한화에어로스페이스 스마트 데이터 분석과정 2조 최종프로젝트

저희 조는 다른 나라의 방산 제품 수출을 예측하여 그 나라에 맞는 솔루션을 주는 것을 목표로 설정했습니다. 그 과정에서 필요한 데이터로 정치, 경제, 분쟁도 데이터가 필요할 것이라 생각했고, 해당 데이터를 구해서 데이터 분석을 진행하였습니다.

데이터는 월드뱅크나 스톡홀름 국제 연구소, ucdp 등 사이트에서 경제, 정치, 분쟁도 데이터를 구해서 분석 진행했습니다.

여기서 구한 데이터중에서 회의를 통하여 방산 수출과 관련된 지표를 선정하여 분석에 이용하기로 하였습니다. 먼저 경제지표에서는 GDP, GDP 대비 군사비 지출 비율, GDP 대비 공공 부채 비율, 외환보유액, CPI, 실업률, 무역수지, 국제 자본 흐름, 소득 불평등 지수 등 9가지 데이터를 활용하여 경제지표를 만들었습니다. 이 데이터는 1991 년 데이터부터 2020 년 데이터까지 9개 변수의 값이 존재하는 데이터로 최근 데이터일수록 가중치를 높게 주는 형식으로 나라별 연도 변수 값을 하나로 만들었습니다. 이 가중치를 주는 방법에는 사이클로이드 곡선을 활용하였습니다.

1991 년부터 2020 년까지 해당 변수 데이터에서 결측치를 보이는 값들도 있었는데, 이의 경우 흐름을 보다 자연스럽게 하기 위해서 선형 보간법으로 결측치를 처리한 후 분석에 들어가기로 하였습니다.

```python
# 사이클로이드 가중치 계산 함수
def cycloid_weight(year, start_year=1991, end_year=2020):
    # 연도를 [0, 2π] 범위로 정규화
    normalized_year = (year - start_year) / (end_year - start_year) * 2 * np.pi
    # 사이클로이드 공식 (r = 1로 설정)
    r = 1
    weight = r * (normalized_year - np.sin(normalized_year))
    return weight

# 연도 리스트 (1991~2020)
years = np.arange(1991, 2021)

# 각 연도에 대해 사이클로이드 가중치 계산
weights = np.array([cycloid_weight(year) for year in years])

# 가중치를 (31,) 크기 배열로 만듦 (연도에 해당하는 가중치)
weights_reshaped = weights.reshape(1, -1)

# 데이터프레임에 가중치 적용

# gdp_growth_1991_2020, gdp_mil_1991_2020 등 각 데이터프레임에 대해 가중치를 곱해줍니다.
def apply_cycloid_weight(df):
    # 'Country Name' 열을 제외한 연도별 값들에 가중치를 곱해줍니다.
    df_values = df.drop(columns='Country Name').values
    # 가중치를 (국가 x 연도) 배열에 적용
    weighted_values = df_values * weights_reshaped
    # 가중치가 적용된 값들을 데이터프레임으로 변환
    weighted_df = pd.DataFrame(weighted_values, columns=df.columns[1:], index=df['Country Name'])
    return weighted_df

# 각 데이터프레임에 가중치 적용
gdp_growth_weighted = apply_cycloid_weight(gdp_growth_1991_2020)
gdp_mil_weighted = apply_cycloid_weight(gdp_mil_1991_2020)
int_cap_weighted = apply_cycloid_weight(int_cap_1991_2020)
income_weighted = apply_cycloid_weight(income_1991_2020)
trade_weighted = apply_cycloid_weight(trade_1991_2020)
unemp_weighted = apply_cycloid_weight(unemp_1991_2020)
cpi_weighted = apply_cycloid_weight(cpi_1991_2020)
dollor_weighted = apply_cycloid_weight(dollor_1991_2020)
gdp_dept_weighted = apply_cycloid_weight(gdp_dept_1991_2020)
```

