**PREPROCESSING FOR MACHINE LEARNING IN PYTHON**

**DataCamp.com**

# Class imbalance

In the volunteer dataset, we're thinking about trying to predict the category\_desc variable using the other features in the dataset. First, though, we need to know what the class distribution (and imbalance) is for that label.

Which descriptions occur less than 50 times in the volunteer dataset?

* The dataset volunteer has been provided.
* The colum you want to check is category\_desc.
* Use the value\_counts() method to check variable counts.

**# Check variable counts**

**In [1]: volunteer['category\_desc'].value\_counts()**

**Out[1]:**

**Strengthening Communities 307**

**Helping Neighbors in Need 119**

**Education 92**

**Health 52**

**Environment 32**

**Emergency Preparedness 15**

**Name: category\_desc, dtype: int64**

# **Stratified sampling**

We know that the distribution of variables in the category\_desc column in the volunteer dataset is uneven. If we wanted to train a model to try to predict category\_desc, we would want to train the model on a sample of data that is representative of the entire dataset. Stratified sampling is a way to achieve this.

**# Create a data with all columns except category\_desc**

**volunteer\_X = volunteer.drop("category\_desc", axis=1)**

**# Create a category\_desc labels dataset**

**volunteer\_y = volunteer[["category\_desc"]]**

**# Use stratified sampling to split up the dataset according to the volunteer\_y dataset**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(volunteer\_X, volunteer\_y, stratify=volunteer\_y)**

**# Print out the category\_desc counts on the training y labels**

**print(y\_train["category\_desc"].value\_counts())**

**<script.py> output:**

**Strengthening Communities 230**

**Helping Neighbors in Need 89**

**Education 69**

**Health 39**

**Environment 24**

**Emergency Preparedness 11**

**Name: category\_desc, dtype: int64**