

휴게소 만족도에 영향을 끼치는 요인 분석

강태웅 양용수 나영비 신재원

CONTENTS

- 01 분석 배경
- 02 데이터 수집
- 03 모델 생성
- 04 정확도
- 05 분석 결과
- 06 분석의 기대효과와 한계점
- 07 Q & A

01

분석 배경

- ❶ 만족도 조사 실시
- ❷ 만족도 기준 불분명, 개선점 파악 불가
- ❸ 진단 모델 구현



01

분석 배경

진단 모델 구현



만족도 등급 예측
모델 구현, 생성

대본

+



중요 변수 파악

대본

+



만족도 향상을 위한
자료 제공

대본

02

데이터 수집

분석을 위한 데이터를 수집

한국도로교통공사

e.g.) 주차장, 화장실, 세차장, 경정비소,
쉼터, 주유소, LPG 충전소

휴게소 별 브랜드 매장

e.g.) 던킨 도너츠, 파스쿠찌, 카페베네,
BBQ, 나부루, 베스킨라빈스

휴게소 등급(5등급)

1등급부터 5등급까지의 휴게소 등급을
수집하였습니다.

휴게소 별 매출액

sale항은 "유동인구 수"를 대체하기 위한
휴게소 별 매출액입니다.

02

데이터 수집

명목형 변수, 수치형 변수, training/test data

	A	B	C	D	
1	name	class	sale	parking	tc
2	건천(서울)	1	6152322	124	
3	건천(부산)	2	4172308	146	
4	경산(서울)	3	7832193	215	
5	경주(부산)	1	4661128	290	
6	금강(부산)	4	13235271	295	
7	기흥(부산)	4	16479680	418	
8	김천(서울)	2	4754764	163	
9	김천(부산)	2	3200314	193	
10	망향(부산)	1	18998845	332	



명목형 변수

6개의 변수를 제외하고 모든 변수
를 Yes/No 로 표기했습니다.

T	U	V	W	X	
angelinus	hollys	lotteria	cobaco	paba	BBC
No	No	No	No	No	No
Yes	No	No	No	No	No
No	Yes	No	No	No	No
No	No	No	No	No	No
Yes	No	Yes	Yes	No	No
No	No	No	Yes	Yes	Yes
No	No	No	No	No	No
No	No	No	No	No	No
No	No	No	No	No	No
No	Yes	No	Yes	No	No
No	No	No	No	No	No



수치형 변수

class, sale, parking, airpressure,
vacuumcle, matcleaner

O	P	Q
airpressure	vacuumcle	matcleaner
1	1	1
1	1	1
1	1	1
2	1	1
1	1	0



test data

193개 중 30개의 데이터를
test data로 설정했습니다.

