고등지능정보공학2 기말고사

지능형데이터·최적화학과
2024321646 김현진

(1) 히스토그램 기반 이상치 제거

회귀문제에서 이상치는 예측 성능을 크게 왜곡시킬 수 있으므로 적절할 수준에서 이상치를 제거해야 한다. 출력변수 y의 histogram 그래프는 아래 그림 1과 같으며, 이 중 백분위수 99.99 이상인 경계는 230.98로 아래 그래프에 빨간 점선으로 표현되어 있다. 백분위수 99.99인 230.98을 초과하는 데이터는 757번 데이터로, 그 값은 265.32이다. 이 값은 전체 데이터의 99.99%보다 높아 이상치로 간주할 수 있으며, 따라서 향후 분석에서는 이 값을 제외하고 사용한다.

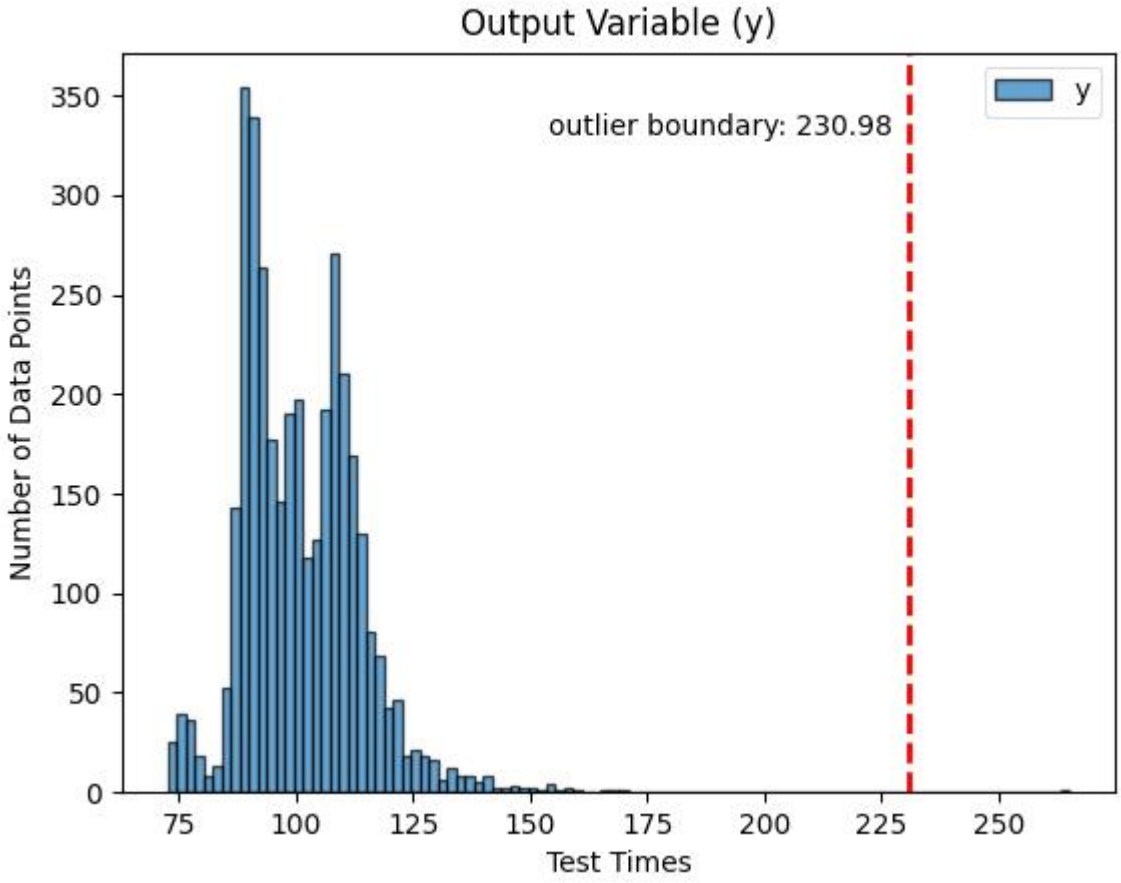

그림 1 출력변수 y의 값에 대한 히스토그램, 백분위수 99.99 기준인 230.98이 빨간 점선으로 표현되어 있으며, 250 이상의 영역에 작게 하나의 데이터 포인트가 관찰된다.

