비트캠프

신용카드
신규 개설 여부 예측 모델 분석

TEAM E조: 신용불량자들
진동선(팀장), 박상현, 권아영, 김재원, 오연수

목차

01 프로젝트 개요

프로젝트 주제 및 선정 배경

- 신용카드를 신규 개설 할 수 있는 고객을 분류하여, 신용카드 개설에 높은 관심을 보이는 고객을 식별, 예측하기 위한 모델을 연구하였습니다.

프로젝트 개요

- 교육과정에서 배운 머신러닝/딥러닝 파이썬 회귀분석을 통해 신규 카드 개설 여부에 대한 정확성을 학습하는 모델을 개발하였습니다.

프로젝트 구조

- 데이터 전처리 및 데이터 시각화 > 모델 선정 > 최적의 모델/파라미터 탐색 > 분석 및 예측

기대 효과

- 데이터 분석을 통해 신규 카드 개설에 관심이 있는 고객들을 식별함으로써,
  은행이 신용카드 개설 여부를 예측하는 데에 효과적으로 활용하는데 도움을 줄 수 있습니다.

데이터 분석

1) 데이터셋 소개

2) 전체 데이터 분포 (pairplot)

3) 라벨 인코딩 적용하기

4) 상관계수 히트맵 분석(heatmap)

모델 선정 및 분석

1) 머신러닝과 딥러닝

2) 머신러닝 - 상관계수에 따른 acc 수치

3) 머신러닝 - boosting에 따른 acc 수치

4) 머신러닝 - 분류기 별 acc 수치

02 프로젝트 팀 구성 및 역할

훈련생	역할	담당업무
진동선	팀장	데이터 분석 및 정규화
박상현	팀원	정제된 데이터 시각화
오연수	팀원	코드 분석 및 문서화
권아영	팀원	파이썬 라이브러리, 기술 적용
김재원	팀원	적용된 기술 분석 및 개선

03 프로젝트 수행 절차 및 방법

구분	기간	활동	비고
사전 기획	7/3(월) ~ 7/4(화)	▶ 프로젝트 기획 및 주제 선정 ▶ 기획안 작성	아이디어 선정
데이터 전처리	7/5(수) ~ 7/6(목)	▶ 데이터 정제 및 정규화	
모델링	7/7(금) ~ 7/8(토)	▶ 사용할 머신러닝 모델의 종류, 구조, 특징 확인 ▶ 모델 학습 및 평가 방법 선정	팀별 중간보고 실시
기능 개선	7/9(일) ~ 7/10(월)	▶ 모델 성능과 신뢰성에 대한 평가 ▶ 최종 결과 도출	최적화, 오류 수정 결과 해석
총 개발기간	7/3(월) ~ 7/10(월) (총 1주)	-	-

1) 데이터셋 소개

신용카드 데이터셋

Happy Customer Bank은 소규모 은행으로서 저축 계좌, 자유 계좌, 투자 상품, 신용 상품 등 다양한 종류의 은행 상품을 제공한다.

이 은행은 기존 고객들에게 다른 상품을 교차 판매하기 위해 전화, 이메일, 인터넷 뱅킹, 모바일 뱅킹 등 다양한 형태의 커뮤니케이션을 활용한다.

Happy Customer Bank에서는 현재 **기존 고객들에게 신용카드를 교차 판매하고자** 한다. 은행은 이미 **신용카드 신청 자격이 있는 고객들의 그룹을 식별**했다.

이제 은행은 추천된 신용카드에 대해 **고객들의 더 높은 관심도를 파악하기 위해 도움이 필요하다.**

1) 데이터셋 소개

신용카드 데이터셋 컬럼 소개 - 11개의 컬럼

컬럼 이름	설명	타입
ID	고유한 행 식별자	object
Gender	고객의 성별	object
Age	고객의 나이 (년 단위)	int64
Region_Code	고객의 지역 코드	object
Occupation	고객의 직업 유형	object
Channel_Code	고객의 획득 채널 코드	object

