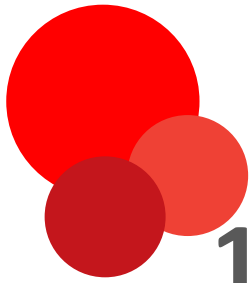
Santander Product Recommendation

2016.12.13

Korean Wave(정현진, 김진아, 전병훈)

Contents



1. 프로젝트 개요

Project Objective

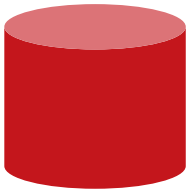
프로젝트 개요

Kaggle에서 스페인 글로벌 은행 Santander가 과제로 제시한 'Santander Product Recommendation' Competition에 참여하여 글로벌 데이터 사이언티스트와 경쟁해보고자 함

Santander ✖ kaggle

"Santander Product Recommendation"

2015.01 – 2016.05
Data



과거 고객의 특성, 행동 데이터를 기반으로
다음 달에 어떤 상품을 구매할 것인지 예측

2016.06
Prediction

?

Introduce Our Team

Kaggle에서 한류를 일으키자!

" Korean Wave "

정 현 진	김 진 아	전 병 훈
EDA, 변수 생성, 데이터 전처리, 알고리즘 적용	시각화, 알고리즘 적용, 평가 방법 연구	EDA, 전처리, 알고리즘 공부&적용, 캐글 제출

Santander

2. 데이터 개요 및 변수 설명

Data Introduction

데이터 개요 및 변수 설명

데이터 셋	데이터 종류	스페인 Santander 은행 고객 정보를 나타내는 정형 데이터 셋
	데이터 기간	2015 1월 ~ 2016 5월 : Training set 2016 6월 : Test set
	데이터 크기	약 2GB
		13,647,309행 48열(고객 정보24 / 상품 가입 정보 24)

데이터 개요

	fecha_dato	ncodpers	ind_empleado	pais_residencia	...	ind_ahor_fin_ult1	ind_aval_fin_ult1	ind_cco_fin_ult1	...
1:	2015-01-28	1049129	N	ES		0	0	1	
2:	2015-01-28	1049988	N	ES		0	0	1	
3:	2015-01-28	1049457	N	ES		0	0	1	
4:	2015-01-28	1055176	N	ES		0	0	0	
5:	2015-01-28	1052983	N	ES		0	0	1	
6:	2015-01-28	1052813	N	ES		0	0	1	

Santander

Feature Description

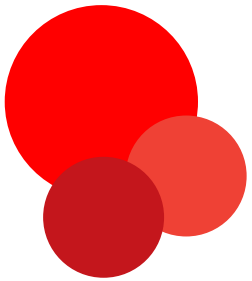
데이터 개요 및 변수 설명

변수 명	영어 명	설명
fecha_dato	date	발생 날짜
ncodpers	cust_id	고객 아이디
ind_empleado	emp_index	은행 직원 여부
pais_residencia	cust_country	거주 국가
sexo	sex	성별
age	age	나이
fecha_alta	cust_firstdate	최초 상품 개설 날짜
ind_nuevo	new_cust	지난 6 개월 동안 새로 등록된 고객 여부
antiguedad	cust_seni	고객 가입 기간
indrel	cust_pri	고객 등급 변화 여부
ult_fec_cli_1t	cust_pri_date	주요고객 등급이었던 최종 날짜
indrel_1mes	cust_type	고객 등급
tiprel_1mes	cust_relation_type	고객 유형
indresi	residence_index	은행이 속한 국가에 거주 여부
indext	foreigner_index	외국인 여부
conyuemp	spouse_index	직원 배우자 여부
canal_entrada	channel	가입 경로
indfall	deceased_index	생존 여부
tipodom	tipodom	고객 주거지 주소
cod_prov	location_code	거주 지역 코드
nomprov	location_name	거주 지역명
ind_actividad_cliente	activity_index	활성화 고객 여부
renta	income	가구 총 소득
segmento	segment	고객 분류

Feature Description

데이터 개요 및 변수 설명

변수 명	영어 명	설명
ind_ahor_fin_ult1	Saving Account	저축 계좌
ind_aval_fin_ult1	Guarantees	지급보증
ind_cco_fin_ult1	Current Accounts	당좌예금
ind_cder_fin_ult1	Derivada Account	파생 계좌
ind_cno_fin_ult1	Payroll Account	급여계좌
ind_ctju_fin_ult1	Junior Account	주니어 통장
ind_ctma_fin_ult1	Más particular Account	특별 상품1
ind_ctop_fin_ult1	particular Account	특별 상품2
ind_ctpp_fin_ult1	particular Plus Account	특별 상품3
ind_deco_fin_ult1	Short-term deposits	단기예금
ind_deme_fin_ult1	Medium-term deposits	중기예금
ind_dela_fin_ult1	Long-term deposits	장기예금
ind_ecue_fin_ult1	e-account	인터넷 계좌
ind_fond_fin_ult1	Funds	펀드
ind_hip_fin_ult1	Mortgage	모기지
ind_plan_fin_ult1	Pensions	연금
ind_pres_fin_ult1	Loans	대출
ind_reca_fin_ult1	Taxes	세금 납부
ind_tjcr_fin_ult1	Credit Card	신용카드
ind_valo_fin_ult1	Securities	증권
ind_viv_fin_ult1	Home Account	주택청약
ind_nomina_ult1	Payroll	급여
ind_nom_pens_ult1	Pensions	연금
ind_recibo_ult1	Direct Debit	자동이체 계좌

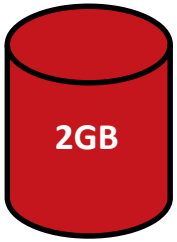


3. 분석 환경

Analysis Environmnet

분석 환경

2GB가 넘는 데이터를 Desktop 이나 Laptop 으로 R을 통해 데이터를 분석하는 과정에서 메모리 부족 문제가 발생하였으며 이를 해결하기 위한 방법으로 아마존 웹 서비스를 이용함



Desktop & Laptop

“ 메모리 부족 문제 ”

amazon web services

Rstudio Server

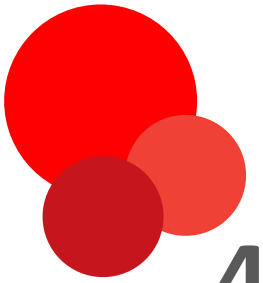
R 3.3.1

Ubuntu 16.04 LTS

Amazon EC2

vCPU: 4
Memory: 30.5 GB(RAM)
SSD : 80GB

Santander



4. SEMMA Process

4-1. Sampling

Sampling: 샘플 추출 및 데이터 분리

Raw Data

원본
Data Set

(nrow=13,647,309)

→ "956,645명의 고객 중 200,000명의 고객 데이터 Sampling"

