Lending Club의 고객 데이터를 활용한 고객의 대출 상환가능 여부 분류 예측

JH-LEE95

Chapter1. 분석 개요

1. 분석의 동기 및 목적

최근 몇 년간 핀테크 기업이 쏟아져 나오고 있다. 그 중에는 금융기관을 거치지 않고 개인과 개인이 직접 연결되어 채무, 채권 관계를 형성할 수 있도록 하는 P2P (Peer to Peer) 대출을 제공하는 기업도 있다.

하지만 개인대출의 경우 상환능력에 의구심이 들 수 있다. 이에 데이터와 통계를 이용하여 고객의 상환능력을 예측하는 모델을 만들어 위험을 줄이고 더 많은 잠재고객을 확보할 수 있도록 한다.

2. 분석 대상 데이터

분석 대상 데이터는 미국의 P2P 대출 기업인 "Lending Club" 이 Kaggle에서 실제로 제공하고 있는 데이터이다. 우리나라의 "공공데이터 포털"을 비롯한 각 기관에서 제공하는 대출에 대한 데이터를 구하려고 했으나, 공공정보가 아닌 사기업에 대한 정보를 구할 수 없었다. 따라서 Kaggle에서 정보를 구하였다.

대상 데이터는 미국 기업이 제공하는 데이터지만, 이를 기반으로 한국 기업도 적용할 수 있으리라 기대한다.

3. 데이터 분석 방법

분석은 다음과 같은 단계로 진행된다.

1. 데이터 전처리를 위한 준비
2. 변수 확인 및 처리 – 삭제 및 변형
3. 결측치 (Missing values) 처리
4. 데이터 모델링 준비
5. 모델링
6. 언더샘플링 하여 모델링

Chapter2. 단계별 분석 진행

1. 데이터 전처리를 위한 데이터셋 준비

데이터셋을 준비하는 단계로 데이터를 불러오고 확인하는 작업을 한다. 데이터는 74개의 변수로 총 887,000개의 관측치로 이루어져 있다. 이 단계에서 특히 중요한 것은 "Target" 변수를 설정하는 작업이다.

	index	loan_status
0	Current	601779
1	Fully Paid	207723
2	Charged Off	45248
3	Late (31-120 days)	11591
4	Issued	8460
5	In Grace Period	6253
6	Late (16-30 days)	2357
7	Does not meet the credit policy. Status:Fully ...	1988
8	Default	1219
9	Does not meet the credit policy. Status:Charge...	761

이 데이터셋에서 "Target" 변수 는 "loan_status" 컬럼으로 위와 같이 총 10개의 범주를 가지고 있는데, 이 분석에서는 이진분류를 사용하므로 변수를 0,1 두개로 나누어 주는 작업이 필요하다.

이때 이미 상환이 완료된 'Fully Paid'와 'Does not meet the credit policy. Status:Fully Paid' 카테고리를 제외한다.

Ex) "Current" : 상환능력 좋음 (0), "Charged off" 를 비롯한 나머지 : 상환능력 나쁨 (1)

2. 변수 확인 및 처리 – 삭제 및 변형

본격적인 데이터 전처리 단계로 각 컬럼의 내용 및 데이터 타입을 확인한다. 분석에 활용될 만한 컬럼인지 아닌지 확인하는 단계이다. 이때 분석에 활용하면서 숫자형 변수로 바꿀 수 있는 경우 적절히 바꾸는 작업을 한다. 또한 분석에 활용되지 않는 컬럼은 삭제한다.

EX1) "title" 컬럼은 대출의 목적을 알려주는 컬럼으로 카테고리가 지나치게 많아 지협적이다. 그리고 "purpose" 라는 대출 목적을 알려주는 컬럼이 또 있으므로, "title" 컬럼은 분석에서 활용하지 않는다.

EX2)"grade" 컬럼은 사용자의 신용등급을 보여주는 컬럼으로 A~G 까지의 스트링 타입으로 이루어져 있다. 이를 0~6 까지 숫자로 바꾸어 준다.

3. 결측치(Missing Values) 처리

결측치를 가지고 있는 변수들을 처리하는 단계이다. 각 컬럼을 확인하여 결측치가 70% 이상인 경우 삭제하였다. 그 외 소수의 결측치가 있는 경우 변수의 특성에 맞게 적절히 바꿔준다.

EX1) "open_acc_6m" 컬럼은 결측치가 96.87% 으로 삭제한다.

EX2)"tot_cur_bal" 컬럼의 결측치들은 mean 값으로 대치하였다.

