Imbalanced Classes

Fintech

Lesson 12.3

1

Class Objectives

By the end of the class, you will be able to:

- Define model evaluation metrics and understand the pros and cons of each metric as applied imbalanced classification problems.
- Define class imbalance and understand why it presents a problem for classification models.
- Demonstrate the ability to undersample and oversample data with imbalanced classes.


WELCOME

Now that we've discussed some techniques to create classification models, we are ready to take the next step and apply those techniques to real-world problems.

Imbalanced Classes

One prominent problem in many classification tasks is **class imbalance**, which occurs when the training data you use to build your classification model is unevenly split.

Imbalanced Classes

Examples include:

- Fraud detection
- Churn prediction
- Medical diagnoses

Can you think of additional examples?

Before diving in, let's review some concepts from Day 1 of this unit, specifically, **confusion matrixes** and some metrics for evaluating models.

Binary Classification Prediction

There are four possible outcomes of a binary classification prediction.

True Positive	When we predict a class (positive) and are correct in that prediction. *For example, we predict someone has cancer, and they do.*
True Negative	When we predict a class (negative) and are correct in that prediction. *For example, we predict someone doesn't have cancer, and they don't.*
False Positive	When we predict a class (positive) and are incorrect in that prediction. *For example, we predict someone has cancer, but they don't.*
False Negative	When we predict a class (negative) and are incorrect in that prediction. *For example, we predict someone doesn't have cancer, but they do.*

Confusion Matrix

A **confusion matrix** is a way to tally and visualize the errors.

This is a special kind of table for the visualization and quantification of model performance.

	Predicted: No (0)	**Predicted: Yes (1)**
Actual=No (0)	True Negatives (TN)	False Positive (FP)
Actual=Yes (1)	False Negative (FN)	True Positives (TP)

Confusion Matrix

We can use our confusion matrix to calculate the model's overall **accuracy**.

- Accuracy is the proportion of correct calls.
- The calculation for Accuracy = (TP + TN)/(TP + TN + FP + FN).
- Treats FP and FNs equally—an issue for unbalanced data.

	Predicted: No (0)	**Predicted: Yes (1)**
Actual=No (0)	True Negatives (TN)	False Positive (FP)
Actual=Yes (1)	False Negative (FN)	True Positives (TP)

Confusion Matrix

We can use our confusion matrix to calculate the model's **precision**.

- Precision is the proportion of positive calls that were correct.
- The calculation for Precision = TP/(TP + FP), using the first column of the confusion matrix.
- A model with no FPs has perfect precision. All of its positive calls are correct!

	Predicted: No (0)	**Predicted: Yes (1)**
Actual=No (0)	True Negatives (TN)	False Positive (FP)
Actual=Yes (1)	False Negative (FN)	True Positives (TP)

If FPs are very undesirable, you want a model with high precision.

Confusion Matrix

We can use our confusion matrix to calculate the model's **recall**.

- Recall is the proportion of truly positive samples that were correct.
- The calculation for Recall = TP/(TP + FN), using the first row of the confusion matrix.
- Recall is a critical metric for optimizing a model with unbalanced data.
- A model with no FNs has perfect recall. All of the positive samples are correctly identified!
- Recall is sometimes called sensitivity.

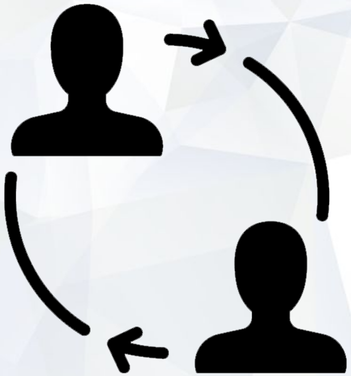
	Predicted: No (0)	**Predicted: Yes (1)**
Actual=No (0)	True Negatives (TN)	False Positive (FP)
Actual=Yes (1)	False Negative (FN)	True Positives (TP)

If FNs are very undesirable, you want a model with high recall.

Instructor Demonstration

Review Model Evaluation

Questions?



Activity: Hypothetical Models

In this activity, you will work in groups of two or three to discuss the relative importance of false positives and negatives.

You’ll also weigh the pros and cons of using each evaluation metric for a set of hypothetical classification models.

Suggested Time:

15 minutes

Time’s Up! Let’s Review.