Sampling Data

분석용
Data

(nrow=2,852,306)

Partitioning Data

Training
Data

2015–01 ~ 2016–04
(nrow=2,657,643)

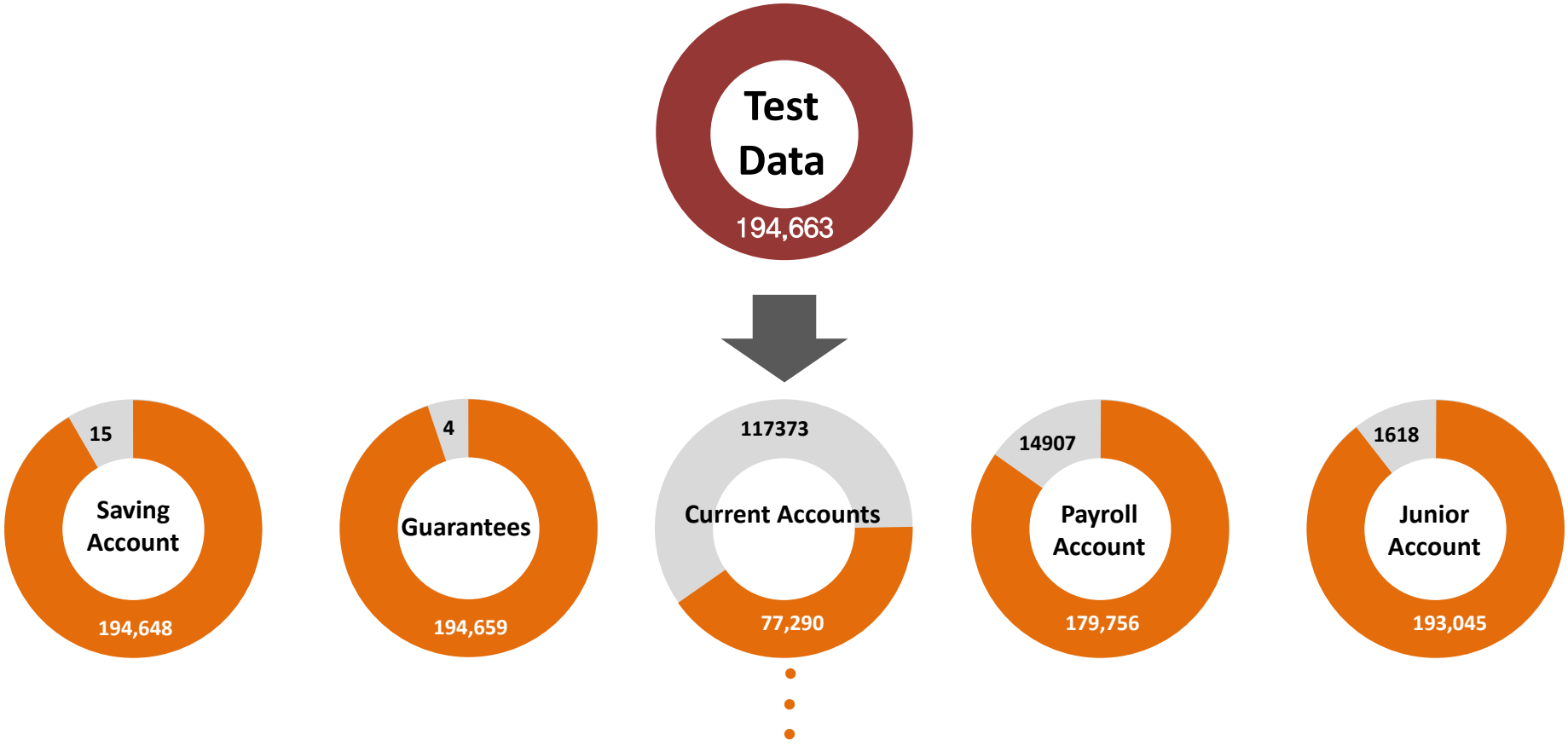
Test Data

2016–05
(nrow=194,663)

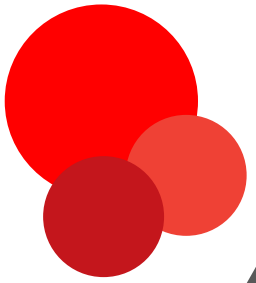
→ 2015년 1월 부터 2016년 4월까지의 Trainig Data로 모델을 생성하여, 최종적으로 2016년 6월에 고객이 가입할 상품을 예측 !

Sampling: Test Data 재가공

Test Data(2016.05)의 직전 달인 4월에 해당 상품을 보유하고 있는 고객인 경우 새로 상품을 구매한 사람이 아니므로 각 상품마다 직전 달에 상품을 보유하고 있던 사람을 제외하여 Test Data 재가공



"총 24개 상품 변수에 대해 동일한 작업 수행한 후, 구매 수가 매우 적은 7개 상품은 예측에서 제외"

