예측애널리틱스 Homework #5

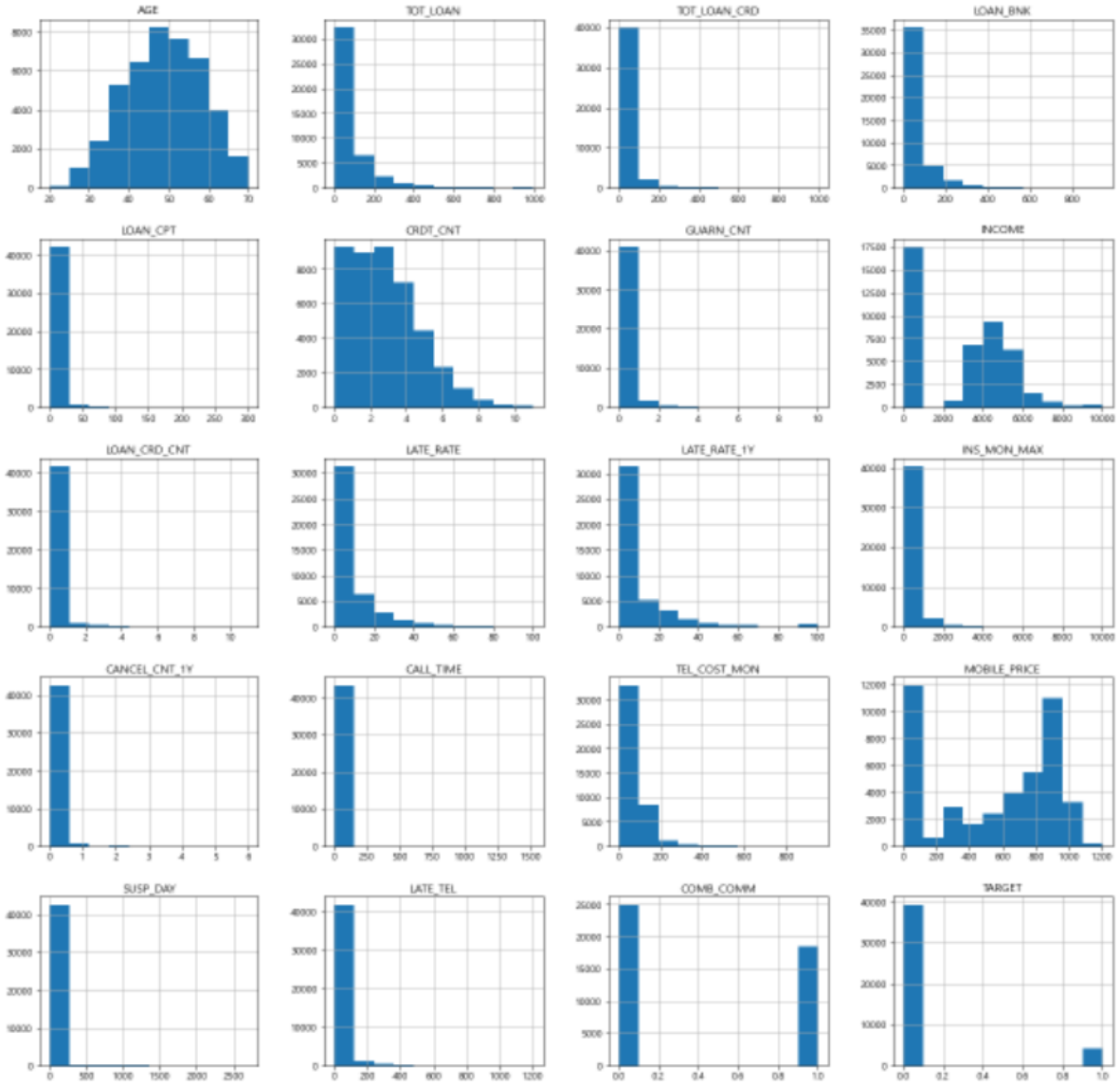
고려대학교
KOREA UNIVERSITY

대학	고려대학교 공과대학
학과	산업경영공학부
학번	2017170819
이름	박상민

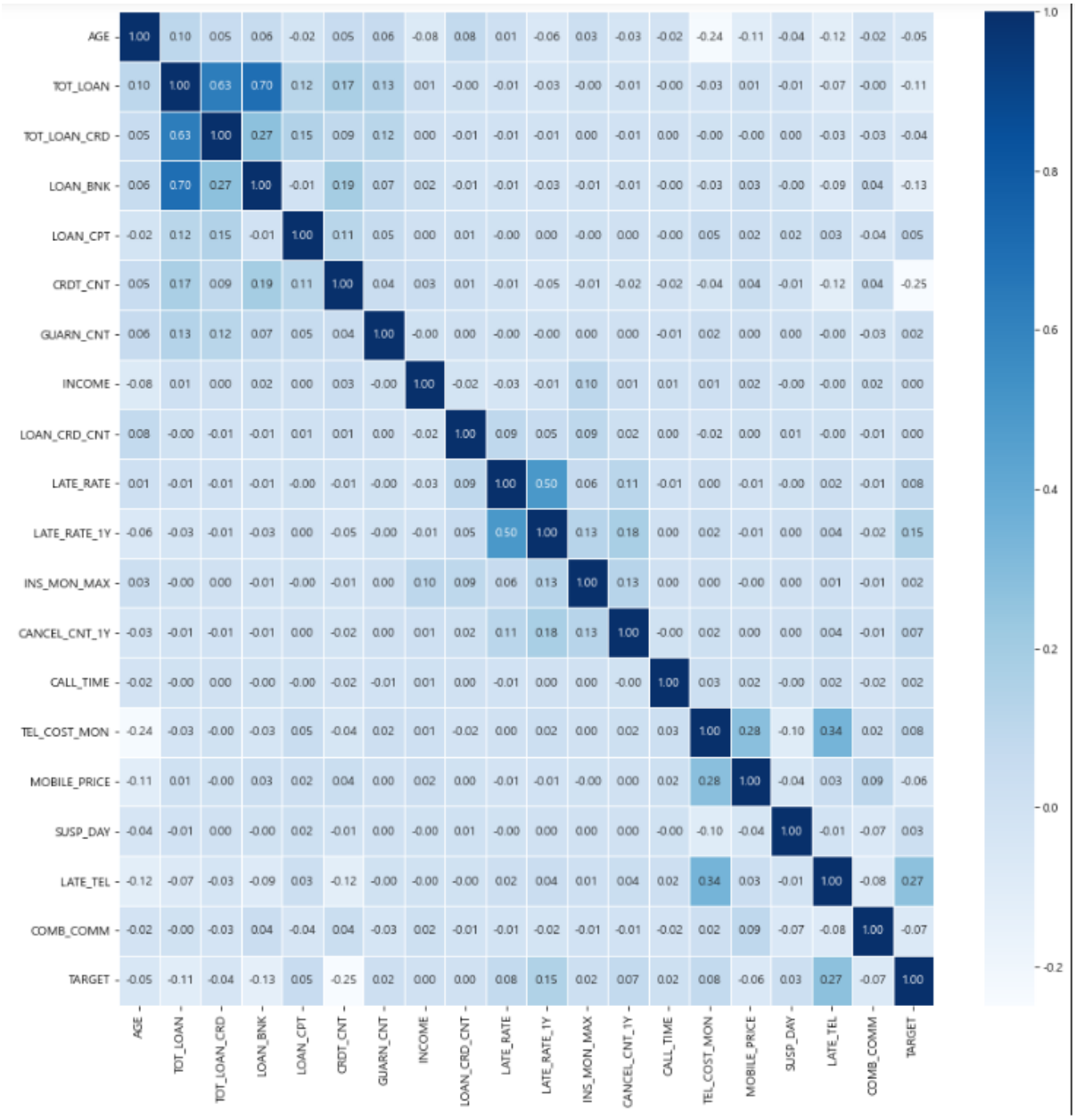
1. 첨부한 loan 데이터를 이용하여 (a) linearly separable SVM, (b) linearly nonseparable SVM, (c) nonlinear SVM 모델을 구축하고 각각 training error 와 testing error 를 비교하시오.

먼저 데이터를 탐색해보았다. 분석할 데이터는 Loan data 로, 변수들에 대한 정보는 다음과 같다.

변수 이름	변수 설명	변수 형태
AGE	연령	연속형
TOT_LOAN	대출 총액	연속형
TOT_LOAN_CRD	신용대출 총액	연속형
LOAN_BNK	은행권에서 발생한 대출 총액	연속형
LOAN_CPT	카드사/캐피탈에서 발생한 대출 총액	연속형
CRDT_CNT	신용카드 발급 수	연속형
GUARN_CNT	보증 건수	연속형
INCOME	소득	연속형
LOAN_CRD_CNT	신용대출 건수	연속형
LATE_RATE	보험료 연체율	연속형
LATE_RATE_1Y	최근 1년 보험료 연체율	연속형
INS_MON_MAX	월납입보험료 (최대값)	연속형
CANCEL_CNT_1Y	최근1년 실효해지건수	연속형
CALL_TIME	월별 통화시간	연속형
TEL_COST_MON	서비스 납부요금	연속형
MOBILE_PRICE	사용중인 핸드폰단말기 가격	연속형
SUSP_DAY	회선의 사용정지일수	연속형
LATE_TEL	핸드폰 요금 연체금액	연속형
COMB_COMM	결합상품가입 여부	이진형
SEX	성별: 남자(M), 여자(F)	명목형
PAY_METHOD	핸드폰 요금 납부방법	명목형
JOB	직업군	명목형
TARGET	대출연체여부: 미발생(0), 발생(1)	이진 (타겟변수)



변수들의 분포를 알아보기 위해 히스토그램을 그려보았다. 일부 연속형 데이터에서 skewed data 분포를 보임을 알 수 있다.



