from chart\_studio.plotly import plot, iplot as py

import plotly.graph\_objects as go

import seaborn as sns

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from plotly.offline import download\_plotlyjs, init\_notebook\_mode, plot, iplot

import time

import warnings

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans, AgglomerativeClustering, AffinityPropagation

from sklearn.mixture import GaussianMixture

import os

import sys

from plotly.offline import iplot

import os

sns.set\_style('dark')

os.getcwd()

os.chdir(r"C:\Users\augus\OneDrive\Desktop\SGH Warszawa\Praca zaliczeniowa 2\Data preparation and graphs")

HappyReport2015 = pd.read\_csv("2015.csv")

HappyReport2016 = pd.read\_csv("2016.csv")

HappyReport2017 = pd.read\_csv("2017.csv")

HappyReport2018 = pd.read\_csv("2018.csv")

HappyReport2019 = pd.read\_csv("2019.csv")

PolandShort = happy1519[happy1519["Country"].str.contains("Poland")]

Poland = PolandShort[["Country", "Happiness Rank", "Year"]]

Poland

#Import Data Sets

os.getcwd()

os.chdir(r"C:\Users\augus\OneDrive\Desktop\SGH Warszawa\Praca zaliczeniowa 2\World Happines Report Dataset")

print("Current Working Directory " , os.getcwd())

WH\_2015 = pd.read\_csv(r"C:\Users\augus\OneDrive\Desktop\SGH Warszawa\Praca zaliczeniowa 2\World Happines Report Dataset\2015.csv", lineterminator='\r')

WH\_2016 = pd.read\_csv(r"C:\Users\augus\OneDrive\Desktop\SGH Warszawa\Praca zaliczeniowa 2\World Happines Report Dataset\2016.csv", lineterminator='\r')

WH\_2017 = pd.read\_csv(r"C:\Users\augus\OneDrive\Desktop\SGH Warszawa\Praca zaliczeniowa 2\World Happines Report Dataset\2017.csv", lineterminator='\r')

WH\_2018 = pd.read\_csv(r"C:\Users\augus\OneDrive\Desktop\SGH Warszawa\Praca zaliczeniowa 2\World Happines Report Dataset\2018.csv", lineterminator='\r')

WH\_2019 = pd.read\_csv(r"C:\Users\augus\OneDrive\Desktop\SGH Warszawa\Praca zaliczeniowa 2\World Happines Report Dataset\2019.csv", lineterminator='\r')

WH\_2015.describe()

WH\_2016.describe()

WH\_2017.describe()

WH\_2017.describe()

WH\_2018.describe()

WH\_2019.describe()

#Happiness Rank in 2017 data = object. The following is to convert it to float64

WH\_2017["Happiness Rank"] = pd.to\_numeric(WH\_2017["Happiness Rank"], errors='coerce')

print("Dimension of dataset: wh.shape")

WH\_2015.dtypes

WH\_2016.dtypes

WH\_2017.dtypes

WH\_2018.dtypes

WH\_2019.dtypes

WH\_2015.head(0)

WH\_2016.head(0)

WH\_2017.head(1)

WH\_2018.head(0)

WH\_2019.head(0)

#Generating single Heat Maps

HeatMap2015 = WH\_2015[['Happiness Score', 'Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)', 'Generosity']] #Subsetting the data

cor2015 = HeatMap2015.corr() #Calculate the correlation of the above variables

sns.heatmap(cor2015, square = True, linewidths=.5) #Plot the correlation as heat map

HM2015 = sns.heatmap(cor2015, square = True, linewidths= 0.5, annot = True, cbar = False, fmt=".1f")

plt.show()

HeatMap2016 = WH\_2016[['Happiness Score', 'Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)', 'Generosity']] #Subsetting the data

cor2016 = HeatMap2016.corr() #Calculate the correlation of the above variables

sns.heatmap(cor2016, square = True, linewidths=.5) #Plot the correlation as heat map

HM2016 = sns.heatmap(cor2016, square = True, linewidths= 0.5, annot = True, cbar = False, fmt=".1f")

plt.show()

HeatMap2017 = WH\_2017[['Happiness Score', 'Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)', 'Generosity']] #Subsetting the data

cor2017 = HeatMap2017.corr() #Calculate the correlation of the above variables

sns.heatmap(cor2017, square = True, linewidths=.5) #Plot the correlation as heat map

HM2017 = sns.heatmap(cor2017, square = True, linewidths= 0.5, annot = True, cbar = False, fmt=".1f")

plt.show()

HeatMap2018 = WH\_2018[['Happiness Score', 'Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)', 'Generosity']] #Subsetting the data

cor2018 = HeatMap2018.corr() #Calculate the correlation of the above variables

sns.heatmap(cor2018, square = True, linewidths=.5) #Plot the correlation as heat map

