LRA Loan Risk Assessment

- 고객 행동에 따른 채무불이행 예측

S2gmo2d

INDEX

개요

- 팀 소개
- 분석배경
- 분석환경

데이터 수집

- 데이터 파악
- 기초통계
- 데이터전처리

데이터 분석

- 분석모델
- 최적모델선택

예측 및 결론

- 예측분석 결과
- 기대효과
- 개선사항

부록

- 웹 시연

1. 개요

1-1. 팀 소개

1-2. 분석배경

1-3. 분석환경

S2gmo2d 멤버를 소개합니다!

김성*

팀장
데이터 분석
예측 분석
웹 사이트 디자인
웹 구현

권은*

기초통계
데이터 분석
데이터시각화
웹 구현

박경*

데이터 전처리
데이터 분석
예측 분석
웹 구현

이승*

기초통계
데이터 분석
데이터 시각화
웹 구현

조경*

데이터 전처리
데이터 분석
데이터 시각화
웹 구현

조광*

부팀장
기초통계
데이터 분석
웹 사이트 디자인
웹 구현

Technology

Facebook, Xiaomi Eye India's $1 Trillion Digital Loan Market

By Suvashree Ghosh

페이스북-샤오미, 인도 '대출업' 앞다퉈 진출…온라인 협력사 위주

인도 온라인 대출 시장, 빅테크 기업 진출로 활기

경제

5대 은행 신용대출 '연봉 이내'로 규제...'대출절벽' 현실화

2021년 08월 27일 22시 28분 댓글

경제 > 경제 일반

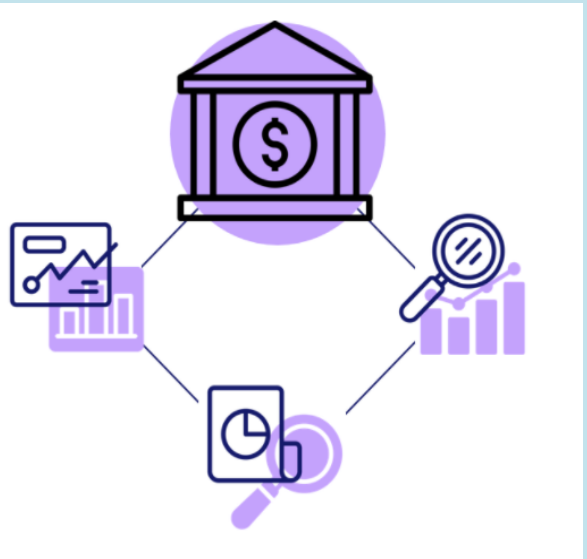
제2금융권도 신용대출 한도 연봉 이내로 제한할 듯

DB손보 "신용대출 중단"…보험업계도 '대출 절벽' 우려

IBK기업은행, 빅데이터 심사로 성장유망기업 1조 대출

'빅데이터' 활용 대출시장 열린다

"AI로 신용평가...중금리 대출 개척"



1-2. 개요 | 분석배경

- ✓ 대출 데이터는 금융 기관에서 비공개
- ✓ 인도 데이터는 구하기가 용이함
- ✓ 데이터의 수가 25만개로, 빅데이터 분석에 사용할 수 있음
- ✓ 인도 온라인 대출 시장에 진출하는 해외 기업들이 늘어나고 있음

- ✓ 대출 규제에 따라 대출에 대한 사람들의 관심 증가
- ✓ 대출 여부를 예측하는 서비스에 대한 수요가 증가할 것으로 예상됨
- ✓ 금융 기관에서도 빅데이터를 활용하여 대출을 심사

Kaggle의 'Loan Prediction Based on Customer Behavior '데이터를 사용, 채무불이행 위험을 예측

데이터 분석

웹 프로그래밍

2. 데이터 수집

2-1. 데이터 파악

2-2. 기초통계

2-3. 데이터 전처리

' Loan Prediction Based on Customer Behavior'

kaggle

Column	Description	Scale	Type
income	소비자의 소득수준	수치형	int
age	소비자의 연령	수치형	int
experience	전체 산업군의 경력 분포	수치형	int
profession	직업(51개 직업->15개 직업분류)	범주형	string
married	결혼여부(결혼/미혼)	범주형	string
house_ownership	주택소유여부(소유/렌트/둘다 아님)	범주형	string
car_ownership	차 소유 여부(있음/없음)	범주형	string
Current *job* years	근속기간	수치형	int
Current *house* years	현재 거주지에서 거주 기간	수치형	int
City	주거 지역(도시)	범주형	string
state	주거 지역(주)	범주형	string
risk_flag	과거 채무불이행 여부(있음/없음)	범주형	string

상관관계 분석

_label(risk_flag)와 feature 간 상관관계 분석

Experience	0.034523
Bhubaneswar	0.030400
rented	0.026647
Kochi	0.024464
Car_Ownership	0.024036
owned	0.023499
Madhya_Pradesh	0.023271
Gwalior	0.022567
Age	0.021809
Married/Single	0.021092
Buxar[37]	0.020689
Barasat	0.020651
Kerala	0.020617
Satna	0.020009
Sikar	0.019850

