The Korean Journal of Applied Statistics
0000, Vol. 00, No. 0, 1–12

DOI: http://dx.doi.org/10.5351/KJAS.0000.00.0.001

Comparison Study of Machine Learning Model for Predicting Continuous Data

A Yeon Yun[a]

[a]Department of Statistics, Korea University

Abstract

A machine learning model is a model in which an algorithm makes predictions about data by learning given data. In this paper, we use five machine learning models, random forest, boosting, adaptive LASSO, SCAD-penalized regression, and elastic net, to find the model that predicts disease progression scores in patients with Lou Gehrig's disease, which is continuous data. After parameter tuning for five models, the prediction performance of each model is evaluated and compared. For the model with the best prediction performance, we proceed with model interpretation by finding variables that play an important role in the prediction process.

Keywords: random forest, boosting, adaptive LASSO, SCAD-penalized regression, elastic net, parameter tuning

1. 서론

머신러닝(machine learning) 모델이란 알고리즘이 주어진 데이터를 학습함으로써 새로운 데이터에 대한 예측을 수행하도록 하는 모델을 뜻한다. 전체 데이터 중 알고리즘의 학습을 위해 주어지는 데이터를 train data라고 하며 학습된 알고리즘의 예측 성능을 확인하기 위해 사용되는 데이터를 test data라고 한다. 머신러닝 모델 적합 시 모델의 구조 및 학습과정에 영향을 미치는 값을 모수(parameter)라고 하는데, 머신러닝 모델의 예측 성능을 향상시키기 위해서는 모수의 값을 조정하는 튜닝(tuning)과정을 거쳐야 한다. 하나의 모수에 대해 여러 가지 값을 사용해보고 가장 좋은 예측 성능을 보여주는 값을 선택해 최종 머신러닝 모델에 사용함으로써 모델의 예측 성능 향상을 기대할 수 있다.

본 논문에서는 다음의 5가지 머신러닝 모델을 사용해 연속형 데이터를 예측하고 각 모델의 예측 성능을 비교하고자 한다: 1) Random forest; 2) Boosting; 3) Adaptive LASSO; 4) SCAD-penalized regression; 5) Elastic net. 모델 적합 및 예측에 사용된 데이터는 루게릭병 환자에 대한 ALS 데이터이다. 해당 데이터는 1822개의 관측치를 포함하며 환자의 루게릭병 진행 점수를 나타내는 반응변수(dFRS)와 369개의 예측변수로 구성된다($n = 1822, p = 369$). 모델 적합 및 예측 성능 평가를 위해 ALS를 1197개의 관측치를 포함한 train data와 625개의 관측치를 포함한 test data로 나누었다.

논문의 구성은 다음과 같다. 2장에서는 elastic net에 대해 간단히 설명한다. 3장에서는 각 모델의 모수를 튜닝하는 과정과 튜닝 후 최종적으로 선택된 모수를 제시한다. 그 후 4장에서는 튜닝 과정을 거쳐 선택된 5가지 모델의 예측 성능에 대한 평가 및 비교를 시행한다. 비교 결과, 가장 예측 성능이 뛰어난 모델에 대해서는 예측 과정에서 중요하게 사용된 변수를 살펴본다. 5장에서는 전체적 내용을 요약한다.

2. Elastic net

반응변수 $\mathbf{y} = (y_1, y_2, \cdots, y_n)'$와 p개의 예측변수를 가지는 설계행렬(design matrix) $X \in \mathbb{R}^{n \times p}$에 대해서 다음과 같은 선형 모델이 존재한다고 가정하자.

$$\mathbf{y} = X\beta + \epsilon \tag{2.1}$$

이때 ϵ은 평균과 분산으로 각각 0과 σ^2를 가지는 분포를 따르는 n차원 노이즈(noise) 벡터이다. 미지의 p차원 모수 벡터 $\beta = (\beta_1, \cdots, \beta_p)'$는 각 예측변수가 반응변수에 미치는 영향에 대한 정보를 담고있으며 β_i는 i번째 예측변수의 계수라고 부른다. 이러한 β를 추정하기 위해 linear regression, LASSO regression, ridge regression 등의 모델이 제안되었다.

Zou와 Hastie (2005)가 제안한 elastic net 역시 선형 모델에서 β를 추정하기 위한 모델이다[7]. Elastic net은 기존의 LASSO regression과 ridge regression을 혼합하면서 두 모델의 단점을 개선했다. Ridge regression은 변수 선택 기능이 없으므로 모든 예측변수가 모델에 사용된다는 단점이 있다. LASSO regression은 변수 선택은 가능하지만, 예측변수가 관측치보다 많은 고차원 데이터 (high-dimensional data)를 분석할 경우 관측치보다 많은 변수를 선택할 수 없다는 제약이 있다. 데이터에 상관관계가 높은 변수 그룹이 존재하는 경우, 해당 변수들의 상관관계를 무시하고 변수 선택을 진행한다는 것도 LASSO regression의 단점이다. 이러한 단점을 보완한 elastic net은 고차원 데이터에서도 자유롭게 변수 선택이 가능하다. 또한, 상관관계가 높은 변수들의 경우 변수들의 상관관계를 고려해 변수 계수 추정치의 shrinkage가 함께 일어난다. Elastic net에서는 다음과 같은 elastic net penalty를 적용해 LASSO regression과 ridge regression을 혼합하는 효과를 낸다.

$$p(x;\alpha) = \frac{1}{2}(1-\alpha)\,\|x\|_2^2 + \alpha\|x\|_1 \tag{2.2}$$

식 (2.2)의 elastic net penalty를 적용한 elastic net 모델을 통해서 다음과 같은 elastic net 계수 추청치 $\hat{\beta}(\alpha,\lambda)^{elastic}$를 얻을 수 있다.

$$\hat{\beta}(\alpha,\lambda)^{elastic} = \arg\min_{\beta} \|\mathbf{y} - X\beta\|_2^2 + \lambda p(\beta;\alpha) \tag{2.3}$$

Elastic net 모델의 경우 elastic net penalty 함수의 모수인 α와 elastic net 모델의 벌점모수인 λ를 튜닝함으로써 예측 성능을 향상시킬 수 있다.

