

# *PROJECT 2*

AI\_11\_김희정

# 주제: 국방비 예측 데이터

## ✂ 데이터 선정 이유 ✂

우크라이나와 러시아의 전쟁을 보며 내가 누리고 있는 안전이 언제든지 없어질 수 있다고 느꼈다.

국가와 국민을 지키기 위해 각 나라마다 국방비에 어느정도 투자하는지 궁금하여 선정함.

## ✂ 가설 ✂

GDP대비 국방비 비율과 국가지출 대비 국방비 비율이 국방비에 큰 영향을 미친다.

## ✂ 베이스라인 모델 ✂

단순선형회귀모델


✂ 2000~2020년 나라별 국방비 데이터를 이용하여 분석해 보았습니다. ✂



## 데이터 수집



<https://www.kaggle.com/datasets/prasertk/military-expenditure-by-country-from-1970-2020>


 Dataset

^

21

# Military expenditure by country from 1970-2020

50 years of military expenditure


 Prasert Kanawattanachai • updated 8 days ago (Version 1)

A country	A iso3c	A iso2c	# year	# Military expenditu...
country	3-digit country code	2-digit country code	year	Military expenditure (current USD)
	# Military expenditu...	# Military expenditu...	A adminregion	A incomeLevel
	Military expenditure (% of general government expenditure)	Military expenditure (% of GDP)	region	income level

# 데이터 전처리

# 🔪 Pandas 이용하여 데이터 전처리 🔪

```
1 df.columns = ['country', 'iso3c', 'iso2c', 'year', 'M_USD', 'M_of_gov', 'M_of_GDP', 'adminregion', 'incomeLevel']
2 df.head()
```

	country	iso3c	iso2c	year	M_USD	M_of_gov	M_of_GDP	adminregion	incomeLevel	
0	Afghanistan	AFG	AF	1970	2.939586e+06	NaN	1.629606	South Asia	Low income	
1	Afghanistan	AFG	AF	1971	NaN	NaN	NaN	South Asia	Low income	
2	Afghanistan	AFG	AF	1972	NaN	NaN	NaN	South Asia	Low income	
3	Afghanistan	AFG	AF	1973	3.341272e+06	NaN	1.868910	South Asia	Low income	
4	Afghanistan	AFG	AF	1974	3.581366e+06	NaN	1.610825	South Asia	Low income	

```
1 df1 = df[df['year'] > 1999]
2 df1 = df1.drop(['iso2c'],axis=1)
3 values = {'M_USD':0,'M_of_gov':0,'M_of_GDP':0}
4 df1 = df1.fillna(value=values)
5 pd.options.display.float_format = '{:.2f}'.format
6 df1 = df1.astype({'M_USD':'int64'})
7 df1.head(100)
```

	country	iso3c	year	M_USD	M_of_gov	M_of_GDP	adminregion	incomeLevel
30	Afghanistan	AFG	2000	0	0.00	0.00	South Asia	Low income
31	Afghanistan	AFG	2001	0	0.00	0.00	South Asia	Low income
32	Afghanistan	AFG	2002	0	0.00	0.00	South Asia	Low income
33	Afghanistan	AFG	2003	0	0.00	0.00	South Asia	Low income
34	Afghanistan	AFG	2004	12511557	16.13	2.43	South Asia	Low income

```
1 idx = df1[df1['incomeLevel']=='Aggregates'].index
2 idx3 = df1[df1['incomeLevel']=='Not classified'].index
3 df1 = df1.drop(idx)
4 df1 = df1.drop(idx3)
```