(2) 분산 기반 설명력이 낮은 변수 제거

분산이 너무 작은 입력변수는 예측 문제 해결에 기여하는 바는 적고 계산 비용은 높여 사전에 제거하는 것이 좋다. 입력변수 X의 변수는 범주형 변수 8개와 이진 변수 368개로 이루어져 있다. 먼저 범주형 변수의 분산은 각 변수의 각 범주별 출력변수 y의 분산을 계산하고 각 범주별 분산의 평균을 변수의 평균으로 활용하며, 이진 변수의 경우 표본 분산을 계산한 뒤 분산이 0이거나 동일한 분산을 갖는 변수를 제거한다.

각 범주형 변수 X0 ~ X8에 대해 출력변수 y의 값을 이용하여 나타낸 Box Plot은 아래 그림 2 와 같으며, 각 변수의 평균 분산은 Box Plot의 우측에 표시된 바와 같으며, 평균값과 중앙값은 Box Plot 좌측에 표시된 바와 같다.

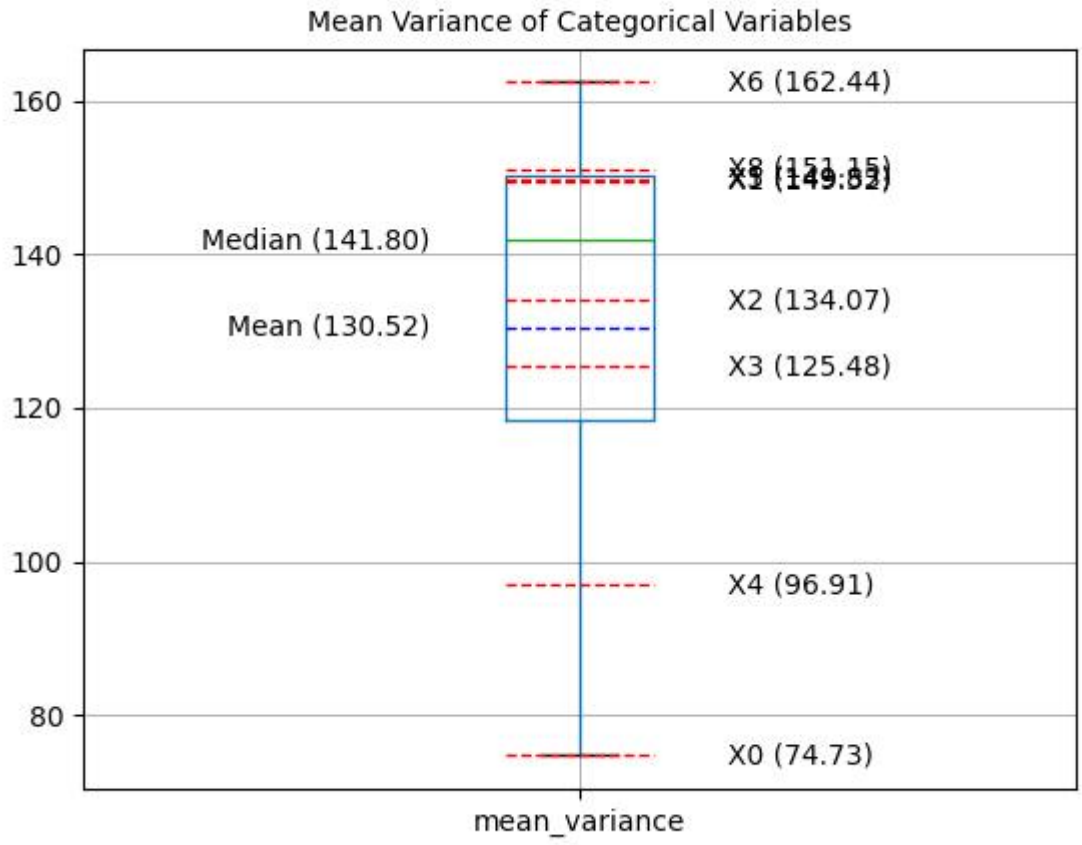

그림 2 범주형 변수의 평균 분산을 나타낸 Box Plot. 각 변수의 평균 분산이 빨간 점선으로 표시되어 있고 우측에 해당 변수명과 평균 분산 값이 적혀 있다. Box의 좌측에는 Median 값과 Mean 값이 각각 표시되어 있으며, 각각 Box 내부에서 초록색 실선과 파란색 점선으로 표시되어 있다. Box 하단, 즉 백분위수 25 이하의 평균 분산 값을 가지는 변수로 X4와 X0가 있음을 알 수 있다. 또한 다수의 변수가 백분위수 75 주변에 모여있는 것이 관찰된다.

