데이터 분석을 통한 창업 전망 예측

Project 2

2022.05

CODE STATES

목 차 Contents

1 Dataset

1.1 목 적

데이터를 통하여 창업 시 전망 예측

1.2 데이터셋 크기

	name	market	funding_total_usd	status	country_code	state_code	city	funding_rounds	founded_at	first_funding_at	last_funding_at
0	Verizon Communications	Mobile	30079503000	ipo	USA	NY	New York	5	2015-04-19	2010-01-26	2014-02-28
1	Uber	Real Time	7007450000	operating	USA	CA	San Francisco	12	NaN	2009-08-01	2015-08-19
2	Sberbank	Finance	5800000000	ipo	RUS	NaN	Moscow	1	2014-04-01	2014-07-07	2014-07-07
3	Clearwire	Mobile	5720000000	acquired	USA	WA	Kirkland	5	2012-06-01	2001-12-11	2013-02-27
4	Charter Communications	Telecommunications	5162513431	ipo	USA	CT	Stamford	2	NaN	2009-11-21	2014-09-15
...	...	...	...	...	...	...	...	...	...	...	...
61393	Zymr, Inc.	Cloud Infrastructure	0	operating	USA	CA	Santa Clara	3	1999-01-01	2012-12-12	2014-01-01
61394	Zynergy Projects & Services	Clean Technology	0	operating	IND	NaN	Chennai	1	2006-01-01	2015-08-19	2015-08-19
61395	Zype	Video	0	operating	USA	NY	New York	1	2013-01-01	2015-04-01	2015-04-01
61396	[24]7	Predictive Analytics	0	operating	USA	CA	Campbell	1	NaN	2003-07-01	2003-07-01
61397	??? ChiShenMa	Restaurants	0	operating	CHN	NaN	Shanghai	1	2014-01-01	2014-05-25	2014-05-25

61398 rows × 11 columns

dataset의 초기상태는 (61398, 11)

1 Dataset

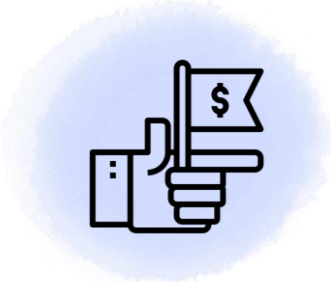
1.3 변수 설명

변수명	해석	타입
name	회사명	Categorical
market	업종(778가지)	Categorical
funding_total_usd	총 투자받은 금액	Numerical
status	상태 ('operating', 'acquired', 'closed', 'ipo')	Categorical
country_code	국가코드(135개국)	Categorical
state_code	지역코드	Categorical
city	도시	Categorical
funding_rounds	투자받은 횟수	Numerical
founded_at	설립일	Categorical
first_funding_at	처음 투자받은 날	Categorical
last_funding_at	마지막 투자받은 날	Categorical

Traget은 status로 클래스는 총 operating, acquired, closed, ipo 4가지로 구성

1 Dataset

1.4 분석목표 및 가설 설정



1) 창업을 한다면 어떤 업종으로 할 것인가?

- 창업 성공에 유리한 업종이 있을 것이다.

2) 성공하기 좋은 국가가 있을까?

- 창업 성공에 유리한 국가가 있을 것이다.

3) 선택한 업종은 어느정도 투자를 받아야 할 것인가?

- 어느정도 투자금액이 있어야 안정적으로 운영이 될 것이다.

창업을 한다면 어떤 조건에 따라 성공 할 수 있을지 목표로 설정하여 진행

2 Data wrangling & EDA

2.1 Feature

2.1.1 Feature 제거

1) Name
- 회사의 고유값으로 불필요하다고 판단하여 제거

2) State_code
- 국가만 해도 135개나 존재하는데 여기서 더 세분화 하면 '차원의 저주'의 위험이 있음으로 제거

3) City_code
- State_code와 동일 이유

4) First_funding_at
- 도메인 지식이 부족하여 분석하는데 오히려 방해가 될 것 같아 제거

5) Last_ funding_at
- First_funding_at와 동일 이유

2.1.2 Feature 추가

1) Average_funding_once
- 1회 평균 투자금액(총 투자 금액 / 투자 횟수)

2.2 Target

2.2.1 Target 설정

1) Status
- 창업 성공 유무를 판단하는게 목적 이므로 ipo, acquired 성공 operating은 운영중, closed는 실패로 총 3가지 클래스로 설정

2.3 Group

2.3.1 그룹화

1) Market
- 업종은 많은 카디널리티를 가지나 중요 feature라 생각하여 제거하진 못하고 상위 20개를 제외하곤 'etc'로 그룹화

2) County_code
- 지역, 도시는 삭제 했지만 상위 카테고리인 국가 코드는 제거 할 수가 없어 상위 20개를 제외하곤 'etc'로 그룹화

3) Founded_at
- 설립일은 일자별로 다루기엔 너무 많아 5년 단위로 그룹화하고 2005년 이전은 하나로 묶음

Target이 status로 분류분석으로 진행할 것이며, 불필요한 Feature들은 삭제하고,
많은 카디널리티를 가진 피처들은 그룹화를 통해 '차원의 저주'에 대한 위험을 방지

2 Data wrangling & EDA

2.4 중복 데이터, 결측치, 이상치

2.4.1 중복데이터 처리

1) Name
- 중복된 이름이 있었으나 이름만 같은 회사임을 확인 이후 Feature 제거

2.4.2 결측치 처리

1) Founded_at
- 설립일이 결측치가 많았으나 최빈값, 평균값 등으로 대체하기엔 분석에 방해가 될 것으로 판단하여 데이터 손실이 많아도 제거

