KBO 승패 예측

목차

1. 문제 선정 배경

2. 목표

3. 데이터 수집 및 분석

4. 데이터 마이닝 모형화

5. 결과 해석

6. 팀원 역할

1. 문제 선정 배경

<닐슨 코리아, 2017년>

[한화 현장노트] 서산 총집합 데이터 야구 … 싹 바뀐 한화 마무리캠프
스포티비뉴스 | 7시간 전 | 네이버뉴스
내년에는 세밀한 작전 야구를 펼칠 수 있을 것 같다는 기대가 된다"고 밝혔다. 한화는 이달 부터 훈련 중 … 한 감독은 "예전에는 위치 감독이 경험과 감으로 이야기를 많이 했는데 데

[프로야구] 키움 히어로즈 '5대 감독' 손혁 취임 … "데이터 기반으로 좋은 팀…
뉴스핌 | 2일 전
이어 "데이터를 기반으로 선수들과 함께 좋은 성적을 거둘 수 있도록 노력하겠다"고 강조했다. [서 … 척스카

[베이스볼 비키니] '스마트' 야구가 우승 비결 주간동아 | 5일 전 | 네이버뉴스
KBO의 데이터 야구, 아직 초기 단계에 불과 0358009999, 이 열 자리 숫자는 무슨 뜻일까요. 빼기 부호(-)를 좀 넣으면 다르게 보일지 모릅니다. 03-5800-9999, 갑자기 전화번호처럼 보

불펜에도 레이더 장비 가동, 기아 데이터야구 본격 시동
스포츠월드 | 2019.11.09. | 네이버뉴스
점차 데이터 야구가 KBO리그에도 도입되면서, 훈련에 사용되는 랩소도, 블라스트 등 데이터 측정 장비를 활용하는 구단이 늘고 있다. 하지만 이와 같은 장비는 모두 카메라 또는 센

프로야구 삼성, 데이터 야구로 승부수 띄워 대구MBC | 2일 전
마무리 캠프를 펼치고 있는 프로야구 삼성 라이온즈가 데이터 야구에 집중하고 있습니다. 이미 지난 12일 경산 볼파크에서 선수단을 대상으로 데이터 야구 교육을 실시한 삼성은 마무리 캠프 기간 동안 데이터 야구…

2. 목표

2019년 KBO 정규 시즌 기록을 기반으로 2019 KBO 와일드카드전의 승패를 예측해 보자.

3. 데이터 수집 및 분석

<데이터 수집> - STATIZ

3월

월	화	수	목	금	토	일
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23 키움 7 4 롯데 KT 4 7 SK 한화 4 5 두산 삼성 0 7 NC LG 2 0 KIA	24 키움 2 6 롯데 KT 3 6 SK 한화 11 1 두산 삼성 4 3 NC LG 9 3 KIA
25	26 삼성 2 7 롯데 LG 6 3 SK 키움 2 7 두산 KT 8 9 NC 한화 13 7 KIA	27 삼성 23 4 롯데 LG 1 2 SK 키움 2 3 두산 KT 1 4 NC 한화 4 9 KIA	28 삼성 12 7 롯데 LG 1 3 SK 키움 5 4 두산 KT 2 6 NC 한화 4 6 KIA	29 SK 5 2 키움 NC 0 9 한화 두산 3 2 삼성 롯데 1 2 LG KIA 3 6 KT	30 SK 1 3 키움 NC 5 13 한화 두산 4 2 삼성 롯데 7 1 LG KIA 2 6 KT	31 SK 8 7 키움 NC 6 3 한화 두산 9 4 삼성 롯데 5 6 LG KIA 4 2 KT

Twins 2 : 0 Tigers

챔피언스필드, 03-23

일일일정 | 미리보기 | 경기요약 | 박스스코어 | 경기로그 | 스탯위키

타격기록 (LG)

	이름	P	TPA	AB	R	H	HR	RBI	BB	HBP	SO	GO	FO	PIT	GDP	LOB	AVG	OPS	LI	WPA	RE24
1	이형종	8	4	4	0	2	0	1	0	0	1	1	0	12	0	2	0.500	1.250	1.08	0.106	0.97
2	오지환	6	4	4	0	0	0	0	0	0	3	1	0	15	1	2	0.000	0.000	1.04	-0.145	-1.73
3	김현수	7	4	4	0	0	0	0	0	0	2	1	1	21	0	1	0.000	0.000	0.66	-0.068	-0.82
4	조셉	3	4	4	0	0	0	0	0	0	3	0	1	20	0	0	0.000	0.000	0.55	-0.057	-0.75
5	채은성	9	4	4	1	2	0	0	0	0	1	0	1	19	0	0	0.500	1.250	0.76	0.035	0.02
6	박용택	0	4	3	0	2	0	0	1	0	0	0	1	14	0	1	0.667	1.750	0.73	0.037	0.43
	신민재	r	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.000	0.000		0.000	0.00
7	양종민	5	2	2	0	0	0	0	0	0	1	1	0	9	0	2	0.000	0.000	1.56	-0.115	-0.94
	서상우	h	1	1	0	0	0	0	0	0	0	1	0	4	1	1	0.000	0.000	1.29	-0.057	-0.57
	윤진호	5	1	1	0	0	0	0	0	0	0	1	0	6	0	1	0.000	0.000	0.29	-0.008	-0.25
8	유강남	2	3	3	0	2	0	1	0	0	1	0	0	11	0	0	0.667	1.333	1.25	0.145	1.00
9	정주현	4	3	1	1	0	0	0	2	0	0	1	0	15	1	1	0.000	0.667	1.06	-0.038	-0.35
팀 합계			34	31	2	8	0	2	3	0	12	7	4	146	3	11	0.258	0.678	0.91	-0.164	-3.00

투구기록 (LG)

이름	IP	TBF	H	R	ER	BB	HBP	K	HR	GO-FO	PIT-S	IR-IS	GSC	ERA	WHIP	LI	WPA	RE24
윌슨 (승, 1-0)	7.0	26	3	0	0	2	1	4	0	7-8	93-58	0-0	73	0.00	0.71	1.3	0.484	3.88
신정락 (홀, 1)	1.0	4	0	0	0	1	0	1	0	2-0	17-9	0-0		0.00	1.00	2.2	0.089	0.56
정찬헌 (세, 1)	1.0	3	0	0	0	0	0	1	0	1-1	9-6	0-0		0.00	0.00	1.2	0.091	0.55
팀 합계	9.0	33	3	0	0	3	1	6	0	10-9	119-73	0-0	73	0.00	0.67	1.4	0.664	4.99

3. 데이터 수집 및 분석

<데이터 합병>

date	team	win	ha	H	RBI	BB	HBP	SO	AVG	OPS	IP	TBF	ER	BB	K	HR	ERA
20190323	KIA	0	1	0.333333	0	0.333333	0.111111	0.666667	0.083333	0.324	1.266667	5.666667	0.333333	0.5	2	0	4.75
20190324	KIA	0	1	0.818182	0.272727	0	0	0.909091	0.151909	0.378545	2.075	11.5	2	1.5	1.25	0.5	4.8375
20190326	KIA	0	1	0.5	0.375	0.375	0.0625	0.5	0.124125	0.3405	2.25	11.5	3.25	1.5	1.75	0.75	8.4875
20190327	KIA	1	1	1.083333	0.75	0.75	0	0.666667	0.197083	0.558333	2.25	10	0.75	1.25	1.75	0.25	2.4375
20190328	KIA	1	1	0.75	0.5	0.583333	0	0.25	0.20225	0.608583	2.075	8.25	1	0.5	2.5	0	3.8775
20190329	KIA	0	0	0.636364	0.272727	0.090909	0	1.181818	0.198	0.596818	2.666667	13.33333	2	0.333333	1.666667	0.333333	4.643333
20190330	KIA	0	0	0.6	0.2	0.4	0.1	1	0.212	0.6218	1.825	10	1.25	1.5	2.5	0.25	5.775
20190331	KIA	1	0	0.7	0.4	0.5	0.1	0.4	0.2009	0.5708	1.52	8.2	0.4	1	2.2	0.2	4.258
20190402	KIA	1	0	0.636364	0.272727	0.545455	0.181818	0.818182	0.197364	0.579273	2.075	9	0.25	0.5	2	0	3.4525
20190403	KIA	0	0	0.75	0.25	0.25	0	1.083333	0.23375	0.64475	1.825	8	1.25	0.5	1	0.25	5.6625
20190404	KIA	0	0	0.615385	0.538462	0.384615	0	0.461538	0.200692	0.587154	1.825	10.25	3	1	1.5	0.75	6.6125
20190405	KIA	1	1	1	0.454545	0.181818	0	0.818182	0.257818	0.705545	2.075	10.75	1	1.5	2.75	0.25	4.5925
20190406	KIA	0	1	0.571429	0.285714	0.214286	0	0.571429	0.1795	0.488	1.66	10.8	2.4	2.2	1.2	0	6.724
20190410	KIA	1	1	0.75	0.166667	0.25	0.166667	0.5	0.234	0.64925	1.72	7.6	0.2	0.8	1.2	0	2.704
20190411	KIA	0	1	0.692308	0.153846	0.230769	0	0.692308	0.213231	0.611615	2.766667	12.66667	1.333333	0	3	0	6.043333

