금융과인공지능 3차과제

2021510010 이상민

1.

1-1) 아래 <표 1>은 수정된 Baby Example에 대한 소프트마진 SVM 모형 결과이다.

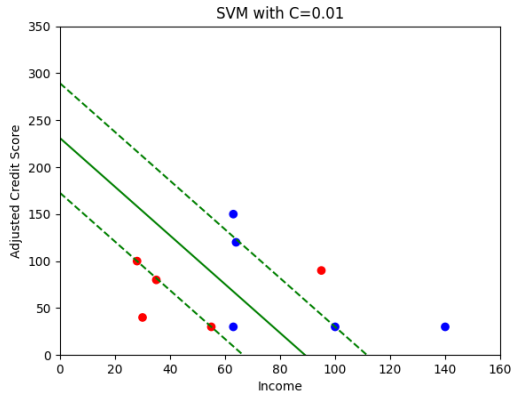
C	w1	w2	b	대출 오분류	통로 너비
0.01	0.044444	0.017143	3.958713	2	41.9852
0.001	0.039669	0.012231	3.333884	2	48.1785
0.0005	0.037869	0.013424	3.399323	2	49.7782
0.0003	0.0231	0.009	2.2785	2	80.6734
0.0002	0.0187	0.0057	1.789	2	102.3048

<표 1. 소프트 마진 SVM 모형 결과>

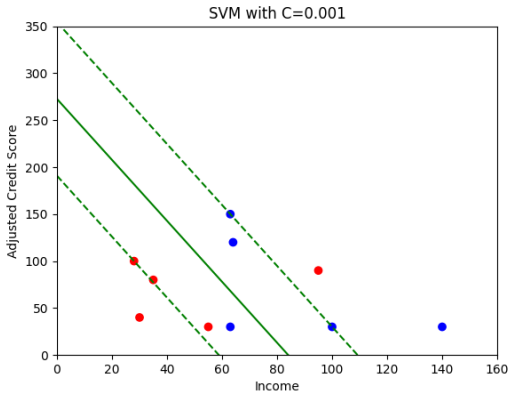
분석 결과, 대출 오분류의 수는 모든 C값에서 2로 동일하게 나타났다. 이는 데이터의 구조적인 특성 때문으로 , 겹치는 데이터 혹은 분류 불가능한 데이터가 존재하기 때문에 C값을 조정하더라도 더이상 오분류를 줄일 수 없는 한계가 있음을 의미한다. C가 클수록 일반적으로 오분류를 줄이기 위해 모델이 더 복잡한 결정 경계를 학습하지만, 본 데이터에서는 이러한 변화가 영향을 미치지 못했다.

C값이 작아질수록 통로 너비(마진)가 증가하는 경향을 보였다. C = 0.01일 때 통로 너비는 41.8952로 가장 좁았다. 이는 모델이 오분류를 최소화하려고 더 복잡한 결정 경계를 학습하며, 마진을 줄인 결과다. 반대로 C = 0.0002일 때 통로 너비는 102.3048로 가장 넓었으며, 이는 모델이 더 단순화되어 마진 내에 더 많은 데이터를 포함하려고 한 결과다. C 값이 작아지면서 결정 경계는 완만해지고, 이는 일반화 성능의 향상을 의미할 수 있다. 하지만, 지나치게 작은 C 값은 모델의 과도하게 단순해져, 결정 경계가 데이터의 세부적인 분포를 반영하지 못할 위험을 내포한다.

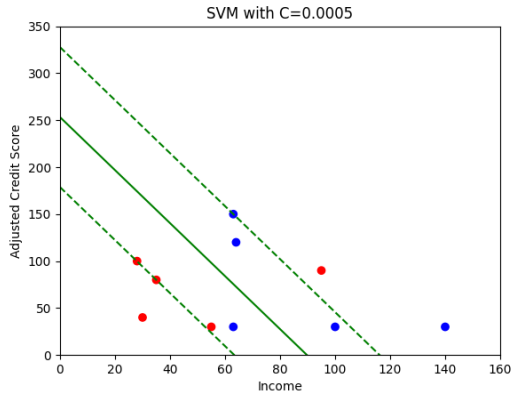
C 값에 따라 SVM의 가중치(w1, w2)와 절편(b)도 변화하였다. C가 클수록 가중치의 값이 증가하였으며, 이는 결정 경계가 더 가파르게 형성됨을 보여준다. C가 작아질수록 가중치의 값이 감소하며, 결정 경계가 더 완만해지고, 더 넓은 마진을 형성한다. 이와 같은 결과를 아래 <그림 1> ~ <그림 5>를 통해 확인 할 수 있다.