2) 공통
- 결측치가 20개 미만인 것들은 제거하고 불필요한 피처들도 제거하여 확인하니 모든 결측치 처리 완료

2.4.3 이상치 유지

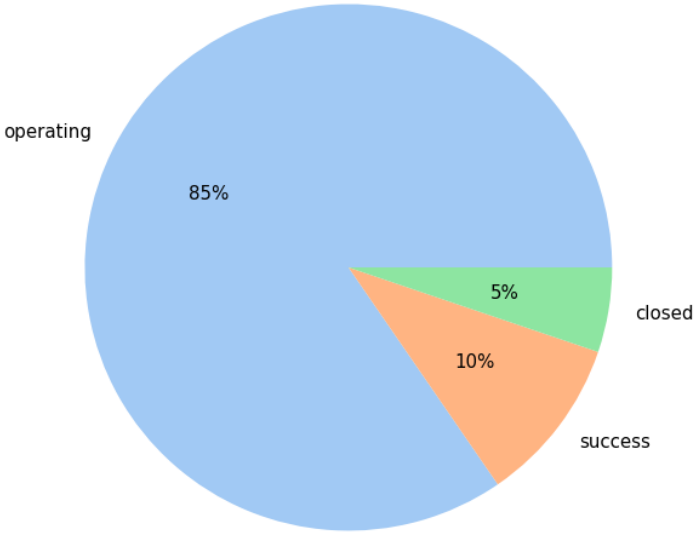
1) 이상치를 상한선, 하한선에 맞추거나 제거하려 했지만 생각을 해보니 트리모델은 외삽이 불가능한걸로 알고 있음. 따라서 이상치도 중요한 데이터라 생각하고 그대로 가져가서 사용.

2.5 데이터 불균형

2.5.1 데이터 불균형 처리

- 현재 데이터의 상태가 매우 불균형하므로 이대로 진행을 한다면 제대로 된 분류분석을 할 수 없음

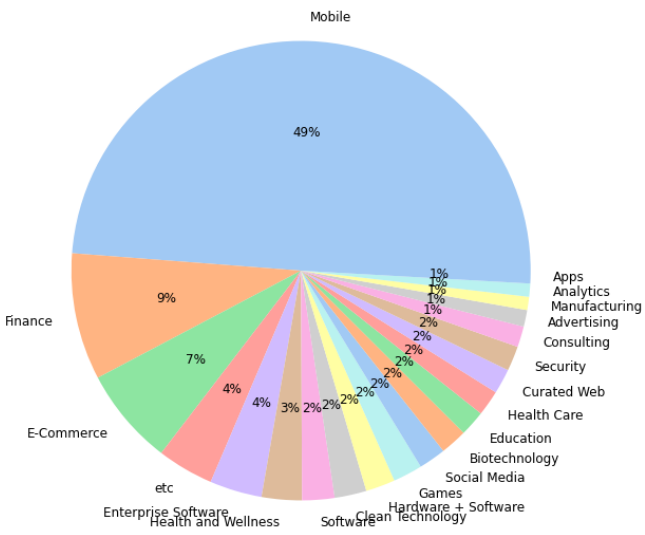
- 오버샘플링, 언더샘플링을 통하여 훈련셋트를 다시 만듬



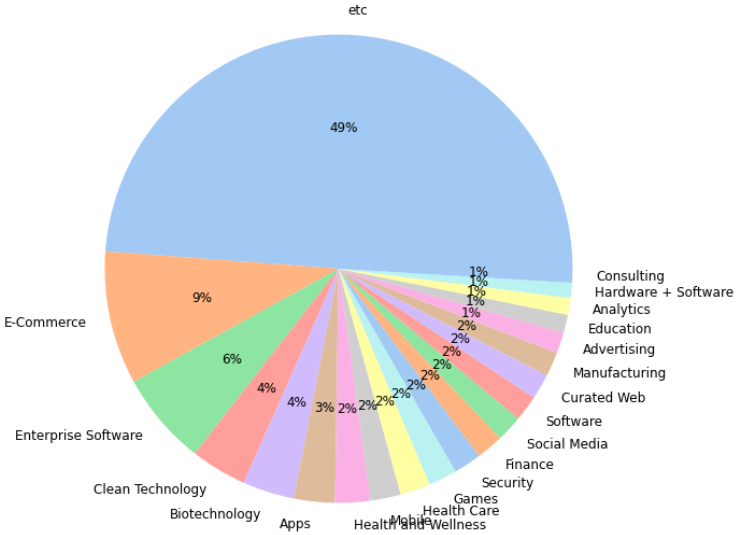
중복데이터, 결측치는 제거하였으며 이상치는 유지하고 타겟이 불균형한 상태로 이후 오버샘플링, 언더샘플링을 시도하여 좋은 쪽으로 선정
또한 훈련,검증,테스트 셋은 8:1:1의 비율로 분할