02

데이터 수집

정리된 엑셀파일

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	name	class	sale	parking	toilet	carwash	lightrepair	freightrou	restarea	cheapgas	LPG	electric	hydrogen	pet	airpressure	vacuumcle	matclease	dunkind	tomandtor	angelinus	hollys
2	건천(서울)	1	6152322	124	40	No	No	No	No	Yes	No	Yes	No	No	1	1	1	Yes	Yes	No	No
3	건천(부산)	2	4172308	146	50	No	No	No	No	Yes	No	Yes	No	No	1	1	1	No	No	Yes	No
4	경산(서울)	3	7832193	215	54	No	Yes	Yes	No	Yes	Yes	Yes	No	No	1	1	1	No	Yes	No	Yes
5	경주(부산)	1	4661128	290	45	No	Yes	Yes	Yes	Yes	No	Yes	No	No	2	1	1	Yes	Yes	No	No
6	금강(부산)	4	13235271	295	44	No	Yes	No	No	Yes	No	Yes	No	No	1	1	0	No	Yes	Yes	No
7	기흥(부산)	4	16479680	418	74	Yes	No	No	No	Yes	No	Yes	No	No	1	2	1	Yes	Yes	No	No
8	김천(서울)	2	4754764	163	46	No	No	Yes	Yes	Yes	No	Yes	No	No	1	1	1	Yes	No	No	No
9	김천(부산)	2	3200314	193	42	No	No	Yes	Yes	Yes	No	Yes	No	No	1	1	1	Yes	No	No	No
10	망향(부산)	1	18998845	332	98	No	Yes	No	Yes	Yes	Yes	Yes	No	No	1	1	1	Yes	Yes	No	No
11	서울만남(부산)	3	7536828	262	51	No	No	No	No	Yes	Yes	Yes	No	No	1	2	1	No	Yes	No	Yes
12	신탄진(서울)	3	9718374	293	94	No	Yes	Yes	No	Yes	No	Yes	No	No	1	1	1	No	Yes	No	No
13	안성(서울)	3	17587832	385	108	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	1	1	0	Yes	No	Yes	Yes
14	안성(부산)	3	25072692	479	117	No	No	No	No	Yes	No	Yes	Yes	No	1	1	2	No	No	No	Yes
15	양산(서울)	3	4509251	207	42	No	No	No	Yes	Yes	No	Yes	No	No	1	2	1	No	No	No	Yes
16	언양(서울)	3	3215452	170	52	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	1	1	0	Yes	Yes	Yes	No
17	옥산(부산)	3	8932052	319	102	No	Yes	Yes	No	Yes	No	Yes	No	No	1	1	1	No	No	No	Yes
18	옥천(서울)	4	3388042	126	48	No	Yes	No	Yes	Yes	No	Yes	No	No	1	1	0	No	No	No	Yes
19	옥천(부산)	2	4813267	145	32	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	1	1	1	No	No	No	Yes
20	옥천만남	5	233840	169	54	Yes	Yes	Yes	Yes	No	No	Yes	No	No	1	1	1	No	No	No	No
21	입장거봉포	2	14049383	309	64	No	Yes	Yes	Yes	Yes	No	Yes	No	No	1	1	1	Yes	Yes	No	No
22	죽암(서울)	3	8934237	240	48	No	Yes	No	No	Yes	Yes	Yes	No	Yes	1	1	1	Yes	No	Yes	No
23	죽암(부산)	3	11171078	175	32	No	No	No	No	Yes	No	Yes	No	No	1	1	1	Yes	Yes	No	No
24	죽전(서울)	2	15314015	186	94	Yes	No	No	No	Yes	No	Yes	No	No	2	1	0	No	No	No	No
25	천안삼거리	2	20243194	257	128	Yes	No	No	Yes	No	Yes	Yes	No	No	1	1	1	Yes	Yes	No	No
26	천안호두(부산)	4	12171188	176	78	No	No	No	Yes	Yes	No	Yes	No	No	1	1	1	Yes	Yes	Yes	No
27	청주(서울)	3	8791222	287	74	No	No	Yes	Yes	No	Yes	Yes	No	No	1	2	1	Yes	No	Yes	No

데이터 수집

명목형/수치형 변수, 열이름 변환

	A	B	C	D	E
1	name	class	sale	parking	toilet
2	건천(서울)	1	6152322	124	40
3	건천(부산)	2	4172308	146	50
4	경산(서울)	3	7832193	215	54
5	경주(부산)	1	4661128	290	45
6	금강(부산)	4	13235271	295	44
7	기흥(부산)	4	16479680	418	74
8	김천(서울)	2	4754764	163	46
9	김천(부산)	2	3200314	193	42
10	망향(부산)	1	18998845	332	98
11	서울만남(서울)	3	7536828	262	51
12	신탄진(서울)	3	9718374	293	94
13	안성(서울)	3	17587832	385	108
14	안성(부산)	3	25072692	479	117
15	양산(서울)	3	4509251	207	42
16	연양(서울)	3	3215452	170	52
17	옥산(부산)	3	8932052	319	102
18	옥천(서울)	4	3388042	126	48
19	옥천(부산)	2	4813267	145	32
20	옥천만남	5	233840	169	54
21	입장거봉포	2	14049383	309	64