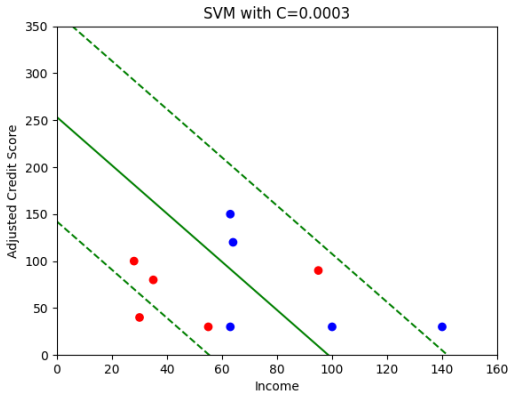
<그림 1. SVM 모형 결과 (C = 0.01)>



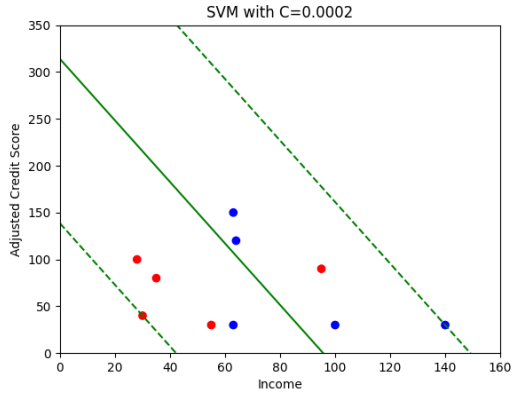
<그림 2. SVM 모형 결과 (C = 0.001)>



<그림 3. SVM 모형 결과 (C = 0.0005)>



<그림 4. SVM 모형 결과 (C = 0.0003)>



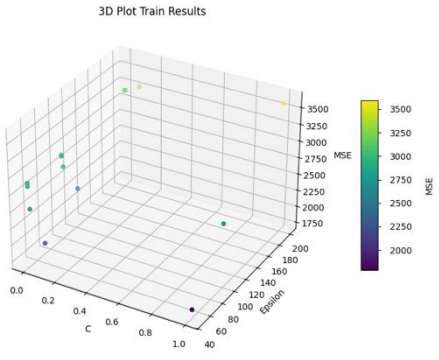
<그림 5. SVM 모형 결과 (C = 0.0002)>

2.

2-1) 아래의 <표 2>와 <그림 6>은 IOWA 훈련 데이터를 이용해 Hyperparameter를 변경해가며 분석을 진행한 결과와 시각화 자료이다.

C	e	w1	w2	b	MSE	통로 너비
0.0001	50	0.118	0.0631	0.0006	2909.3972	100
0.0001	100	0.1225	0.0257	0.0006	2930.9266	200
0.0001	200	0.1298	0.0034	0.0001	3168.4293	400
0.001	50	0.1162	0.6095	0.0039	2859.3814	100
0.001	100	0.1217	0.2533	0.0053	2911.6415	200
0.001	200	0.1297	0.0324	0.0011	3169.0712	400
0.01	50	0.0997	4.9189	-0.0988	2530.1087	100
0.01	100	0.1151	2.2899	0.0409	2762.5472	200
0.01	200	0.1293	0.3222	0.0104	3176.8603	400
0.1	50	0.0612	16.0956	-4.1981	2082.9824	100
0.1	100	0.0815	12.987	-0.1402	2493.4371	200
0.1	200	0.1259	2.2777	0.0309	3281.7109	400
1	50	0.0537	24.0687	-40.3937	1786.0145	100
1	100	0.0493	23.5047	-8.4159	2614.57	200
1	200	0.0979	11.815	0.2017	3595.2326	400