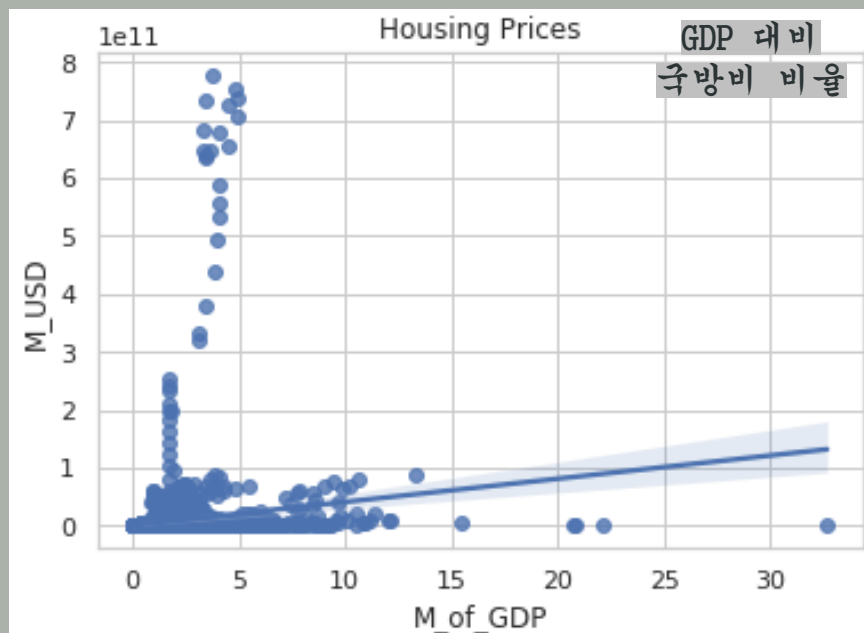
1 df1								
	country	iso3c	year	M_USD	M_of_gov	M_of_GDP	adminregion	incomeLevel
30	Afghanistan	AFG	2000	0	0.00	0.00	South Asia	Low income
31	Afghanistan	AFG	2001	0	0.00	0.00	South Asia	Low income
32	Afghanistan	AFG	2002	0	0.00	0.00	South Asia	Low income
33	Afghanistan	AFG	2003	0	0.00	0.00	South Asia	Low income
34	Afghanistan	AFG	2004	125111557	16.13	2.43	South Asia	Low income

```
1 df1 = df1[pd.notnull(df1['incomeLevel'])]
2 df1.reset_index(drop=True)
```

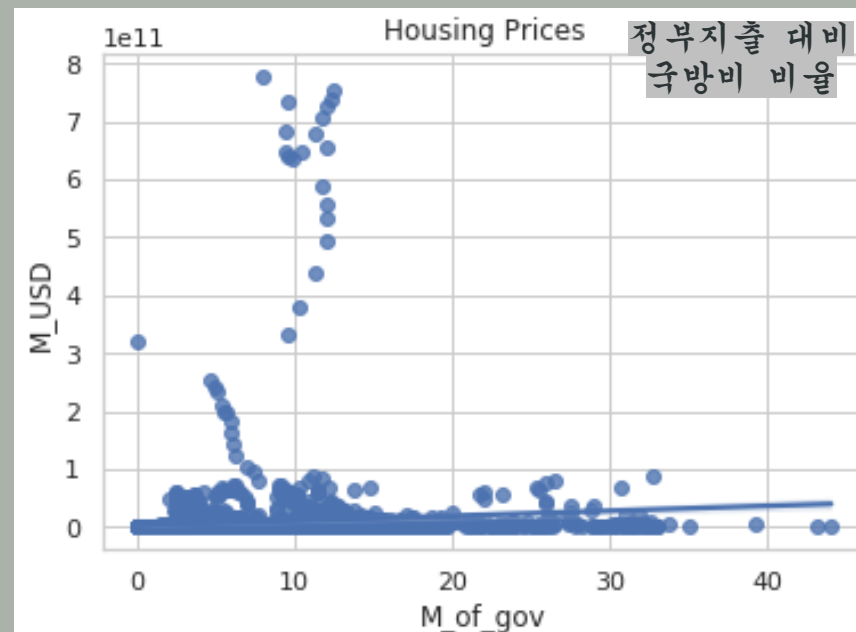
	country	iso3c	year	M_USD	M_of_gov	M_of_GDP	adminregion	incomeLevel
0	Afghanistan	AFG	2000	0	0.00	0.00	South Asia	Low income
1	Afghanistan	AFG	2001	0	0.00	0.00	South Asia	Low income
2	Afghanistan	AFG	2002	0	0.00	0.00	South Asia	Low income
3	Afghanistan	AFG	2003	0	0.00	0.00	South Asia	Low income
4	Afghanistan	AFG	2004	125111557	16.13	2.43	South Asia	Low income