이후 이 변수들을 가지고 중요해보이는 변수에는 가중치를 높게 주고 그렇지 않은 변수는 가중치를 적게 주어서 경제 지표를 나라마다 한 값으로 만들고, 그 값을 로그변환 시킨 후 스케일링을 통해 점수로 만들었습니다. 점수화를 시키는 과정에서 어느 나라도 완벽하거나 최악인 나라는 없을 것이라 판단하고, 최고 점수를 80, 최저 점수를 20을 줘서 스케일링 후 점수로 표현하였습니다.

```python
weights = {
    'GDP Growth Weighted': 0.2,        # GDP 비중을 적당히 유지
    'GDP Military Weighted': 0.1,      # 군사력 비중을 어느 정도 반영
    'Int. Cap Weighted': 0.1,          # 해외 투자 비중 유지
    'Income Weighted': 0.25,           # 소득 불평등 비중은 높게 설정
    'Trade Weighted': 0.1,             # 무역 비중은 적당히 반영
    'Unemployment Weighted': 0.1,      # 실업률 비중을 적당히 반영
    'CPI Weighted': 0.05,              # 물가 상승률 비중은 낮게 설정
    'Dollar Weighted': 0.1,            # 외화 보유량 비중을 적당히 반영
    'GDP Debt Weighted': 0.1           # 부채 비중은 적당히 설정
}

# 경제 지표 계산
merged_df['Economic Indicator'] = (
    merged_df['GDP Growth Weighted'] * weights['GDP Growth Weighted'] +
    merged_df['GDP Military Weighted'] * weights['GDP Military Weighted'] +
    merged_df['Int. Cap Weighted'] * weights['Int. Cap Weighted'] +
    merged_df['Income Weighted'] * weights['Income Weighted'] +
    merged_df['Trade Weighted'] * weights['Trade Weighted'] +
    merged_df['Unemployment Weighted'] * weights['Unemployment Weighted'] +
    merged_df['CPI Weighted'] * weights['CPI Weighted'] +
    merged_df['Dollar Weighted'] * weights['Dollar Weighted'] +
    merged_df['GDP Debt Weighted'] * weights['GDP Debt Weighted']
)

# 'Economic Indicator' 기준으로 내림차순 정렬
merged_df = merged_df.sort_values(by='Economic Indicator', ascending=False)

# 순위(Rank) 열 추가
merged_df['Rank'] = range(1, len(merged_df) + 1)
```

```python
from sklearn.preprocessing import MinMaxScaler

# 'Economic Indicator' 열을 로그 변환한 값(Log_Economic_Indicator)을 20과 80 사이로 스케일링
scaler = MinMaxScaler(feature_range=(20, 80))
merged_df['Scaled Economic Indicator'] = scaler.fit_transform(merged_df[['Log_Economic_Indicator']])

# 'Scaled Economic Indicator' 기준으로 내림차순 정렬
merged_df = merged_df.sort_values(by='Scaled Economic Indicator', ascending=False)

# 상위 30개 국가 출력
top_30 = merged_df[['Country Name', 'Log_Economic_Indicator', 'Scaled Economic Indicator', 'Rank']]

# 출력에 반복문 사용
for index, row in top_30.iterrows():
    print(f"Country Name: {row['Country Name']}, Scaled Economic Indicator: {row['Scaled Economic Indicator']:.2f}, Rank: {row['Rank']}")
```

다음은 정치 안정도 점수입니다. 이 데이터에는 기본 데이터에 순위화시켜놓은 데이터가 있어서, 6개 정치 지표에 대하여 이 순위를 평균을 낸 다음 20-80 스케일링을 진행하였습니다.

```
# 모든 나라들의 'scaled_pctrank' 값을 비례적으로 스케일링
country_scores['scaled_pctrank'] = country_scores['scaled_pctrank'] * scaling_factor

# 'scaled_pctrank'의 최소값과 최대값 계산
min_value = country_scores['scaled_pctrank'].min()
max_value = country_scores['scaled_pctrank'].max()

# 새로운 최소값(20)과 최대값(80)
new_min = 20
new_max = 80

# 최소-최대 스케일링을 적용하여 [20, 80] 범위로 변환
country_scores['scaled_pctrank'] = (
    (country_scores['scaled_pctrank'] - min_value) / (max_value - min_value)  # 0~1로 정규화
) * (new_max - new_min) + new_min  # 새로운 범위로 변환
```

