## Objectives

Some classification models are better suited than others to outliers, low occurrence of a class, or rare events. The most common methods to add robustness to a classifier are related to stratified sampling to re-balance the training data. This module will walk you through both stratified sampling methods and more novel approaches to model data sets with unbalanced classes.

* Identify class weights and sampling as methods to deal with unbalanced classes in a data set.
* Recognize the syntax for building for sampling, blagging, and nearest neighbor methods for modeling unbalanced classes.

Unbalanced classes

Classifiers – are usually built to optimimze accuracy and hence will often perform poorly on unbalanced classes.

For unbalanced datasets, we can balance the size of the classes by either downsampling the larger class or upsampling the small one.

We can also do a mix of the two.

Unbalanced classes

Steps for unbalanced datasets:

* Do a stratified test-train split
* Up or down sample the full dataset
* Build models

Unbalanced classes

With unbablanced classes, the data often isn’t easily separable. We must choose to make sacrifices to one class or the other

For every mino-class data-point identified as such, we might wrongly label a few major-class points as minor –class

Precision will likely go down

Downsampling adds importance to the minor class, typically shooting recall and brining down precision

Values like .8 recall and .15 precision ins’t uncommon

Upsampling mitigates some of the excessive weight on the minor class recall is still typically higher than precision, but the gap is lesser

Values like.7 recall and .4 precision isn’t uncommon and are often considered good results for an unbbalanced dataset.

Cross validation works for any global model-making choice including sampling

Every classifier used produces a diff. Model

Everydataset we use (prouced by carious sampling, say) produces a diff. Model.

We can choose the best model using any criteria including AUC (area under the curve). Each model produces a diff. ROC curve

Once a model is chose. You can walk along the ROC curve and pick any point on it. Each point has diff. Precision/recall values.

Approaches

* General sklearn approaches
* Oversampling
* Undersampling
* Combination
* Ensembles
* Scikit-learn.org/imbalanced

Weighting

Many models allow weighted ovservations

* Adjust these so total weights are equal across classes
* Easy to do, when it’s available
* No need to sacrifice data

Stratified sampling

* Train-test split, ‘stratify’ option
* ShuffleSplit -> StratifiedShuffleSplit
* KFold -> StratifiedKFold -> RepeatedStatifiedKFold

Random oversampling

* Simplist oversampling approach
* Resample with replacement from minority class
* No concerns about geometry of feature space
* Good for categorical data

Synthetic oversampling

* Start with a point in the minority class
* Choose one of k nearest neighbors
* Add a new point between them
* Two main approaches:
  + Smote
  + Adasyn

SMOTE: Synthetic Minority Oversampling Technique

* Regular: connect minority class points to any neighbor (evern other calsses)
* Borderline: classify points as outlier, safe, or in-danger
  + Connect minority in-danger points only to minority points
  + Connect minority in-danger points to whatever is nearby
* SVM: use minority support vectors to generate new points

ADASYN: ADAptive SYNthetic sampling

* For each minority point:
  + Look at classes in neighborhood
  + Generate new samples proportional to competing classes
* Motivated by KNN, but helps other classifiers as well

NearMiss-1

Tomek links: mixed mutual nearest neighbors

Edited nearest neighbors

* Remove points that don’t agree with neighbors

Combination over/under

* Smote + tomek’s link
* Smote + edited Nearest Neighbors

Blagging (Balanced Bagging)

* Take bootstrap samples from the original population
* Balanced each sample by downsampling
* Learn a decision tree from each
* Majority vote
* Result: bagged decision trees learned on balanced populations

Unbalanced classes:

* All of this happens after the test set has been split
* Use sensible metrics
  + Auc
  + F1
  + Cohen's kappa
  + Not accuracy – to easily fooled

## Course Project Self-Review

Self-Evaluate your course project submission.

1. Does the report include a section describing the data? The summary of the data should help understand the features available and how they will be used for prediction or interpretation.

* No points awarded if there is no summary or it is hard to put together what variables are available or how they might be used. (0 pts)
* One point if there is a basic summary, like a data dictionary. (1 pt)
* One extra point if the summary of the data is presented with graphs of distributions and plots that show the relation between features and the outcome variable. (2 pts)

2. Does the report include a paragraph detailing the main objective of this analysis?

* This report is missing a planning section for the data analysis (0 pts)
* Yes. This plan includes a detailed subtask section or a good vision of what is possible to do with this data set. (1 pt)
* This plan exceeds expectations. In addition to plan out subtasks and vision for this analysis, it also anticipates possible snags that might be incorporated into preliminary hypothesis of the data. (2 pts)

3. Classification models - Does the report include a section with variations of classification models and which one is the model that best suits the main objectives of this analysis?

* No. There was no classification model on this analysis. (0 pts)
* Yes. At least one classification model is included and it discusses findings and results appropriately. (1 pt)
* Yes, there are at least 2 different classifier models. One of them is presented as the better alternative, and some findings are presented. (2 pts)

4. Insights and key findings - Does the report include a clear and well presented section with key findings about the problem and next steps?

* No. There are no takeaways, insights, or findings about this problem. (0 pts)
* Yes. Some good takeaways and findings derived from the model are presented. (1 pt)
* Yes. Takeaways and findings derived from the model are well presented. The quality of insights or the next steps section award this section an extra point. (2 pts)

5. Next steps - No model is perfect and it is valuable to highlight aspects of the analysis worth revisiting. Does the report highlight possible flaws in the model and a plan of action to revisit this analysis with additional data or different predictive modeling techniques?

* No. There is no mention of possible flaws or plans to revisit the analysis. (0 pts)
* Yes. There is some discussion presented on possible flaws of this model and a plan to revisit this with additional data or different predictive modeling techniques. (1 pt)
* Yes. There is a comprehensive list of possible flaws of this model and a detailed plan to revisit this with additional data or different predictive modeling techniques. The quality of this section awards it an extra point. (2 pts)

If you scored less than 5 points, please review the lessons taught in this course again.

## End of module review: Modeling Unbalanced Classes

### **Modeling Unbalanced Classes**

Classification algorithms are built to optimize accuracy, which makes it challenging to create a model when there is not a balance across the number of observations of different classes. Common methods to approach balancing the classes are:

* Downsampling or removing observations from the most common class
* Upsampling or duplicating observations from the rarest class or classes
* A mix of downsampling and upsampling

### **Modeling Approaches for Unbalanced Classes**

Specific algorithms to upsample and downsample are:

* Stratified sampling
* Random oversampling
* Synthetic oversampling, the main two approaches being Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic sampling (ADASYN)
* Cluster Centroids implementations like NearMiss, Tomek Links, and Nearest Neighbors