

R을 통한 머신러닝 프로젝트

H사 보험 사기꾼 예측 모델 생성



김현정

CONTENTS

순서

01

- 프로젝트 개요
-

02

- 데이터 EDA 및 전처리
 - 데이터 변수 삭제 / 파생변수 추가
-

03

- 데이터 모델링
-

04

- 결과

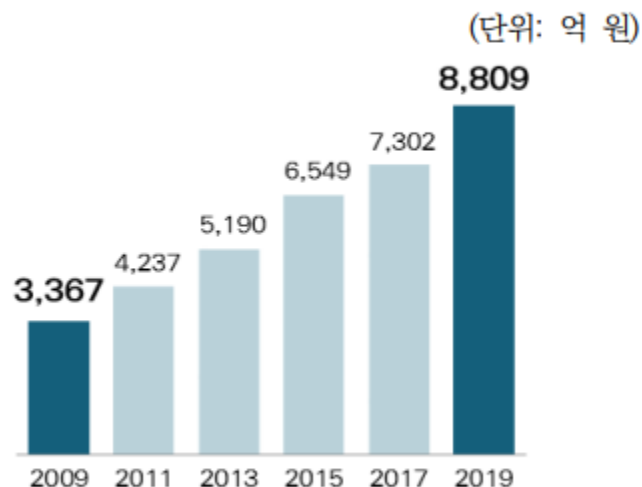
CONTENTS

01

프로젝트 개요

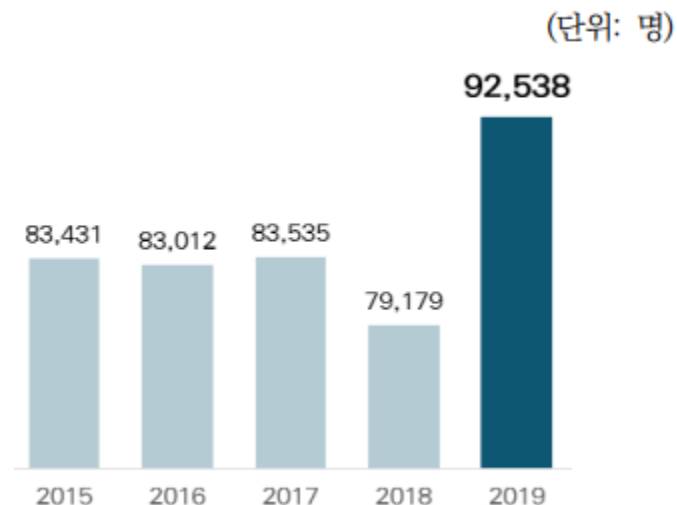


〈그림 1〉 국내 보험사기 적발금액 추이



자료: 금융감독원

〈그림 2〉 국내 보험사기 적발인원 추이



자료: 금융감독원

CAUTION · WARNING

금융감독원에 따르면 2019년 기준으로 1인당 적발금액이 천만 원 미만인 경우가 보험사기의 약 84%를 차지하였으며, 상해·질병, 자동차사고 피해를 과장하거나 왜곡하여 보험금을 청구하는 보험사기가 증가하였다.

SNS나 블로그에 구인광고를 가장하여 보험사기 공범을 모집하거나, 보험사기를 조장하는 글들이 게시되어 일반 보험계약자를 보험사기로 유인하는 사례가 증가하였다.

출처: 금융감독원 보도자료(2020. 5. 19), “코로나19 상황을 틈타 급전 필요한 분, 고액 일당 지급 등을 미끼로 한 보험사기 주의!”

출처: 금융감독원 보도자료(2020. 4. 9), “2019년 보험사기 8,809억원 적발, 전년대비 10.4% 증가”



정상 고객의 피해


“보험 사기꾼 예측 확률 향상 필요”

사회 구성원간의 신뢰도



No	변수영문명	변수타입	변수명	변수 설명
1	CUST_ID	N	고객ID	고객을 구분하는 고유번호
2	POLY_NO	N	증권번호	청약서번호이면서 동시에 계약성립후에는 증권번호로 사용
3	ACCI_OCCP_GRP1	C	직업그룹코드1	총 8개직업군으로 분류한 코드(사고 당시)
4	ACCI_OCCP_GRP2	C	직업그룹코드2	총 25개직업군으로 분류한 코드(사고 당시)
5	CHANG_FP_YN	C	FP 변경 여부	모집 FP와 청구 당시 수급 FP와의 동일 여부
6	CNTT_REC_P_SQNO	C	계약별접수일련번호	사고접수에 대
7	REC_P_DATE	C	사고접수일자	사고가 접수된
8	ORIG_RESN_DATE	C	원사유일자	사고접수시 해
9	RESN_DATE	C	사유일자	보험금 지급사
10	CRNT_PROG_DVSN	C	현재진행구분	현재진행구분 - 접수(11), 손
11	ACCL_DVSN	C	사고구분	사고원인을 구 - 재해(1), 교
12	CAUS_CODE	C	원인코드	사고의 원인에
13	CAUS_CODE_DTL	C	원인코드상세	사고의 원인에
14	DSAS_NAME	C	병명	병명
15	DMND_RESN_CODE	C	청구사유코드	지급청구의 원 - 사망(01), 은
16	DMND_RSCD_SQNO	N	청구사유코드일련번호	동일 증번, 동
17	HOSP_OTPA_STDT	C	입원/통원시작일자	입원시작일, 통
18	HOSP_OTPA_ENDT	C	입원/통원종료일자	입원종료일, 통
19	RESL_CD1	C	결과코드1	사고원인에 대
20	RESL_NM1	C	결과명1(사인내용)	결과내용
21	VLID_HOSP_OTDA	N	유효입원/통원일수	보험금지급대상인 입원일수 또는 통원일수
22	HOUSE_HOSP_DIST	N	고객병원거리	고객 거주지와 병원까지의 거리(km)

