# PCSE 595
# Special Topics in Machine Learning

Dr. Sam Henry

samuel.henry@cnu.edu

Luter 325

# Office Hours

I expect to see everyone at office hours at some point

- Please stop by!
- This is an **Open Question Answering Time**

Monday, Wednesday, Friday

11:00-12:00, 1:00-1:30

**Or by appointment**

Office hours are in-person (just come by my office, LUTR 325)

# Pizza My Mind

- You can get extra credit
  - Up to two extra points on your final grade for one PCSE course
- Attend them! They’re fun, informative, and employers present
  - Don’t wait until you need a job or internship, go now!
- Thursdays at 12:20

… and you get free pizza!!

Students in PCSE classes can get extra credit if they attend at least 10 events. 10-11 events: 1 extra point; 12-13 events: 2 extra points.

# Data Balance

### Breast Cancer Dataset

- 458 benign and 241 Malignant

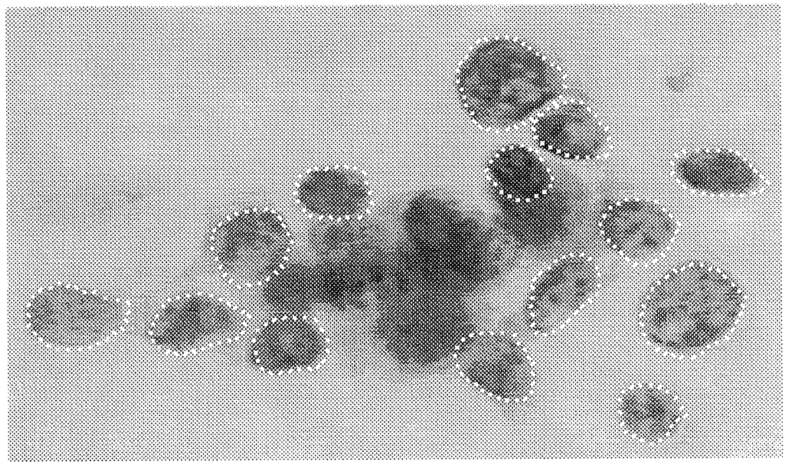
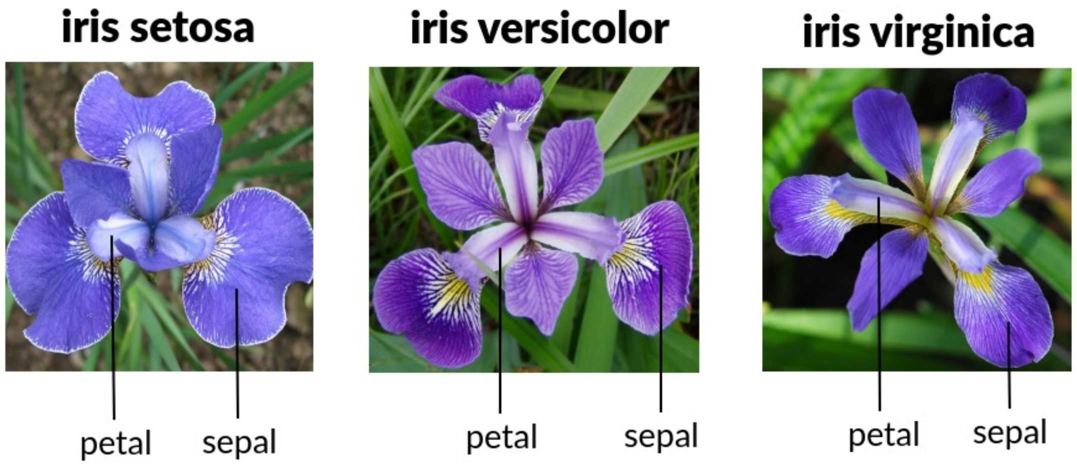