상관관계 matrix 를 그려보면 TOT_LOAN 과 LOAN_BANK 가 강한 상관관계를 가짐을 알 수 있다.

	AGE	TOT_LOAN	TOT_LOAN_CRD	LOAN_BNK	LOAN_CPT	CRDT_CNT	GUARN_CNT	INCOME	LOAN_CRD_CNT	LATE_RATE	L
count	43386.000000	43386.000000	43386.000000	43386.000000	43386.000000	43386.000000	43386.000000	43386.000000	43386.000000	43386.000000	
mean	46.250980	82.019407	32.829622	49.324897	4.288641	3.004264	0.098695	2778.629051	0.163855	8.216406	
std	9.693741	126.702976	83.419760	92.443944	12.660968	1.842478	0.529664	2470.097227	0.617522	12.120840	
min	20.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	40.000000	12.000000	0.000000	0.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000	
50%	45.000000	36.000000	9.000000	9.000000	0.000000	3.000000	0.000000	3600.000000	0.000000	3.000000	
75%	55.000000	102.000000	27.000000	60.000000	3.000000	4.000000	0.000000	4700.000000	0.000000	11.000000	
max	70.000000	994.000000	994.000000	944.000000	301.000000	11.000000	10.000000	10000.000000	11.000000	100.000000	

기초통계량 분석을 진행했다. SVM 은 거리 기반 데이터분석 기법이고, X 변수들 각각 단위가 다르기 때문에 스케일링을 통해 단위를 통일시켜줄 필요가 있다.

(1) Linearly separable SVM 은 선형 SVM 이지만 오차를 허용하지 않는 Hard margin SVM 이다.

Margin 은 각 클래스에서 가장 가까운 관측치 사이의 거리를 나타내며, margin 을 최대화하는 것이 목적식이다. 이 때 hyperplane 은 $w^T x + b = 0$ 로 표현한다.

$$\underset{w,b}{minimize}\ \frac{1}{2}\|w\|_2^2$$
$$subject\ to\ y_i(w^T x_i + b) \geq 1, i = 1, 2, \dots\dots, n$$

라그랑주 승수법을 통해 목적식과 제약식을 하나의 식으로 표현했다.

$$\max_{\alpha}\min_{w,b}\mathcal{L}(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{n}\alpha_i(y_i(w^T x_i + b) - 1)$$
$$subject\ to\ \ \alpha_i \geq 0, i = 1,2, \dots\dots, n$$

Maxmin 문제로 바뀐다. 이 목적식은 convex 함수이기 때문에 미분을 통해 파라미터를 추정할 수 있다.

$$\underset{\alpha}{maximize}\ \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$$

quadratic, Inner product

$$subject\ to\ \sum_{i=1}^{n}\alpha_i y_i = 0,$$

linear

$$\alpha_i \geq 0, i = 1,2, \dots\dots, n$$

따라서 이런 식으로 목적식을 표현할 수 있고, KKT 조건의 4 번째 조건인 complementary slackness 에 의해

$$w^* = \sum_{i \in SV}\alpha_i^* y_i x_i$$

로 w 를 구할 수 있다.

$$\underset{w,b}{minimize}\ \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i$$
$$subject\ to\ y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1,2, \dots\dots, n$$

soft margin SVM 은 완벽히 선형으로 분리할 수 없을 때, error 를 일부 허용해서 모델을 구축하는 방식이다. C 는 하이퍼파라미터이며, C 를 크게 설정하면 Error 를 거의 허용하지 않는다는 뜻이고, C 를 작게 설정하면 error 를 많이 허용한다는 의미이다. 따라서 hard margin SVM 은 C 가 매우 큰 경우이고, soft margin SVM 은 C 가 작은 경우이다.