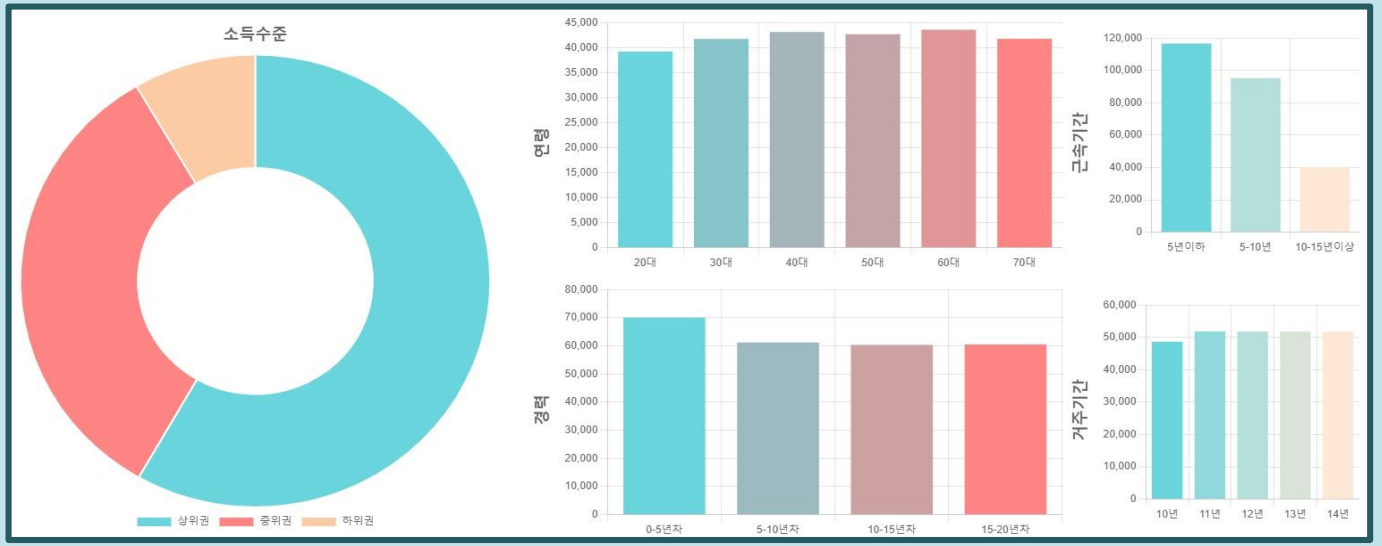
✓ 상위 15개만 추출

✓ 최대값이 0.03인 것으로 보아

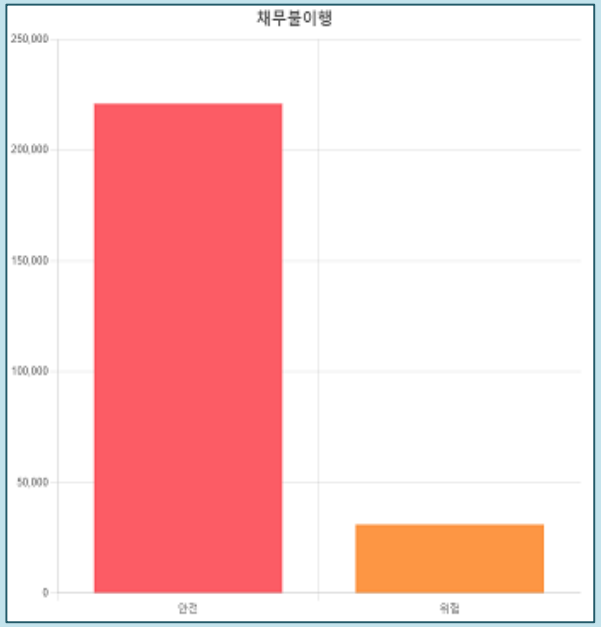
모든 변수와 label 간

상관관계가 거의 없다고 판단

- ✓ Feature 11개를 시각화
- ✓ Feature를 각 범주 레이블 당 인원수로 표현 - 분포를 파악하기 위함
- ✓ 그리드 형식으로 배치 - 그래프를 한눈에 파악하기 용이
- ✓ RandomForest 분석모델의 주요변수 5개를 추려서 상위에 위치
  (소득수준-연령-경력-근속기간-거주기간)



<참고>
Label
(Risk_Flag)