3. 모수 튜닝

3절에서는 각 모델의 모수를 튜닝하는 과정을 설명한다. 모수 튜닝은 모델에 따라 교차검증(cross validation)과 아웃-오브-백(out-of-bag; OOB) 방법을 사용하며 가장 작은 평균제곱오차(mean squared error; MSE)를 보여주는 모수를 최종 모수로 선택한다.

3.1. Random forest

Random forest 모델은 분산이 큰 decision tree의 단점을 보완하기 위해 여러 개의 decision tree를 생성한 후 평균을 내는 모형이다. 동질적인 decision tree를 평균 내는 것은 분산 감소 효과가 거의 없기 때문에 random forest 모델에서는 서로 독립적인 decision tree를 생성한다. 하나의 decision tree를 적합할 때마다 train data에서 추출한 bootstrap 데이터와 전체 예측변수 중 임의로 선택된 변수를 사용한다.

Random forest 모델 적합 시 필요한 모수는 적합하는 decision tree의 수(B)와 각 decision tree에 사용되는 예측변수의 수(m)이 있다. 두 모수 중 B는 충분히 큰 수이기만 하면 되지만 m은 모델 적합에 영향을 미치는 모수이므로 튜닝이 필요하다. 본 논문에서는 모수 B를 충분히 큰 수인 1000으로 선택하고 모수 m을 튜닝하도록 한다.

모수 m을 튜닝하기 위해 10-folds 교차검증을 실시한다. 교차검증을 위해 R의 `randomForest` 패키지[4]에서 제공하는 `rfcv`함수를 사용한다. 모수 B는 1000으로 고정한 상태에서 모수 m이 $1, 2, 4, 7, 15, 30, 61, 123$일 때의 MSE를 계산한 후 가장 작은 MSE를 보여주는 m값을 최종 m값으로 결정한다.

표 3.1을 통해 $m = 30$일 때 가장 작은 MSE가 나타난다는 것을 알 수 있다. 따라서 random forest의 모수는 최종적으로 $(B, m) = (1000, 30)$로 결정한다.

Table 3.1: Random forest 모수 튜닝 : 10-folds 교차검증을 통해 계산된 모수 m에 따른 MSE
표에 제시된 MSE는 실제 MSE에 1000을 곱한 값이며 최소 MSE와 그에 해당하는 m값은 진하게 표시했다.

m	1	2	4	7	15	**30**	61	123
MSE	330.640	280.883	264.110	257.897	252.625	**250.612**	250.737	253.066

3.2. Boosting

부스팅 모형은 편향(bias)이 큰 decision tree의 단점을 보완하기 위해 여러 개의 decision tree를 합하는 모형이다. 서로 독립적인 decision tree를 생성하는 random forest와는 달리 boosting에서는 하나의 decision tree가 다음 차례에 생성되는 decision tree에 영향을 미친다. Boosting에서는 정해진 깊이(depth)에서 가장 좋은 적합을 보여주는 decision tree를 생성한 후, 해당 decision tree가 가진 설명력의 일부만 반영해 전체 모델에 더해준다. 그 후 해당 decision tree의 설명력을 제거한 데이터를 사용해 다음 차례의 decision tree를 적합한다.

Boosting 모델 적합 시 필요한 모수는 생성하는 decision tree의 수(B), decision tree 생성 시 최대 깊이(d), decision tree가 가진 설명력이 boosting 모델에 반영되는 정도(ϵ)가 있다. Random forest와는 달리 boosting에서는 B가 커질수록 과적합(over-fitting)이 발생하기 때문에 모수 d, ϵ와 함께 모수 B도 튜닝이 필요하다.

모수 B, ϵ, d를 튜닝하기 위해 OOB 오차를 계산한다. OBB 오차의 계산을 위해 R의 `gbm` 패키지[3]에서 제공하는 `gbm`함수를 사용한다. `gbm`함수는 `train.fraction` 속성에 설정된 비율에 따라 주어진 train data에서 decision tree 적합에 사용될 데이터를 추출하고 나머지 데이터는 OOB 오차를 계산하는데 사용한다. 본 논문의 튜닝 과정에서는 `train.fraction`를 0.5로 설정한다. 튜닝을 위해 모수 ϵ는 구간 $[0.005, 0.1]$에서 동일한 간격으로 생성된 10개의 값을, 모수 d는 $2, 4, 7$의 세 가지 값을 사용한다. 모수 (ϵ, d)의 모든 조합에 대해 1500개의 decision tree를 생성해 하나의 decision tree가 생성될 때 마다 OBB 오차를 계산한다. 그 후 가장 작은 오차를 보여준 (ϵ, d) 조합과 그 조합에 해당하는 decision tree의 개수를 최종 B, ϵ, d값으로 결정한다.

Table 3.2: Boosting 모수 튜닝
: 10-folds 교차검증 결과 각 모수 (ϵ, d)에서의 최소 MSE와 최소 MSE가 나타난 B값
표에 제시된 MSE는 실제 MSE에 1000을 곱한 값이며 최소 MSE와 그에 해당하는 B, ϵ, d값은 진하게 표시했다.