V = 승패 / ha = 홈, 어웨이 /
H = 안타 / RBI = 타점 / BB = 볼넷 / HBP = 몸에 맞는 공 / SO = 삼진 / AVG = 타율 / OPS = 장타율 + 출루율 /
IP = 투구 수 / TBF = 상대한 타자 수 / ER = 자책점 / BB = 볼넷 / K = 삼진 / HR = 홈런 / ERA = 평균 자책점

3. 데이터 수집 및 분석

<데이터 전처리>

date	team	v	ha	H	RBI	BB
20190323	KIA	0	1	0.333333	0	0.333333
20190324	KIA	0	1	0.818182	0.272727	0
20190326	KIA	0	1	0.5	0.375	0.375
20190327	KIA	1	1	1.083333	0.75	0.75
20190328	KIA	1	1	0.75	0.5	0.583333
20190329	KIA	0	0	0.636364	0.272727	0.090909
20190330	KIA	0	0	0.6	0.2	0.4
20190331	KIA	1	0	0.7	0.4	0.5
20190402	KIA	1	0	0.636364	0.272727	0.545455
20190403	KIA	0	0	0.75	0.25	0.25
20190404	KIA	0	0	0.615385	0.538462	0.384615
20190405	KIA	1	1	1	0.454545	0.181818
20190406	KIA	0	1	0.571429	0.285714	0.214286
20190410	KIA	1	1	0.75	0.166667	0.25
20190411	KIA	0	1	0.692308	0.153846	0.230769
20190413	KIA	1	0	0.571429	0.428571	0.071429
20190414	KIA	1	0	0.9	0.4	0.2

= rand()

	date	team	v	ha	H	RBI	BB
0.272381	20190704	두산	1	0	0.583333	0.333333	0.166667
0.6333	20190905	NC	0	1	0.333333	0	0.083333
0.187688	20190329	NC	0	0	0.5	0	0.142857
0.940218	20190417	키움	1	0	1	0.454545	0.272727
0.834748	20190809	KIA	1	1	1.076923	0.692308	0.230769
0.530474	20190501	두산	0	0	0.538462	0.076923	0.461538
0.592269	20190323	롯데	0	1	0.5	0.285714	0.071429
0.008304	20190615	KIA	0	0	0.307692	0	0.307692
0.136371	20190606	키움	1	1	1	0.545455	0.454545
0.37535	20190809	두산	1	1	0.6	0.3	0.5
0.612159	20190623	KIA	1	0	1.2	0.466667	0.4
0.985511	20190727	KT	0	1	0.642857	0.142857	0.285714
0.233951	20190828	키움	0	0	0.357143	0.071429	0.5
0.115151	20190416	SK	0	0	0.727273	0.272727	0
0.280486	20190412	NC	1	1	0.666667	0.222222	0.333333
0.734749	20190417	한화	1	0	0.8	0.8	0.7
0.226141	20190804	삼성	1	0	0.777778	0.222222	0.111111
0.047061	20190818	LG	1	0	0.733333	0.533333	0.266667

date	team	win	ha	H	RBI	BB
20190522	SK	1	0	0.5	0.166667	0.416667
20190630	SK	1	0	1.285714	0.857143	0.5
20190803	KIA	1	1	0.833333	0.416667	0.166667
20190810	SK	1	0	0.636364	0.181818	0.363636
20190916	키움	1	0	1	0.454545	0.181818
20190907	롯데	0	0	0.818182	0.090909	0
20190627	두산	1	0	1.142857	0.642857	0.214286
20190713	SK	1	1	1	0.4	0.3
20190504	NC	1	1	0.615385	0.307692	0.153846
20190712	KIA	1	1	0.777778	0.555556	0.333333
20190509	한화	1	0	0.909091	0.545455	0.181818
20190830	두산	1	0	0.8	0.3	0.3
20190818	KIA	0	1	0.692308	0.076923	0.153846
20190830	롯데	0	0	0.4	0.1	0.2
20190420	NC	0	0	0.909091	0.363636	0.090909
20190930	한화	0	1	0.692308	0.153846	0
20191001	NC	0	0	0.944444	0.222222	0.111111
20190406	롯데	1	1	0.727273	0.727273	0.454545

RAND() 함수를 이용하여 난수를 생성하고 이를 통해 전체 데이터를 랜덤하게 섞어준다.

4. 데이터 마이닝 모형화

1 단계 STEP1	2 단계 STEP2	3 단계 STEP3	4 단계 STEP4
Decision Tree	Random Forest	Logistic Regression	ANN

4. 데이터 마이닝 모형화 - 데이터 준비 -

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	date	team	win	ha	H	RBI	BB	HBP	SO	AVG	OPS	IP	TBF	ER	BB	K	HR	ERA
2	20190522	SK	1	0	0.5	0.166667	0.416667	0.083333	0.416667	0.241	0.6165	2.25	8.25	0	0.25	2.75	0	2.8925
3	20190630	SK	1	0	1.285714	0.857143	0.5	0.071429	0.214286	0.237714	0.656357	1.66	8.6	1.2	0.6	1.6	0.4	3.074
4	20190803	KIA	1	1	0.833333	0.416667	0.166667	0.083333	0.5	0.256833	0.691833	2.075	8.25	0	0.75	2	0	4.3375
5	20190810	SK	1	0	0.636364	0.181818	0.363636	0.090909	0.454545	0.272455	0.727091	2.25	8.25	0	0.5	2	0	2.495
6	20190916	키움	1	0	1	0.454545	0.181818	0	0.818182	0.285818	0.778909	3	12	0	1	1.666667	0	2.99
7	20190907	롯데	0	0	0.818182	0.090909	0	0	0.909091	0.257455	0.688545	1.825	10	1.25	0.75	1.25	0	5.4925
8	20190627	두산	1	0	1.142857	0.642857	0.214286	0.142857	0.357143	0.264929	0.7235	3	10.33333	0.333333	0.333333	2.666667	0.333333	3.943333
9	20190713	SK	1	1	1	0.4	0.3	0	0.2	0.2696	0.7241	1.66	7.6	0.4	0.6	1.2	0.2	1.936
10	20190504	NC	1	1	0.615385	0.307692	0.153846	0.076923	0.461538	0.251923	0.699923	2.25	7.75	0.25	0.25	1.75	0	2.5475
11	20190712	KIA	1	1	0.777778	0.555556	0.333333	0	0.444444	0.295222	0.795111	2.766667	11	0	1	2.333333	0	2.976667
12	20190509	한화	1	0	0.909091	0.545455	0.181818	0	0.909091	0.238545	0.681091	2.075	10.25	0.25	0.75	1.5	0	2.7775
13	20190830	두산	1	0	0.8	0.3	0.3	0.1	0.5	0.254	0.6875	2.075	8.75	0.25	1	1.5	0	3.395
14	20190818	KIA	0	1	0.692308	0.076923	0.153846	0	0.769231	0.263	0.696538	3	13	0.666667	2.666667	2.333333	0	3.863333
15	20190830	롯데	0	0	0.4	0.1	0.2	0	0.7	0.259	0.695	1.825	9.25	1.25	0.5	1.75	0.25	5.465
16	20190420	NC	0	0	0.909091	0.363636	0.090909	0.090909	0.636364	0.278636	0.794545	2.433333	13.66667	3.333333	0.333333	1.666667	1	6.656667
17	20190930	한화	0	1	0.692308	0.153846	0	0	0.384615	0.248846	0.662	1.66	8.2	1.2	1.2	1.8	0.2	4.924
18	20191001	NC	0	0	0.944444	0.222222	0.111111	0	0.388889	0.275944	0.734333	0.744444	4.666667	0.555556	0.111111	0.666667	0	3.81
19	20190406	롯데	1	1	0.727273	0.727273	0.454545	0	0.272727	0.247545	0.694818	1.8	8.2	1.4	0.6	2.6	0.2	3.652
20	20190601	키움	1	0	0.916667	0.333333	0.5	0.083333	0.5	0.27625	0.737833	1.266667	6.5	0	0.666667	1.333333	0	3.243333
21	20190707	SK	0	0	0.692308	0.230769	0.307692	0	0.384615	0.248	0.685769	1.85	8.333333	0.833333	0.833333	1.833333	0.166667	2.813333
22	20190706	두산	0	1	0.545455	0.181818	0.090909	0	0.545455	0.268545	0.732091	3	12.66667	1.333333	1	1	0	2.823333