또한 앞서 2단계와 지금의 3단계를 거치면서 887379개의 데이터가 677668개로 감소했다.

4. 데이터 모델링 준비

본격적인 모델링에 앞서 데이터를 준비하는 단계이다.

먼저 데이터 프레임에서 종속변수와 독립변수를 분리한다. 종속변수는 앞서 "Target"으로 설정한 변수이고, 독립변수는 데이터프레임에서 "Target"을 제외한 것들이다.

또한 변수들간 스케일이 다르므로 적절한 비교를 위해 정규화를 진행한다.

마지막으로 숫자형 변수가 아닌 경우 더미화 한다.

5. 모델링

실제로 모델링을 적용하고 비교한다. 사용된 모델은 다음과 같다.

[Logistic Regression, SVM, Random Forest, KNN, Naïve Bayes]

(5가지 모델을 모두 측정하려고 했으나, 실제 코딩 과정에서 SVM과 KNN 모델은 로딩에 너무 오랜 시간이 걸려서 포기하였습니다. 양해 부탁드립니다.)

모델간 비교를 위해 사용될 기준은 다음과 같다.

1) Accuracy : 전체 데이터 중 제대로 분류된 비율이다. 즉 0은 0(상환능력 좋음)으로 1은 1(상환능력 나쁨)로 제대로 분류했는지에 대한 척도이다.

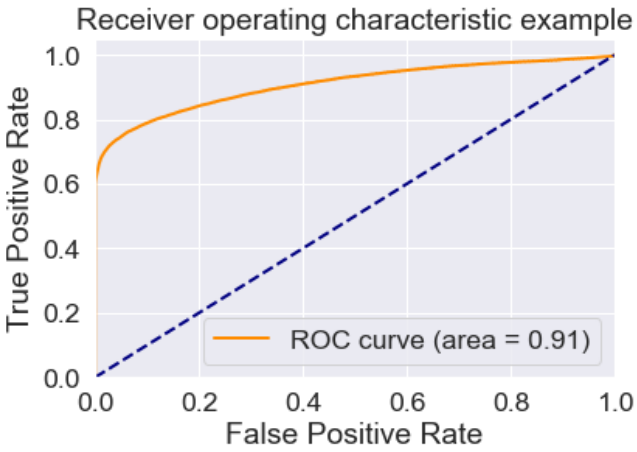
2) Recall : 전체 정답 중 실제로 정답을 몇 개를 맞추었는지에 대한 척도이다. 이는 오답을 개수는 고려하지 않는다.

3) Precision : 정답이라고 예측한 것 중 실제 정답의 비율이다. 이 분석의 목적은 상환능력을 예측하여 위험을 줄이는 데 있다. 잘못된 예측이 회사에 큰 위험을 가져올 수 있는 것을 감안하여, 오답을 고려하지 않는 "Recall" 보다 "Precision"을 더 우선순위에 둔다.

4) ROC , AUC : 민감도와 특이도를 고려하여 한 눈에 모델의 성능을 알아 볼 수 있게 해주는 척도이다. 만약 모델이 정답을 정답이라고 예측할 확률이 높고 오답을 오답이라고 예측할 확률이 높다면 Roc 커브의 그래프는 좌측 상단에 수렴하게 된다. Roc curve의 밑면적이 넓을수록 AUC가 1에 가까울수록 성능이 좋다. 모델 선택의 최우선기준으로 삼는다.

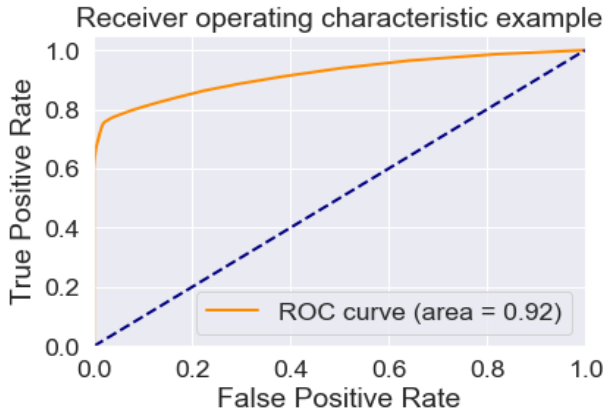
1) Logistic Regression

```
Accuracy:  0.96
Recall:  0.62
Precision 0.97
ROC AUC score:  0.9080997826383126
```