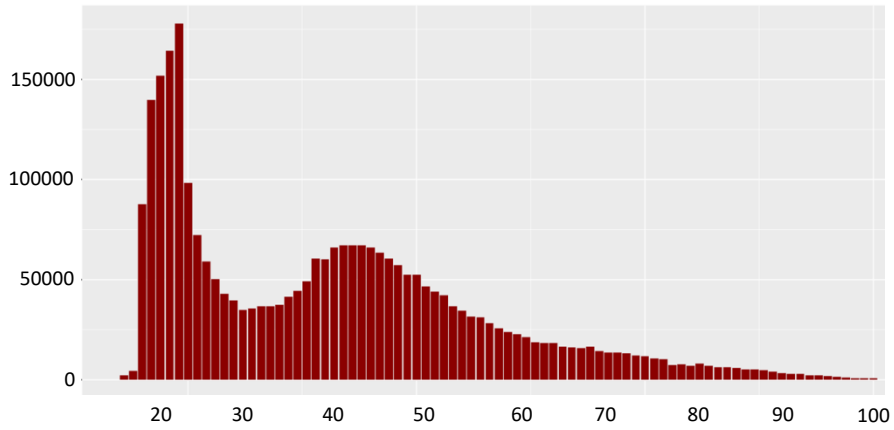


4-2.Exploring

Exploring

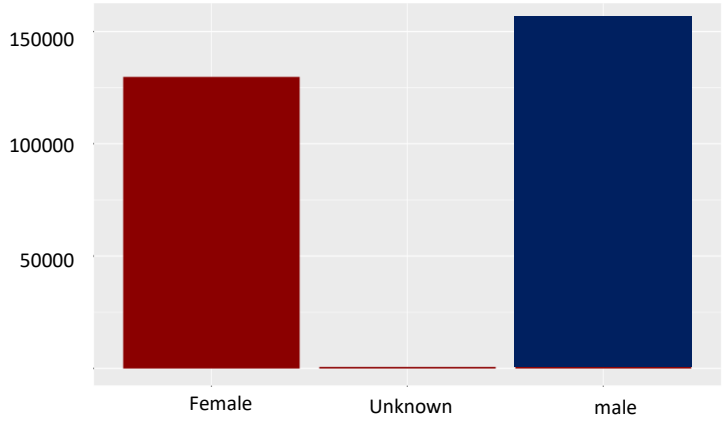
전체 고객들의 분포 특성

age



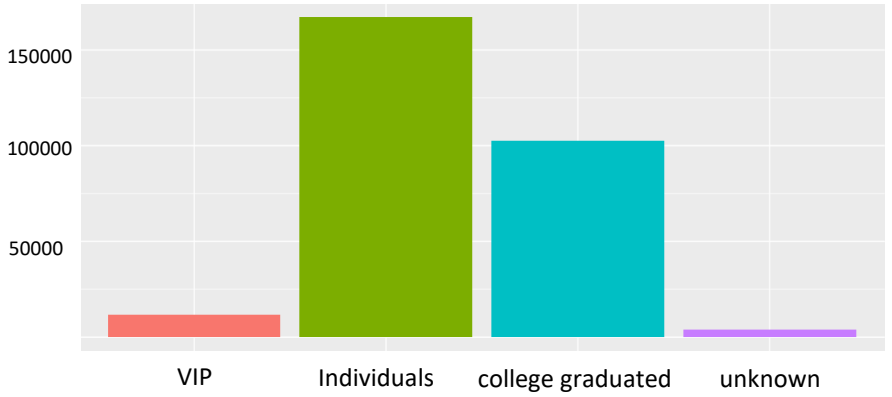
18~30세와 40~50세 사이에 높은 분포를 보임

sex



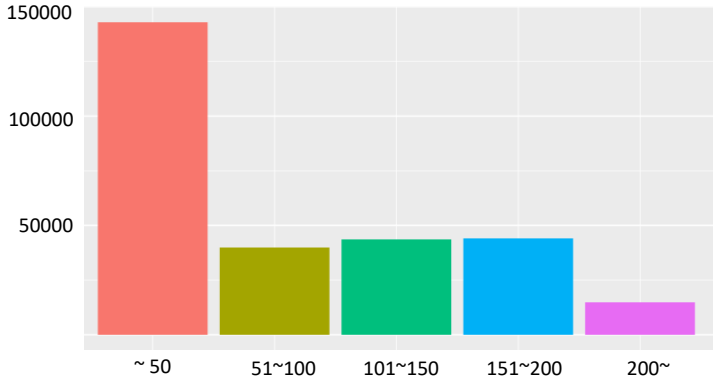
남녀의 비율은 비슷함

segment



개인고객, 사회초년생, VIP 고객 순으로 분포가 나타남

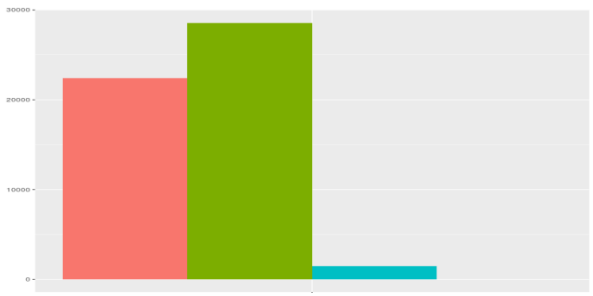
seniority



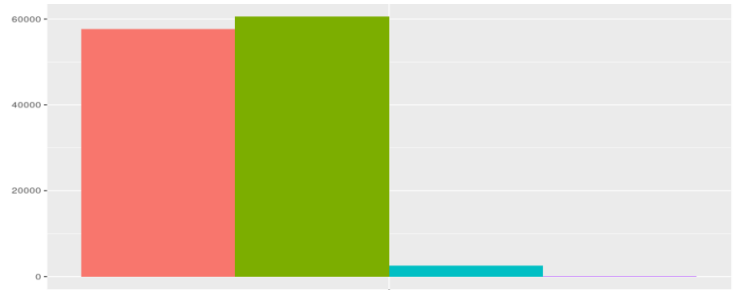
가입기간이 50개월 이하인 고객들의 분포가 높음

Exploring

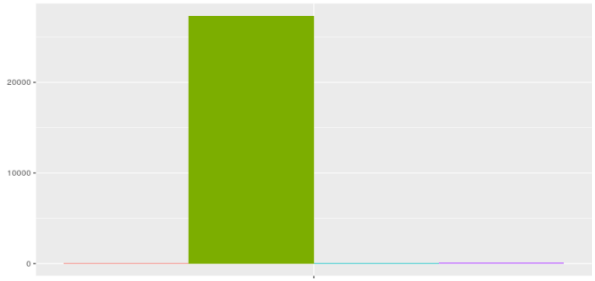
각 상품에 따른 고객 분류(Segment) 비율



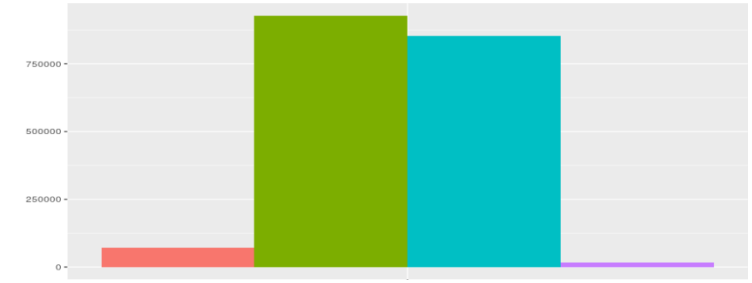
Fond



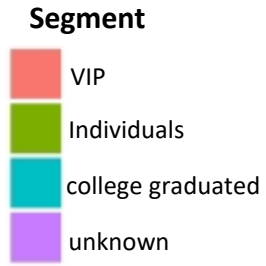
Long Term Deposit



Junior Account



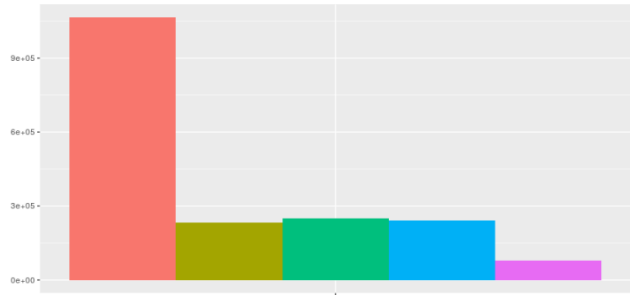
Current Account



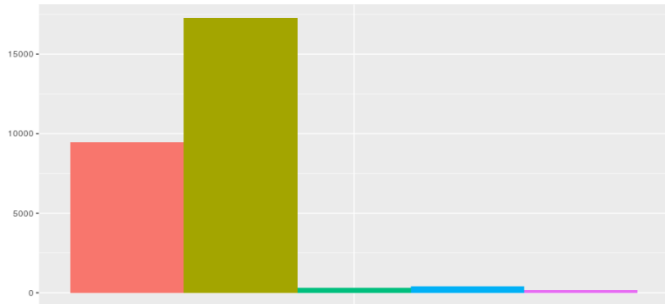
- 다른 상품들에 비해 Long-term deposit과, Fund 상품에서 VIP 고객 비율이 높음
- Junior Account 상품은 대부분 개인 고객들이 가입
- Current Account 상품에 대학 졸업자 고객들의 비율이 높음

