ELITE DATA SCIENCE

PYTHON CHEATSHEET: HANDLING IMBALANCED CLASSES

This Python cheatsheet will cover some of the most useful methods for handling machine learning datasets that have a disproportionate ratio of observations in each class. These "imbalanced" classes render standard accuracy metrics useless.

To see the most up-to-date full tutorial and download the sample dataset, visit the online tutorial at elitedatascience.com.

SETUP

Make sure the following are installed on your computer:

- Python 2.7+ or Python 3
- NumPy
- Pandas
- Scikit-Learn (a.k.a. sklearn)

LOAD SAMPLE DATASET

```
import pandas as pd
import numpy as np

df = pd.read_csv('balance-scale.data',
                 names=['balance', 'var1', 'var2', 'var3', 'var4'])
```

*Up-to-date link to the sample dataset can be found here.

UP-SAMPLE MINORITY CLASS

```
df_majority = df[df.balance==0]
df_minority = df[df.balance==1]

df_minority_upsampled = resample(df_minority,
                                 replace=False,
                                 n_samples=49,
                                 random_state=123)

df_upsampled = pd.concat([df_majority, df_minority_upsampled])
```

DOWN-SAMPLE MAJORITY CLASS

```
df_majority = df[df.balance==0]
df_minority = df[df.balance==1]

df_majority_downsampled = resample(df_majority,
                                   replace=False,
                                   n_samples=49,
                                   random_state=123)

df_downsampled = pd.concat([df_majority_downsampled, df_minority])
```

CHANGE YOUR PERFORMANCE METRIC

```
from sklearn.metrics import roc_auc_score

prob_y_2 = clf_2.predict_proba(X)
prob_y_2 = [p[1] for p in prob_y_2]
print( roc_auc_score(y, prob_y_2) )
```

USE COST-SENSITIVE ALGORITHMS

```
from sklearn.svm import SVC
clf = SVC(kernel='linear', class_weight='balanced', probability=True)
```

USE TREE-BASED ALGORITHMS

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier()
```

Honorable Mentions

- Create Synthetic Samples (Data Augmentation) - A close cousin of upsampling.
- Combine Minority Classes - Group together similar classes.
- Reframe as Anomaly Detection - Treat minority classes as outliers.

To see the most up-to-date full tutorial, explanations, and additional context, visit the online tutorial at elitedatascience.com. We also have plenty of other tutorials and guides.