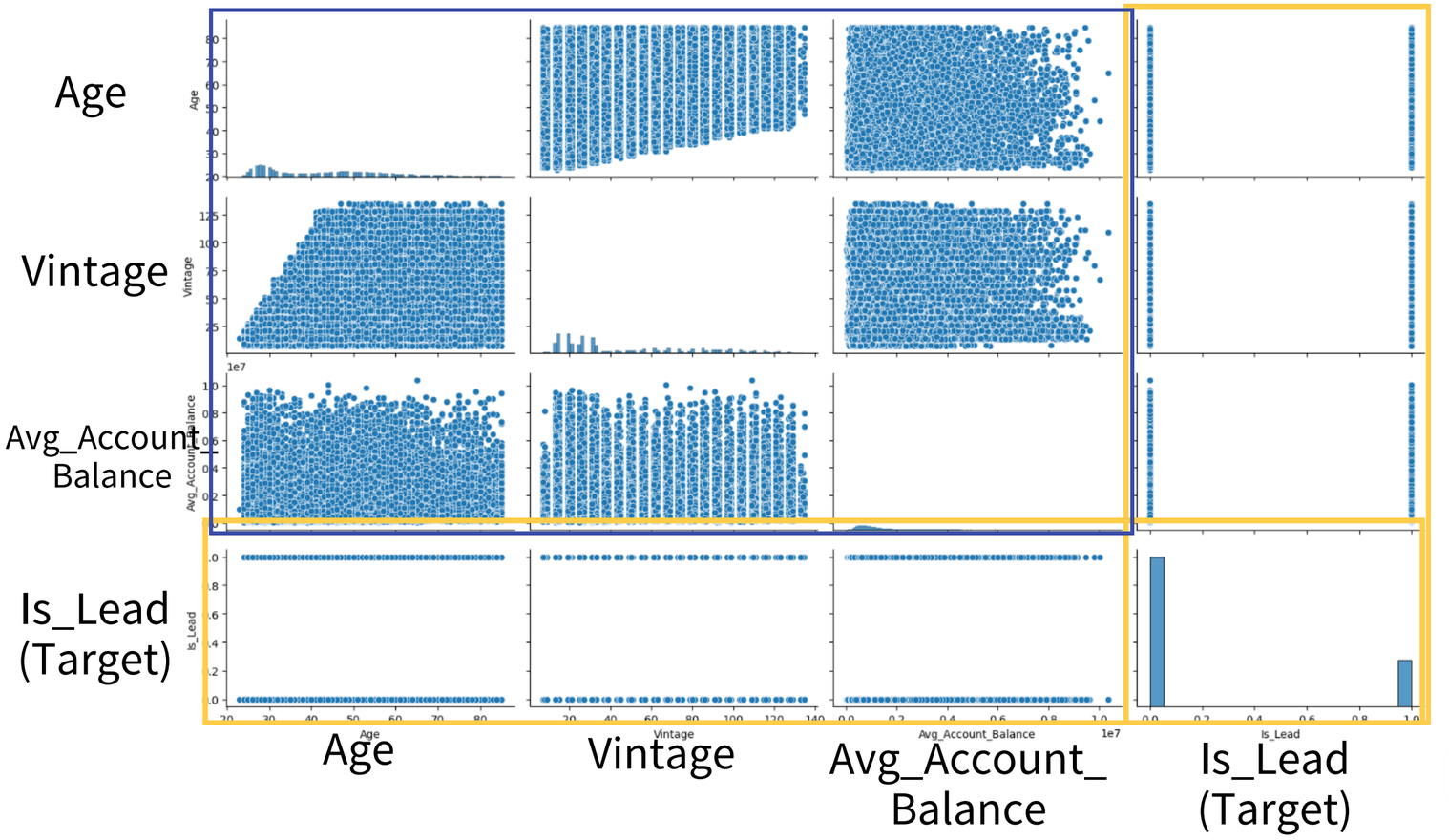
컬럼 이름	설명	타입
Vintage	고객의 고객 관계 기간 (개월 단위)	int64
Credit_Product	고객이 활성 크레딧 제품 (주택 대출, 개인 대출, 신용 카드 등)을 가지고 있는지 여부	object
Avg_Account_Balance	지난 12개월 동안의 평균 계좌 잔액	int64
Is_Active	지난 3개월 동안 고객이 활성인지 여부	object
Is_Lead(Target)	고객이 크레딧 카드에 관심이 있는지 여부 0: 고객이 관심이 없음 1: 고객이 관심이 있음	int64

1) 데이터셋 소개

데이터 예시

ID	Gender	Age	Region_Code	Occupatio	Channel_Code	Vintage	Credit_Product	Avg_Account_Balance	Is_Active	Is_Lead
NNVBBKZB	Female	73	RG268	Other	X3	43	No	1045696	No	0
IDD62UNG	Female	30	RG277	Salaried	X1	32	No	581988	No	0
HD3DSEMC	Female	56	RG268	Self_Emplc	X3	26	No	1484315	Yes	0
BF3NC7KV	Male	34	RG270	Salaried	X1	19	No	470454	No	0
TEASRWXV	Female	30	RG282	Salaried	X1	33	No	886787	No	0
ACUTYTWS	Male	56	RG261	Self_Emplc	X1	32	No	544163	Yes	0
ETQCZFEJ	Male	62	RG282	Other	X3	20		1056750	Yes	1
JJNJUQMQ	Female	48	RG265	Self_Emplc	X3	13	No	444724	Yes	0
ZMQFYKCB	Female	40	RG283	Self_Emplc	X2	38	No	1274284	No	0
NVKTFBA2	Female	55	RG268	Self_Emplc	X2	49	Yes	2014239	No	0

