LOCATING THE NEXT STARBUCKS IN SEOUL

1팀 범주형 자료분석

고은영 계승환

전연호 오희준

김민구 김주영

TABLE OF CONTENTS

1. 방향설정

2. 데이터
정제 과정

3. 로지스틱
회귀모형

4. SVM

5. GAM

6. 최종모형
선택 및 결론

1. 2주차 방향설정

1주차 에서는...

- 서울을 각 **행정동**으로 나누어

  동 별로 스타벅스 입점에 영향을 미치는 요인 분석

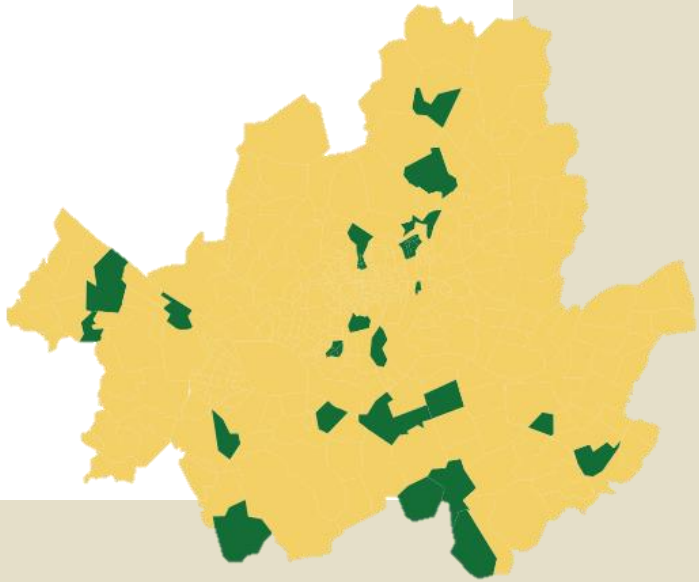
- 사용했던 분석

  : 패턴분석, 선형회귀모형, 포아송 회귀모형, Regression Tree

  Random Forest

1주차 에서는...

Random Forest에서 중요하게 사용된 변수	입점 후보 총 31개 동
여성 직장인구 수 의류 가게 매출액 20대 유동인구 수 카페 점포 수 …	삼청동 신당동 가양 제 1동 … 시흥 제 1동 잠실 7동



2주차 에서는...

- 그러나 동 별 분석은 포괄적이므로

  행정동 분석만으로는 유의미한 결론을 내리기 어렵다고 판단

- 서울을 상권으로 다시 나누어 **상권 별 영향 요인** 분석!

2. 데이터 정제과정

데이터 형성

1. 좌표계 변환

	A	B	C	D
1	골목상권 번호	골목상권 명	엑스좌표 값	와이좌표 값
2	11947	당산로44길	191269	448084
3	11948	동일로242길	205010	463934
4	11949	동일로_A	204944	464247
5	11950	동일로_B	204819	464224
6	11953	도봉로_A	203803	463702
7	11954	동일로_C	204780	463969
8	11955	동일로_D	204846	464465
9	11956	동일로242가길	204967	464104
10	11957	도봉로181길_A	203849	465051
11	11958	도봉로181길_B	203927	464905
12	11959	한글비석로_A	206461	462238
13	11960	노원로	205864	462038
14	11961	해등로_A	203166	462123
15	11962	상계로	205751	461973
16	11963	한글비석로20길_A	206519	462257
17	11964	상계로23다길	206077	462136

TM좌표계

	I	J
	위도	경도
	37.53221	126.9012
	37.67505	127.0568
	37.67787	127.056
	37.67766	127.0546
	37.67296	127.0431
	37.67536	127.0542
	37.67983	127.0549
	37.67658	127.0563
	37.68512	127.0436
	37.6838	127.0445
	37.65976	127.0732
	37.65796	127.0665
	37.65874	127.0359
	37.65738	127.0652
	37.65993	127.0739
	37.65884	127.0689

경 · 위도

스타벅스의 위치가 **위도와 경도**로 표현 되어 있기 때문에 **골목상권의 위치**도 위도와 경도로 변환!

데이터 형성

2. 스타벅스가 위치한 상권 매칭

	A	B	C	D	E	F	G	H	I	J	K
1	상호명	지점명	행정동코드	행정동명	법정동코드	법정동명	상권코드	도로명주소	층정보	경도	위도
2	스타벅스	IFC몰점	1156054000	여의동	1156011000	여의도동		서울특별시 영등포구 국제금융로 10	1	126.9249	37.52517
3	스타벅스	W-MALL점	1154551000	가산동	1154510100	가산동		서울특별시 금천구 디지털로 188	1	126.8877	37.47732
4	스타벅스	가든파이브	1171064200	문정2동	1171010800	문정동		서울특별시 송파구 충민로 10	1	127.1194	37.47826
5	스타벅스	가락본동점	1171062000	가락본동	1171010700	가락동		서울특별시 송파구 송파대로30길 13	2	127.1188	37.49488
6	스타벅스	가락시장역점	1171062000	가락본동	1171010700	가락동		서울특별시 송파구 중대로 121	1	127.1215	37.49441
7	스타벅스	가로수길점	1168051000	신사동	1168010700	신사동		서울특별시 강남구 논현로175길 94	2	127.0216	37.52318
8	스타벅스	가산그레이트	1154551000	가산동	1154510100	가산동		서울특별시 금천구 디지털로9길 32		126.8873	37.47963
9	스타벅스	가산디지털단지점	1154551000	가산동	1154510100	가산동		서울특별시 금천구 가산디지털1로 168	1	126.8826	37.47995
10	스타벅스	가산브이타워	1154551000	가산동	1154510100	가산동		서울특별시 금천구 가산디지털1로 128		126.8837	37.47722
11	스타벅스	가양이마트점	1150060500	가양3동	1150010400	가양동		서울특별시 강서구 양천로 559	1	126.8618	37.55818
12	스타벅스	강남GT타워점	1165053100	서초4동	1165010800	서초동		서울특별시 서초구 서초대로 411	1	127.0259	37.4981
13	스타벅스	강남II점	1165053100	서초4동	1165010800	서초동		서울특별시 서초구 서초대로77길 27	1	127.0255	37.50011
14	스타벅스	강남구청역점	1168053100	논현2동	1168010800	논현동		서울특별시 강남구 선릉로 669	1	127.0412	37.51661
15	스타벅스	강남대로점	1168064000	역삼1동	1168010100	역삼동		서울특별시 강남구 강남대로 456	2	127.0256	37.50313
16	스타벅스	강남비젼타워점	1168064000	역삼1동	1168010100	역삼동		서울특별시 강남구 테헤란로2길 27	1	127.0297	37.49648
17	스타벅스	강남삼성타운점	1165052000	서초2동	1165010800	서초동		서울특별시 서초구 서초대로78길 24	2	127.0277	37.49554

데이터 형성

2. 스타벅스가 위치한 상권 매칭 - KNN algorithm

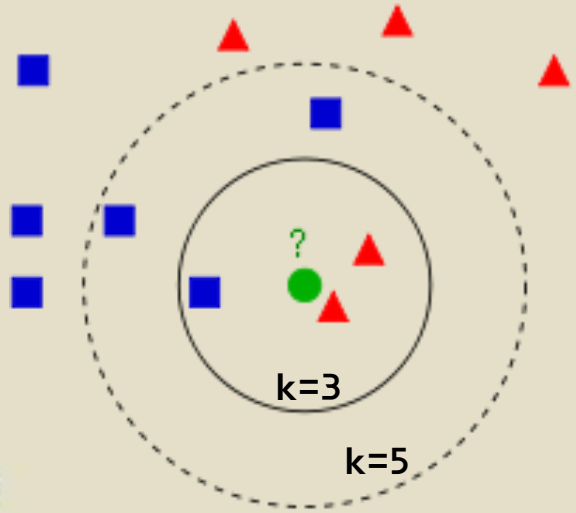
골목상권 명과 스타벅스의 주소를 비교해볼까?

남부순환로_A, 남부순환로_B, 남부순환로_C,
남부순환로_D, 남부순환로_E, 남부순환로_F, 남부순환로_G,
남부순환로_H … 남부순환로_W 등
같은 이름의 골목상권 명이 다수 존재

K-Nearest Neighbors(KNN) Method를 사용해보자!

데이터 형성

2. 스타벅스가 위치한 상권 매칭 - KNN algorithm



KNN Classifier란?

☞ Euclidean distance를 이용해서
가장 가까운 점 k개를 찾고 많이 선택된 그룹으로 분류

ex. 왼쪽 그림에서 k=3일 때는 red로 분류
k=5일 때는 blue로 분류

데이터 형성

2. 스타벅스가 위치한 상권 매칭 - KNN algorithm

스타벅스의 위·경도와 가장 가까운 골목 상권의 위·경도를 k=1인 KNN classifier로 찾아 보면

```
> knn
  [1] 12895 13387 13357 13222 13204 12906 13387 13370
 [14] 13044 13298 13298 13044 13044 12741 12048 12470
 [27] 12366 12366 13003 12239 13577 12320 12324 12320
 [40] 12287 12339 13163 13217 13217 13247 13363 13363
 [53] 12803 12049 12497 13476 13169 12755 12819 12772
 [66] 12021 12015 13043 13027 13102 13102 12752 11947
 [79] 12535 12517 12898 12779 13508 12912 12532 12895
 [92] 12462 12518 12518 12518 12518 12518 12658 12364
[105] 12106 12890 12903 13256 12455 12102 12079 13043
[118] 12518 13324 13324 13515 13268 13340 12299 12310
[131] 13025 13255 13598 13514 13048 13028 12314 13508
[144] 12339 12599 12572 13155 12493 12493 12493 12493
```

골목상권 코드로 표현

	A	B	C	D	E	F	G	H	I
1	상호명	지점명	행정동코드	행정동명	법정동코드	법정동명	상권코드	도로명주소	층정보
2	스타벅스	IFC몰점	1156054000	여의동	1156011000	여의도동	12895	서울특별시 영등포구 국	
3	스타벅스	W-MALL점	1154551000	가산동	1154510100	가산동	13387	서울특별시 금천구 디지	
4	스타벅스	가든파이브	1171064200	문정2동	1171010800	문정동	13357	서울특별시 송파구 충민	
5	스타벅스	가락본동점	1171062000	가락본동	1171010700	가락동	13222	서울특별시 송파구 송파	
6	스타벅스	가락시장역점	1171062000	가락본동	1171010700	가락동	13204	서울특별시 송파구 중대	
7	스타벅스	가로수길점	1168051000	신사동	1168010700	신사동	12906	서울특별시 강남구 논현	
8							13387	서울특별시 금천구 디지털로9길	
9	스타벅스	가산디지털단지점	1154551000	가산동	1154510100	가산동	13370	서울특별시 금천구 가산	
10	스타벅스	가산브이타워	1154551000	가산동	1154510100	가산동	13370	서울특별시 금천구 가산디지털1로	
11	스타벅스	가양이마트점	1150060500	가양3동	1150010400	가양동	12490	서울특별시 강서구 양천	
12	스타벅스	강남GT타워점	1165053100	서초4동	1165010800	서초동	13298	서울특별시 서초구 서초	
13	스타벅스	강남II점	1165053100	서초4동	1165010800	서초동	13044	서울특별시 서초구 서초	
14	스타벅스	강남구청역점	1168053100	논현2동	1168010800	논현동	12985	서울특별시 강남구 선릉	
15	스타벅스	강남대로점	1168064000	역삼1동	1168010100	역삼동	13044	서울특별시 강남구 강남	
16	스타벅스	강남비젼타워점	1168064000	역삼1동	1168010100	역삼동	13298	서울특별시 강남구 테헤	