ϵ \ d	**2**		4		7	
	MSE	B	MSE	B	MSE	B
0.005	271.051	1487	276.092	925	276.035	754
0.016	269.952	660	275.789	353	275.609	222
0.026	269.564	431	275.64	172	274.707	151
0.037	269.27	279	274.717	118	276.543	93
0.047	269.112	219	277.381	117	276.59	119
0.058	268.479	219	275.265	71	274.947	67
0.068	269.342	158	276.501	71	276.263	47
0.079	**268.452**	**150**	274.098	64	275.629	43
0.089	271.451	109	272.93	88	276.445	37
0.100	270.111	113	276.336	56	274.616	38

표 3.2에 제시된 MSE값을 살펴보면 $(\epsilon, d) = (0.079, 2)$일 때 가장 작은 것을 알 수 있으며 이때 B값은 150으로 나타난다. 또한 모든 B값이 1500보다 작으므로 boosting 모델이 모수 (ϵ, d)의 모든 조합에서 최소 MSE를 찾은 후 과적합 단계로 넘어갔음을 알 수 있다. 교차검증 결과에 따라서 boosting 모델의 모수 (B, ϵ, d)를 $(150, 0.079, 2)$로 결정한다.

3.3. Adaptive LASSO

Adaptive LASSO 모델은 모든 변수에 동일한 벌점모수(penalty parameter)가 적용되는 LASSO regression을 개선한 모형이다. Adaptive LASSO는 변수마다 다른 가중치 $w = (w_1, \cdots, w_p)'$를 벌점모수 λ에 곱하는 방법을 통해 변수마다 계수 추정치의 shrinkage가 다르게 일어나도록 한다.

$$\hat{\beta}(\lambda, w)^{adaptive} = \arg\min_{\beta} \|\mathbf{y} - X\beta\|_2^2 + \lambda\|w'\beta\|_1 \tag{3.1}$$

LASSO regression 모델에 가중치 w를 적용한 adaptive LASSO를 통해 식 (3.1)와 같은 adaptive LASSO 계수 추정치 $\hat{\beta}(\lambda, w)^{adaptive}$를 구할 수 있다.

통상적으로 사용되는 가중치는 linear regression 적합 시 계산되는 계수의 최소제곱법(ordinary least squares; OLS) 추정치를 사용한다. OLS 추정치 $\hat{\beta}^{OLS}$의 절댓값에 역수를 취한 값을 가중치로 두어 반응변수에 미치는 영향과 계수 추정치의 shrinkage 정도가 반비례하도록 하는 방법이다. 해당 가중치를 식으로 나타내면 다음과 같다.

$$w_i = \frac{1}{|\hat{\beta}_i|^{OLS}}, \quad for\ i = 1, 2, ..., p. \tag{3.2}$$

본 논문에서도 식 (3.2)과 같은 가중치를 사용하기 위해 ALS 데이터에 linear regression 모델을 적합한 후 $\hat{\beta}^{OLS}$를 계산했다. 모델 적합에는 R의 `lm`[5]함수를 사용했다. Linear regression 적합 결과 369개의 예측변수 중 10개의 변수는 $\hat{\beta}^{OLS}$가 `NA`값으로 출력되었고 이 변수들은 adaptive LASSO 모델에 불필요하다고 판단했다. 해당 10개 변수의 $\hat{\beta}^{OLS}$를 정상적으로 계산된 나머지 359개의 $\hat{\beta}^{OLS}$ 중 최솟값인 2310.553×10^{-9}으로 대체함으로써 계수 추정지의 shrinkage가 크게 일어나도록 했다. $\hat{\beta}^{OLS}$의 최솟값을 식 (3.2)에 따라 가중치로 변환한 값은 432796.7이다.

가중치를 설정한 후 벌점모수 λ를 튜닝하기 위해 10-folds 교차검증을 실시한다. 교차검증을 위해 R의 `glmnet` 패키지[2][6]가 제공하는 `cv.glmnet` 함수를 사용한다. $[-6, -1.5]$ 구간에서 동일한 간격으로 생성된 100개의 $\log\lambda$값을 사용해 각 λ마다 MSE를 계산하고 가장 작은 MSE를 보여주는 λ값을 최종 λ로 결정한다.

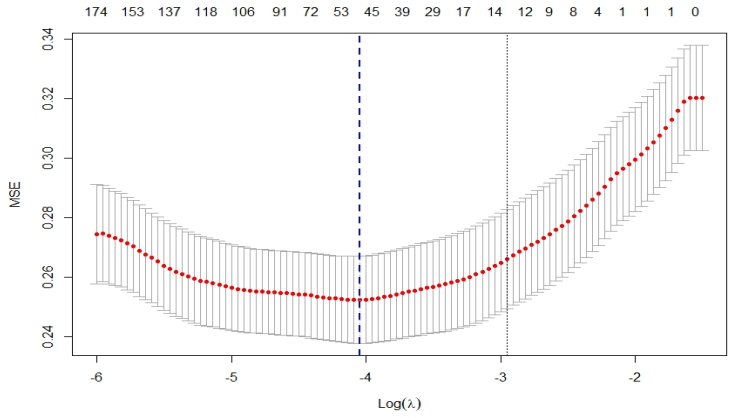

Figure 3.1: *Adaptive LASSO 모수 튜닝 : $\log\lambda$에 따른 MSE값*
최소 MSE가 나타난 $\log\lambda$값을 파란 점선으로 표시하였다.

그림 3.1과 표 3.3을 통해 $\log\lambda = -4.045$일 때의 MSE가 252.372로 가장 작음을 알 수 있다. 따라서 $\lambda = e^{-4.045}$를 adaptive LASSO의 최종 모수로 결정한다.

Table 3.3: Adaptive LASSO 모수 튜닝 : 10-folds 교차검증을 통해 계산된 $\log(\lambda)$에 따른 MSE
표에 제시된 MSE는 실제 MSE에 1000을 곱한 값이며 최소 MSE와 그에 해당하는 $\log\lambda$값은 진하게 표시했다.