Exploring

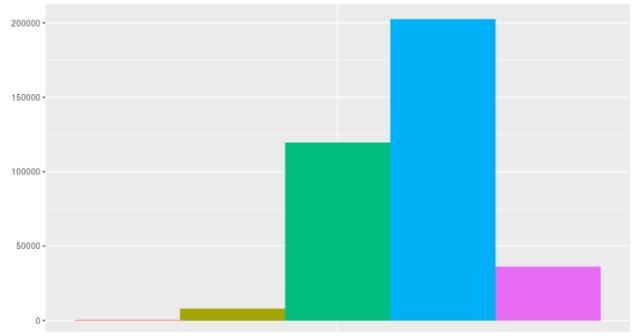
각 상품에 따른 고객 가입 기간(Seniority) 비율



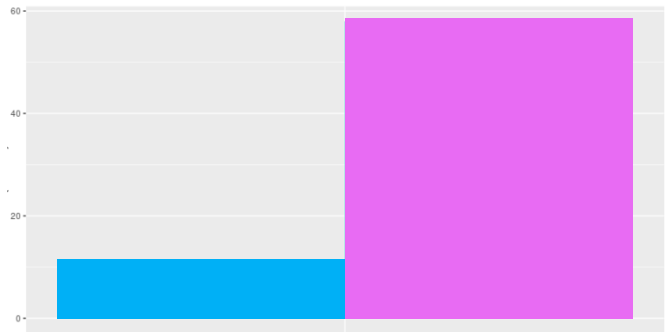
Current Account



Mas particular Account



Particular Account

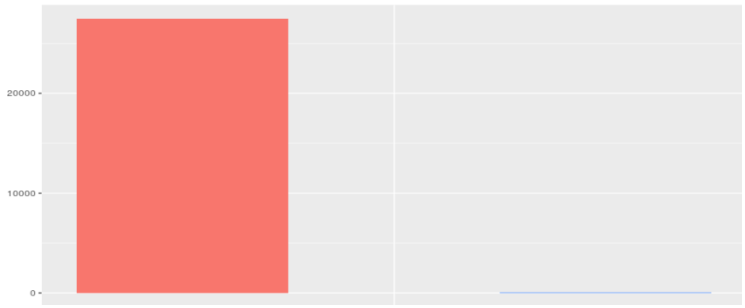


Guarantees

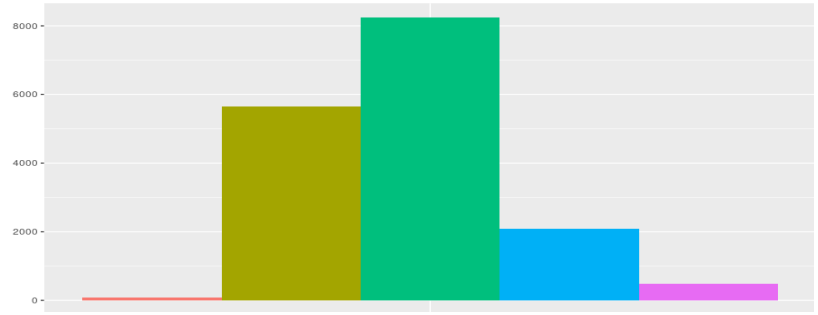
- Current Account 상품은 가입 기간이 50개월 이하인 고객의 비율이 가장 높음
- Mas particular Account 상품은 가입 기간이 51~100 개월 사이인 고객의 비율이 높음
- Particular Account상품은 가입 기간이 151~200 개월 사이인 고객의 비율이 높음
- Guarantees 상품은 상품 가입 기간이 200개월 이상인 고객의 비율이 높음