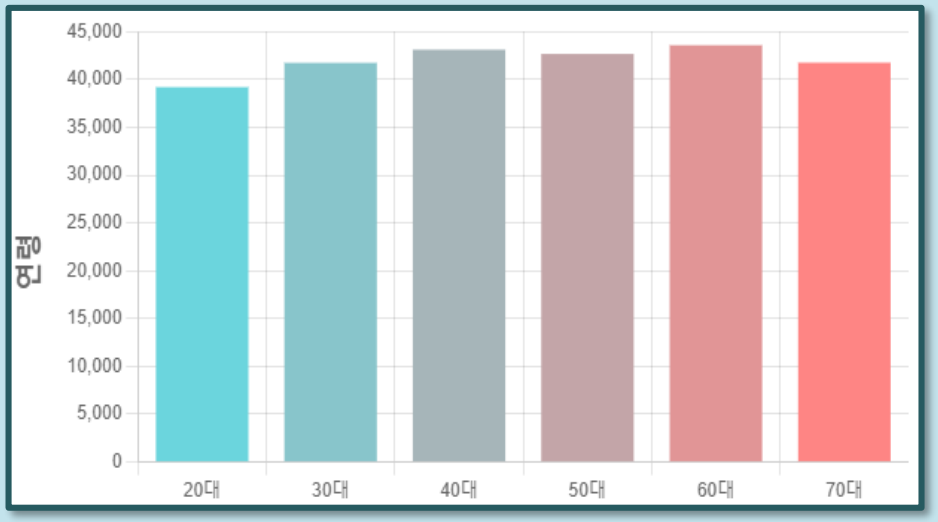


2-2. 데이터 수집 | 기초통계

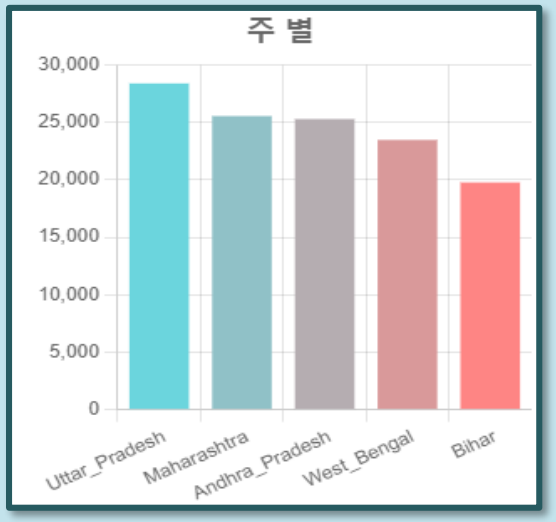
✓ 연속형변수 -> 구간으로 나누어 범주화

✓ 범주형변수 -> 상위 몇 개만 표현

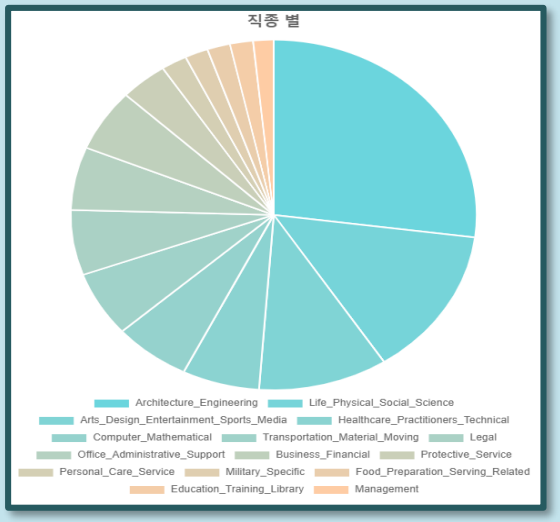
✓ 직업변수 -> 미국 산업군 기준 15개의 산업군으로 재분류



<연속형>



<범주형>



<직종>

데이터 구성

1. id(아이디)
2. income(소득)
3. age(연령)
4. experience(전체 산업군 경력 분포)
5. profession(직업)
6. married(결혼유무)
7. house_ownership(집 소유)
8. car_ownership(차 소유)
9. risk_flag (대출 위험도)
10. currentjobyears(근속기간)
11. currenthouseyears(거주기간)
12. city(도시)
13. state(주)

	Id	Income	Age	Experience	Married/Single	House_Ownership	Car_Ownership	\
0	1	1303834	23	3	single	rented	no	
1	2	7574516	40	10	single	rented	no	
2	3	3991815	66	4	married	rented	no	

	Profession	CITY	STATE	CURRENT_JOB_YRS	\
0	Mechanical_engineer	Rewa	Madhya_Pradesh	3	
1	Software_Developer	Parbhani	Maharashtra	9	
2	Technical_writer	Alappuzha	Kerala	4	

	CURRENT_HOUSE_YRS	Risk_Flag
0	13	0
1	13	0
2	10	0

데이터 타입

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 252000 entries, 0 to 251999
Data columns (total 13 columns)

4. married(결혼유무)
5. house_ownership(집 소유)
6. car_ownership(차 소유)
7. profession(직업)
8. city(도시)
9. state(주)