데이터 형성

road.code	road.name	candidate	starbucks	mlive10	mlive20	mlive30	mlive40	mlive50	mlive60	flive10
11947	당산로44길	0	1	14	15	20	13	12	14	14
11948	동일로242길	0	0	158	121	121	137	143	156	150
11949	동일로_A	0	0	28	22	21	25	26	28	27
11950	동일로_B	0	0	17	14	13	15	16	17	16
11953	도봉로_A	0	0	135	132	147	162	185	272	125
11954	동일로_C	0	0	182	140	138	157	166	179	170
11956	동일로242	0	0	65	52	50	56	60	65	61
11957	도봉로181길	0	0	60	57	66	72	81	119	55
11958	도봉로181길	0	0	88	86	99	106	121	179	82
11962	상계로	0	0	27	26	20	26	28	27	26
11963	한글비석로	1	0	24	15	13	20	17	19	20
11964	상계로23다	0	0	56	50	41	52	57	55	52
11965	상계로1길	0	0	165	137	133	144	150	140	160
11966	해등로_B	0	0	190	154	176	175	175	213	173
11970	상계로27길	0	0	63	56	48	61	62	67	60
11971	덕릉로83길	1	0	206	130	113	176	152	160	180
11972	도봉로_B	0	0	151	139	129	153	168	187	142
11974	노원로34길	0	0	157	135	123	155	154	176	152
11975	한글비석로	0	0	171	142	133	168	167	190	164
11978	한글비석로	0	0	34	29	27	33	33	38	32

상권 별 스타벅스 수와 기타 정보들을 결합 ☞ 최종 데이터셋 완성

데이터 변환 및 분할

상권 별 스타벅스 입점 가능성에 관심

☞ 스타벅스의 유무를 고려하는 분류 모델을 사용하는 것이 적절

☞ 스타벅스 개수 정보를 바탕으로 binary data를 형성하자!

starbucks
2
0
2
1
0
1
1
0
0
0
0
0
0
2
4

starbucks
1
0
1
1
0
1
1
0
0
0
0
0
0
1
1

스타벅스 지점 유무에 따른 binary data 형성

데이터 변환 및 분할

전체 상관 data

- 비후보군 data
  - Training Data 7
  - Test Data 3
- 최종 예측을 위한 후보군 행정동 상권 (109곳) data

3. 로지스틱 회귀모형

로지스틱 회귀모형

왜 로지스틱 회귀 모형부터?

✓ 선형회귀에서 확장된 GLM 모형 중 하나

✓ 반응변수 Y가 이항변수일 때 사용 가능한
가장 쉽고 효율적인 **분류 모형 (classification model)**

☞ 우선 로지스틱 회귀 모형을 활용하여
스타벅스 입점 가능성이 높은 상권들을 분류해보자!

로지스틱 회귀모형

✓ Stepwise-method

- 기존 변수 43개 ☞ 변수 선택 필요
- AIC를 기준으로 변수 선택

```
null = glm(starbucks~1,data=train,family="binomial")
full = glm(starbucks~.-lat-lon,data=train,family="binomial")
step(null,scope=list(lower=null,upper=full),direction="both")
```

(위경도 data는 제외!)

- 최종 선택된 모형 (총 12개 변수 선택)

```
Call:  glm(formula = starbucks ~ drink.sales + cloth.num + theater +
    apt.area + fwork + move50 + market + bank + high + mmove +
    drink.num + drink.month, family = "binomial", data = train)
```

: 카페 음료점 수, 의류점 수, 극장 수, 아파트 평균 면적, 여성 직장 인구 수, 50대 유동 인구, 마켓, 은행, 고등학교, 남성 유동인구, 카페 수, 카페 평균 영업 개월 수

로지스틱 회귀모형

```
> lrtest(fit)
Likelihood ratio test

Model 1: starbucks ~ drink.sales + cloth.num + theater + apt.area + fwork +
    move50 + market + bank + high + mmove + drink.num + drink.month
Model 2: starbucks ~ 1
  #Df  LogLik  Df  Chisq Pr(>Chisq)
1  13 -278.49
2   1 -339.62 -12 122.26  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> dwtest(fit)

        Durbin-Watson test

data:  fit
DW = 1.8643, p-value = 0.04152
alternative hypothesis: true autocorrelation is greater than 0
```

✓ 가능도 비 검정
- < 0.001

- 해당 모형이 유의

✓ 독립성 검정
- 0.04152

- 독립성 가정을 어느정도 만족!

로지스틱 회귀모형

```
> summary(fit)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.109e+00  4.932e-01  -4.277 1.90e-05 ***
drink.sales  1.211e-08  5.341e-09   2.268  0.02331 *
cloth.num   -9.405e-02  3.521e-02  -2.671  0.00756 **
theater      8.379e-01  4.217e-01   1.987  0.04695 *
apt.area     1.324e-02  5.814e-03   2.277  0.02281 *
fwork        7.731e-04  3.266e-04   2.367  0.01794 *
move50      -4.148e-03  7.202e-04  -5.759 8.47e-09 ***
market       1.172e+00  4.144e-01   2.828  0.00468 **
bank        -4.511e-01  2.254e-01  -2.001  0.04535 *
high        -1.475e+01  4.820e+02  -0.031  0.97558
mmove        1.822e-03  2.987e-04   6.099 1.06e-09 ***
drink.num   -2.676e-01  1.291e-01  -2.073  0.03822 *
drink.month  3.560e-03  2.222e-03   1.602  0.10915
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 679.24  on 653  degrees of freedom
Residual deviance: 556.98  on 641  degrees of freedom
AIC: 582.98

Number of Fisher Scoring iterations: 13
```

+

카페음료점 총매출액 / 극장 수
아파트평균면적 / 여성직장인구 수
슈퍼마켓 수 / 남성유동인구 수

−

의류점 수 / 카페음료점 수
은행 수 / 50대 유동인구 수

로지스틱 회귀모형

```
> summary(fit)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.109e+00  4.932e-01  -4.277 1.90e-05 ***
drink.sales  1.211e-08  5.341e-09   2.268  0.02331 *
cloth.num   -9.405e-02  3.521e-02  -2.671  0.00756 **
theater      8.379e-01  4.217e-01   1.987  0.04695 *
apt.area     1.324e-02  5.814e-03   2.277  0.02281 *
fwork        7.731e-04  3.266e-04   2.367  0.01794 *
move50      -4.148e-03  7.202e-04  -5.759 8.47e-09 ***
market       1.172e+00  4.144e-01   2.828  0.00468 **
bank        -4.511e-01  2.254e-01  -2.001  0.04535 *
high        -1.475e+01  4.820e+02  -0.031  0.97558
mmove        1.822e-03  2.987e-04   6.099 1.06e-09 ***
drink.num   -2.676e-01  1.291e-01  -2.073  0.03822 *
drink.month  3.560e-03  2.222e-03   1.602  0.10915
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 679.24  on 653  degrees of freedom
Residual deviance: 556.98  on 641  degrees of freedom
AIC: 582.98

Number of Fisher Scoring iterations: 13
```

- ✓ 대부분 유의한 변수들 선택
- ✓ 그렇다면 해당 모형의 예측력은 어떨까?

예측력 검정

- 스타벅스가 존재하는 상권 비율 확인

```
> sum(data$starbucks)/nrow(data)
[1] 0.2096257
```

- 스타벅스가 없는 상권이 더 많으므로 unbalanced data

- 균형이 좋지 않으므로 cutoff line을 단순히 0.5로 설정하지 않고 데이터 비율에 맞춰 0.21 수준으로 설정하자!

- test data 예측 확률 결과 분포

실제 비율과 유사한 분포

☞ 0.2 아래의 예측 값 많음!

예측력 검정

- **Test MCR 확인**

  Cutoff line은 0.21로 설정!

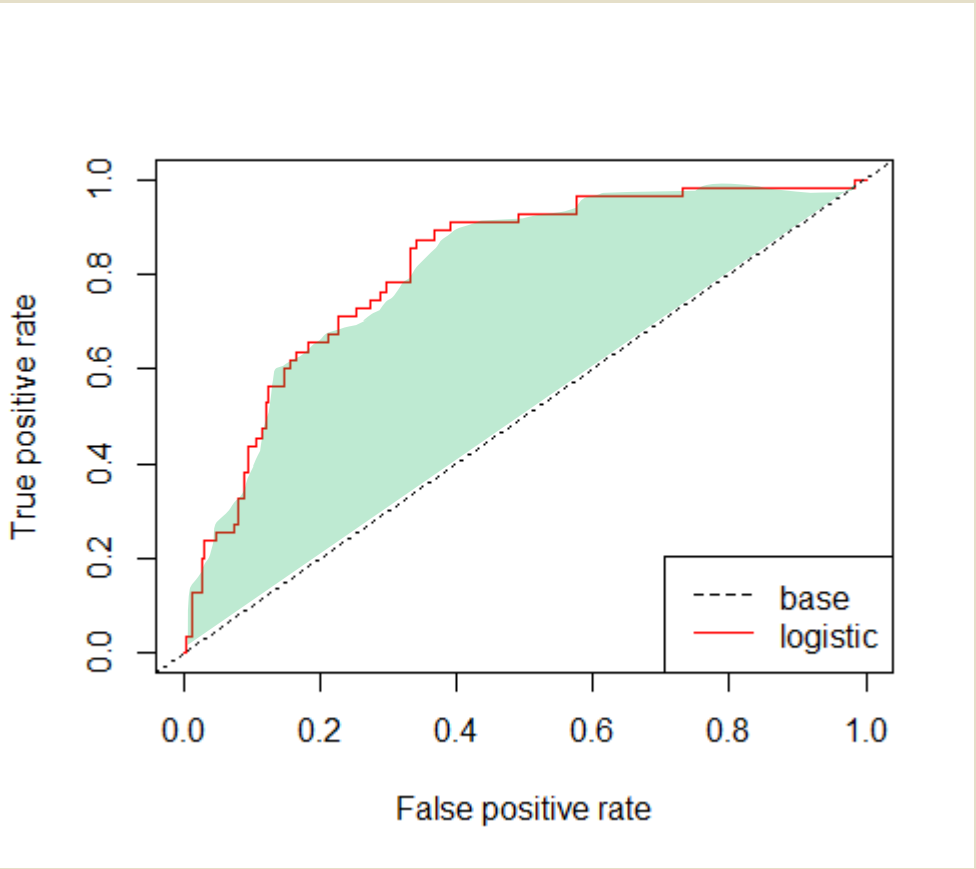
```
> sb.hat = ifelse(pred.y>=0.21,1,0)
> table(sb.hat,test$starbucks)

sb.hat   0   1
     0 157  12
     1  68  44
> sum(sb.hat==test$starbucks)/nrow(test)
[1] 0.7153025
> table(sb.hat,test$starbucks)[2,2]/sum(test$starbucks)
[1] 0.7857143
```

