Unbalanced multiclass data with XGBoost

datascience.stackexchange.com/questions/16342/unbalanced-multiclass-data-with-xgboost

shda 48511 gold badge55 silver badges1010 bronze badges

For sklearn version < 0.19

Just assign each entry of your train data its class weight. First get the class weights with `class_weight.compute_class_weight` of sklearn then assign each row of the train data its appropriate weight.

I assume here that the train data has the column `class` containing the class number. I assumed also that there are `nb_classes` that are from 1 to `nb_classes`.

```
from sklearn.utils import class_weight
classes_weights = list(class_weight.compute_class_weight('balanced',
                                                 np.unique(train_df['class']),
                                                 train_df['class']))

weights = np.ones(y_train.shape[0], dtype = 'float')
for i, val in enumerate(y_train):
    weights[i] = classes_weights[val-1]

xgb_classifier.fit(X, y, sample_weight=weights)
```

Update for sklearn version >= 0.19

There is simpler solution

```
from sklearn.utils import class_weight
classes_weights = class_weight.compute_sample_weight(
    class_weight='balanced',
    y=train_df['class']
)

xgb_classifier.fit(X, y, sample_weight=classes_weights)
```