→ Object type 변수 6개 확인

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 252000 entries, 0 to 251999
Data columns (total 13 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   Id                 252000 non-null  int64
 1   Income             252000 non-null  int64
 2   Age                252000 non-null  int64
 3   Experience         252000 non-null  int64
 4   Married/Single     252000 non-null  object
 5   House_Ownership    252000 non-null  object
 6   Car_Ownership      252000 non-null  object
 7   Profession         252000 non-null  object
 8   CITY               252000 non-null  object
 9   STATE              252000 non-null  object
 10  CURRENT_JOB_YRS    252000 non-null  int64
 11  CURRENT_HOUSE_YRS  252000 non-null  int64
 12  Risk_Flag          252000 non-null  int64
dtypes: int64(7), object(6)
memory usage: 25.0+ MB
```

변수 삭제, 값 대체

1. Unique함수사용 NaN값 없음 확인
2. id칼럼 제거
3. married(결혼유무)
   ['single' 'married']-> 0,1값 대체
4. car_ownership(차 소유)
   ['no' 'yes']-> 0,1값 대체

	Income	Age	Experience	Married/Single	House_Ownership	Car_Ownership	\
0	1303834	23	3	0	rented	0	
1	7574516	40	10	0	rented	0	
2	3991815	66	4	1	rented	0	

	Profession	CITY	STATE	CURRENT_JOB_YRS	\
0	Mechanical_engineer	Rewa	Madhya_Pradesh	3	
1	Software_Developer	Parbhani	Maharashtra	9	
2	Technical_writer	Alappuzha	Kerala	4	

	CURRENT_HOUSE_YRS	Risk_Flag
0	13	0
1	13	0
2	10	0

라벨인코딩

변수 unique()확인

1. House_Ownership - 3종류
   ['rented' 'norent_noown' 'owned']
2. Profession - 51종류
3. CITY - 317종류
4. STATE - 29종류

dataLabel.shape : (252000, 12)

```
data_la = data[['House_Ownership','Profession', 'CITY', 'STATE']]
#print(data_la.head(3))
data_la.loc[:,'House_Ownership'] = LabelEncoder().fit_transform(data_la['House_Ownership'])
data_la.loc[:,'Profession'] = LabelEncoder().fit_transform(data_la['Profession'])
data_la.loc[:,'CITY'] = LabelEncoder().fit_transform(data_la['CITY'])
data_la.loc[:,'STATE'] = LabelEncoder().fit_transform(data_la['STATE'])
```

	Income	Age	Experience	Married/Single	Car_Ownership	CURRENT_JOB_YRS	\
0	1303834	23	3	0	0	3	
1	7574516	40	10	0	0	9	
2	3991815	66	4	1	0	4	

	CURRENT_HOUSE_YRS	Risk_Flag	House_Ownership	Profession	CITY	STATE
0	13	0	2	33	251	13
1	13	0	2	43	227	14
2	10	0	2	47	8	12

원핫인코딩

1. House_Ownership - 3종류
2. Profession - 51종류
3. CITY - 317종류
4. STATE - 29종류

dataOneHot.shape : (252000, 408):
51(직업)+317(도시)+29(주)+3(집여부)+13(원본변수)-5(ID, House, Prof, CITY, STATE)

```
#----------------------------집여부 원핫인코딩-------------------
housenames=sorted(data['House_Ownership'].unique())
data_x1 = pd.DataFrame(OneHotEncoder().fit_transform(
    data_x['House_Ownership'].values[:, np.newaxis]).toarray(),\
    columns=housenames, index = data_x.index)
data_x = pd.concat([data_x, data_x1], axis = 1)
```

```
 Income  Age  Experience  Married/Single  Car_Ownership  CURRENT_JOB_YRS  \
1303834   23           3               0              0                3

CURRENT_HOUSE_YRS  Risk_Flag  norent_noown  owned  rented  \
               13          0           0.0    0.0     1.0

Air_traffic_controller  Analyst  Architect  Army_officer  Artist  Aviator  \
                   0.0      0.0        0.0           0.0     0.0      0.0

Biomedical_Engineer  Chartered_Accountant  Chef  Chemical_engineer  \
                0.0                   0.0   0.0                0.0

Civil_engineer  Civil_servant  Comedian  Computer_hardware_engineer  \
           0.0            0.0       0.0                         0.0
```

산업군 별 라벨·원핫 인코딩

Profession
51종류 -> 15종류의 산업군

dataLabel_pro.shape : (252000, 12)
→ 산업군별 Label인코딩된 데이터

dataOneHot_pro.shape:(252000, 372)
→ 산업군별 OneHot인코딩된 데이터