2) 전체 데이터 분포 (pairplot)



Seaborn 라이브러리의 pairplot 사용
정수형 타입 데이터의 분포도 분석 결과

1. 이진분류이다.
Is_Lead(Target) 행과 열의 값이 0과 1로 나뉨

2. 분포 양상을 보았을 때 모든 데이터가 고르게 분포되어 있어 이상치 처리 없이 모든 데이터를 사용하기로 했다.

3) 라벨 인코딩 적용하기

1. 라벨 인코딩 적용 전

******Labeling 전 데이터*****

	ID	Gender	Age	Region_Code	Occupation	Channel_Code	Vintage	Credit_Product	Avg_Account_Balance	Is_Active
66674	HGDFMRDO	Male	46	RG251	Self_Employed	X2	80	No	863006	No
167206	PFKGSRRE	Male	27	RG281	Self_Employed	X1	15	NaN	1464797	No
196396	CJPJZQ7W	Male	37	RG270	Self_Employed	X3	19	No	630235	No
92642	DPZH4TT9	Female	42	RG283	Other	X1	80	NaN	274787	Yes
229630	DW7XUUFR	Male	51	RG254	Self_Employed	X4	13	No	1102941	No
74441	WYAB8TSY	Male	78	RG251	Other	X2	20	No	697065	Yes
177997	DDBCACK2	Male	29	RG282	Other	X1	20	No	342614	No
171331	NNVOJAHX	Male	57	RG284	Self_Employed	X3	111	NaN	1324916	No
60174	NLKFZEUK	Male	59	RG254	Other	X3	115	NaN	1149105	Yes
101363	CGVUXHTP	Female	42	RG284	Self_Employed	X3	69	Yes	456543	Yes
83994	5VY5CMKB	Female	32	RG258	Salaried	X1	26	No	183188	No