Hypothetical Model 1 — Flagging SPAM Emails

If we define spam emails as positives, false positives are more costly than false negatives. (A spam email getting through is not the end of the world, but an important email that gets flagged as spam might be disastrous for the user.)

- We should review **precision** and specificity for this reason.
- Spam emails probably make up a relatively small (but not tiny) proportion of all emails.
- Because of this, a high accuracy or F1 score might be misleading.

	Predicted: No (0)	Predicted: Yes (1)
Actual=No (0)	True Negatives (TN)	False Positive (FP)
Actual=Yes (1)	False Negative (FN)	True Positives (TP)

Precision = TP/(TP + FP)

Hypothetical Model 2 — Targeting Who Applies for a Credit Card

Here, we should be weighting false positives and false negatives evenly, but true positives are likely to be small when compared to the amount of true negatives. (A lot of people may receive a new credit card application, but only a few people may sign up).

In this situation:

High accuracy may still be misleading, and we should examine all other evaluation metrics for different models to understand their relative strengths and weaknesses.

	Predicted: No (0)	**Predicted: Yes (1)**
Actual=No (0)	True Negatives (TN) 100	False Positive (FP) 50
Actual=Yes (1)	False Negative (FN) 2	True Positives (TP) **5**

$$Acc = (TP + TN)/(TP + TN + FP + FN)$$

Hypothetical Model 3 — Predicting Up or Down Stock Movement

There does not seem to be any obvious reason why false negatives or positives should be weighted more than the other. (On any given day, stocks are just about as likely to go up as they are down).

- Assuming a random, representative sample, we would expect the two classes to be roughly equal in size.
- Therefore, accuracy or the F1 score would likely be an effective summary metric to compare models.

	Predicted: No (0)	**Predicted: Yes (1)**
Actual=No (0)	True Negatives (TN)	False Positive (FP)
Actual=Yes (1)	False Negative (FN)	True Positives (TP)

$$F1 \text{ Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Hypothetical Model 4 — Predicting a Rainy Day

If we define rain as positive, false negatives are likely to be more costly than false positives.

- The cost of being without an umbrella in the rain is a lot higher than the cost of carrying one when it's not needed.
- This makes recall a metric of special interest because the classes are likely to be imbalanced, but not overwhelmingly so.
- The F1 score is probably a useful measure for comparing metrics.

	Predicted: No (0)	Predicted: Yes (1)
Actual=No (0)	True Negatives (TN)	False Positive (FP)
Actual=Yes (1)	False Negative (FN)	True Positives (TP)

$$\text{Recall} = TP/(TP + FN)$$

$$F1\ \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Hypothetical Model 5 — Which Startup Should a VC Invest In?

Venture capitalists (VCs) will probably view false negatives as more costly than false positives.

- VCs invest with the knowledge that the majority of companies will fail, and they get large returns from those that don't.
- Recall is likely to be the metric of most interest in this case.

	Predicted: No (0)	**Predicted: Yes (1)**
Actual=No (0)	True Negatives (TN)	False Positive (FP)
Actual=Yes (1)	$\boxed{\text{False Negative (FN)}}$	True Positives (TP)

Recall = TP/(TP + FN)

Questions?

Imbalanced Data

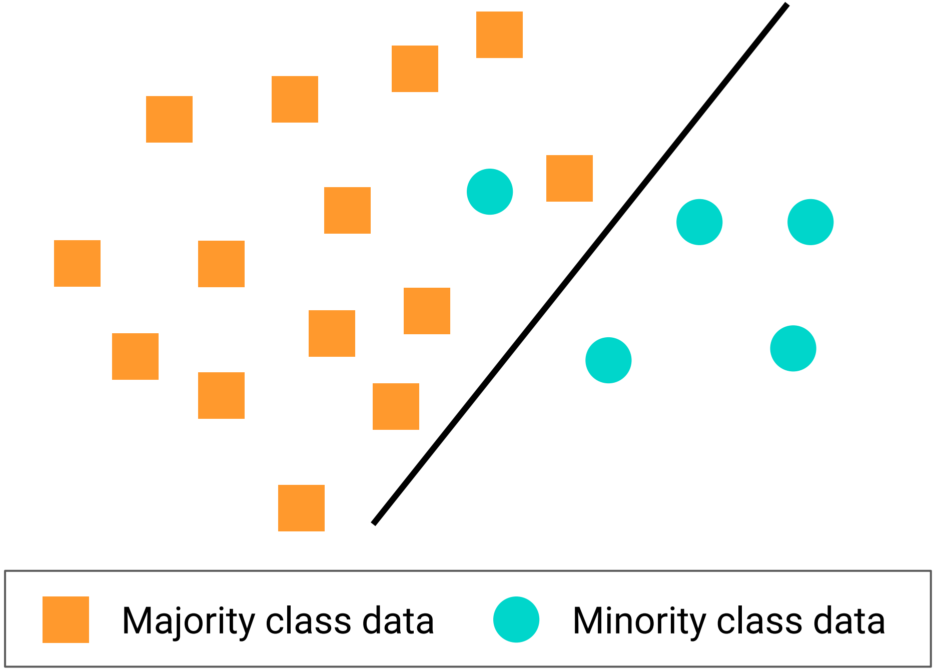
Imbalanced data describes cases when one or more classes in the data are much more or much less frequent than the other class(es).