```
data_tt['Profession'] = data_tt['Profession'].replace([23,15,7], 'Business and Financial Operations Occupations')
data_tt['Profession'] = data_tt['Profession'].replace([47,42,11], 'Office and Administrative Support Occupations')
data_tt['Profession'] = data_tt['Profession'].replace([36,17,29,33,10,2,9,18,46,19,48,21,49,6], 'Architecture and Engineering Occupations')
data_tt['Profession'] = data_tt['Profession'].replace([34,26,1,40,44,41,20], 'Life, Physical, and Social Science Occupations')
data_tt['Profession'] = data_tt['Profession'].replace([43,13,14], 'Computer and Mathematical Occupations')
data_tt['Profession'] = data_tt['Profession'].replace(31, 'Education, Training, and Library Occupations')
data_tt['Profession'] = data_tt['Profession'].replace([32,30,39], 'Legal Occupations')
data_tt['Profession'] = data_tt['Profession'].replace(3, 'Military Specific Occupations')
data_tt['Profession'] = data_tt['Profession'].replace([16,45,37], 'Healthcare Practitioners and Technical Occupations')
data_tt['Profession'] = data_tt['Profession'].replace([12,22,27,50,4], 'Arts, Design, Entertainment, Sports, and Media Occupations')
data_tt['Profession'] = data_tt['Profession'].replace(28, 'Personal Care and Service Occupations')
data_tt['Profession'] = data_tt['Profession'].replace(8, 'Food Preparation and Serving Related Occupations')
data_tt['Profession'] = data_tt['Profession'].replace([0,5,25], 'Transportation and Material Moving Occupations')
data_tt['Profession'] = data_tt['Profession'].replace(35, 'Management Occupations')
data_tt['Profession'] = data_tt['Profession'].replace([24,38], 'Protective Service Occupations')
```

최종 전처리 데이터

dataLabel : (252000, 12)
→ Label인코딩된 데이터

dataOneHot : (252000, 408)
→ OneHot인코딩된 데이터

dataLabel_pro : (252000, 12)
→ 산업군별 Label인코딩된 데이터

dataOneHot_pro : (252000, 372)
→ 산업군별 OneHot인코딩된 데이터

```
 Income  Age  Experience  Married/Single  Car_Ownership  CURRENT_JOB_YRS  \
1303834   23           3               0              0                3

CURRENT_HOUSE_YRS  Risk_Flag  House_Ownership  Profession  CITY  STATE
               13          0                2          33   251     13
```

```
 Income  Age  Experience  Married/Single  Car_Ownership  CURRENT_JOB_YRS  \
1303834   23           3               0              0                3

CURRENT_HOUSE_YRS  Risk_Flag  norent_noown  owned  rented  \
               13          0           0.0    0.0     1.0

Air_traffic_controller  Analyst  Architect  Army_officer  Artist  Aviator  \
                   0.0      0.0        0.0           0.0     0.0      0.0
```

```
 Income  Age  Experience  Married/Single  Car_Ownership  CURRENT_JOB_YRS  \
1303834   23           3               0              0                3

CURRENT_HOUSE_YRS  Risk_Flag  House_Ownership  Profession  CITY  STATE
               13          0                2           0   251     13
```

```
 Income  Age  Experience  Married/Single  Car_Ownership  CURRENT_JOB_YRS  \
1303834   23           3               0              0                3

CURRENT_HOUSE_YRS  Risk_Flag  norent_noown  owned  rented  \
               13          0           0.0    0.0     1.0

Architecture and Engineering Occupations  \
                                     1.0

