# Reprot for CA3 (상용 온라인 게임 데이터 분석)

20165174 Jwa Younkyung

## 1 Introduction and Related Work

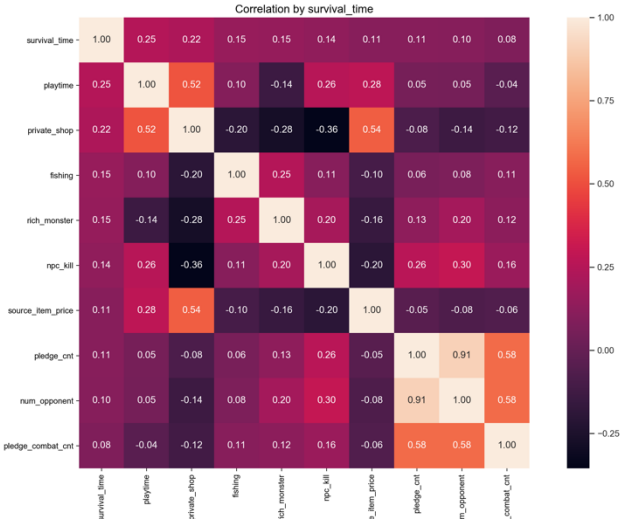
상용 온라인 게임(리니지)의 데이터를 활용하여 잔존 가치를 고려해 이탈 예측 모델을 제작하였다. preprocessing을 통해 필요한 input 데이터를 만들고 적절한 model을 활용하여 예상되는 y값 데이터를 만들었다.

## 2 Approach

1. Preprocess
    1. activity, combat : server 데이터를 제외한 나머지 데이터를 같은 day, acc_id를 가진 것들끼리 합하였다. 합한 값을 다시 acc_id를 기준으로 평균을 내고 day값들을 제거하였다.
    2. pledge :server, pledge_id를 제외한 나머지 데이터를 같은 day, acc_id를 가진 것들끼리 합하였다. 합한 값을 다시 acc_id를 기준으로 평균을 내고 day값들을 제거하였다.
    3. trade : source_acc_id와 target_acc_id 각각을 기준으로 price, item_amount, item_price 값을 가진 데이터를 만들고 이를 같은 day, acc_id를 가진 것들끼리 합하였다. 합한 값을 다시 acc_id를 기준으로 평균을 내고 day값들을 제거하였다.

    1,2,3의 데이터를 acc_id를 기준으로 하나의 csv파일을 만들었다.

    다음은 survival_time에 기준한 correlation table이다. Playtime, private_shop, fishing 등이 높은 관계를 보여주었다.



2. Trainig
    1. Find best regression

        /etc/findRegression.py를 활용하여 y값인 survival_time과 amount_spent를 예측하는데 가장 적은 error를 만들어내는 regression을 찾았다. Regression은 sklearn에 있는 LinearRegression, Lasso, ElasticNet, KneighborsRegressor, DecisionTreeRegressor 그리고 GradientBoostingRegreesor(GB)을 이용하였다. 정확도를 향상시키기 위해서 위 데이터들을 검증할 때 k cross validation기법을 사용하였다.

        a. survival_time

            여러 regression을 이용해 survival_time을 예측하면 neg_mean_squared_error는 다음과 같다. 따라서 가장 좋은 regression은 GB이다.

```
ScaledLR: -427.446974 (7.617748)
ScaledLASSO: -446.615602 (5.246596)
ScaledEN: -452.823152 (5.751694)
ScaledKNN: -398.864295 (8.758335)
ScaledCART: -645.130205 (15.932648)
ScaledGBM: -360.469484 (7.687708)
```

        b. amount_spent

            amount_spent의 neg_mean_squared_error는 다음과 같다. 이 또한 가장 좋은 regression은 GB이다.

```
ScaledLR: -0.513663 (0.378008)
ScaledLASSO: -0.521744 (0.383309)
ScaledEN: -0.521744 (0.383309)
ScaledKNN: -0.582247 (0.353891)
ScaledCART: -1.004271 (0.401919)
ScaledGBM: -0.487257 (0.365181)
```

        위 결과에 따라서 두가지 모두 GB Model을 사용하여 모델을 만들었다.
        GB Model은 hyperparmeter로 'n_estimators'가 있다 어떤 값이 각 y값에 최적화 되어있는지 알아보기 위해서 /etc/findEstimator.py를 이용하여 최적화된 값을 찾아보았다.

