Brain Age Prediction Using Machine Learning

한지훈
경희대학교 소프트웨어융합학과
jhhan0208@khu.ac.kr

요 약

한국정보과학회는 정보과학에 관한 기술을 발전, 보급시키고 회원상호간의 친목을 도모하기 위하여 1973년 3월 3일에 설립되었으며, 정보통신부에 '사단법인 한국정보과학회'로 등록되었다. 학회의 주요 활동은 1) 컴퓨터 기술 및 이론에 관한 새로운 연구결과를 발표하는 기회를 제공하고, 2) 국내의 컴퓨터 관련 기술 개발에 참여 하며, 3) 국제적 학술 교류 및 협력 증진을 도모 하고, 4) 회원 상호간의 친목을 증진시키는 것이다.

1. 서 론

프로젝트 소개

해당 프로젝트의 궁극적인 목표는 최대한 정확하게 뇌 나이(brain age)를 예측하는 것이다. 뇌 나이를 계산하면 실제 나이(chronological age)와 비교하여 뇌가 건강한지를 진단할 수 있다. 또한 뇌 관련 질병의 발병 시기를 예측하는 데에도 쓰일 수 있다. 따라서 이 brain age를 최대한 정확하게 예측하는 모델을 train하고, 그 모델을 바탕으로 임의의 사람의 brain age를 예측하는 것이 이 프로젝트이다.

프로젝트 개요

프로젝트의 train data로 여러 사람들의 68개 regional cortical thickness 와 intracranial volume이 주어진다. 이 데이터들은 brain age와 관련이 있다. 이 train data를 preprocess하고, data의 특성에 맞는 machine learning model을 선택할 것이다. 선택한 Model에 data를 fit 하고, 그 data를 바탕으로 model의 parameter을 tuning하는 과정으로 최적의 model을 찾을 것이다. 완성된 model로 주어진 test data의 뇌 나이를 예측하고, 결과를 xlsx 파일로 export하면 프로젝트가 마무리될 것이다.

2. 방 법

2.1. Train Data Preprocessing

2.1.1. Nan value 제거

사용할 data에 Nan value가 있으면 사용하기 전에 손을 봐야한다. Nan value에 대해서 다른 값들의 평균값 등을 대입하는 방법도 있지만, 해당 사람의 정확한 data 가 아니므로 값을 대입하는 대신 해당 row를 drop하는 방식을 택하였다. Pandas의 dropna() 함수를 사용하여 Nan 값이 있는 열들을 제거해주었다.

2.1.2. Target 분리

Data 중 target 값을 다른 값을 바탕으로 예측하기 때문에 train data를 X, y(target)로 분리하는 과정이 필요하다. Numpy array의 indexing을 통해 data를 X, y로 분리하였다.

2.1.3. Scale data

Data의 크기에 따른 비중의 차이를 최소화하기 위해 data의 scaling을 진행했다. 특히 data중 ICV 값은 다른 값에 비해 매우 커서 먼저 네 제곱근을 취한 후에 scaling하였다. Scaling은 sklearn의 여러 scaler 중에 StandardScaler을 이용하였다.

2.2. Applying machine learning model

2.2.1. 데이터 특성 분석

최적의 모델을 찾기 위해 우선 train data의 특성을 분석해보았다. 우선 data의 feature들은 모두 categorical하지 않고 continuous하기 때문에 별도의 encoding 과정 없이 분석에 사용할 수 있음을 알았다. 가장 눈에 띄는 점은 train data의 target 또한 continuous하다는 것이었다. 이전의 프로젝트에서 target는 yes/no, 또는 특정 class의 형태로 주어졌는데, target이 continuous한 나이라는 점에 주목했다.

2.2.2. 데이터에 알맞은 모델 선정

우선 data의 target이 continuous하기 때문에 관련 모델들에 대해 알아보았다. 기본적으로 continuous target의 예측에는 regression model이 많이 쓰인다. 여러 regression model을 둘러보았고, 그 중 Gaussian Process Regression 모델을 사용하기로 결정했다.

Gaussian Process Regression 모델은 multivariate Gaussian 분포를 기반으로 한다. multivariate Gaussian 분포는 mean vector μ, covariance matrix Σ에 의해 다음과 같이 정의된다.[1]

$$\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

그림 1 multivariate Gaussian 분포

즉 mean과 variance로 정의되는 Gaussian 분포를 다차원 공간에 대해 확장한 분포이다.