- 고객정보데이터 (CUST_DATA) : 22,400행 25열
- 보험청구데이터 (CLAIM_DATA) : 119,020행 39열

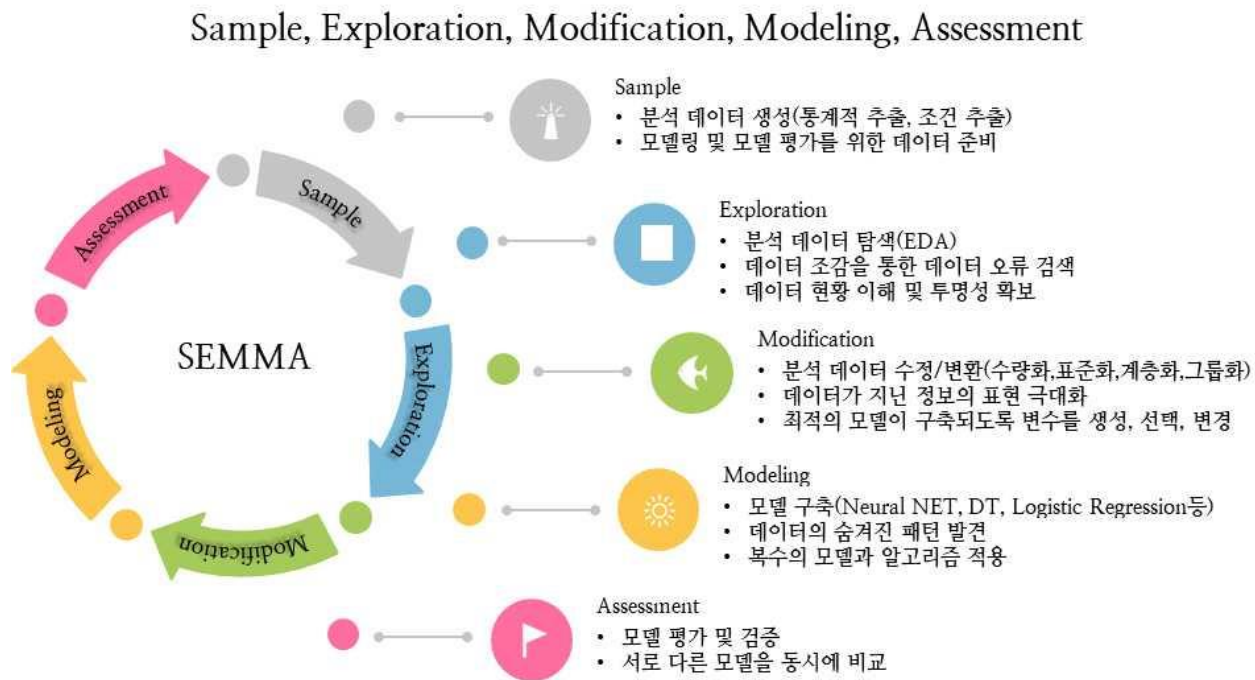


The factors that have shown strength concerning reducing fraud are auditing, monitoring, sanction and control, nurses' role (supported by applied studies), the economy, politics and social conditions, the medical record, and the commercial implication (supported by theoretical studies). On the other hand, the factors that have shown strength concerning increasing fraud are sex, age, predominant race, have health insurance, place of residence, medical and surgical treatments, chronic health conditions, risk of illness, deductibles and coinsurance, the complicity between the provider and the insurer, the relationship between the provider and the consumer, the relationship between the consumer and the insurer, the influence of the bosses and the Guanxi, (supported by applied studies), the geography, reimbursement processes and billing characteristics, information asymmetry, and poor economic situation of the patient (supported by theoretical studies).

증가하는 사기에 대한 강점을 보인 요소들은 성별, 나이, 지배적인 인종, 건강 보험, 거주지, 의료 및 외과 치료, 만성적인 건강 상태, 질병의 위험, 공제 및 동전 보험, 제공자와 보험사 사이의 합병증, 그리고 보험사 사이의 관계이다

변수영문명	변수타입	변수명
SIU_CUST_YN	C	보험사기자여부
SEX	N	성별
AGE	N	연령
RESI_COST	N	주택가격
RESI_TYPE_CODE	C	거주TYPE
FP_CAREER	C	FP경력
CUST_RGST	C	고객등록년월
CTPR	C	시도구분
OCCP_GRP1	C	직업그룹코드1
OCCP_GRP2	C	직업그룹코드2
TOTALPREM	N	납입총보험료
MINCRDT	C	신용등급(최소)
MAXCRDT	C	신용등급(최대)
WEDD_YN	C	결혼여부
MATE_OCCP_GRP1	C	배우자직업그룹코드1
MATE_OCCP_GRP2	C	배우자직업그룹코드2
CHLD_CNT	N	자녀수
LTBN_CHLD_AGE	N	막내자녀연령
MAX_PAYM_YM	C	최대보험료연월
MAX_PRM	N	최대보험료
CUST_INCM	N	고객추정소득
RCBASE_HSHD_INCM	N	추정가구소득1
JPBASE_HSHD_INCM	N	추정가구소득2

변수영문명	변수타입	변수명
CRNT_PROG_DVSN	C	현재진행구분
ACCI_DVSN	C	사고구분
CAUS_CODE	C	원인코드
CAUS_CODE_DTAL	C	원인코드상세
DSAS_NAME	C	병명
DMND_RESN_CODE	C	청구사유코드
DMND_RSCD_SQNO	N	청구사유코드일련번호
HOSP_OTPA_STDT	C	입원/통원시작일자
HOSP_OTPA_ENDT	C	입원/통원종료일자
RESL_CD1	C	결과코드1
RESL_NM1	C	결과명1(사인내용)
VLID_HOSP_OTDA	N	유효입원/통원일수
HOUSE_HOSP_DIST	N	고객병원거리
HOSP_CODE	N	병원코드
ACCI_HOSP_ADDR	C	병원지역(시도)
HOSP_SPEC_DVSN	C	병원종별구분



- 프로젝트에 SEMMA 방법론을 이용했습니다.

CONTENTS

02

DATA EDA

DATA 전처리



02 데이터 전처리 [문자형 데이터 → 숫자형 데이터]

데이터 EDA 및 전처리

- Y값은 1로, N값은 0으로 변환
- NULL 값은 0으로 대체

CUST_ID	DIVIDED_SET	SIU_CUST_YN	SEX	AGE	RESI_COST	RESI_TYPE_CODE	FP_CAREER
1	1	N	2	47	21111	20	N
2	1	N	1	53	40000	20	N
3	1	N	1	60	0	NA	N

CUST_ID	DIVIDED_SET	SIU_CUST_YN	SEX	AGE	RESI_COST	RESI_TYPE_CODE
1	1	0	2	47	21111	
2	1	0	1	53	40000	
3	1	0	1	60	0	
5	1	0	2	54	0	
6	1	0	1	62	6218	
7	1	1	2	60	11388	
8	1	0	1	57	86527	
9	1	0	1	54	22638	
12	1	0	1	58	37222	
13	1	1	1	63	8140	
15	1	0	1	59	23055	

TOTALPREM	MINCRDT	MAXCRDT	WEDD_YN
146980441	NA	NA	Y
94600109	1	6	Y
18501269	NA	NA	N

MINCRDT	MAXCRDT	WEDD_YN	MATE_OCCP_GRP
6	6	1	3
1	6	1	1
6	6	0	0
8	8	1	3
6	6	1	1
6	6	1	0
6	6	1	2
6	6	1	4
6	7	1	5
6	6	1	1

CODE

```
| yn.fun <- function(data){ if(data == 'Y'){ data=1 }else if  
| (data == 'N')  
| { data=0 } else {data=' '} }  
| cust_data$SIU_CUST_YN <- sapply(cust_data$SIU_CUST_YN,  
| yn.fun)  
| cust_data$WEDD_YN <- sapply(cust_data$WEDD_YN, yn.fun)
```

>> 적용된 변수: SIU_CUST_YN / WEDD_YN

+ `yn.fun`을 생성한 후 `Supply` 함수를 이용해 문자형 데이터를 1과 0으로 표현합니다.

+SIU_CUST_YN의 NULL 값은 추후 분석이 필요한 관측치라 변경 X

02 데이터 전처리 [문자형 데이터 → 숫자형 데이터]

데이터 EDA 및 전처리

- » 지역이름을 Number 형으로 변환
- » 직업이름을 코드(숫자)만 추출

CTPR	OCCP_GRP_1	OCCP_GRP_2	TOTALPREM	MI
충북	3.사무직	사무직	146980441	
서울	3.사무직	사무직	94600109	
서울	5.서비스	2차산업 종사자	18501269	
경기	2.자영업	3차산업 종사자	317223657	
광주	2.자영업	3차산업 종사자	10506072	
충남	3.사무직	고위 공무원	22313040	
서울	5.서비스	3차산업 종사자	46522197	
서울	2.자영업	자영업	151085847	
서울	4.전문직	공무원	3666050	
서울	4.전문직	대학교수/강사	135719262	
경기	6.제조업	운전직	33261687	

CTPR	OCCP_GRP_1	OCCP_GRP_2	TOTALPREM	MINCRDT	MAXCR
18	3	사무직	146980441	6	
10	3	사무직	94600109	1	
10	5	2차산업 종사자	18501269	6	
6	2	3차산업 종사자	10506072	8	
17	3	고위 공무원	22313040	6	
10	5	3차산업 종사자	46522197	6	
10	2	자영업	151085847	6	
10	4	공무원	3666050	6	
10	4	대학교수/강사	135719262	6	
3	6	운전직	33261687	6	
10	3	사무직	33336919	6	

CODE

```
area.name <- levels(as.factor(cust_data$CTPR))
cust_data$CTPR <- as.numeric(as.factor(cust_data$CTPR))
occp.fun <- function(data){ data = substr(data, 1,1)
  if(data==' '){ data='0' }else{data=data}}
cust_data$OCCP_GRP_1 <- apply(cust_data$OCCP_GRP_1, occp.fun)
cust_data$OCCP_GRP_1 <-
as.numeric(cust_data$OCCP_GRP_1)cust_data$MATE_OCCP_GRP_1
<- apply(cust_data$MATE_OCCP_GRP_1, occp.fun)
```

» 적용된 변수 : CTPR

지역이름은 Factor형으로 변환해 단순 레이블 인코딩 합니다.
이때 지역이름은 'area.name'에 별도 저장합니다.