Figure 1: Initial Approximate Boundaries of Cell Nuclei

**Fairly Balanced Dataset**
**(~2 to 1 class balance)**

### Iris Dataset

- 50 of each flower type




**Perfectly Balanced Dataset**
**(1 to 1 to 1 balance)**

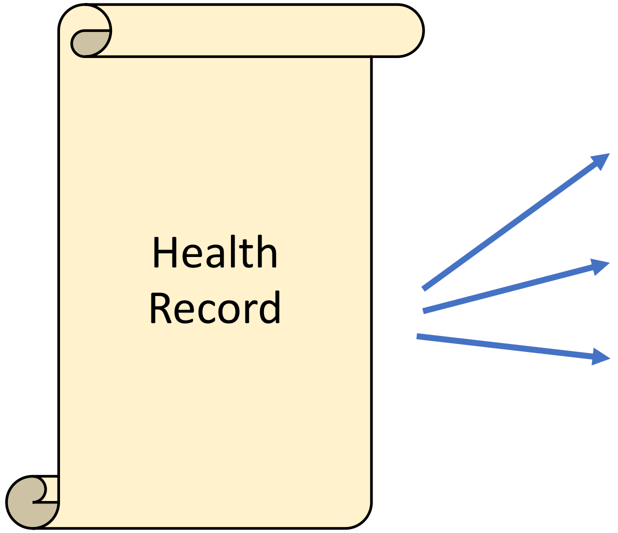
# Consider the Following Problem

- Goal – Relationship Extraction
  - Given an electronic health record, extract all of the following relationships

| Relationship | Definition | Example |
|---|---|---|
| **TrIP** | Treatment improves or cures medical problem | Her chest pain was controlled with morphine |
| **TrWP** | Treatment worsens or does not improve medical problem | … who presented with acute coronary syndrome refractory to medical treatment and TNK … |
| **TrCP** | Treatment caused medical problem | Narcotics can cause constipation |
| **TrAP** | Treatment administered for medical problem, but outcome is not mentioned in the sentence | HTN elev. chol. right facial droop with metal plate secondary to GSW to face right nephrectomy |
| **TrNAP** | Treatment is not administered or discontinued because of a medical problem | Discharge Instructions : may shower , no bathing or swimming for 1 month no creams , lotions or powders to any incisions no driving for 1 month no lifting > 10 # for 10 weeks |
| **TeRP** | Test reveals a medical problem | Cath at Kindred/North Shore today showed 80% LM lesion with normal LAD , CX , RCA . |
| **TeCP** | Test given to investigate a medical problem | He also has a diagnosis of interstitial cystitis and more significantly has a history of coronary artery disease with a recent catheterization , supraventricular tachycardia , and a progressive mitochondrial myopathy . |
| **PIP** | Medical problem indicates or reveals aspects of another medical problem | Monitor wounds for infection - redness , drainage , or increased pain |

# Relationship Extraction

- Goal – Relationship Extraction
  - Given an electronic health record, extract relationships




**TrIP** Her chest pain was controlled with morphine

**TrIP** Her acid reflux was controlled with pepto

**PIP** Monitor wounds for infection - redness , drainage , or increased pain

# Relationship Extraction as Sentence Classification

- Cast the problem as a sentence classification task
- For each sentence in the document, label it with each of the relations it contains

| | |
|---|---|
| **None** | Patient Admitted on 03/15/12 |
| **None** | Primary Complaints are chest pain and acid reflux |
| **TrIP** | Her chest pain was controlled with morphine |
| **TrIP** | Her acid reflux was controlled with pepto |
| **TrCP** | During treatment, she was wounded |
| **PIP** | Monitor wounds for infection - redness , drainage , or increased pain |

# Sentence Classification

- This is a Complicated Problem:

16,316 sentences

This is not binary classification – a sentence can contain 1 or more relationship types

| Relationship | TrIP | TrWP | TrCP | TrAP | TrNAP | TeRP | TeCP | PIP | Total Relations | No Relation |
|---|---|---|---|---|---|---|---|---|---|---|
| Total Count | 203 | 133 | 526 | 2617 | 174 | 3053 | 504 | 2203 | 9413 | 10,308 |
| Unique Sentence Count | 140 | **86** | 370 | 1643 | 119 | 1858 | 345 | 1447 | **6008** | **10,308** |

