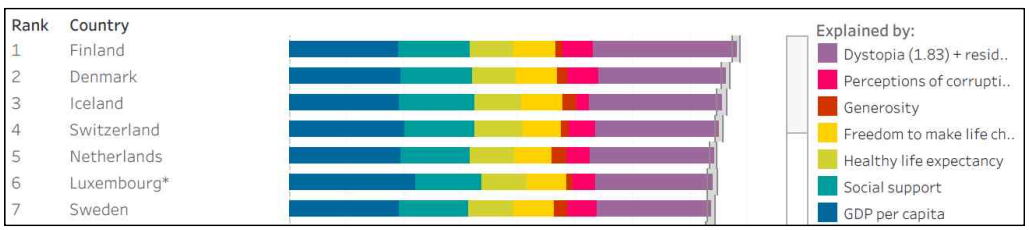
기계학습 02분반 Term project: Final Report	
주제	기계학습을 통한 행복지수 분석
팀원	황수인, 이한별, 김소민

1. 문제 정의

World Happiness Report라는 기관에서는 매년 설문조사를 시행해 150개가 넘는 국가의 사람들이 자신의 삶을 평가한 자료를 수집한다. 이들이 수집하는 자료는 [Country name, year, Life Ladder, Log GDP per capita, Social support, Healthy life expectancy at birth, Freedom to make life choices, Generosity, Perceptions of corruption, Positive affect, Negative affect, Confidence in national government]로, 총 12개의 column으로 이루어져 있다. 행복지수를 나타내는 Life Ladder는 0~10 사이의 숫자로 설문자들이 직접 정한다. 이를 평균을 낸 후, 수집한 데이터가 대표성을 갖도록 가중치가 곱해져서 최종적인 Life Ladder 값이 정해진다. 수집한 자료 중 Positive affect와 Negative affect가 가중치를 구할 때 사용된다.

[그림 1] 각 나라 설문자들의 직접 정한 Life Ladder (Average Life Evaluation)

이후, 가장 행복하지 않은 가상의 나라 Dystopia와 비교해서 각 [Log GDP per capita, Social support, Healthy life expectancy at birth, Freedom to make life choices, Generosity, Perceptions of corruption] 값이 Life Ladder에서 어느 정도의 비중을 차지하는지를 확인한다. 여기서 주의해야 할 점은 행복지수가 6가지 요소들의 점수로 정해지지 않는다는 점이다. (원문: As already noted, our happiness rankings are not based on any index of these six factors) 그러므로 6가지 요소로 나타낸 행복지수는 원래의 행복지수 값보다 항상 작고, 그 차이를 메꾸기 위해 Dystopia라는 residual 값이 사용된다.



[그림 2] 행복의 6요소가 각각 어느 정도의 비중을 차지하는지 나타낸 도표

기존의 공식적인 Life Ladder가 나타내는 행복이 행복을 구성한다고 여겨지는 6가지 요소의 값으로부터 도출되지 않았다는 점에서 의문을 품게 되었고, 여기서 프로젝트의 주제를 정하게 되었다. Life Ladder를 먼저 구하고 이후 행복요소의 비중을 구하는 기존의 방식 대신, 행복요소의 값으로부터 Life Ladder를 구하는 것을 프로젝트의 핵심 목표로 잡았다. 여기서 사용

할 데이터가 가공되지 않은 원 데이터이기 때문에, weight vector를 구하는 데에서 새로운 의미를 도출할 수 있을 것으로 예상한다. 이렇게 전체 국가에 통용되는 Life Ladder 예측 모형을 제작한 후에는 대륙별로 데이터를 나누어 새로운 Life Ladder 예측 모형을 만들 예정이다. 가까운 곳에 있는 나라일수록 행복의 요소가 실질 행복지수에 미치는 영향이 비슷할 것으로 추정되기 때문이다.