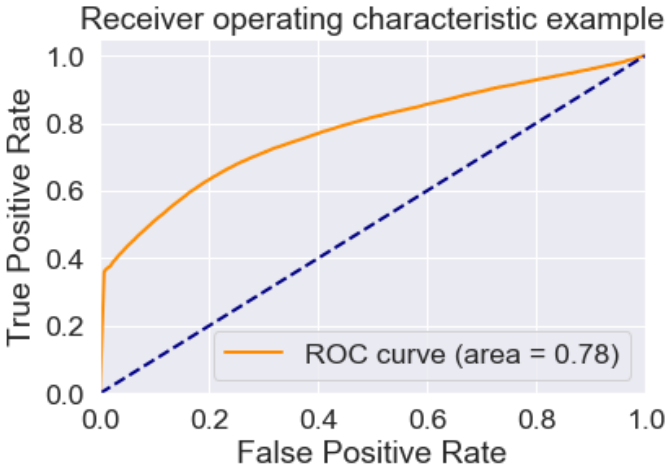
2) Random Forest

```
Accuracy:  0.96
Recall:  0.68
Precision 0.93
ROC AUC score:  0.9175468766052938
```



3) Naïve Bayes

```
Accuracy:  0.91
Recall:  0.39
Precision 0.67
ROC AUC score:  0.7754610833971224
```



3 가지 모델 모두 상당한 수준의 "Accuracy"를 보여주고 있다. 하지만 Naïve Bayes 모델은 "Recall" 과 "Precision" 이 다른 두 모델에 비해 현저히 낮게 나오고 있다. 따라서 Naïve Bayes 모델은 고려대상에서 제외한다.

ROC 면적을 기준으로 살펴보았을 때 Random Forest 모델이 0.92로 가장 높은 성능을 발휘하는 모델로 판명되므로 이를 채택한다.

만약 기업이 채무불이행 위험에 대하여 조금 더 민감하게 고려한다면 Precision이 0.97로 가장 높은 Logistic Regression 모델을 채택할 수 있다.

하지만 이 데이터셋은 타겟변수의 클래스간 비율 차이가 심하게 난다. 다음과 같이 0(상환능력 좋음)과 1(상환능력 나쁨) 이 601779:75889의 차이를 보여주고 있다. 이 경우 0을 우선시하는 모델의 성능이 높게 측정될 수밖에 없다. 따라서 클래스의 비율을 적절히 조절하여 측정할 필요가 있다. 이에 대한 방법은 다음 과정에서 소개한다.

	rating
0	601779
1	75889

6. 언더샘플링을 통한 모델링

분석에 활용될 데이터는 다음과 같이 클래스 간의 비율 차이가 심하게 난다.

	rating
0	601779
1	75889

이 경우 단순히 우세한 클래스를 택하는 모형의 정확도가 높아지므로,

모형 간의 비교가 어려워진다. 이런 "비대칭 데이터 문제"를 해결하기 위해

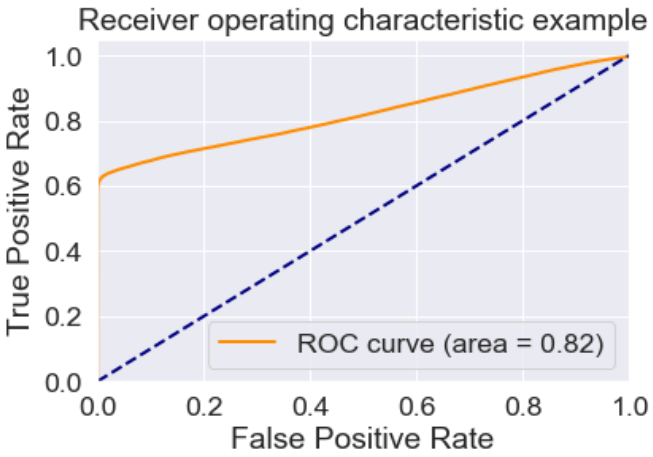
다수 클래스의 데이터에서 일부만 사용하는 언더샘플링을 사용하여 다시 한번 모델링을

시도한다. 그리고 모델 간의 비교를 통해 가장 좋은 성능을 가진 모델을 찾아낸다.

또한 언더샘플링을 하기 전과 후의 성능차이를 비교한다.