Hard margin SVM 을 구현하기 위해 C=1000 으로 설정하고 모델을 구현했다. 먼저 데이터셋을 train set 과 test set 으로 분리하고, 스케일링을 한 후 모델을 구현했다.

```
(30370, 26)
(13016, 26)
```

Training set 에는 30370 개의 관측치가 할당됐고, Testing set 에는 13016 개의 관측치가 할당됐다. Unbalanced data 이기 때문에 stratify 조건을 통해 y class 가 training set 과 testing set 에 균등한 비율로 포함되도록 했다. 이후 hyperplane 을 도출해 SVM 모델을 구축했다.

```
[-0.1487437  -0.10069907 -0.10023554 -0.22137082  0.11776318 -0.35709655
 -0.18399666 -0.02534254  0.02071259 -0.01956855  0.06050311 -0.01145468
  0.05891653 -0.04803647  0.12598106 -0.19546308 -0.04887342 -0.05105826
 -0.24406999  0.03293287  0.20413289  0.05572275 -0.29133392 -0.06977767
  0.26038895 -0.00325427]
```

hyperplane 의 기울기인 w 벡터는 다음과 같다.

hyperplane 의 y 절편이라고 할 수 있는 b 도 구할 수 있다.

```
-1.050060218439756
```

```
Train set에 대한 성능
정확도:0.8871

Test set에 대한 성능
정확도:0.8871
```

Training accuracy, testing accuracy 를 구해보았다. 둘 다 0.8871 로 동일했고, 꽤 높은 정확도를 보였다.

```
정확도:0.8871 | 민감도:0.3927 | 정밀도:0.4048
```

	예측 Y=1	예측 Y=0
실제 Y=1	True Positive	False Negative
실제 Y=0	False Positive	True Negative

$$\text{정확도} = \frac{TP+TN}{TP+FN+FP+TN}$$

$$\text{민감도} = \frac{TP}{TP+FN}$$

$$\text{정밀도} = \frac{TP}{TP+FP}$$

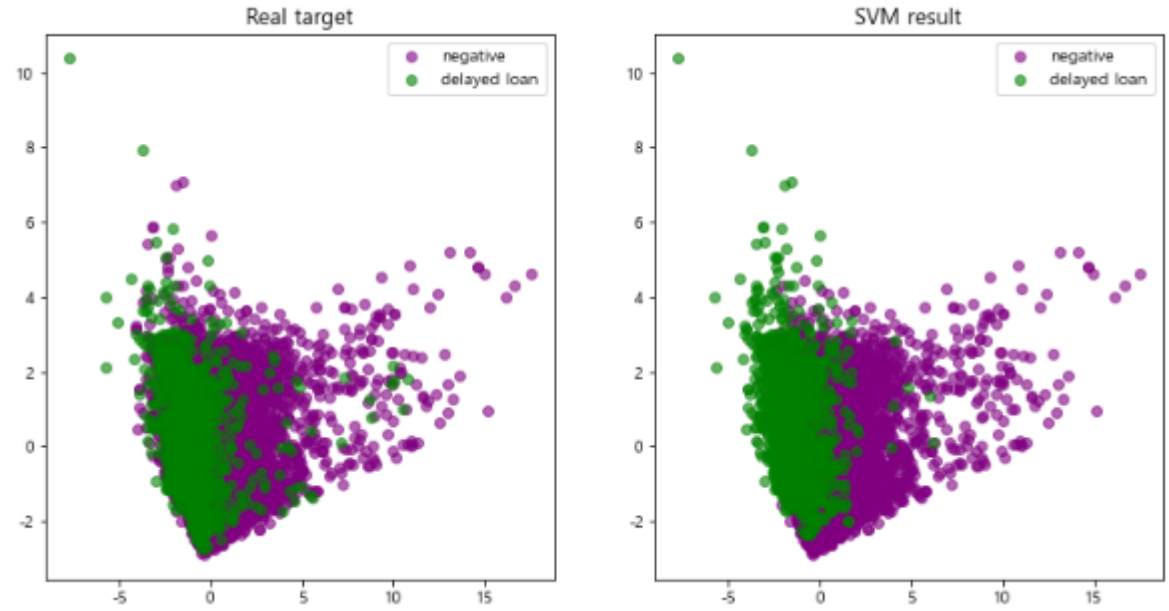
True Positive(TP) : 실제 Y=1이고 Y=1로 예측한 경우

True Negative(TN) : 실제 Y=0이고 Y=0으로 예측한 경우

False Negative(FN) : 실제 Y=1이고 Y=0으로 예측한 경우

Fasle Positive(FP) : 실제 Y=0 이고 Y=1 로 예측한 경우

Confusion matrix 를 통해 정확도, 민감도, 정밀도를 구했다. 정확도는 전체 경우 중 정확히 분류한 확률이다. 0.8871 로 꽤 높은 정확도를 보였다. 하지만, 실제 loan 해준 경우 대비 loan 해주었다고 올바르게 예측한 경우의 비율이 민감도인데, 0.3927 로 높지는 않았다. 이는 Y=1 인 데이터수가 애초에 많지 않기 때문이라고 예상했다. 정밀도는 loan 해주었다고 예측한 경우 대비 실제 loan 해준 경우의 비율이다. 0.4048 로 그리 높지는 않기 때문에, 민감도와 마찬가지의 이유라고 예상했다.



가시화해서 분석해보면, 실제 target 이 분류된 양상과 비슷하게 SVM 이 분류했음을 알 수 있다.