$\log(\lambda)$	MSE	$\log(\lambda)$	MSE	$\log(\lambda)$	MSE	$\log(\lambda)$	MSE	$\log(\lambda)$	MSE
-1.500	320.216	-2.409	282.151	-3.318	258.68	-4.227	252.774	-5.136	257.969
-1.545	320.216	-2.455	280.417	-3.364	258.12	-4.273	252.985	-5.182	258.418
-1.591	320.166	-2.500	278.803	-3.409	257.582	-4.318	253.21	-5.227	258.812
-1.636	318.944	-2.545	277.271	-3.455	257.094	-4.364	253.49	-5.273	259.475
-1.682	315.959	-2.591	275.829	-3.500	256.675	-4.409	253.775	-5.318	260.256
-1.727	312.926	-2.636	274.446	-3.545	256.308	-4.455	254.045	-5.364	261.036
-1.773	310.157	-2.682	273.134	-3.591	255.937	-4.500	254.246	-5.409	261.867
-1.818	307.628	-2.727	271.939	-3.636	255.509	-4.545	254.405	-5.455	262.849
-1.864	305.32	-2.773	270.819	-3.682	255.088	-4.591	254.556	-5.500	263.835
-1.909	303.211	-2.818	269.643	-3.727	254.636	-4.636	254.695	-5.545	265.19
-1.955	301.286	-2.864	268.491	-3.773	254.135	-4.682	254.824	-5.591	266.436
-2.000	299.528	-2.909	267.34	-3.818	253.692	-4.727	254.96	-5.636	267.685
-2.045	297.923	-2.955	266.101	-3.864	253.326	-4.773	255.146	-5.682	268.871
-2.091	296.445	-3.000	264.897	-3.909	252.981	-4.818	255.269	-5.727	270.263
-2.136	294.933	-3.045	263.796	-3.955	252.7	-4.864	255.463	-5.773	271.268
-2.182	292.882	-3.091	262.729	-4.000	252.484	-4.909	255.723	-5.818	272.292
-2.227	290.46	-3.136	261.748	**-4.045**	**252.372**	-4.955	256.019	-5.864	273.092
-2.273	288.124	-3.182	260.863	-4.091	252.381	-5.000	256.413	-5.909	273.82
-2.318	285.962	-3.227	260.068	-4.136	252.486	-5.045	256.902	-5.955	274.568
-2.364	283.977	-3.273	259.335	-4.182	252.615	-5.091	257.43	-6.000	274.36

3.4. SCAD-penalized regression

SCAD-penalized regression은 adaptive LASSO와 같이 변수마다 다른 벌점모수가 적용되도록 해서 각 변수의 계수 추정치의 shrinkage 정도를 다르게 하는 방법이다. Adaptive LASSO는 가중치를 사용하지만 SCAD-penalized regression은 다음과 같은 SCAD-penalty 함수를 사용한다.

$$p(x;\gamma,\lambda) = \begin{cases} \lambda|x|, & \text{if} \quad |x| \le \lambda \\ \frac{2\gamma\lambda|x| - x^2 - \lambda^2}{2(\gamma-1)}, & \text{if} \quad \lambda < |x| < \gamma\lambda \\ \frac{\lambda^2(\gamma+1)}{2}, & \text{if} \quad |x| \ge \gamma\lambda \end{cases} \tag{3.3}$$

식 (3.3)의 SCAD-penalty 함수를 적용한 SCAD-penalized regression 모델을 통해 다음과 같은 SCAD-penalized regression 계수 추정치 $\hat{\beta}(\lambda,\gamma)^{SCAD}$를 구할 수 있다.

$$\hat{\beta}(\gamma,\lambda)^{SCAD} = \arg\min_{\beta} \|\mathbf{y} - X\beta\|_2^2 + p(\beta;\gamma,\lambda) \tag{3.4}$$

SCAD-penalized regression에서는 SCAD-penalty 함수의 두 모수 γ, λ에 대한 튜닝이 필요하지만 본 논문에서는 모수 γ를 튜닝하지 않고 통상적으로 많이 사용되는 3.7을 사용한다. 모수 λ를 튜닝하기 위해 10-folds 교차검증을 실시한다. 교차검증을 위해 R의 `ncvreg` 패키지[1]에서 제공하는 `cv.ncvreg` 함수를 사용한다. 모수 γ를 3.7로 고정한 후 구간 $[-6.363, -1.619]$에서 동일한 간격으로 생성된 69개의 $\log\lambda$값을 사용해 튜닝을 진행한다. 각 λ에서 MSE를 계산한 후 가장 작은 MSE를 보여준 λ값을 최종 λ로 선택한다.

Table 3.4: SCAD-penalized regression 모수 튜닝 : 10-folds 교차검증을 통해 계산된 log(λ)에 따른 MSE
표에 제시된 MSE는 실제 MSE에 1000을 곱한 값이며 최소 MSE와 그에 해당하는 log λ값은 진하게 표시했다.

log(λ)	MSE	log(λ)	MSE	log(λ)	MSE	log(λ)	MSE
-1.619	319.656	-2.875	265.109	-4.131	257.059	-5.387	301.205
-1.689	315.493	-2.944	263.218	-4.2	260.087	-5.456	308.22
-1.758	311.008	-3.014	261.583	-4.27	262.141	-5.526	389.462
-1.828	307.107	-3.084	260.304	-4.34	262.563	-5.596	416.272
-1.898	303.715	-3.154	259.192	-4.41	264.834	-5.666	411.902
-1.968	300.763	-3.224	258.254	-4.48	267.211	-5.736	415.401
-2.037	298.196	-3.293	257.472	-4.549	268.447	-5.805	418.062
-2.107	295.91	-3.363	256.761	-4.619	269.443	-5.875	412.741
-2.177	293.154	-3.433	256.359	-4.689	272.255	-5.945	415.472
-2.247	289.449	-3.503	256.644	-4.759	274.125	-6.015	423.514
-2.317	286.006	-3.572	257.202	-4.828	276.113	-6.084	435.566
-2.386	282.399	-3.642	256.778	-4.898	281.311	-6.154	429.632
-2.456	278.879	-3.712	256.219	-4.968	285.915	-6.224	434.996
-2.526	275.931	**-3.782**	**255.231**	-5.038	289.585	-6.294	410.111
-2.596	273.485	-3.852	255.232	-5.108	285.953	-6.363	400.241
-2.665	271.381	-3.921	255.678	-5.177	287.658		
-2.735	269.365	-3.991	255.842	-5.247	292.479		
-2.805	267.168	-4.061	256.502	-5.317	296.287		