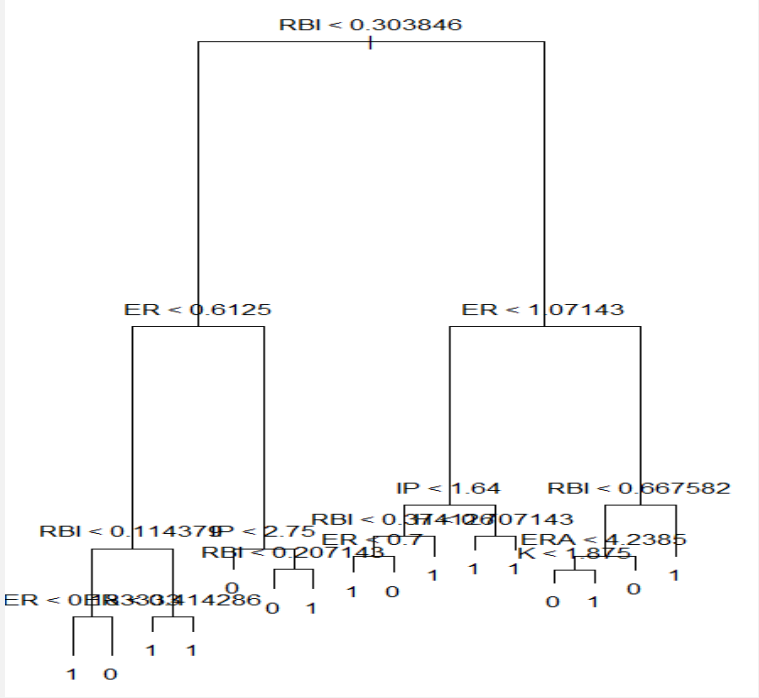
```
> ba <- ba[,-c(1,2)]
> ba$win=factor(ba$win)
> ba$ha=factor(ba$ha)
> summary(ba)
   win       ha            H                 RBI                BB
 0   :694   0:693   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1   :691   1:694   1st Qu.:0.5420   1st Qu.:0.1538   1st Qu.:0.1538
 NA's:  2           Median :0.7273   Median :0.3077   Median :0.2500
                    Mean   :0.7499   Mean   :0.3518   Mean   :0.2714
                    3rd Qu.:0.9231   3rd Qu.:0.5000   3rd Qu.:0.3636
                    Max.   :1.9167   Max.   :1.5833   Max.   :1.0909
      HBP                SO                AVG                OPS
 Min.   :0.00000   Min.   :0.0000   Min.   :0.08333   Min.   :0.2928
 1st Qu.:0.00000   1st Qu.:0.3846   1st Qu.:0.24177   1st Qu.:0.6562
 Median :0.00000   Median :0.5385   Median :0.25890   Median :0.6985
 Mean   :0.04020   Mean   :0.5534   Mean   :0.25523   Mean   :0.6932
 3rd Qu.:0.07692   3rd Qu.:0.7000   3rd Qu.:0.27204   3rd Qu.:0.7360
 Max.   :0.33333   Max.   :1.6000   Max.   :0.41509   Max.   :1.0351
       IP                TBF               ER                BB.1
 Min.   :0.7444   Min.   : 4.250   Min.   :0.000   Min.   :0.0000
 1st Qu.:1.6171   1st Qu.: 7.750   1st Qu.:0.400   1st Qu.:0.5000
 Median :2.0000   Median : 9.000   Median :0.800   Median :0.7143
 Mean   :2.1886   Mean   : 9.843   Mean   :0.974   Mean   :0.7779
 3rd Qu.:2.4333   3rd Qu.:10.775   3rd Qu.:1.333   3rd Qu.:1.0000
 Max.   :9.0000   Max.   :34.000   Max.   :4.667   Max.   :4.0000
       K                HR                ERA
 Min.   : 0.000   Min.   :0.0000   Min.   : 0.000
 1st Qu.: 1.000   1st Qu.:0.0000   1st Qu.: 3.071
 Median : 1.500   Median :0.0000   Median : 3.777
 Mean   : 1.723   Mean   :0.1687   Mean   : 4.127
 3rd Qu.: 2.000   3rd Qu.:0.2500   3rd Qu.: 4.863
```

- 데이터 셋을 불러와 date와 team을 제외한 모든 요소를 Attribute로 가짐.
- win(승/패)과 ha(홈/어웨이)요소는 factor를 통해 범주형 변수로 바꿔준다.
  -> 승리 팀 : 691팀, 패배 팀 : 694팀, NA's : 승패를 모르는 두 팀
  -> 홈 : 693팀, 어웨이 팀 694팀

4. 데이터 마이닝 모형화 - Decision Tree -

〈의사결정나무 형성〉

```
train <- D[1:831,]
test <- D[832:1385,]
summary(D)
```



- **1385개의 데이터 셋**

- train_data -> 60%
  -> 1385 * 0.6 = 831

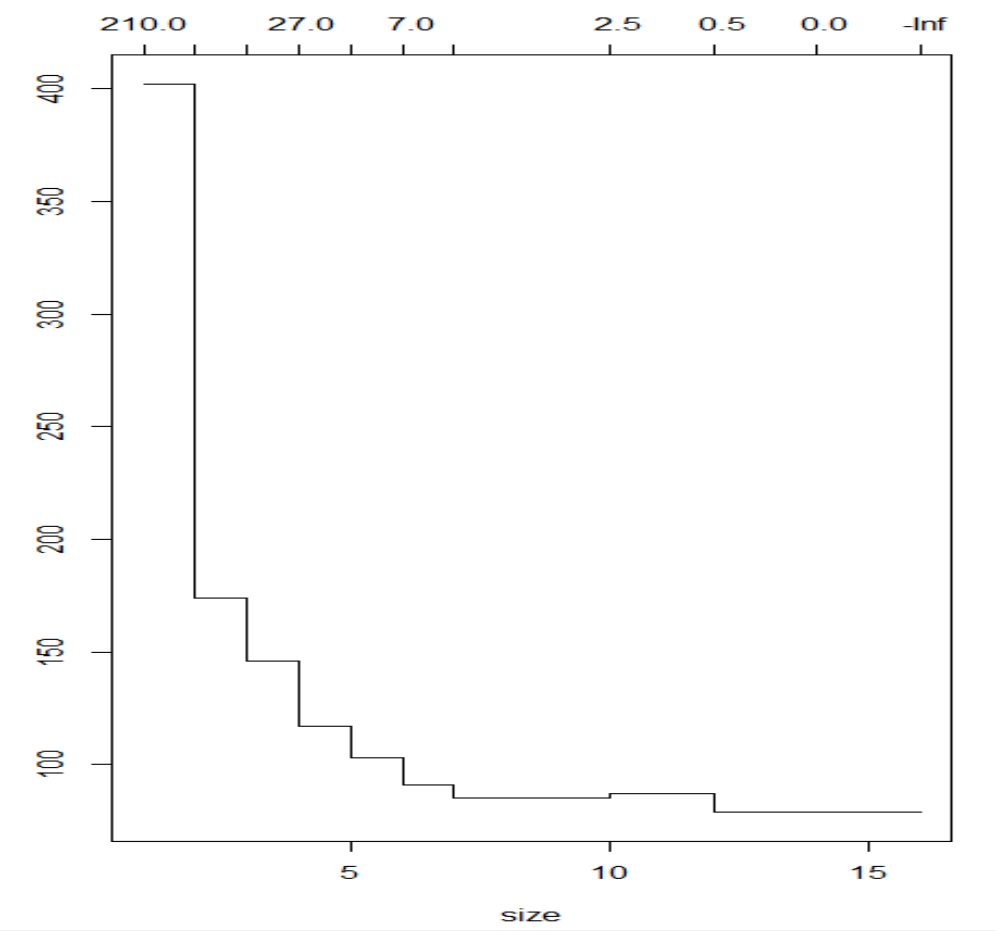
- test_data -> 40%

Tree 패키지를 이용하여 의사결정 나무 형성

```
> library(tree)
> treemod=tree(win~., data=train)
> plot(treemod)
> text(treemod)
```