Imbalanced Data

Imbalanced data is problematic because it can cause your model to be biased toward the majority class.

- Basically, the model will be better at predicting the majority class as compared to the minority class because model fitting algorithms are designed to minimize the number of **total** incorrect classifications.

- If data is imbalanced, accuracy scores can be a misleading indicator of model quality.

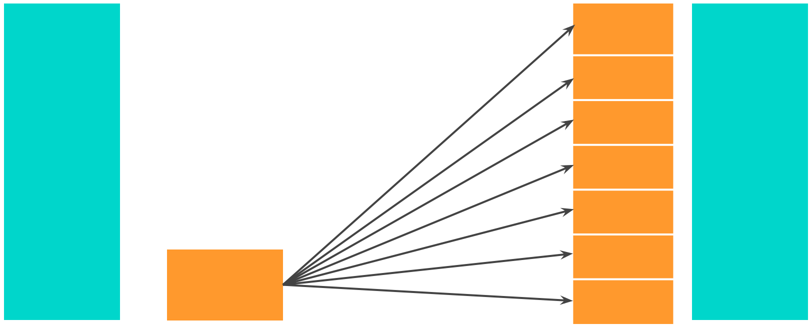


Imbalanced Data

The rest of the material will cover strategies for dealing with imbalanced classes.
We will work mostly with two methods:

Oversampling

We sample the **minority class** with greater-than-random chance.

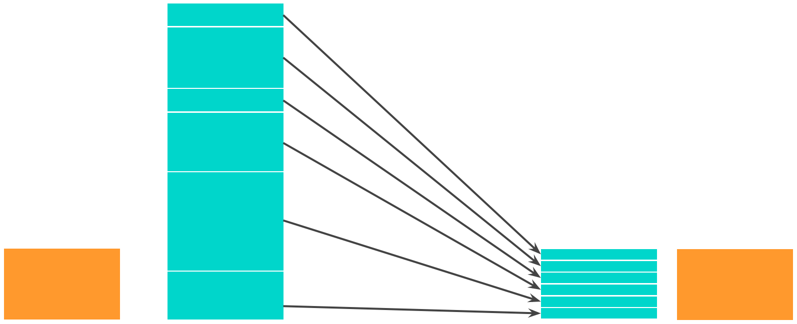


Original dataset

Final dataset

Undersampling

We sample the **majority class** with less-than-random chance.



Original dataset

Final dataset

Imbalanced Data

We will also:

Explain
why ensemble methods may be more suitable for imbalanced data than other classification methods.

