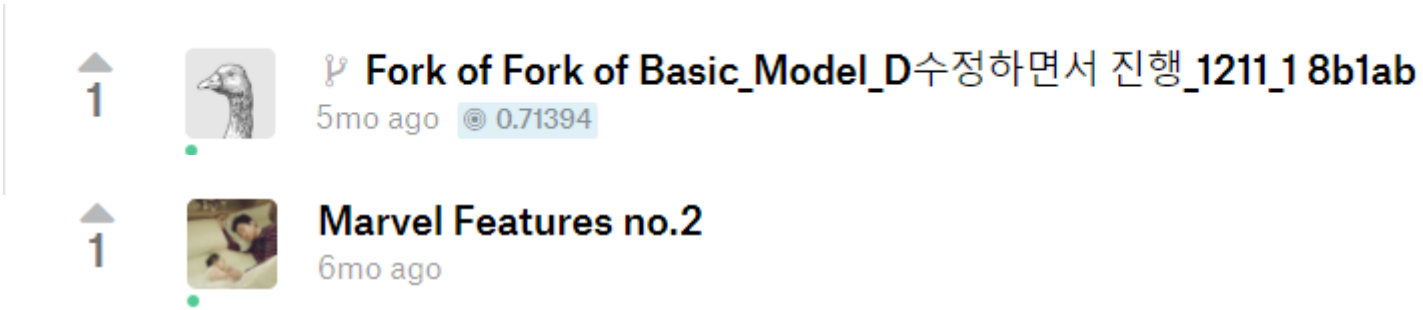
1Round 분석과정

김태욱 정재엽

1. Feature



X_train1

Fork of Fork의 첫번째 feature 목록

X_train2

Marvel Features no.2

X_train3

Fork of Fork의 PCA적용 feature

2. Features Scaling

수치가 크게 다른 변수들이 있어 scaling -> 데이터셋 3개 생성(X_train_scaled 1,2,3)

Scaling

In [142]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
C:₩Users₩user₩Anaconda3₩lib₩site-packages₩sklearn₩preprocessing₩data.py:645: DataConversionWarning: Data with input dtype int32, int64, float64 were all converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
C:₩Users₩user₩Anaconda3₩lib₩site-packages₩sklearn₩base.py:464: DataConversionWarning: Data with input dtype int32, int64, float64 were all converted to float64 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
C:₩Users₩user₩Anaconda3₩lib₩site-packages₩ipykernel_launcher.py:5: DataConversionWarning: Data with input dtype int32, int64, float64 were all converted to float64 by StandardScaler.
  """
```

In [146]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train2_scaled = scaler.fit_transform(X_train_new)
X_test2_scaled = scaler.transform(X_test_new)
```

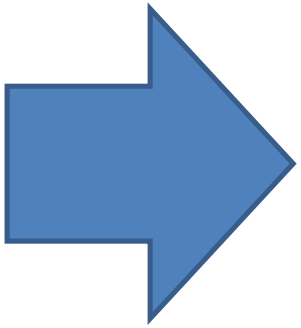
In [144]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train3_scaled = scaler.fit_transform(X_train3)
X_test3_scaled = scaler.transform(X_test3)
```