분쟁도 데이터는 나라, 분쟁 별 사망자의 데이터 합에 사이클로이드 가중치를 적용시켜 값을 만들었습니다. 이 과정에서 분쟁 자체가 없어 사망자가 없는 나라도 있는데, 이와 같은 나라의 경우 100점을 부여하였고, 사망자가 있는 나라들은 99점부터 0점까지 사망자 수의 역순으로 점수를 매겼습니다. 점수를 매기는 과정에서는 가중치를 매긴 값을 로그 변환을 이용하여 점수를 매겼습니다.

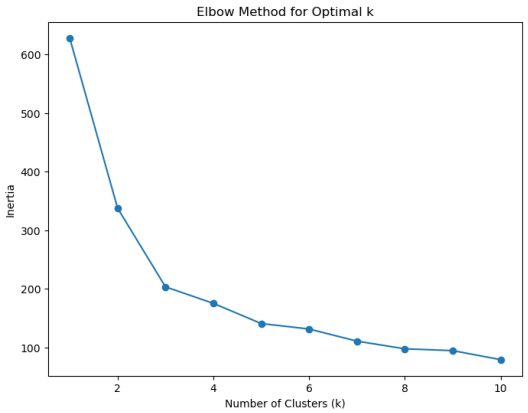
```
# 연도에 대해 사이클로이드 가중치 계산
start_year = 1989
end_year = 2020
years_range = end_year - start_year + 1

# 사이클로이드 함수에 대한 가중치 계산 (0에서 1 사이로 변환)
a = 1  # 가중치의 진폭을 설정
b = years_range  # 주기를 설정
bd['cycle_weight'] = 0.5 * (1 + np.cos(np.pi * (bd['year'] - start_year) / b))  # 0에서 1 사이로 변환

# 가중치를 적용하여 bd_high 값 계산
bd['weighted_bd_high'] = bd['bd_high'] * bd['cycle_weight']

# location별로 weighted_bd_high 값을 합산
location_bd_sum = bd.groupby('location')['weighted_bd_high'].sum().reset_index()


# 로그 변환 (0이 아닌 값들에 대해서만 적용)
location_bd_sum['log_weighted_bd_high'] = np.log(location_bd_sum['weighted_bd_high'] + 1)  # +1을 하는 이유는 0에 로그를 적용할 수 없기 때문
```

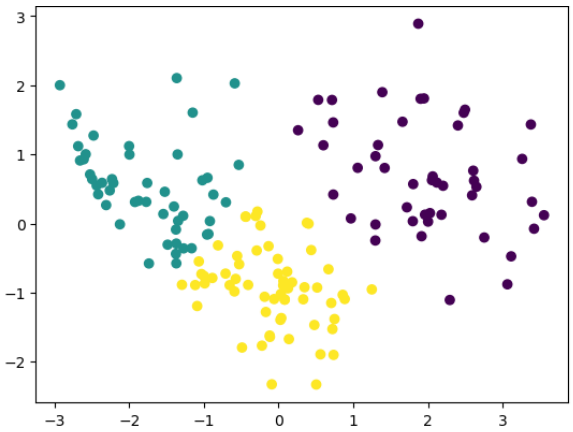
```
# MinMaxScaler를 사용하여 'log_weighted_bd_high' 열을 1과 100 사이로 스케일링
scaler = MinMaxScaler(feature_range=(1, 100))

# 'log_weighted_bd_high' 열을 1과 100 사이로 스케일링하고 결과를 'scaled_weighted_bd_high'에 저장
location_bd_sum['scaled_weighted_bd_high'] = scaler.fit_transform(location_bd_sum[['log_weighted_bd_high']])

# 'scaled_weighted_bd_high' 기준으로 내림차순 정렬
location_bd_sum_sorted = location_bd_sum.sort_values(by='scaled_weighted_bd_high', ascending=False)
```

```
location_bd_sum_sorted['scaled_weighted_bd_high'] = 100 - location_bd_sum_sorted['scaled_weighted_bd_high']

# 소수점 둘째 자리까지 출력
location_bd_sum_sorted['scaled_weighted_bd_high'] = location_bd_sum_sorted['scaled_weighted_bd_high'].round(2)
```