4. 데이터 마이닝 모형화 - Decision Tree -

<가지치기(PRUNING)>



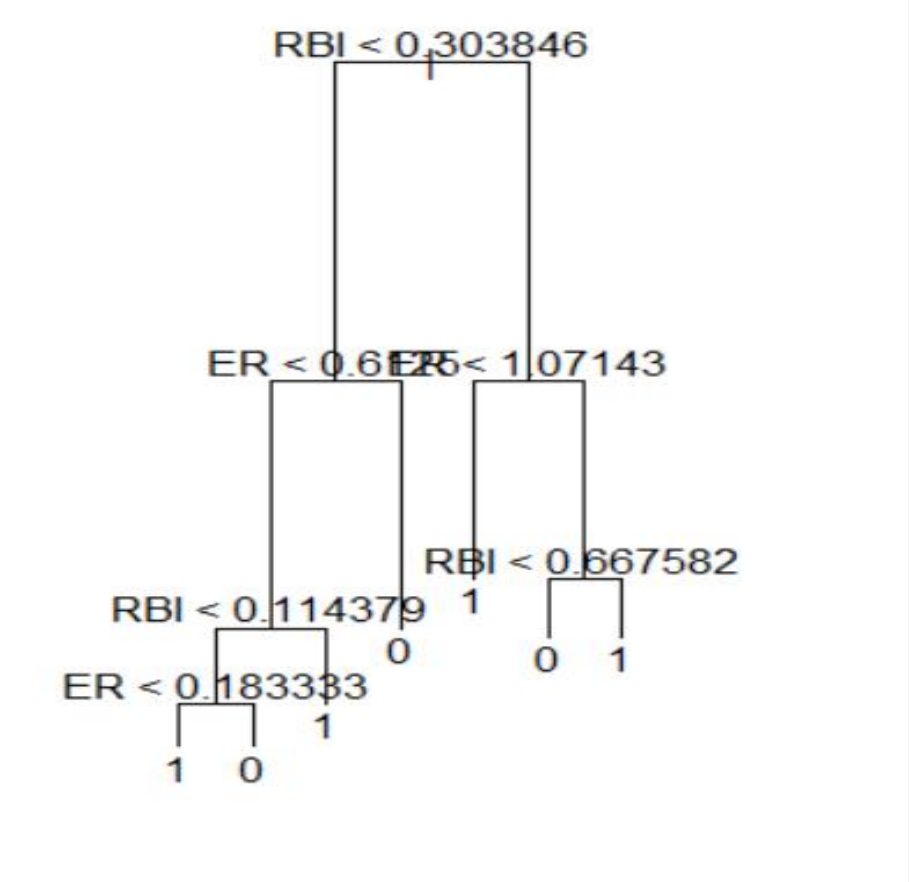
Overfitting 문제를 해결하기 위해 Pruning 단계 진행

CV그래프를 통해 분산이 낮은 6-10개 가지 중 7개로 설정
(총 가지 수가 15개 이기 때문에 10-15개는 무의미)

```
> cv.trees=cv.tree(treemod, FUN=prune.misclass)
> plot(cv.trees)
```

4. 데이터 마이닝 모형화 - Decision Tree -

〈최종 의사 결정 나무 모형〉




```
> prune.trees=prune.misclass(treemod, best = 7)
> plot(prune.trees)
> text(prune.trees,pretty=0)
```

4. 데이터 마이닝 모형화 - Decision Tree -

<예측하기 & 모델 평가>

```
> confusionMatrix(treepred, test$win)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 256  15
         1  36 246

               Accuracy : 0.9078
                 95% CI : (0.8805, 0.9306)
    No Information Rate : 0.528
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8158

 Mcnemar's Test P-Value : 0.005101

            Sensitivity : 0.8767
            Specificity : 0.9425
         Pos Pred Value : 0.9446
         Neg Pred Value : 0.8723
             Prevalence : 0.5280
         Detection Rate : 0.4629
   Detection Prevalence : 0.4901
      Balanced Accuracy : 0.9096

       'Positive' Class : 0
```

정확도 : 90.78%로 비교적 정확하게 승패 여부를 예측한다.

4. 데이터 마이닝 모형화 - Decision Tree -

NC vs LG

<데이터 셋>

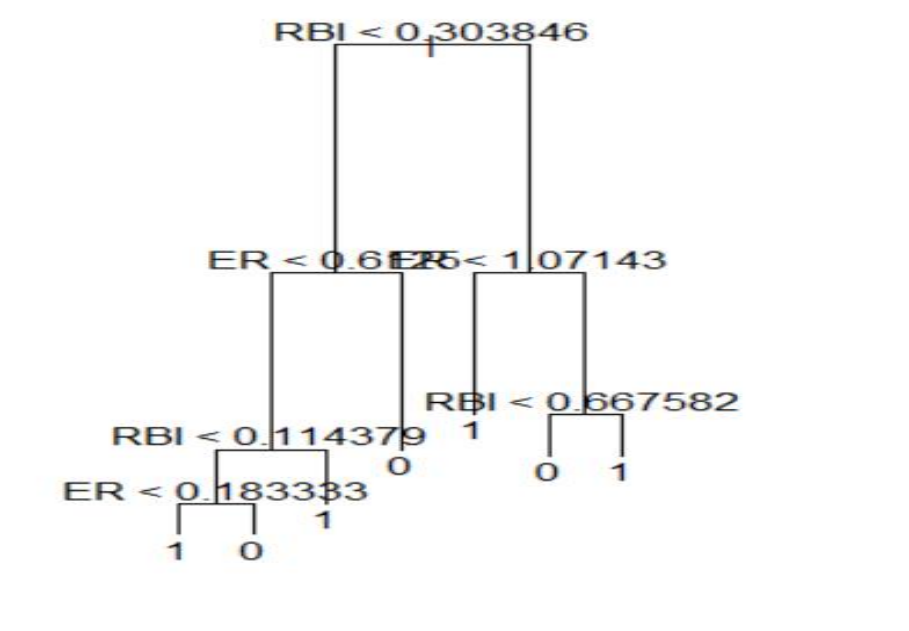
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	date	team	win	ha	H	RBI	BB	HBP	SO	AVG	OPS	IP	TBF	ER	BB	K	HR	ERA
1387	20191003	NC		0	0.773405	0.354252	0.25967	0.046723	0.50877	0.265679	0.736989	2.224378	9.969236	0.923276	0.755193	1.760543	0.190605	3.972512
1388	20191003	LG		1	0.729368	0.331821	0.240571	0.042827	0.536832	0.240214	0.639584	2.196358	9.936348	0.960295	0.842874	1.712958	0.157443	3.205396

```
> predict_data=D[1386:1387,]
> predict_data
      win ha         H       RBI         BB        HBP        SO       AVG
1386 <NA>  0 0.7734054 0.3542516 0.2596701 0.04672261 0.5087697 0.2656792
1387 <NA>  1 0.7293679 0.3318214 0.2405708 0.04282742 0.5368319 0.2402142
           OPS       IP      TBF        ER      BB.1        K        HR
1386 0.7369891 2.224378 9.969236 0.9232757 0.7551930 1.760543 0.1906047
1387 0.6395842 2.196358 9.936348 0.9602946 0.8428743 1.712958 0.1574426
          ERA
1386 3.972512
1387 3.205396
> pred=predict(prune.trees, predict_data,type='class')
> pred
[1] 1 1
Levels: 0 1
```

- 두 모형 모두 의사결정나무에 의해 승리로 예측

4. 데이터 마이닝 모형화 - Decision Tree -

〈한계〉



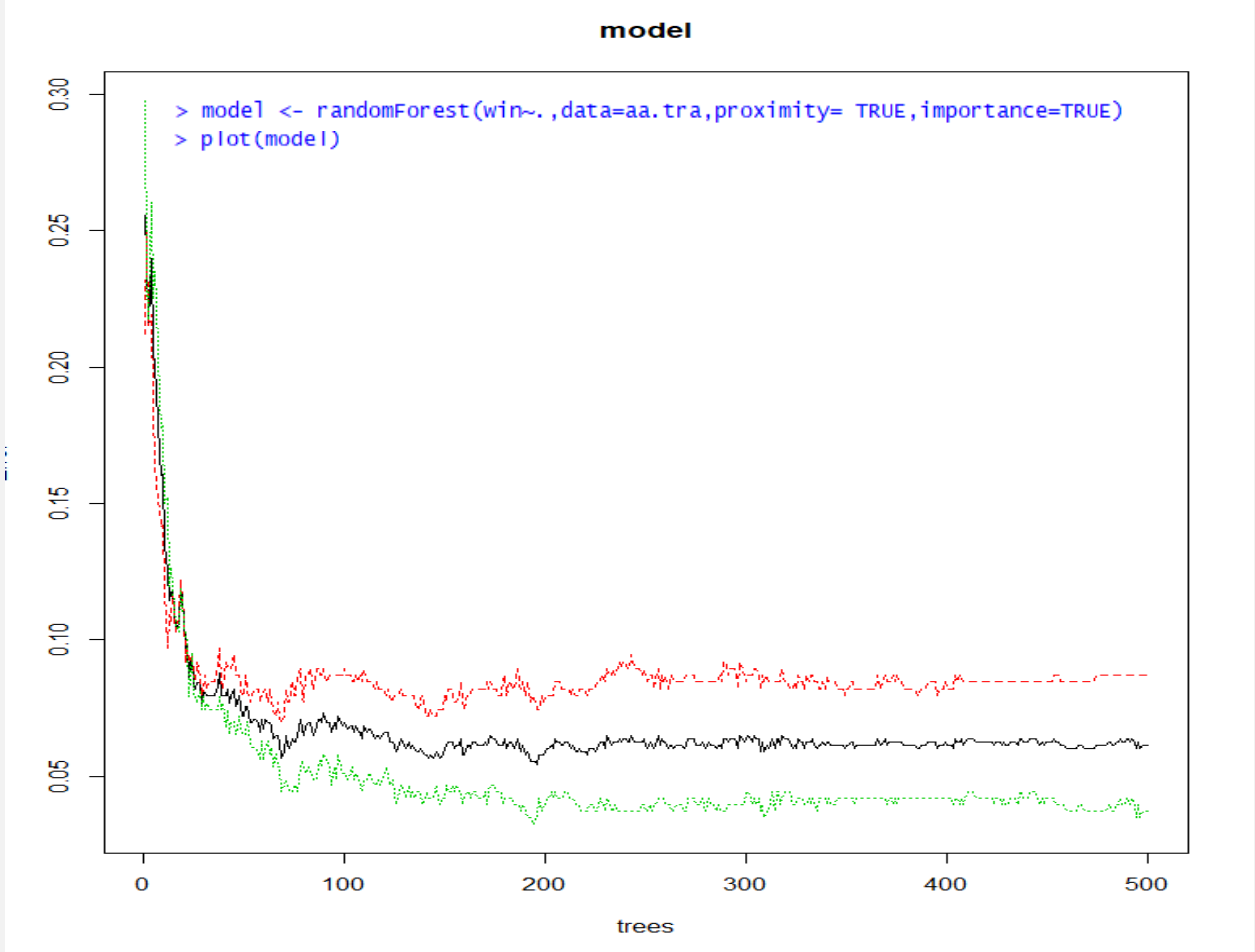

```
> predict_data=D[1386:1387,]
> predict_data
      win ha         H       RBI        BB        HBP        SO       AVG
1386 <NA>  0 0.7734054 0.3542516 0.2596701 0.04672261 0.5087697 0.2656792
1387 <NA>  1 0.7293679 0.3318214 0.2405708 0.04282742 0.5368319 0.2402142
           OPS       IP      TBF        ER      BB.1        K        HR
1386 0.7369891 2.224378 9.969236 0.9232757 0.7551930 1.760543 0.1906047
1387 0.6395842 2.196358 9.936348 0.9602946 0.8428743 1.712958 0.1574426
          ERA
1386 3.972512
1387 3.205396
> pred=predict(prune.trees, predict_data,type='class')
> pred
[1] 1 1
Levels: 0 1
```