```
C:₩Users₩user₩Anaconda3₩lib₩site-packages₩sklearn₩preprocessing₩data.py:645: DataConversionWarning: Data with input dtype int64, float64 were all converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
C:₩Users₩user₩Anaconda3₩lib₩site-packages₩sklearn₩base.py:464: DataConversionWarning: Data with input dtype int64, float64 were all converted to float64 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
C:₩Users₩user₩Anaconda3₩lib₩site-packages₩ipykernel_launcher.py:5: DataConversionWarning: Data with input dtype int64, float64 were all converted to float64 by StandardScaler.
  """
```

3. Imbalanced Learning.

성별	비율
여성(0)	69.6067%
남성(1)	30.3933%

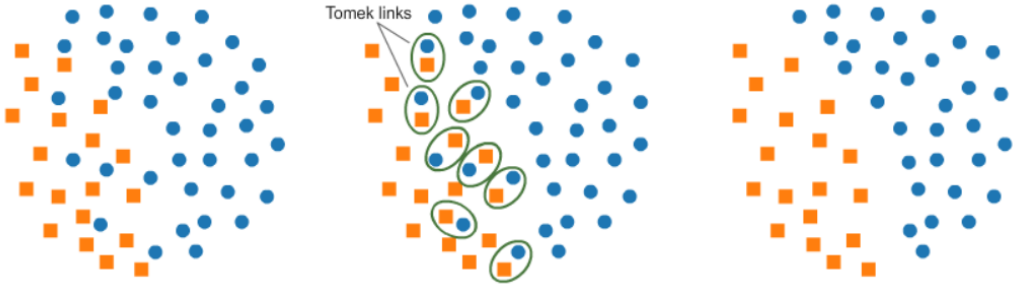


Imbalaned 처리
method 사용

3. Imbalanced Learning.

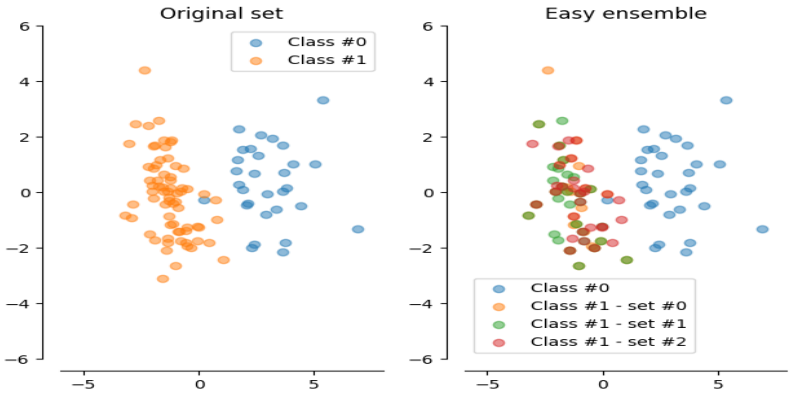
사용모델 -> 모델들을 Train_scaled1,2,3 에 모두 적용함(데이터셋 9개)

1. TomekLink



2. RandomUnder Sampler

3. EasyEnsemble



Train3에 적용할 때는 memory error 때문에 샘플링을 통해 데이터를 1/5로 줄임

4. Model selection

- ## XGBClassifier

  성능이 잘나옴!

- ## LGBMClassifier

  튜닝 및 적합속도가 빠름!

5. Model Parameter Tuning

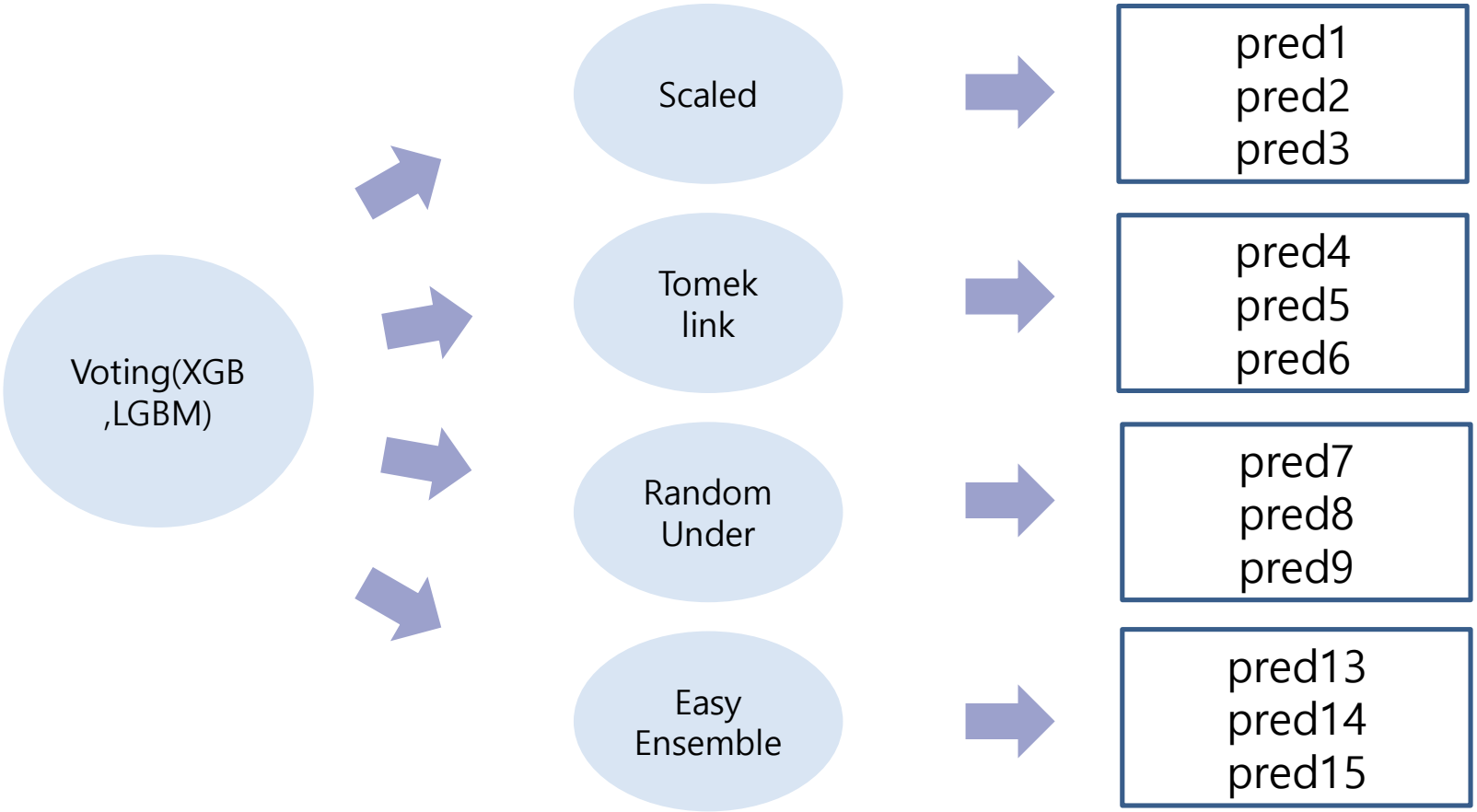
```
#각 케이스 XGB 파라미터 튜닝
XGB1 =XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=1, eta=0.09, gamma=0, learning_rate=0.1,
       max_delta_step=0, max_depth=4, min_child_weight=1, missing=None,
       n_estimators=100, n_jobs=1, nthread=None,
       objective='binary:logistic', random_state=0, reg_alpha=0,
       reg_lambda=1, scale_pos_weight=0.5, seed=None, silent=True,
       subsample=0.7999999999999999, xgb__max_depth=14, xgb__subsample=0.4)
XGB2 = XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=1, eta=0.09, gamma=0, learning_rate=0.1,
       max_delta_step=0, max_depth=4, min_child_weight=1, missing=None,
       n_estimators=100, n_jobs=1, nthread=None,
       objective='binary:logistic', random_state=0, reg_alpha=0,
       reg_lambda=1, scale_pos_weight=0.5, seed=None, silent=True,
       subsample=0.7999999999999999, xgb__max_depth=14, xgb__subsample=0.4)
XGB3 = XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=1, eta=0.02, gamma=0, learning_rate=0.1,
       max_delta_step=0, max_depth=4, min_child_weight=1, missing=None,
       n_estimators=100, n_jobs=1, nthread=None,
       objective='binary:logistic', random_state=0, reg_alpha=0,
       reg_lambda=1, scale_pos_weight=0.5, seed=None, silent=True,
       subsample=0.8999999999999999, xgb__max_depth=14, xgb__subsample=0.4)
```