이에 따라 분산이 가장 작은 변수는 X0이며, 백분위수 25 이하인 다른 변수로는 X4가 있음을 알 수 있다. 따라서 변수 X0와 X4를 제거하여 계산의 효율성과 모델의 성능 향상을 도모한다.

또한 X10 ~ X385는 이진 변수로 각 값이 0과 1로 이루어져 있다. 따라서 각 변수별로 분산을 계산할 수 있다. 먼저 각 변수의 표본평균 $\overline{X}$는 1의 등장 확률과 동일하며 X_i는 데이터 X의 i번째 데이터 포인트, n은 표본의 크기일 때 표본분산 s^2은 아래의 수식와 같이 계산할 수 있다.

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

이제 위의 식으로 구한 각 변수의 분산을 산점도로 나타내면 아래의 그림 3과 같이 표현할 수 있다. 변수의 수가 너무 많아 그림 내 변수의 이름은 생략하였다.

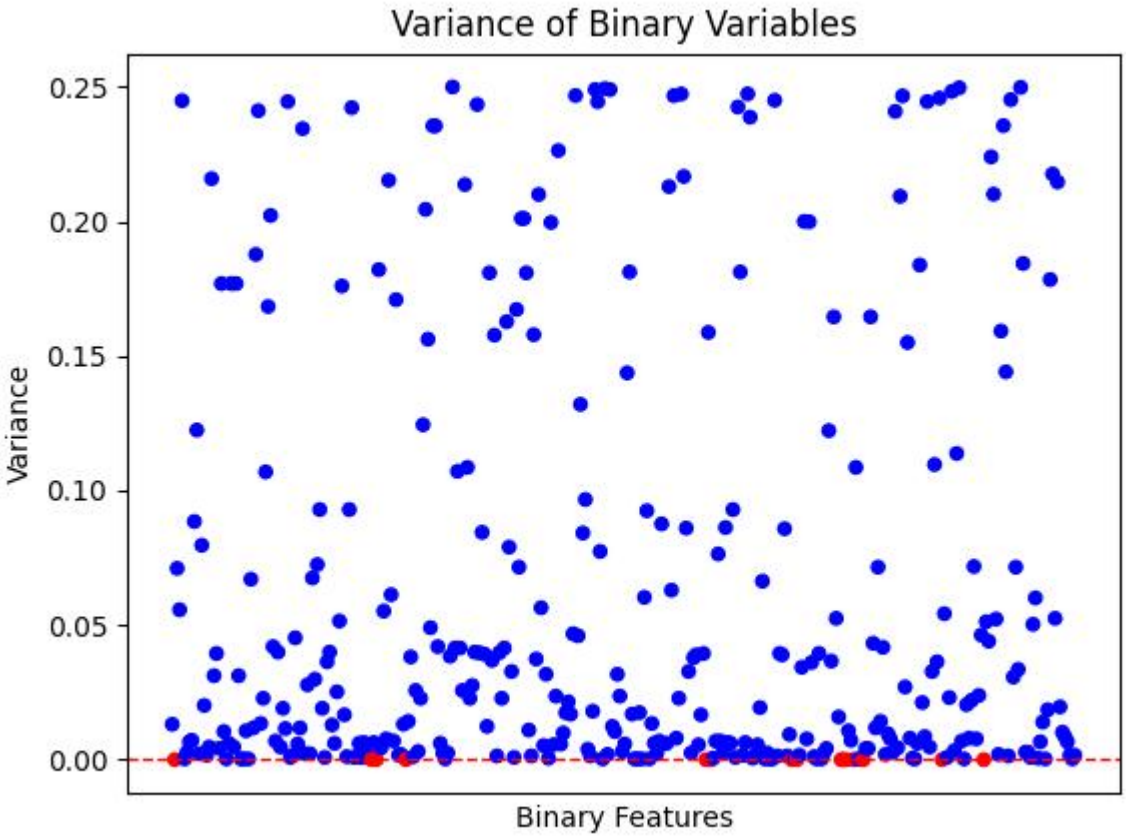

그림 3 이진 변수별 분산의 산점도 그래프, 빨간 점선으로 표시된 분산 0 위치에 총 15개의 점이 위치해 있으며, 이외에도 서로 동일한 분산을 가지는 변수가 187개 존재한다.

빨간 점선으로 표시된 $y=0$을 분산으로 가지는 변수가 X11 등 15개, 그리고 서로 동일한 분산을 가지는 변수가 X10 등 185개 있으며, 동일한 분산값을 가지는 변수에는 분산을 0으로 가지는 변수가 모두 포함된다. 따라서 총 185개의 변수를 제거하여 모델의 계산 효율성과 성능 향상을 도모한다.

따라서 최종적으로, 범주형 변수 2개, 이진 변수 185개, 즉 187개의 변수를 제거하여 189개의 변수로 회귀 분석을 실시한다.

(3) 범주형 변수 대상 Label Encoding

Label Encoding이란 범주형 변수의 범주를 각각 고유한 수로 encoding하여 컴퓨터가 효율적으로 처리할 수 있도록 하는 방법론을 말한다. Label Encoding을 위해서 sklearn 라이브러리에서 제공하는 LabelEncoder를 활용하였으며, 결과는 아래와 같다.

ID	X1	X2	X3	X5	X6	X8
0	v	at	a	u	j	o
1	t	av	e	y	l	o
2	w	n	c	x	j	x