# 모델링을 통한 성능 비교 방법(선형회귀)

[선형회귀: 예측하고자 하는 변수(종속변수)가 다른 특성(독립변수)과 선형관계를 이루는 것]



$R^2 : 0.02517293924773356$

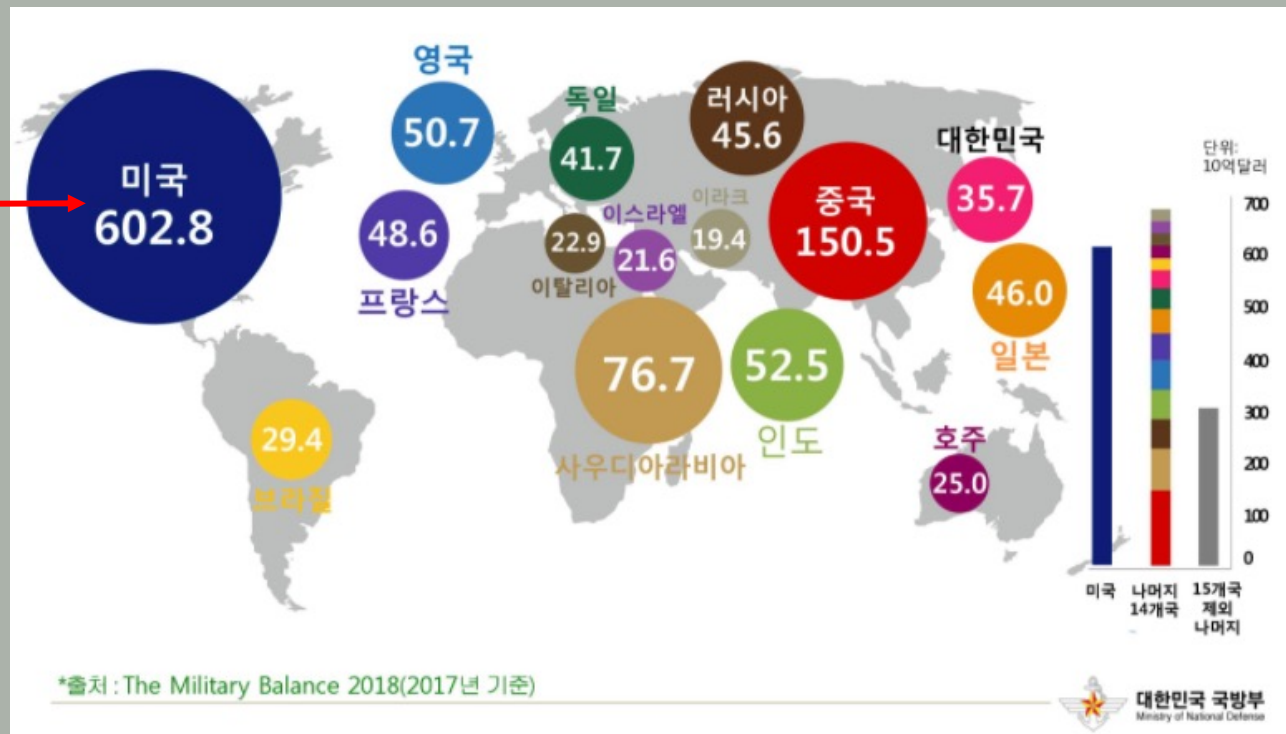
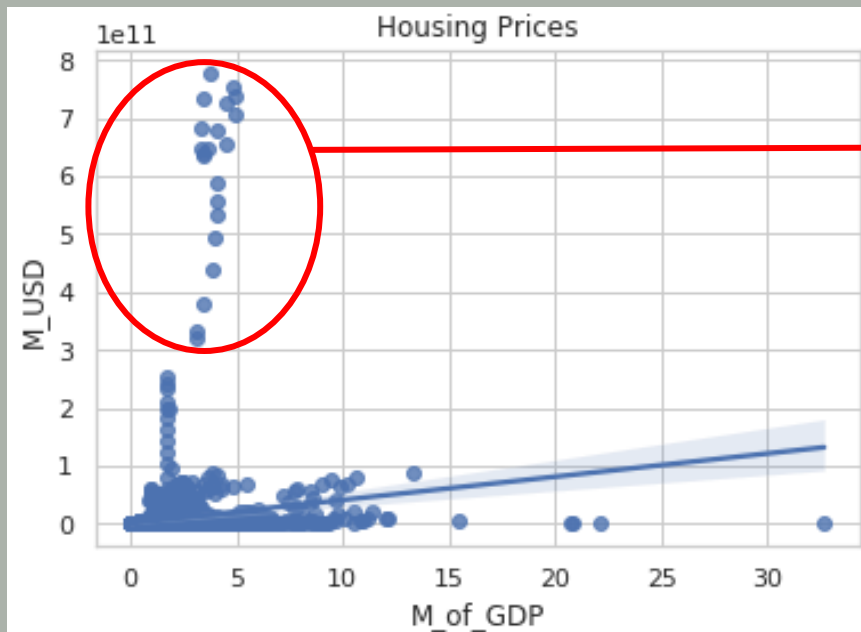