<표 2. 각 Hyperparameter에 대한 분석 결과>

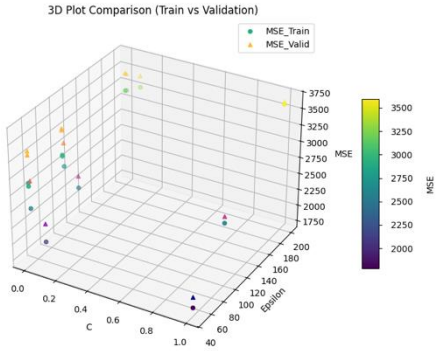


<그림 6. 훈련 데이터 분석 결과>

2-2) 아래의 <표 3>은 검증 데이터로 평가하고 각 모형에 대한 MSE 결과이다. 비교를 쉽게 하기 위해 훈련 데이터와 검증 데이터의 MSE의 차에 대한 열도 추가했다.

C	Epsilon	MSE_Train	MSE_Valid	Passage Width	MSE Difference
0.0001	50	2909.0839	3385.661	100	476.5771
0.0001	100	2930.9267	3330.2458	200	399.3192
0.0001	200	3168.4293	3440.0185	400	271.5892
0.001	50	2859.3661	3326.3287	100	466.9626
0.001	100	2911.3476	3307.5297	200	396.1821
0.001	200	3169.0712	3439.3432	400	270.272
0.01	50	2529.1928	2945.8322	100	416.6394
0.01	100	2761.9524	3119.3532	200	357.4008
0.01	200	3176.8603	3434.0552	400	257.1949
0.1	50	2083.072	2359.6927	100	276.6207
0.1	100	2494.0786	2673.7845	200	179.7059
0.1	200	3281.7109	3457.9547	400	176.2437
1	50	1784.0992	1944.4898	100	160.3906
1	100	2614.57	2713.2393	200	98.6693
1	200	3595.2326	3629.0791	400	33.8465

<표 3. 각 Hyperparameter에 대한 분석 결과>



<그림 7. 훈련 데이터와 검증 데이터 분석 결과>

2-3) 위의 결과는 IOWA Data에 대한 훈련 데이터와 훈련 데이터로 학습된 모델을 검증 데이터에 적용한 결과를 보여준다. 각각 다른 C값과 Epsilon값을 사용하여 모델을 학습하였으며, 이에 따른 MSE 값과 두 데이터셋 간의 MSE 차이를 제시한다. 먼저, 훈련 데이터와 검증 데이터의 성능 차이를 살펴보면, MSE Difference값이 상당히 큰 경우가 존재함을 알 수 있다. 특히 C 값이 작은 경우에는 MSE Difference 값이 상대적으로 높게 나타나는데, 이는 훈련 데이터에 비해 검증 데이터에서의 오차가 더 크다는 것을 의미한다. 이는 모델이 훈련 데이터에 충분히 적합하지 못하여 일반화 성능이 떨어지는 현상으로 해석될 수 있다.

반면, C 값이 큰 경우에는 MSE Difference값이 현저히 감소하는 경향을 보인다. 특히 C = 1인 경우 MSE Difference 값이 가장 낮게 나타나는데, 이는 모델이 훈련 데이터와 검증 데이터 모두에 대해 우수한 성능을 보이고 있음을 나타낸다. 이는 모델이 데이터의 패턴을 잘 학습하여 새로운 데이터에도 효과적으로 적용될 수 있음을 시사한다.

또한, Epsilon 값이 증가할수록 각 데이터에 대한 MSE 값이 전반적으로 상승하는 경향을 보인다. 이는 모델의 허용 오차가 커짐에 따라 예측 정확도가 떨어지기 때문이다. 따라서 Epsilon 값을 적절하게 설정하는 것이 모델의 성능을 향상시키는 데 중요하다고 볼 수 있다.

이러한 분석을 종합해 봤을 때, 작은 C값과 큰 Epsilon값을 사용할 때 모델이 과소적합되는 징후를 보인다. 과소적합은 모델이 데이터의 복잡한 패턴을 충분히 학습하지 못하여 훈련 데이터와 검증 데이터 모두에서 높은 오차를 보이는 현상이다. 이는 모델의 복잡도가 낮거나 규제가 과도하게 적용될 때 발생한다.