3	t	n	f	x	l	e
5	b	e	c	g	h	s
6	r	e	f	f	h	s
8	s	as	e	f	i	h
10	r	r	f	f	h	p
11	r	e	f	f	h	o

$\Rightarrow$

X1	X2	X3	X5	X6	X8
23	16	0	23	9	14
21	18	4	27	11	14
24	32	2	26	9	23
21	32	5	26	11	4
3	23	2	11	7	18
19	23	5	10	7	18
20	15	4	10	8	7
19	36	5	10	7	15
19	23	5	10	7	14

표 1 범주형 변수 X1 ~ X8의 원래 값(좌측)과 대한 Label Encoding 결과(우측). Label Encoding 결과가 크기 혹은 순서로 해석되지 않도록 모델 설계에 주의가 필요하다.

물론 이 값은 범주의 크기나 순서를 표현하는 값이 아니므로 모델이 이를 크기나 순서로 해석하지 않도록 주의하여야 하며, 특히 선형 모델이나 거리 기반 알고리즘을 사용할 때 더더욱 주의를 기울여서 모델을 설계해야 한다.

(4) 교호작용 변수의 선형회귀를 활용한 변수 제거

교호작용 변수란 두 개 이상의 입력변수의 상호작용을 나타내는 변수로 변수 X_i와 X_j의 교호변수는 X_iX_j, 즉 두 데이터 벡터의 내적으로 나타낼 수 있다. 이 때 생성되는 교호작용 변수는 총 입력변수의 수 p에 대하여 $\frac{p(p+1)}{2}$만큼 생성되며, 따라서 189개의 변수에서 교호작용 변수를 구성하면 총 17,955개의 교호작용 변수가 생성된다.

(1) ~ (3) 단계에서 제거한 입력변수로 교호작용 변수를 구성하고 일반 선형회귀를 수행함으로써 각 교호작용 변수의 설명력을 구할 수 있다. 선형회귀 결과는 각 입력변수가 출력변수에 유의미한 영향을 미치지 않는다는 귀무가설을 기각하기 위해 p-value를 검정하는데, p-value가 0.05 이상인 경우 교호작용 변수 X_iX_j가 출력변수 y에 유의미한 영향을 미치지 않는다는 귀무가설을 기각할 수 없으므로, p-value가 0.05 이상인 교호작용 변수를 모두 제거한다. 이 과정을 통해 총 17,955개의 교호작용 변수 중 15,403개의 변수를 제거하고 2,552개의 변수를 남길 수 있다.