» 적용된 변수 : OCCP_GRP_1

Sub_str()을 사용해 앞 숫자를 추출한 후 숫자로 치환합니다.
※ MATE_OCCP_GRP_1 도 동일하게 진행합니다.

» MIN/MAXCRDT 결측치를 특정 값으로 대체

TOTALPREM	MINCRDT	MAXCRDT	WEDD_YN
NA	NA	NA	N
NA	6	6	N
NA	NA	NA	Y
NA	NA	NA	N
NA	NA	NA	N
NA	NA	NA	Y
NA	7	8	Y
NA	NA	NA	Y
NA	6	6	N
NA	NA	NA	N

TOTALPREM	MINCRDT	MAXCRDT	WEDD_YN
0	7	8	0
0	6	6	1
0	6	6	1
0	6	6	0
0	6	6	0
0	6	6	0
0	6	6	0
0	6	6	1
0	6	6	0

CODE

```
min <- cust_data$MINCRDT
max <- cust_data$MAXCRDT

cust_data$MINCRDT <- ifelse(is.na(min), 6, min)
cust_data$MAXCRDT <- ifelse(is.na(max), 6, max)
```

변수 정의서 설명서에 따라
ifelse() 를 이용해 NA 값을 6으로 대체했습니다

» RESI_TYPE_CODE 및 TOTALPREM의 결측치를 0으로 대체

RESI_TYPE_CODE	FP_CAREER	CUST_RGST	CTPR	OCCP_GRP_1	OCCP_GRP_2	TOTALPREM
NA	N	200312	서울	8.기타	박생	NA
NA	N	200306	전남	8.기타	하생	NA
11	N	200306	전남	3.사무		
NA	N	201205		3.사무		
NA	N	201305		3.사무		
NA	N	200306				
20	N	201310	서울	8.기타		
NA	N	200306	경북	4.전문		
NA	N	200607	대구	5.서비스		
NA	N	200306	경기	8.기타		
NA	N	200503	인천	8.기타		
NA	N	200306	울산	8.기타		

RESI_TYPE_CODE	FP_CAREER	CUST_RGST	CTPR	OCCP_GRP_1	OCCP_GRP_2	TOTALPREM
12	N	200305	18	1	주부	0
12	N	200306	4	1	주부	0
13	N	200306	3	1	주부	0
0	N	200306	1	0		0
11	N	NA	1	0		0
13	N	200306	12	5	3차산업 종사자	0
99	N	200306	10	1	주부	0
0	N	200306	1	0		0
11	N	200306	14	1	주부	0
20	N	200306	14	2	자영업	0
12	N	200306	3	3	3차산업 종사자	0

CODE

```
resicode <- cust_data$RESI_TYPE_CODE
cust_data$RESI_TYPE_CODE <- ifelse(is.na(resicode), 0,
resicode)
```

```
total <- cust_data$TOTALPREM
cust_data$TOTALPREM <- ifelse(is.na(total), 0, total)
```

»RESI_TYPE_CODE

결측치를 기타(99)로 대체했을 때의 F1-score 보다 결측치를 0으로 대체했을 때 값이 더 높아서 0으로 대체했습니다.

»TOTALPREM

고객이 지금까지 당사에 실제 납입한 총 보험료의 합계의 결측치를 0으로 대체했습니다.

» CUST_INCM의 NA 값을 같은 직업군의 평균 소득으로 대체

PRM	CUST_INCM	RCBASE_HS
NA	3062	
NA	0	
NA	4937	
NA	NA	
NA	0	
NA	NA	
NA	5599	

M	MAX_PRM	CUST_INCM	RCBASE_HSHD_IN
000402	161200	0	
000511	170223	0	
000209	29	0	
000208	29	4169	
000312	193950	4169	
000000	70700	3103	
000305	86600	0	
000210	13	3691	
000508	80000	0	

> occpavg 같은 직업군의 평균 소득

	0	1	2	3	4	5	6	7	8
4169	0	4621	4319	4173	3950	4182	4026	416	

CODE

```

occpavg <- round(tapply(cust_data$CUST_INCM, cust_data$OCCP_GRP_1,
mean, na.rm=TRUE))
cust_data$OCCP_GRP_1 <- as.numeric(cust_data$OCCP_GRP_1)
occpna.fun <- function(income, occp){
  if(is.na(income)){ income=occpavg[occp+1] } else {income=income} }

cust_data$CUST_INCM<- mapply(FUN = occpna.fun, cust_data$CUST_INCM,
cust_data$OCCP_GRP_1)
    
```

고객의 소득 변수 중 값을 알 수 없는 “결측치”를
각 고객이 속한 직업군의 평균 소득으로 대체했습니다.

> Tapply(x,y,fun) 를 이용해 “OCCPAVG” 변수에
각 직업별 평균 소득 입력
> Occpna.fun 함수를 생성한 다음, Mapply(fun, x,y)를
이용해 값을 대체

» 연속형 변수인 AGE를 연령대로 구간화

SIU_CUST_YN	SEX	AGE	RESI_COST
N	2	30	0
N	2	21	0
N	1	45	0
N	1	3	0
N	1	20	0
N	2	65	0
N	1	55	0
N	1	59	0

SIU_CUST_YN	SEX	AGE	RESI_COST
0	2	4	21111
0	1	5	40000
0	1	6	0
0	2	5	0
0	1	6	6218
0	2	6	11388
0	1	5	86527

CODE

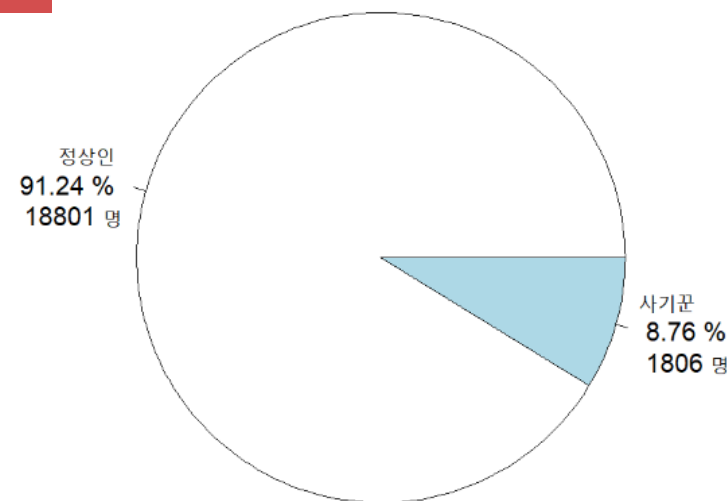
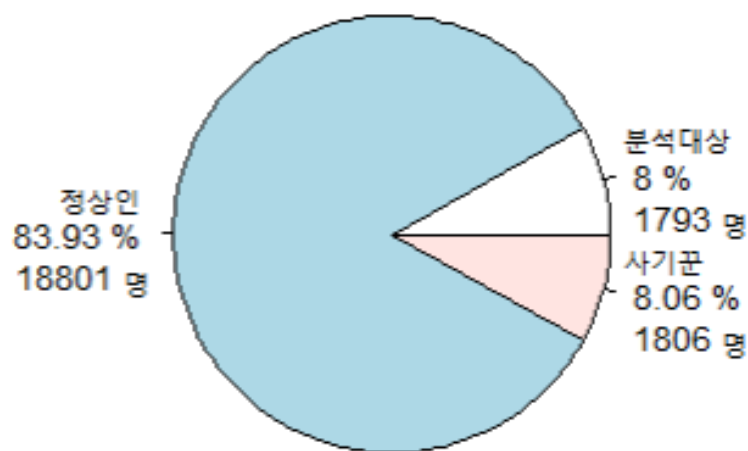
```
age.fun <- function(data){ data%/%10 }
cust_data$AGE <- sapply(cust_data$AGE, age.fun)

names(a_pc)<-
c("10세미만","10대","20대","30대","40대","50대","60대","70대","80대")
```