Test 예측	0	1
0	157	12
1	68	44

정분류율: 약 71.5%

민감도: 약 78.6%

예측력 검정



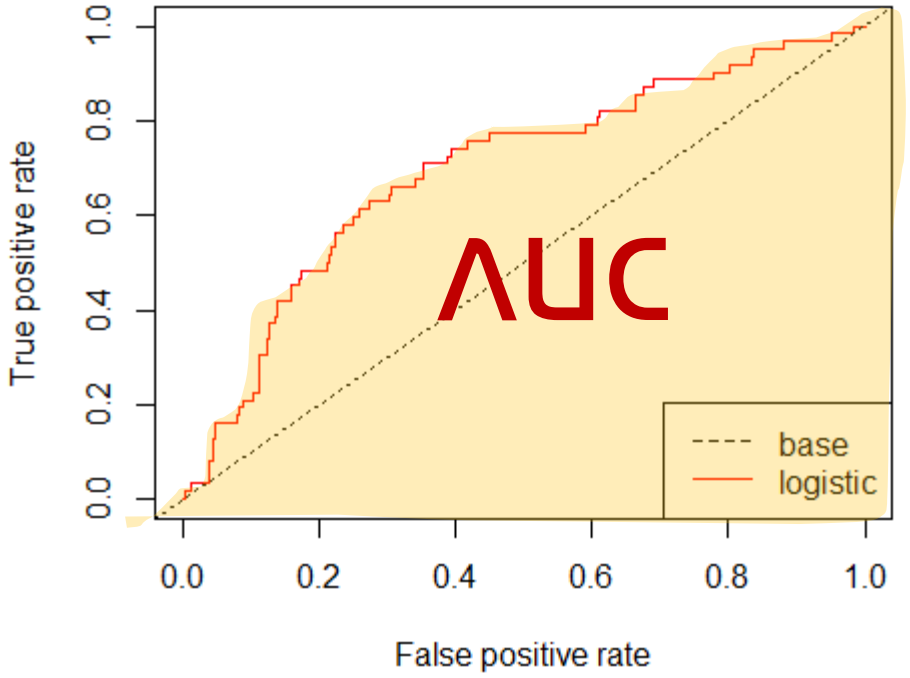
- ROC curve 사용 배경
  - 전체 예측력 뿐만 아니라, 스타벅스가 실제로 입점했을 때 입점할 것을 예측하는 **민감도**도 중요

☞ 관련 평가 지표 ROC, AUC 확인!

- ROC curve plotting 결과
  - 곡선이 'y=x' 축에서 상당히 떨어져 있음

☞ **예측력이 준수해 보인다!**

예측력 검정




```
> auc = performance(pred,measure='auc')
> auc = auc@y.values[[1]]
> auc
[1] 0.8193651
```

AUC : 0.8194

☞ 실제로 예측력이 준수하다

로지스틱 회귀모형

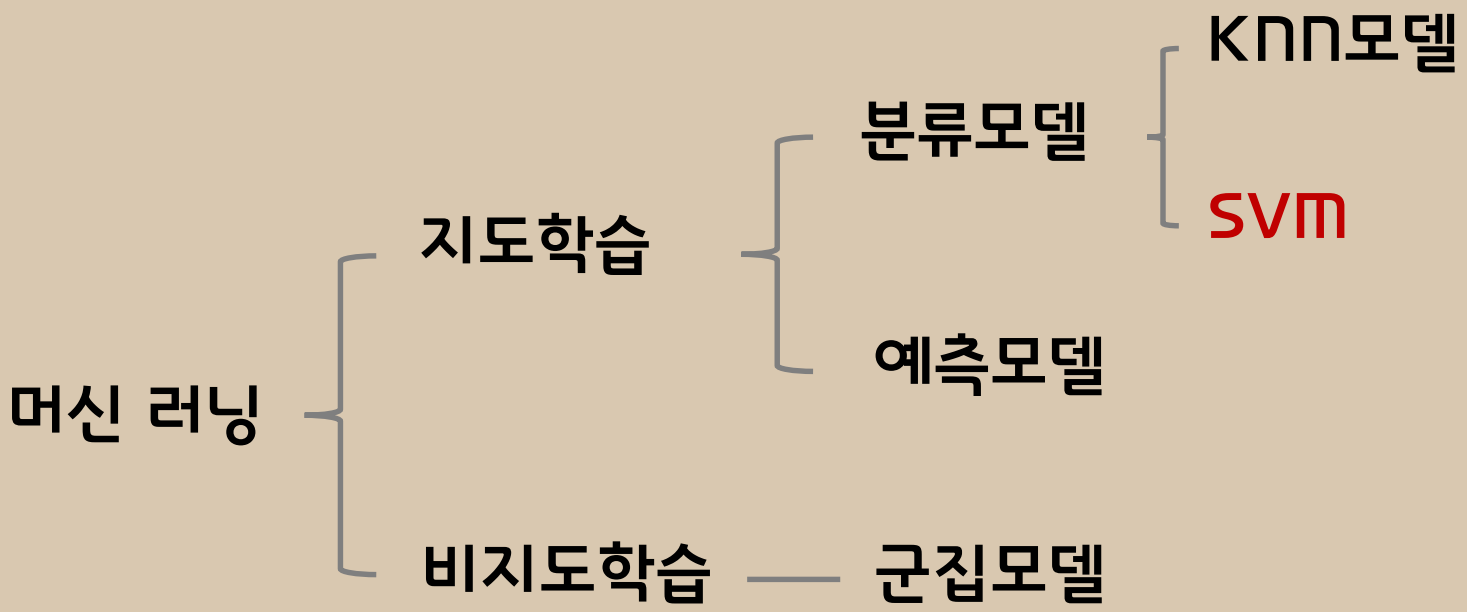
정분류율 71.5%, 민감도 78.6%로 준수한 예측력을 보임

최대한 정분류율과 민감도가 더 높아질 수 있도록

예측력이 좋은 다른 모형들도 고안해보았다!

4. 서포트벡터머신 (SVM)

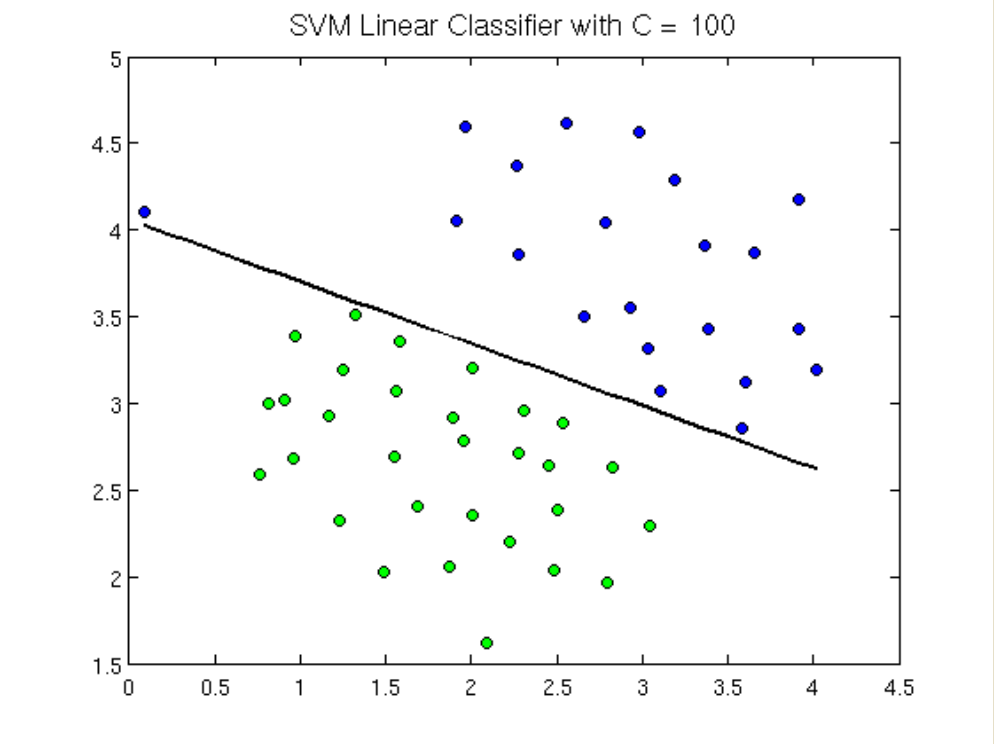
SVM이란?

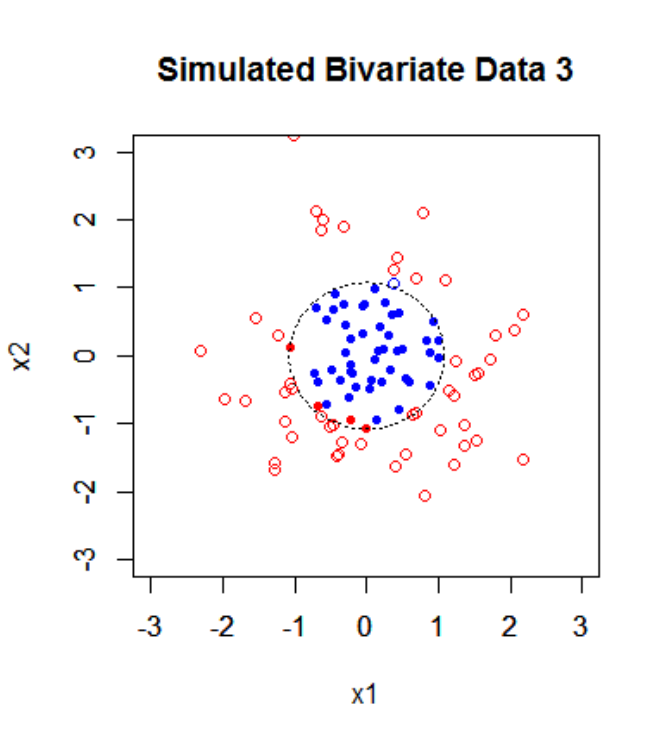


두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때,
새로운 데이터가 어느 카테고리에 속할지 분류하는 비확률적 이진 선형 분류 모델