이번 프로젝트에서 시도해 볼 과제는 다음과 같다.
1. 전체 데이터를 사용하여 Life Ladder 예측 모델 제작
2. Weight 벡터 분석
3. 대륙별 Life Ladder 예측 모델 제작

2. 전체 데이터를 사용하여 Life Ladder 예측 모델 제작

1. 데이터

```
df = pd.read_csv("http://.../Data.csv")
df.shape    # (2089, 12)

df.fillna(df.mean(), inplace=True)
```

사용한 데이터는 2089 * 12의 배열로, 처음 세 열은 각각 나라, 연도, Life Ladder를 나타내며, 나머지 9개의 열은 GDP, Social support 등의 feature 값으로 이루어져 있다. 해당 데이터에는 결측치들이 존재했는데 이는 각 열의 평균값을 가져와 대체하였다.

2. Cross-Validation

```
df = df.sample(frac=1).reset_index(drop=True)    # shuffle

df_train = df.iloc[:1253]     # 1253 train data
df_val   = df.iloc[1253:1671] # 418  validation data
df_test  = df.iloc[1671:]     # 418  test data
```

교차 검증을 위해 사용한 첫 번째 방식은 데이터셋을 랜덤한 Train, Validation, Test 셋으로 나누는 것이었다. 비율은 6 : 2 : 2로 조정하였다.

3. Parameters, Preprocessing

```
train = df_train.iloc[:, 2:]
train = (train - train.mean()) / train.std()

x_t = np.c_[np.ones((Nt, 1)), train.iloc[:, 1:].to_numpy()]
r_t = train.iloc[:, 0].to_numpy()[np.newaxis].T

valid = df_val.iloc[:, 2:]
valid = (valid - valid.mean()) / valid.std()

x_v = np.c_[np.ones((Nv, 1)), valid.iloc[:, 1:].to_numpy()]
r_v = valid.iloc[:, 0].to_numpy()[np.newaxis].T

w = np.random.randn(d + 1, 1) # weight vector
```

Training에서 learning rate는 0.02로 설정하였다. 또한 각 feature의 값들이 차이가 존재

하였기 때문에 각 feature의 평균과 편차를 이용해 정규화하였다. 학습에 사용될 weight는 표준정규분포 난수를 사용하여 초기화하였다.

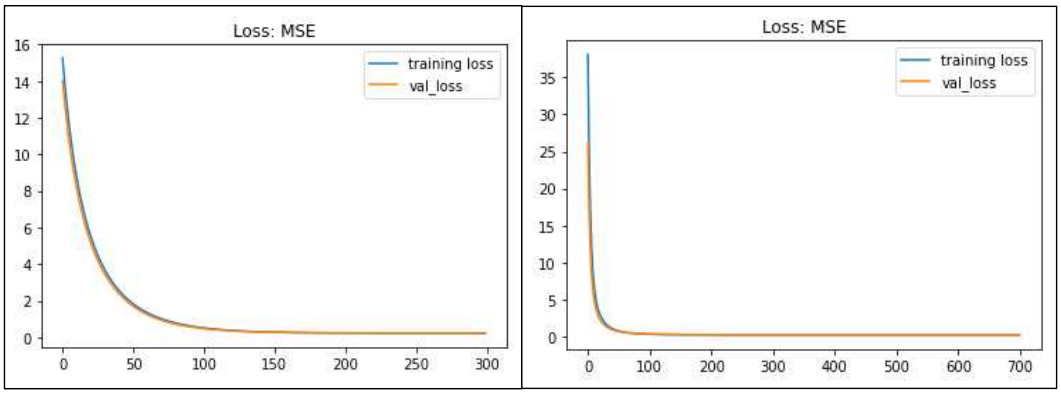
4. Training

```
for epoch in range(epo):
  E = x_t.dot(w) - r_t
  grad = (2 / Nt) * x_t.T.dot(E)
  w = w - (eta * grad)

  loss = np.sum(np.power(E, 2)) / Nt
  train_loss.append(loss)

  E = x_v.dot(w) - r_v
  loss = np.sum(np.power(E, 2)) / Nv
  val_loss.append(loss)
```

사용한 최적화 방식은 Gradient-Descent이고, Error measure는 MSE를 사용하였다.