2. 라벨 인코딩 적용 후

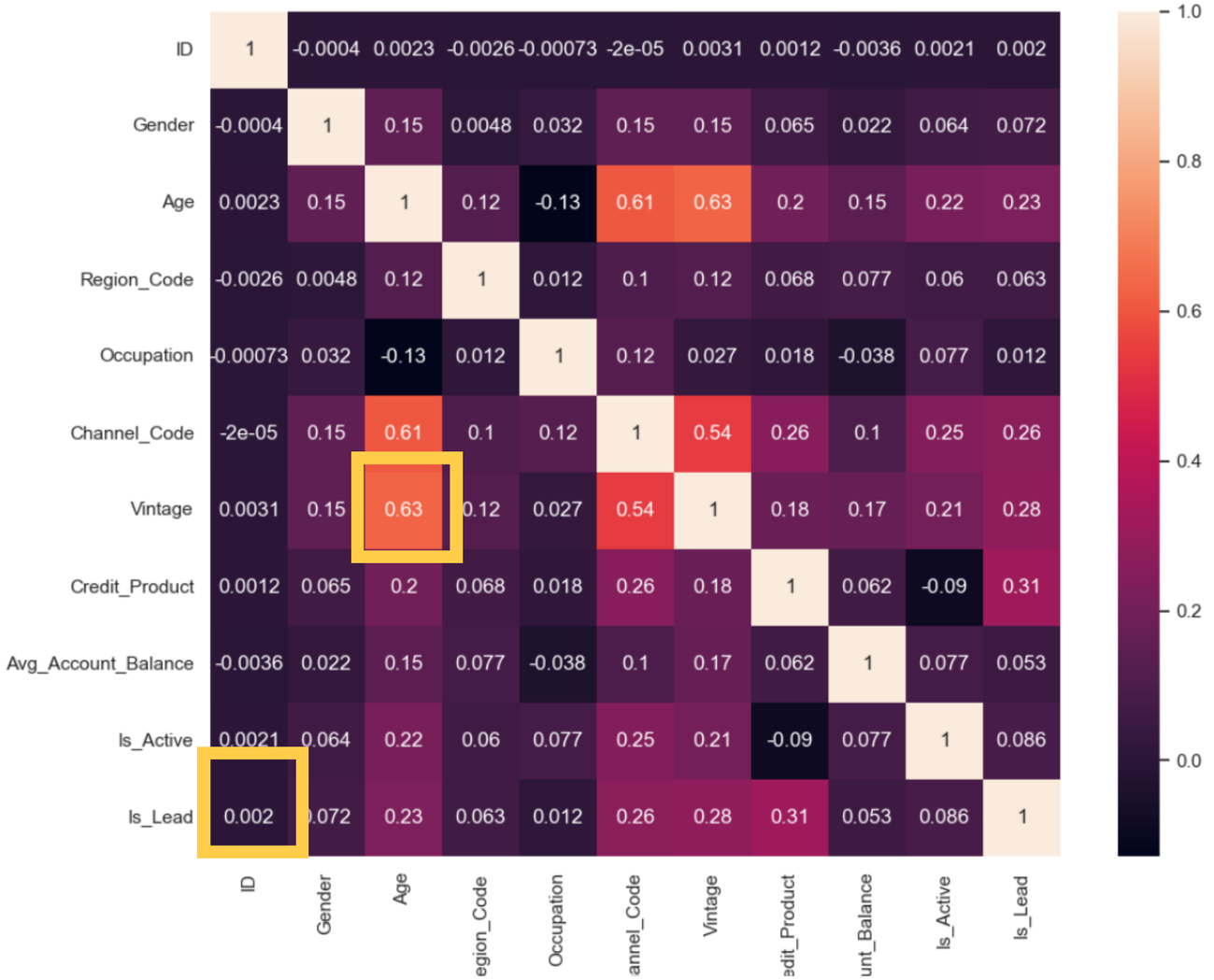
******Labeling 후 데이터*****

	ID	Gender	Age	Region_Code	Occupation	Channel_Code	Vintage	Credit_Product	Avg_Account_Balance	Is_Active
66674	12265	1	46	1	3	1	80	0	863006	0
167206	18938	1	27	31	3	0	15	2	1464797	0
196396	7007	1	37	20	3	2	19	0	630235	0
92642	8283	0	42	33	1	0	80	2	274787	1
229630	8492	1	51	4	3	3	13	0	1102941	0
74441	22970	1	78	1	1	1	20	0	697065	1
177997	7880	1	29	32	1	0	20	0	342614	0
171331	18024	1	57	34	3	2	111	2	1324916	0
60174	17930	1	59	4	1	2	115	2	1149105	1
101363	6919	0	42	34	3	2	69	1	456543	1
83994	2065	0	32	8	2	0	26	0	183188	0

NaN 값 처리 관련

- "Credit_Product" 특성은 활성 크레딧 제품 보유 여부를 나타내는데, 결측치(누락된 값)는 응답을 거부하는 의사 표현의 일부로 가정함
- 결측치(NaN) 값을 별도로 처리하지 않고, "Credit_Product" 특성에 라벨 인코딩을 적용하여 숫자로 변환함

4) 상관계수 히트맵 분석(heatmap)



Seaborn 라이브러리의 heatmap 사용 feature 간의 상관계수 분석 결과

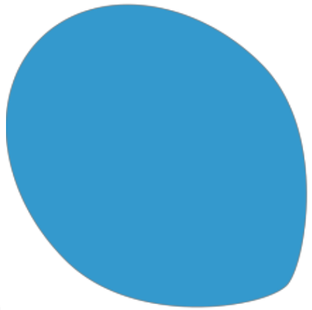
1. ID는 상관계수가 0.002로 가장 낮다.
ID는 고유한 행 식별자이므로, 데이터 분석에 중요한 정보를 제공하지 않음
-> ID 특성 제거