HM2018 = sns.heatmap(cor2018, square = True, linewidths= 0.5, annot = True, cbar = False, fmt=".1f")

plt.show()

HeatMap2019 = WH\_2019[['Happiness Score', 'Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)', 'Generosity']] #Subsetting the data

cor2019 = HeatMap2019.corr() #Calculate the correlation of the above variables

sns.heatmap(cor2019, square = True, linewidths=.5) #Plot the correlation as heat map

HM2019 = sns.heatmap(cor2019, square = True, linewidths= 0.5, annot = True, cbar = False, fmt=".1f")

plt.show()

#Creating World Maps

data2015 = dict(type = 'choropleth',

           locations = WH\_2015['Country'],

           locationmode = 'country names',

           z = WH\_2015['Happiness Score'],

           text = WH\_2015['Country'],

           colorbar = {'title':'Happiness'})

layout2015 = dict(title = 'Happiness Index 2015',

             geo = dict(showframe = False,

                       projection = {'type': 'mercator'}))

choromap2015 = go.Figure(data = [data2015], layout=layout2015)

plot(choromap2015)

data2016 = dict(type = 'choropleth',

           locations = WH\_2016['Country'],

           locationmode = 'country names',

           z = WH\_2016['Happiness Score'],

           text = WH\_2016['Country'],

           colorbar = {'title':'Happiness'})

layout2016 = dict(title = 'Happiness Index 2016',

             geo = dict(showframe = False,

                       projection = {'type': 'mercator'}))

choromap2016 = go.Figure(data = [data2016], layout=layout2016)

plot(choromap2016)

data2017 = dict(type = 'choropleth',

           locations = WH\_2017['Country'],

           locationmode = 'country names',

           z = WH\_2017['Happiness Score'],

           text = WH\_2017['Country'],

           colorbar = {'title':'Happiness'})

layout2017 = dict(title = 'Happiness Index 2017',

             geo = dict(showframe = False,

                       projection = {'type': 'mercator'}))

choromap2017 = go.Figure(data = [data2017], layout=layout2017)

plot(choromap2017)

data2018 = dict(type = 'choropleth',

           locations = WH\_2018['Country'],

           locationmode = 'country names',

           z = WH\_2018['Happiness Score'],

           text = WH\_2018['Country'],

           colorbar = {'title':'Happiness'})

layout2018 = dict(title = 'Happiness Index 2018',

             geo = dict(showframe = False,

                       projection = {'type': 'mercator'}))

choromap2018 = go.Figure(data = [data2018], layout=layout2018)

plot(choromap2018)

data2019 = dict(type = 'choropleth',

           locations = WH\_2019['Country'],

           locationmode = 'country names',

           z = WH\_2019['Happiness Score'],

           text = WH\_2019['Country'],

           colorbar = {'title':'Happiness'})

layout2019 = dict(title = 'Happiness Index 2019',

             geo = dict(showframe = False,

                       projection = {'type': 'mercator'}))

choromap2019 = go.Figure(data = [data2019], layout=layout2019)

plot(choromap2019)

#Clustering

ss.fit\_transform(HeatMap2015)

ss = StandardScaler()

C\_2015 = ss.fit\_transform(HeatMap2015)

C\_2016 = ss.fit\_transform(HeatMap2016)

C\_2017 = ss.fit\_transform(HeatMap2017)

C\_2018 = ss.fit\_transform(HeatMap2018)

C\_2019 = ss.fit\_transform(HeatMap2019)

x = pd.DataFrame(C\_2015)

HeatMap2015.dropna()

#Plot the clusters obtained using k means

fig = plt.figure()

ax = fig.add\_subplot(111)

scatter = ax.scatter(HeatMap2015['Economy (GDP per Capita)'],HeatMap2015['Trust (Government Corruption)'],

                     c=kmeans2015[0],s=50)

ax.set\_title('K-Means Clustering')

ax.set\_xlabel('GDP per Capita')

ax.set\_ylabel('Corruption')

plt.colorbar(scatter)

plt.show()

fig = plt.figure()

ax = fig.add\_subplot(111)

scatter = ax.scatter(ClusterDropNA2016['Economy (GDP per Capita)'],ClusterDropNA2016['Trust (Government Corruption)'],

                     c=kmeans[0],s=50)

ax.set\_title('K-Means Clustering 2016')

ax.set\_xlabel('GDP per Capita')

ax.set\_ylabel('Corruption')

plt.colorbar(scatter)

plt.show()

fig = plt.figure()

ax = fig.add\_subplot(111)

scatter = ax.scatter(ClusterDropNA2017['Economy (GDP per Capita)'],ClusterDropNA2017['Trust (Government Corruption)'],