There is a huge class imbalance

$\frac{6008}{86} \sim 70\ to\ 1\ class\ ratio$

There are a lot of negative samples

If we consider "No Relation" a class,

then $\frac{10308}{86} \sim 120\ to\ 1\ class\ ratio$

# Complex Problems

- Problems with multiple classes require:
  - Different classification architectures
  - Different evaluation metrics
    - Metrics for each class, metrics to combine results for all classes
  - More complex error analysis

- Imbalanced Datasets require:
  - Different evaluation measures
  - May have trouble training due to class imbalance
  - Results may need to be interpreted more carefully

# Evaluation vs. Error Analysis

## Evaluation

- Methods to quantify the performance of an algorithm

## Error Analysis

- Methods to determine what needs to be improved to increase performance

# Evaluation Metrics for Classifiers

- Primary Metrics:
  - Accuracy
  - Precision
  - Recall
  - F-measure
  - Receiver Operator Characteristic (ROC) Curves
  - Precision and Recall Curves
  - Correlation Coefficients

…and more

# Why not just use accuracy?

**Accuracy = Total Correct / Size of the dataset**

- Probably the most popular measure
- Biased in favor of the majority class
  - assumes a 1 to 1 class distribution

***Use with Caution!***

90 samples of class 1
10 samples of class 2

Classify all as class 1

90/100 = 90% accuracy

defined more formally on subsequent slides

# Confusion Matrix and Types of Errors

**confusion matrix**

| | | Predicted Class | | |
|---|---|---|---|---|
| | | Class 1 (pos) | Class 2 (neg) | |
| True Class | Class 1 (pos) | TP | FN | P=TP+FN |
| | Class 2 (neg) | FP | TN | N=FP+TN |

Type II Error

Type I Error

**TP** = True Positive = predicted positive and is positive

**FP** = False Positive = predicted positive but is negative

**TN** = True Negative = predicted negative and is negative

**FN** = False Negative = predicted negative but is positive

**P** = Number Positive = TP + FN

**N** = Number Negative = FP+TN

# Accuracy

| | | | | |
|---|---|---|---|---|
| | | Predicted Class | | |
| | | Class 1 (pos) | Class 2 (neg) | |
| True Class | Class 1 (pos) | TP | FN | P=TP+FN |
| | Class 2 (neg) | FP | TN | N=FP+TN |

$$Accuracy = \frac{TP + TN}{P + N}$$

Measures the percent of predictions that were correct
Places an equal importance on positive and negative classes
Appropriate for balanced datasets
Range: (0,1); 1 is best

# Classes aren’t always the same

Accuracy assumes classes are balanced and are equally important

- In many cases, one class is more important than another
  - Fraud detection, cancer diagnosis, intrusion detection, relationship extraction
  - In these cases, we may tolerate greater overall error, in return for better predictions of the more important class
- Often the classes are imbalanced
  - In these cases, the performance on the minority class is hidden by the size of the majority class

# Precision

| | | Predicted Class | | |
|---|---|---|---|---|
| | | Class 1 (pos) | Class 2 (neg) | |
| True Class | Class 1 (pos) | TP | FN | P=TP+FN |
| | Class 2 (neg) | FP | TN | N=FP+TN |

$$Precision = \frac{TP}{TP + FP}$$

How good is my performance on the samples I am predicting to be true?

What percent of samples that I predict are true are actually true?

Range: (0,1); 1 is best

# Recall

| | | Predicted Class | | |
|---|---|---|---|---|
| | | Class 1 (pos) | Class 2 (neg) | |
| True Class | Class 1 (pos) | TP | FN | P=TP+FN |
| | Class 2 (neg) | FP | TN | N=FP+TN |

$$Recall = \frac{TP}{TP + FN}$$

What percentage of true samples am I predicting to be true?

Am I missing a lot of true samples?