$R^2 : 0.012574960107511912$

왼쪽 그래프의  $R^2$ 값과 기울기가 더 크다.

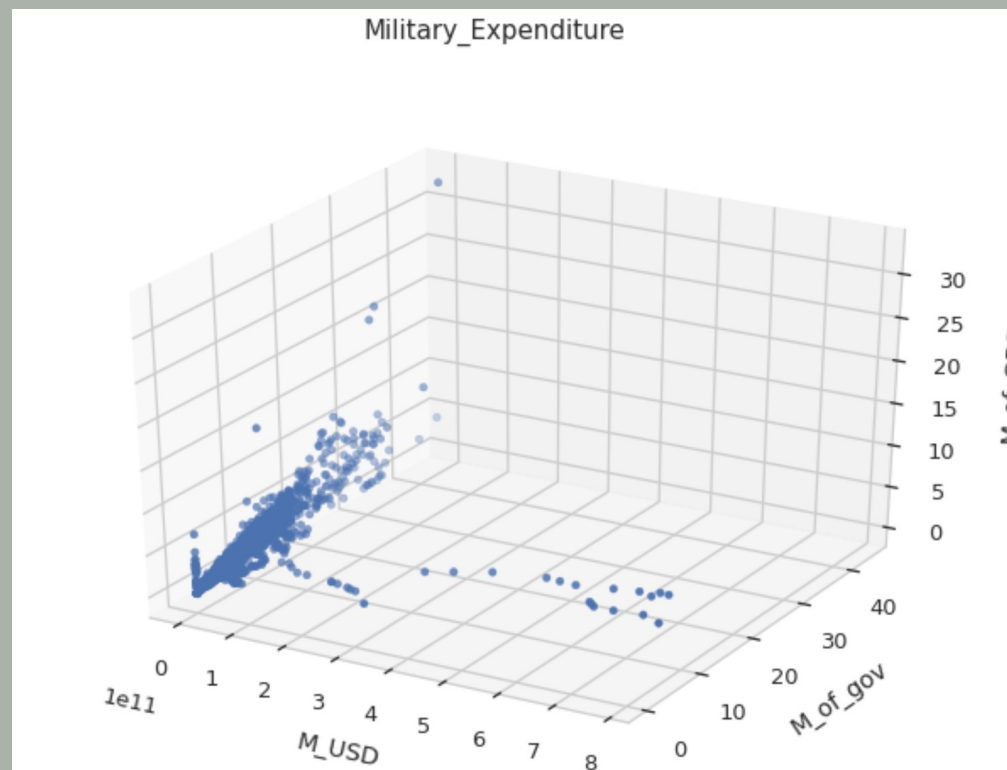


# 모델링을 통한 성능 비교 방법(선형회귀)



국방비는 그 나라의 힘을 보여주는 지표

# 모델링을 통한 성능 비교 방법(다중 선형회귀)



$R^2$  0.025941858164886034

단순선형회귀모델보다 다중선형회귀모델의 정확도가 더 높음

# 모델링을 통한 성능 비교 방법(결정트리)

## ❌결정트리 점수 확인❌

[결정트리:스무고개 하듯 특성들의 수치를 가지고 질문을 통해 정답클래스를 찾아가는 과정]

```
[150] 1 from sklearn.tree import DecisionTreeRegressor
      2 pipe = make_pipeline(DecisionTreeRegressor(min_samples_leaf=54, random_state=10))
      3 pipe.fit(X_train, y_train)
      4 print('훈련 정확도: ', pipe.score(X_train, y_train))
      5 print('테스트 정확도: ', pipe.score(X_test, y_test))
```