```
from lightgbm import LGBMClassifier
LGBM1 = LGBMClassifier(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
        importance_type='split', learning_rate=0.1, max_depth=6,
        min_child_samples=20, min_child_weight=0.001, min_data_in_leaf=600,
        min_split_gain=0.0, n_estimators=100, n_jobs=1, num_leaves=90,
        objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,
        silent=True, subsample=1.0, subsample_for_bin=200000,
        subsample_freq=0)
LGBM2 = LGBMClassifier(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
        importance_type='split', learning_rate=0.1, max_depth=6,
        min_child_samples=20, min_child_weight=0.001, min_data_in_leaf=900,
        min_split_gain=0.0, n_estimators=100, n_jobs=1, num_leaves=70,
        objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,
        silent=True, subsample=1.0, subsample_for_bin=200000,
        subsample_freq=0)
LGBM3 = LGBMClassifier(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
        importance_type='split', learning_rate=0.1, max_depth=7,
        min_child_samples=20, min_child_weight=0.001, min_data_in_leaf=800,
        min_split_gain=0.0, n_estimators=100, n_jobs=1, num_leaves=70,
        objective=None, random_state=None, reg_alpha=0.0, reg_lambda=0.0,
        silent=True, subsample=1.0, subsample_for_bin=200000,
        subsample_freq=0)
```

각 Train 데이터별로
따로 Tunning

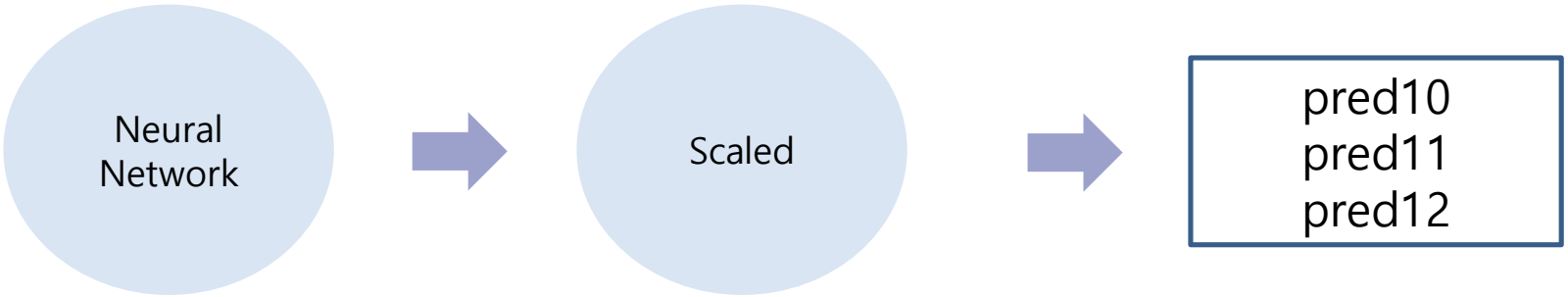
6. Model Ensemble

데이터셋 : 12개(scaled 3개, imbalanced 9개)
앙상블 기법 : Voting classifier



6. Model Ensemble

데이터셋 : 3개 (neural network 3개)



6. Model Ensemble (pred 선택)

데이터셋 : 12개(scaled 2개, imbalanced 7개 , neural network 3개)
앙상블 기법 : 산술평균

7. Result

- **Feature**

  Imbalanced Learning

- **Model**

  Trial & Error