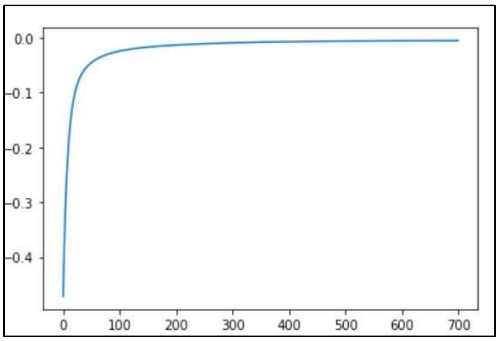
$$E = \frac{1}{N}(Y - R)^2$$
$$Y = XW$$
$$\frac{\partial E}{\partial W} = \frac{\partial E}{\partial Y}\frac{\partial Y}{\partial W} = \frac{2}{N}X^T(XW - R)$$

위 식에 따라, Error 벡터의 weight 벡터에 대한 gradient를 계산하여 learning rate를 곱해 매 epoch마다 weight를 업데이트하였다.

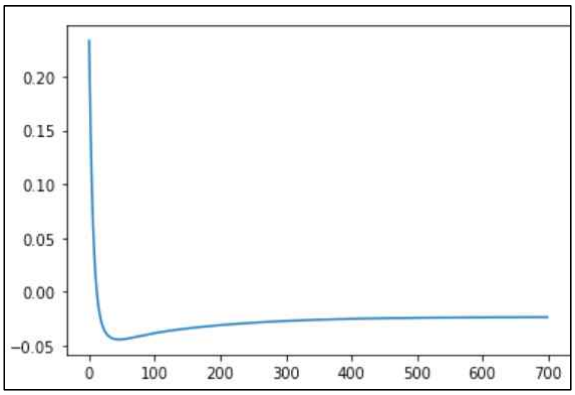


x축은 epoch, y축은 loss로 두었을 때의 graph이다. 우리의 loss는 regularization을 포함하지 않은, 단순 Squared Error 값이다. 300, 700 등의 여러 epoch으로 training을 진행해 보았으며 공통적으로 100번의 iteration 사이에 elbow point를 관측할 수 있었다.

5. k-fold cross validation



위 그래프의 x축은 epoch, y축은 training loss와 validation loss의 차(training error-validation error)인데, 예상했던 대로 validation error가 더 크고 시간이 지남에 따라 두 값 사이의 차이가 점점 줄어드는 것을 볼 수 있다.



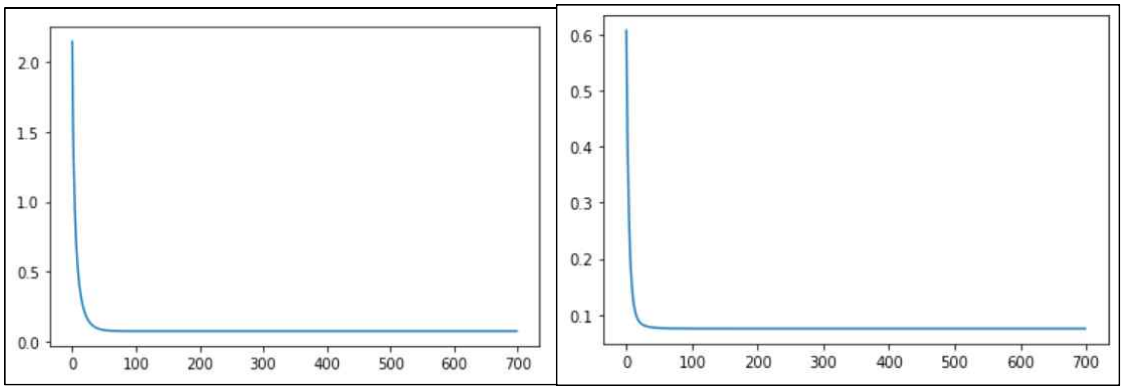
그런데 여러 번 실행해보니, 간혹 이렇게 training error가 더 크고, 두 값 사이의 차가 점차 커지는 경우가 있었다. Training set을 이용해 model을 fitting했기 때문에 validation error가 더 클 것이라고 예상했는데, training error가 더 크게 나오는 부분이 이상했기에 몇 가지 이유를 생각해보았다.

- training error는 weight vector를 update하기 전에 계산하고 validation error는 weight vector를 update한 후에 계산하기에 training error가 더 클 수 밖에 없다.
- loss function에 regularization term이 포함되어 있다.
- validation set에 더 쉬운 example들이 편중되었다.