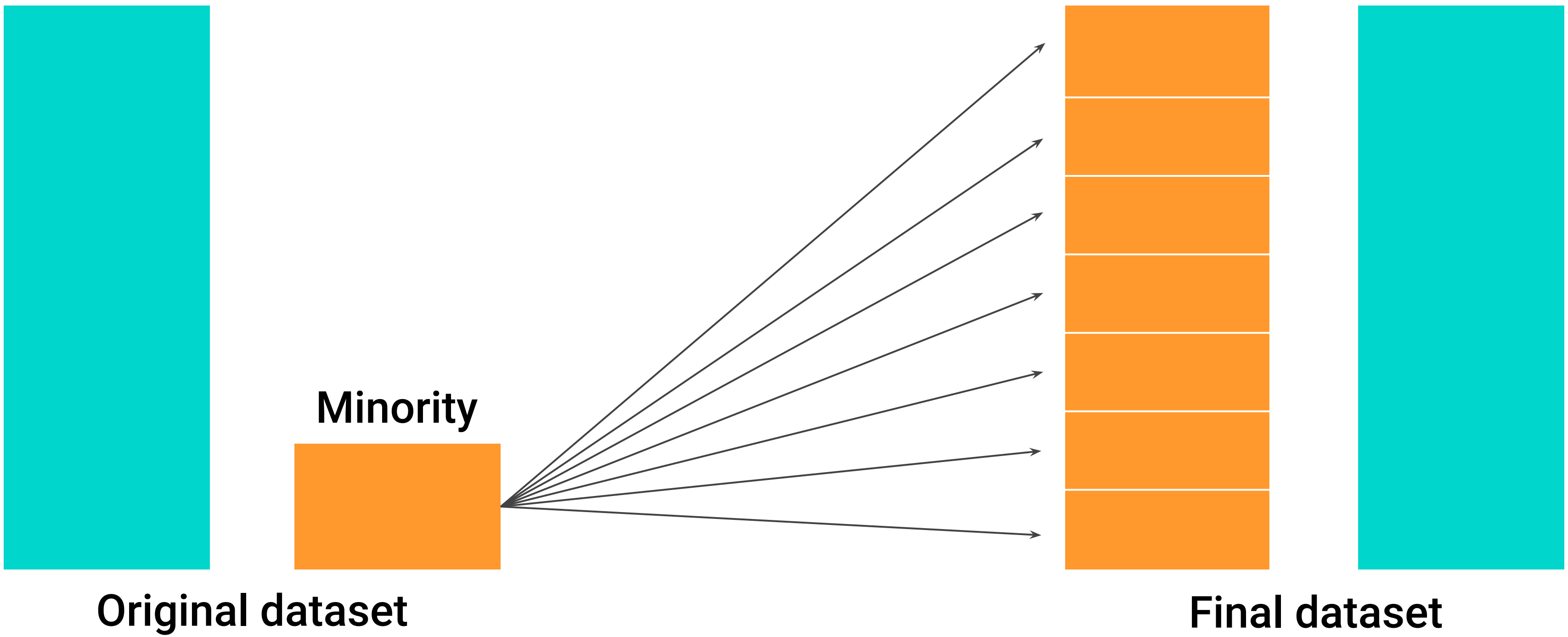
Introduce
a classification report specifically created for imbalanced data.

Random Sampling

Class imbalance refers to a situation in which the existing classes in a dataset aren’t equally represented.

Oversampling

Creating more instances of a class label, usually for the smaller class.



Oversampling

Potential strategies:

Add additional samples of the minority class until

---

instances of minority = instances of majority.

Random oversampling: Randomly choose minority class instances (with replacement).