        a. survival_time

            n_estimator = 400 이 가장 좋은 모델을 만들었다.

```
-369.720374 with {'n_estimators': 50}
-360.493259 with {'n_estimators': 100}
-352.310570 with {'n_estimators': 200}
-347.327646 with {'n_estimators': 300}
-343.806671 with {'n_estimators': 400}
Best: -369.720374 using {'n_estimators': 400}
```

        b. amount_spent

            n_estimator = 50 이 가장 좋은 모델을 만들었다.

```
-0.485084 with: {'n_estimators': 50}
-0.485588 with: {'n_estimators': 100}
-0.498588 with: {'n_estimators': 200}
-0.505727 with: {'n_estimators': 300}
-0.520982 with: {'n_estimators': 400}
```

```
Best: -0.485084 using {'n_estimators': 50}
```

따라서 데이터 모델을 형성 하기 위해 다음과 같은 코드를 작성하였다.

```
model_Grad_time = GradientBoostingRegressor(random_state=21, n_estimators=400)
model_Grad_spent = GradientBoostingRegressor(random_state=21, n_estimators=50)
```

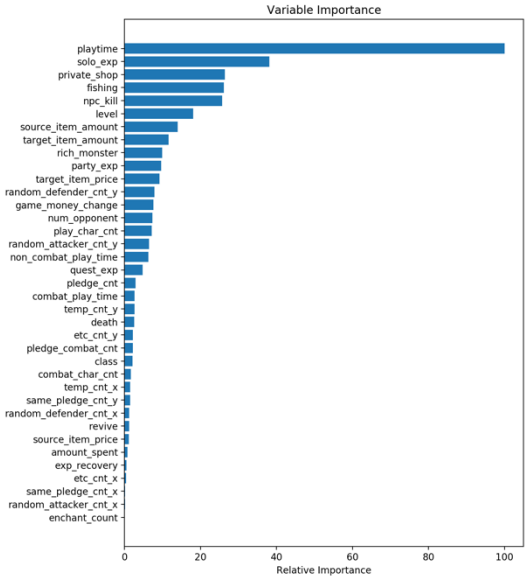
그 결과 채점 결과 값이 다음과 같다.

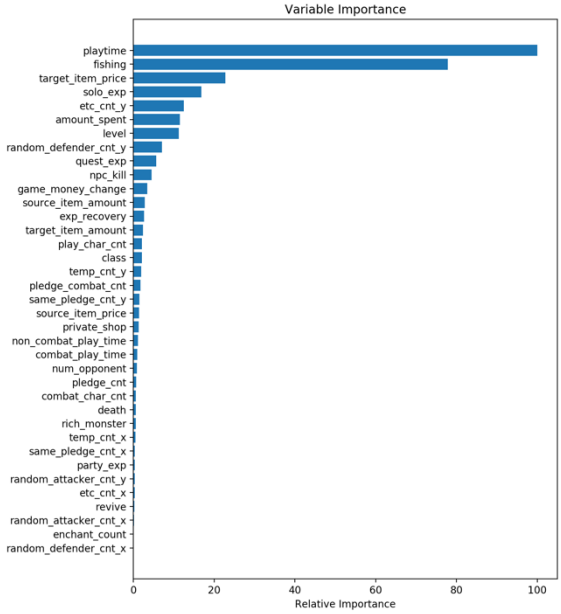
# 3 Experimental Results

1. Feature importance

   왼쪽 그래프는 survival_time을 모델에 따라 예측했을때 feature importance이다. Playtime, solo_exp, private_shop순으로 높은 중요도를 나타낸다.

   오른쪽 그래프는 amount_spent를 모델에 따라 예측했을 때의 feature importance이다. Playtime, fishing, target_item_price순으로 높은 중요도를 나타낸다.




중요도가 높지 않은 feature들은 향후 model을 tuning할 때 삭제를 할 수 있을 것이라 여겨진다.

2. Leaderboard result

| Test1 | Test2 | Total |
|---|---|---|
| 1263.3773 | 1404.2437 | 2667.6210 |

# 4 Conclusion

게임 이탈 고객의 데이터 예측을 위한 최적의 regression모델은 GradientBoostingRegreesor이고 feature deletion을 통해서 모델을 더 tuning할 수 있을 것이다. Survival_time은 discrete한 값을 가지고 있으므로 향후 연구를 진행한다면 regression이 아닌 classification을 통한 예측이 가능 할 것이다.