                     c=kmeans[0],s=50)

ax.set\_title('K-Means Clustering 2017')

ax.set\_xlabel('GDP per Capita')

ax.set\_ylabel('Corruption')

plt.colorbar(scatter)

plt.show()

fig = plt.figure()

ax = fig.add\_subplot(111)

scatter = ax.scatter(ClusterDropNA2018['Economy (GDP per Capita)'],ClusterDropNA2018['Trust (Government Corruption)'],

                     c=kmeans[0],s=50)

ax.set\_title('K-Means Clustering 2018')

ax.set\_xlabel('GDP per Capita')

ax.set\_ylabel('Corruption')

plt.colorbar(scatter)

plt.show()

fig = plt.figure()

ax = fig.add\_subplot(111)

scatter = ax.scatter(ClusterDropNA2019['Economy (GDP per Capita)'],ClusterDropNA2019['Trust (Government Corruption)'],

                     c=kmeans[0],s=50)

ax.set\_title('K-Means Clustering 2019')

ax.set\_xlabel('GDP per Capita')

ax.set\_ylabel('Corruption')

plt.colorbar(scatter)

plt.show()

print("Średnia wartość PKB w roku 2015:", np.mean(HappyReport2015["Economy (GDP per Capita)"]))

print("Średnia wartość PKB w roku 2016:", np.mean(HappyReport2016["Economy (GDP per Capita)"]))

print("Średnia wartość PKB w roku 2017:", np.mean(HappyReport2017["Economy (GDP per Capita)"]))

print("Średnia wartość PKB w roku 2018:", np.mean(HappyReport2018["Economy (GDP per Capita)"]))

print("Średnia wartość PKB w roku 2019:", np.mean(HappyReport2019["Economy (GDP per Capita)"]))

HappyReport2015

HappyReport2016

HappyReport2017 = HappyReport2017.rename(columns=({'Happiness.Rank':'Happiness Rank',

'Happiness.Score':'Happiness Score','Economy..GDP.per.Capita.':'Economy (GDP per Capita)',

'Health..Life.Expectancy.':'Health (Life Expectancy)',

'Trust..Government.Corruption.':'Trust (Government Corruption)',

'Dystopia.Residual':'Dystopia Residual','Whisker.high':'Whisker high',

'Whisker.low':'Whisker low'}))

HappyReport2018 = HappyReport2018.rename(columns=({'Overall rank':'Happiness Rank','Score':'Happiness Score',

 'Country or region':'Country', 'GDP per capita':'Economy (GDP per Capita)',

 'Healthy life expectancy':'Health (Life Expectancy)','Freedom to make life choices':'Freedom',

 'Perceptions of corruption':'Trust (Government Corruption)',"Social support":'Family'}))

HappyReport2019 = HappyReport2019.rename(columns=({'Overall rank':'Happiness Rank','Score':'Happiness Score',

 'Country or region':'Country', 'GDP per capita':'Economy (GDP per Capita)',

 'Healthy life expectancy':'Health (Life Expectancy)','Freedom to make life choices':'Freedom',

 'Perceptions of corruption':'Trust (Government Corruption)',"Social support":"Family"}))

HappyReport2018.info()

#2018 has null value in column = "Perceptions of corruption"

#Happines

plt.figure(figsize =(10,5))

sns.kdeplot(HappyReport2015['Happiness Score'], color='red', label='2015')

sns.kdeplot(HappyReport2016['Happiness Score'], color='blue', label='2016')

sns.kdeplot(HappyReport2017['Happiness Score'], color='limegreen', label='2017')

sns.kdeplot(HappyReport2018['Happiness Score'], color='orange', label='2018')

sns.kdeplot(HappyReport2019['Happiness Score'], color='pink', label='2019')

plt.title('Happiness 2015-2019', size=25)

plt.legend()

plt.show()

plt.figure(figsize=(10,5))

sns.kdeplot(HappyReport2015['Economy (GDP per Capita)'],color='red', label="2015")

sns.kdeplot(HappyReport2016['Economy (GDP per Capita)'],color='blue', label="2016")

sns.kdeplot(HappyReport2017['Economy (GDP per Capita)'],color='limegreen', label="2017")

sns.kdeplot(HappyReport2018['Economy (GDP per Capita)'],color='orange', label="2018")

sns.kdeplot(HappyReport2019['Economy (GDP per Capita)'],color='black', label="2019")

plt.title("Economy (GDP Per Capita) 2015-2019", size=20)

plt.legend()

plt.show()

plt.figure(figsize=(10,5))

sns.kdeplot(HappyReport2015['Family'],color='red', label='2015')

sns.kdeplot(HappyReport2016['Family'],color='blue', label='2016')