Exploring

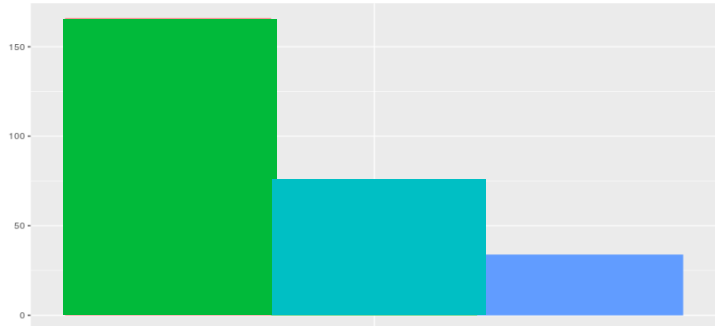
각 상품에 따른 고객 연령(age) 그룹별 비율



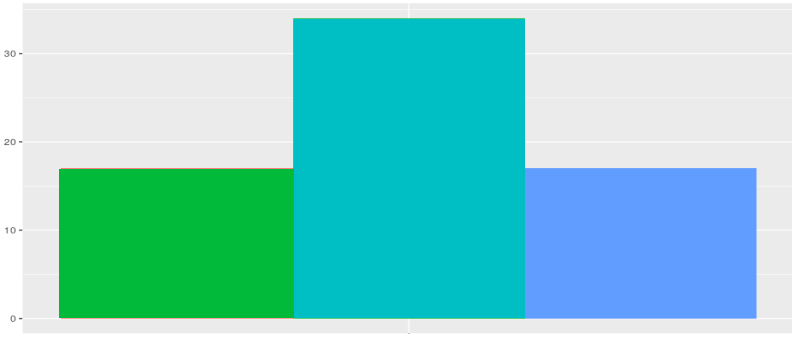
Junior Account



Mortgage

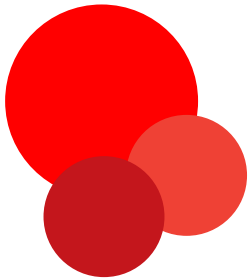


Saving Account



Guarantees

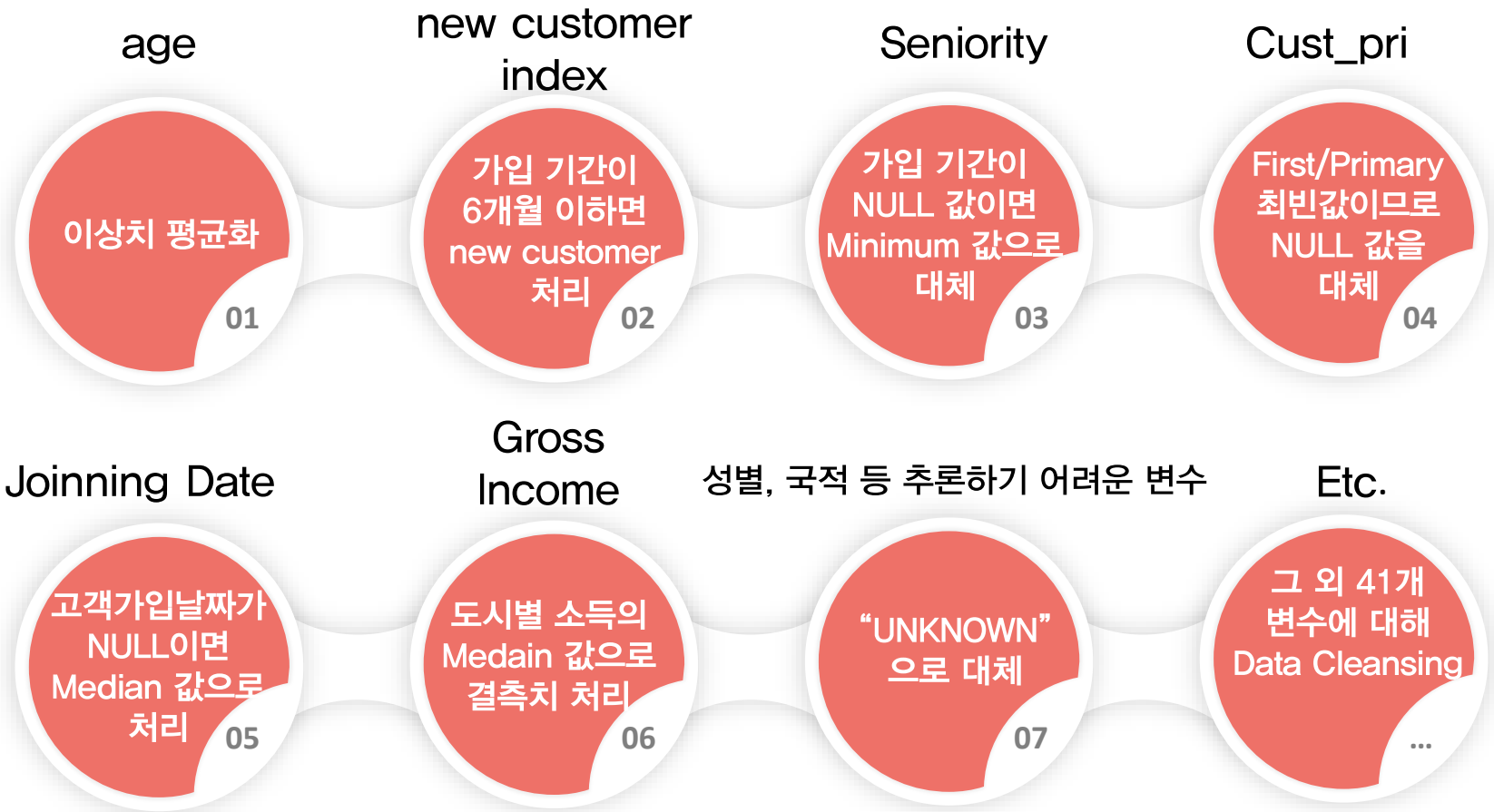
- Junior Account 상품은 25세 이하인 고객들이 가입
- Saving Account, Mortgage상품은 36~45 사이의 고객들이 많이 가입
- Guarantees 40대 중반 고객들의 비율이 높음



4-3. Modify

Data Cleansing: 이상치와 결측치 처리

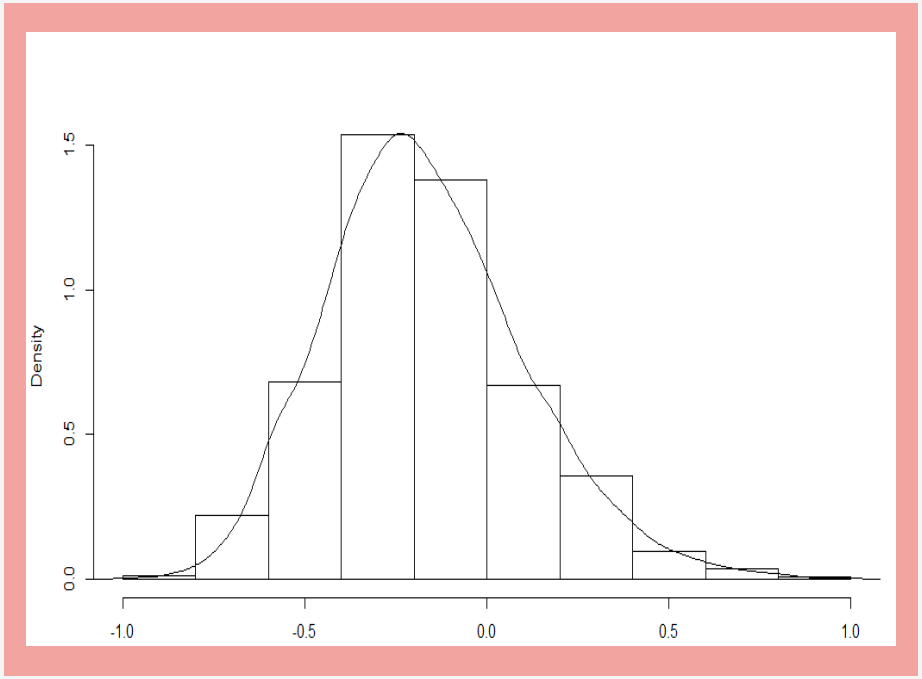
원본 데이터 SET의 이상치와 결측치(NULL)를 Average, Median, Frequency 등을 바탕으로 처리하고 값을 추정할 수 없는 변수인 경우는 “UNKNOWN” 으로 대체함



Feature Engineering: 정규화

분석을 위해서 연속형 변수들 각각을 정규화(Age, Gross income, Seniority)

나이, 소득, 가입 기간 정규화



Feature Engineering: 독립변수 재가공

Modify

연속형 변수의 범주화하는 작업과 Level이 많은 범주형 변수의 Level을 축소하거나 원 핫 인코딩을 통해 Dummy 변수로 변환하여 독립변수를 재가공