2. Age와 Vintage의 상관계수가 0.63으로 높다.
-> 추후 데이터 분석 시 Age와 Vintage 특성을 제거한 후 수치 비교 필요

사용 라이브러리

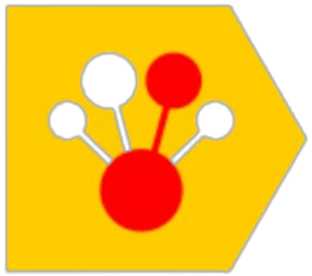
NumPy

pandas



scikit learn

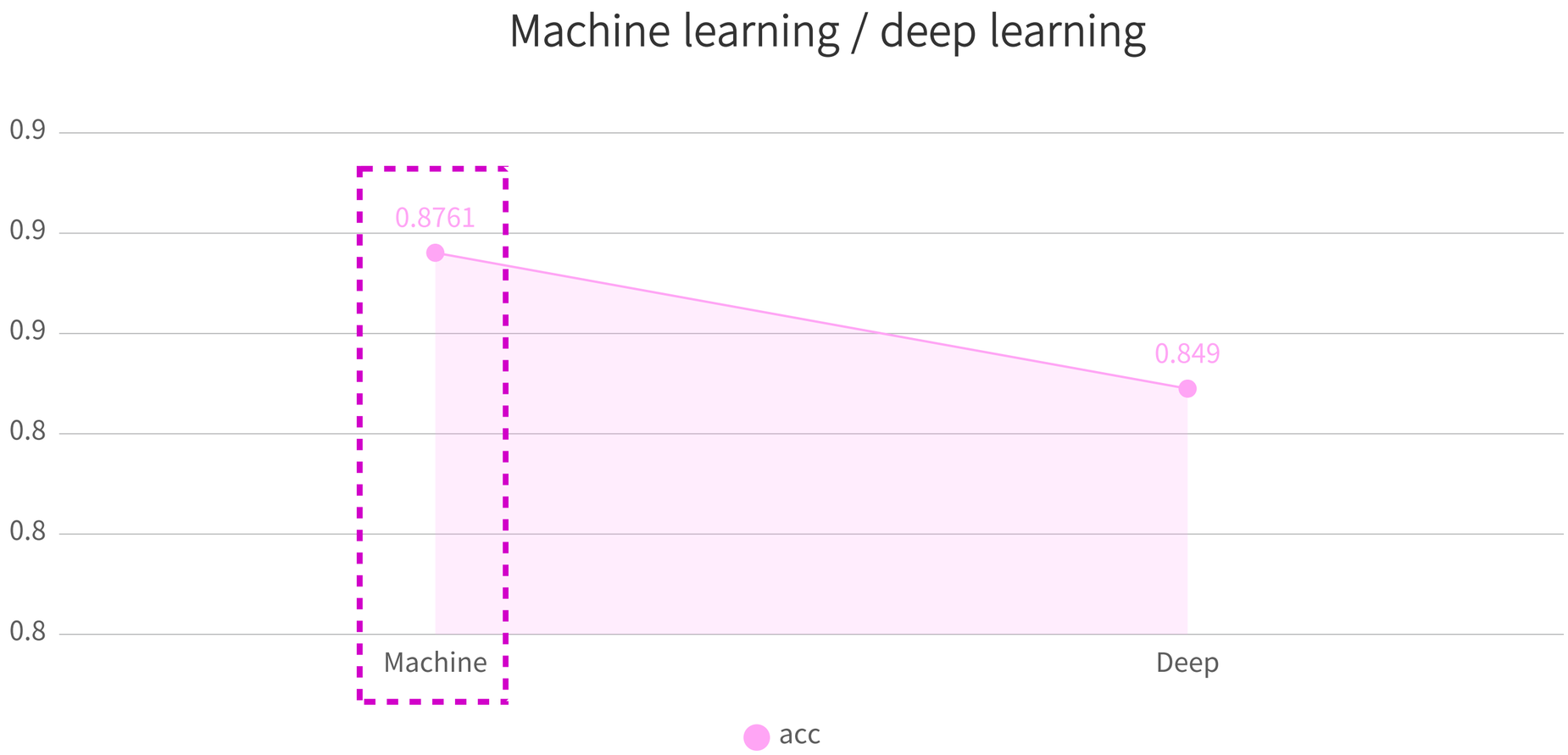
LightGBM



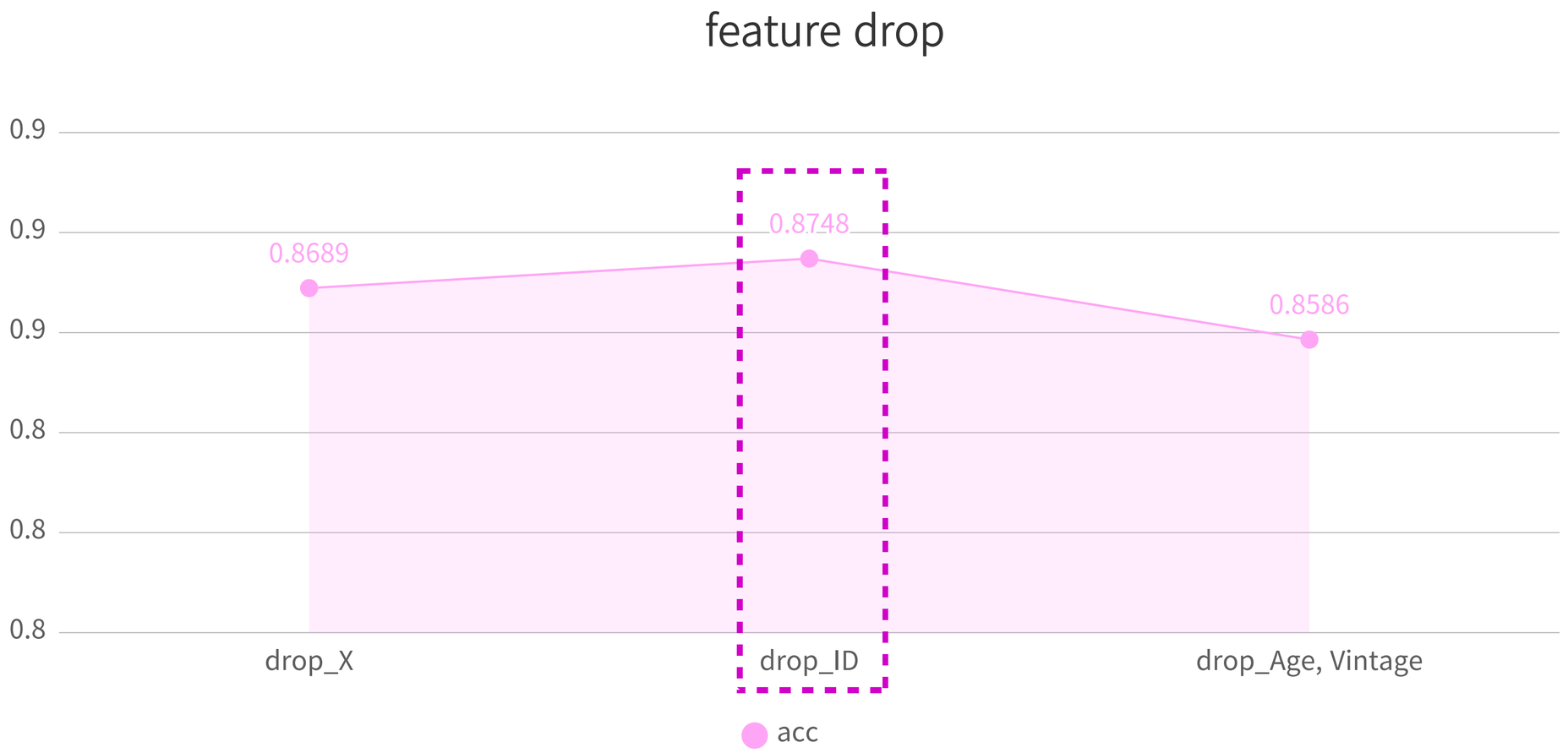
CatBoost

dmlc XGBoost

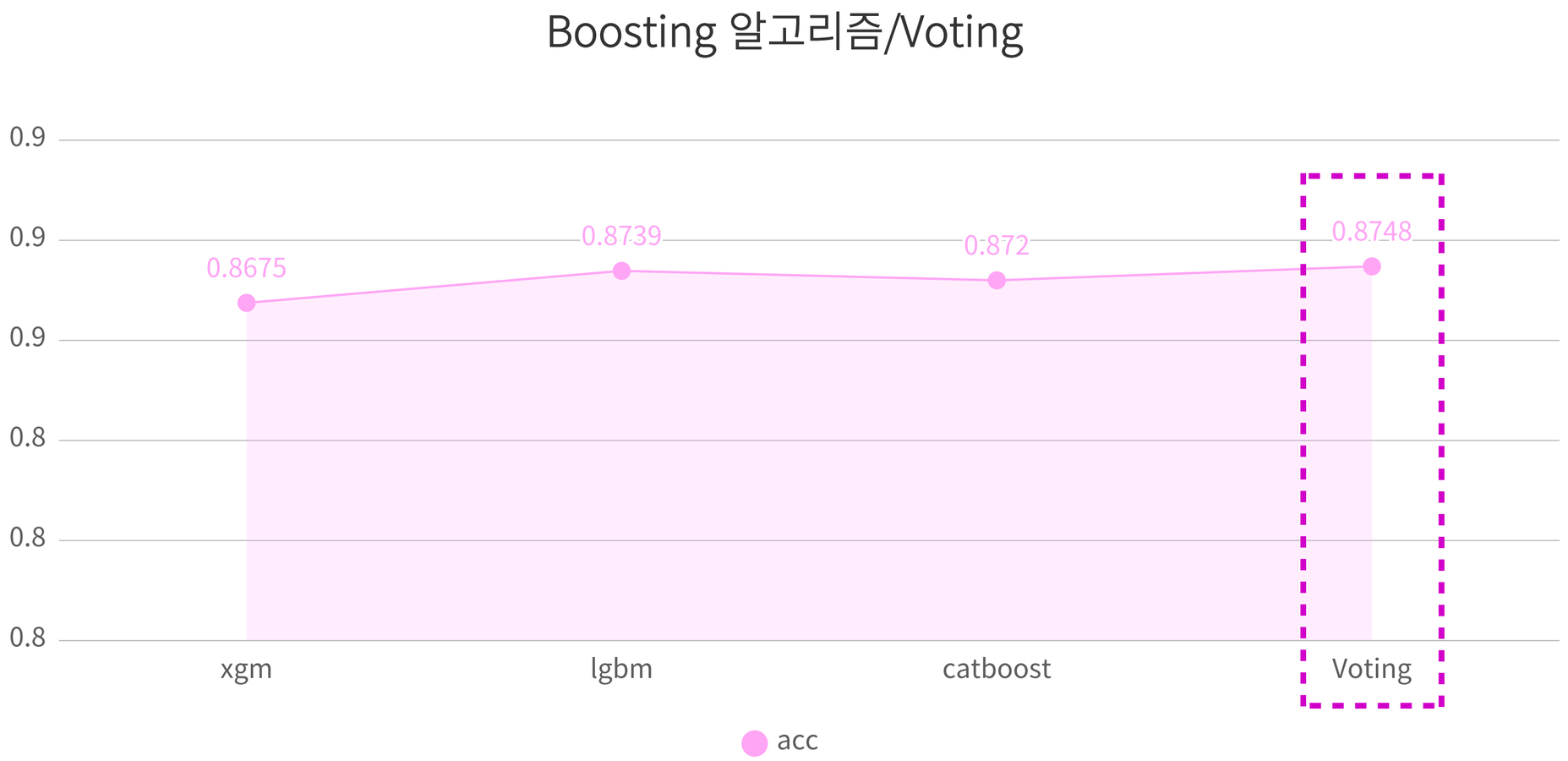
1) 모델 선정 및 분석 **머신러닝과 딥러닝 acc 수치**



2) 모델 선정 및 분석 **머신러닝 - 상관계수에 따른 acc 수치**

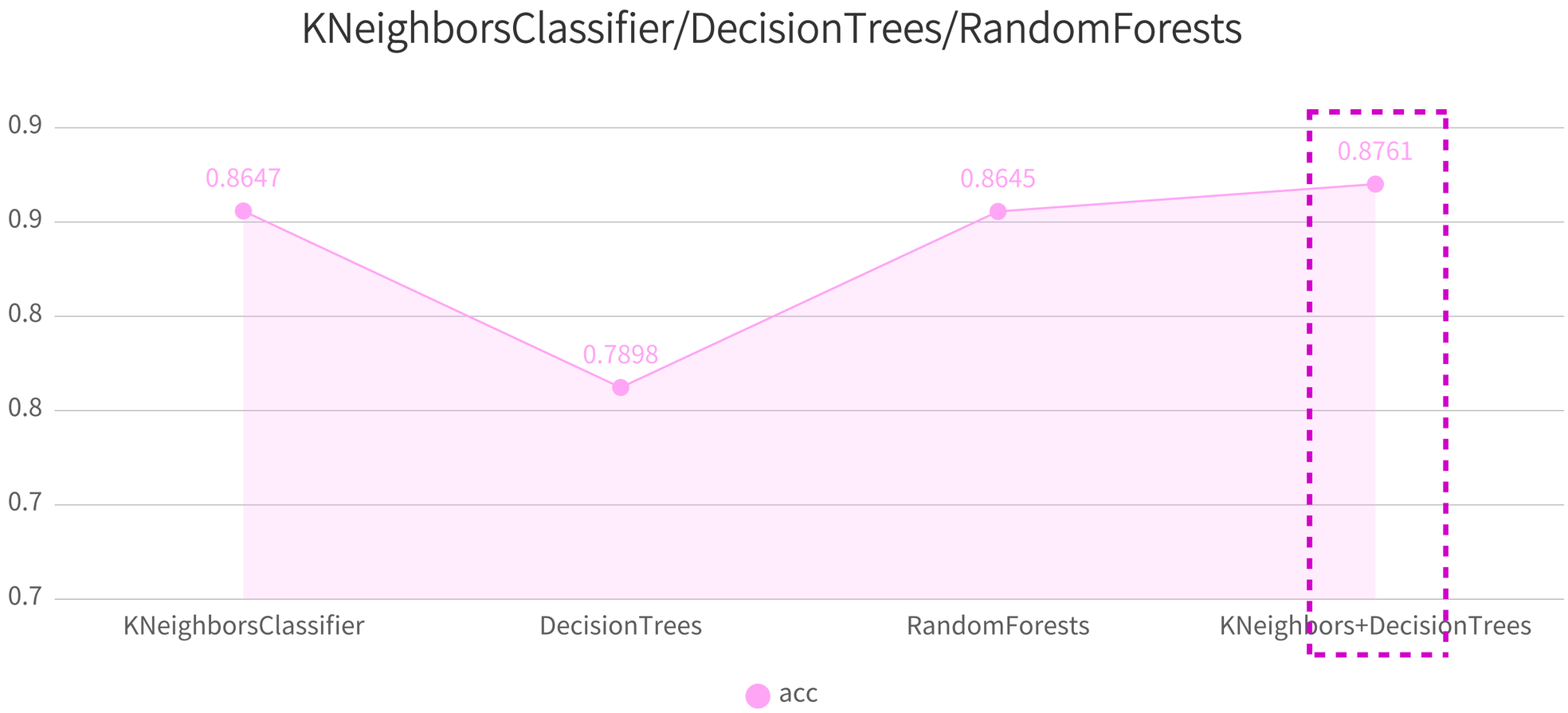


3) 모델 선정 및 분석 **머신러닝 - boosting에 따른 acc 수치**



4) 모델 선정 및 분석

머신러닝 - 분류기 별 acc 수치



04 프로젝트 수행 결과

1) 사용된 주요 모델

CatBoostClassifier

LGBMClassifier

XGBClassifier

2) 코드 설명

사용 라이브러리

데이터 로드 및 전처리

데이터 분할

파이프라인 정의 및 모델 훈련

테스트 세트 예측 및 정확도 평가

1) 사용된 주요 모델

CatBoostClassifier

범주형 특성을
처리하는 데 특화된
그래디언트 부스팅 알고리즘

LGBMClassifier