노드값이 RBI, ER로 반복적으로 나타난다.

두 팀 모두 조건을 만족시킬 시 승패 예측이 불가능해진다.

4. 데이터 마이닝 모형화 - Random Forest-

모든 예측모형 같은 방식으로 변수 가공,데이터 분리



1) 최적의 트리 개수 선정

- 오류가 최소화 되는 트리 개수를 알아 보기 위해 plot생성
- 트리개수가 약 200개에 못 미칠 때 오류가 최소인 것으로 추정
- 이후에는 트리 개수가 많아져도 오차율 개선에 특별히 도움되지 않음

4. 데이터 마이닝 모형화 - Random Forest-

```
> model2 <- randomForest(win~.,data=aa.tra, ntree = which.min(model$err.rate[, 1]),proximity= TRUE,importance=TRUE)
> model2

Call:
 randomForest(formula = win ~ ., data = aa.tra, ntree = which.min(model$err.rate[,      1]), proximity = TRUE, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 196
No. of variables tried at each split: 3

        OOB estimate of  error rate: 6.5%
Confusion matrix:
    0   1 class.error
0 369  33  0.08208955
1  21 408  0.04895105
```

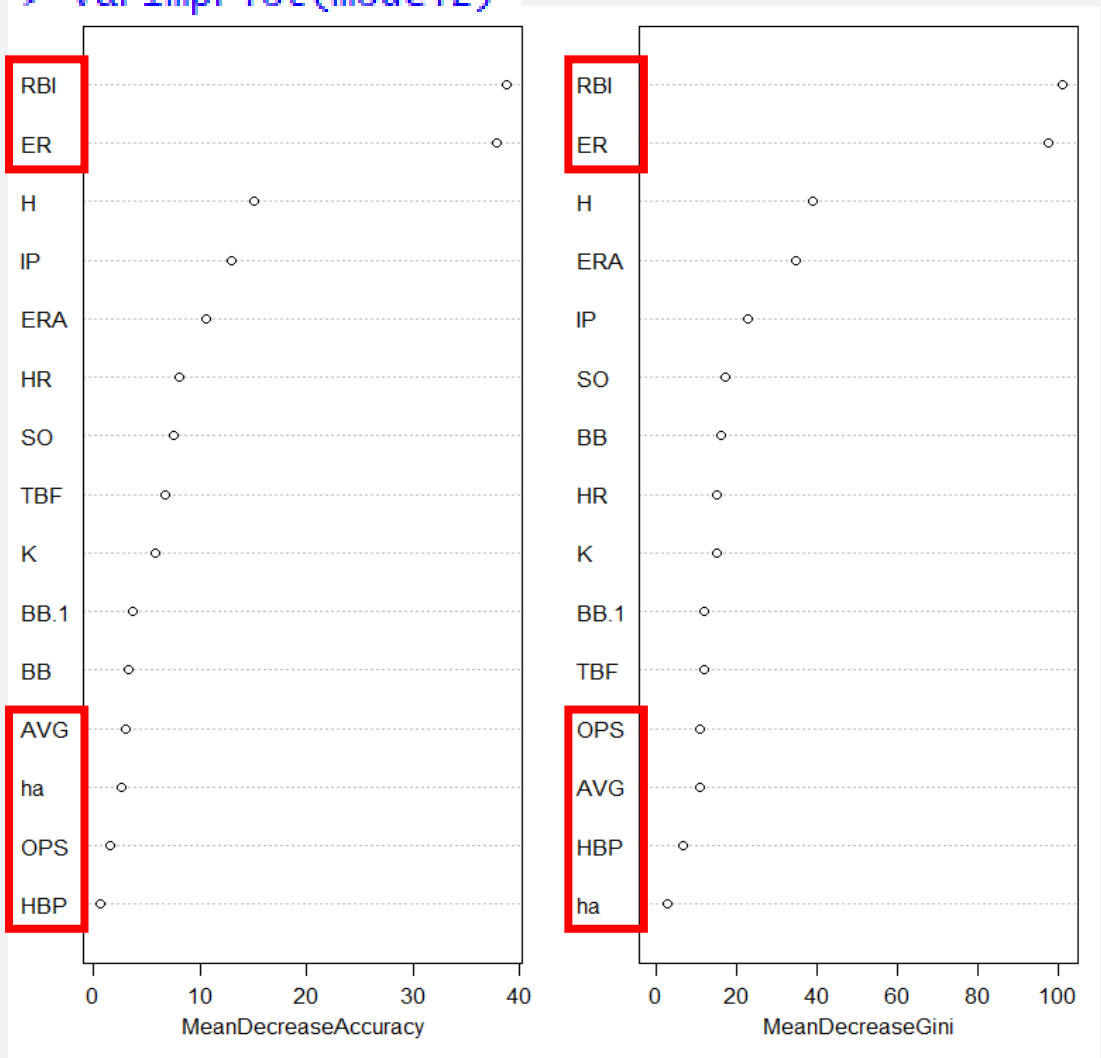
```
> pre <- predict(model2,aa.test)
> t<-table(pre,aa.test$win)
> t

pre   0   1
  0 271   8
  1  21 254
> (t[1,1]+t[2,2])/sum(t)
[1] 0.9476534
```

2) 모델생성

- 오류가 최소가 되는 ntree 개수의 최소값 -> 196개 사용
- 정확도 94.76%

4. 데이터 마이닝 모형화 - Random Forest-

```
> varImpPlot(model2)
```



3) 요인 분석

주요 변인 알아보기 위해 Importance plot 생성

- 모델 정확도와 노드 불순도 개선에 타자의 타점(RBI) 과 투수의 자책점(ER)이 가장 많이 기여

- 타자와 투수 중 어느 쪽의 영향이 더 큰지는 확실히 판단 할 수 없음

- 타격 지표로 널리 쓰이는 OPS(출루율+장타율)의 영향 전체 변수 중 하위권

4. 데이터 마이닝 모형화 - Random Forest-

NC vs LG

<데이터 셋>

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	date	team	win	ha	H	RBI	BB	HBP	SO	AVG	OPS	IP	TBF	ER	BB	K	HR	ERA
1387	20191003	NC		0	0.773405	0.354252	0.25967	0.046723	0.50877	0.265679	0.736989	2.224378	9.969236	0.923276	0.755193	1.760543	0.190605	3.972512
1388	20191003	LG		1	0.729368	0.331821	0.240571	0.042827	0.536832	0.240214	0.639584	2.196358	9.936348	0.960295	0.842874	1.712958	0.157443	3.205396

```
> predict_data <- aa[1386:1387,]
> win_predict <- predict(model2, predict_data, type="response")
> win_predict
1386 1387
   1    1
Levels: 0 1
```