연속형 변수 범주화	Level이 많은 범주형 변수의 Level 축소	Dummy Variable 생성
Age: 6 level 범주형 변수로 변환 Gross_income: 4 level 범주형 변수로 변환 Seniority: 5 level 범주형 변수로 변환	Province name(nomprov): 5 level 범주형 변수로 변환 Channel: 5 level 범주형 변수로 변환 Reidence(pais_residencia): 9 level 범주형 변수로 변환	Date: 17개의 더미 변수 생성 Province name(nomprov): 53개의 더미 변수 생성 Residence(pais_residencia): 93개의 더미 변수 생성

“ 모델의 정확도에 따라 판단하기로 결정 ”

Santander

Feature Engineering: 새로운 변수 생성

기존의 독립변수를 바탕으로 다음과 같은 새로운 변수들을 생성

Feature Engineering: 연관규칙 분석

Modify

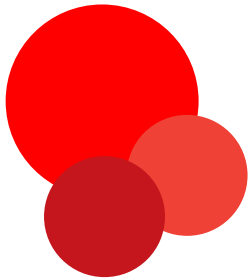
상품에 대한 연관규칙 분석으로 높은 빈발 패턴(높은 향상도)를 가지는 상품을 독립 변수로 선정

				Support	Confidence	Lift
[1]	Mortgage	=>	Direct Debit	0.0061232	0.8603142	5.960670
[2]	Payroll	=>	Pensions	0.0625038	1.0000000	**14.813015**
[3]	Pensions	=>	Payroll	0.0625038	0.9258697	**14.813015**
[4]	Payroll	=>	Payroll Account	0.0592032	0.9471936	10.451423
[5]	Pensions	=>	Payroll Account	0.0638628	0.9460006	**10.438259**

⋮

	종속 변수	추가한 독립 변수(상품)
1	Direct Debit	Credit Card, Pensions추가
2	Payroll	Pensions 추가
3	Payroll Account	Payroll 추가
4	Pensions	Payroll 추가

Santander



4-4. Modeling

Feature Choice: Near Zero Variance 제거

독립변수 중 분산 값이 0에 가까운 값(한쪽으로 치우친 변수)들을 제거

컬럼명	설명
pais_residencia	고객 거주 국가
ult_fec_cli_1t	Primary 고객이었던 최근 날짜
conyuemp	은행 직원의 배우자 여부
indfall	생존 여부

“Near Zero Variance 변수 제거(분산이 0에 가까운 값)”

Feature Choice: 변수 선정

기존의 변수와 기존 변수를 재가공한 변수, 새로 생성한 변수를 종합하여 모델을 위한 변수 선정

기존 변수 : 20 개

+

기존 변수를 변환한 변수

Province name(nomprov):
5 level 범주형 변수로 변환

Channel:
5 level 범주형 변수로 변환

Reidence(pais_residencia):
9 level 범주형 변수로 변환

Date:
17개의 더미 변수 생성

Province name(nomprov):
53개의 더미 변수 생성

+

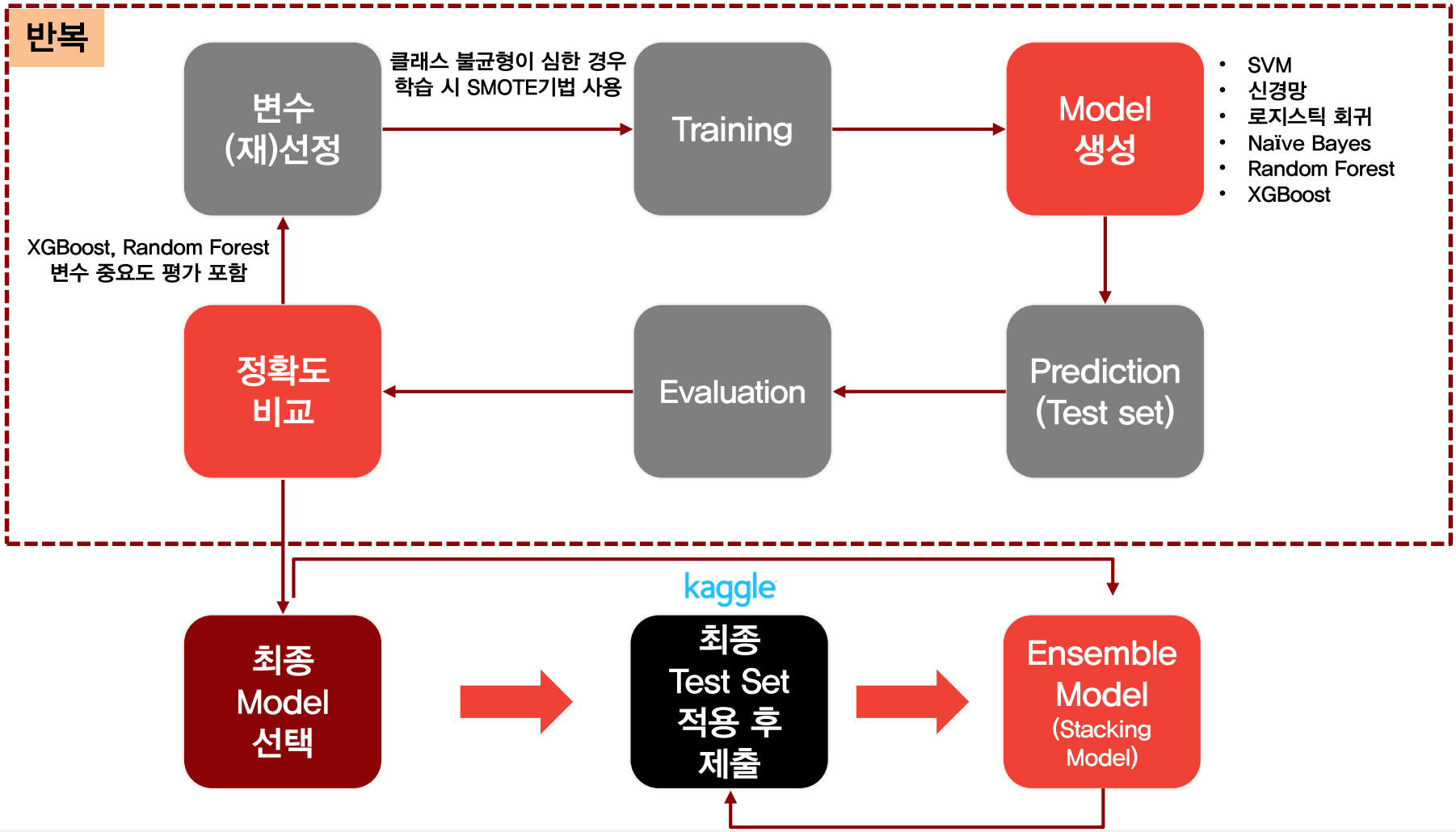
새로 생성한 변수

연령 그룹별 소득 (범주형) =〉 연령 그룹 + 소득 그룹	과거 금융 상품 구매 누적 횟수 (연속형)
지역별 소득 (범주형) =〉 고소득/중산층/저소득	금융 상품 특성별 구매 횟수 (연속형) ⇒ 예금 / 증권 / 대출 / 편의
나이, 소득, 고객 년수 Clustering (범주형) =〉 6 Level	전달에 구입한 상품 (범주형)