(2) linearly nonseparable SVM

다음으로, soft margin SVM 을 구현해보았다.

```
[-7.76703924e-14  1.14020425e-13 -4.43588128e-14 -1.31414242e-13
 -2.42344999e-15 -1.96856169e-14  6.48623031e-16 -2.93857359e-14
  7.13441192e-15  1.57589814e-14  9.80183241e-15  1.53501147e-14
  3.86880008e-14  1.01295649e-13 -7.18042870e-15 -3.95569788e-14
  3.24426977e-14  8.22472890e-14  3.12325203e-14 -4.88078128e-14
  2.66968039e-14  2.07842871e-13  6.51917740e-14 -2.29266099e-14
  3.07526752e-14  8.58846793e-14]
```

마찬가지로 w 벡터와 b 값을 구할 수 있다.

-1.0000000000002198

```
Train set에 대한 성능
정확도:0.9048

Test set에 대한 성능
정확도:0.9047
```

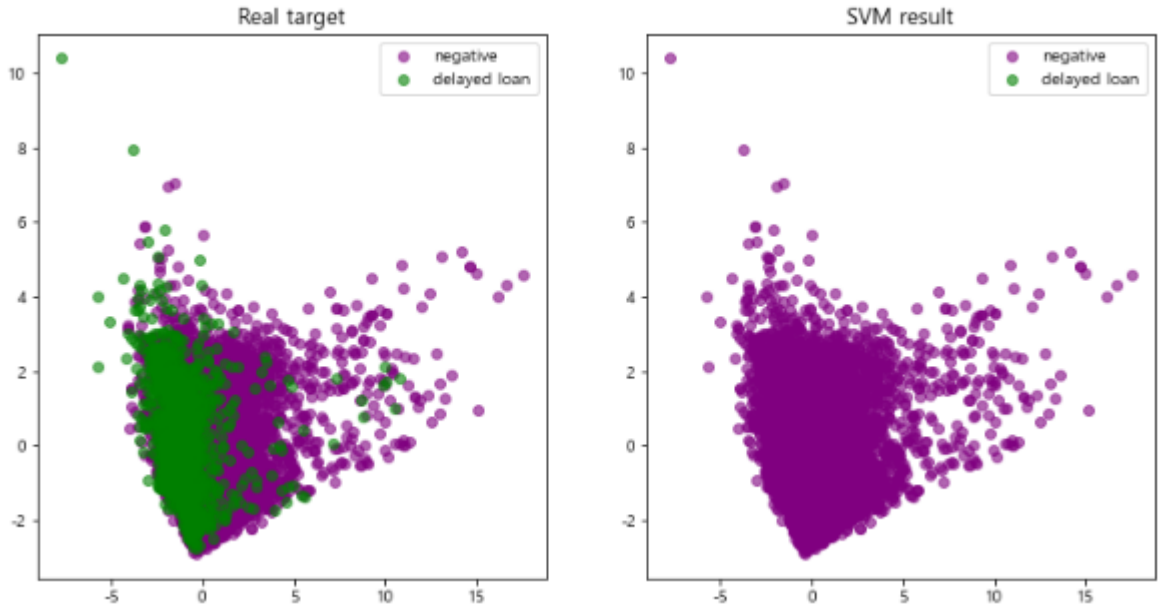
Training accuracy 와 testing accuracy 도 구해보았다. 각각 0.9048, 0.9047 로 더 높은 값을 가지는 것을 알 수 있다. 따라서 오히려 error 를 허용한 것이 overfitting 을 방지해 정확도를 높였다고 할 수 있다.

```
정확도:0.9047 | 민감도:0.0000 | 정밀도:0.0000
```

그러나 민감도와 정밀도는 0 을 기록했다. 그 이유는, 실제 true positive 로 분류한 경우가 거의 없었기 때문이다. 실제로 이 데이터는 90%가 class 0 이고, 10%가 class 1 이기 때문에 class1 로 전혀 분류하지 않아도 약 90% 정도의 정확도를 가질 수 있다는 것이다. 따라서 이 모델은 좋은 모델이 아니다.

```
data['TARGET'].value_counts()
```

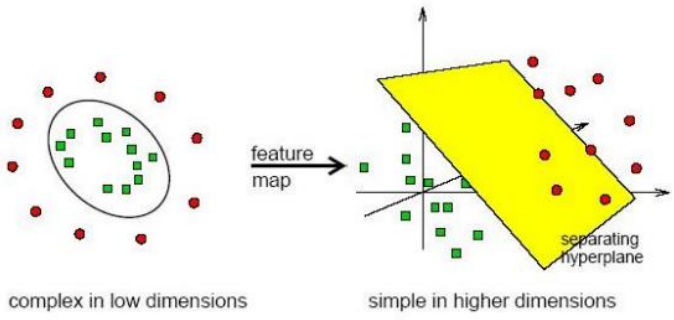
```
0    39254
1     4132
Name: TARGET, dtype: int64
```



시각적으로 가시화해보면, 실제 delayed loan 된 target 값들을 전혀 잡지 못하고, negative 로 분류했음을 알 수 있다.