반대로, C값을 높게 설정하고 Epsilon값을 적절히 조절하면 모델의 복잡도가 증가하여 데이터의 패턴을 더 잘 학습하게 된다. 이는 훈련 데이터와 검증 데이터에서 모두 낮은 MSE 값을 가져오며, 두 데이터셋 간의 성능차이도 최소화된다. 이러한 경우 모델이 과대적합 없이 데이터에 잘 적용되었다고 볼 수 있다.

2-4) 최적의 C와 ϵ 값을 선택하기 위해 다양한 접근 방식을 활용하여 검증 데이터를 분석한 결과, 검증 데이터에서 MSE 값이 가장 낮은 경우와 MSE 차이가 가장 적은 경우를 각각 도출하였다. 먼저, 검증 데이터의 MSE 값이 가장 낮은 경우는 C = 1.0, ϵ = 50 일 때로, 이 설정에서 검증데이터의 MSE는 1944.5111로 나타났다. 해당 모델의 훈련 데이터 MSE는 1782.2306이며, 훈련 데이터와 검증 데이터 간의 MSE 차이는 162.2805로 비교적 작았다.

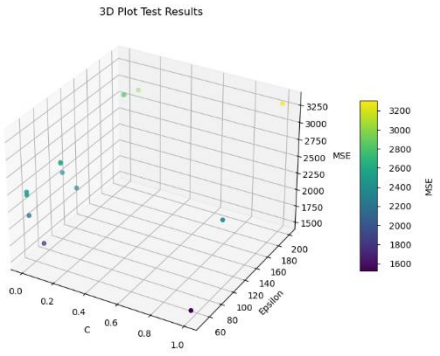
검증 데이터에서 MSE 차이가 가장 적은 경우는 C = 1.0 ϵ = 200일 때로, 이 설정에서는 MSE Difference가 33.8465로 나타났다. 훈련 데이터 MSE는 3595.2326, 검증 데이터 MSE는 3629.0791로 두 데이터셋 간의 성능 차이가 매우 작았지만 각각의 MSE값이 다른 모델에 비해 높았다.

최적의 C와 ∈값을 교차 검증 방식(GridSearchCV를 사용)을 통해 도출하였다. 교차 검증 결과, 최적의 Hyperparameter는 C = 1.0, ∈= 50으로 보여지고, 이 모델에서 훈련 데이터의 MSE = 1782.2306, 검증 데이터의 MSE는 1944.5111, MSE 차이는 162.2805인 것을 확인할 수 있다. 이는 검증 데이터에서 가장 낮은 MSE 값을 기록한 경우와 일치하며, 해당 모델이 가장 적합한 모델임을 나타낸다.

위에서 결정한 최적의 모델 Linear SVR(C = 1, epsilon = 50)과 이외의 다른 Hyperparameter 값들을 테스트 데이터에 적용하여 최종 성능을 평가했다. 아래 <표 4>는 테스트 데이터에 대한 각 Hyperparameter로 평가한 결과이고, <그림 8>은 이를 시각화 한 것이다.

C	Epsilon	MSE_Test
0.0001	50	2509.7992
0.0001	100	2550.6957
0.0001	200	2818.2882
0.001	50	2465.5778
0.001	100	2537.4341
0.001	200	2819.2685
0.01	50	2172.4279
0.01	100	2404.2219
0.01	200	2830.3403
0.1	50	1824.2555
0.1	100	2237.5199
0.1	200	2952.3126
1	50	1522.5406
1	100	2405.3412
1	200	3306.4188

<표 4. 테스트 데이터 결과>



<그림 8. 테스트 데이터 결과>

2-5) 검증 데이터와 테스트 데이터의 성능을 비교한 결과, 두 데이터에서 최적의 Hyperparameter 조합은 C = 1.0, ∈ = 50으로 동일하게 나타났다. 검증 데이터에서의 MSE 값은 1944.5111이고, 테스트 데이터에서는 1522.5406으로 두 값은 유사한 수준이지만 테스트 데이터에서 성능이 약간 더 우수하게 나타났다. 이 차이는 데이터의 분포 차이에 의한 것일 것이라고 예상된다. 결과적으로 두 데이터의 성능은 유사한 경향을 보이며, 모델의 일반화 성능은 적절하다.