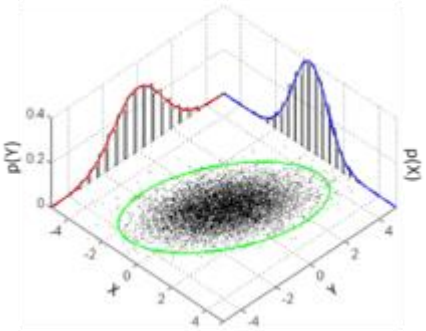


그림 2 multivariate Gaussian 분포 시각화

Gaussian Process(GP)도 이와 유사하게 mean function m(x), covariance function k(x,x′)에 의해 다음과 같이 정의된다.

$$f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$$

그림 3 Gaussian Process 분포

여기서 mean function은 임의의 데이터 x의 mean을 나타낸다. 또한 covariance function은 임의의 두 데이터 x1, x2의 상관관계를 나타낸다.

2.2.3. 모델을 scikit-learn으로 적용

Gaussian Process를 실제로 적용할 때 주목해야 할 것은 covariance function k(x,x′)이다. kernel라고도 부르는데, 이 kernel의 설정이 model initiation에 중요하기 때문이다. kernel에는 여러가지 종류가 있고, 두개 이상의 kernel을 합쳐서 사용할 수도 있다. 나는 Gaussian 형태를 취하는 RBF(Radial Basis Function) kernel에 noise를 추가한 형태의 kernel을 사용하기로 하였다. [2]

```
kernel = ConstantKernel() + \
ConstantKernel() * RBF() + WhiteKernel()
```

각각의 kernel에 대해서 hyperparameter는 먼저 정하지 않고 data에 맞는 최적값을 찾기 위해 초깃값을 설정하지 않았다. 또한 모델 선언시에 최적의 parameter을 찾기 위해서 전범위의 다른 값에 대해 30번 모델을 돌려서 최적값을 찾았다.

```
n_restarts_optimizer = 30)
```

2.2.4. Visualization

Matplotlib을 이용하여 모델의 예측 결과를 간단하게 시각화 해보았다. x축에는 실제 값(actual age), y축에는 예측 값(predicted age)을 mapping하여 시각화 하였다. 총 두개의 그래프를 만들었다.

- older/younger brain의 군 표시 그래프
- 각 data의 brain-PAD값 표시 그래프

2.2.5. 성능 평가

Test set들의 target value는 비공개 되어있기 때문에 직접 test set에 대한 Evaluation을 하는 것은 불가능하다. 따라서 성능 평가를 위해 target value가 존재하는 Training set를 다시

80%의 training set와 20%의 임시 test set로 나누어서 임시 test set에 대한 정확도를 계산하였다. 정확도는 sklearn의 r2_score 함수를 이용하여 임시 test set의 실제 나이 값과 예측 나이 값이 얼마나 차이나는 가를 기준으로 계산하였다.

3. 결 과

3.1. r2_score

임시 test set의 실제 값과 예측 값 간의 차이를 계산한 결과, r2_score은 0.72가 나왔다.

```
R2_score = 0.72
```

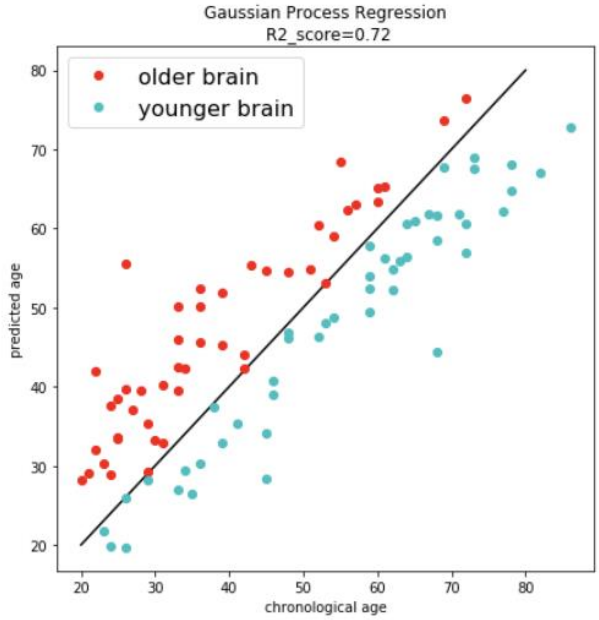
최상의 정확도를 보이지는 않지만, fitting된 모델이 여러 data에 대해 어느 정도 적합하고, 유의미한 분류를 해낼 수 있음을 나타내는 정확도라고 볼 수 있다.

3.2. Visualization

3.2.1. Visualize older/younger brain groups

older/younger brain을 나눈 그래프는 다음과 같이 나왔다.