(3) nonlinear SVM

마지막으로, nonlinear SVM 을 구축해보았다. Nonlinear SVM 은 original space 가 아닌 feature space 에서 학습하는 것이다. Kernel function 을 이용해 inner product 에 해당하는 값만 정의할 수 있으면 Kernel 함수를 이용해 같은 효과를 얻을 수 있다.



Kernel function 은 polynomial kernel 을 사용했다. 먼저 degree=3, C=100 으로 설정하고 구현해보았다.

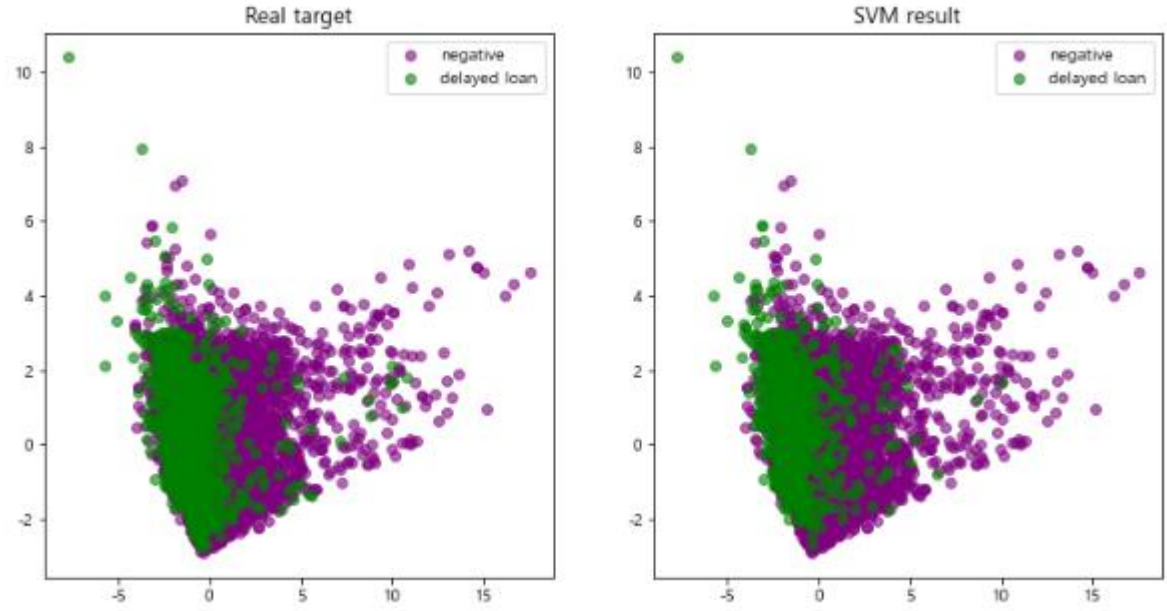
```
Train set에 대한 성능
정확도:0.9467

Test set에 대한 성능
정확도:0.9076
```

Training accuracy 와 testing accuracy 도 구해보았다. 각각 0.9467, 0.9076 이기 때문에 꽤 높은 예측정확도를 보인다.

```
정확도:0.9076 | 민감도:0.3226 | 정밀도:0.5242
```

그러나 마찬가지로 민감도, 정밀도는 각각 0.3226, 0.5242 로 높은 편이 아니다. 이는 Y=1 인 데이터수가 애초에 많지 않기 때문에 class 1 으로 정확히 분류된 경우가 적기 때문이다. 그래도 비선형 SVM 이기 때문에 비선형 hyperplane 을 가지고 있어 정확한 분류 기능을 보여주었다고 할 수 있다.



시각적으로 가시화해보면, 실제 target 값들을 SVM 이 잘 분류했다고 할 수 있다.

하이퍼파라미터를 바꿔서 degree=3, C=3 으로 정하고 nonlinear SVM 을 구축해보았다.