효율성과 속도가 높은
트리 기반
그래디언트 부스팅 프레임워크

XGBClassifier

확장성과 성능이 높은
최적화된 그래디언트
부스팅 라이브러리

CatBoostClassifier, LGBMClassifier, XGBClassifier 세 가지 다른 분류기가 있는 파이프라인을 생성하고,
이 분류기들은 VotingClassifier를 사용하여 결합한다.
파이프라인은 입력 특성을 표준화하는 전처리 단계로 StandardScaler도 포함한다.

2) 코드 설명

사용 라이브러리

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import VotingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.pipeline import Pipeline
from catboost import CatBoostClassifier
from lightgbm import LGBMClassifier
from xgboost import XGBClassifier
from sklearn.neighbors import KNeighborsClassifier
```

2) 코드 설명

데이터 로드 및 전처리

```
path = './team_project/credit_card_prediction/'
datasets = pd.read_csv(path + 'train.csv')
data = datasets.sample(frac=0.1, random_state=123)

x = data[['ID', 'Gender', 'Age', 'Region_Code', 'Occupation', 'Channel_Code',
        'Vintage', 'Credit_Product', 'Avg_Account_Balance', 'Is_Active']]
y = data[['Is_Lead']]

ob_col = list(x.dtypes[x.dtypes=='object'].index)
for col in ob_col:
    x[col] = LabelEncoder().fit_transform(x[col].values)

x = x.drop(['ID'], axis=1)
```

2) 코드 설명

데이터 분할

```
train_data_ratio = 0.8
x_train, x_test, y_train, y_test = train_test_split(
    x, y,
    test_size=1-train_data_ratio,
    random_state=123)
```

2) 코드 설명

파이프라인 정의 및 모델 훈련

```
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('model', VotingClassifier(
        estimators=[
            ('cat', CatBoostClassifier()),
            ('lgbm', LGBMClassifier()),
            ('xgb', XGBClassifier()),