4) 결과 예측

- 정규시즌 평균을 토대로 와일드카드전의 승패 예측

- 각 팀의 승리 확률을 계산하여 비교 할 수 없으므로 두 팀의 예측 결과가 같게 나올 경우 승패를 예측할 수 없음

- 두 팀 모두 승리로 예측

4. 데이터 마이닝 모형화 - Logistic Regression -

```
> # 데이터 1385개중 60%의 train 데이터
> train_data <- ba[1:831,]
> # 데이터 1385개중 40%의 test데이터
> test_data <- ba[832:1385,]
> logit.ba=glm(win~., family=binomial, data=train_data)
> logit.ba

Call:  glm(formula = win ~ ., family = binomial, data = train_data)

Coefficients:
(Intercept)          ha1            H          RBI           BB
     5.4088       1.1399       0.7588      21.3446      -0.3785
        HBP           SO          AVG          OPS           IP
     0.9824      -3.2757      36.1797     -22.2675       3.5424
        TBF           ER         BB.1            K           HR
    -0.6698      -8.0436       0.6315       0.4739       0.2632
        ERA
    -0.3045
```

- **1385개의 데이터 셋**

- train_data -> 60%
  -> 1385 * 0.6 = 831

- test_data -> 40%

BB : 타자
BB.1 : 투수

Logit = 5.4088 + 1.1399*ha + 0.7588*H + 21.3446*RBI + (-0.3785*BB) + 0.9824*HBP + (-3.2757*SO) + 36.1797*AVG + (-22.2675*OPS) + 3.5424*IP + (-0.6698*TBF) + (-8.0436*ER) + 0.6315*BB.1 + 0.4739*K + (-0.2632*HR) + (-0.3045*ERA)

4. 데이터 마이닝 모형화 - Logistic Regression -

<유의성 검사>

```
> summary(logit.ba)

Call:
glm(formula = win ~ ., family = binomial, data = train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9712  -0.0333   0.0002   0.0764   2.1607

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   5.4088     2.1513   2.514  0.01193 *
ha1           1.1399     0.3957   2.881  0.00397 **
H             0.7588     1.0400   0.730  0.46560
RBI          21.3446     2.3337   9.146  < 2e-16 ***
BB           -0.3785     1.1169  -0.339  0.73473
HBP           0.9824     2.8617   0.343  0.73139
SO           -3.2757     0.8867  -3.694  0.00022 ***
AVG          36.1797    18.7460   1.930  0.05361 .
OPS         -22.2675     7.1891  -3.097  0.00195 **
IP            3.5424     0.8379   4.228  2.36e-05 ***
TBF          -0.6698     0.2474  -2.707  0.00678 **
ER           -8.0436     0.9062  -8.876  < 2e-16 ***
BB.1          0.6315     0.4804   1.315  0.18864
K             0.4739     0.2645   1.792  0.07316 .
HR            0.2632     1.0150   0.259  0.79537
ERA          -0.3045     0.1639  -1.858  0.06314 .
---
```

p-value의 절댓값이 0.05 이하인 ha, RBI, SO, AVG, OPS, IP, ER, K값이 유의한 것으로 나타났다.

4. 데이터 마이닝 모형화 - Logistic Regression -

1) 유의성 검사와 상관없이 분석, 예측할 때

```
> coef(logit.ba)
(Intercept)         ha1           H         RBI          BB         HBP
  5.4088124   1.1398888   0.7588295  21.3445924  -0.3784719   0.9823519
         SO         AVG         OPS          IP         TBF          ER
 -3.2757479  36.1796775 -22.2674578   3.5424204  -0.6698363  -8.0435575
       BB.1           K          HR         ERA
  0.6315002   0.4738737   0.2632278  -0.3044904
> exp(coef(logit.ba))
 (Intercept)          ha1            H          RBI           BB
2.233662e+02 3.126421e+00 2.135775e+00 1.861396e+09 6.849072e-01
         HBP           SO          AVG          OPS           IP
2.670730e+00 3.778860e-02 5.159817e+15 2.134843e-10 3.455044e+01
         TBF           ER         BB.1            K           HR
5.117924e-01 3.211644e-04 1.880429e+00 1.606204e+00 1.301123e+00
         ERA
7.374991e-01
```

ha가 1이고(홈) H, RBI, HBP, AVG, IP, BB(투수), K가 클수록 BB(타자) SO, OPS, TBF, ER, HR, ERA가 작을수록 승리할 확률이 높아진다는 것을 알 수 있다.

```
> group = win[832:1385]
> logitpred = (predict(logit.ba,test_data,type="response")>=0.5)
> ba_table = table (logitpred, group)
> ba_table
         group
logitpred   0   1
    FALSE 278  18
    TRUE   14 244
```

<Cut-off를 0.5로 잡았을 경우>

예측 정확도 => (278+244) / (278+18+14+244) = 94.22%

4. 데이터 마이닝 모형화 - Logistic Regression -

2) 유의성 검사 결과를 가지고 분석, 예측할 때

```
> summary(logit.ba)

Call:
glm(formula = win ~ ., family = binomial, data = train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9712  -0.0333   0.0002   0.0764   2.1607

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   5.4088     2.1513   2.514  0.01193 *
ha1           1.1399     0.3957   2.881  0.00397 **
H             0.7588     1.0400   0.730  0.46560
RBI          21.3446     2.3337   9.146  < 2e-16 ***
BB           -0.3785     1.1169  -0.339  0.73473
HBP           0.9824     2.8617   0.343  0.73139
SO           -3.2757     0.8867  -3.694  0.00022 ***
AVG          36.1797    18.7460   1.930  0.05361 .
OPS         -22.2675     7.1891  -3.097  0.00195 **
IP            3.5424     0.8379   4.228  2.36e-05 ***
TBF          -0.6698     0.2474  -2.707  0.00678 **
ER           -8.0436     0.9062  -8.876  < 2e-16 ***
BB.1          0.6315     0.4804   1.315  0.18864
K             0.4739     0.2645   1.792  0.07316 .
HR            0.2632     1.0150   0.259  0.79537
ERA          -0.3045     0.1639  -1.858  0.06314 .
---
```

```
> logit.ba=glm(win~ha+RBI+SO+OPS+IP+TBF+ER, family=binomial, data=train_data)
> summary(logit.ba)

Call:
glm(formula = win ~ ha + RBI + SO + OPS + IP + TBF + ER, family = binomial,
    data = train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9362  -0.0505   0.0003   0.0818   2.4985

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.6970     1.9463   2.413 0.015810 *
ha1           0.9750     0.3715   2.624 0.008678 **
RBI          21.0303     2.0596  10.211  < 2e-16 ***
SO           -2.7855     0.8046  -3.462 0.000536 ***
OPS          -8.9030     2.6048  -3.418 0.000631 ***
IP            3.3704     0.7472   4.510 6.47e-06 ***
TBF          -0.5108     0.2124  -2.405 0.016192 *
ER           -7.7987     0.8074  -9.659  < 2e-16 ***
```

P-value의 절댓값이 0.05이하인 ha, RBI, SO, OPS, IP, TBF, ER 값만을 가지고 데이터 분석

- logit.ba=glm(win~ha+RBI+SO+OPS+IP+TBF+ER, family=binomial, data=train_data)
- 유의한 속성만을 가지고 test_data 예측

4. 데이터 마이닝 모형화 - Logistic Regression -

```
> coef(logit.ba)
(Intercept)         ha1         RBI          SO         OPS          IP
  4.6970039   0.9750058  21.0302788  -2.7854784  -8.9029906   3.3703907
        TBF          ER
 -0.5107556  -7.7987263
> exp(coef(logit.ba))
 (Intercept)          ha1          RBI           SO          OPS
1.096183e+02 2.651183e+00 1.359359e+09 6.169957e-02 1.359817e-04
          IP          TBF           ER
2.908989e+01 6.000420e-01 4.102572e-04
```

ha가 1(홈)일수록, RBI, IP가 클수록 SO, OPS, TBF, ER이 작을수록 승리 확률이 높아진다.

```
> group = win[832:1385]
> logitpred = (predict(logit.ba,test_data,type="response")>=0.5)
> ba_table = table (logitpred, group)
> ba_table
         group
logitpred   0   1
    FALSE 275  15
    TRUE   17 247
```

<Cut-off를 0.5로 잡았을 경우>

➔ 예측 정확도 : (275+247) / (275+15+17+247)

= 94.22%

유의성을 고려하지 않고 Logistic Regression분석을 실시 하였을 경우와 동일한 예측 값을 가진다.

4. 데이터 마이닝 모형화 - Logistic Regression -

NC vs LG

<데이터 셋>

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	date	team	win	ha	H	RBI	BB	HBP	SO	AVG	OPS	IP	TBF	ER	BB	K	HR	ERA
1387	20191003	NC		0	0.773405	0.354252	0.25967	0.046723	0.50877	0.265679	0.736989	2.224378	9.969236	0.923276	0.755193	1.760543	0.190605	3.972512
1388	20191003	LG		1	0.729368	0.331821	0.240571	0.042827	0.536832	0.240214	0.639584	2.196358	9.936348	0.960295	0.842874	1.712958	0.157443	3.205396

- 모든 Attribute 사용

```
> predict_data <- ba[1386:1387,]
> logitpred_prob = predict(logit.ba, predict_data, type="response")
> logitpred_prob
     1386      1387
0.2624356 0.6538842
```

- NC가 승리할 확률 : 26.24%
- LG가 승리할 확률 : 65.39%

=> LG가 NC보다 승리할 확률이 더 높다.