나이를 10으로 나누는 age.fun을 생성하여 AGE 데이터를 모두 연령대로 구간화시킵니다.

» 사기꾼 비율

- + 전체 : 22,400명
- + 정상인: 18,801명 (91.24%) / 미분류 :1,793명
- + 사기꾼 :1,806명(8.76%)



CODE

```

cust_data <- subset(cust_data, subset=
(cust_data$DIVIDED_SET==1))
siu <- table(cust_data$SIU_CUST_YN)
siu_pc <- round(siu/sum(siu)*100,2)
pie(siu_pc, main='사기꾼 비율', labels = paste(c('정상인',
'사기꾼'), '\n', siu_pc, '%', '\n', siu, '명'), cex=1.2)
    
```

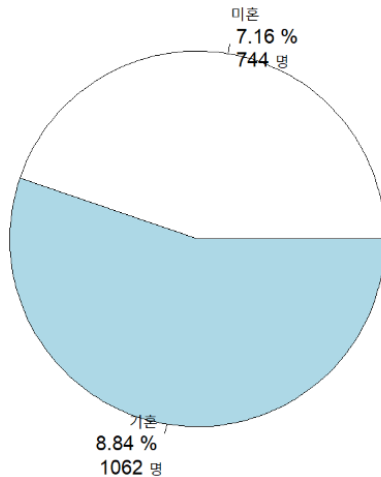
DIVIDED SET ==1인, 사기꾼 여부가 밝혀진 데이터를 Cust_data 변수에 저장합니다.

전체 2,2400명 중
정상인은 18,801명(89.93%), 사기꾼은 1,806명(8.06%),
미분류(분석대상)은 1,793명(8%)입니다.

미분류를 제외한 고객의 수로 비교시,
사기꾼은 20,607명 중 8.76%의 비율을 보입니다.

» 사기꾼 기혼자 비율

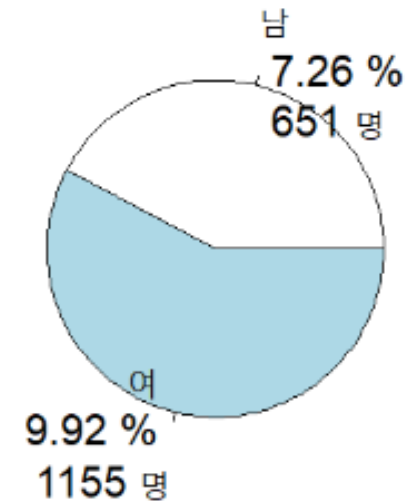
- + 전체 : 1,806명
- + 미혼 : 744명 (7.16%)
- + 기혼 : 1,062명(8.84%)



사기꾼 1,806명 중 미혼인 고객은 733명으로, 전체 미혼 고객 중 사기꾼 비율이 7.5%이고, **기혼고객은 1,030명으로 전체 기혼고객 중 9.54%입니다.** 이를 통해 기혼 고객이 미혼고객보다 사기꾼 분포가 더 높음을 알 수 있습니다.

» 사기꾼 성별 비율

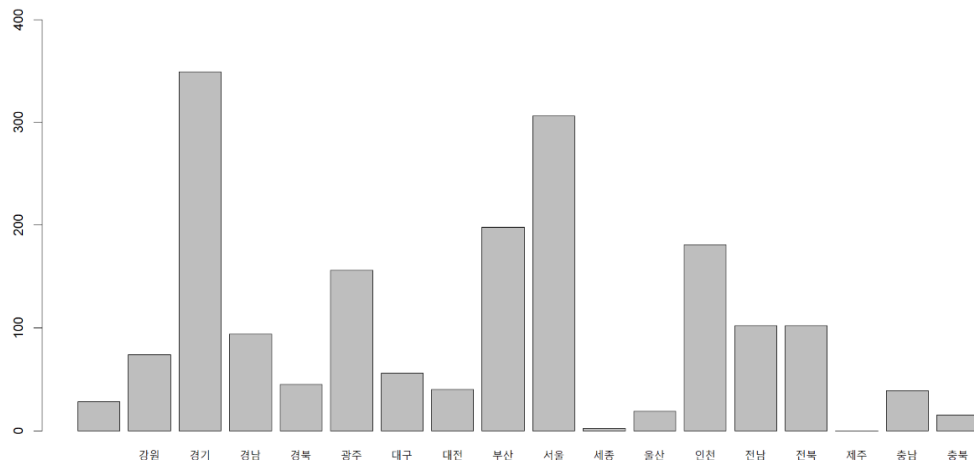
- + 전체 : 1,806명
- + 남성 : 651명 (7.26%)
- + 여성 : 1,151명(9.92%)



사기꾼 1,806명 중 여성은 1,151명으로 전체 여성 중 사기꾼 비율이 9.92%이고, 남성은 651명으로 전체 남성 중 사기꾼 비율이 7.26%입니다. 이를 통해 **여성이 남성보다 사기꾼 분포가 더 높음을 알 수 있습니다.**

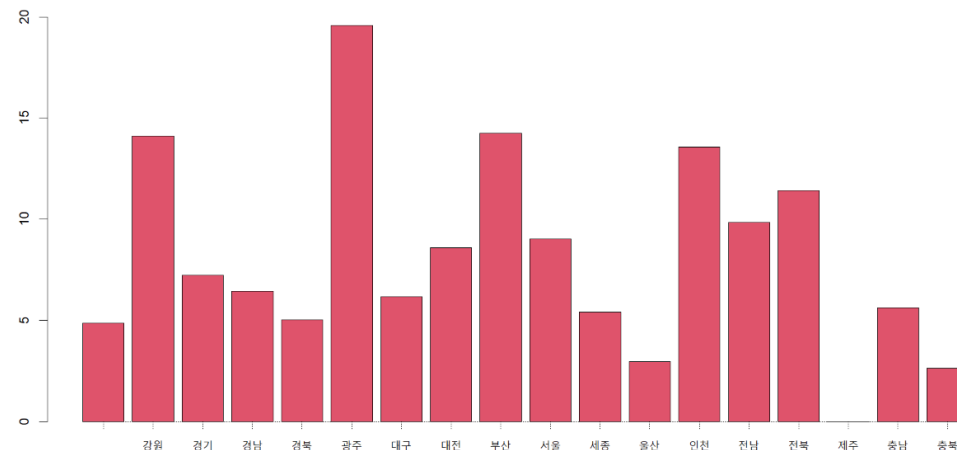
» 지역별 사기꾼 수

- + 1위 : 경기 349명
- + 2위 : 서울 306명
- + 3위 : 부산 198명



» 지역별 사기꾼 분포 비율

- + 1위 : 광주 19.57% (156명)
- + 2위 : 부산 14.24% (74명)
- + 3위 : 강원 14.12% (198명)



거주 지역별로 사기꾼 비율을 확인했을 때, 광주에 거주하는 사기꾼이 전체 광주고객 중 19.57%인 156명으로 분포가 가장 높습니다. 그 뒤로 강원, 부산, 인천이 각각 13.73%(74명), 13.55%(198명), 13.15%(181명) 분포를 보여주고 있습니다.