2 Data wrangling & EDA

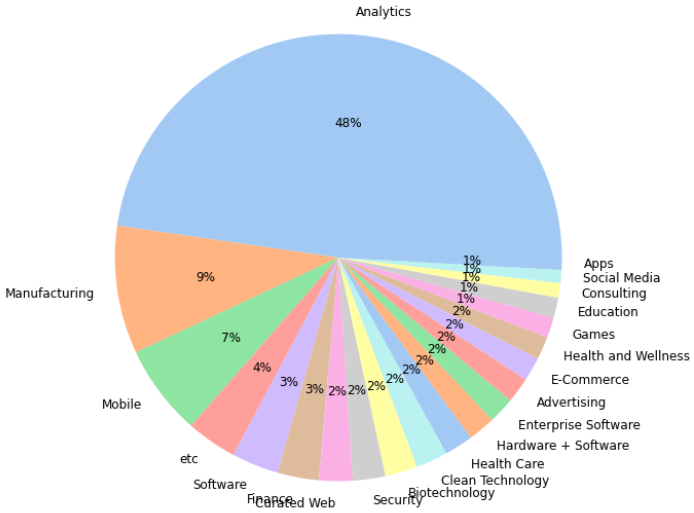
2.6 연도별 업종 선호도 그래프



2011~2015



2006~2011

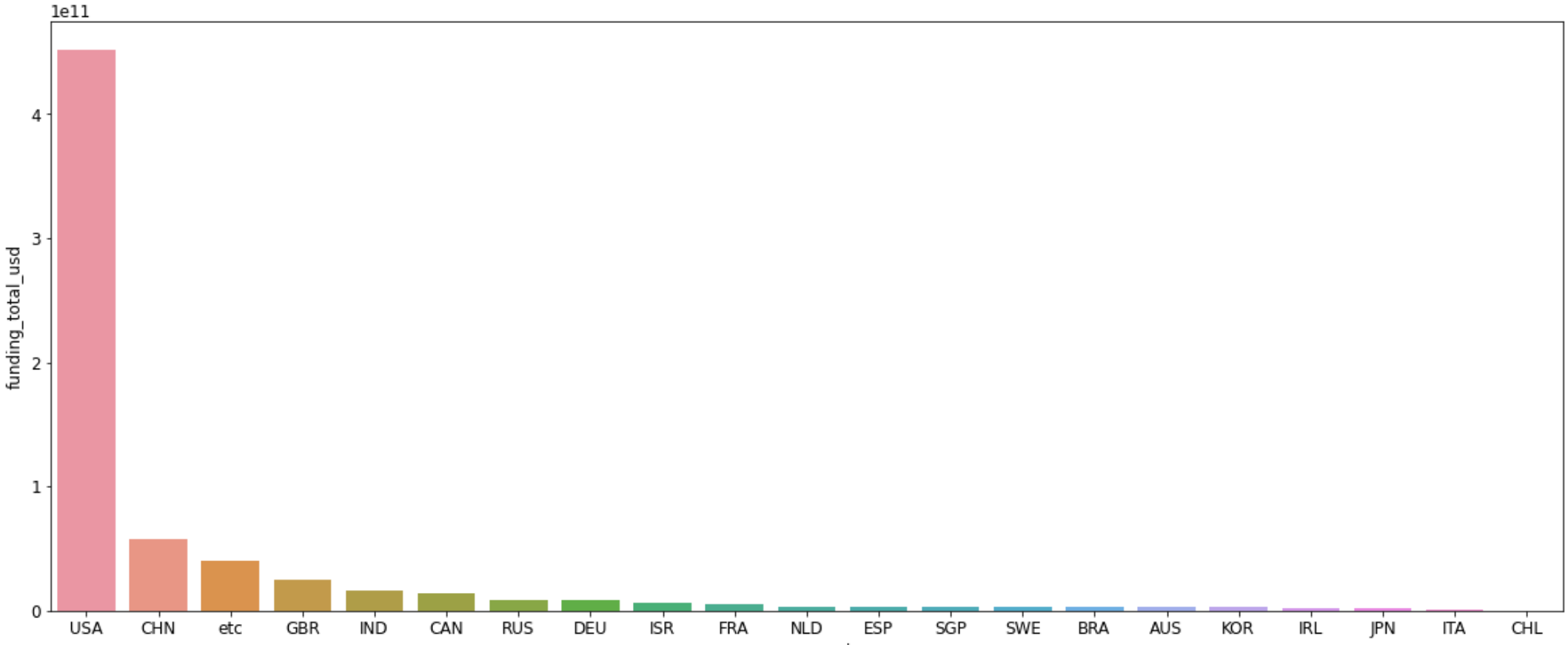


~ 2005

2011~2015년까지는 모바일시장, 2006~2011년은 기타 업종, 2005년 이전엔 분석시장이 활발
따라서 연도별 업종 선호도가 존재

2 Data wrangling & EDA

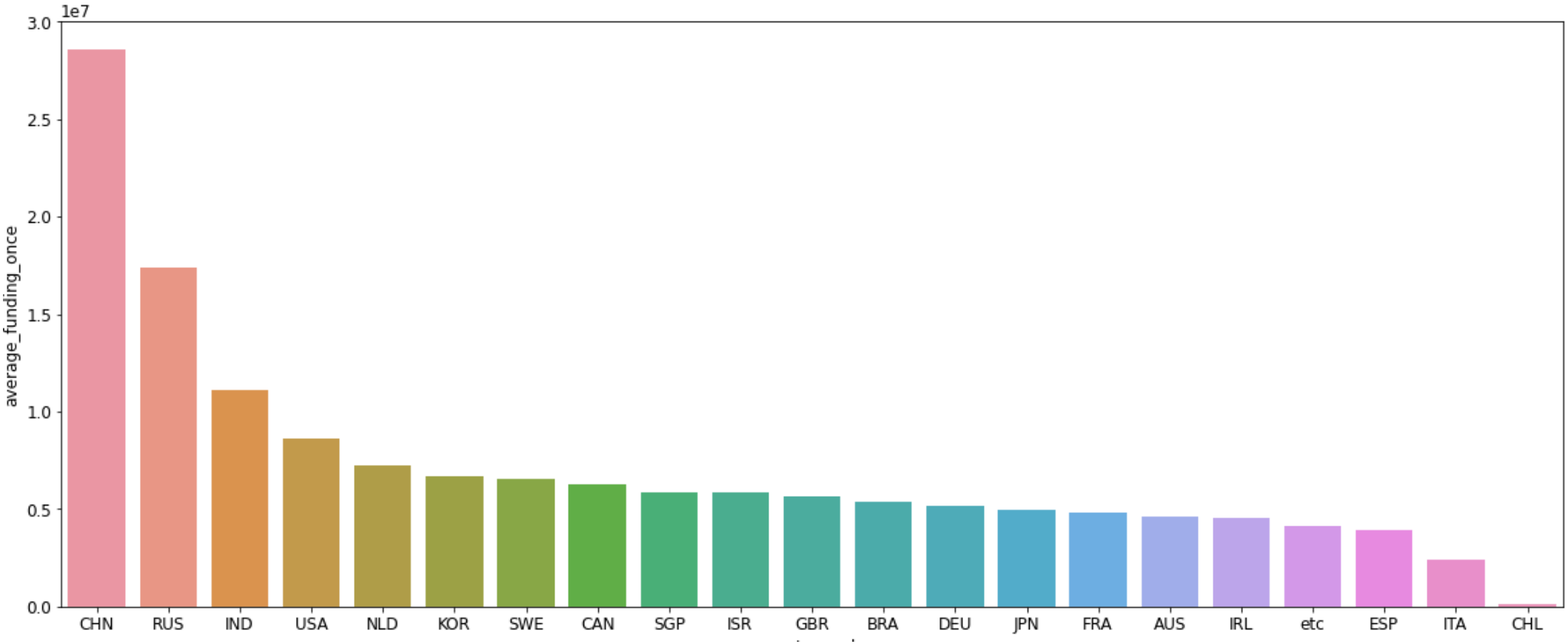
2.7 국가별 총 투자 금액 그래프



총 투자금액으로 보면 미국, 중국, 기타, 영국 순이지만 미국엔 창업 회사들도 많아 총 투자 금액이 높은 것으로 예상
사용하기엔 부적합하여 새로운 피처 생성