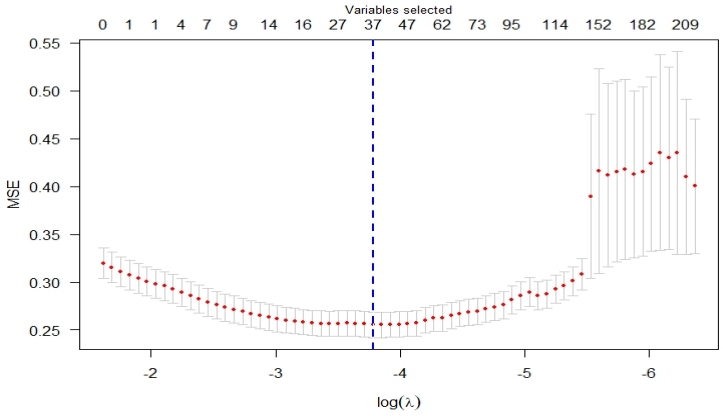

Figure 3.2: *SCAD-penalized regression 모수 튜닝 : $\log\lambda$에 따른 MSE값*
최소 MSE가 나타난 log λ값을 파란 점선으로 표시하였다.

표 3.4와 그림 3.2를 통해 $\log\lambda = -3.782$일 때의 MSE가 255.231로 가장 작음을 알 수 있다. 따라서 SCAD-penalized regression의 최종 모수는 $(\gamma, \lambda) = (3.7, e^{-3.782})$로 결정한다.

3.5. 엘라스틱 넷

2절에서 설명한 바와 같이 elastic net 모델에서는 모수 α, λ에 대한 튜닝이 필요하다. elastic net 모델의 두 모수를 튜닝하기 위해 10-folds 교차검증을 실시한다. 교차검증을 위해 R의 `glmnet` 패키지[2][6]가 제공하는 `cv.glmnet` 함수를 사용한다. 튜닝 과정에서 $0.1, 0.3, 0.5, 0.7, 0.9$의 5가지 값을 모수 α값으로 사용하고, 구간$[-6, -1.5]$에서 동일한 간격으로 생성된 100개의 값을 $\log \lambda$로 사용한다. 5개의 α값마다 교차검증을 통해 가장 작은 MSE가 계산되는 λ값을 고른 후, 최종적으로는 5가지 조합의 (α, λ)에서 가장 작은 MSE를 가지는 조합을 선택한다.

Table 3.5: Elastic net 모수 튜닝 : 10-folds 교차검증을 통해 선택된 5가지 조합의 (α, λ)와 그에 대한 MSE
표에 제시된 MSE는 실제 MSE에 1000을 곱한 값이며 최소 MSE와 그에 해당하는 (α, λ)값은 진하게 표시했다.

(α, λ)	$(0.1, e^{-2.227})$	$(0.3, e^{-2.909})$	$(0.5, e^{-3.364})$	$(0.7, e^{-3.727})$	$(0.9, e^{-3.955})$
MSE	254.536	252.883	252.532	252.434	**252.369**

표 3.5를 통해 $(\alpha, \lambda) = (0.9, e^{-3.955})$일 때 MSE값이 252.369로 가장 작음을 알 수 있다. 교차검증 결과에 따라 elastic net의 최종 모수를 $(\alpha, \lambda) = (0.9, e^{-3.955})$로 결정한다.

4. 모형의 비교

3절에서의 튜닝 결과 다음의 5가지 모델이 최종적으로 선택되었다.

Table 4.1: 튜닝 과정을 거쳐 선택된 5가지 모델의 모수 설정

모델	모수 설정
Random forest	$B = 1000, m = 30$
Boosting	$B = 150, \epsilon = 0.079, d = 2$
Adaptive LASSO	$\lambda = e^{-4.045}$
SCAD-penalized regression	$\gamma = 3.7, \lambda = e^{-3.782}$
Elastic net	$\alpha = 0.9, \lambda = e^{-3.955}$

4절에서는 표 4.1의 5가지 모델의 예측 성능을 평가 및 비교하고자 한다. 가장 예측 성능이 좋다고 판단되는 최적 모델에 대해서는 예측 과정에서 중요하게 사용된 변수를 살펴본다.

4.1. 예측 성능 비교

5가지 모델의 예측 성능을 비교하기 위해 다음과 같은 bootstrap 방법을 사용한다.