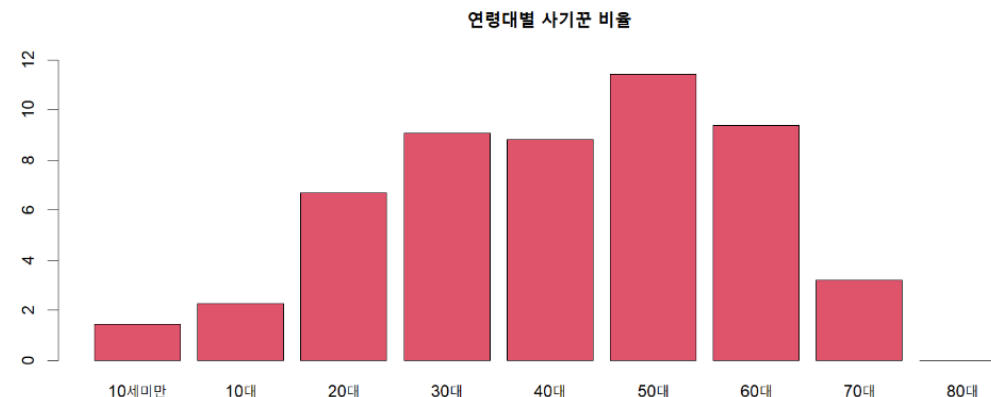
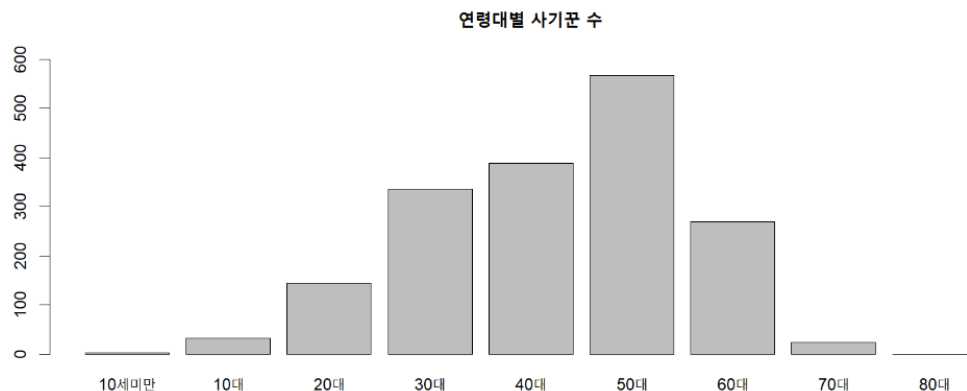
보험사기꾼의 거주지별 인원수를 확인할 경우, “경기”와 “서울” 지역에 몰려있어보이나, 분포비율을 계산하면 **광주, 강원, 부산, 인천 지역이 서울 및 경기 지역보다 높음을 알 수 있습니다.**

02 데이터 EDA

데이터 EDA 및 전처리

» 연령대별 사기꾼 수
+ 1위 : 50대 584명
+ 2위 : 40대 398명
+ 3위 : 30대 340명

» 연령대별 사기꾼 분포 비율
+ 1위 : 50대 11.50% (584명)
+ 2위 : 60대 10% (277명)
+ 3위 : 30대 9.17% (340명)



CODE

```
age.fun <- function(data){data%/%10}
cust_data1$AGE <- sapply(cust_data1$AGE, age.fun)
a<- table(sagi_data$AGE) ; a2 <- table(cust_data1$AGE)
a_pc <- round(a/a2*100,2) ; a <- append(a,0)
names(a)<-
c("10세미만","10대","20대","30대","40대","50대","60대","70대","80대")
barplot(a_pc, ylim=c(0,12), main = '연령대별 사기꾼 비율',
xlab='나이구간', ylab='비율(%)', col=10)
```

사기꾼 1,806명 중 50대가 584명으로, 전체 50대 중 11.50% 이고, 그 다음으로 높은 분포를 보여주는 연령대는 60대(9.37%, 269명)와 30대(9.06%, 336명) 입니다.

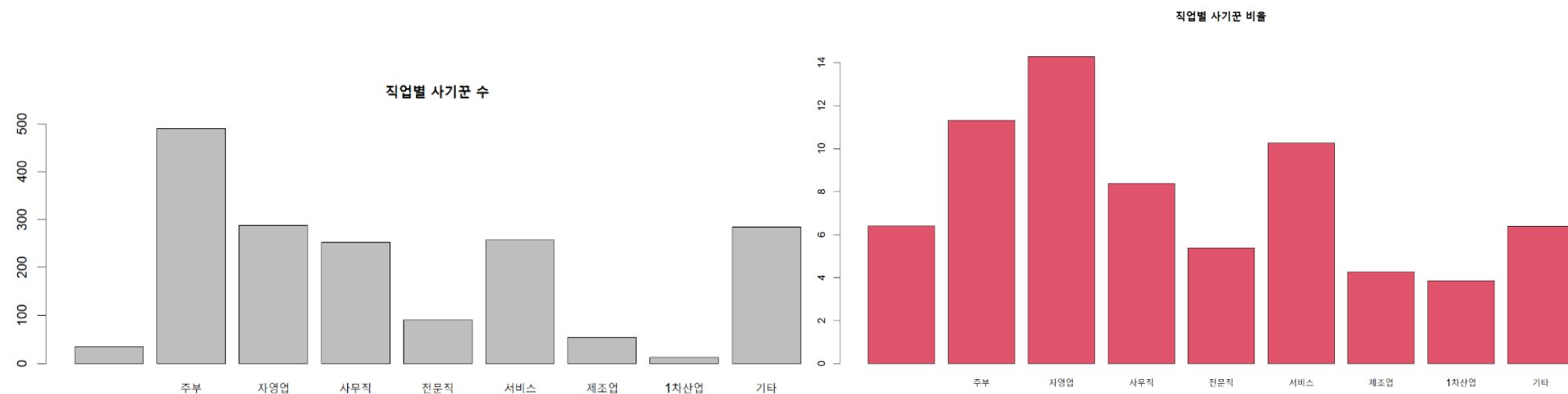
사기꾼 수 및 비율 모두 50대가 가장 높음을 알 수 있습니다.

02 데이터 EDA

데이터 EDA 및 전처리

» 직업군별 사기꾼 수
+ 1위 : 주부 498명
+ 2위 : 자영업 301명
+ 3위 : 기타 288명

» 직업군별 사기꾼 분포 비율
+ 1위 : 자영업 14.27%
+ 2위 : 주부 11.31%
+ 3위 : 서비스업 10.25%



CODE

```
o <- table(sagi_data$OCCP_GRP_1)
o2 <- table(cust_data1$OCCP_GRP_1)
o_pc <- round(o/o2*100,2)
barplot(o, ylim = c(0,500), main = '직업별 사기꾼 수', xlab='직업종류',
ylab='명',names.arg = c("", occp_name))
barplot(o_pc, ylim=c(0,15), main = '직업별 사기꾼 비율',
xlab='직업종류', ylab='비율(%)', col=10, names.arg = c("",
occp_name))
```

사기꾼 1,806명 중 주부가 498명으로 가장 수가 많으나,
직업별 인원 수 대비 사기꾼 비율을 확인했을 때
자영업이 14.27%로 가장 높습니다.