2 Data wrangling & EDA

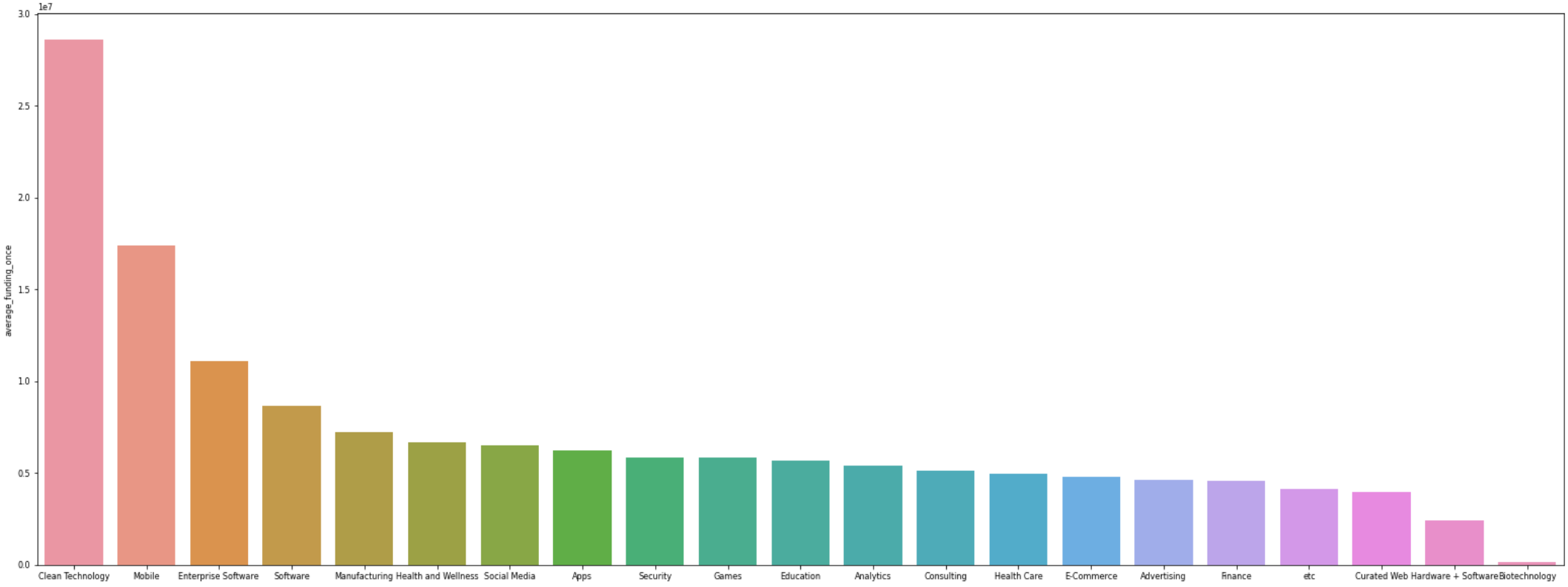
2.8 국가별 1회 평균 투자 금액



1회 평균 투자금액(총 투자 금액 / 투자 횟수)을 보면 중국이 1등이고 미국은 4위
따라서 모델링을 할 땐 총 투자 금액보단 1회 평균 투자금액으로 설정

2 Data wrangling & EDA

2.9 업종별 1회 평균 투자 금액

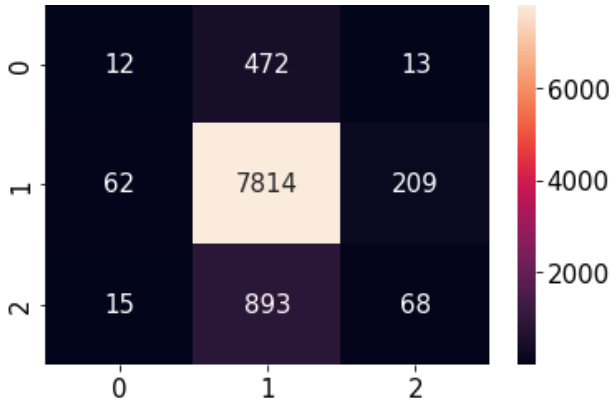


업종들의 1회 평균 투자금액으로 생산시설, 원자재값 등은 업종별로 다를 것으로 예상

3 Modeling

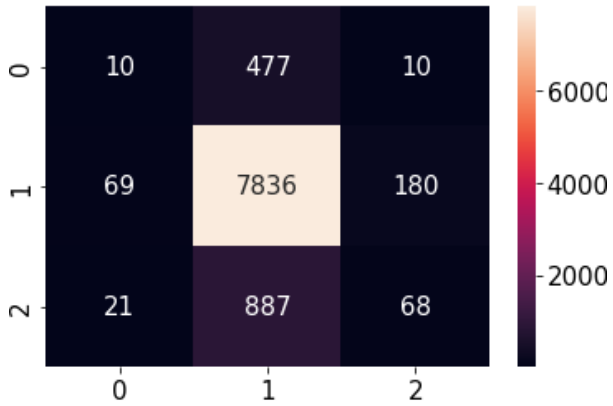
3.1 Modeling 성능 비교

3.1.1 Robustscaler & MinMaxScaler & StandardScaler



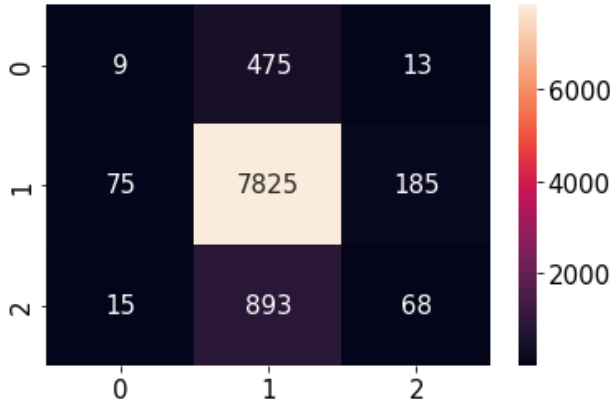
	precision	recall	f1-score	support
closed	0.13	0.02	0.04	497
operating	0.85	0.97	0.91	8085
success	0.23	0.07	0.11	976
accuracy			0.83	9558
macro avg	0.41	0.35	0.35	9558
weighted avg	0.75	0.83	0.78	9558