1. $i = 1, \cdots, 200$에 대해서 다음 (a)~(c)의 과정을 반복한다.
   (a) train data에서 bootstrap 데이터 ALS_i^B를 생성한다.
   (b) ALS_i^B를 사용해 표 4.1의 5가지 모델을 적합한다.
   (c) test data를 사용해 (b)에서 적합한 5가지 모델의 i번째 MSE(MSE_i)를 계산한다.
2. 1.(c)에서 얻은 200개의 MSE를 사용해 평균 MSE($\overline{MSE}$), $\overline{MSE}$의 표준오차(standard error of $\overline{MSE}$; $SE(\overline{MSE})$),

$\overline{MSE}$의 95%신뢰구간(95% $CI^{\overline{MSE}}$)을 다음 식에 따라 계산한다.

$$\overline{MSE} = \frac{1}{200}\sum_{i=1}^{200} MSE_i \tag{4.1}$$

$$SE(\overline{MSE}) = \sqrt{\frac{1}{200}\sum_{i=1}^{200}(MSE_i - \overline{MSE})^2} \tag{4.2}$$

$$95\%\ CI^{\overline{MSE}} = (\overline{MSE} + 2SE(\overline{MSE}),\quad \overline{MSE} - 2SE(\overline{MSE})) \tag{4.3}$$

식 (4.1), (4.2), (4.3)를 통해 계산된 각 모델의 $\overline{MSE}$, $SE(\overline{MSE})$, 95% $CI^{\overline{MSE}}$를 통해 예측 성능을 평가하고 모델 간 예측 성능 비교를 진행한다. $\overline{MSE}$는 각 모델의 예측 오차에 대한 추정치로 $\overline{MSE}$가 작을수록 모델의 예측 성능이 좋다고 판단한다. $SE(\overline{MSE})$와 95% $CI^{\overline{MSE}}$는 $\overline{MSE}$의 안정성을 나타내 모델의 예측 성능이 얼마나 안정적인지 보여주는 지표이다. $SE(\overline{MSE})$ 가 작을수록, 95% $CI^{\overline{MSE}}$ 가 좁을수록 모델의 예측 성능이 안정적이라고 판단한다. 추가적으로 95% $CI^{\overline{MSE}}$ 가 중첩되는 모델은 서로 비슷한 예측 성능을 가진다고 판단한다.

Table 4.2: 200번의 bootstrap 과정을 통해 계산된 $\overline{MSE}$, $SE(\overline{MSE})$, 95% $CI^{\overline{MSE}}$
표에 제시된 $\overline{MSE}$, $SE(\overline{MSE})$, 95% $CI^{\overline{MSE}}$는 실제 값에 1000을 곱한 값이다.

예측 성능 / 모델	$\overline{MSE}$	$SE(\overline{MSE})$	95% $CI^{\overline{MSE}}$
Random forest	275.472	0.247	(274.979, 275.965)
Boosting	281.508	0.665	(280.177, 282.838)
Adaptive LASSO	294.665	1.281	(292.103, 297.228)
SCAD-penalized regression	297.928	1.687	(294.555, 301.301)
Elastic net	294.964	1.286	(292.391, 297.536)

표 4.2에 나타난 $\overline{MSE}$의 경우 random forest가 275.472로 가장 작으며 SCAD-penalized regression이 297.928로 가장 크다. $SE(\overline{MSE})$의 경우도 random forest가 0.247로 가장 작고 SCAD-penalized regression가 1.687로 가장 크다. Random forest와 boosting은 1보다 작은 $SE(\overline{MSE})$를 보여주는 반면 나머지 세 모델은 1보다 큰 $SE(\overline{MSE})$가 나타난다. 95% $CI^{\overline{MSE}}$의 경우 adaptive LASSO, SCAD-penalized regression, elastic net의 95% $CI^{\overline{MSE}}$는 서로 중첩되는 부분이 존재하지만 random forest와 boosting의 95% $CI^{\overline{MSE}}$는 다른 모델의 95% $CI^{\overline{MSE}}$와 중첩되는 부분이 존재하지 않는다. 표 4.2를 시각화해서 나타낸 그림 4.2을 보면 5가지 모델의 예측 성능을 한눈에 비교할 수 있다. $\overline{MSE}$의 경우 random forest, boosting, adaptive LASSO, elastic net, SCAD-penalized regression 순서로 작은 $\overline{MSE}$값이 나타난다. 각 모델의 $\overline{MSE}$의 95% $CI^{\overline{MSE}}$를 살펴보면 random forest가 눈에 띄게 좁은 95% $CI^{\overline{MSE}}$를 가지고 있으며 random forest와 boosting이 나머지 세 모델보다 상대적으로 좁은 95% $CI^{\overline{MSE}}$를 가짐을 알 수 있다.

이러한 결과를 통해 5가지 모델의 예측 성능을 평가 및 비교하고자 한다. 먼저 random forest, boosting, adaptive LASSO, elastic net, SCAD-penalized regression의 순서로 높은 예측 성능을 보인다. Random forest와 boosting이 나머지 세 모델(adaptive LASSO, SCAD-penalized regression, elastic net)보다 안정적인 예측 성능을 보인다. 마지막으로 adaptive LASSO, SCAD-penalized regression, elastic net의 예측 성능은 비슷하다고 판단되며 random forest와 boosting의 예측 성능은 다른 모델과 유의미한 차이를 보인다. 결론적으로 random forest의 예측 성능이 가장 좋고 안정적이며 adaptive LASSO, SCAD-penalized regression, elastic net은 random forest와 boosting에 비해 예측 성능과 안정성이 떨어진다.