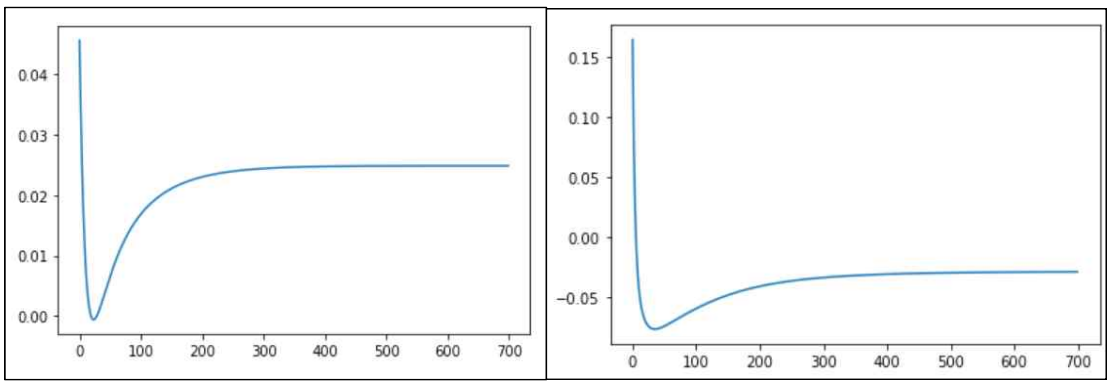
```
for epoch in range(epo):
  E = x_t.dot(w) - r_t
  grad = (2 / Nt) * x_t.T.dot(E)
  w = w - (eta * grad)

  E = x_t.dot(w) - r_t
  loss = np.sum(np.power(E, 2)) / Nt
  train_loss.append(loss)

  E = x_v.dot(w) - r_v
  loss = np.sum(np.power(E, 2)) / Nv
  val_loss.append(loss)
```

우리 모델은 1번을 막기 위해, weight vector를 update한 후 training error를 재계산하여 loss로 처리하였다. 따라서 1번은 해당하지 않으며, regularization term 또한 포함하고 있지 않기에 2번도 해당하지 않았다.

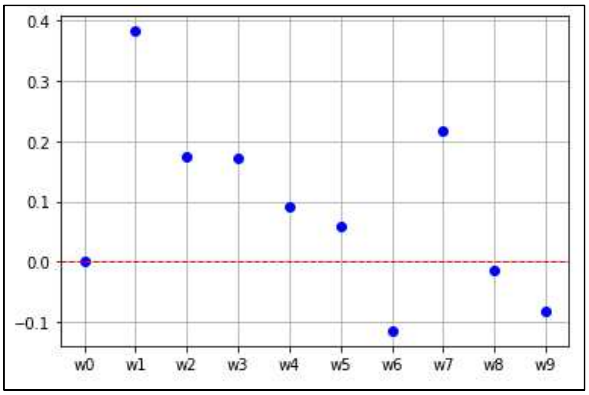
3번의 경우 해당할 가능성이 있다고 생각하였다. 모델을 적용하기 전 데이터를 한 번 섞는데, 예측과 다른 결과가 나온 경우는 validation set에 쉬운 example들이 편중된 경우라고 보았고, k-fold cross validation을 적용하면 이런 경우를 줄일 수 있을 것으로 생각하였다. 따라서 기존의 cross validation(training:validation:test = 6:2:2) 대신 k=4로 설정하고 k-fold cross validation을 수행한 결과는 다음과 같다.



오히려 더 안정적으로 training error가 크게 나오는 것을 볼 수 있었다. 아래의 기존 cross validation을 사용한 model의 [training error-validation error] 그래프와 비교했을 때, k-fold cross validation을 사용하면, training error와 validation error 사이의 갭이 0으로 수렴하지 않거나 중간에 오히려 더 커지거나 하는 경우는 없지만, 언제나 training error가 validation error보다 더 크게 나왔다. validation error가 training error보다 더 크게 나오도록 k-fold cross validation을 수행한 것인데 반대의 결과가 나온 것이다. 이 점은 아직 원인을 파악하지 못했고, 이후 다시 분석할 예정이다.