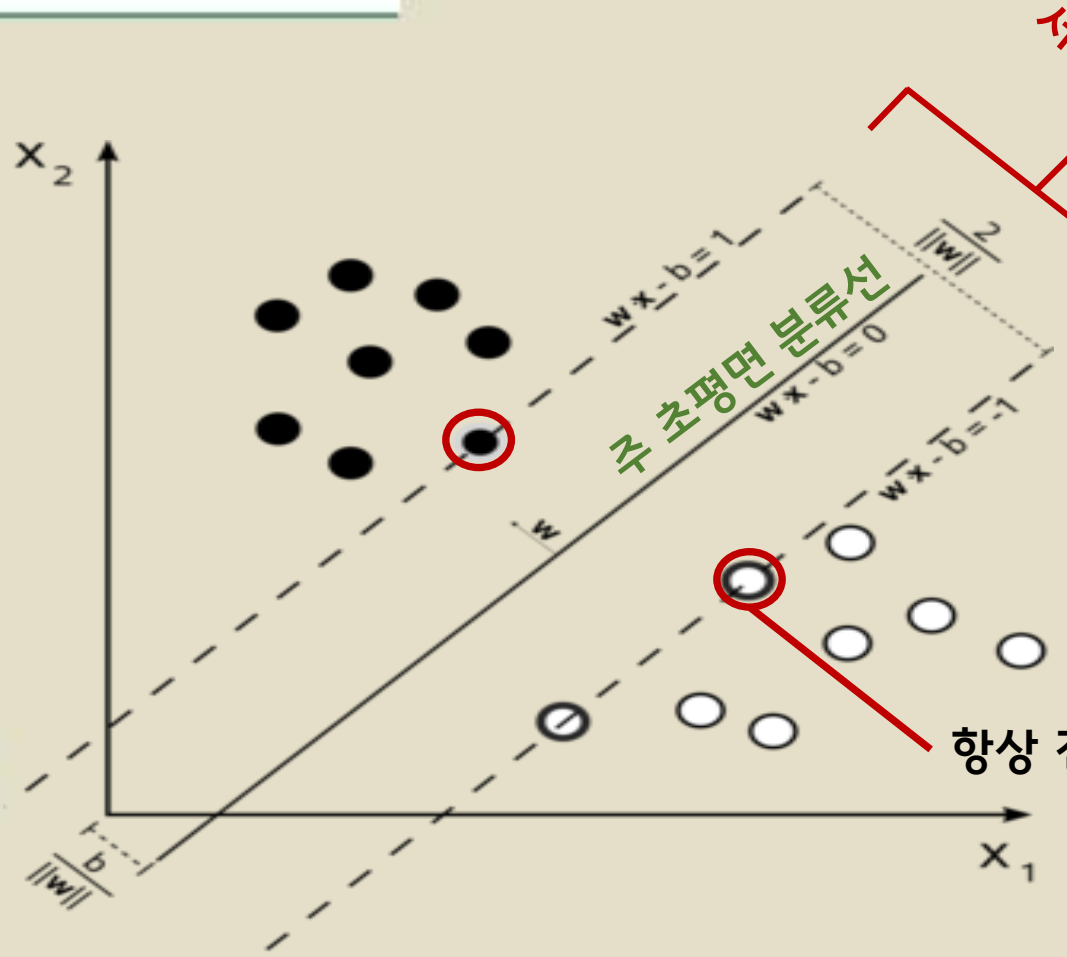
SVM이란?

◆ SVM의 예시





SVM의 원리

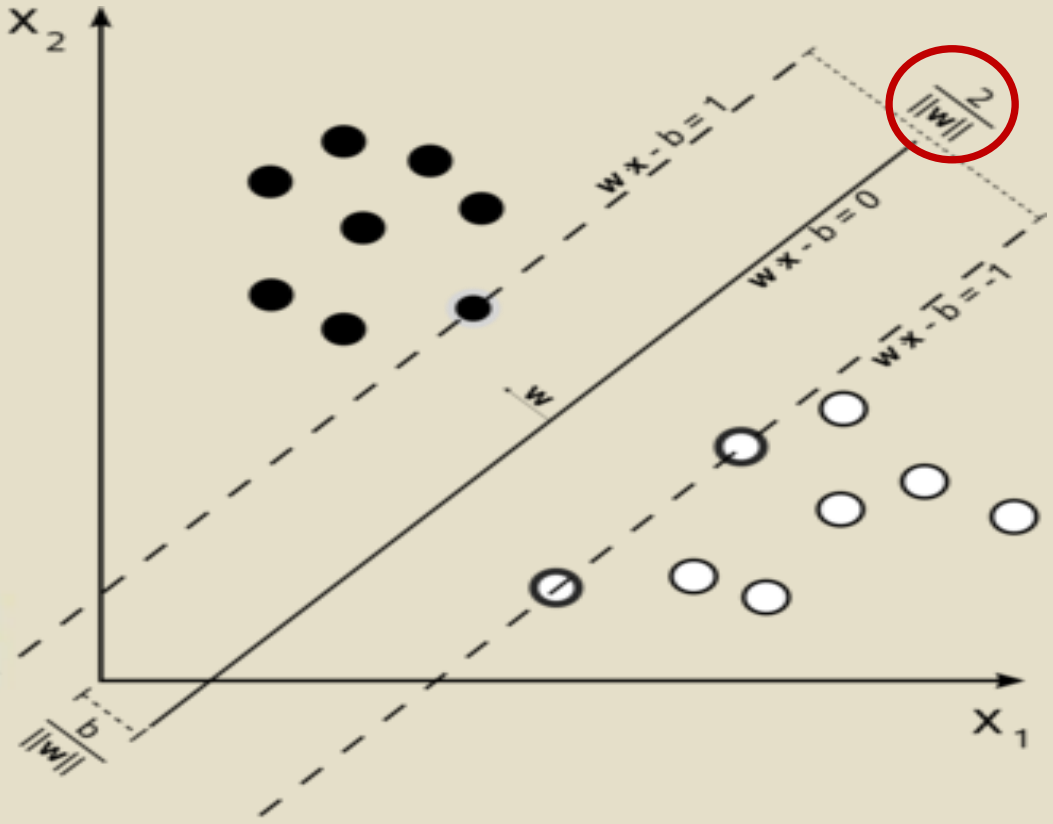


'주 초평면 분류선(Main Hyperplane)'은 두 서포트 벡터 사이의 직선으로 결정!

서포트 벡터는 절편 값이 각 1, -1이 되도록 가중치 w와 b를 조정한 직선

항상 점을 지나게끔 설정!

SVM의 원리



분류선의 위치 결정

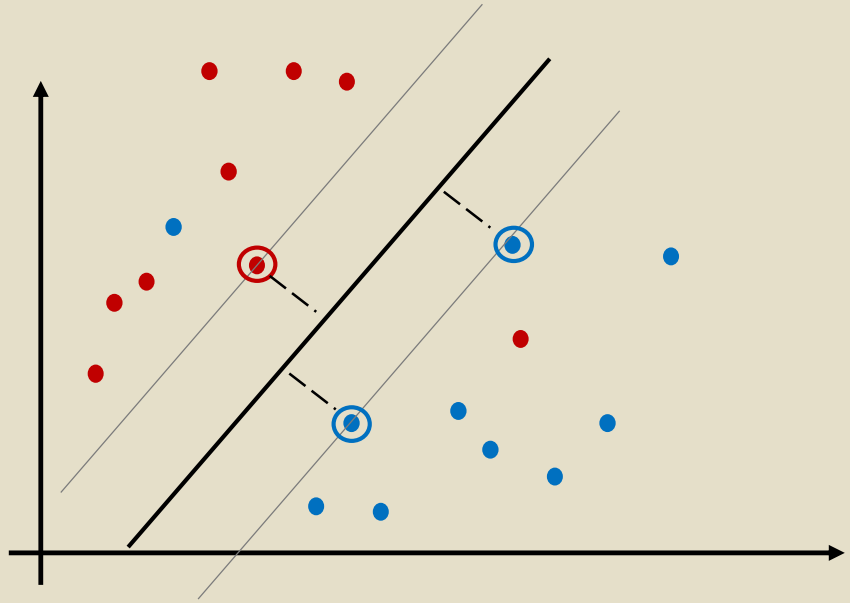
두 집합의 마진인 $\frac{2}{||w||}$ 가 클수록,

잘못된 분류를 할 가능성(Generalization Error)이 줄어든다.

✓ 분류가 뚜렷하게 잘 된다!

SVM의 원리

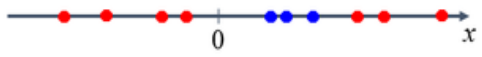
데이터가 뚜렷하게 나뉘지지 않는 경우 : 소프트 마진 방법



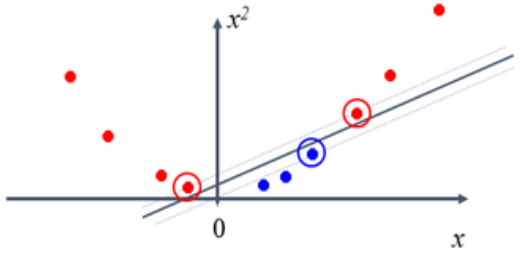
제대로 분리된 point들 중
가장 가까운 point들 간의 **거리를 최대화** 하면서,
주어진 자료들을 최대한 제대로 분리하는 초평면을 찾는 방법.

SVM의 원리

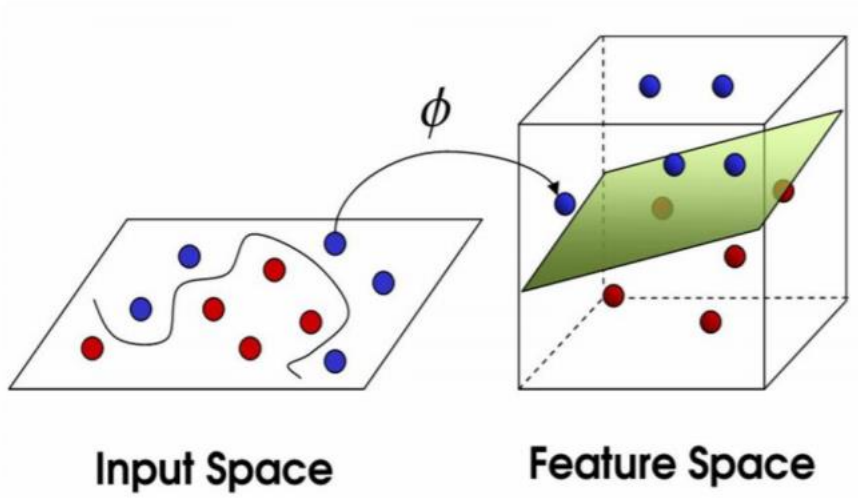
- *However, sometimes data aren`t linearly separable*



- *In this case, we can use Kernel to make data linearly separable*



(1차원 데이터의 경우)