sns.kdeplot(HappyReport2017['Family'],color='limegreen', label='2017')

sns.kdeplot(HappyReport2018['Family'],color='orange', label='2018')

sns.kdeplot(HappyReport2019['Family'],color='black', label='2019')

plt.title('Family - Social Support Over The Years',size=20)

plt.legend()

plt.show()

plt.figure(figsize=(10,5))

sns.kdeplot(HappyReport2015['Health (Life Expectancy)'],color='red', label='2015')

sns.kdeplot(HappyReport2016['Health (Life Expectancy)'],color='blue', label='2016')

sns.kdeplot(HappyReport2017['Health (Life Expectancy)'],color='limegreen', label='2017')

sns.kdeplot(HappyReport2018['Health (Life Expectancy)'],color='orange', label='2018')

sns.kdeplot(HappyReport2019['Health (Life Expectancy)'],color='pink', label='2019')

plt.title('Health (Life Expectancy) Over The Years',size=20)

plt.legend()

plt.show()

plt.figure(figsize=(10,5))

sns.kdeplot(HappyReport2015['Freedom'],color='red', label='2015')

sns.kdeplot(HappyReport2016['Freedom'],color='blue', label='2016')

sns.kdeplot(HappyReport2017['Freedom'],color='limegreen', label='2017')

sns.kdeplot(HappyReport2018['Freedom'],color='orange', label='2018')

sns.kdeplot(HappyReport2019['Freedom'],color='black', label='2019')

plt.title('Freedom To Choose Things 2015-2019',size=20)

plt.legend()

plt.show()

plt.figure(figsize=(10,5))

sns.kdeplot(HappyReport2015['Trust (Government Corruption)'],color='red', label='2015')

sns.kdeplot(HappyReport2016['Trust (Government Corruption)'],color='blue', label='2016')

sns.kdeplot(HappyReport2017['Trust (Government Corruption)'],color='limegreen', label='2017')

sns.kdeplot(HappyReport2018['Trust (Government Corruption)'],color='orange', label='2018')

sns.kdeplot(HappyReport2019['Trust (Government Corruption)'],color='black', label='2019')

plt.title('Perceptions Of Corruption 2015-2019',size=20)

plt.legend()

plt.show()

plt.figure(figsize=(10,5))

sns.kdeplot(HappyReport2015['Generosity'],color='red', label='2015')

sns.kdeplot(HappyReport2016['Generosity'],color='blue', label='2016')

sns.kdeplot(HappyReport2017['Generosity'],color='limegreen', label='2017')

sns.kdeplot(HappyReport2018['Generosity'],color='orange', label='2018')

sns.kdeplot(HappyReport2019['Generosity'],color='black', label='2019')

plt.title('Generosity 2015-2019',size=20)

plt.legend()

plt.show()

ScoreList = list(HappyReport2019.Score)

for i in range(0,5):

    print(ScoreList[i])

dataAbove=[]

dataBelow=[]

for i in range(len(ScoreList)):

    if ScoreList[i]>np.mean(ScoreList):

        dataAbove.append(ScoreList[i])

    else:

        dataBelow.append(ScoreList[i])

print(len(dataAbove))

print(len(dataBelow))

HappyReport2019.drop(["Overall rank", "Country or region"], axis=1, inplace=True)

HappyReport2019.Score=[1 if each>np.mean(ScoreList) else 0 for each in HappyReport2019.Score]

y=HappyReport2019.Score.values

x = HappyReport2019["GDP per capita"]

x\_data=HappyReport2019.drop(["Score", "Social support", "Healthy life expectancy", "Freedom to make life choices","Generosity","Perceptions of corruption"],axis=1)

x=(x\_data - np.min(x\_data))/(np.max(x\_data) - np.min(x\_data)).values

scaler = StandardScaler()

x\_train = scaler.fit\_transform(x\_train)

model = LogisticRegression(solver='liblinear', random\_state=0).fit(x, y)

model.classes\_

model.intercept\_

model.coef\_

model.predict\_proba(x)

model.predict(x)

model.score(x, y)

confusion\_matrix(y, model.predict(x))

cm = confusion\_matrix(y, model.predict(x))

fig, ax = plt.subplots(figsize=(8, 8))

ax.imshow(cm)

ax.grid(False)

ax.xaxis.set(ticks=(0, 1), ticklabels=('Predicted 0s', 'Predicted 1s'))

ax.yaxis.set(ticks=(0, 1), ticklabels=('Actual 0s', 'Actual 1s'))

ax.set\_ylim(1.5, -0.5)

for i in range(2):

    for j in range(2):

        ax.text(j, i, cm[i, j], ha='center', va='center', color='red')

plt.show()