Robustscaler



	precision	recall	f1-score	support
closed	0.10	0.02	0.03	497
operating	0.85	0.97	0.91	8085
success	0.26	0.07	0.11	976
accuracy			0.83	9558
macro avg	0.41	0.35	0.35	9558
weighted avg	0.75	0.83	0.78	9558

MinMaxScaler



	precision	recall	f1-score	support
closed	0.09	0.02	0.03	497
operating	0.85	0.97	0.91	8085
success	0.26	0.07	0.11	976
accuracy			0.83	9558
macro avg	0.40	0.35	0.35	9558
weighted avg	0.75	0.83	0.78	9558

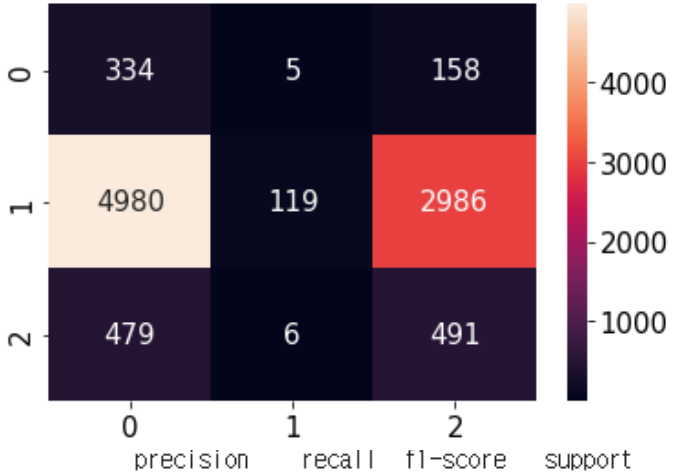
StandardScaler

3가지 스케일러 모두 정확도, f1_score가 동일하지만 이상치를 제거 안하고 사용함에 따라 영향을 적게 받는 Robustscaler를 사용

3 Modeling

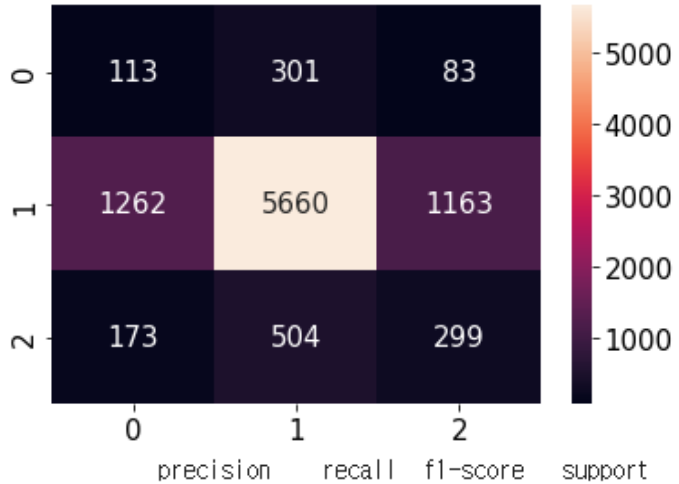
3.1 Modeling 성능 비교

3.1.2 Undersampling(Nearmiss) & Oversampling(SMOTE)



	precision	recall	f1-score	support
closed	0.06	0.67	0.11	497
operating	0.92	0.01	0.03	8085
success	0.14	0.50	0.21	976
accuracy			0.10	9558
macro avg	0.37	0.40	0.12	9558
weighted avg	0.79	0.10	0.05	9558

언더샘플링(nearmiss)



	precision	recall	f1-score	support
closed	0.07	0.23	0.11	497
operating	0.88	0.70	0.78	8085
success	0.19	0.31	0.24	976
accuracy			0.64	9558
macro avg	0.38	0.41	0.38	9558
weighted avg	0.76	0.64	0.69	9558

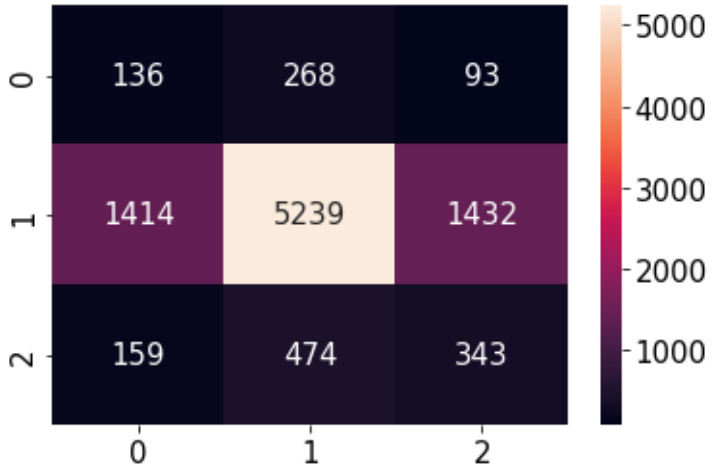
오버샘플링(smote)