(5) Feature Engineering 방법론

(1) ~ (4)까지의 단계를 통해 최종적으로 2,552개의 입력변수를 확보한다. 이는 최초 376개의 입력변수에서 상당히 증가한 양인데, 2,552개의 입력변수를 그대로 활용하거나 Feature Engineering을 통해 입력변수의 수를 줄여 예측 모델의 입력 변수로 사용할 수 있다. 각 방법론은 아래와 같다.

(가) Linear PCA

Linear PCA는 분산이 큰 변수가 데이터에 대한 설명력이 높다는 가정에서 시작하며, 각 변수의 선형 결합을 활용하여 주성분을 구성한 뒤 일정 정도 이상의 설명력을 가지는 만큼의 주성분만으로 데이터를 재구성함으로써 입력변수의 차원을 축소한다.

(나) Kernel PCA

Kernel PCA는 입력변수 간의 비선형적 관계를 Linear PCA가 잘 반영하지 못한다는 문제점에서 출발한다. 이를 위해 데이터를 비선형 변환을 통해 고차원 feature space로 mapping하고, 이 고차원 feature space에서 다시 Linear PCA를 수행한다. 하지만 이 과정은 입력변수의 수와 데이터의 증가에 따라 계산량이 무한대로 증가한다. 따라서 실제로 mapping하지 않더라도 같은 효과를 얻어내는 것이 Kernel PCA의 핵심이다.

Kernel 함수는 Original Space L에서 정의된 두 개의 feature vector x_i, x_j에 대해 $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$를 만족하는 변환함수 ϕ가 존재할 때 $k(x_i, x_j)$를 일컫는 말이다. 대표적인 종류와 그 수식은 아래와 같다.

Kernel 함수의 종류	수 식	비 고
Polynomial	$k(x_i, x_j) = (x_i \cdot x_j)^d$	-
Gaussian(Radial Basis)	$k(x_i, x_j) = e^{(-\gamma \| x_i - x_j \|^2)}$	$\gamma > 0$ or $\gamma = \frac{1}{2\sigma^2}$
Hyperbolic Tangent(Sigmoid)	$k(x_i, x_j) = \tanh(x x_i \cdot x_j + c)$	주로 $x > 0, c < 0$

표 2 Kernel 함수의 종류와 그 계산식

이러한 Kernel 함수를 사용하여 변환함수 ϕ가 없어도 데이터를 고차원 feature space에 mapping한 뒤 Linear PCA를 수행한 것과 같은 결과를 수학적으로 확보할 수 있다.

(다) Gaussian Random Projection

Gaussian Random Projection은 Gaussian 분포에서 샘플링된 무작위 행렬을 새로운 기저로 하여 입력변수를 투영하는 방식이다. 고차원 데이터를 저차원으로 임의로 투영하더라도 일정 수준의 데이터 간의 거리가 보존됨이 알려져 있다. 즉, $n \times d$ 입력변수 X에 대해 무작위 $d \times k$ 행렬 R은 $r_{ij} \sim N(0, \frac{1}{d})$로 정의되며, 이때 d는 입력변수의 차원, k는 목표 차원이다. k차원으로 투영된 데이터 X'은 $X' = X \cdot R$로 계산된다.

이때 k값은 아래와 같이 결정할 수 있다. ϵ은 거리 왜곡 허용 오차 값으로, 작을수록 더 많은 차원이 필요하지만 거리 왜곡이 줄어들고 크면 차원을 많이 줄이지만 왜곡이 커지는 hyperparameter이다. 본 실험에서는 $\epsilon = 0.2$로 실험을 진행하였다.

$$k \geq \frac{4\ln(n)}{\epsilon^2/2 - \epsilon^3/3}$$

이 방법은 기존 PCA 방식에 비해 계산 비용이 매우 낮아 대용량 데이터를 처리하는데 적합하며, 어느 정도 거리를 보존하기에 유용하다. 다만 데이터 간의 비선형 구조를 고려하지 않고 무작위로 투영하기 때문에 비선형 구조가 보존되어야 하는 경우 부적합할 수 있다.