훈련 정확도: 0.14580786547071256  
테스트 정확도: 0.11429540876904143

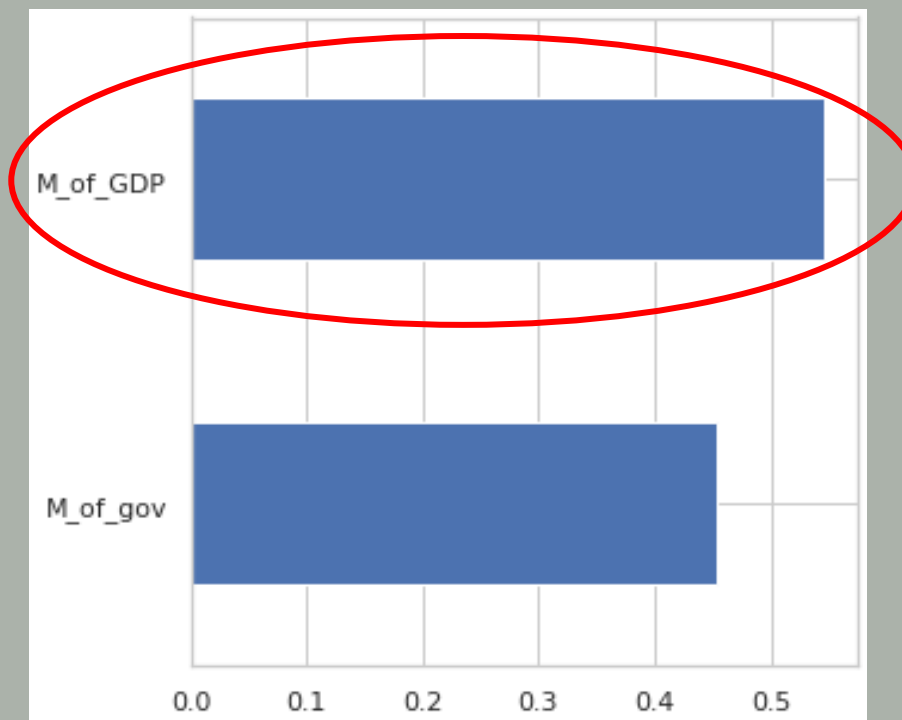
Min\_samples\_split을 이용하여 과적합 제어  
단순선형회귀모델에 비해 정확도가 낮음



# 모델링을 통한 성능 비교 방법(결정트리)

✂결정트리를 통해 특성중요도 확인✂

[결정트리:스무고개 하듯 특성들의 수치를 가지고 질문을 통해 정답클래스를 찾아가는 과정]



# 모델링을 통한 성능 비교 방법(랜덤포레스트)

## ✂랜덤포레스트 점수 확인✂

[랜덤포레스트: 훈련과정에서 구성한 다수의 결정 트리로부터 분류, 평균 예측치를 출력함]

```
[169] 1 from sklearn.ensemble import RandomForestRegressor
      2
      3 pipe = make_pipeline(RandomForestRegressor(min_samples_leaf=54, random_state=10))
      4 pipe.fit(X_train, y_train)
      5 print('훈련 정확도: ', pipe.score(X_train, y_train))
      6 print('테스트 정확도: ', pipe.score(X_test, y_test))
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/pipeline.py:394: DataConversionWarning
  self._final_estimator.fit(Xt, y, **fit_params_last_step)
```

훈련 정확도: 0.12921626223184268

테스트 정확도: 0.10846905311647859

Min\_samples\_split을 이용하여 과적합 제어  
단순선형회귀모델에 비해 정확도가 낮음

# 모델링을 통한 성능 비교 방법(하이퍼파라미터 튜닝)

[하이퍼파라미터 튜닝: 모델의 성능을 확보하기 위해 조절하는 주요 설정값]