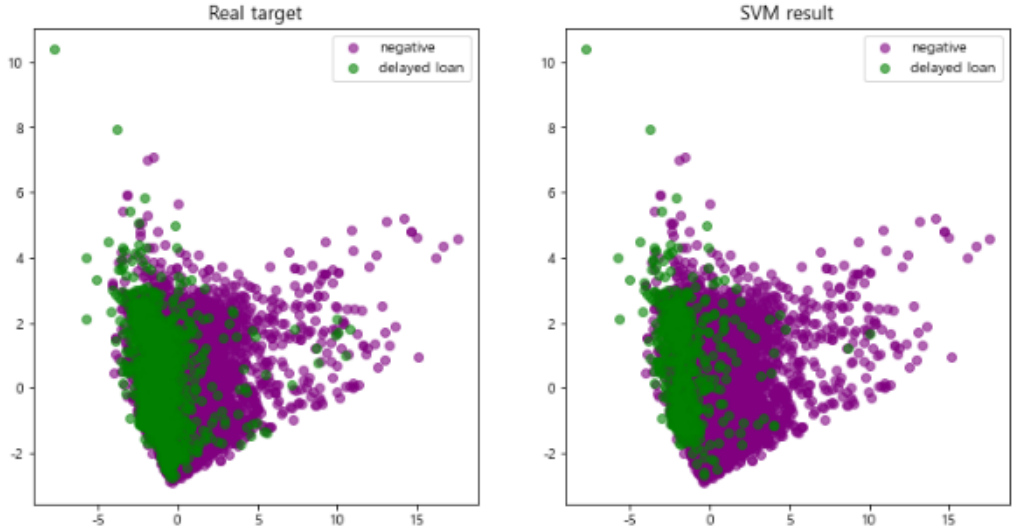
```
Train set에 대한 성능
정확도:0.9325

Test set에 대한 성능
정확도:0.9129
```

Training accuracy 와 testing accuracy 도 구해보았다. 각각 0.9325, 0.9129 이기 때문에 꽤 높은 예측정확도를 보인다.

```
정확도:0.9129 | 민감도:0.2613 | 정밀도:0.5978
```

민감도와 정밀도는 각각 0.2613, 0.5978 을 나타냈다. 민감도와 정밀도는 trade off 관계가 있기 때문에 민감도와 정밀도를 동시에 어느 수준까지 증가시키는 것은 한계가 있음을 알 수 있다.



시각적으로 가시화해보면 실제 target 값들을 SVM 이 잘 분류했다고 할 수 있다.

정리하면, Linearly separable SVM 은 정확도, 민감도, 정밀도 면에서 모두 괜찮은 성능을 보여줬다. Linearly nonseparable SVM 은 정확도는 높았지만, 데이터 자체가 심하게 unbalanced data 이기 때문에 delayed loan 을 정확히 분류하지 못하면서 민감도, 정밀도가 너무 낮게 나왔다. Nonlinear SVM 은 비선형 hyperplane 통해 분류할 수 있기 때문에 linear SVM 보다 더 유연하고, 분류 정확도도 더 높으며, 민감도와 정밀도도 높기 때문에 delayed loan 을 비교적 정확히 분류했음을 알 수 있다. 종합적으로 평가했을 때, C=100, degree=3 인 nonlinear SVM 의 예측정확도가 가장 좋다고 결론지었다.

2. KKT conditions 에 대해 조사하고 요약하시오.

KKT condition : 등식의 제약식이 있는 최적화 문제를 푸는 것을 라그랑주 승수법이라 하고, 부등식의 제한 조건에서도 쓸 수 있게 확장시킨 것을 KKT condition 이라 한다.

$$p^\star = \min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{subject to : } g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, k$$

등식과 부등식 제약조건이 있는 일반적인 최적화 문제에 대해서 KKT 조건(Karush-Kuhn-Tucker conditions)은 다음과 같다.

$$\text{1. stationarity: } \nabla_{\mathbf{x}} f(\mathbf{x}) + \sum_{i=1}^{m} \mu_i \nabla_{\mathbf{x}} g_i(\mathbf{x}) + \sum_{j=1}^{k} \lambda_j \nabla_{\mathbf{x}} h_j(\mathbf{x}) = 0$$

$$\text{2. primal constraints: } g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, k$$

$$\text{3. dual constraints: } \mu_i \geq 0, \quad i = 1, \ldots, m$$

$$\text{4. complementary slackness: } \mu_i g_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m$$

여기서 1 번과 2 번항인 stationarity 와 primal constraints 조건은 등식 제약조건만 있는 최적화문제의 경우와 동일한 것으로서, 최적화 변수 x 와 라그랑주 승수 μ, λ 에 대해서 각각 계산한 라그랑지안,

$$L(\mathbf{x}, \mu, \lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} \mu_i g_i(\mathbf{x}) + \sum_{j=1}^{k} \lambda_j h_j(\mathbf{x})$$

의 그래디언트가 0 이 되는 정류점(stationary point)이 최적해가 됨을 나타낸다.

3 번과 4 번항은 부등식 제약조건에서 나타나는 추가적인 조건이다.
만약 최적해 x⋆가 gi(x⋆)<0 의 영역에 있다면 이 때의 부등식 제약조건을 슬랙 제약(slack constraints)이라고 하는데 이 경우는 최적해를 계산하는데 아무런 영향을 주지 않는다. 따라서 이에 대한 라그랑주 승수는 μi=0 으로 주어진다. 만약 최적해 x⋆가 부등식 제약조건의 경계선인 gi(x⋆)=0 의 영역에 있다면 이 때의 제약조건을 바운딩 제약(bounding constraints)이라고 하는데 이 경우는 등식 제약조건과 다를 바 없게 된다.