(6) 각 Feature Engineering 방법별 예측모델 조합 및 최종 모델 선정

(5)에서 설명한 Feature Engineering 방법론들을 각 예측모델과 함께 활용할 수 있다. 활용할 예측모델은 Random Forest, PLS, Kernel Ridge, Gaussian Process Regression이며, 각 모델에 대한 설명은 아래와 같다.

(가) Random Forest

원 입력변수 X의 변수 중 q개를 random sampling을 통해 추출한 뒤, 정해진 depth와 leaf node의 수를 가진 여러 개의 Decision Tree 모델에 base learner로서 동시에 학습하여 그 결과를 bagging 하여 최종 class를 결정한다. 여러 Decision Tree를 동시에 학습함으로써 병렬화를 통해 학습 속도를 향상하고 구조도 단순하여 대중적으로 많이 활용된다. 본 실험에서는 200개의 Decision Tree를 구성하여 모델의 설명력을 측정하였다.

(나) PLS(Partial Least Square Regression)

PLS는 입력변수의 수가 데이터의 수보다 많을 때(large p, small n) 많이 활용되는 회귀모델이다. PLS는 입력변수 X와 출력변수 Y에 대해 각각 주성분 분석을 수행하면서 서로 간의 연관성이 최대화되는 각자의 Score Matrix를 찾는 방식이다. 이 과정에서 입력변수와 출력변수의 loading 벡터를 활용하여 차원을 축소한 뒤 OLS를 수행한다. 본 실험에서는 차원 축소의 한계는 eigenvalue의 누적 설명력이 0.95를 넘는 범위까지로 설정하였다.

(다) Kernel Ridge

Kernel Ridge는 Ridge 회귀에 Kernel 기법을 활용하는 모델이다. Ridge 회귀는 일반적인 선형회귀 모델에 L2 정규화항을 도입한 회귀 분석 방법론으로, 기존 선형회귀 방법론에 비해 과적합 방지에 효과적이면서도 계산량이 많이 늘어나지 않는다.

하지만 Ridge 회귀 모델은 기본적으로 선형 모델이기 때문에, 입력 변수가 비선형 관계를 가지고 있을 때 이를 포착하기 어려운데, Ridge 회귀에 Kernel 기법을 도입함으로써 이 문제를 효과적으로 처리할 수 있다. 이를 Kernel Ridge라고 부른다. 본 실험에서는 Gaussian Kernel, 즉 RBF(Radial Basis Function)을 활용하였으며, $\gamma = 0.5$로 설정하였다.

(라) Gaussian Process Regression

Gaussian Process Regression은 비선형 회귀 문제를 해결하는 방법 중 하나이다. 입력 변수의 데이터를 통해 어떤 함수 f의 함숫값의 분포를 학습하고 새로운 입력에 대해 확률 분포 형태의 예측을 반환함으로써, 예측값뿐만 아니라 그 불확실성도 표현하는 방법론이다.

이때, 함수 f의 함숫값이 평균이 0, 분산이 Kernel 함수인 Gaussian 분포를 따른다고 가정함으로써 예측값의 분포와 표준편차를 구할 수 있다.

각 Feature Engineering 방법론별 선택된 입력변수의 개수와 각 모델별 설명력(R^2)은 아래의 표 3과 같다. 이때 모델의 설명력은 5-fold Cross Validation을 통해 평균 R^2값을 계산하였으며, Linear PCA와 Kernel PCA에서의 target variance는 0.9로 설정하였다.

Feature Engineering 방법론	입력 변수 개수	모델별 설명력(R^2)			
		Random Forest	PLS	Kernel Ridge	Gaussian Process Regression
(1) ~ (4)	2,552	0.49	-7.05	-60.71	-59.35
(1) ~ (4) + Linear PCA	30	0.47	0.49	-59.37	-57.56
(1) ~ (4) + Kernel PCA	2,299	0.02	-2865.05	-0.27	-0.77
(1) ~ (4) + Gaussian Random Projection	1,890	0.48	0.50	-60.69	-59.32

표 3 각 Feature Engineering 방법론과 모델에 따른 설명력(R^2)

표 3의 결과에 따라 최초 설정 상태에서 가장 설명력이 우수한 조합은 Gaussian Random Projection으로 입력 변수를 축소한 뒤 PLS 모델로 회귀 분석을 실시하는 것이다. 이는 교호작용 변수가 도입되면서 변수의 수가 급격하게 늘어나 PLS의 설명력이 높아진 것에 기인한다고 판단된다.