표 1 older/younger brain distribution graph

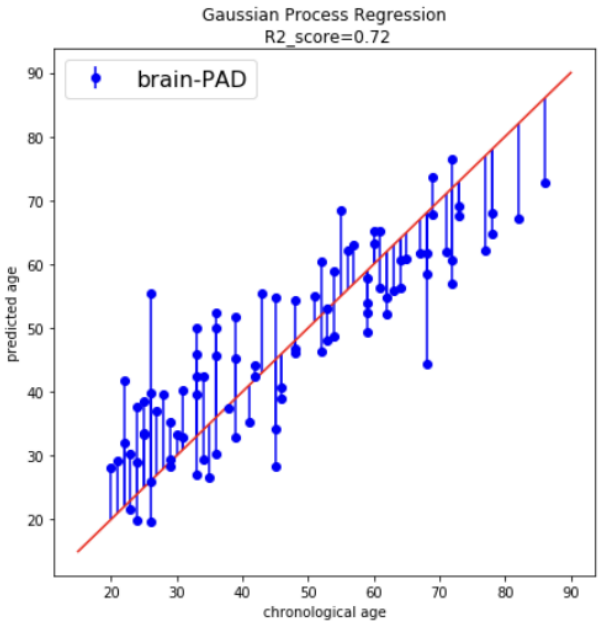


실제 나이보다 측정 나이가 높으면 older brain, 낮으면 younger brain으로 구분한 것이다. 두 그룹을 다른 색으로 plot하니 각 그룹의 분포를 더 직관적으로 알 수 있었다. 두 그룹에 속한 data의 수는 거의 같았다. 또한 특정 나이대에 data가 편중 되어있기보다는 전반적으로 고른 분포를 보였다.

3.2.2. Visualize 'brain-PAD' value

각 datapoint의 brain-PAD 값을 나타낸 그래프는 다음과 같이 나왔다.

표 2 'brain-PAD' value distribution graph



각 datapoint가 어떤 group에 속하는지는 신경 쓰지 않고 실제 값과 예측 값의 차이에만 주목하였다. 데이터의 분포를 보면 전반적으로 예측 값이 실제 값과 크게 차이 나지 않는다는 것을 알 수 있다(전반적으로 y = x 그래프를 따라 분포하고 있다.).

몇 개의 data point는 y = x 그래프로부터 꽤나 멀게 분포함을 알 수 있다. 해당 포인트들에 대해서는 정확도가 높지는 않았던 것이다. 그러나 오히려 모든 data에 대해 매우 정확한 예측을 하는 것은 test set에 대한 overfitting으로 이어질 수 있기 때문에 해당 측면에서 봤을 때는 적당하게 fitting된 model이라고 판단하였다.

3.3. Predict brain age of IXI test data, COBRE test data

Train data에 의해 fitting된 모델을 통해서 주어진 두개의 test set(IXI test data set, COBRE test data set)의 brain age를 예측하였다. 또한 예측 값을 excel 파일로 export하여 저장하였다.

앞서 언급한 것처럼 이 두 개 set의 실제 age는 비공개 되어있기 때문에 직접 evaluation하는 것이 불가능했다. 따라서 예측 결과 excel 파일을 제출하고 결과를 기다렸다. 평가 결과는 다음과 같다.

IXI		COBRE		RANK
MAE	R	MAE	R	
8.277385	0.763008	13.15278	0.535718	14

4. 소 감

먼저, 한학기 동안 수업에서 배운 machine learning 내용들을 직접 프로젝트에 적용해볼 수 있어서 좋았다. 특히 프로젝트를 진행하면서 공부했던 것들에 대해 더 세세하게 알아본 것 같다. 예를 들어서 내가 개발한 모델의 정확도를 높이기 위해서 preprocessing의 방법을 더 찾아보거나, 모델의 적합한 parameter을 찾기 위한 방법들을 알아보는 과정들이 있었다. 배운 것들을 적용해보니 수업 내용이 더 체계적으로 정리되는 느낌이었다.

또한 data science 프로젝트의 한 cycle을 경험해보았다는 것이 의미 있는 경험으로 남을 것 같다. 소프트웨어융합학과의 데이터 사이언스 트랙을 선택한 이후에 data scientist는 어떤 과정들을 거쳐서 하나의 결과를 만들어내는 지에 대해 많이 찾아보았다. 데이터 전 처리 과정부터 결과 시각화까지의 이론적으로만 알고 있었던 과정을 직접 해본 이 프로젝트를 통해 어떤 방식으로 일을 해 나가야 하는지에 대한 감이 조금은 잡히게 된 것 같다. 이번 학기가 끝나면 2개월 후에 군대를 가는데, 군대 가기 전, 그리고 가서도 여러 프로젝트들을 해볼 계획이다. Kaggle 등에서 진행하는 대회, 혹은 내가 직접 data를 가져와서 분석하는 방향으로 진행해보고 싶다. 이 PAC Challenge가 앞으로 진행하는 프로젝트들의 기반이 되겠다고 생각하였다.

참고문헌

[1] "Multivariate Normal Distribution", BRILLIANT, accessed June, 15, 2021, URL

[2] "Quick Start to Gaussian Process Regression", Towards data science, accessed June, 15, 2021, URL