O	P	Q
airpressure	vacuumcle	matcleane
1	1	1
1	1	1
1	1	1
2	1	1
1	1	0
1	2	1
1	1	1
1	1	1
1	1	1
1	2	1
1	1	1
1	1	0
1	1	2
1	2	1
1	1	0
1	1	1
1	1	0
1	1	1

[illegible]

	A	B	C
1	name	휴게소명	
2	class	등급	
3	sale(1,000)	매출액(천원)	
4	parking	주차장 수	
5	toilet	변기 수	
6	carwash	세차장	
7	lightrepair	경정비소	
8	freightrou	화물차라온지	
9	restarea	쉼터	
10	cheapgas	알뜰주유소	
11	LPG	LPG	
12	electric	전기차충전소	
13	hydrogen	수소차충전소	
14	pet	반려동물 편의시설	
15	airpressure	공기압정검기	
16	vacuumcle	진공청소기	
17	matcleaner	매트청소기	
18	dunkind	던킨도너츠	
19	tomandtor	탐앤탐스	
20	angelinus	엔젤리너스	
21	hollys	할리스	
22	lotteria	롯데리아	
23	cobaco	코바코	
24	paba	파리바게뜨	
25	BBQ	BBQ	
26	pascucci	파스구찌	
27	droptop	드롭탑	
28	natuur	나뚜루	
29	baskin	배스킨라빈스	
30	chrispy	크리스피크림도넛	
31	tousles	뚜레쥬르	
32	cafebene	카페베네	
33	dalcomm	달콤커피	
34	antiand	앤티앤스프레즐	
35	macnal	맥도날드	
36	starbucks	스타벅스	
37	rotvun	로띠번	
38	mangosix	망고식스	
39	yongudon	용우동	
40	kalguksu	명동칼국수	

03

모델 생성

- ❶ 데이터 전처리
- ❷ training/test data에 대한 정확도
- ❸ 분석 결과



03

모델 생성

데이터 전처리



5개의 클래스를 3개로 줄이기

1등급과 2등급은 1등급, 3등급은 2등급, 4등급과 5등급을 3등급으로 줄여서 모델을 생성했습니다.

```
for(i in 1:193){
  rest_n[i,"class"]=ifelse((rest_n[i,"class"]==1)|(rest_n[i,"class"]==2),1,ifelse((rest_n[i,"class"]==3),2,3))
}
```

training data

test data 나누기

총 193개의 데이터 중 30개의 test data
163개의 training data로 나눴습니다.

```
#데이터 쪼개기- 총193개 row, test를 30개 나머지 train
rest_train=rest_n[1:163,]
rest_test=rest_n[164:193,]
table(rest_train$class)
table(rest_test$class)
rest_test
```

04

training data 정확도

training data 성능평가



training data 정확도

```
m3=C5.0(rest_train[,c(-1)],rest_train$class)
summary(m3)
```

+



중요 변수 파악

Evaluation on training data (163 cases):

Decision Tree			

Size	Errors		
46	20(12.3%)	<<	
(a)	(b)	(c)	<-classified as
-----	-----	-----	
43	4	2	(a): class 1
2	60	4	(b): class 2
4	4	40	(c): class 3

04

test data 정확도

test data에 대한 성능평가

```
#prunning
cv.trees= cv.tree(rest_tree, FUN=prune.misclass)
plot(cv.trees) #20일 때 missclass 가장 적음

rest_tree_prun=prune.misclass(rest_tree, best=20)
```

```
#rpart
library(rpart)
model_rpart=rpart(class~., data=rest_train)
rest_rpart_pred= predict(model_rpart, rest_test,type="class")
```

```
#부스팅 >> 모델의 성능을 개선시키는
rest_boosting=C5.0(rest_train[,-1], rest_train$class, trials=9)
```

```
#bagging
library(ipred)
set.seed(123)
model <- bagging(formula = class ~., data = rest_train, coob = TRUE, nbag=160)

pred bagging <- predict(object = model, newdata = rest_test, type = "class")
```



prunning

accuracy: 0.466667



rpart

accuracy: 0.333333



boosting

accuracy: 0.566667



bagging

accuracy: 0.466667

05

분석 결과

01

의미 없는 sale 변수

유동인구 대신 넣은 변수였지만 결과에서는 유의미한 해석이 어려워졌습니다.