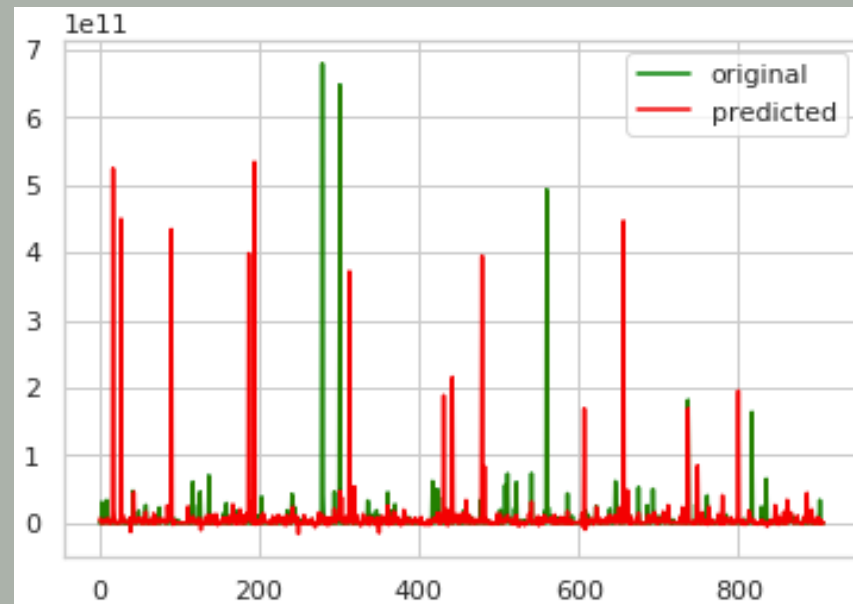
✂ 하이퍼파라미터 튜닝을 적용하여 회귀평가지표 확인하기 ✂

[회귀평가지표로 MAE, MSE, RMSE, R2이 있다. 이 지표를 보고 모델의 성능을 평가한다. 그중 R2는 0~1사이의 값으로 표현한다.]

```
1 #테스트 데이터 스코어
2 y_pred = boosting.predict(X_test)
3 MAE=mean_absolute_error(y_test, y_pred)
4 MSE=mean_squared_error(y_test, y_pred)
5 RMSE=np.sqrt(MSE)
6 R2= r2_score(y_test, y_pred)
7 print(MAE, MSE, RMSE, R2)

11511621855.084803 3.17779356358649e+21 56371921765.95091 -1.3116378337319663
```

R2값이 -1.3이므로  
성능이 아주 낮은 비정상적인 모델이다.

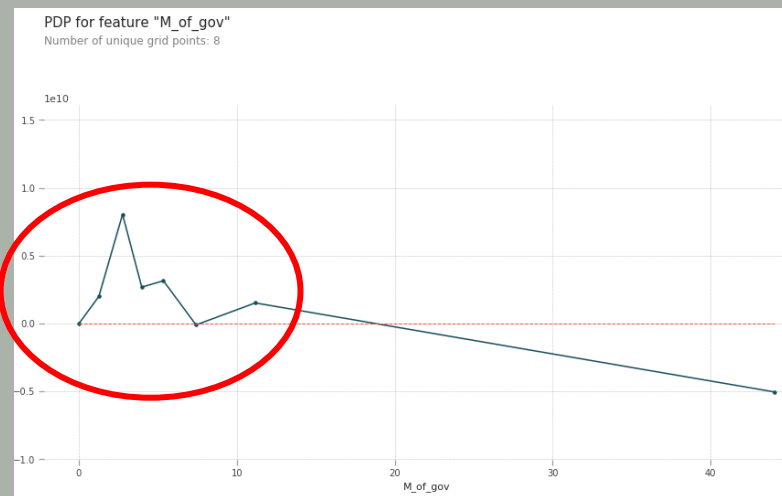




# 머신러닝 모델 해석(PDP)

## ✂PDP로 머신러닝 모델 해석하기✂

[PDP: 모델의 예측값이 특정 피처의 변화에 따라 평균적으로 어떻게 변화하는지 보여주는 그래프]