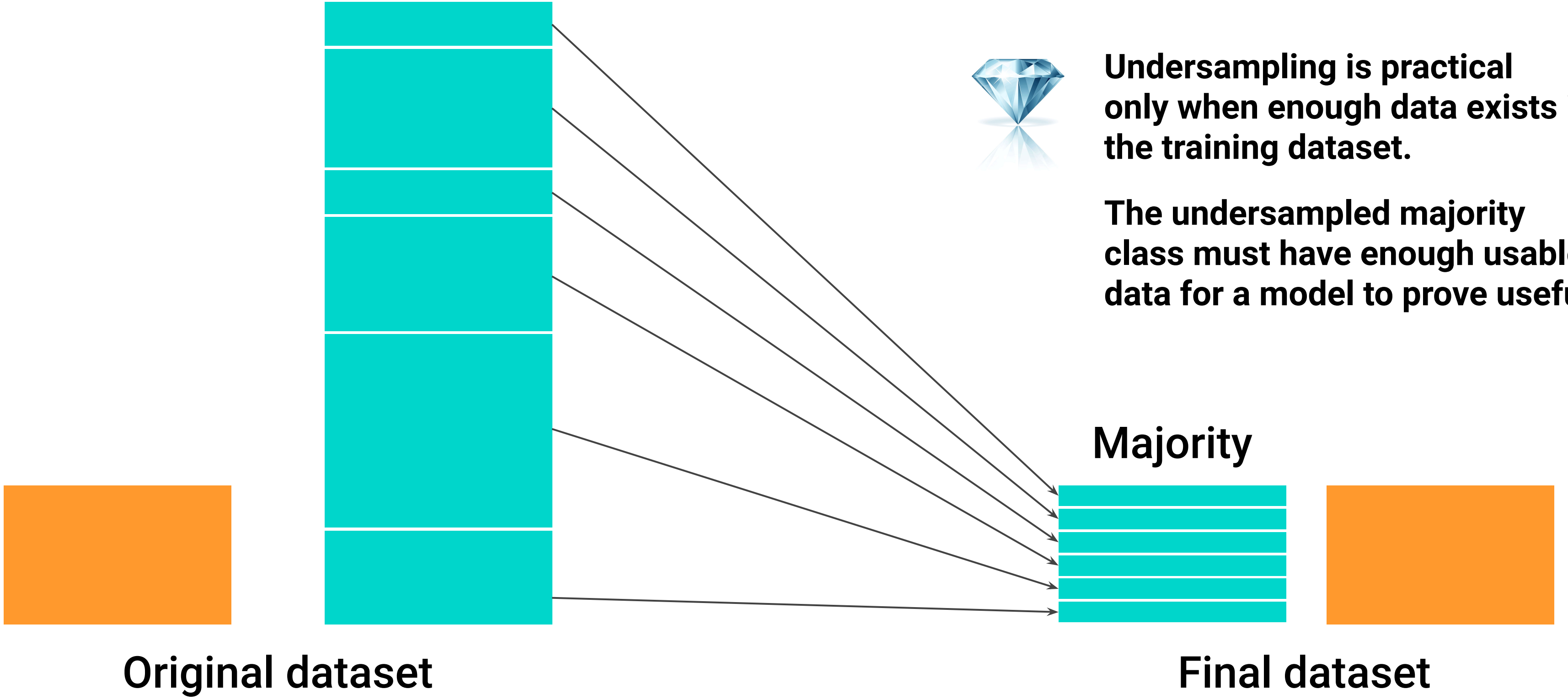
SMOTE (synthetic minority oversampling technique): Creates synthetic data from minority samples through k-nearest neighbors.

Undersampling

Creating fewer instances of a class label, usually for the larger class.

Undersampling is practical only when enough data exists in the training dataset.

The undersampled majority class must have enough usable data for a model to prove useful.



Undersampling

Potential strategies:

Remove instances of the majority class until

----

instances of minority = instances of majority.

Random undersampling: Randomly choose majority class instances to remove from the training set.

Cluster centroid: Undersampling: first create N clusters, where N is the number of minority class training instances; then take the centroids from those clusters as the majority class training data.

Random Sampling

Two methods that are commonly used to obtain new samples:

Random sampling

Our algorithm chooses **random instances** from the existing dataset.

We can use either oversampling or undersampling when sampling randomly, but we are using existing instances in our dataset and not creating new ones.