02

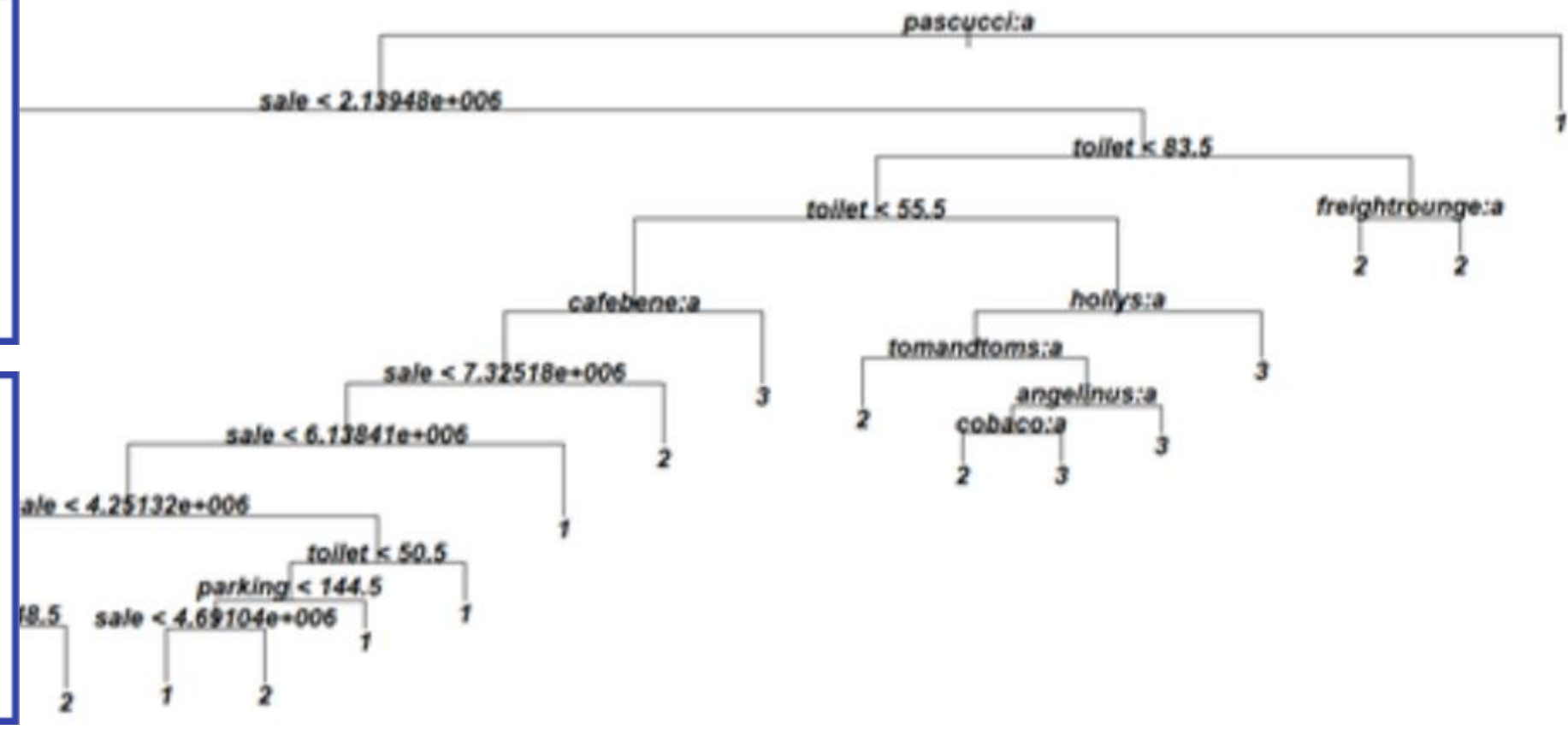
난잡한 모양의 결과

보기가 편하다는 장점이 있는 방법론이었지만 결과물이 너무 지저분하게 나왔습니다.