여기서 타겟의 비율을 1:1:1로 설정, 따라서 기준모델은 0.33
이번 데이터는 오버샘플링이 언더샘플링보다 더 좋은 성능임을 확인

3 Modeling

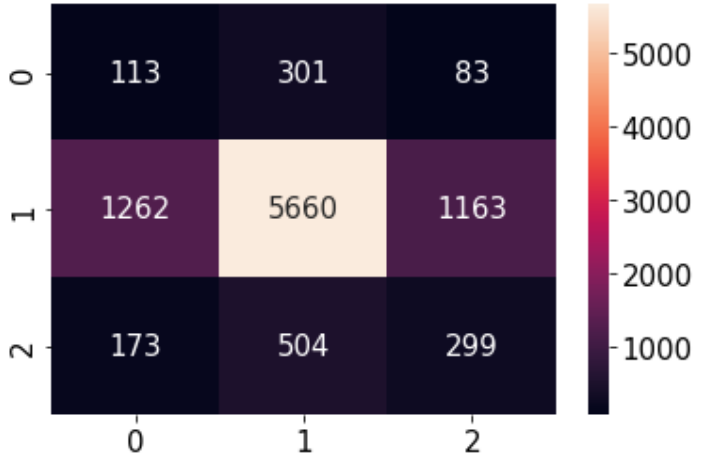
3.1 Modeling 성능 비교

3.1.3 RandomForestClassifier & KNN(K-Nearest Neighbours)



	precision	recall	f1-score	support
closed	0.08	0.27	0.12	497
operating	0.88	0.65	0.74	8085
success	0.18	0.35	0.24	976
accuracy			0.60	9558
macro avg	0.38	0.42	0.37	9558
weighted avg	0.76	0.60	0.66	9558

RandomForestClassifier



	precision	recall	f1-score	support
closed	0.07	0.23	0.11	497
operating	0.88	0.70	0.78	8085
success	0.19	0.31	0.24	976
accuracy			0.64	9558
macro avg	0.38	0.41	0.38	9558
weighted avg	0.76	0.64	0.69	9558

KNN(K-Nearest Neighbours)

KNN이 랜덤포레스트보다 성능이 더 좋음을 확인

3 Modeling

3.2 Modeling 최적 파라미터

3.2.1 RandomizedSearchCV 설정

- KNeighborsClassifier는 파라미터가 크게 3가지 사용
- n_neighbors = 최근접 이웃 수 설정 후 주변데이터가 더 많이 포함된 것으로 설정 따라서 짝수로 설정 시 동점이 나올 가능성이 있어 홀수 추천
- weights = 분류할 때 인접한 샘플의 거리에 따라 다른 가중치 부여
- metric = 거리계산 방식
- Scoreing = f1_score가 너무 떨어짐 따라서 f1으로 설정

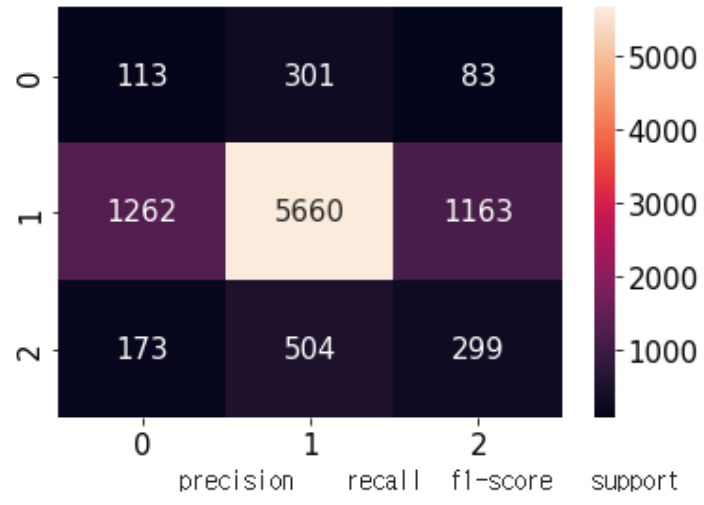
```
pipe3 = make_pipeline(
        RobustScaler(),
        KNeighborsClassifier())

dists = {
    'kneighborsclassifier__n_neighbors' : [3,5,7,9],
    'kneighborsclassifier__weights' : ['uniform','distance'],
    'kneighborsclassifier__metric' : ['euclidean','manhattan','chebyshev','seuclidean','minkowski']
}

clf= RandomizedSearchCV(
    pipe3,
    param_distributions=dists, # 파라미터 입력
    n_iter = 50,   # random search 탐색 횟수
    cv = 5,        # cv 검증을 위한 분할 검증 횟수
    scoring='f1',  # 오차 평가방법
    verbose=1,     # 진행상황
    random_state = 2
  )

clf.fit(X_train_over, y_train_over)
```

최적 하이퍼파라미터 {'weights': 'uniform', 'n_neighbors': 3, 'metric': 'euclidean'}으로 나왔으나 distance가 스코어가 더 좋게 나옴

3 Modeling

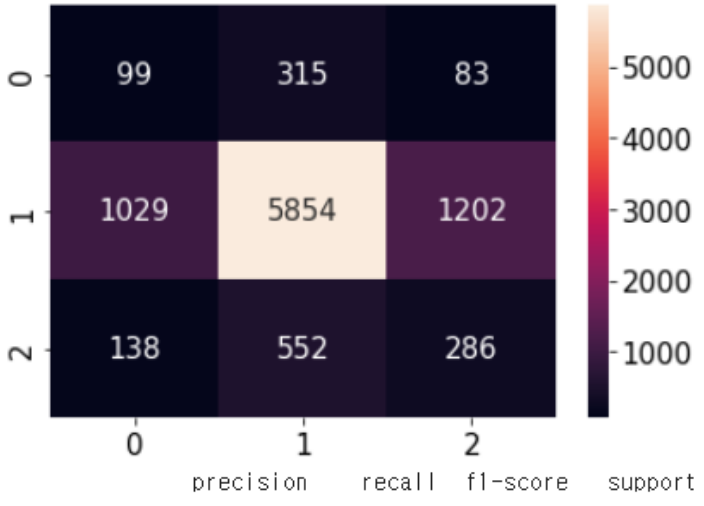
3.2 Modeling 최적 파라미터

3.2.2 RandomizedSearchCV 결과



	precision	recall	f1-score	support
closed	0.07	0.23	0.11	497
operating	0.88	0.70	0.78	8085
success	0.19	0.31	0.24	976
accuracy			0.64	9558
macro avg	0.38	0.41	0.38	9558
weighted avg	0.76	0.64	0.69	9558

파라미터 적용 전



	precision	recall	f1-score	support
closed	0.08	0.20	0.11	497
operating	0.87	0.72	0.79	8085
success	0.18	0.29	0.22	976
accuracy			0.65	9558
macro avg	0.38	0.41	0.38	9558
weighted avg	0.76	0.65	0.70	9558