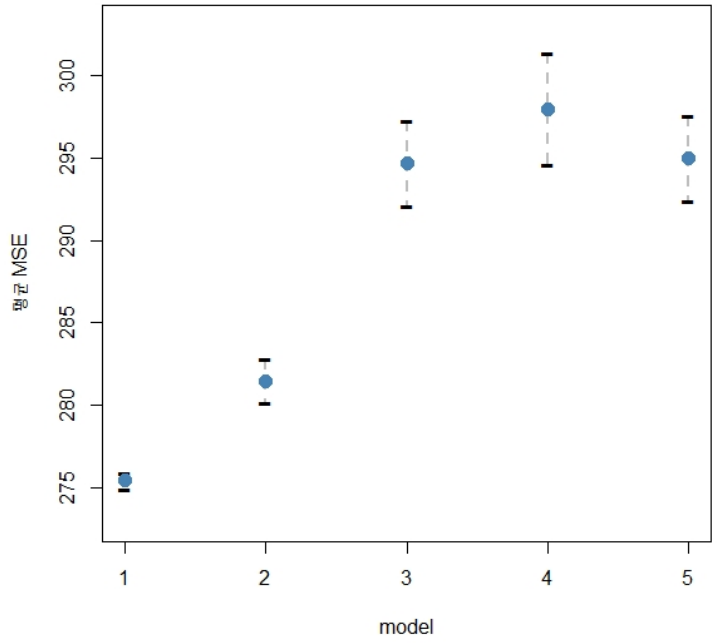

Figure 4.1: *표 4.2의 결과를 시각화한 그래프*
모델 1, 2, 3, 4, 5는 순서대로 random forest, boosting, adaptive LASSO, SCAD-penalized regression, elastic net이다.
5가지 모델의 $\overline{MSE}$는 파란색 점으로 표시되어있고 95% $CI^{\overline{MSE}}$는 회색 점선으로 표시되어있다.
y축의 단위는 실제 값에 1000을 곱해 나타냈다.

4.2. 최적 모델에 대한 해석

모델의 예측 성능은 예측 결과만을 제공하므로 모델의 예측 결과가 어떤 과정을 거쳐 도출되는지 알기 위해서는 모델에 대한 해석이 필수적이다. 이 소절에서는 4.1절에서 가장 예측 성능이 뛰어나다고 판단된 random forest 모델의 해석에 초점을 맞추고자 한다. Random forest 모델의 경우 여러 개의 decision tree를 적합하는 과정에서 각 변수가 예측 오차를 줄이는데에 기여한 정도를 합산해 변수 중요도(variable importance)를 산출하는데, 이를 통해 random forest 모델의 예측 과정에서 어떤 변수가 중요한 역할을 하는지 알 수 있다. 또한 모델에 모든 변수가 사용되는 random forest와는 달리 나머지 4개의 모델(boosting, adaptive LASSO, SCAD-penalized regression, elastic net)에서는 변수 선택이 일어난다. Random forest의 변수 중요도와 나머지 모델의 변수 선택 결과를 비교함으로써 random forest 모델과 나머지 모델의 예측 성능이 차이나는 원인을 탐구할 수 있다.

표 4.3와 그림 4.2은 random forest 모델에서의 변수 중요도를 보여준다. 그림 4.2를 보면 random forest의 경우 가장 아래에 위치한 Onset.Delta 변수의 변수 중요도가 압도적으로 높게 나타났다. 표 4.3에서도 Onset.Delta 변수의 변수 중요도가 압도적으로 높음을 확인할 수 있는데, Onset.Delta 변수의 변수 중요도는 2위인 alsfrs.score.slope 변수의 변수 중요도의 약 3.7배이고 20위인 max.slope.fvc.liters 변수의 변수 중요도의 약 9배로 나타난다. Random forest 모델의 예측 과정에서 Onset.Delta 변수가 매우 중요한 역할을 하고있음을 알 수 있다.

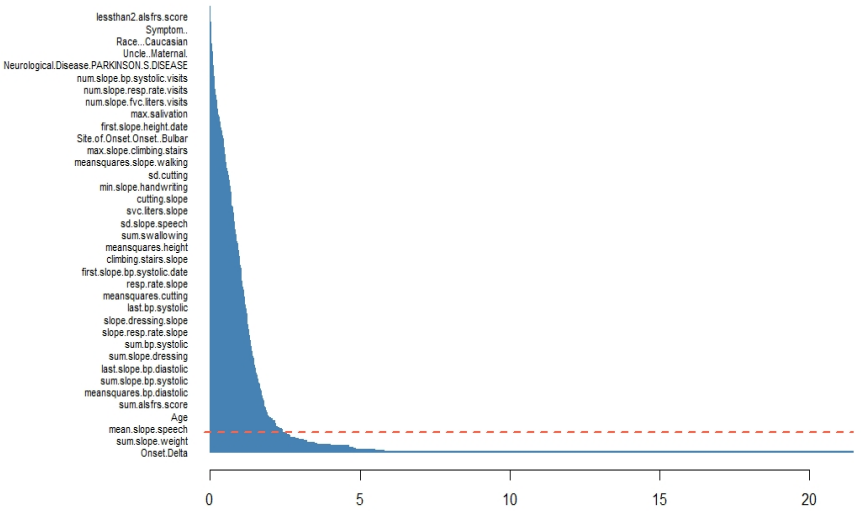

Figure 4.2: *표 4.1의 모수 설정과 train data를 사용해 적합한 random forest의 변수 중요도*
빨간 점선은 변수 중요도 상위 20위를 나타낸다.

Table 4.3: Radom forest의 상위 20개 변수 중요도와 해당하는 변수명
진하게 표시된 변수는 boosting의 decision tree 적합 시 한 번 이상 사용된 변수, Adaptive LASSO, SCAD-penalized regression, elastic net에서 선택된 변수에 포함되지 않는 변수이다. 변수 중요도 및 변수 선택 결과는 표 4.1의 모수 설정과 train data를 사용해 모델을 적합했을 때의 결과이다.