PLS에서 현재 활용할 수 있는 Hyperparameter는 입력 변수의 설명력 threshold로, 기본 값은 0.95로 설정하였으나 0.9, 0.95, 0.99로 변화시키며 학습을 진행하였다. 또한 Gaussian Random Projection의 경우 ϵ 값에 따라 선택되는 변수의 수가 다른데, 각각 0.1, 0.2, 0.3로 설정해 진행하였다.

Gaussian Random Projection		PLS threshold					
ϵ	선택된 변수 개수	0.9		0.95		0.99	
		R^2	변수 개수	R^2	변수 개수	R^2	변수 개수
0.1	7,019	-8.60	4,671	-8.60	4,671	-8.60	4,671
0.2	1,890	0.50	57	0.50	57	0.50	57
0.3	910	0.49	56	0.49	56	0.49	56

표 4 각 Feature Engineering 방법론과 모델에 따른 설명력(R^2)

따라서 표 4의 결과에 따라 최적의 조건은 threshold와 상관없이 $\epsilon = 0.2$ 인 것을 알 수 있다. 이 결과에서는 ϵ의 값이 커질수록 선택된 변수의 개수가 줄어들기는 하지만, 일정 수준 이상에서 유의미한 변수 축소 영향을 주지 못하는 것을 알 수 있다. 이에 따라 PLS 모델의 결과에도 일정 수준 이상의 ϵ의 값은 오히려 모델의 성능을 감소시키는 것으로 추정된다. 해당 최종 결과물은 threshold를 0.95로, ϵ을 0.2로 설정하여 X_test에 대한 예측을 수행하였으며, 그 결과값은 1번.csv에 정리하였다.

(7) 자신만의 모델과 전처리 방법을 통한 자유로운 최종 모델 조합 선정

(가) 전처리 방법

입력 데이터의 전처리를 위해 위에서 사용했던 전처리방법론을 일부 차용하였다. 백분위 수 99.99 이상의 데이터를 이상치로 간주하여 제거하고, 범주형 변수 중 분산이 낮은 변수 2개를 제거한다. 이후 각 범주에 대해 label encoding을 실시한 뒤, 교호작용 변수를 구성한다.

다만, 이진 변수의 분산 중 0인 값과 동일한 값뿐만 아니라 box plot을 그려 분산이 백분위수 25 이하, 즉 IQR 박스보다 아래의 whisker에 해당하는 변수 92개를 제거한다.