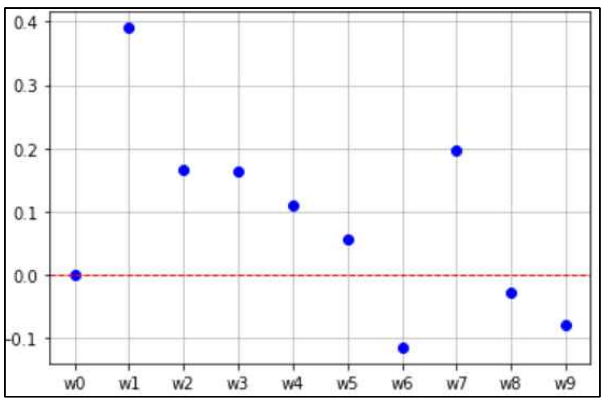
3. 결과 가중치 벡터 해석



학습을 수행한 방식이 Linear regression이기 때문에 weight 벡터를 통해 직관적으로 각 feature가 life ladder score에 영향을 미치는 정도를 분석해 볼 수 있었다. 이에 따르면 행복지수에는 Log GDP per capita가 양의 방향으로, Perceptions of corruption이 음의 방향으로 가장 큰 영향을 주는 feature라는 점을 확인할 수 있다.

GDP는 일정 기간 한 나라에서 생산된 최종생산물의 시장가치 합계로 정의되는데, GDP가 높을수록 해당 국가의 경제적인 성적 수준이 높다고 볼 수 있고 이는 개인이 느끼는 행복함에 긍정적으로 영향을 미칠 수 있다. Perceptions of corruption은 정부나 기업체가 어느 정도 부패했는지 느끼는 정도를 나타내는 지표인데, 이 수치가 높을수록 행복함에는 부정적인 영향을 미칠 것이다. 해당 feature들은 우리의 직관적 사고와 기계학습을 통해 얻은 가중치가 서로 들어맞는다.

그런데 기계학습에서는 가중치들 서로의 절대적인 크기를 비교하여 어느 특성이 좀 더 강하게 긍정적으로 작용하는지, 혹은 그 반대인지를 확인할 수 있다. 하나의 예시로 w5에 해당하는 Generosity는 w1에 해당하는 GDP보다 행복지수에 대한 그 영향이 적다는 것을 그래프를 통해 정량적으로 비교할 수 있다.



위 그래프는 k-fold cross validation을 적용했을 때의 weight vector를 나타낸 것이다. 해당 그래프가 먼저 사용한 방법으로 도출된 그래프와 큰 차이가 없는 것으로 보아, 두 결과 모두 잘 도출된 것으로 보인다.