이렇게 3개의 지표를 점수화시킨 후, 3개의 점수를 평균을 내어 해당 국가의 점수를 만들었습니다. 그리고 국가 등급도 매겨보았습니다. 국가 등급을 매기는 방법은 엘보우 방법을 사용하여 몇 개의 군집이 가장 적절할지 고르고, 3개의 군집이 가장 좋다는 결과가 나와서 3개의 군집으로 knn 방법으로 군집화하였습니다. 이 군집을 바탕으로 각 국가에 A,B ,C 등급으로 국가 등급을 매겨보았습니다.

```python
# 데이터 준비 (정규화된 데이터 사용)
data = df[['scaled_weighted_bd_high', 'Scaled Economic Indicator', 'scaled_pctrank', 'average_score']]
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)

# KMeans 군집화 (군집 수 3)
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(data_scaled)

# 군집 레이블을 'Cluster' 열에 추가 (기존의 'cluster' 열이 있으면 삭제)
df['Cluster'] = kmeans.labels_

# 중복된 'cluster' 열 삭제
df = df.drop(columns=['cluster'], errors='ignore')

# 결과 확인
print(df.head())

# 군집화된 데이터 시각화 (2D로 차원 축소 후 시각화)
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
data_pca = pca.fit_transform(data_scaled)
```