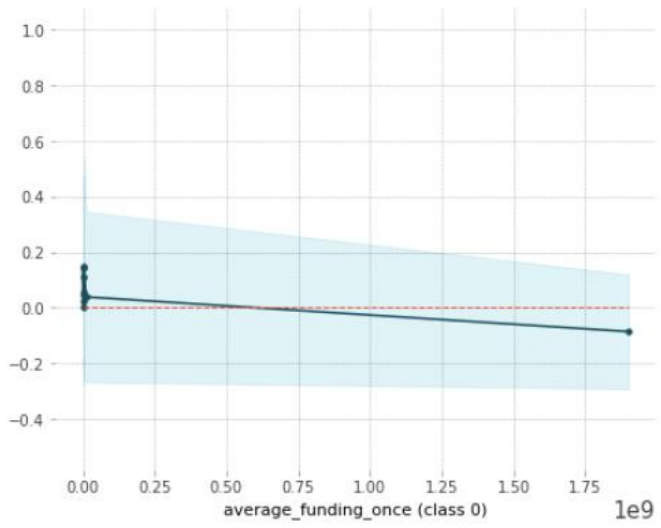
파라미터 적용 후

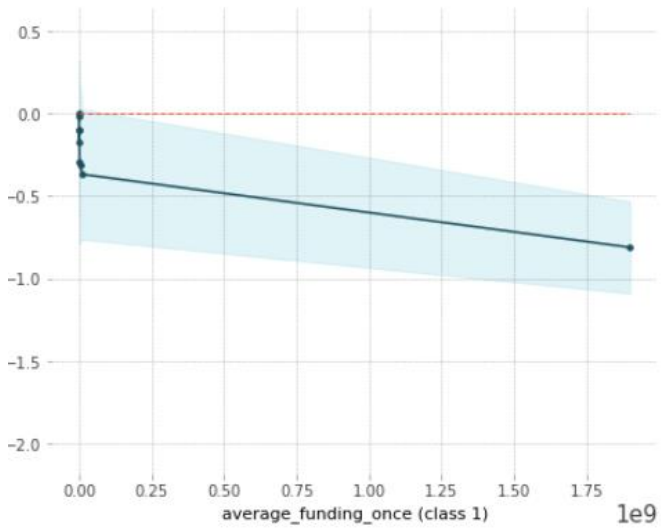
F1_score는 동일하지만 정확도는 0.01 상승
최종 모델의 성능은 정확도 0.65, f1 score 0.38

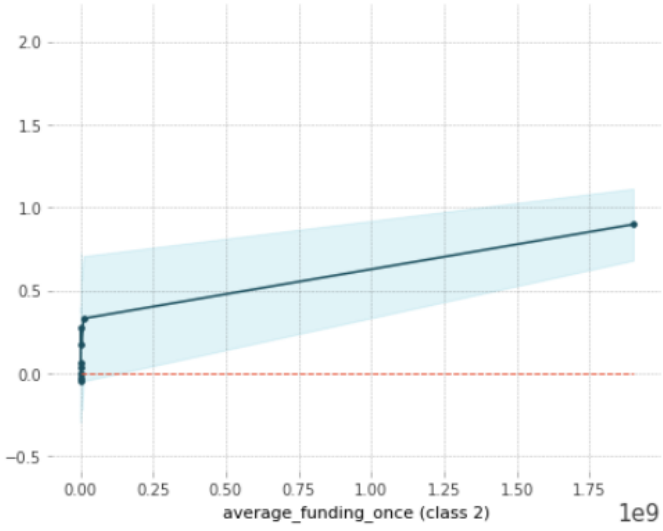
4 모델 해석

4.1 PDP PLOT

4.1.1 Average Funding Once(1회 평균 투자 받은 금액)







Class 0(closed)

Class 1(success)

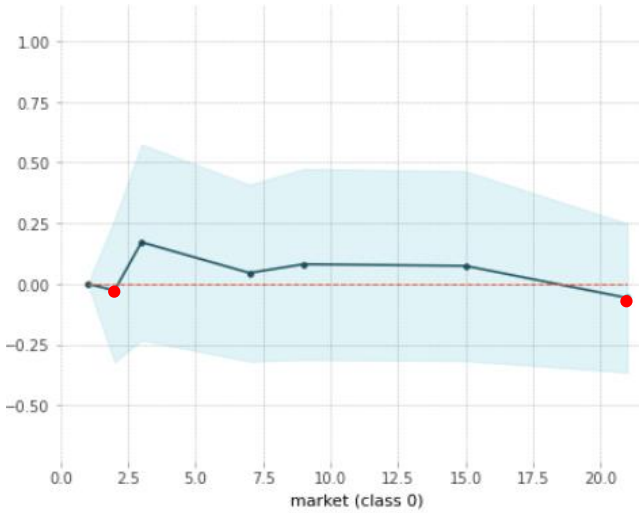
Class 2(operating)

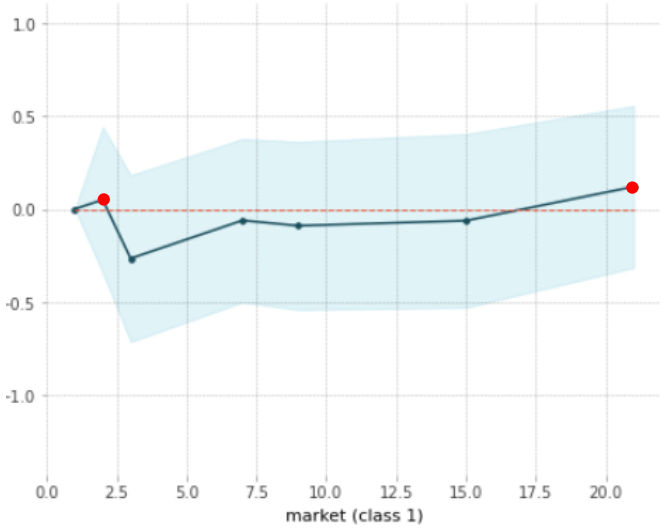
1회 평균 투자 받은 금액이 증가할수록 실패 또는 성공할 확률은 적어지고, 운영에 대한 확률은 증가