이 두 경우를 합치면 4 번항 complementary slackness(상호 보완적인 슬랙) 조건이 성립한다.

3. SVM 와 logistic regression model 과의 차이점을 명시하시오.

SVM 과 logistic regression model 은 둘 다 지도학습에 해당하지만, 몇가지 차이점이 존재한다.

1. 다른 목적 함수 : 로지스틱회귀는 Log Likelihood functioin 을 최소화하는 방향으로 파라미터를 추정하는 목적식을 가진다. SVM 은 초평면으로부터 기하학적 간격, 즉 각 클래스에서 가장 가까운 관측치 사이의 거리를 뜻하는 margin 을 극대화하는 방향으로 목적식을 가진다. 라그랑주 승수법을 통해, SVM 목적 함수를 작성할 수 있다.

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(w^T x_i + b) - 1 \right)$$

2. 로지스틱회귀는 목적식이 convex 하다고 알려져 있지 않기 때문에 수치최적화 알고리즘을 통해 파라미터를 추정한다. SVM 은 목적식과 제약식이 quadratic programming 형태이기 때문에, 목적식이 convex 함수이며 전역 최적해가 존재한다. 따라서 미분을 통해 파라미터를 도출할 수 있다.

3. SMV 은 로지스틱 회귀와는 어디에 강조를 두느냐가 다르다. 비용함수에 방점을 두는가, 정규화에 방점을 두는가가 서로 다른 것이다. SVM 의 기본 성질이 Large Margin, 즉 학습 데이터에 **최적화 fit 되지 않게끔** 하여 실제 데이터의 분류 정확도를 향상시키는 것이므로 어쩌면 당연한 말이다. C 와 λ 는 개념적으로 반비례 관계이다.

로지스틱 회귀: **A + λB**

SVM: **CA + B**

로지스틱 회귀 문제에서는 **A + λB** 를 작게 만드는 것이 목표였다. **람다가 커지면 정규화에 치중**되어 람다 B 는 0 에 가까워진다. 람다가 커지면 B 안에 들어있는 파라미터들이 작아져서 고차항들이 많이 사라진다. 차수가 올라갈수록 그래프 모양이 구불구불해졌던 것이니 차수가 내려가면 그래프가 smoothing 된다. 람다에 무게를 너무 많이 주면 underfit 이 될 수 있는 이유다. 즉, **λB** 가 0 에 가까워지게 만드는 최소화 문제로, 정규화항은 없어지는 것이나 마찬가지다. 이 과정에서는 A 항만 남게 된다. SVM 문제에서는 **CA + B** 를 작게 만드는 것이 목표로, **C 가 커지면 피팅에 치중**(overfitting 가능성 높음)되어 CA 가 0 에 가까워지고, CA 가 0 에 가까워지게 하는 최소화 문제가 되어 B 항만 남게 된다. 즉 우선 파라미터를 줄이는 것을 강조하는 것이다.

4. 고려대학교 인촌기념관, 대학원, 사범대, 본관을 방문하여 사진을 찍고 첨부하세요 (독사진이 아니어도 좋습니다. 여러명이 함께 찍었을 경우 본인을 표시하세요.)

<인촌기념관>


(맨 왼쪽)

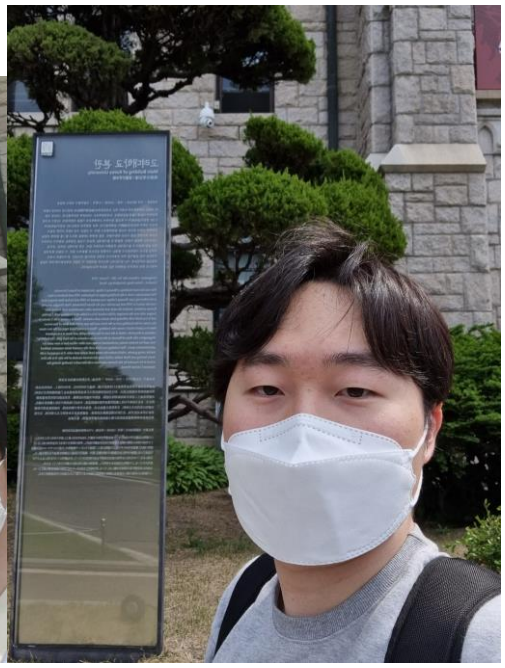
<대학원>


(맨 왼쪽)

<사범대>

(맨 왼쪽)

<본관>



(가운데)