03

의미 없는 브랜드 음식점

카페 종류에는 유의미한 결과를 도출할 수 있었지만 용우동과 명동칼국수는 그렇지 않았습니다.



모델 2

- ❶ 데이터 전처리
- ❷ training/test data에 대한 정확도
- ❸ 분석 결과

TVU division			FRT division		
254	550	254	274	154	415
320	754	273	825	154	
114	304	954	174		
		175	274		

05

모델 생성

데이터 전처리



"프랜차이즈 카페 매장 수"
라는 새로운 변수 생성

```
#카페 수 변수 생성
rest_new$cafe_n=0
cafe=c("dunkind","tomandtoms","angelinus","hollys","paba","pascucci","droptop","natuur",
       "baskin","chrispy","tousles","cafebene","dalcomm","starbucks","rotvun","mangosix")

for(i in 1:length(rest_new$cafe_n)){
  for(j in 1:length(cafe)){
    rest_new[i,"cafe_n"]=rest_new[i,"cafe_n"]+ifelse(rest_new[i,cafe[j]]=="Yes",1,0)|
  }}
}
```

+



음식점, 매출액 제외

매출액과 음식점에서 유의미한 결과를
도출해내기 어렵고, 결과물을 더 난잡하게만
만들었다고 판단하여 제외시켰습니다.

05

모델 생성

모델 2의 데이터 전처리

training data

test data 나누기

총 193개의 데이터 중 30개의 test data
163개의 training data로 나눴습니다.

```
#데이터의 행순서를 한번 섞어서 데이터 쪼개기
set.seed(12345)
rest_rand_3=rest_new[order(runif(193)),]
rest_rand_3
str(rest_rand_3)
#데이터 쪼개기- 총193개 row, test를 30개 나머지 train
rest_train=rest_rand_3[1:163,]
rest_test=rest_rand_3[164:193,]
table(rest_train$class)
table(rest_test$class)
rest_test
```


05

training data 정확도

모델2의 training data 성능평가



training data 정확도

```
m3=C5.0(rest_train[,c(-1)],rest_train$class)
summary(m3)
```

+



중요 변수 파악

Evaluation on training data (163 cases):

Decision Tree			

Size	Errors		
28	39 (23.9%)	<<	
(a)	(b)	(c)	<-classified as
----	----	----	
37	6	6	(a): class 1
7	56	3	(b): class 2
8	9	31	(c): class 3

05

test data 정확도

test data에 대한 성능평가

```

library(tree)
rest_tree=tree(rest_train$class~.,data=rest_train)

#prunning
cv.trees= cv.tree(rest_tree, FUN=prune.misclass)
plot(cv.trees) #5일 때 missclass 가장 적음

#부스팅 >> 모델의 성능을 개선시키는 >> 성능 향상 안됨
rest_boosting=C5.0(rest_train[,-1], rest_train$class, trials=5) #16,18,19
rest_boosting
summary(rest_boosting)

boost_predict=predict(rest_boosting, rest_test)

#부스팅 >> 모델의 성능을 개선시키는 >> 성능 향상 안됨
rest_boosting=C5.0(rest_train[,-1], rest_train$class, trials=5) #16,18,19
rest_boosting
summary(rest_boosting)

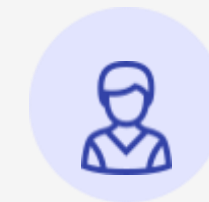
```