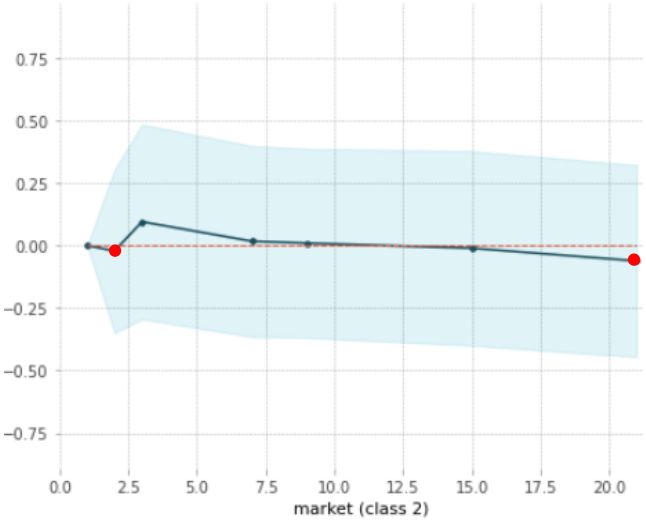
4 모델 해석

4.1 PDP PLOT

4.1.2 Market(업종)







Class 0(closed)

Class 1(success)

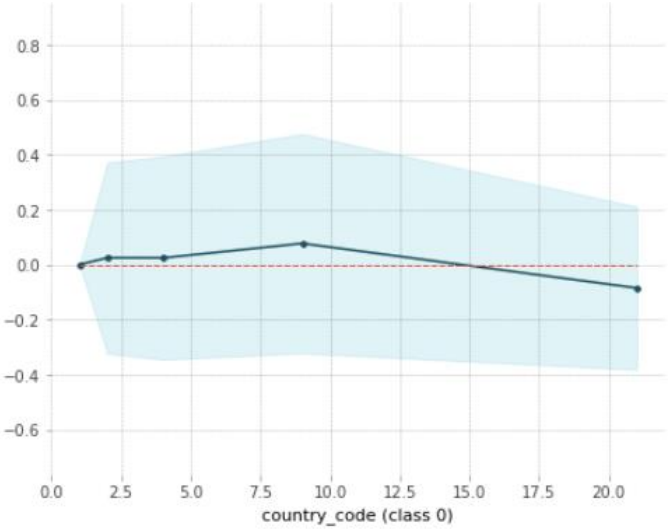
Class 2(operating)

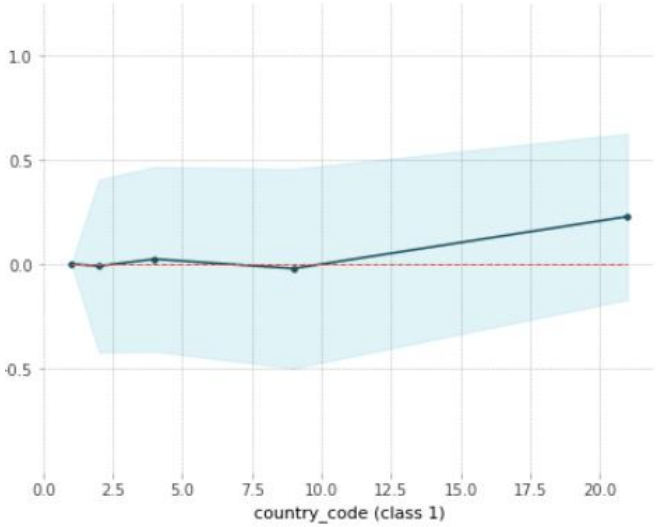
빨간 포인트는 성공확률이 높은 곳을 표시, 성공할 확률이 높은 업종이 존재

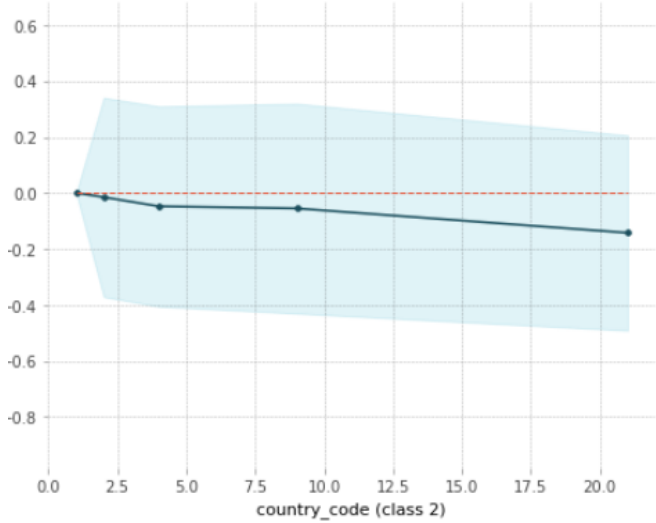
4 모델 해석

4.1 PDP PLOT

4.1.3 County code(국가)







Class 0 (closed)

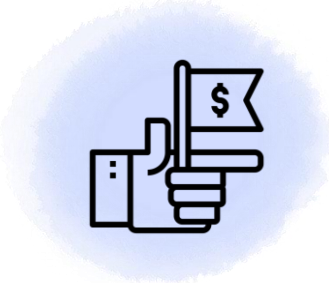
Class 1 (success)

Class 2 (operating)

Class0 ,Class1 그래프를 보면 성공 확률이 높아지는 국가는 존재

4 모델 해석

4.2 결 론



1) 지금 창업을 한다면 어떤 업종으로 할 것인가?

- 성공할 확률이 높은 업종이 있음

2) 성공하기 좋은 국가가 있을까?

- 투자를 많이 하는 국가가 있음

3) 선택한 업종은 어느정도 투자를 받아야 할 것인가?

-투자 금액은 성공여부를 판단하기엔 어려움

데이터를 통해(2015년까지) 스타트업 회사가 늘어나는 추세임을 알 수 있고 성공(인수,주식상장)을 하려면 업종과 국가선택이 중요