Range: (0,1); 1 is best

Worksheet

# Evaluation Example

I took an aspirin for a headache and neckpain

You predict:
aspirin, for
aspirin, and
aspirin, headache

You Miss:
Aspirin, neckpain
headache, aspirin
neckpain, aspirin

There is a possible relationships between each word and every other word in the sentence

**Calculate the confusion matrix, Accuracy, Precision, and Recall**
**What is the class ratio?**

# Precision and Recall Tradeoff

There is a trade-off between precision and recall

There is typically a threshold you can adjust adjust the trade-off

High Recall = Fishing with a big net

High Precision = Spear fishing

- As you increase the size of your net, you catch more fish
- You'll catch a lot of of what you wanted, but you'll get increasingly more bycatch

# F-Measure = $F_1$ Measure = F Score

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

"Harmonic mean" between precision and recall

A measure that balances the trade-off between precision and recall

Range: (0,1); 1 is best

This allows us to quantify precision and recall performance with a single number

Bonus Material

# $F_\beta$ Measure

$$F_\beta = (1+\beta^2) * \frac{precision * recall}{\beta^2 * precision + recall}$$

Generalized F-Measure

You can weight the importance of precision and recall in your metric by changing $\beta$

$\beta = 2$ weights recall more

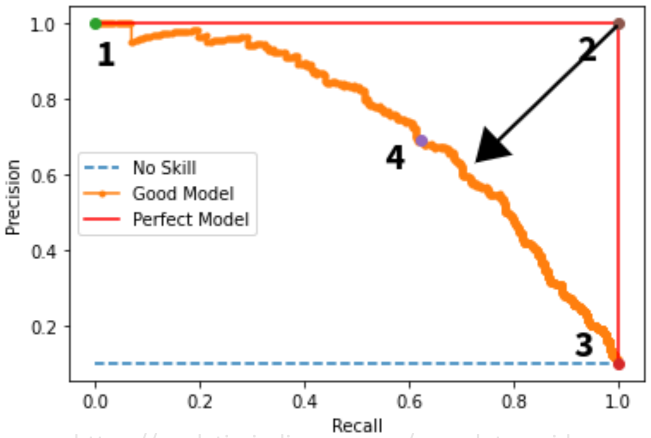
$\beta = 0.5$ weights precision more

Good to know, but F1-Measure is most commonly used

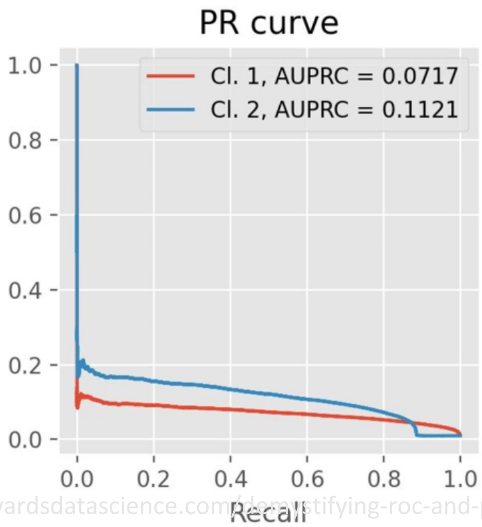
# Precision and Recall (PR) Curve

- Plots the trade-off between precision and recall
- **Area under the Curve (AUC)** quantifies performance with a single number
  - Higher AUC means better, more robust system (less trade-off between precision and recall)
- Generate by varying some threshold with some interval
  - For example, threshold of a logistic function output
- The optimum threshold is the Euclidean distance from the perfect model
  - E.g. Euclidean distance between (1,1), and (recall, precision) of the threshold

- The look of PR Curves can vary a lot
- There isn't a 1-to-1 relationship between precision and recall, and the plot can be jagged.
- Plotting a majority class baseline is useful for interpretation
  - It will be a line at the positive class percentage




https://analyticsindiamag.com/complete-guide-to-understanding-precision-and-recall-curves/




https://towardsdatascience.com/demystifying-roc-and-precision-recall-curves-d30f3fad2cbf