- 유의성을 가진 Attribute만을 사용

```
> predict_data <- ba[1386:1387,]
> logitpred_prob = predict(logit.ba, predict_data, type="response")
> logitpred_prob
     1386      1387
0.3482473 0.5742530
```

- NC가 승리할 확률 : 34.82%
- LG가 승리할 확률 : 57.42%

=> LG가 NC보다 승리할 확률이 더 높다.

4. 데이터 마이닝 모형화 - Artificial Neural Network -

```
> Dnnet <- nnet(win ~ ., data = train, size = 8, rang = 0.1, decay = 5e-4, maxit = 100)
# weights:  137
initial  value 580.593617
iter  10 value 377.994081
iter  20 value 99.036573
iter  30 value 84.396654
iter  40 value 77.363539
iter  50 value 73.689078
iter  60 value 70.883936
iter  70 value 69.729201
iter  80 value 68.101478
iter  90 value 62.168560
iter 100 value 53.185598
final  value 53.185598
stopped after 100 iterations
> Dnnet
a 15-8-1 network with 137 weights
inputs: ha1 H RBI BB HBP SO AVG OPS IP TBF ER BB.1 K HR ERA
output(s): win
options were - entropy fitting  decay=5e-04
```

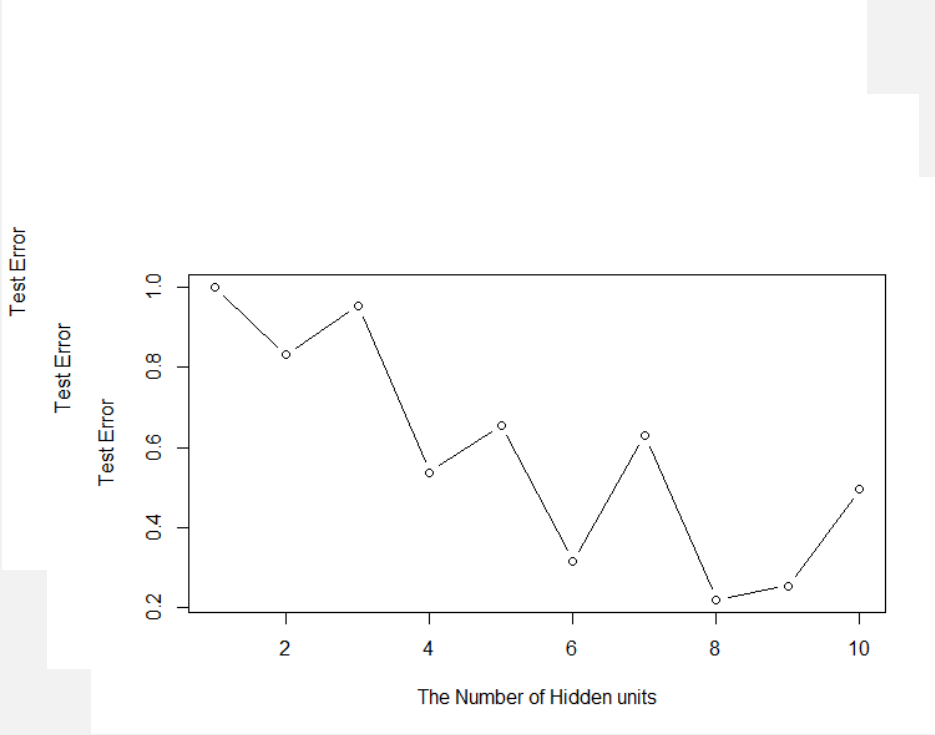
- **1385개의 데이터 셋**

- train_data -> 60%
  -> 1385 * 0.6 = 831

- test_data -> 40%

4. 데이터 마이닝 모형화 - Artificial Neural Network -

<Hidden Node의 크기 결정>



- hidden node수에 따른 test 자료의 오차를 알아보기 위한 그래프
- 여러 번 그려본 다음 공통되게 오차가 작게 나온 부분의 hidden node 개수로 정함.
- 적합한 Hidden Node 의 수 = 8

```
test.err=function(h.size,maxit0){
  Dnnet1 = nnet(win ~ ., data = train, size = h.size)
  y = test$win
  p = predict(Dnnet1,test)
  err = mean(y!=p)
  c(h.size,err)
}
out = t(sapply(1:10,FUN = test.err,maxit0=200))
plot(out,type='b',xlab="The Number of Hidden units",ylab = "Test
Error")
```

4. 데이터 마이닝 모형화 - Artificial Neural Network -

- 인공신경망 모형 해석

```
> summary(Dnnet)
a 15-8-1 network with 137 weights
options were - entropy fitting
  b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->h1 i10->h1 i11->h1 i12->h1
   2.17    0.67   13.98 -122.41  -13.87   -4.00    9.67   -3.93   -8.03  -13.87    3.23   27.67   -2.12
i13->h1 i14->h1 i15->h1
  -2.11    3.47    1.03
  b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9->h2 i10->h2 i11->h2 i12->h2
   0.81   -0.29   -0.39   -0.54    0.36   -0.29    0.36   -0.65   -0.37    0.49    1.17    0.65    0.37
i13->h2 i14->h2 i15->h2
   0.69   -0.29    1.01
  b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3  i9->h3 i10->h3 i11->h3 i12->h3
 -39.51   19.50   -6.47  -11.99   22.64    9.68   18.92    5.25   25.46  -34.74  -17.26   63.66  -39.37
i13->h3 i14->h3 i15->h3
  16.84    3.10   38.17
  b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4  i9->h4 i10->h4 i11->h4 i12->h4
  -0.19   -0.33    0.29    0.07    0.55    0.13   -0.15    0.01    0.36   -1.16   -3.29   -0.37   -0.25
i13->h4 i14->h4 i15->h4
  -1.05   -0.07   -1.33
  b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5  i9->h5 i10->h5 i11->h5 i12->h5
  15.33  -12.28   46.27  108.35  -56.99  -22.49  -62.80   -2.72  -10.63   69.83  -23.74   43.08  -38.14
i13->h5 i14->h5 i15->h5
  -7.02    2.52   15.36
  b->h6  i1->h6  i2->h6  i3->h6  i4->h6  i5->h6  i6->h6  i7->h6  i8->h6  i9->h6 i10->h6 i11->h6 i12->h6
   0.40   -0.21   -0.06   -0.10    0.47   -0.21    0.64   -0.37    0.52    0.61    0.95    0.85   -0.36
i13->h6 i14->h6 i15->h6
  -0.47    0.07    1.20
  b->h7  i1->h7  i2->h7  i3->h7  i4->h7  i5->h7  i6->h7  i7->h7  i8->h7  i9->h7 i10->h7 i11->h7 i12->h7
  -0.98   -2.60   -3.47   -3.75   -0.36    0.17    0.45   -0.93   -0.90   -1.14   -0.88    3.73    0.65
i13->h7 i14->h7 i15->h7
  -1.47    1.25   -1.87
  b->h8  i1->h8  i2->h8  i3->h8  i4->h8  i5->h8  i6->h8  i7->h8  i8->h8  i9->h8 i10->h8 i11->h8 i12->h8
  59.99  -23.15  -16.51 -160.22   32.13   33.52   31.94   58.06   18.43  -26.62  -15.11  121.21  -12.51
i13->h8 i14->h8 i15->h8
  -2.74   -9.69    4.66
   b->o   h1->o   h2->o   h3->o   h4->o   h5->o   h6->o   h7->o   h8->o
   2.32  -73.59    2.05   -4.54   -2.95   69.92    0.10    0.23 -147.39
```