Arts, Design, Entertainment, Sports, and Media Occupations  \
                                                       0.0
```

3. 데이터 분석

3-1. 분석모델

3-2. 최적모델 선택

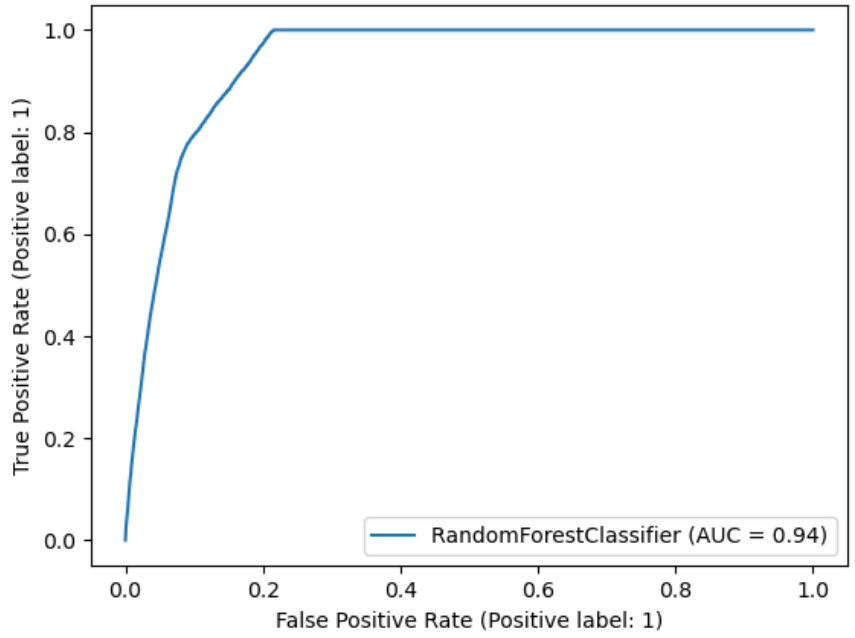
3-1. 데이터 분석 | 분석모델

모델명	정확도(Train)	정확도(Test)	과적합여부
LogisticRegression	87.65 %	87.81 %	X
KNN	90.06 %	89.41 %	X
SVM	87.65 %	87.80 %	X
MLP	89.33 %	88.96 %	X
NaiveBayes	87.65 %	87.81 %	X
DecisionTree	87.80 %	87.71 %	X
Randomforest	93.71 %	90.06 %	X
XGBClassifier	91.38 %	89.83 %	X
LGBMClassifier	90.41 %	89.59 %	X
Tensor	87.73 %	87.60 %	X

3-2. 데이터 분석 | 최적모델 선택

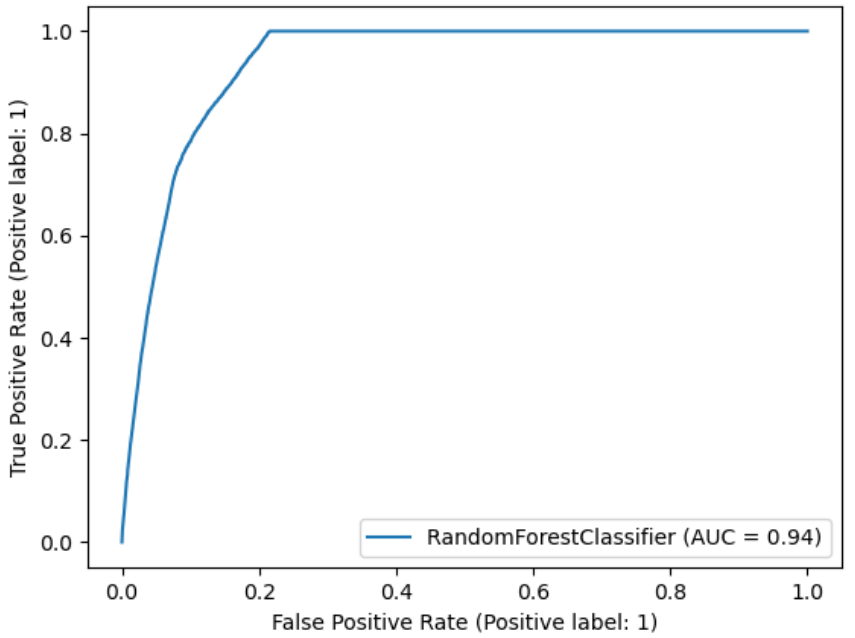
모델명	모델 정확도	GridSearchCV의 최적 파라미터	GridSearchCV의 최고 정확도	Cross Validation 이후 평균 정확도
LogisticRegression	87.81 %	'C': 0.01	87.81 %	87.70 %
KNN	89.41%	'n_neighbors': 15	89.41%	89.41%
SVM	87.80 %	'C': 0.01	87.80%	87.80 %
MLP	88.82 %	'hidden_layer_sizes': 4, 'learning_rate_init': 0.1	88.96 %	88.15%