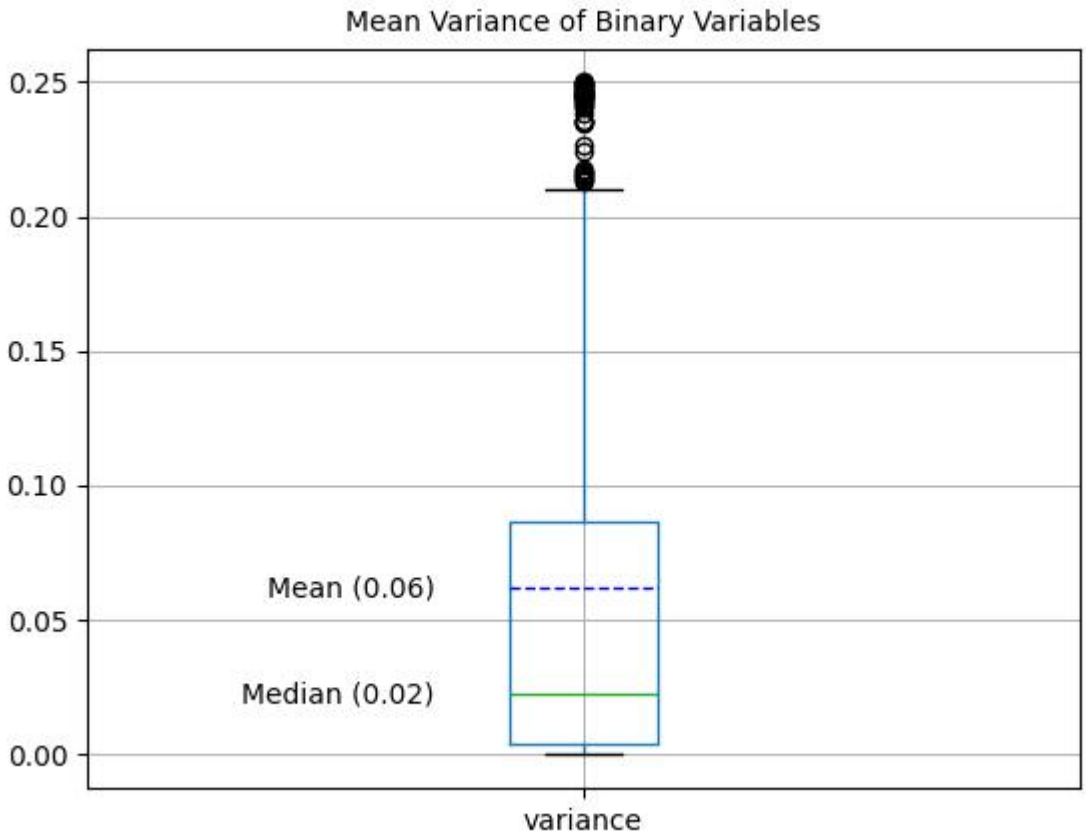

그림 4 이진 변수의 분산에 대한 Box Plot, IQR 박스 이하의 변수 92개를 제거하여 분석에 활용한다.

그리고 교호작용 변수를 구성한 뒤, 전체 변수의 Variable Importance Ranking in Random Forest를 구해 변수별 중요도를 계산하고, 전체 중요도의 95%를 설명하는 변수를 선택한다. 이러한 전처리 과정을 통해 최종적으로 57개의 변수를 선택한다.

(나) 회귀 분석 모델

회귀 분석 모델로는 Gaussian Process Regression(GPR)을 활용한다. 위에서 언급했듯, Gaussian Process Regression은 비선형 회귀문제를 해결하기 위한 회귀 분석 모델이다. 본 실험에서는 교호작용 변수까지 고려하기 때문에 데이터가 선형적 구조이기보다 비선형적 구조일 것임을 가정하는 것이 타당하다.

GPR은 3개의 hyperparameter $\theta = \left\{l, \sigma, \sigma_y^2\right\}$를 가진다. l은 Gaussian Kernel에서의 길이의 범위를 의미하며, σ는 kernel의 높이(amplitude)를 의미한다. σ_y^2는 $y_i = f_i + \epsilon$일 때 ϵ의 분산, 즉 noise의 분산을 의미한다. 각각의 hyperparameter는 Kernel에 결합되어 사용되며, kernel k는 아래와 같이 정의할 수 있다.

$$k = \sigma^2 \times e^{-\frac{\| x - x' \|^2}{2l^2}} + \sigma_y^2 \delta(x, x')$$

이때, $\sigma^2 \times e^{-\frac{\| x - x' \|^2}{2l^2}}$는 입력된 데이터 간 거리를 기반으로 유사도를 측정하여 데이터의 패턴을 반영하며, $\sigma_y^2 \delta(x, x')$ 항을 더함으로써 노이즈 항을 추가하여 데이터의 불확실성을 모델링할 수 있다. 각 변수의 범위는 아래 표와 같이 설정하였으며, 해당 범위 내에서 Log-Marginal Likelihood(LML)을 최대화하게끔 optimizer를 실행하여 최적 parameter를 선정하였다.

변 수	의 미	초기값	최소 범위	최대 범위	최적값
l	입력 데이터 간 거리의 영향	0.1	1e-2	1e2	10.3
σ	출력값의 진폭	0.1	1e-3	1e3	31.6
σ_y^2	데이터의 노이즈의 진폭	1e-10	1e-10	1e1	10

표 5 각 hyperparameter 별 의미와 초기값 및 범위

표 5에 따라 최적의 hyperparameter 값은 $\left\{l = 10.3, \sigma = 31.6, \sigma_y^2 = 10\right\}$임을 알 수 있으며, 이에 따른 X_test 데이터의 예측 결과는 2번.csv에 정리하였다.