# Receiver Operating Characteristic (ROC) Curve

- Plots the trade off between:
  - True Positive Rate (Recall) and False Positive Rate
    - Equivalent to Sensitivity vs. 1-Specificity
    - Lots of names for the same measures:
      - see: https://en.wikipedia.org/wiki/Precision_and_recall
- Area Under the ROC Curve (AUROC) can be used to quantify performance with a single number
  - Higher AUROC means better system (less trade-off between TP rate and FP rate)
- Generate

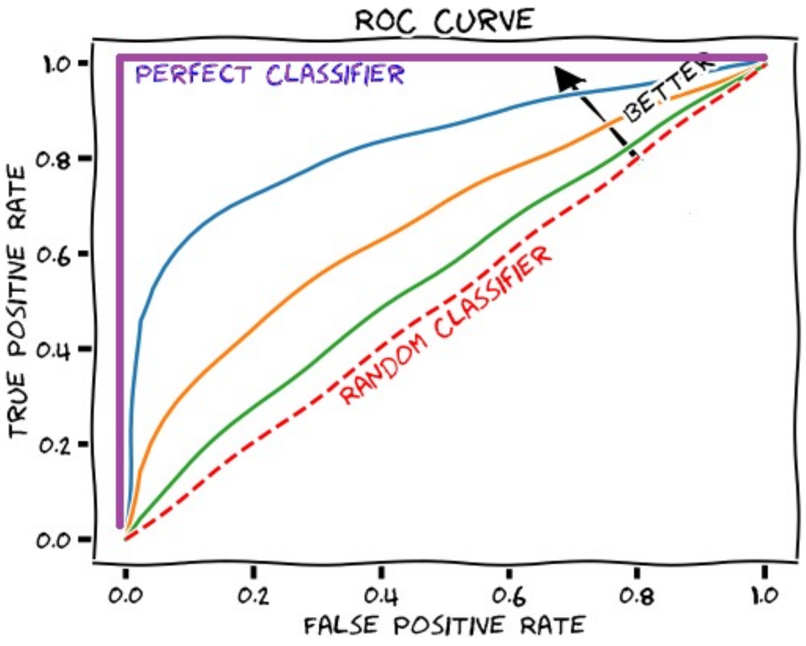



Image from: https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

**Use with Caution:**

research* shows that **ROC Curves are misleading (overly optimistic) for problems with a high class imbalance**. For these problems, precision and recall curves are preferred

*"The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets" by Saito and Rehmsmeier https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0118432

# Multiclass/label Evaluation

**Confusion Matrix for Multiple Classes**

| | | Predicted Class | | |
|---|---|---|---|---|
| | | Class 1 | Class 2 | Class 3 |
| True Class | Class 1 | 4 | 50 | 1 |
| | Class 2 | 10 | 290 | 65 |
| | Class 3 | 0 | 10 | 2 |

- Show the count of samples classified as each class
- The example table tells us that there is:
  - There is confusion between classes 1 and 2
  - Most of class 3 is being classified as class 2

# Multiclass/label Evaluation

- For Multiclass classification and Multilabel classification evaluation:
  - Report evaluation metrics for each class/label individually
  - Report Micro and/or Macro averaging between all classes
- Macro Averaged Precision, Recall, F1
  - All classes contribute equally
  - **Equally weights the performance of each class**
  - Tells us the average performance over all classes
- Micro Averaged Precision, Recall, F1
  - Classes with large number of samples dominate
  - **Equally weights the performance of each sample**
  - Tells us the performance over all samples

# Macro Averaging

...It's just the average of the metric for each class

- Macro Averaged Accuracy

$$\frac{1}{C}\sum_{i}^{C} Accuracy_i$$

- Macro Averaged Recall

$$\frac{1}{C}\sum_{i}^{C} Recall_i$$

- Macro Averaged Precision

$$\frac{1}{C}\sum_{i}^{C} Precision_i$$

- Macro Averaged F1

$$\frac{1}{C}\sum_{i}^{C} F1_i$$

Where C = the number of classes

# Micro Averaging

- Micro Averaged Accuracy

$$\frac{\sum_{i=1}^{C} TP_i}{Total\ Number\ of\ Predictions}$$

- Micro Averaged Precision

$$\frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FP_i}$$

Where $C$ = the number of classes

- Micro Averaged Recall

$$\frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FN_i}$$

- Micro Averaged F1

$$2 * \frac{\sum_{i=1}^{C} precision_i * recall_i}{\sum_{i=1}^{C} precision_i + recall_i}$$