이를 통해, 고객 및 고객의 배우자 직업이 “자영업” 일 때 사기꾼 분포가
높음을 알 수 있습니다.

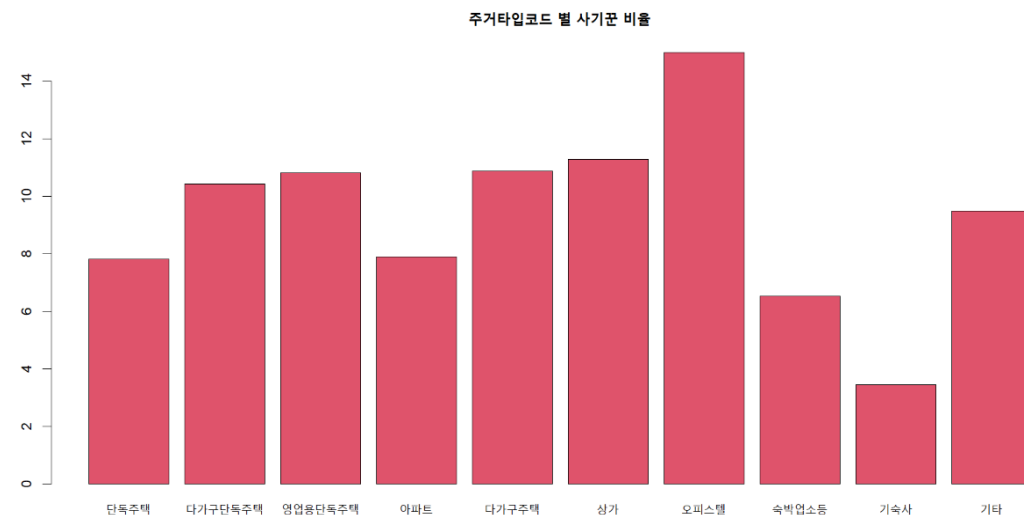
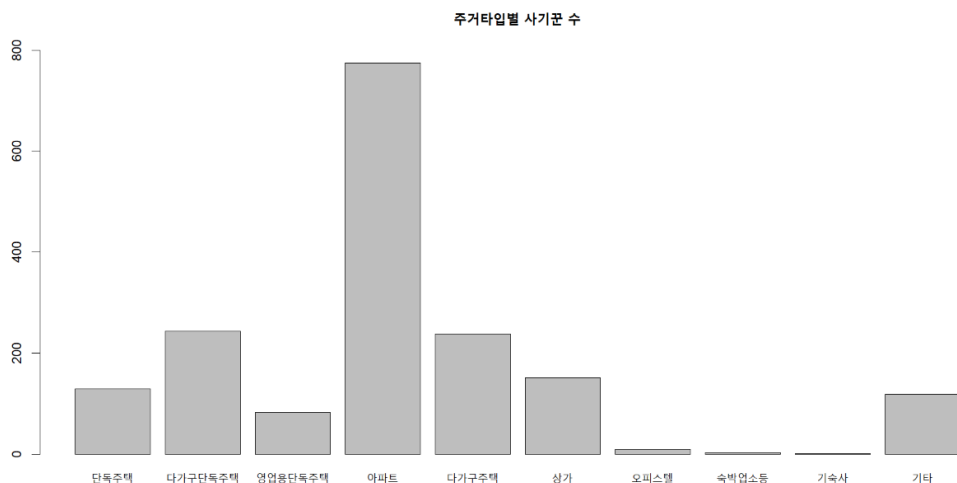
+ 배우자 직업의 사기꾼 분포 비율에서도 “자영업” 비율이 가장 높습니다.

02 데이터 EDA

데이터 EDA 및 전처리

» 주거코드별 사기꾼 수
+ 1위 : 아파트 774명
+ 2위 : 오피스텔 9명

» 주거코드별 사기꾼 분포 비율
+ 1위 : 오피스텔 15%
+ 2위 : 아파트 7.89%



CODE

```
r2<- table(cust_data1$RESI_TYPE_CODE)
r<- table(sagi_data$RESI_TYPE_CODE)
r_pc <- round(r/r2*100,2)
resi.name <- c('단독주택','다가구단독주택','영업용단독주택',
'아파트','다가구주택','상가','오피스텔','숙박업소등','기숙사','기타')
barplot(r, ylim = c(0,800), main = '주거타입별 사기꾼 수', xlab='주거타입',
ylab='명', names.arg = resi.name)
barplot(r_pc, ylim=c(0,15), main='주거타입코드 별 사기꾼 비율',
xlab='주거타입', ylab='비율(%)', col=10, names.arg = resi.name )
```

사기꾼 1,806명 중 아파트 거주자는 774명으로,
다른 주거타입 대비 높습니다.

그러나 거주 타입 별 분포비율을 확인할 경우
오피스텔 거주자의 사기꾼 분포 비율이 15% 로 제일 높습니다.

이를 통해, 오피스텔에 거주하는 보험 고객 중 사기꾼 비율이 높음을 알 수
있습니다.

02 데이터 변수 제거

데이터 EDA 및 전처리

» 불필요한 DATA 제거

주택가격, FP경력,
고객등록년월
직업그룹코드2
배우자직업코드2
자녀수
막내자녀연령
최대보험료연월
최대보험료
추정가구소득1
추정가구소득2

SIU_CUST_YN	SEX	AGE	RESI_COST	RESI_TYPE_CODE	OCCP_GRP_1	TOTALPREM	MINCRDT	MAXCRDT
0	2	4	21111	20	3	146980441	6	6
0	1	5	40000	20	3	94600109	1	6
0	1	6	0	0	5	18501269	6	6
0	2	5	0	0	2	10506072	8	8
0	1	6	6218	99	3	22313040	6	6
1	2	6	11388	30	5	46522197	6	6
0	1	5	86527	20	2	151085847	6	6
0	1	5	22638	20	4	3666050	6	6
0	2	8	29018	99	0	0	6	6
0	1	6	18408	20	3	15628337	6	6
0	1	5	37222	20	4	135719262	6	8
1	1	6	8140	50	6	33261687	6	7
1	1	6	10697	40	2	0	6	6
0	1	5	23055	20	3	33336919	6	6
0	2	5	6684	12	1	77429702	6	6
0	1	6	14027	20	5	10122595	6	6
0	1	4	33333	20	5	10317997	10	8

CODE

```
colnames(cust_data)
cust_data <- subset(cust_data,
  select=-c(CTPR,FP_CAREER,CUST_RGST,
    OCCP_GRP_2, MATE_OCCP_GRP_2,
    CHLD_CNT,LTCN_CHLD_AGE,
    MAX_PAYM_YM,MAX_PRM))
```

Subset을 이용해 분석에서 제외할 변수들을 제거합니다.

» 변수 제거 수정사항

1차적으로 추정가구소득1,2를 제거했었으나,
두 변수가 xgboost 모델에 영향을 많이 주는 변수이기에
두 변수 모두 포함하여 모델을 생성했습니다.

>> 단순레이블링

Var.26	강원	경기	경남	경북	광주	대구	대전	부산	서울	세종	울산	인천	전남	전북	제주	충남	충북
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

CODE

```
temp <- subset(cust_data, select=c('CUST_ID','CTPR'))
temp$value <- 1
library(reshape2)
temp_dummy <- dcast(data=temp, CUST_ID~CTPR, fun=sum)
names(temp_dummy) <- c('CUST_ID', area.name)
cust_data <-
merge(cust_data, temp_dummy, by='CUST_ID')
cust_data <- subset(cust_data, select=-CTPR)
```

>> 적용 변수 : CTPR

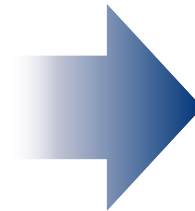
지역별 사기자 변수를 원-핫 인코딩을 이용해
지역 값을 변수열로 변경합니다.