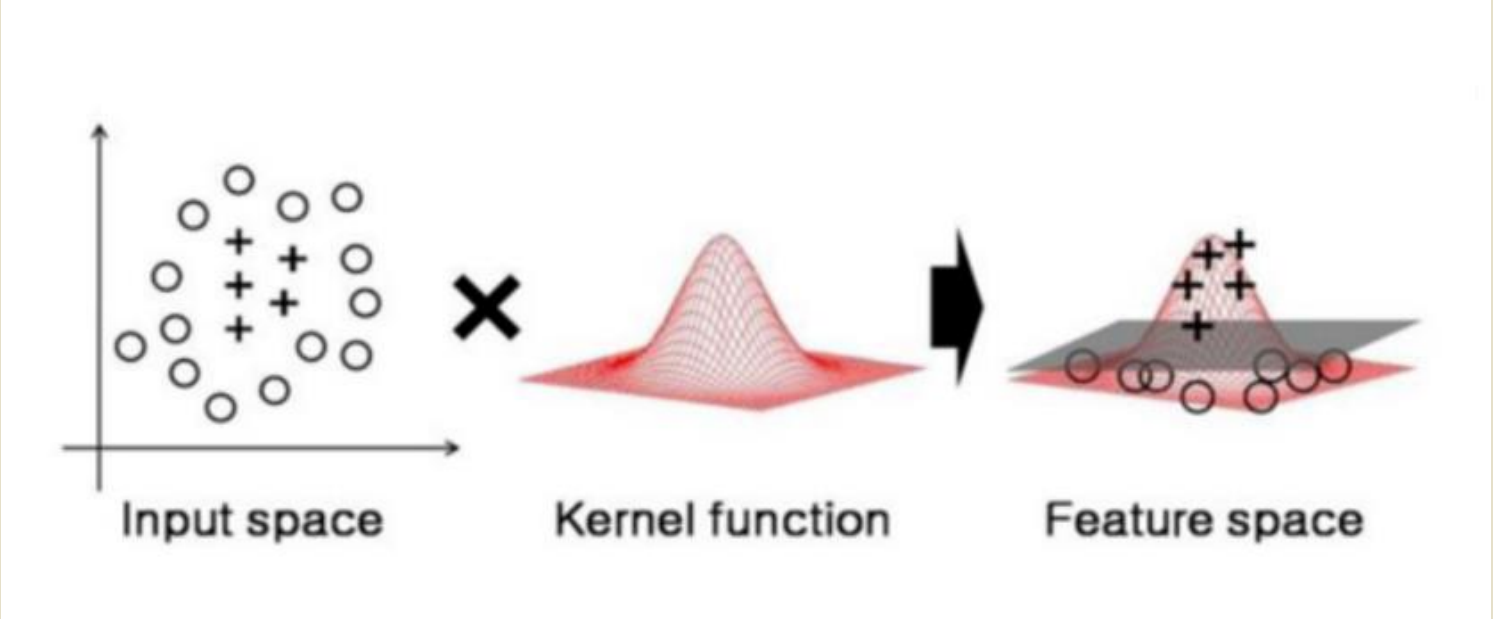


(2차원 데이터의 경우)

만약 현 데이터의 차원의 수로는 뚜렷한 초평면 분류선을 찾을 수 없다면,
커널 트릭(Kernel Trick)을 통해 데이터의 차원을 늘려 분류선을 찾는다!

SVM의 원리

데이터가 비선형적으로 분류되는 경우



(가우시안 커널 사용 예시)

커널의 종류: 데이터변환이 거의 없는 '선형 커널', 신경망과 유사한 '시그모이드 커널', 정규분포의 확률밀도함수를 쓰는 '가우시안 RBF 커널'이 있다.

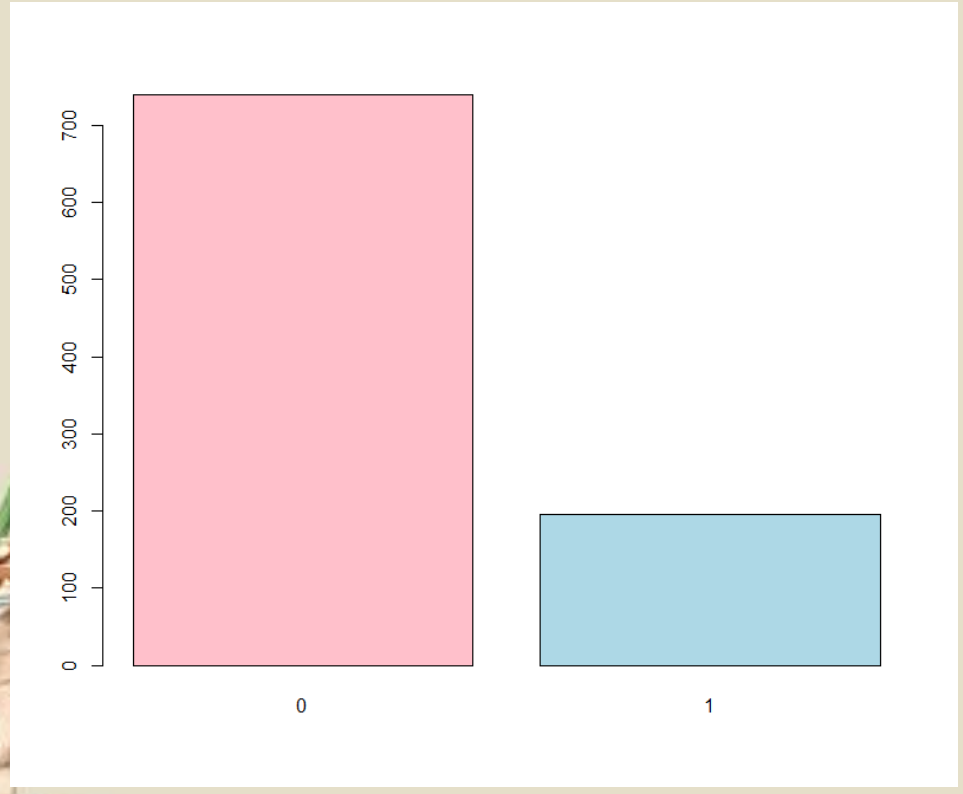
SVM의 장단점

장점	단점
범주나 수치 예측문제에 사용 가능	변수가 많다면 훈련이 느림
노이즈의 영향을 크게 받지 않아 과적합화 되지 않음	해석이 어렵고 복잡한 블랙박스
잘 지원되는 일부 SVM 알고리즘 덕에 신경망보다 사용하기 쉬움	최적의 서포트벡터를 찾기 위해 여러가지 조합의 테스트 필요

SVM in R

위경도 data를 제외한 모든 변수를 활용해 SVM 적합

```
> svm.model = svm(factor(starbucks)~.-lat-lon,data=train,kernel="linear",class.weights=c("0"=0.25))
```



0	1
514	140

0의 비율이 80% 정도 차지
☞ 0과 1의 비율이 4:1편향 완화하기 위해
0에 0.25의 weight를 부여

SVM in R

```
> pred.y = predict(svm.model,newdata=test)
> ct = table(pred.y,test$starbucks); ct

pred.y   0   1
     0 159  12
     1  66  44
> sum(diag(ct))/sum(ct)
[1] 0.7224199
> sum(diag(ct)[2])/sum(ct[,2])
[1] 0.7857143
```

<Full Model SVM>

Test 예측	0	1
0	159	12
1	66	44

정분류율: 약 72.2%

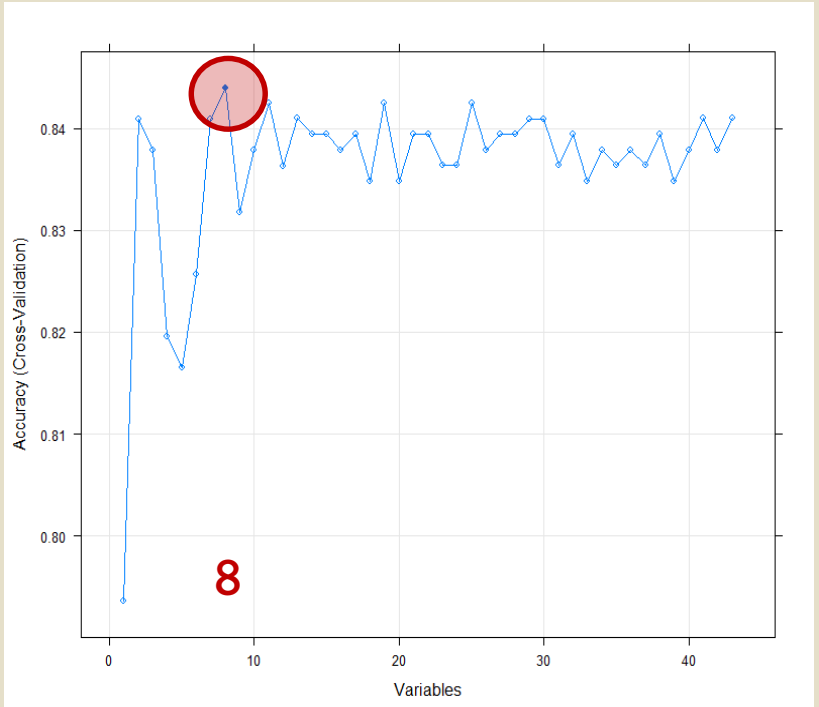
민감도: 약 78.6%

로지스틱 모형과 비교했을 때, 민감도에선 차이가 없지만 전체 예측력이 약간 개선됨

☞ 변수 선택을 하면 어떻게 될까?