NaiveBayes	87.81 %	x	x	87.70 %
DecisionTree	87.71%	x	x	87.71%
Randomforest	90.06 %	'n_estimators': 500	90.06%	90.06 %
XGBClassifier	88.25 %	'max_depth': 10, 'n_estimators': 300	89.83 %	89.69 %
LGBMClassifier	87.94 %	'max_depth': 30, 'n_estimators': 1000	89.59 %	89.83 %
Tensor	87.60 %	x	x	87.60 %

RandomForest



[onehot ver, 산업군 미포함, 직업/도시/주 포함]

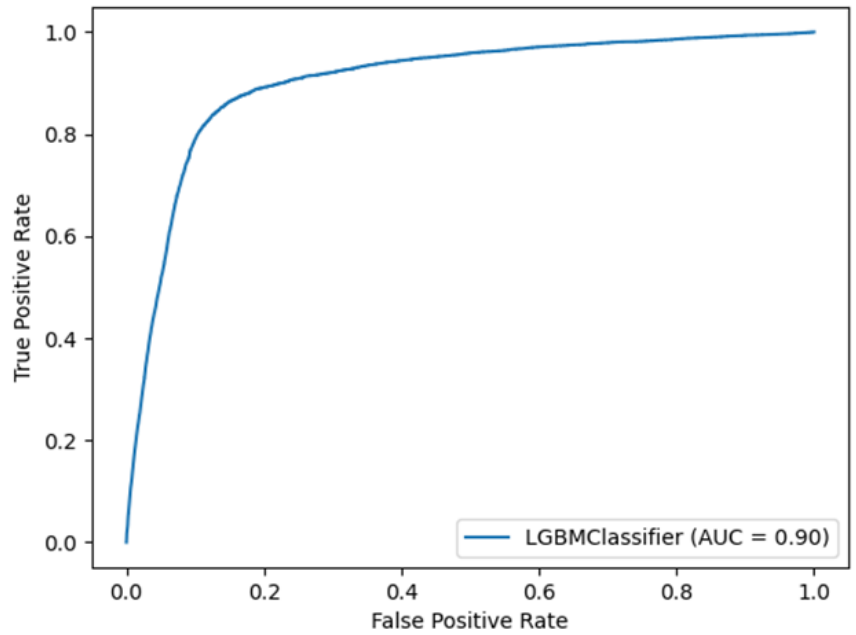
- 정확도 : 90.06 %
- 총 갯수:75600, 오류수:7870
- confusion_matrix:
  [63123 3215]
  [4297 4965]



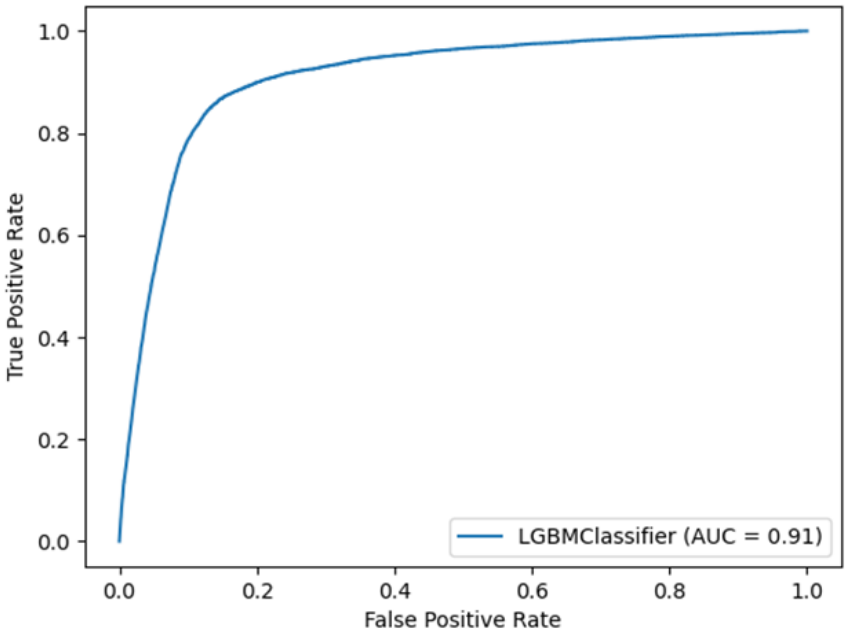
[simple ver, 산업군 포함, 직업/도시/주 제외]

- 정확도 : 90.01 %
- 총 갯수:75600, 오류수:7797
- confusion_matrix:
  [63067 3271]
  [4281 4981]

LGBM Classifier

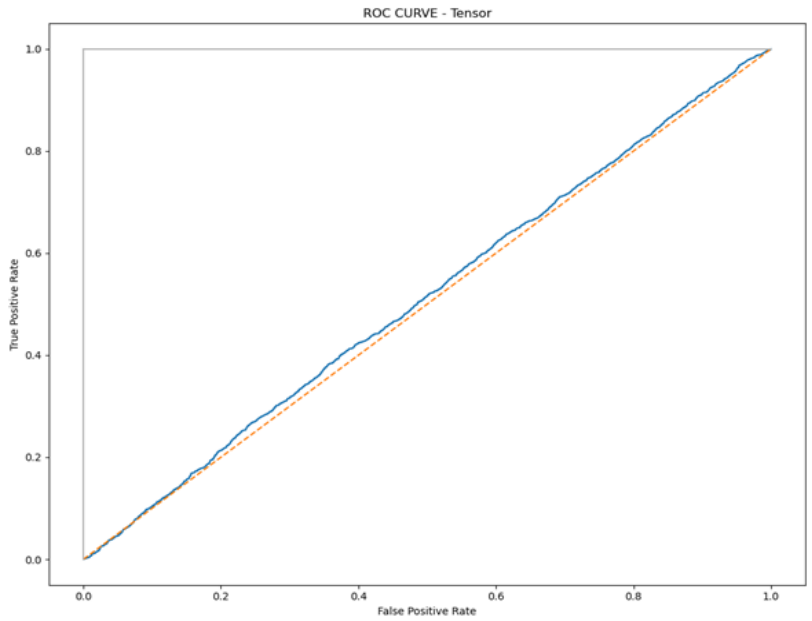


[onehot ver, 산업군 미포함, 직업/도시/주 포함]
- 정확도 : 89.83 %
- 총 갯수:75600, 오류수:7870
- confusion_matrix:
  [64412 1971]
  [5899 3318]



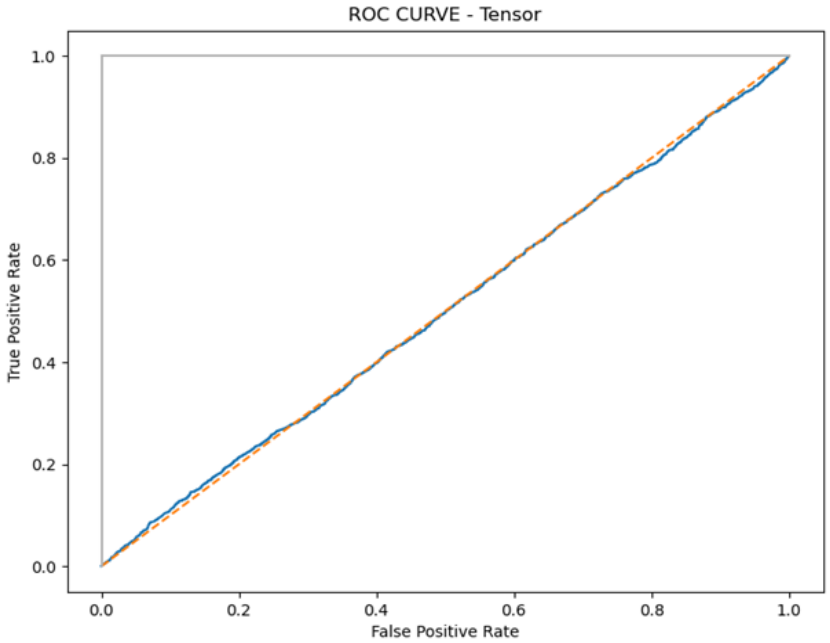
[simple ver, 산업군 포함, 직업/도시/주 제외]