4. 대륙별 Life Ladder 예측 모델 제작

1. 데이터 가공 및 학습

```
Europe = ['Ukraine', 'Malta', 'Sweden', 'Finland', 'United Kingdom', 'Moldova', 'Serbia', 'France', 'Albania',
          'Lithuania', 'North Macedonia', 'Germany', 'Spain', 'Belgium', 'Iceland', 'Ireland', 'Bulgaria',
          'Norway', 'Bosnia and Herzegovina', 'Romania', 'Czechia', 'Italy', 'Slovakia', 'Hungary', 'Croatia',
          'Estonia', 'Poland', 'Montenegro', 'Luxembourg', 'Argentina', 'Greece', 'Netherlands', 'Portugal',
          'Belarus', 'Austria', 'Switzerland', 'Latvia', 'Denmark', 'Kosovo', 'Slovenia', ]
North_America = ['Honduras', 'Canada', 'United States', 'Costa Rica', 'Nicaragua', 'Dominican Republic', 'Mexico',
                 'Guatemala', 'Haiti', 'Panama', 'El Salvador', 'Jamaica', 'Belize', 'Cuba', ]
South_America = ['Chile', 'Bolivia', 'Uruguay', 'Peru', 'Venezuela', 'Suriname', 'Paraguay', 'Colombia', 'Ecuador',
                 'Brazil', 'Trinidad and Tobago', 'Guyana', ]
Africa = ['South Africa', 'Tanzania', 'Chad', 'Afghanistan', 'Zambia', 'Guinea', 'Liberia', 'Togo', 'Mozambique',
          'Kenya', 'Mauritania', 'Ivory Coast', 'Sierra Leone', 'Somalia', 'Ghana', 'Morocco', 'Cameroon',
          'Madagascar', 'Tunisia', 'Mauritius', 'Comoros', 'Botswana', 'Nigeria', 'Congo (Brazzaville)', 'Namibia',
          'Libya', 'Uganda', 'Burkina Faso', 'Ethiopia', 'Malawi', 'Lesotho', 'Senegal', 'Mali', 'Sudan', 'Zimbabwe',
          'Rwanda', 'Benin', 'Djibouti', 'Gabon', 'Algeria', 'Burundi', 'Niger', 'South Sudan', 'Congo (Kinshasa)',
          'Gambia', 'Somaliland region', 'Angola', 'Central African Republic', 'Eswatini', ]
Asia = ['Bangladesh', 'Saudi Arabia', 'Singapore', 'Japan', 'Turkey', 'Israel', 'Philippines', 'Bahrain',
        'Pakistan', 'Indonesia', 'Egypt', 'Lebanon', 'Mongolia', 'Kyrgyzstan', 'Myanmar', 'Sri Lanka', 'Georgia',
        'United Arab Emirates', 'Azerbaijan', 'Cambodia', 'Laos', 'Malaysia', 'Yemen', 'Vietnam', 'Armenia',
        'Palestinian Territories', 'South Korea', 'Russia', 'Jordan', 'China', 'Syria', 'Cyprus', 'India',
        'Turkmenistan', 'Kazakhstan', 'Uzbekistan', 'Tajikistan', 'Thailand', 'Taiwan Province of China', 'Bhutan',
        'Nepal', 'Iran', 'Qatar', 'Hong Kong S.A.R. of China', 'Kuwait', 'Iraq', 'North Cyprus', 'Oman', 'Maldives']
Oceania = ['New Zealand', 'Australia', ]
```

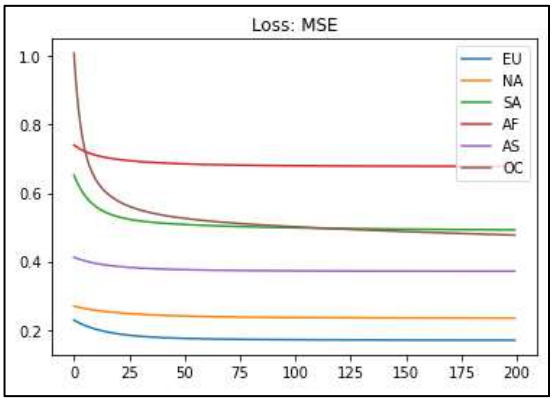
데이터에 있는 나라들을 대륙별로 분류하고 각각 따로 데이터 프레임을 생성하였다. 학습 과정은 전체 데이터를 사용할 때처럼 똑같은 선형회귀법과 MSE loss를 사용한다. 학습 에폭은 각 대륙 모두 200으로 설정하였으며, learning rate 값은 이전과 같이 고정된 0.02이다.

```
for i, data_train in enumerate(data):
  data_train = data_train.iloc[:, 2:]
  data_train = (data_train - data_train.mean()) / data_train.std() # 정규화
  N = len(data_train)

  x = np.c_[np.ones((N, 1)), data_train.iloc[:, 1:].to_numpy()]
  r = data_train.iloc[:, 0].to_numpy()[np.newaxis].T
  weight = w # transfer, from previous training

  for epoch in range(epo):
    E = x.dot(weight) - r
    grad = (2 / N) * x.T.dot(E)
    weight = weight - (eta * grad)

    E = x.dot(weight) - r
    l = np.sum(np.power(E, 2)) / N
```

가중치 벡터를 만들 때, 전체 데이터로 충분한 양의 학습을 거친 가중치가 존재하였기 때문에, 전이학습의 아이디어로 이를 대륙별 학습 과정에서 초기 가중치로 사용하였다.