SVM in R

```
> predictors(results)
[1] "market"      "move20"      "mwork"       "fwork"       "drink.sales" "move30"
[7] "bank"        "theater"
> plot(results,type=c("g","o"))
```



- 변수 선택 결과 선택된 변수

  20대/30대 유동인구 수,
  남성/여성 직장인구 수,
  카페 음료점 수, 은행 수,
  슈퍼마켓 수

- 5-Fold CV 결과,
  변수 subset 중에서는
  변수가 8개 일 때 정분류율 극대화

SVM in R

변수 선택 결과 선택된 변수들로 적합한 SVM

```
> sel.svm.model = svm(factor(starbucks)~market+move20+mwork+fwork+drink.sales+bank+
+                       move30+theater,data=train,kernel="linear",class.weights=c("0"=0.25))
>
> pred.y = predict(sel.svm.model,newdata=test)
> ct = table(pred.y,test$starbucks); ct

pred.y   0   1
     0 156  17
     1  69  39
> sum(diag(ct))/sum(ct)
[1] 0.6939502
> sum(diag(ct)[2])/sum(ct[,2])
[1] 0.6964286
```

- Full SVM과 동일하게 스타벅스가 없는 경우(0)에 0.25의 weight penalty 부여
- Full model SVM과 결과를 비교해보자

SVM in R

Full model vs. Selected model

<Full Model SVM>

Test 예측	0	1
0	159	12
1	66	44

정분류율: 약 72.2%

민감도: 약 78.6%

<Selected SVM>

Test 예측	0	1
0	156	17
1	69	39

정분류율: 약 69.4%

민감도: 약 69.6%

변수 선택 후 오히려 정분류율과 민감도가 모두 감소

SVM in R

Full model vs. Selected model

<Full Model SVM>

```
> pred.y = predict(svm.model,newdata=test)
> pred = prediction(as.integer(pred.y)-1,test$starbucks)
> auc = performance(pred,measure='auc')
> auc = auc@y.values[[1]]
> auc
[1] 0.7461905
```

AUC : 0.7462

<Selected SVM>

```
> pred = prediction(as.integer(pred.y)-1,test$starbucks)
> auc = performance(pred,measure='auc')
> auc = auc@y.values[[1]]
> auc
[1] 0.694881
```

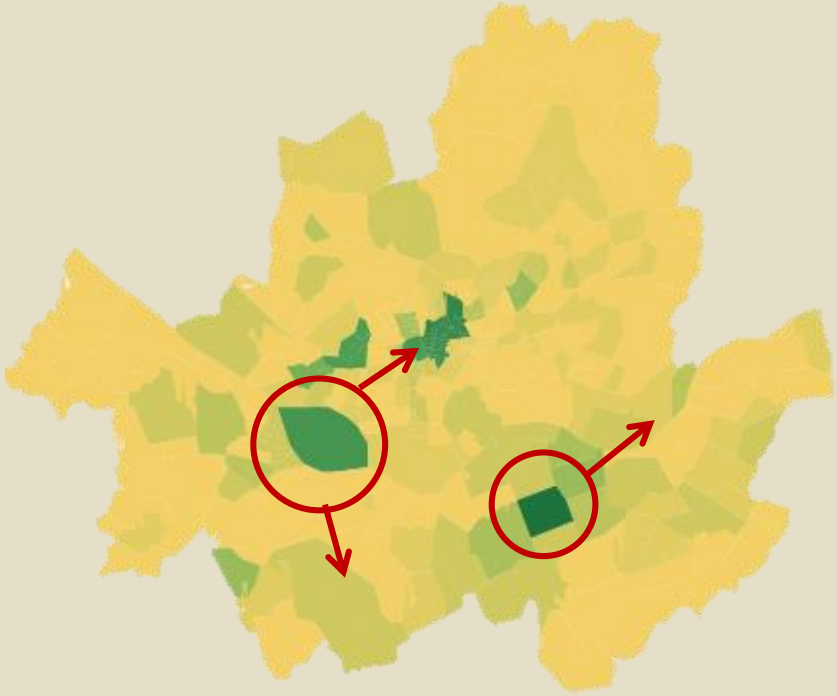
AUC : 0.6949

변수 선택 후 AUC 역시 감소

☞ SVM 모형에서는 모든 변수를 활용하는 것이 가장 적절해 보인다!

5. GAM

공간적 자기상관성



< 서울 지역 스타벅스 분포도 >

한 지역의 특성이 주변 지역에도 영향을 미침

공간적 자기상관성
(Spatial Autocorrelation)

우리 데이터도
공간적 자기상관성을 갖지는 않을까?

Moran's I

```
> Moran.I(raw.dat$starbucks,sb.dist.inv)
$observed
[1] 0.03025967

$expected
[1] -0.0009587728

$sd
[1] 0.00264951

$p.value
[1] 0
```

- H_0 : 공간적 자기 상관성이 없다.
- H_1 : 공간적 자기 상관성이 존재한다.

P-value가 유의수준 보다 작으므로
귀무가설 기각!

공간적 자기상관이 존재!

GAM이란?

공간적 자기상관성 해결을 위한 GAM의 도입

GAM (Generalized Additive Model) : 일반화 가법 모형

① 일반화 선형 모형(GLM : generalized linear model)과
② 가법모형(additive model)의 속성을 혼합한 통계적 모형

$$g(E(Y)) = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p)$$

(① : $g(E(Y))$, ② : $\beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p)$)

변수를 transforming ☞ 선형 이외의 관계 표현
(non-parametric smooth function)

GAM이란?

공간적 자기상관성 해결을 위한 GAM의 도입

선형회귀 —유연화→ GLM —유연화→ GAM

- GLM보다 더 비선형적 관계 표현하는 매우 유연한 모형
  (각 설명 변수 모두에게 smooth function 가정)

- 위경도의 비선형적 효과 고려
  ☞ 다른 변수에 영향을 주던 공간적 자기 상관성 해결 기대

GAM이란?

공간적 자기상관성 해결을 위한 GAM의 도입

앞서 적합한 로지스틱 모형을 바탕으로
로지스틱에서 선택 안 된 변수 + 위경도의 비선형적 효과도 고려한다면
더 좋은 모형이 되지 않을까?

- 로지스틱에서 선택되지 않은 기타 변수들의 비선형적 관계 모두 고려
- 비선형적 관계가 유의한 변수들 만을 활용해 GAM 재 적합
- 두 모형의 예측력 확인 및 비교

GAM 적합

Full GAM 적합

- 위경도 + 로지스틱에서 선택 안 된 변수에 모두 smooth function 적용한 결과

위경도 고려 ☞ 다른 변수들에 대한 공간적 영향력 제거

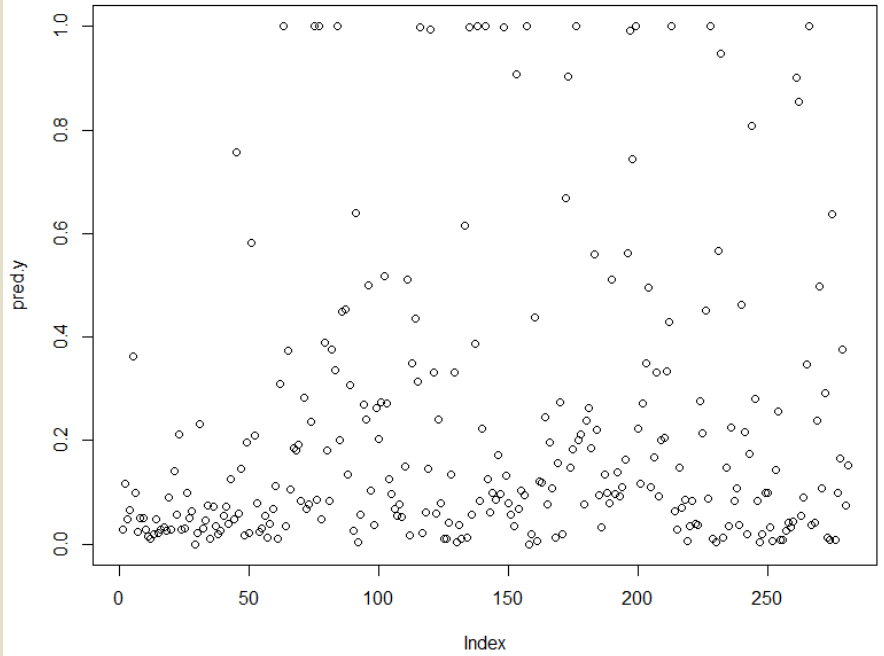
```
gam.full = gam(starbucks~drink.sales+cloth.num+theater+apt.area+fwork+move50+
               market+bank+high+mmove+drink.num+drink.month+s(lat,lon)+
                s(mlive10)+s(mlive20)+s(mlive30)+s(mlive40)+s(mlive50)+
                s(mlive60)+s(flive10)+s(flive20)+s(flive30)+s(flive40)+
                s(flive50)+s(flive60)+s(cos.month)+s(cos.sales)+s(cos.num)+
                s(cloth.month)+s(cloth.sales)+s(apt.num)+s(apt.price)+
                s(fmove)+s(move10)+s(move20)+s(move30)+s(move40)+s(move60)+s(mwork)+
                s(govern)+s(market)+s(subway)+s(bus),data=train,family="binomial")
```

☞ 공간적 자기 상관성 + 비선형 관계를 고려한 모형의 예측력은 어떨까?

GAM 적합

Full GAM 예측력 확인

```
> pred.y = predict.gam(gam.full,newdata=test,type="response")
> plot(pred.y)
```




```
> sb.hat = ifelse(pred.y>=0.21,1,0)
> table(sb.hat,test$starbucks)

sb.hat   0   1
     0 172  19
     1  53  37
> sum(sb.hat==test$starbucks)/nrow(test)
[1] 0.7437722
> table(sb.hat,test$starbucks)[2,2]/sum(test$starbucks)
[1] 0.6607143
```

Test 예측	0	1
0	172	19
1	53	37

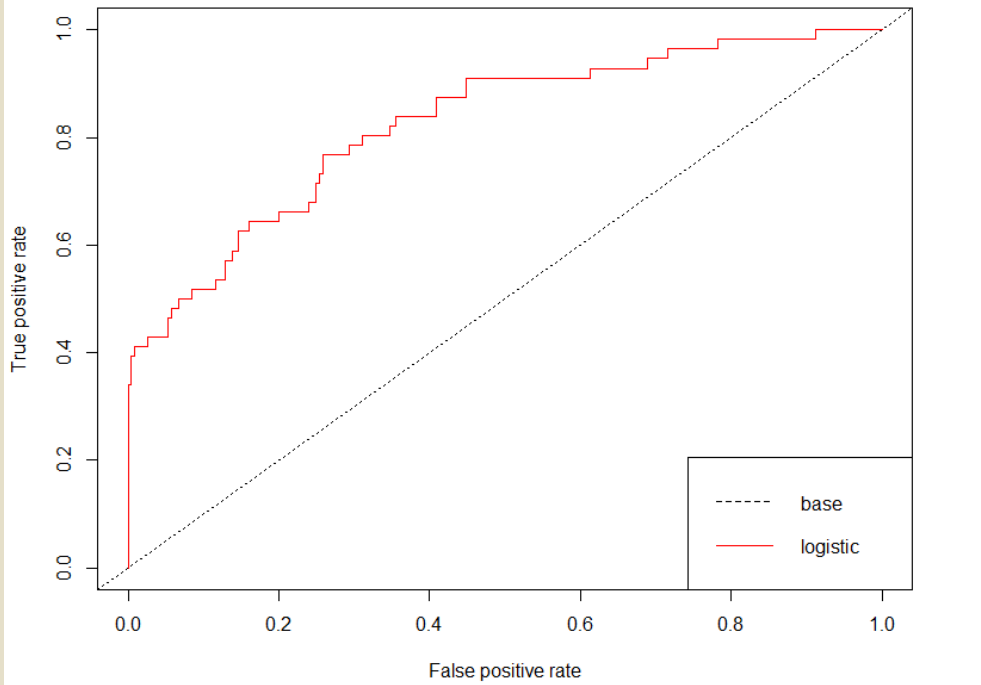
정분류율: 약 74.4%

민감도: 약 66.1%

GAM 적합

Full GAM 예측력 확인

```
> pred = prediction(as.vector(pred.y),test$starbucks)
> roc = performance(pred,measure='tpr',x.measure='fpr')
> plot(roc,col='red')
> legend('bottomright',c('base','logistic'),col=1:2,lty=2:1)
> abline(a=0,b=1,lty='dotted')
```



