fit(X_test, y_test)

eli5.show_weights(estimator= perm, feature_names = X_test.columns.tolist())
```

미래 예측 - 어떤 기준이 앞으로의 행복에 더 큰 영향을 줄까?

Weight	Feature
0.3884 ± 0.1075	GDP per capita
0.1527 ± 0.0554	Social support
0.1069 ± 0.0420	Freedom to make life choices
0.0430 ± 0.0187	Healthy life expectancy
0.0101 ± 0.0072	Perceptions of corruption
0.0077 ± 0.0039	Generosity

기존의 6가지 columns가 보여주고 있는 영향력, 즉 weight와
해당 weight에 대해 앞으로 상승하거나 하락할 오차범위를 예측하여 보여준다.
GDP per capita, 즉 개인의 총생산량이 기존 weight도 높고, 상승 하락의 예측폭도 크다.

Social support의 최대치와 GDP per capita의 최하치 예측이 비슷한 값인 것으로 보아
'Social Support 의 영향력이 점점 커지고 있다.' 라는
이전의 데이터 분석이 실현될 가능성도 보이고 있다.

마무리

느낀 점

느낀 점

Kaggle에서 데이터셋을 본 뒤 코드를 하나하나 읽고 해석도 해보고, 검색까지 해가며
각각의 라이브러리나 함수가 무엇을 하는 것인지,
그것을 통해 보여주고자 하는 것이 무엇인지에 집중하고자 했다.
처음 고민처럼 행복이란 단어에 대해 생각했던 것과는
약간 멀게 느껴지는 물리적 환경의 행복 요소가 대부분이었다.
물론 물리적인 요소도 굉장히 중요한 행복의 기준이지만,
궁금증이 다 해소된것은 아니라서 좀 더 자료조사와 탐색을 해볼 필요를 느꼈다.
경제성장이나 개인의 총생산량이 크게 행복지수에 작용하는 것을 볼 수 있었지만,
시간이 지날수록 돈보다 개인의 건강이나 가족, 사회에 대한
행복지수 영향력이 커지는 것을 보고 앞으로의 미래가 어떻게 변할지
조금 상상해보기도 했다. 매년마다 나오고 있는 전세계 행복지수.
단순히 행복하다라는 지표뿐 아니라
전세계가 어떠한 모습으로 변해가는지를 보여줄 수 있는 척도라고 느낄 수 있었다.

감사합니다