i□은 □번째 input node

h□은 □번째 hidden node

o□은 □번째 output node

밑의 값은 가중치

20이 넘으면 영향력이 크다고 판단

3, 4, 6, 9, 11, 12 input node가 다른 변수들에 비해 영향력이 큼

4. 데이터 마이닝 모형화 - Artificial Neural Network -

- 인공신경망 모형 해석

```
> summary(Dnnet)
a 15-8-1 network with 137 weights
options were - entropy fitting
  b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->h1 i10->h1 i11->h1 i12->h1
   2.17    0.67   13.98 -122.41  -13.87   -4.00    9.67   -3.93   -8.03  -13.87    3.23   27.67   -2.12
i13->h1 i14->h1 i15->h1
  -2.11    3.47    1.03
  b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9->h2 i10->h2 i11->h2 i12->h2
   0.81   -0.29   -0.39   -0.54    0.36   -0.29    0.36   -0.65   -0.37    0.49    1.17    0.65    0.37
i13->h2 i14->h2 i15->h2
   0.69   -0.29    1.01
  b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3  i9->h3 i10->h3 i11->h3 i12->h3
 -39.51   19.50   -6.47  -11.99   22.64    9.68   18.92    5.25   25.46  -34.74  -17.26   63.66  -39.37
i13->h3 i14->h3 i15->h3
  16.84    3.10   38.17
  b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4  i9->h4 i10->h4 i11->h4 i12->h4
  -0.19   -0.33    0.29    0.07    0.55    0.13   -0.15    0.01    0.36   -1.16   -3.29   -0.37   -0.25
i13->h4 i14->h4 i15->h4
  -1.05   -0.07   -1.33
  b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5  i9->h5 i10->h5 i11->h5 i12->h5
  15.33  -12.28   46.27  108.35  -56.99  -22.49  -62.80   -2.72  -10.63   69.83  -23.74   43.08  -38.14
i13->h5 i14->h5 i15->h5
  -7.02    2.52   15.36
  b->h6  i1->h6  i2->h6  i3->h6  i4->h6  i5->h6  i6->h6  i7->h6  i8->h6  i9->h6 i10->h6 i11->h6 i12->h6
   0.40   -0.21   -0.06   -0.10    0.47   -0.21    0.64   -0.37    0.52    0.61    0.95    0.85   -0.36
i13->h6 i14->h6 i15->h6
  -0.47    0.07    1.20
  b->h7  i1->h7  i2->h7  i3->h7  i4->h7  i5->h7  i6->h7  i7->h7  i8->h7  i9->h7 i10->h7 i11->h7 i12->h7
  -0.98   -2.60   -3.47   -3.75   -0.36    0.17    0.45   -0.93   -0.90   -1.14   -0.88    3.73    0.65
i13->h7 i14->h7 i15->h7
  -1.47    1.25   -1.87
  b->h8  i1->h8  i2->h8  i3->h8  i4->h8  i5->h8  i6->h8  i7->h8  i8->h8  i9->h8 i10->h8 i11->h8 i12->h8
  59.99  -23.15  -16.51 -160.22   32.13   33.52   31.94   58.06   18.43  -26.62  -15.11  121.21  -12.51
i13->h8 i14->h8 i15->h8
   2.74    9.69    4.66
   b->o   h1->o   h2->o   h3->o   h4->o   h5->o   h6->o   h7->o   h8->o
   2.32  -73.59    2.05   -4.54   -2.95   69.92    0.10    0.23 -147.39
```

- H1->O는 음수, H5->O는 양수,

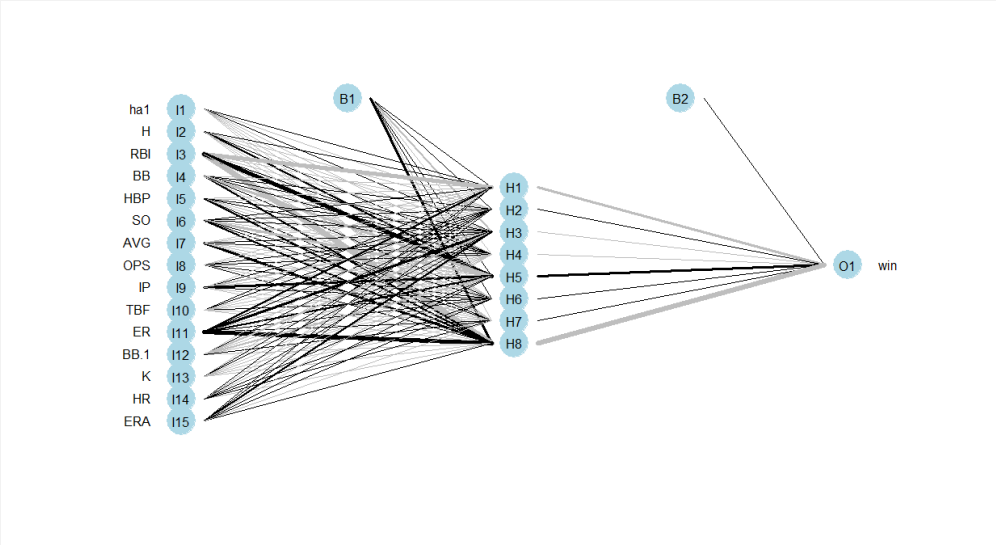
  H8->O는 음수

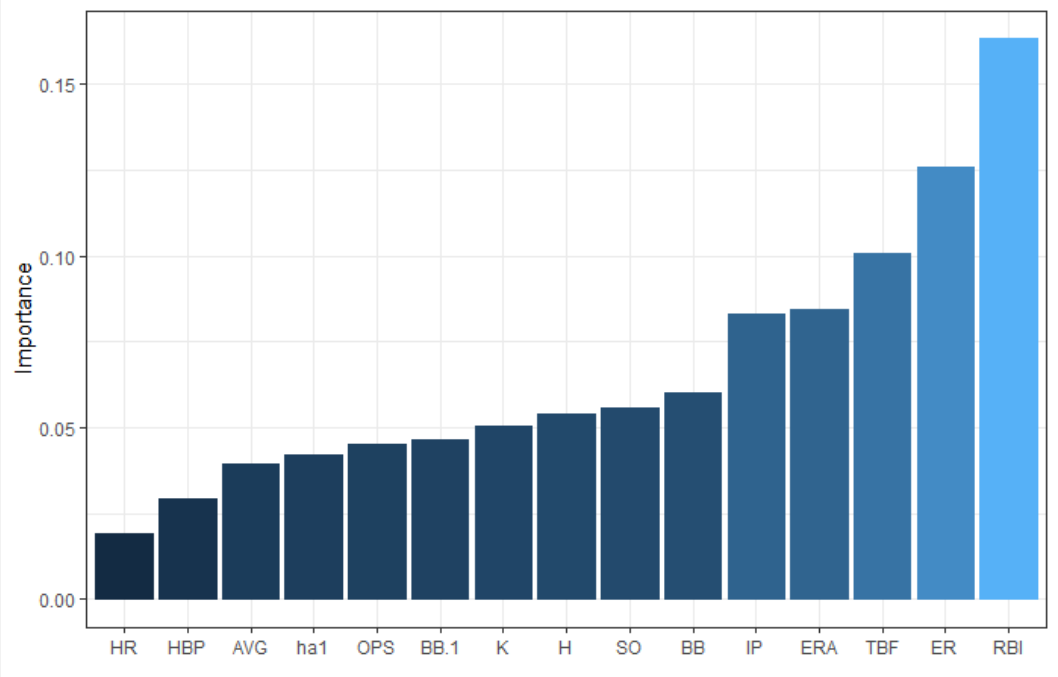
- 이 값은 최종 모형에서sigmoid 함수의 계수 값으로 weighted sum을 대입한 activation function의 가중치를 나타냄

→결국 첫 번째, 여덟 번째 hidden node의 값이 작을 수록, 다섯 번째 hidden node의 값이 클수록 1로 분류할 확률이 높아진다. (여기서 1은 승리)

4. 데이터 마이닝 모형화 - Artificial Neural Network -

인공신경망 모형화





plotnet(Dnnet)

→ H5가 가장 가중치가 높은 걸 알 수 있음

RBI, ER, TBF, ERA, IP, BB 순으로 중요도가 나열되었음. 인공신경망의 특징에 따라 실행할 때마다 조금씩 다른 순서로 나옴

4. 데이터 마이닝 모형화 - Artificial Neural Network -

```
> pre=predict(Dnnet, test, type = "class")
> actual = test$win
> tab=table(actual, pre)
> tab
      pre
actual   0   1
     0 276  16
     1  13 249
> accuracy=(tab[1,1]+tab[2,2])/sum(tab)
> accuracy
[1] 0.9476534
```

예측 정확도 : (276+249)/
(276+16+13+249)
= 94.76534%

4. 데이터 마이닝 모형화 - Artificial Neural Network -

NC vs LG

<데이터 셋>

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	date	team	win	ha	H	RBI	BB	HBP	SO	AVG	OPS	IP	TBF	ER	BB	K	HR	ERA
1387	20191003	NC		0	0.773405	0.354252	0.25967	0.046723	0.50877	0.265679	0.736989	2.224378	9.969236	0.923276	0.755193	1.760543	0.190605	3.972512
1388	20191003	LG		1	0.729368	0.331821	0.240571	0.042827	0.536832	0.240214	0.639584	2.196358	9.936348	0.960295	0.842874	1.712958	0.157443	3.205396

```
> predict_data <- D[1386:1387,]
> win_predict <- predict(Dnnet, predict_data, type="class")
> win_predict1 <- predict(Dnnet, predict_data, type="raw")
> win_predict
[1] "0" "1"
> win_predict1
           [,1]
1386 0.02290139
1387 0.97443427
```

=> NC 승리 확률 = 2.29%
=> LG 승리 확률 = 97.44%
=> LG가 NC를 이길 것으로 예측.

5. 결과 해석

모델	정확도
Decision Tree	90.78%
Random Forest	94.76%
Logistic Regression	94.22%
ANN	94.77%

서로 상반된 결과를 도출해야 하는 모형인 경우,
Decision Tree와 Radom Forest를 쓰기에 한계가 존재한다.

6. 역할

“

임세훈

데이터 수집 및 분석

Logistic Regression 모형화

발표

”

Q & A