- ✓ ROC curve가 'y=x'선과 멀리 떨어져 있음
- ✓ AUC 값 확인

```
> auc = performance(pred,measure='auc')
> auc = auc@y.values[[1]]
> auc
[1] 0.8255556
```

AUC : 0.8256

GAM 적합

비선형 관계 유의성 확인

```
Approximate significance of smooth terms:
                edf Ref.df Chi.sq  p-value
s(lat,lon)    1.1856     49  2.715 0.082274 .
s(mlive10)    0.3149      9  0.142 0.424742
s(mlive20)    0.3603      9  0.223 0.263780
s(mlive30)    0.1931      9  0.078 0.517975
s(mlive40)    0.1689      9  0.038 0.754797
s(mlive50)    0.1079      9  0.053 0.447256
s(mlive60)    0.1496      9  0.067 0.491726
s(flive10)    0.1926      9  0.139 0.245496
s(flive20)    0.5007      9  0.857 0.061115 .
s(flive30)    0.2163      9  0.035 0.842915
s(flive40)    0.1935      9  0.070 0.544026
s(flive50)    0.1907      9  0.064 0.595673
s(flive60)    0.1197      9  0.013 1.000000
s(cos.month)  0.4871      9  0.327 0.355306
s(cos.sales)  0.6536      9  0.942 0.207620
s(cos.num)    0.6711      9  0.900 0.166995
s(cloth.month) 0.2850     9  0.091 0.598210
s(cloth.sales) 0.8209     9  2.387 0.075821 .
s(apt.num)    0.3653      9  0.129 0.615470
s(apt.price)  0.3081      8  0.017 1.000000
s(fmove)      3.8826      9 18.325  1.2e-06 ***
s(move10)     0.2777      9  0.014 1.000000
s(move20)     1.2984      9  4.779 0.012432 *
s(move30)     1.4349      9  1.679 0.200108
s(move40)     2.1964      9  9.760 0.000342 ***
s(move60)     0.6382      9  0.932 0.186719
s(mwork)      0.5541      9  0.519 0.323839
s(govern)     2.9172      9 14.571 0.001078 **
s(market)     0.8508      5  0.877 0.309110
s(subway)     1.7732      9  4.243 0.087302 .
s(bus)        0.2757      9  0.008 1.000000
```

많은 변수들의 비선형적 관계가 유의하지 않음

☞ 유의한 결과를 가진 변수만으로 모형 재 적합 후 예측력을 다시 확인해보자!

GAM 적합

Selected GAM 적합

```
Approximate significance of smooth terms:
                edf Ref.df Chi.sq  p-value
s(lat,lon)    1.1856     49  2.715 0.082274 .
s(mlive10)    0.3149      9  0.142 0.424742
s(mlive20)    0.3603      9  0.223 0.263780
s(mlive30)    0.1931      9  0.078 0.517975
s(mlive40)    0.1689      9  0.038 0.754797
s(mlive50)    0.1079      9  0.053 0.447256
s(mlive60)    0.1496      9  0.067 0.491726
s(flive10)    0.1926      9  0.139 0.245496
s(flive20)    0.5007      9  0.857 0.061115 .
s(flive30)    0.2163      9  0.035 0.842915
s(flive40)    0.1935      9  0.070 0.544026
s(flive50)    0.1907      9  0.064 0.595673
s(flive60)    0.1197      9  0.013 1.000000
s(cos.month)  0.4871      9  0.327 0.355306
s(cos.sales)  0.6536      9  0.942 0.207620
s(cos.num)    0.6711      9  0.900 0.166995
s(cloth.month) 0.2850     9  0.091 0.598210
s(cloth.sales) 0.8209     9  2.387 0.075821 .
s(apt.num)    0.3653      9  0.129 0.615470
s(apt.price)  0.3081      8  0.017 1.000000
s(fmove)      3.8826      9 18.325  1.2e-06 ***
s(move10)     0.2777      9  0.014 1.000000
s(move20)     1.2984      9  4.779 0.012432 *
s(move30)     1.4349      9  1.679 0.200108
s(move40)     2.1964      9  9.760 0.000342 ***
s(move60)     0.6382      9  0.932 0.186719
s(mwork)      0.5541      9  0.519 0.323839
s(govern)     2.9172      9 14.571 0.001078 **
s(market)     0.8508      5  0.877 0.309110
s(subway)     1.7732      9  4.243 0.087302 .
s(bus)        0.2757      9  0.008 1.000000
```

- H_0 : 비선형 관계가 유의하지 않다
- H_1 : 비선형 관계가 유의하다

✓ 비선형 관계가 존재하는 변수들

- 위경도
- 20대 여성 거주인구 수
- 의류점 총 매출액
- 여성 유동인구 수
- 20대 유동인구 수
- 40대 유동인구 수
- 관공서 수
- 지하철 역 수

GAM 적합

```
gam.sel = gam(starbucks ~ drink.sales + cloth.num + theater + apt.area +
                fwork + move50 + market + bank + high + mmove + drink.num +
                drink.month + s(lat,lon)+
                s(flive20)+s(cloth.sales)+s(fmove)+s(move20)+s(move40)+
                s(govern)+s(subway),data=train,family="binomial")
```

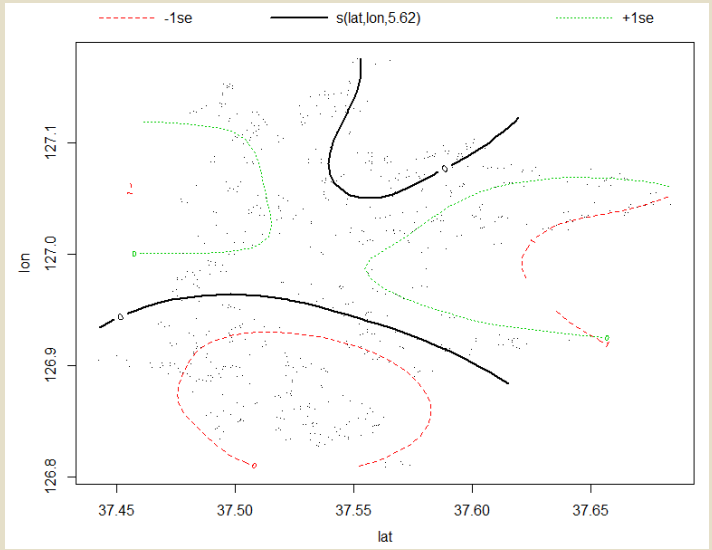
```
Approximate significance of smooth terms:
                edf Ref.df Chi.sq p-value
s(lat,lon)    5.6226     29  9.239 0.03315 *
s(flive20)    1.3823      9  7.996 0.04109 *
s(cloth.sales) 0.3937     9  0.366 0.32557
s(fmove)      2.1024      9  4.838 0.01806 *
s(move20)     1.4087      9  2.374 0.05529 .
s(move40)     2.8579      9 10.602 0.00158 **
s(govern)     2.4684      9  7.757 0.02695 *
s(subway)     1.8494      8  6.970 0.02224 *
```

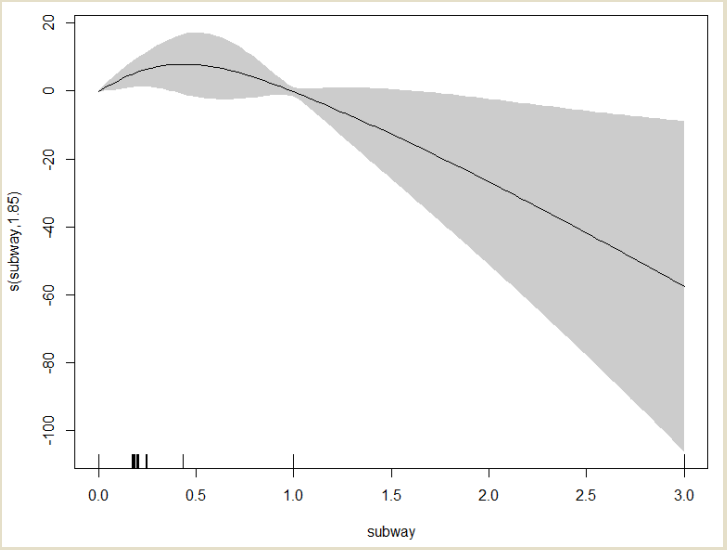
- H_0 : 비선형 관계가 유의하다
- H_1 : 비선형 관계가 유의하지 않다

☞ 거의 모든 비선형 관계 변수들이 유의한 것을 확인!

GAM 적합

```
> plot(gam.sel,se=T,shade=T)
```





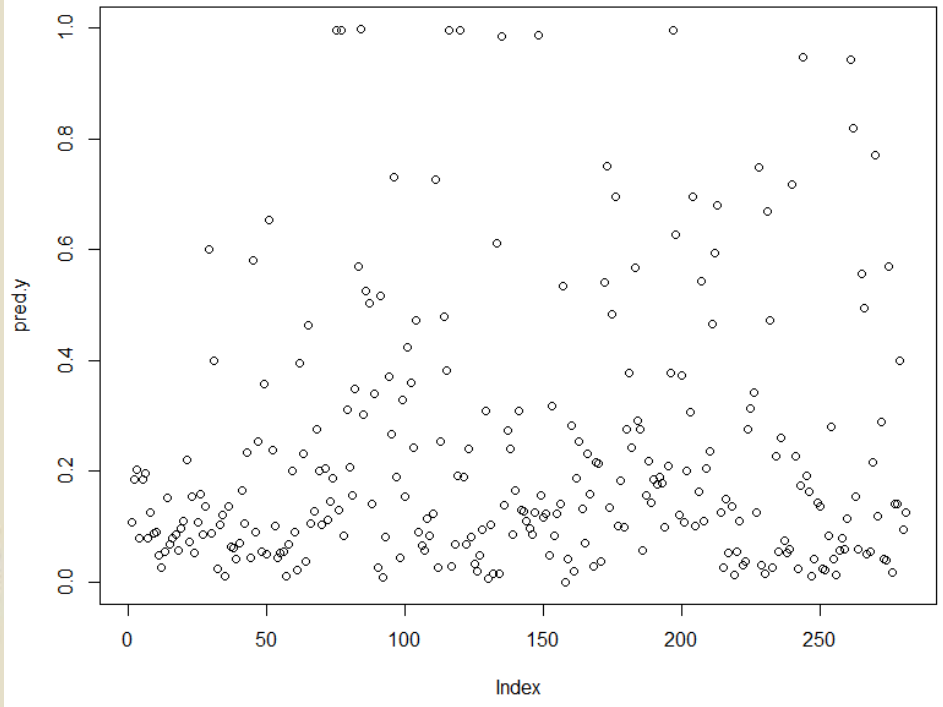
변수들의 비선형 관계 확인 가능

☞ **유의한 비선형 관계들 만을 고려한 모형의 예측력은 어떨까?**

GAM 적합

Selected GAM 예측력 확인

```
> pred.y = predict.gam(gam.sel,newdata=test,type="response")
> plot(pred.y)
```




```
> sb.hat = ifelse(pred.y>=0.21,1,0)
> table(sb.hat,test$starbucks)

sb.hat   0   1
     0 172  13
     1  53  43
> sum(sb.hat==test$starbucks)/nrow(test) # 0.7651246
[1] 0.7651246
> table(sb.hat,test$starbucks)[2,2]/sum(test$starbucks)
[1] 0.7678571
```