02 데이터 파생변수 추가

데이터 EDA 및 전처리

» 입원 평균일수를 원-핫 인코딩을 통해 파생변수로 추가

인 천	전 남	전 북	제 주	충 남	충 북
0	0	0	0	0	1
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0



전 남	전 북	제 주	충 남	충 북	HOSP_DAYS
0	0	0	0	1	1
0	0	0	0	0	3
0	0	0	0	0	16
0	0	0	0	0	25
0	0	0	1	0	2
0	0	0	0	0	32
0	0	0	0	0	2
0	0	0	0	0	2
0	0	0	0	0	60
0	0	0	0	0	7

CODE

```
hospday <- aggregate(claim_data$VLID_HOSP_DAYS,
                     by=list(claim_data$CUST_ID), mean)
names(hospday) <- c("CUST_ID", "HOSP_DAYS")
hospday$HOSP_DAYS <- round(hospday$HOSP_DAYS)
cust_data <- merge(cust_data, hospday)
```

» 추가 변수 : HOSP_DAYS

CLAIMDATA에서 개별 평균 입원일수를 이용해 평균 입원일을 계산하여 기존 데이터에 평균 입원일수 데이터 열을 추가합니다.

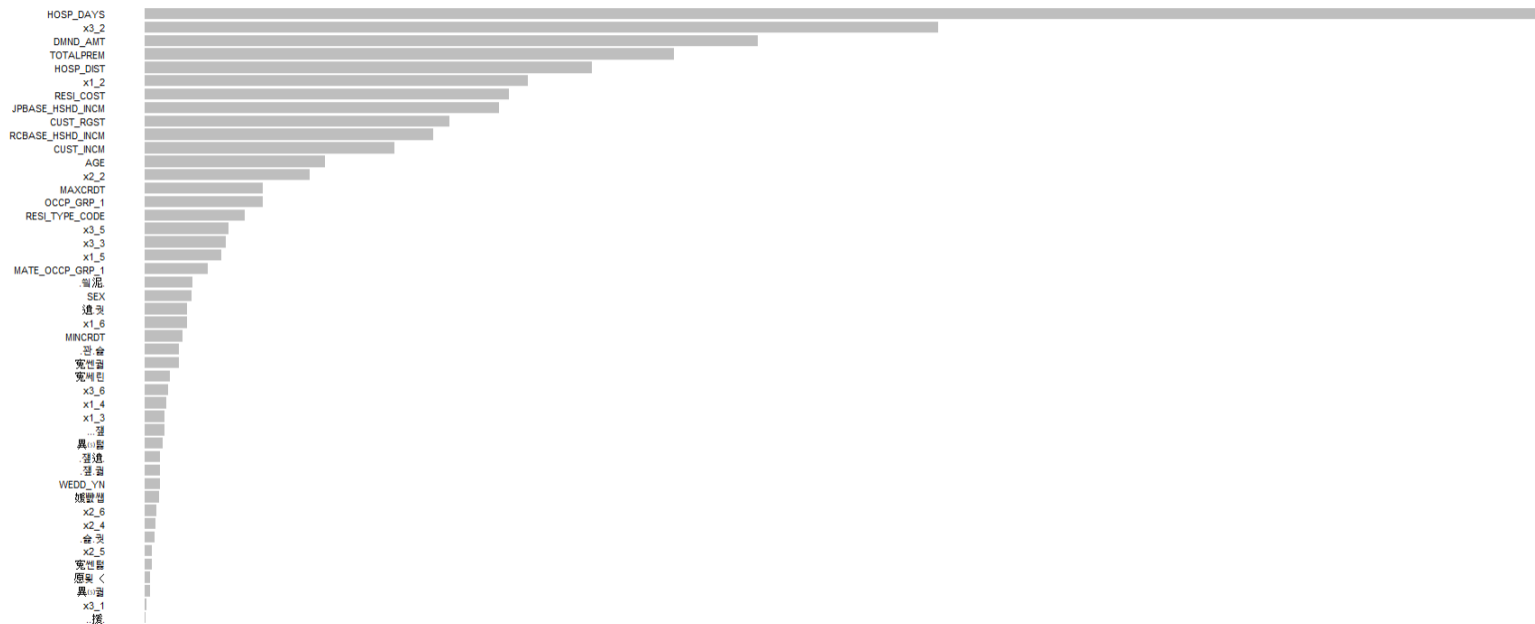
※ XGBOOST 모델링에 영향을 가장 많이 준 변수

데이터 EDA 및 전처리

A horizontal number line with a red dot at 0. The line has tick marks every 1 unit, with labels 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50. The red dot is at 0.



※ MERGE 했을 때, 열 이름이 문자로 시작해야하는데,



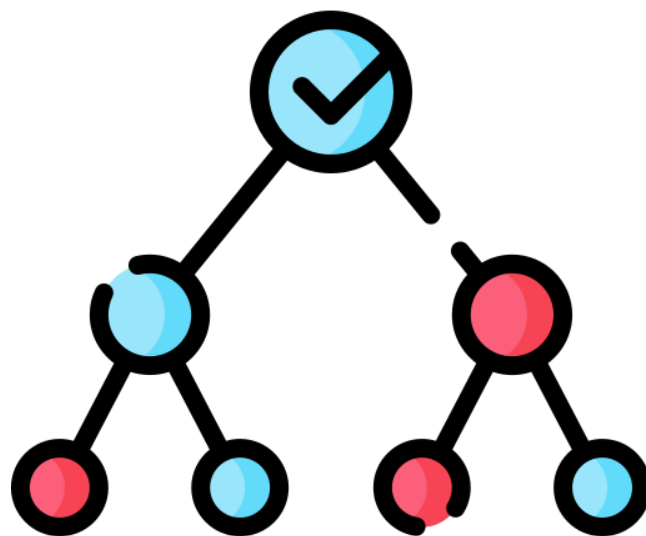
CONTENTS

03

모델링

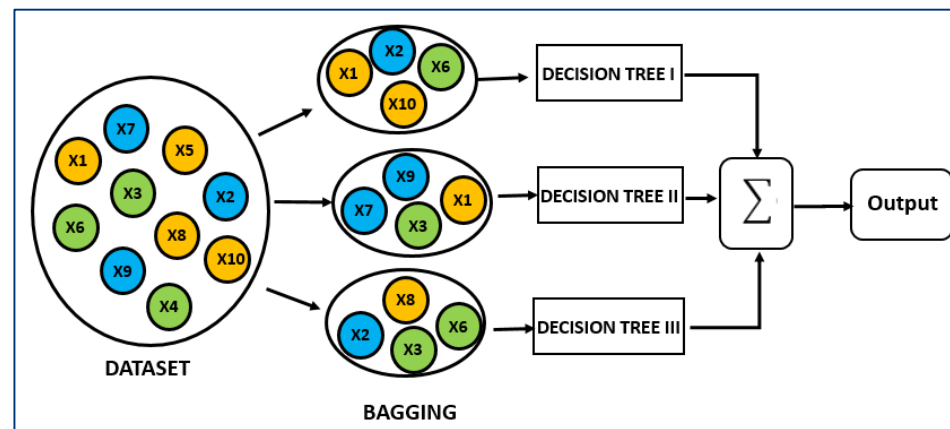
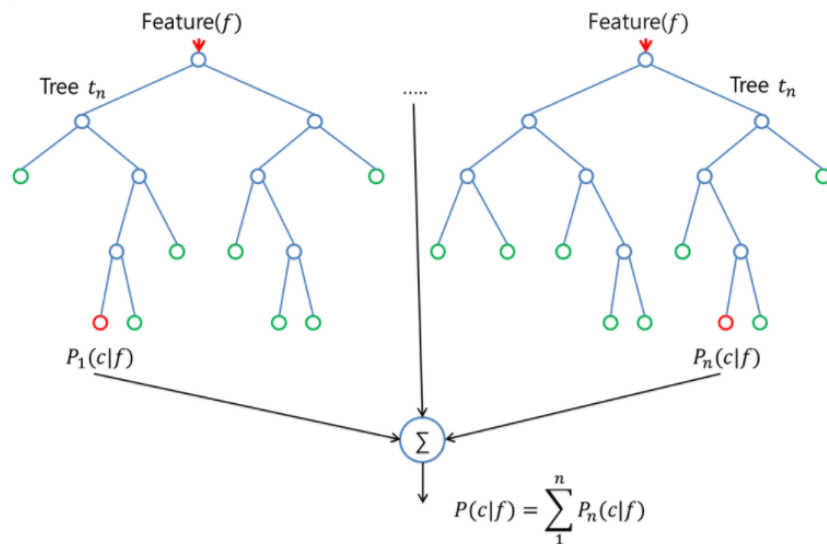