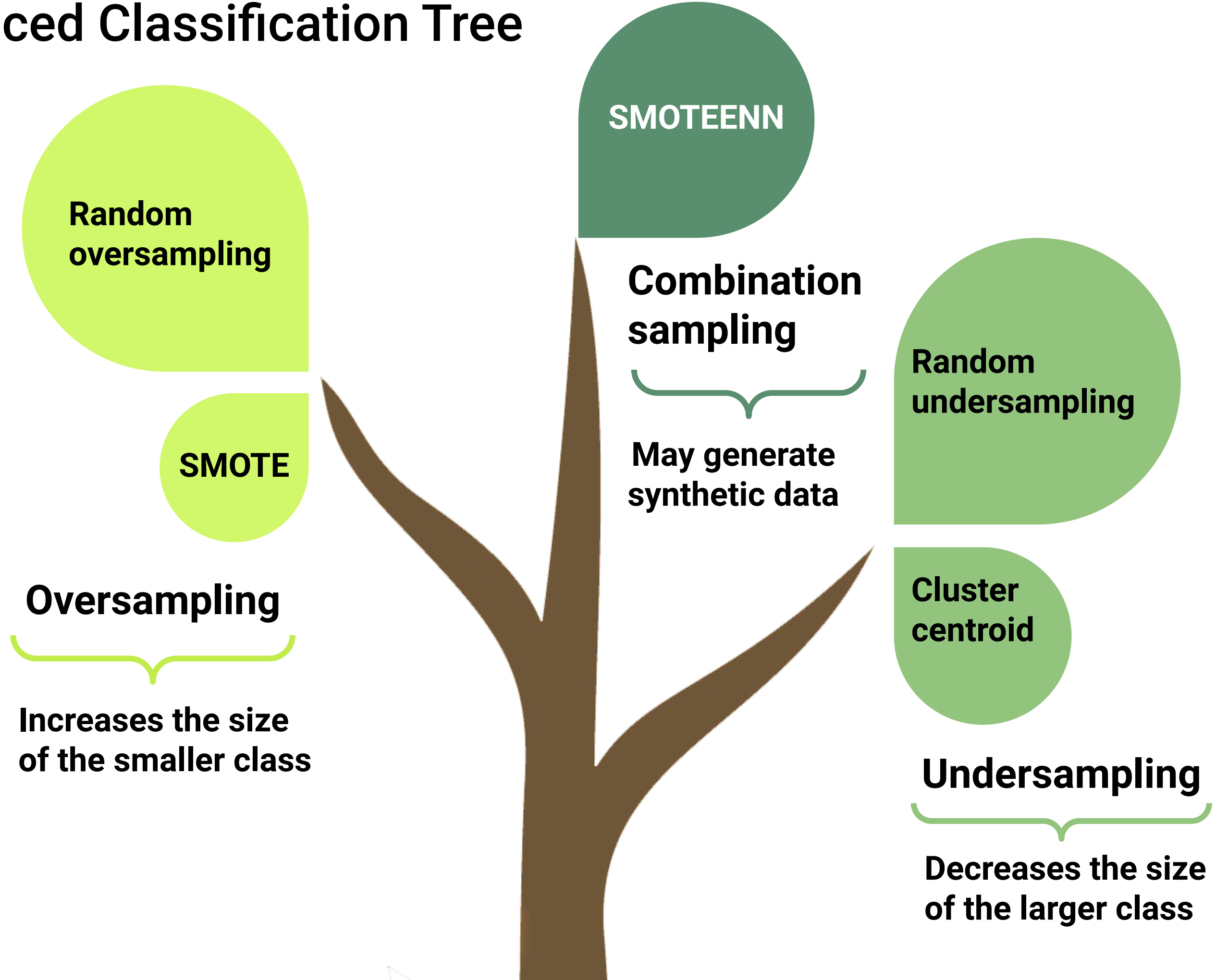
Synthetic sampling

Our algorithm generates **new instances** from observations about existing data.

In predicting loan defaults, we could use k-nearest neighbors to simulate the characteristics of a borrower who defaulted.

We would then add this simulated data to our original dataset.

The Imbalanced Classification Tree



Instructor Demonstration

Random Sampling

Questions?

Activity: Random Resampling

In this activity, you will use the provided dataset of a bank's telemarketing campaign to:

- Compare the effectiveness of random resampling methods using a random forest.
- Measure the random forest's recall of the minority class for both a random forest fitted to the resampled data and to the original dataset.

Suggested Time:

20 minutes

Time’s Up! Let’s Review.

Questions?

Countdown timer

15:00

(with alarm)

Instructor Demonstration

Synthetic Resampling

Questions?

Activity: Synthetic Resampling

In this activity, you will again use the provided dataset of a bank’s telemarketing campaign, but this time you'll compare the effectiveness of synthetic resampling methods using a random forest.

Suggested Time:

20 minutes

Time’s Up! Let’s Review.

What are the high-level steps in the process for making predictions with resampled data?

Synthetic Resampling Review

- Import the data.
- Separate the data into target and feature datasets.
- Encode the data.
- Split the data into training and testing datasets.
- Scale the data.
- Import the resampling technique (random oversampling, cluster centroids, SMOTE, etc.).
- Resample the X and y training sets to make the value counts between classes approximately equal.
- Instantiate the classifier model.
- Fit the model with the resampled data
- Generate the predictions.
- Review the results.

Questions?

Instructor Demonstration

Balanced Random Forest

Questions?

Activity: Comparing Imbalanced Classifiers

In this activity, you will:

- Apply the balanced random forest model that you just learned.
- Deploy a regular random forest and an additional imbalanced model of your choice.

Suggested Time:

20 minutes

Time’s Up! Let’s Review.

Questions?

The End