“ 위와 같은 최종 변수로 선정하고 모델을 만들고
모델을 활용한 변수 중요도 평가를 병행하여
높은 정확도를 보이는 변수를 최종 선택하기로 판단 ”

Modeling Process

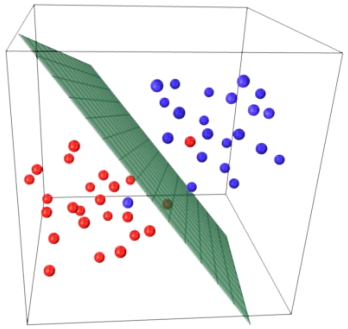
변수의 조합을 다양하게 하여 모델을 생성하고 정확도를 검증하는 과정을 반복하여 최종 모델을 선택



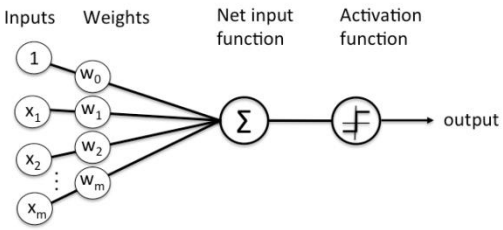
Model Description

미래의 상품 구매 여부를 예측하는 것은 분류 문제(살 것이다, 사지 않을 것이다)로 규정하고 다음과 같은 머신러닝 분류 알고리즘을 바탕으로 모델링 하기로 결정

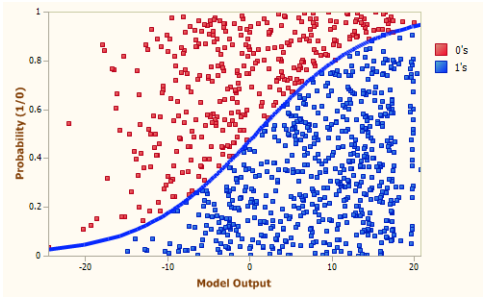
SVM



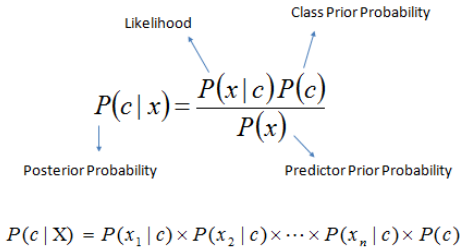
신경망



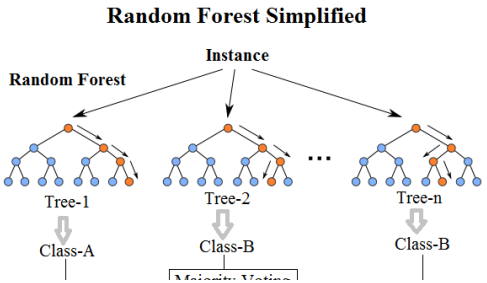
로지스틱 회귀



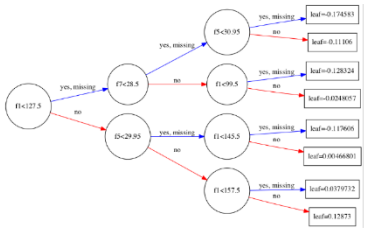
Naive Bayes



랜덤 포레스트



XGBoost



Model Choice

모델 정확도 평가는 Recall 값(실제 구매자 중 모델이 맞춘 구매자)을 사용하기로 결정하고 6가지 알고리즘에 대한 Recall 값을 상품마다 비교하여 최종적으로 사용할 알고리즘을 선택

		Prediction	
		0	1
Actual	0	TN	FP
	1	FN	TP

$$\text{Recall} : \frac{TP}{FN + TP} = \frac{\text{실제 구매자를 모델이 맞춘 수}}{\text{실제 구매한 사람들}}$$

SVM

신경망

XGBoost

"평균 Recall 값"

<

알고리즘 선택

로지스틱 회귀

랜덤 포레스트

Naive Bayes

Modeling : 최종 변수 선택

Feature Engineering을 통해 모델의 Recall 값이 가장 높았던 변수 조합을 선택

최종 모델에 사용된 변수 (22 + α)

emp_index	activity_index	z.gross_income
sex	segment	z.seniority
new_cust	age_groupBy6	prov_group
cust_pri	income_groupBy4	channel
cust_type	country	income_nomprov
cust_relation_type	resident	seniority_group
foreigner_index	z.age	age_incone
residence_index	association_rule_var	

“ 최종 변수 조합으로 22개 독립 변수와
연관규칙 분석을 적용한 상품 변수(1~2개)를 추가하여 독립변수로 선택 ”

Modeling : 최종 모델 선정

Recall 값이 높은 모델을 예측을 위한 최종 모델로 선정

Particular Account

Long Term Deposit

Payroll Account

Junior Account

Current Account

Payroll

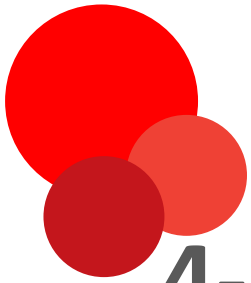
“총 17개 상품에 대해 3가지 알고리즘을 적용하여 그 중 Recall 값이 높은 모델을 선택”