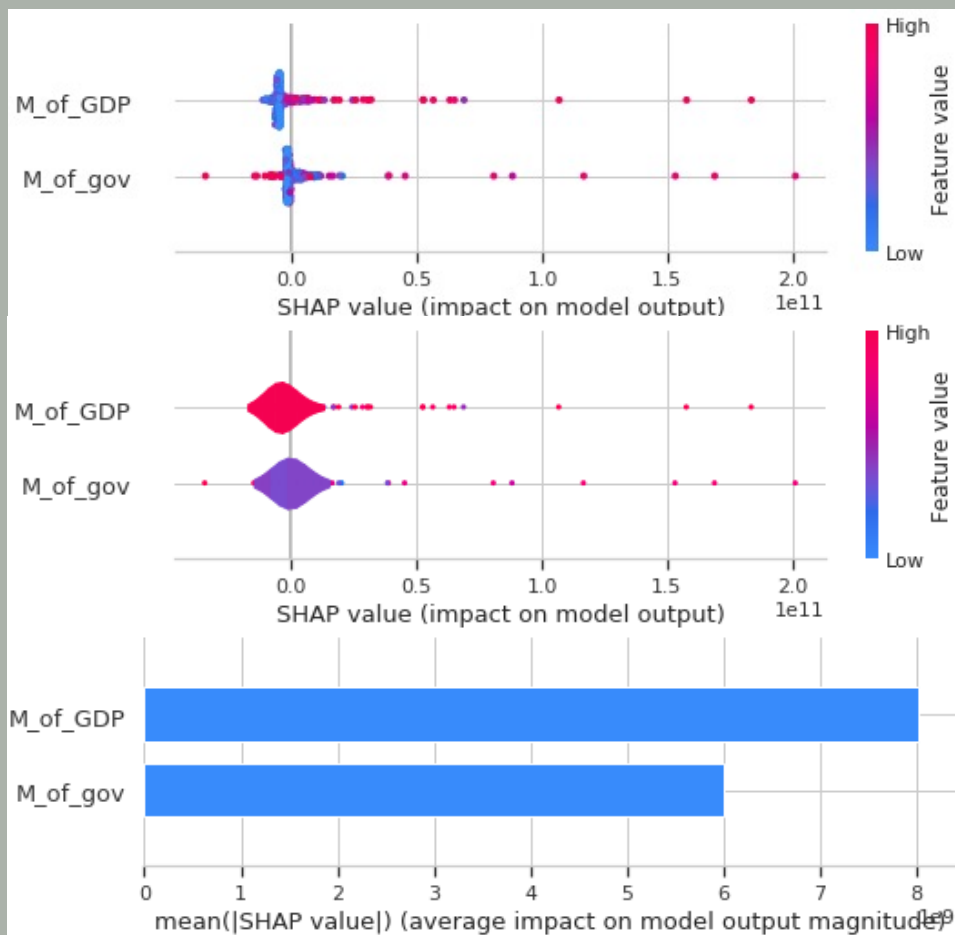


변화가 매우 불규칙적이므로 좋은 모델이 아니다.

# 머신러닝 모델 해석(SHAP)

## ✂SHAP로 머신러닝 모델 해석하기✂

[SHAP:특정 데이터에 대해 모델의 예측값에 각 피쳐들이 얼마나 기여했는지 보여줌]



전반적으로 SHAP value가 커야  
예측값에 영향을 많이 준다.  
분포가 작으므로 예측값에  
영향을 작게 준다.

# 종합 결론

## ✂결론✂

1. GDP 대비 국방비 비율( $M\_of\_GDP$ )과 국가 지출 대비 국방비 비율( $M\_of\_gov$ )은 국방비에 큰 영향을 주지 않는다.
2. 예측 성능이 낮은 모델이므로 성능이 좋은 모델이 아니다.

## ✂반성✂

1. 이상치 제거를 하면 좀 정확도가 올라갈 것이라고 생각합니다.  
->미국이란 나라도 중요하지만, 전체적인 정확도를 위해선 제거가 필수인 것 같습니다.
2. 독립변수( $M\_of\_GDP$ ,  $M\_of\_gov$ )를 이용하여 국방비(USD)단위로 맞췄다면 더 좋은 성능이 나올 것이다.
3. 데이터가 부족했고 독립변수 설정과 가설에 문제가 있었던 거 같습니다.
4. 데이터를 선정할 때 주제도 중요하지만, 다양한 정보가 있는 데이터를 선정하고 EDA와 전처리를 잘 하는 것이 중요합니다.