- 정확도 : 89.81 %
- 총 갯수:75600, 오류수:7797
- confusion_matrix:
  [64294 2089]
  [5708 3509]

Tensor



[onehot ver, 산업군 미포함, 직업/도시/주 포함]

- 정확도 : 87.60 %
- 총 갯수:75600, 오류수:9368
- confusion_matrix:
  [66232 0]
  [9368 0]



[simple ver, 산업군 포함, 직업/도시/주 제외]

- 정확도 : 87.60 %
- 총 갯수:75600, 오류수:9368
- confusion_matrix:
  [66232 0]
  [9368 0]

4. 예측 및 결론

4-1. 예측분석 결과

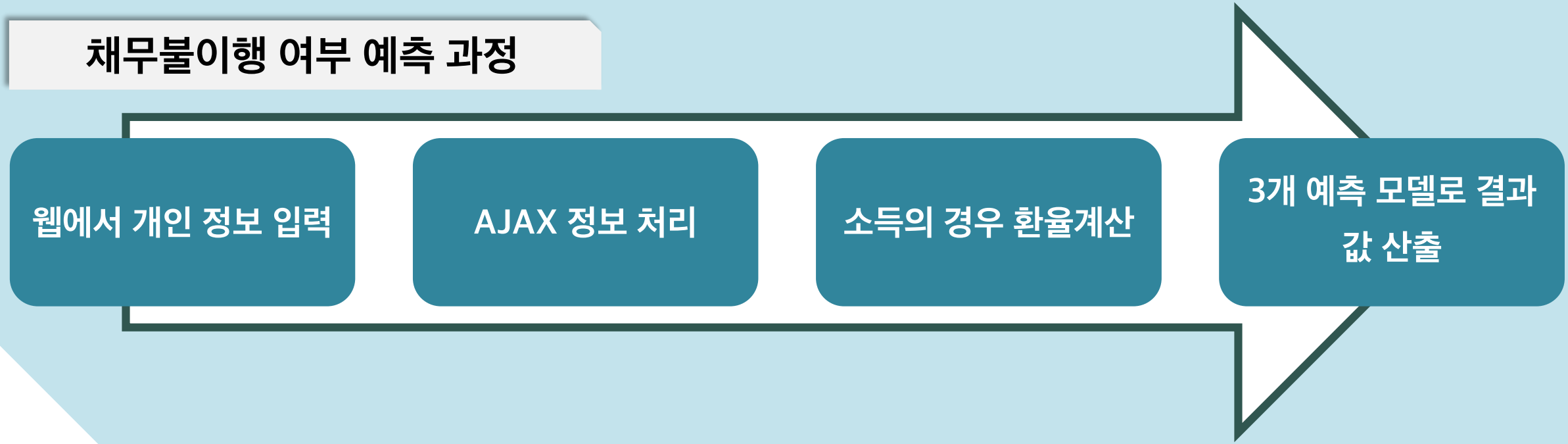
4-2. 기대효과 및 개선사항

✓ RandomForest, LGBM(분류정확도 상위 2개) + tensor 예측 모델

✓ 예측 모델 변수 추출 - 주, 도시, 직업 변수를 제외, 산업군 추가 새로운 예측 모델 생성

✓ 본인의 정보 입력 시, 각 모델 별로 채무불이행 예측

채무불이행 여부 예측 과정



기대효과

- ✓ 대출심사 시간 및 장소의 제약이 줄어든다
- ✓ 대출심사의 접근성이 좋다
- ✓ 자가 대출심사 가능
- ✓ 인도 데이터로 인도시장 진출 가능성
- ✓ 대출업무 효율성 증대

개선사항

- ✓ 한국 데이터가 아닌 인도데이터
  → 한국 실정에 맞지 않을 수 있음
- ✓ 비지도학습 제외(군집화)
- ✓ 새로운 데이터의 DB저장을 통한 예측 모델 업데이트
- ✓ 챗봇 구현
- ✓ 변수의 효과가 확연히 드러나지 않는다
  → 리스크가 큰 사람과 낮은 사람의 구분이 두드러지지 않는다

LRA 웹 구경하러 가겠습니다

감사합니다!