순위	변수명	변수 중요도
1	Onset.Delta	21.466
2	alsfrs.score.slope	5.798
3	**sum.slope.alsfrs.score**	**5.478**
4	**mean.slope.alsfrs.score**	**4.847**
5	sum.slope.fvc.liters	4.751
6	**mean.slope.fvc.liters**	**4.61**
7	mean.slope.weight	4.01
8	last.slope.weight	3.554
9	last.slope.fvc.liters	3.458
10	fvc.liters.slope	3.226

순위	변수명	변수 중요도
11	sum.slope.weight	3.109
12	**min.slope.alsfrs.score**	**2.932**
13	sd.alsfrs.score	2.825
14	max.slope.weight	2.666
15	**meansquares.alsfrs.score**	**2.626**
16	last.slope.bp.systolic	2.537
17	mean.speech	2.488
18	**min.slope.fvc.liters**	**2.396**
19	**mean.alsfrs.score**	**2.395**
20	max.slope.fvc.liters	2.368

그림 4.2의 빨간 점선을 통해 변수 중요도 상위 20위 이전과 이후의 변수 중요도 변화를 살펴볼 수 있다. 상위 20위 이전의 변수들은 변수 중요도가 매우 가파르게 감소하고 있는 반면 상위 20위 이후의 변수들은 상대적으로 변수 중요도가 완만하게 감소하고 있음을 확인할 수 있다.

표 4.3에서 진하게 표시된 7개의 변수는 random forest 모델에서의 변수 중요도는 상당히 높음에도 불구하고 나머지 4개의 모델에서 선택되지 않은 변수이다. 해당 7개 변수가 random forest 모델에서 예측 오차를 줄이는데 큰 도움을 준 것으로 보아, 해당 변수들의 포함 여부가 random forest와 나머지 모델의 예측 성능 차이를 야기한 것으로 보인다.

5. 결론

본 논문에서는 5가지 머신러닝 모델 random forest, boosting, adaptive LASSO, SCAD-penalized regression, elastic net 모델을 사용해 연속형 데이터를 예측해보고 각 모델의 예측 성능을 비교하였다. 모델 적합 및 예측에 사용된 데이터는 ALS 데이터로, 예측 대상인 반응변수는 루게릭병 환자의 병 진행 점수로 나타나있다. ALS 데이터를 train data와 test data로 나눠 각각 모수 튜닝 및 모델의 적합과 예측 성능 평가에 사용하였다.

5가지 모델 중 elastic net은 선형모델에 대한 추정을 위해 Zou와 Hastie (2005)[7]가 제안하였다. Elastic net은 elastic net penalty를 사용해 LASSO regression과 ridge regression을 혼합함으로써 두 모델의 단점을 보완했다. Elastic net의 장점으로는 고차원 데이터에서의 자유로운 변수 선택, 변수들의 상관관계를 고려한 변수 선택 등이 있다.

모델의 예측 성능을 향상시키기 위해 5가지 모델에 대해서 모수 튜닝을 진행하였다. 모수의 튜닝은 교차검증 또는 OOB 방법을 사용해 가장 작은 MSE를 보이는 모수를 최종 모수로 선택하는 방법으로 진행하였다. 모수 튜닝 결과 표 4.1와 같이 모수가 결정되었다.

튜닝된 모수 설정을 사용한 5가지 모델의 예측 성능을 비교하기 위해 bootstrap 방법을 사용하였다. 예측 성능을 평가하기 위해 bootstrap 과정에서 $\overline{MSE}$, $SE(\overline{MSE})$, 95% $CI^{\overline{MSE}}$의 세 가지 값을 계산하였다. 그 결과 random forest의 예측 성능이 가장 뛰어나고 안정적이며 adaptive LASSO, SCAD-penalized regression, elastic net은 random forest와 boosting에 비해 예측 성능과 예측 안정성이 떨어진다는 결론을 내릴 수 있었다.

예측 성능이 가장 뛰어난 모델로 선택된 random forest 모델의 해석을 위해 random forest 모델의 변수 중요도를 살펴보았다. Onset.Delta 변수의 변수 중요도가 압도적으로 높았으며 변수 중요도 상위 20위 이후로는 변수 중요도가 원만하게 감소함을 확인할 수 있었다. 또한 random forest의 변수 중요도 상위 20개 변수 중 7개 변수 sum.slope.alsfrs.score, mean.slope.alsfrs.score, mean.slope.fvc.liters, min.slope.alsfrs.score, meansquares.alsfrs.score, min.slope.fvc.liters, mean.alsfrs.score는 나머지 4개 모델의 적합과정에서 선택되지 않았으며 이는 random forest와 나머지 모델의 예측 성능이 차이나는 원인으로 생각할 수 있다.

References

[1] Patrick Breheny and Jian Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Annals of Applied Statistics*, 5(1):232–253, 2011.

[2] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[3] Brandon Greenwell, Bradley Boehmke, Jay Cunningham, and GBM Developers. *gbm: Generalized Boosted Regression Models*, 2020. R package version 2.1.8.

[4] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[5] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.

[6] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011.

[7] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

연속형 자료 예측을 위한 머신러닝 모델 비교 연구

윤아연[a]

[a]고려대학교 통계학과

요 약

머신러닝(machine learning) 모델은 알고리즘이 주어진 데이터를 학습함으로써 새로운 데이터에 대한 예측을 수행하도록 하는 모델을 뜻한다. 본 논문은 5가지 머신러닝 모델인 랜덤 포레스트(random forest), 부스팅(boosting), 어댑티브 라쏘(adaptive LASSO), 스캣-벌점 회귀(SCAD-penalized regression), 엘라스틱 넷(elastic net)을 사용해 연속형 데이터인 루게릭병 환자의 병 진행점수를 예측하는 모델을 적합하고 가장 예측 성능이 뛰어난 모델을 찾고자 한다. 5가지 모델에 대한 모수 튜닝(tuning)을 진행한 후 결정된 모수 설정을 이용해 각 모델의 예측 성능을 평가 및 비교한다. 가장 예측 성능이 좋다고 판단된 모델에 대해서는 예측 과정에서 중요한 역할을 하는 변수를 찾아 모델 해석을 진행한다.

주요용어: 랜덤 포레스트, 부스팅, 어댑티브 라쏘, 스캣-벌점 회귀, 엘라스틱 넷, 모수 튜닝