            ('knn', KNeighborsClassifier(n_neighbors=5)),
            ('dt', DecisionTreeClassifier())
        ],
        voting='hard',
        n_jobs=-1
    ))
])

pipeline.fit(x_train, y_train)
```

2) 코드 설명

테스트 세트 예측 및 정확도 평가

```
accuracy = pipeline.score(x_test, y_test)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.8760935910478128
PS D:\ai study>
```

05 자체 평가 의견

1. 프로젝트 기획 의도와의 부합 정도:
   - 프로젝트의 기획 의도인 신용카드 신규 개설에 관심이 있는 고객들을 식별하는 모델을 개발하였다.

2. 달성도 및 완성도:
   - 주어진 데이터를 기반으로 모델을 학습하고 평가하는 코드를 구현하였다.
   - 결과적으로 모델은 정확도 87.60%를 달성하였다.

3. 자체평가 의견과 느낀 점:
   - 이 프로젝트를 통해 데이터 전처리, 모델 구성, 파이프라인 구축 등의 기본적인 머신러닝 프로세스를 경험할 수 있었다.
   - VotingClassifier를 활용한 앙상블 모델의 성능과 활용 가능성에 대해 이해할 수 있었다.

4. 개선할 점
 - 더 다양한 모델 사용, 하이퍼파라미터 튜닝 등을 시도해 더 높은 정확도의 성능을 찾을 수 있었으면 좋았을 것 같다.

감사합니다.