Test 예측	0	1
0	172	13
1	53	43

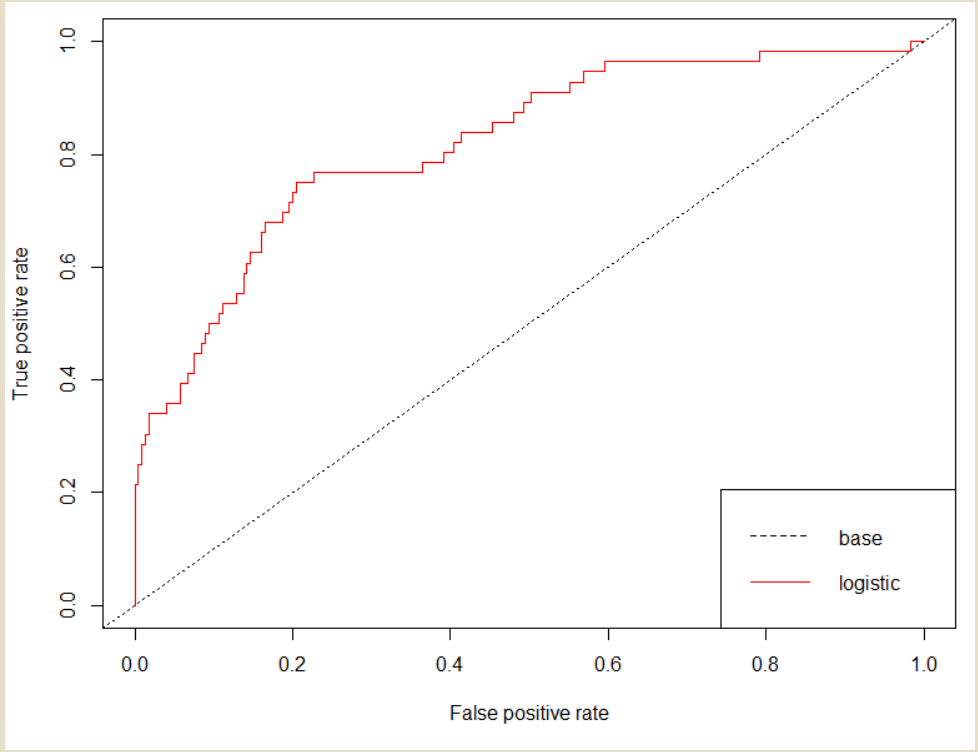
정분류율: 약 76.5%

민감도: 약 76.8%

GAM 적합

Selected GAM 예측력 확인

```
> pred = prediction(as.vector(pred.y),test$starbucks)
> roc = performance(pred,measure='tpr',x.measure='fpr')
> plot(roc,col='red')
> legend('bottomright',c('base','logistic'),col=1:2,lty=2:1)
> abline(a=0,b=1,lty='dotted')
```



- ✓ ROC curve가 'y=x'선과 멀리 떨어져 있음
- ✓ AUC 값 확인

```
> auc = performance(pred,measure='auc')
> auc = auc@y.values[[1]]
> auc
[1] 0.8193651
```

AUC : 0.8194

☞ 양호한 예측력을 보인다

GAM 적합

Full GAM vs. Selected GAM

<Full Model GAM>

Test 예측	0	1
0	172	19
1	53	37

정분류율: 약 74.4%

민감도: 약 66.1%

<Selected GAM>

Test 예측	0	1
0	172	13
1	53	43

정분류율: 약 76.5%

민감도: 약 76.8%

최종적으로 selected gam 모델 선택!

6. 최종모형 선택 및 결론

모형 별 결과값 비교

<Logistic>

Test 예측	0	1
0	157	12
1	68	44

정분류율: 약 71.5%

민감도: 약 78.6%

AUC: 81.9%

<SVM>

Test 예측	0	1
0	159	12
1	66	44

정분류율: 약 72.2%

민감도: 약 78.6%

AUC: 74.6%

<GAM>

Test 예측	0	1
0	172	13
1	53	43

정분류율: 약 76.5%

민감도: 약 76.8%

AUC: 81.9%

AUC가 현저히 떨어지는 SVM 1차 탈락!

GLM vs GAM

<GLM>

```
> cvstat = numeric(K)
> for (k in 1:K)
+ {
+   index = cvf$subsets[cvf$which == k]
+   tr = data[-index,]
+   te = data[index,]
+   fit = glm(formula = starbucks ~ drink.sales + cloth.num + theater +
+               apt.area + fwork + move50 + market + bank + high + mmove +
+               drink.num + drink.month, family = "binomial", data = tr)
+   pred = predict(fit, newdata = te, type='response')
+   yhat = ifelse(pred>=0.21,1,0)
+   cvstat[k] = 1 - sum(te$starbucks == yhat)/nrow(te)
+ }
> mean(cvstat)
[1] 0.2673797
```

<GAM>

```
> cvstat = numeric(K)
> for (k in 1:K)
+ {
+   index = cvf$subsets[cvf$which == k] ## index of data which will be used as test set
+   tr = data[-index,]
+   te = data[index,]
+   gam.sel = gam(starbucks ~ drink.sales + cloth.num + theater + apt.area +
+                   fwork + move50 + market + bank + high + mmove + drink.num +
+                   drink.month + s(lat,lon)+
+                   s(flive20)+s(cloth.sales)+s(fmove)+s(move20)+s(move40)+
+                   s(govern)+s(subway),method="REML",select=T,data=tr,family="binomial")
+   pred = predict(fit, newdata = te, type='response')
+   yhat = ifelse(pred>=0.21,1,0)
+   cvstat[k] = 1 - sum(te$starbucks == yhat)/nrow(te)
+ }
> mean(cvstat)
[1] 0.2545455
```

GLM과 GAM 은 CV error 값으로 비교해보자!

GLM vs GAM Cross-Validation

<GLM>

Test 예측	0	1
0	157	12
1	68	44

정분류율 : 약 71.5%

민감도 : 약 78.6%

AUC : 81.9%

CV error : 0.2673

<GAM>

Test 예측	0	1
0	172	13
1	53	43

정분류율 : 약 76.5%

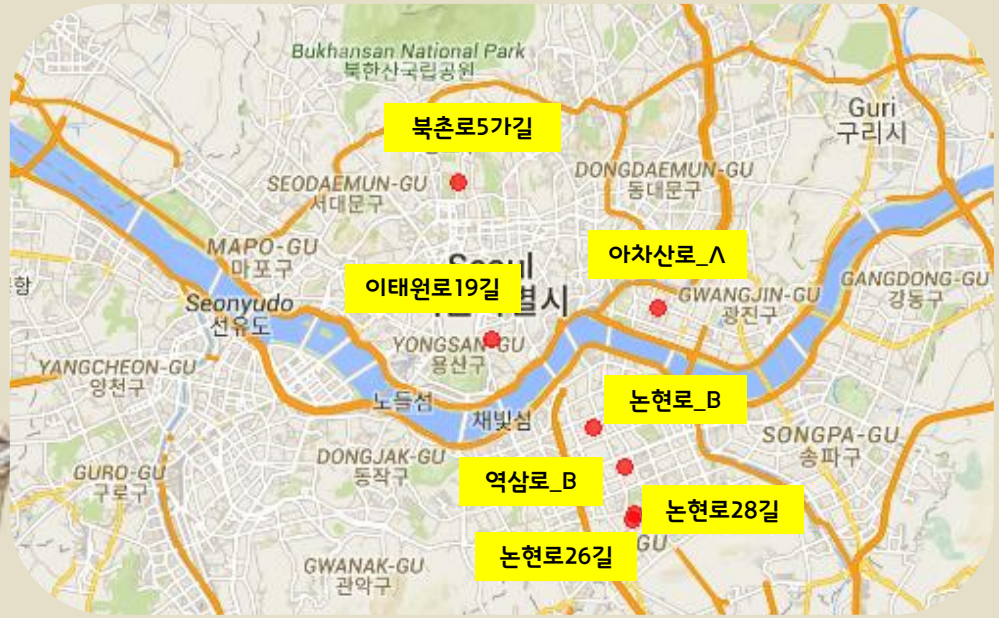
민감도 : 약 76.8%

AUC : 81.9%

CV error : 0.2545 (최종 선택!)

최종 예측

```
> final.sb = predict(gam.sel,newdata=candidate,type="response")
> final.sb = ifelse(final.sb>=0.8,1,0)
> gam.result = candidate[which(final.sb==1&candidate$starbucks==0),"road.name"]
> gam.result
[1] 북촌로5가길  아차산로_A  이태원로19길 논현로_B  역삼로_C  논현로28길  논현로26길
```



왜 스타벅스가 0개인 곳만 추출했나?

☞ "집중적 초토화" 전략을 사용하는 스타벅스,
이미 입점한 곳 보다는
새롭게 입점할 곳에 집중!

분석 한계점

1. 스타벅스 개별 지점의 특성까지 고려한 보다 세부적인 분석을 원했으나,
   - 세부적인 데이터를 모으는 것이 어려웠고
   - 지점 별 데이터를 수집할 경우에는 적절한 분석법을 찾을 수 없었다

2. 데이터 수집 상의 한계
   - 공시지가 데이터가 결측률이 심해 아파트 가격으로 대체했다.
   - 매출 데이터를 얻는 것이 불가능했다.

3. 데이터 가공 시 발생한 error 들이 존재
   - 법정동 코드와 행정동 코드 매칭 문제
   - 스타벅스를 각 상권에 매칭할 때 KNN 방식을 적용하는 과정에서 현실과 차이가 존재할 수 있었다.

4. 분석 과정 및 결과의 한계
   - GAM 모형을 최종적으로 선택했으나 로지스틱에서 크게 개선된 결과를 보이지는 못함 (비용 상의 문제)
   - GAM 모형에서 가장 이상적인 변수 subset selection을 하는게 현실적으로 불가능했음
   - 통계적인 분석 방법론만을 사용했기 때문에, 실제 기업에서 고려할 마케팅 등등의 요소는 고려되지 못함

분석 의의

1. 수집 가능한 데이터 내에서 다양한 프로그램을 활용한 통계적 분석을 시도

SVM, GAM 등등..

2. 최종선택 모형의 예측률이 좋았기 때문에 납득 가능하고 유의미한 결론을 내릴 수 있었다

>> 타 경쟁 업체 및 부동산 투자자들에게 참고 지표가 되겠다는
본래의 목적을 어느정도 달성했다고 생각 함!

가 암 사 합 니 다