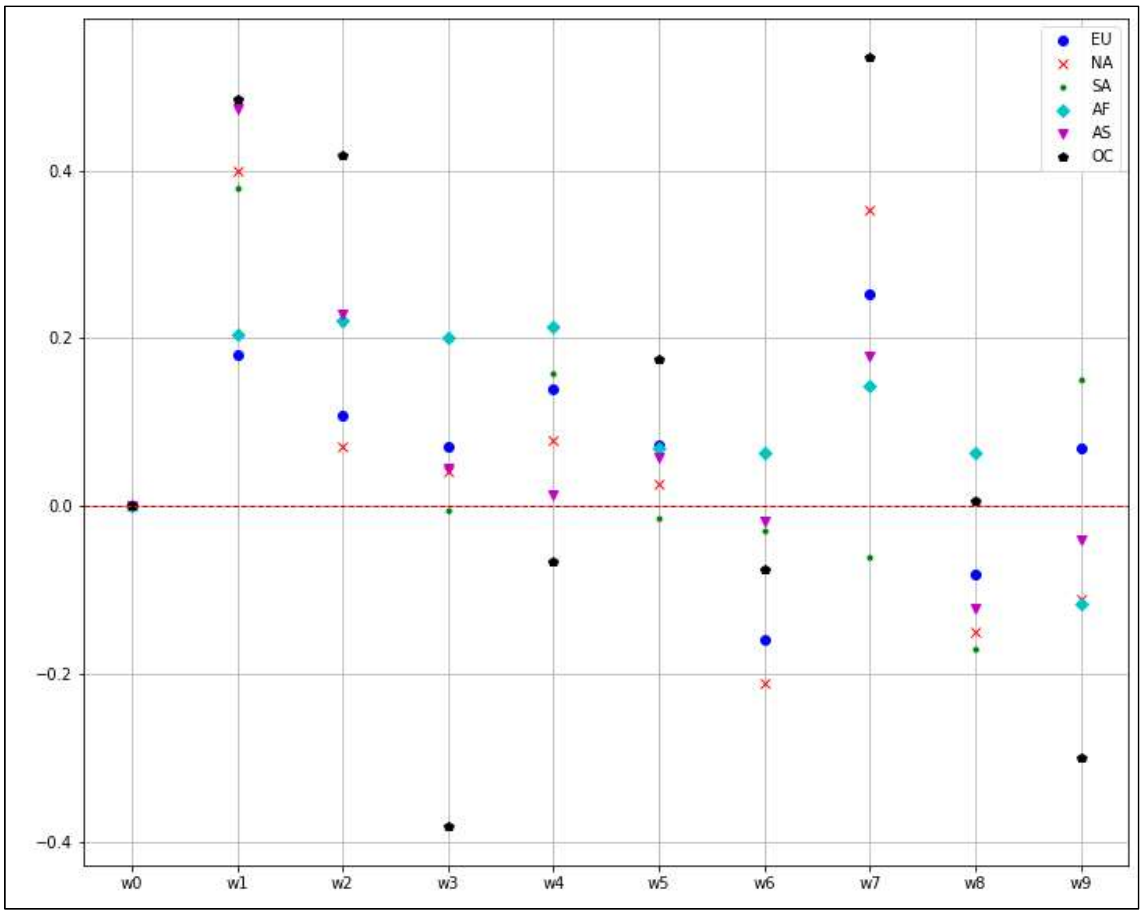
2. 학습 결과

각 데이터의 수가 이전보다 적기 때문에 Cross-validation 방법은 사용하지 않고 오직 Training만 진행하였다. 진행한 후의 최종 loss값은 아래와 같다.




```
Train loss(MSE):
  Europe:          0.1695
  North America:   0.2343
  South America:   0.4924
  Africa:          0.6783
  Asia:            0.3714
  Oceania:         0.477
```

그리고 최종 대륙별 가중치 값은 아래와 같다.



3. 가중치 해석

전체 데이터를 이용해 제작한 Life Ladder 모델에서는 Perceptions of corruption, Negative affect, Confidence in national government의 가중치들이 음의 방향으로 결과에 영향을 주었고 나머지 feature들은 양의 방향으로 영향을 끼친 것을 확인할 수 있었다. 하지만 대륙별로 제작한 모델의 경우 대륙에 따라 Healthy life expectancy at birth, Freedom to make life choices, Generosity와 Positive affect까지 음의 방향으로 결과의 영향을 줄 수 있음을 알 수 있다. 주목할 만한 점은 오직 남아메리카와 오세아니아에서만 이런 반대 방향의 가중치가 나왔다는 것이다. 이런 식으로 직관과 반대되는 결과가 나오는 이유를 알기 위해서는 해당 대륙들의 사회•문화적 배경을 추가적으로 조사해야 할 것이다.