**prunning**

accuracy: 0.300000

**rpart**

accuracy: 0.433333

**boosting**

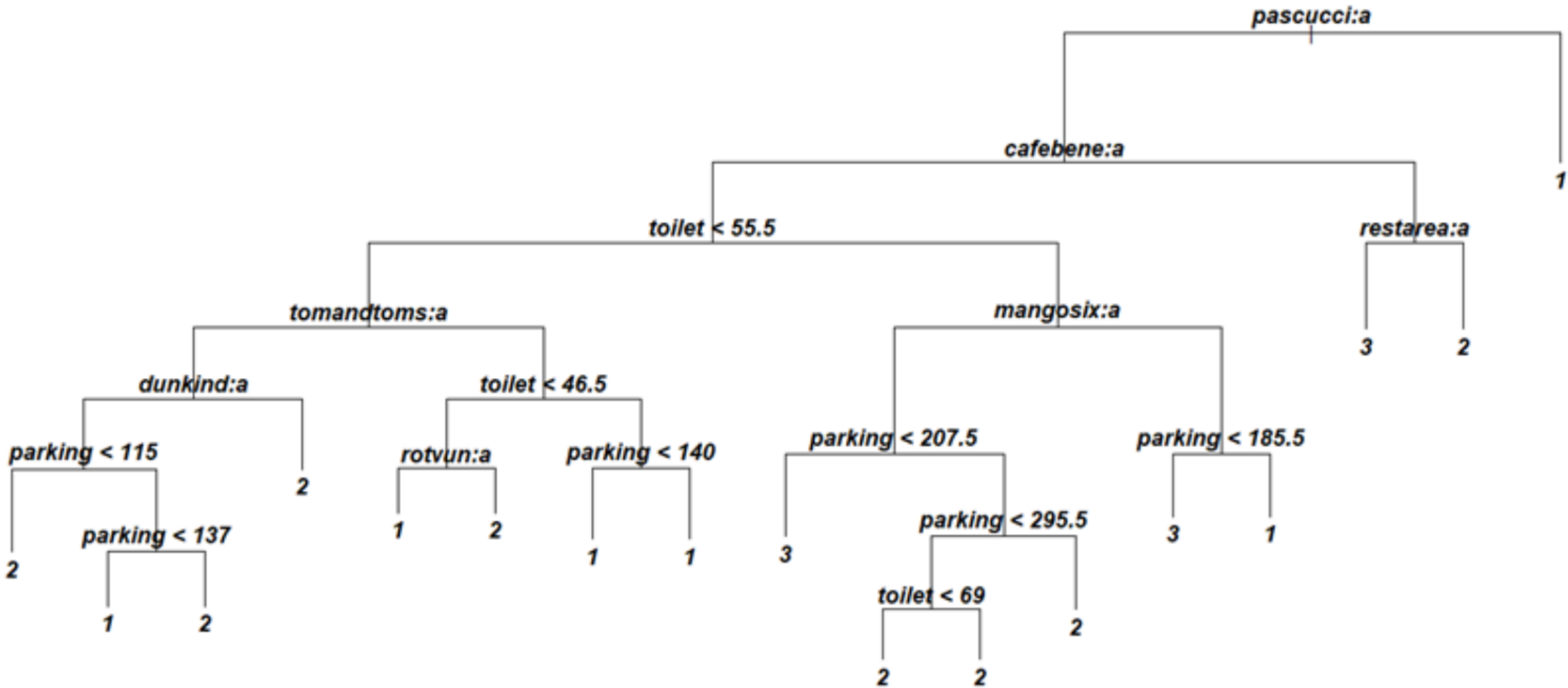
accuracy: 0.5000000

**bagging**

accuracy: 0.5000000

분석 결과

모델 2의 분석 결과



06

분석의 기대효과와 한계점



참고자료

새로운 휴게소를 만들 때의 하나의
참고자료로 활용이 가능합니다



만족도 개선

휴게소 만족도 개선을 위한
지표로 활용 가능합니다.



낮은 정확도

유동인구 데이터가 있었다면
더 높은 정확도가 나올 것으로 예상됩니다.



데이터 부족

충분하지 않은 데이터를 활용한 모델이다보
니, 정확도 개선에 실패하였습니다.

Q & A

감사합니다.

강 태 웅	16102155
양 용 수	16101157
나 영 비	17100081
신 재 원	20102028
