Note

Click here to download the full example code

Multiclass classification with under-sampling

Some balancing methods allow for balancing dataset with multiples classes. We provide an example to illustrate the use of those methods which do not differ from the binary case.

Out:

```
Training target statistics: Counter({1: 38, 2: 38, 0: 17})
Testing target statistics: Counter({1: 12, 2: 12, 0: 8})
                   pre       rec       spe        f1       geo       iba       sup

          0       1.00      1.00      1.00      1.00      1.00      1.00         8
          1       1.00      0.83      1.00      0.91      0.91      0.82        12
          2       0.86      1.00      0.90      0.92      0.95      0.91        12

avg / total       0.95      0.94      0.96      0.94      0.95      0.90        32
```

```python
# Authors: Guillaume Lemaitre <g.lemaitre58@gmail.com>
# License: MIT

from collections import Counter

from sklearn.datasets import load_iris
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split

from imblearn.datasets import make_imbalance
from imblearn.under_sampling import NearMiss
from imblearn.pipeline import make_pipeline
from imblearn.metrics import classification_report_imbalanced

print(__doc__)

RANDOM_STATE = 42

# Create a folder to fetch the dataset
iris = load_iris()
X, y = make_imbalance(iris.data, iris.target,
                      sampling_strategy={0: 25, 1: 50, 2: 50},
                      random_state=RANDOM_STATE)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, random_state=RANDOM_STATE)

print('Training target statistics: {}'.format(Counter(y_train)))
print('Testing target statistics: {}'.format(Counter(y_test)))

# Create a pipeline
pipeline = make_pipeline(NearMiss(version=2),
                         LinearSVC(random_state=RANDOM_STATE))
pipeline.fit(X_train, y_train)

# Classify and report the results
print(classification_report_imbalanced(y_test, pipeline.predict(X_test)))
```

Total running time of the script: (0 minutes 0.041 seconds)

Download Python source code: plot_multi_class_under_sampling.py

Download Jupyter notebook: plot_multi_class_under_sampling.ipynb