» 모델링에 사용될 3가지 모델



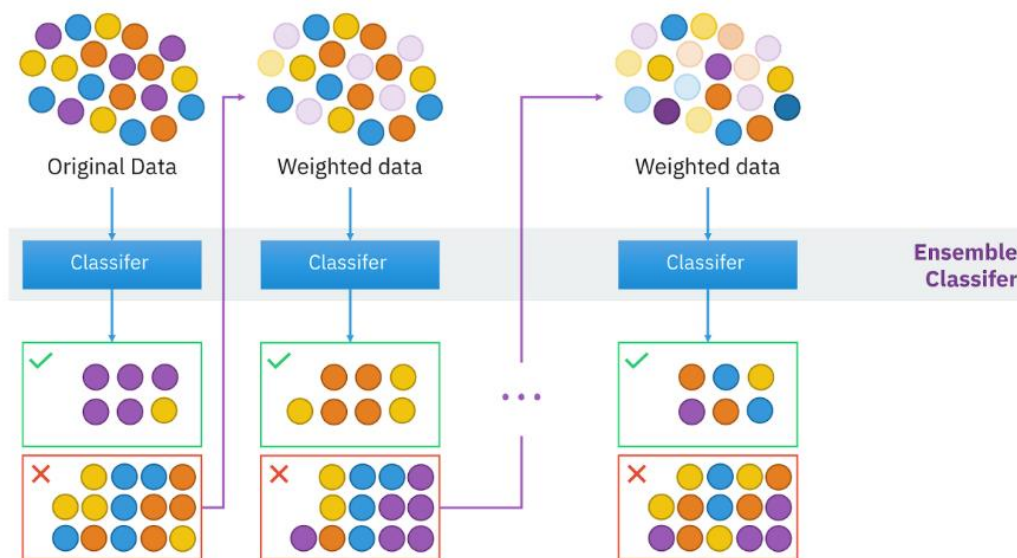
- 방법 : 여러가지 질문들을 통해 분류
- 목표 : 최소의 질문으로 정확하게 분류
- 특징 : 질문의 순서가 중요하기 때문에, 가장 영향력있는 변수를 판단하여 루트노드, 규칙노드, 리프노드로 구분.
- 약점 : 과적합 가능성이 높음

>> 랜덤 포레스트(앙상블 - Bagging)



- 방식 : 여러 개의 의사결정트리를 형성해 각 트리가 분류한 결과를 투표해 최적의 예측치를 선택
- 목표 : 의사결정트리의 과적합 문제를 해결

>> XGBoost (앙상블 - Boosting)



- 방법 : 이전 모델의 오류를 보완(가중치 부여)하여 결과값 예측모형 생성
- 특징 : 라벨이 적어 분류하기 어려운 데이터셋에 적용 가능
- 강점 : 정확도 UP / 결측치를 내부적으로 처리

>> XGBoost (앙상블 - Boosting)

랜덤 포레스트

```
Call:
  randomForest(formula = SIU_CUST_YN ~ ., data = train, ntree = 300,      na.action = na.omit)
    Type of random forest: regression
    Number of trees: 300
No. of variables tried at each split: 17

    Mean of squared residuals: 0.05319738
      % Var explained: 37.74
> |
```

의사결정나무

```
1) X3_2 <= 6; criterion = 1, statistic = 1948.915
2) X1_2 <= 2; criterion = 1, statistic = 1211.976
3) HOSP_DAYS <= 10; criterion = 1, statistic = 455.401
4) X2_2 <= 2; criterion = 1, statistic = 116.082
5) HOSP_DAYS <= 5; criterion = 1, statistic = 75.719
6)* weights = 7407
5) HOSP_DAYS > 5
7) X1_2 <= 0; criterion = 0.999, statistic = 19.567
8) X3_2 <= 2; criterion = 0.999, statistic = 23.91
9)* weights = 1358
```

XGBoost

```
##### xgb.Booster
raw: 172.2 Kb
call:
  xgb.train(params = params, data = dtrain, nrounds = nrounds,
    watchlist = watchlist, verbose = verbose, print_every_n = print_every_n,
    early_stopping_rounds = early_stopping_rounds, maximize = maximize,
    save_period = save_period, save_name = save_name, xgb_model = xgb_model,
    callbacks = callbacks, eta = 1, objective = "binary:logistic",
    nthread = 8)
params (as set within xgb.train):
```


» 모델 평가

랜덤 포레스트

ACCURACY

0.93

PRECISION

0.72

RECALL

0.45

F1-SCORE

0.55

의사 결정 트리

ACCURACY

0.93

PRECISION

0.66

RECALL

0.44

F1-SCORE

0.53

XGBoost

ACCURACY

0.92

PRECISION

0.61

RECALL

0.49

F1-SCORE

0.54

CODE

```
f1score <- function(m){
  ac <- (m[1,1]+m[2,2])/ sum(m)
  pr <- m[2,2]/(m[1,2]+m[2,2])
  re <- m[2,2]/(m[2,1]+m[2,2])
  f1 <- 2*pr*re/(pr+re)
  print(ac) ; print(pr) ; print(re) ; return(f1) }
m <- table(test_lab, round(pred_xg))
d <- table(test$SIU_CUST_YN, round(pred_tr))
r <- table(test$SIU_CUST_YN, round(pred_rf))
f1score(m) #xgboost ; 1score(d) #나무 ; f1score(r) #랜덤트리
```

모델 평가 결과 다음과 같은 결과가 나왔습니다.

- 정확도 : 랜덤포레스트 > 의사결정트리 > XGBoost
- 정밀도 : 랜덤포레스트 > 의사결정트리 > XGBoost
- 재현율 : 의사결정트리 > XGBoost > 랜덤포레스트
- F1-SCORE : 랜덤포레스트 > XGBoost > 의사결정트리

>> 랜덤포레스트 모델링을 이용한 TEST_SET 분석

- 보험사에서 제공한 고객정보와 보험청구데이터를 처리하여 사기꾼 예측 모델을 구축
- 사기꾼 판별여부를 모르는 1,645 데이터에 모델을 적용하면, 78명을 사기꾼이라고 예측

CODE

```
f_test <- subset(cust_data, subset=
(cust_data$DIVIDED_SET==2))
pred_rf <- predict(rf_model, f_test)
result <- as.data.frame(round(pred_rf))
table(result)
```

Thank You