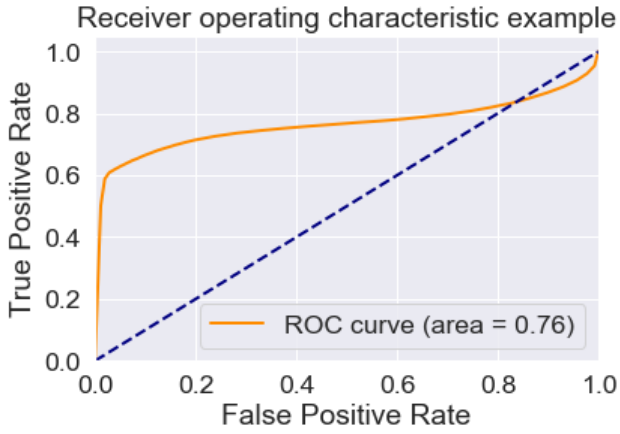
1) Logistic Regression

```
Accuracy score :  0.8954310960529345
Recall :  0.6674485103242894
Precision: 0.5245005692992444
ROC AUC score : 0.8204189263001109
```



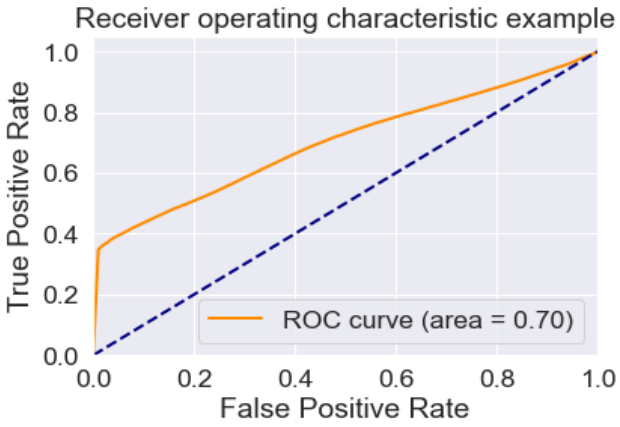
2) Random Forest

```
Accuracy score :  0.5949683325758336
Recall :  0.7587133840213997
Precision: 0.18293454585923466
ROC AUC:  0.7637561656946266
```



3) Naïve Bayes

```
Accuracy score :  0.9006814546356032
Recall :  0.3788691378197104
Precision: 0.5868519726116727
ROC AUC score:  0.701340630951475
```



이전 단계에서 모델 성능의 인플레이션이 심했던 것에 비하여 전체적으로 성능이 하향되었다. 언더샘플링을 통해 클래스간 데이터 비율을 균등하게 맞추어놨다. 이로 인해 단순히 우세한 클래스를 택하여 모델 성능이 높게 측정되는 문제를 해결하였다.

주목할 만한 점은 언더샘플링 이후 Random Forest의 성능이 급격히 낮아졌다는 것이다. 오히려 이전 단계에서 가장 성능이 좋지 않았던 Naïve Bayes가 언더샘플링 이후에도 준수한 성능을 보여주고 있다. 또한 Logistic Regression 은 성능의 급격한 하향없이 모델 중 가장 높은 수준의 성능을 보여주고 있다. 따라서 Logistic Regression 모델을 채택한다.

Chapter3. 결론 및 한계점과 향후 개선점

채무자의 신용정보를 이용하여 대출상환 가능 여부를 예측하는 모델을 만들어보았다.

"Lending Club" 과 같은 핀테크 기업뿐만 아니라 일반 금융기관도 이 모델을 이용하면

채무불이행으로부터 오는 위험을 줄이고, 신뢰 있는 잠재고객을 확보할 수 있다.

고객의 입장에서 살펴보자면, 대출이 필요한 사람이 자신의 신용정보를 입력하면 자신의 상환능력을 확인할 수 있으므로, 은행의 대출 여부를 예측할 수 있다.

만약 모델이 상환능력이 없다고 판단한다면, 고객이 자신의 신용정보 값을 적절히 조절하여 어떤 점을 개선하면 상환가능여부를 "YES"로 만들 수 있는지도 예측 가능하다.

하지만 모델 설정에 있어서 각 모델마다 파라미터값을 조정하지 않고 단순 비교를 하였기 때문에 파라미터값의 조절에 따른 성능의 변화와 비교는 측정하지 못하였다.

두번째로 이 모델은 단순히 상환가능과 불가능을 예측하는 이진분류이기 때문에 고객의 상환능력에 대한 정밀한 판단이 어렵다.

이를 개선하기 위해 다중클래스분류 모델을 사용하여 더 세분화한다면 좀 더 많은 잠재고객을 확

보할 수 있을 것이다.

또한 분류가 아닌 상환 확률을 예측하는 모델을 만들 수도 있을 것이다. 이 경우 각 변수에 기업의 방침에 따라 가중치를 설정하여 기업이 원하는 고객을 확보할 수 있으리라 기대한다.