w0는 bias를 의미하는데, 전체 데이터를 사용한 회귀와 같이 0에 가까운 값을 보여주는데, 이는 데이터 정규화 과정을 거쳤기 때문으로 보인다. w1, GDP에 대한 가중치는 오세아니아, 아시아가 가장 크게 나왔고, 유럽, 아프리카가 가장 낮게 나왔다. 상대적으로 아시아와 오세아니아에서는 국가의 경제적 능력 또는 개개인의 부가 행복함에 크게 영향을 미치고 있으며, 이와 반대로 유럽과 아프리카에서는 상대적으로 앞 두 대륙에 비교해 덜 중요하게 생각하고 있다는 점을 알 수 있다. w2는 자신의 문제 상황을 도울 사회적 관계를 의미하는 지표인데, 오세아니아에서 매우 높게 나타나지만, 북아메리카와 유럽에선 0에 가깝게 나타난 것으로 보인

다. w3~w5인 예상 건강수명, 선택의 자유도, 사회에 대한 기부는 전반적으로 다른 features에 비교해 그 가중치가 크지 않았다. 그런데 오세아니아는 예상 건강수명에 대해 아주 크고 부정적인 음의 가중치를 갖고 있는데, 이는 YOLO의 성향을 보여주는 것인지 상식과는 조금 다른 모습을 보여준다. w6는 정부나 비즈니스에 얼마나 부정부패가 만연한지 인식하는 지표에 대한 가중치다. 대부분의 대륙들은 이에 들어맞게 음의 가중치를 갖고 있지만, 아프리카는 오히려 양의 가중치를 보여주고 있다. 그 크기는 상대적으로 작지만, 만연한 부정부패에 대해 개개인의 행복이 크게 상관하지 않는다는 성향을 알 수 있었다. w7과 w8은 각각 개인의 긍정적, 부정적 경험의 지표들이다. 그러나 남아메리카는 이상하게도 긍정적 경험에 음의 가중치를 두고 있고, 아프리카는 부정적 경험에 양의 가중치를 두고 있다. 둘 다 그 크기가 크지 않은 것으로 보아, w6와 마찬가지로 타 대륙들에 비해 해당 features를 크게 신경쓰지 않는 듯하다. 마지막으로 w9은 정부에 대한 믿음에 해당하는 가중치이다. 남아메리카와 유럽은 그 믿음의 정도가 행복함에 긍정적으로 영향을 미치는 반면, 다른 대륙들은 부정적으로 영향을 미치고 있는 것으로 보인다.

5. 본 결과를 바탕으로 향후 가능한 추가 연구 개발 방향

대륙별 데이터가 매우 적은 탓에, 대륙별 life ladder estimation model의 경우 training loss가 전체 데이터를 이용한 life ladder estimation model에 비해 큰 값으로 수렴한다. 더 많은 데이터가 필요할 것으로 보인다.

또한, 아프리카 대륙의 경우 데이터가 적지 않았음에도 다른 대륙의 모델에 비해 training loss가 크게 나타나는데, 한 대륙이지만 나라별 정세 차이가 특히 심한 탓에 통합된 모델을 제작할 수 없는 것으로 보인다. 따라서 종교 등 상황이 유사한 국가별로 재차 묶어 모델을 제작하면 더 의미 있는 분석이 가능할 것으로 생각된다.

해당 연구를 통해 제작한 두 종류의 모델은 각 feature가 인간 삶에 미치는 정도를 학습하였다. 이에 transfer learning을 적용해 prediction layer를 바꿔 자살률/출산율/이민율 예측 등 다른 유사한 task에 적용할 수 있을 것이라 기대된다.

또한, linear regression을 사용하였으므로 각 feature가 행복지수에 미치는 영향에 대해 직관적으로 분석할 수 있다. 어떤 feature가 어느 정도가량 증가/감소 했을 때 미래 행복지수가 얼마나 변화할지 예측 가능하므로 이를 이용해 성장 혹은 개선 방향을 제안 할 수 있을 것이다.