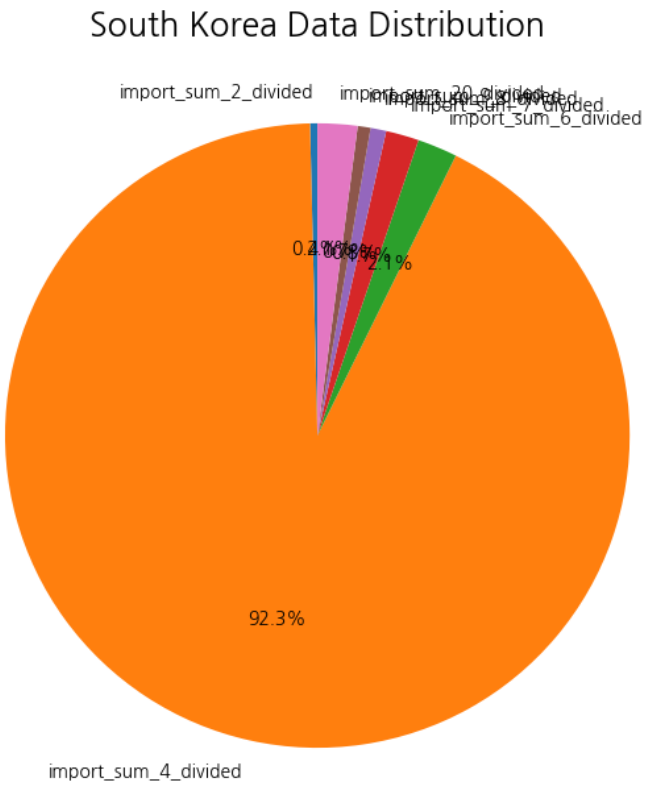




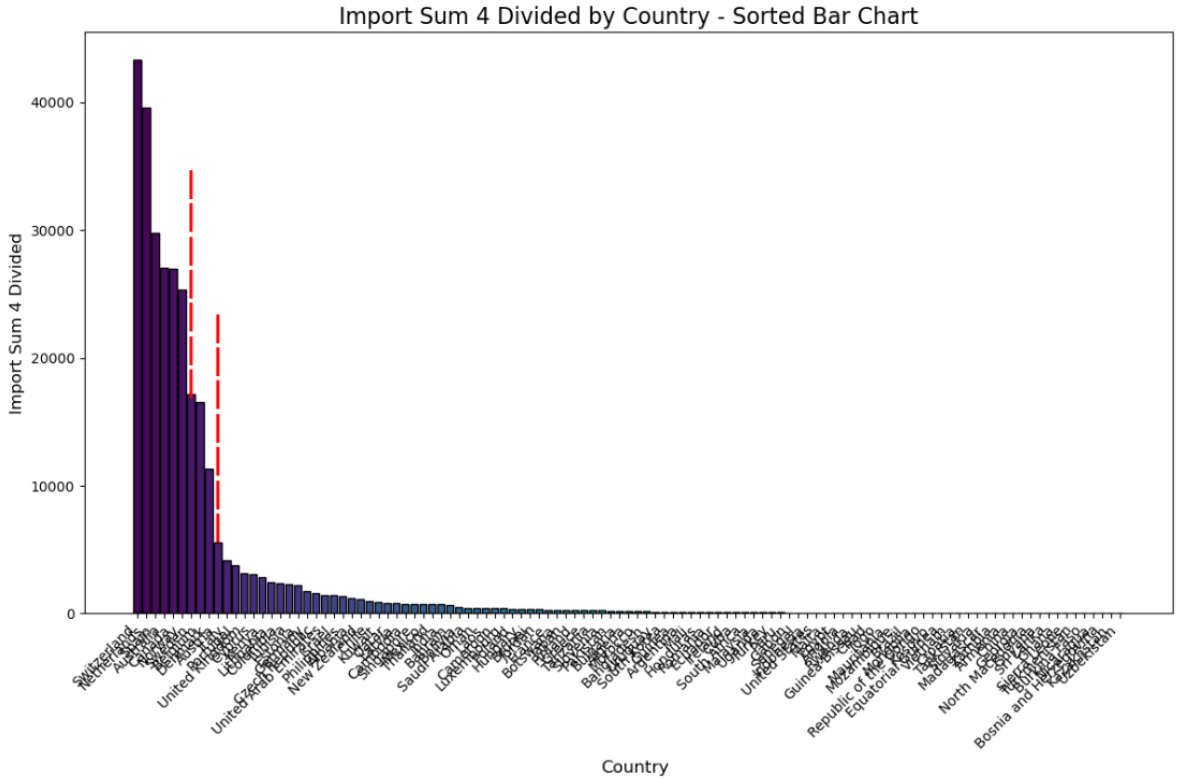
그 후 무기 수입 데이터와 무기 보유 데이터를 가지고 무기 보유 대비 수입으로 그 나라가 무기 체계를 얼마나 수입에 의존하고 있는지를 나타내어 보았습니다. 그 후 특정 나라의 무기 수입 체계나 특정 카테고리에서 어떤 나라가 수입에 많이 의존하는지를 조회해 보았습니다.

수입 체계의 비율을 조회한 원그래프 옆에는 저희가 점수 매긴 것과 GDP 를 함께 볼 수 있도록 표시하였습니다.

무기 수입 의존도 그래프에서는 상위권 나라와 중위권 나라 부분에 점선으로 표시를 해보았고, 그 밑에 해당 나라와 짧은 요약 코멘트를 달아보았습니다.



항목	값
분쟁 안정도 점수	100.00
경제지표 점수	66.67
정치 안정도 점수	64.12
평균 점수	76.93
국가 등급	A
GDP (단위 : 억)	17130




```
상위 6개 나라:
['Switzerland', 'Netherlands', 'Spain', 'Australia', 'Canada', 'Norway']
상위 9개 나라:
['Switzerland', 'Netherlands', 'Spain', 'Australia', 'Canada', 'Norway', 'Sweden', 'Belgium', 'Denmark']
```

Import Sum 4 코멘트

카테고리 4에서는 스위스와 네덜란드가 다른 나라에 비해 많은 보유량 대비 수요량이 그래프에 보이는 것을 확인할 수 있다. 스페인, 호주, 캐나다와 노르웨이도 이 두 나라보다는 아니지만 꽤 높은 수요량이 있는 것으로 예측이 되고 있고, 스웨덴과 벨기에, 덴마크도 상위 6개 나라보다는 떨어지지만 수요량이 꽤 예측되고 있는 것을 확인할 수 있다.