Modeling: 모델 별 Recall 수치

각 상품 변수에 적용한 모델의 Recall 값

Particular Account	Particular Plus Account	Long Term Deposit	e-account	Payroll Account
Random Forest : 0.305 Naive Bayesian : 0.938 Logistic Regression : 0.290	Random Forest : 0.058 Naive Bayesian : 0.663 Logistic Regression : 0	Random Forest : 0.333 Naive Bayesian : 0.764 Logistic Regression : 0.356	Random Forest : 0.091 Naive Bayesian : 0.597 Logistic Regression : 0.108	Random Forest : 0.718 Naive Bayesian : 0.613 Logistic Regression : 0.713
Junior Account	**Más Particular Account**	**Funds**	**Pensions**	**Loans**
Random Forest : 0.7 Naive Bayesian : 0.4 Logistic Regression : 0.3	Random Forest : 0.917 Naive Bayesian : 0.904 Logistic Regression : 0.949	Random Forest : 0.533 Naive Bayesian : 0.533 Logistic Regression : 0.466	Random Forest : 0 Naive Bayesian : 0.166 Logistic Regression : 0.166	Random Forest : 0.666 Naive Bayesian : 0.666 Logistic Regression : 0
Taxes	**Credit Card**	**Securities**	**Pensions2**	**Payroll**
Random Forest : 0.034 Naive Bayesian : 0.258 Logistic Regression : 0.017	Random Forest : 0.038 Naive Bayesian : 0.671 Logistic Regression : 0.002	Random Forest : 0 Naive Bayesian : 0.75 Logistic Regression : 0	Random Forest : 0.969 Naive Bayesian : 0.963 Logistic Regression : 0.969	Random Forest : 0.967 Naive Bayesian : 0.812 Logistic Regression : 0.977
Direct Debit	**Current Account**			
Random Forest : 0.169 Naive Bayesian : 0.541 Logistic Regression : 0.204	Random Forest : 0.841 Naive Bayesian : 0.609 Logistic Regression : 0.919			

Random Forest : 0.431 Naive Bayesian: 0.638 Logistic Regression : 0.378



4-5. Assessment

Making Submission: MAP@7 평가

Assessment

예측한 결과를 Kaggle Competition에 제출하고 MAP@7 방식에 따라 결과를 Scoring

예측 모델

- Random Forest
- Naive Bayes

2016.06

최종 Test Set

상품 구매 예측

Kaggle 평가방식:
(Mean Average Precision)

$$MAP@7 = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{min(m,7)}\sum_{k=1}^{min(n,7)}P(k)$$

EX)

고객ID	예측 상품 7개	실제 구매한 상품 (정답)	Precision	Average Precision
1	A, B, C, D, E, F, G	B	0.5	0.5 / 1(정답 개수)
2	A, B, C, D, E, F, G	A	1	1 / 1 (정갑 개수)
3	A, B, C, D, E, F, G	G	0.14	0.14 / 1(정답개수)

$$MAP@7 = \frac{0.5 + 1 + 0.14\ (\text{Sum of Average Precision})}{3\ (\text{전체 고객 수})} = 0.2466$$

Santander

Scoring Result: Naive Bayes, Random Forest

Naïve Bayes 와 Random Forest로 예측한 결과를 MAP@7으로 평가한 결과는 다음과 같음

Naive Bayes

963	↓149	Himanshu Jain	0.0143527	4
964	↓149	SonyaS	0.0143443	2
965	new	abc1234	0.0141137	6
966	↓149	Gene D'Angelo	0.0141034	5
967	↓148	Junyan Gao	0.0140635	5
968	↓148	meetdestiny	0.0140341	1
969	↓148	JimmyLafontaine	0.0139944	1
970	new	idarthvader	0.0139573	8
971	new	DaveZahedi	0.0137892	3
972	new	steelrose	0.0137723	1
973	new	Korean Wave	0.0137514	3
974	new	DuraQ	0.0136747	2
975	↓152	Telcontar120	0.0136259	2
976	new	ZhejunZheng	0.013504	1
977	↓151	rowout	0.0134037	22
978	new	Cyborgus	0.0133523	1

MAP@7: 0.013514

(12/09 기준)

Random Forest

875	↓160	papadopc	0.0176327	3
876	↓160	PUTAJ	0.017586	8
877	↓157	pbys	0.0175043	3
878	↓157	TeeratP	0.0174272	5
879	↓157	strivingwen	0.0173473	3
880	new	**Korean Wave**	**0.0173095**	**4**
881	new	Data.Beyer	0.0172966	4
882	↓159	Prisma Liu	0.0172829	2
883	↓159	GabrielSantucci	0.0172424	4
884	↓159	YiZhengFang	0.017112	3
885	↓94	messiNo1	0.0171023	9
886	↓159	MonDai	0.0170922	5

Your Best Entry ↑
You improved on your best score by 0.0035581.

You just moved up 111 positions on the leaderboard. Tweet this!

MAP@7: 0.0173095

(12/09 기준)

Santander

Scoring Result: Ensemble Model

Naive Bayes 와 Random Forest를 Ensemble한 모델을 사용하여 MAP@7으로 평가한 결과 예측율이 상승하였음(두 가지 모델이 예측한 상품 구매 확률의 평균값으로 구매 우선 순위를 재조정)

Naive Bayes + Random Forest

914	↓146	Paolo M.	0.0188127	8	Fri, 09 Dec 2016 19:06:21
915	↓156	gengh!s	0.0186866	1	Wed, 16 Nov 2016 05:34:43
916	↓156	pras	0.0186516	1	Wed, 16 Nov 2016 12:24:07
917	↓155	TuTuTu	0.0186003	2	Thu, 27 Oct 2016 19:12:56
918	↓155	Chris	0.0185913	11	Wed, 16 Nov 2016 00:17:46 (-8.2d)
919	↓155	sinosora	0.0185584	13	Tue, 08 Nov 2016 18:54:56 (-2.6d)
920	↓155	shareone	0.0185567	3	Wed, 16 Nov 2016 03:42:16
921	new	**Korean Wave**	**0.0185308**	**7**	**Sun, 11 Dec 2016 14:19:25**
922	↓155	luisfer	0.0184804	2	Tue, 15 Nov 2016 23:37:03
923	↓154	ArifAziz	0.0184492	2	Mon, 31 Oct 2016 18:43:24
924	↓154	gmobaz	0.0183876	14	Wed, 07 Dec 2016 04:35:49 (-14.1d)

Your Best Entry ↑
You improved on your best score by 0.0012213.

You just moved up 20 positions on the leaderboard. Tweet this!

MAP@7: 0.0185308

(12/11 기준)

Scoring Result: Top Ranker 점수 확인

함께 경쟁하고 있는 상위 Ranker들의 점수를 확인하고 결과를 비교해보았음

Top 10 Ranker

1	–	Tom Van de Wiele *	0.0309537	88
2	↑6	Jared Turkewitz *	0.0307929	112
3	↑6	Alejo y Miro *	0.0307807	121
4	↓2	In Public Leaderboard We Trust	0.0307772	51
5	–	idle_speculation	0.0307696	46
6	↓3	lb overfitting snail	0.030758	71
7	↓3	yoniko	0.0307392	47
8	↓2	BreakfastPirate	0.0306229	92
9	↓2	Jack (Japan)	0.0305614	50
10	–	Sameh Faidi	0.0305373	128

→ Research Engineer at Google DeepMind

→ PhD in Physics University of Minesota, BS in Physics MIT

→ Data Scientist at Donoho Anlytics

→ Data Scientist at Nokia

12/09 기준

"MAP@7: 0.0305 이상의 Score가 선두 그룹"

감사합니다

Korean Wave(정현진, 김진아, 전병훈)