이후 이 세 지표와 무기 수출에 상관관계가 있을지 확인해보기 위해서 선형회귀 모델로 분석해보았습니다. 종속변수로는 해당 국가의 총 수입량을 넣었고, 독립변수로는 분쟁도, 경제, 정치 지표와 클러스터링한 등급을 넣어 분석을 진행해보았습니다.

```python
# 종속변수와 독립변수 분리
X = df_merge.drop(columns=['Country', 'import_sum'])
y = df_merge['import_sum']

# 학습 데이터와 테스트 데이터 분리
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 수치형 변수 스케일링
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# 상수항 추가 (statsmodels는 명시적으로 상수항이 필요)
X_train_sm = sm.add_constant(X_train)
X_test_sm = sm.add_constant(X_test)

# 선형 회귀 모델 학습 (OLS)
model = sm.OLS(y_train, X_train_sm).fit()

# 모델 요약 출력 (계수와 p-값 포함)
print(model.summary())

# 테스트 데이터로 예측
y_pred = model.predict(X_test_sm)

# 모델 평가
print("R2 (Test):", r2_score(y_test, y_pred))
print("MSE (Test):", mean_squared_error(y_test, y_pred))
```

```python
X.columns
```

```
Index(['scaled_weighted_bd_high', 'Scaled Economic Indicator',
       'scaled_pctrank', 'Cluster_B', 'Cluster_C'],
      dtype='object')
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             import_sum   R-squared:                       0.366
Model:                            OLS   Adj. R-squared:                  0.340
Method:                 Least Squares   F-statistic:                     13.75
Date:                Fri, 13 Dec 2024   Prob (F-statistic):           1.35e-10
Time:                        10:22:23   Log-Likelihood:                -1466.6
No. Observations:                 125   AIC:                             2945.
Df Residuals:                     119   BIC:                             2962.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.794e+04   2764.331      6.490      0.000    1.25e+04    2.34e+04
x1         -1.335e+04   7885.513     -1.694      0.093    -2.9e+04    2259.497
x2          2.317e+04   3756.989      6.167      0.000    1.57e+04    3.06e+04
x3          1099.6534   4142.125      0.265      0.791   -7102.168    9301.475
x4          3475.2721   5100.174      0.681      0.497   -6623.582    1.36e+04
x5          -1.06e+04   9140.314     -1.160      0.249   -2.87e+04    7500.437
==============================================================================
Omnibus:                       96.736   Durbin-Watson:                   2.153
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              773.485
Skew:                           2.676   Prob(JB):                    1.10e-168
Kurtosis:                      13.948   Cond. No.                         6.78
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
R2 (Test): 0.46146672651544896
MSE (Test): 520567197.73583007
```

해당 분석 모델을 보면, 설명력이 0.366으로 그리 높게 나오지 않는 것을 볼 수 있습니다. 이는 아무래도 무기 수입건수를 비교한 결과인데 이 수입건수는 무기 체계의 가격을 고려하지 않았고, 분쟁도와 정치, 경제지표만으로는 많은 설명이 어려워서 이렇게 설명력이 나온 것으로 추정됩니다. 변수 설명력을 보면 2번째 변수인 경제지표가 p값이 0에 가까워서 변수가 의미 있는것을 확인할 수 있습니다. 그리고 분쟁도 변수의 경우 p값이 0.093으로 변수 자체는 유의해보이지 않으나, 완전히 무시하기에는 위험할 만한, 간단하게 의미가 있을 수도 있는 변수로 보여집니다. 이외의 세 변수는 p값이 너무 높게 나와서 유의미한 변수로 보여지지 않습니다. 따라서, 이 모델에서는 경제지표와 분쟁도 정도가 무기 수입에 의미가 있어 보이는 것으로 분석할 수 있습니다.