…This looks really complicated, but we are just considering each sample regardless of class, determining if it’s a TP, FP, TN, or FN and then calculating as each metric as before

# Multiclass Evaluation

| | | Predicted Class | | |
|---|---|---|---|---|
| | | Class 1 | Class 2 | Class 3 |
| True Class | Class 1 | 4 | 50 | 1 |
| | Class 2 | 10 | 290 | 65 |
| | Class 3 | 0 | 10 | 2 |

- **Calculate Class 1,2,3 precisions, the micro, and macro precisions**
- **Calculate Class 1 Recall**
- **Calculate Micro Recall**
- **Calculate Micro Accuracy**

Precision $= \frac{TP}{TP+FP}$ Recall = $\frac{TP}{TP+FN}$

- Micro Averaged Precision

$$\frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FP_i}$$

- Macro Averaged Precision

$$\frac{1}{C}\sum_{i}^{C} Precision_i$$

- Micro Averaged Accuracy

$$\frac{\sum_{i=1}^{C} TP_i}{Total\ Number\ of\ Predictions}$$

Where $C$ = the number of classes

# Multilabel Evaluation

- For multiclass problems (when each sample belongs to exactly 1 class), micro precision = micro recall = micro accuracy
- This is **not** the case for multi-label problems
  - One sample may have multiple labels, and a sample may have NO labels
- Consider the following True ($Y$) and Predicted ($\hat{Y}$)labels

$$Y = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \qquad \hat{Y} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

**Construct the confusion matrix**

**Calculate micro precision, recall, f1, accuracy**

Including a "None" in the confusion matrix is useful, but it isn't counted towards the true positives

# Other Evaluation Measures

- There are a lot of evaluation measures
- Common Regression Problem evaluation measures:
  - Mean Squared Error (MSE)
  - Root-Mean-Squared-Error (RMSE)
  - Mean-Absolute-Error (MAE)
  - $R^2$ = Coefficient of Determination

- Common Ranking Algorithm Measures
  - Precision at K
  - Mean average precision

  - Example of ranking systems are information retrieval and recommendation systems

# Compare Against Baseline Systems

- Majority class baseline for classification
  - Classify everything as the majority class
- Mean baseline for regression
  - Output the mean of the data for every sample

- Can you think of another, problem specific simple classifier?
  - Compare against that
  - Sometimes a simple solution is frustratingly hard to beat

# Recap:

- Different problem types:
  - Regression, binary classification, multiclass classification, multilabel classification
- How to construct and interpret a confusion matrix
  - This can help with error analysis
  - The error types can be used to derive new evaluation metrics
- How to evaluate the performance of a model using a variety of metrics
  - accuracy, F1, precision, recall, ROC curves
  - macro vs. micro measures
- We should always compare against:
  - A majority class baseline for classification, mean baseline for regression
  - and if possible against some other a simple baseline (which you implement)

# Discussion

| Class | Precision | Recall | F1 |
|---|---|---|---|
| Micro | 0.80 | 0.81 | 0.81 |
| Macro | 0.78 | 0.66 | 0.69 |
| TrIP | 0.70 | 0.56 | 0.62 |
| TrWP | 1.00 | 0.30 | 0.46 |
| TrCP | 0.75 | 0.79 | 0.77 |
| TrAP | 0.81 | 0.91 | 0.86 |
| TrNAP | 0.68 | 0.55 | 0.61 |
| TeRP | 0.86 | 0.88 | 0.87 |
| TeCP | 0.68 | 0.54 | 0.60 |
| PIP | 0.77 | 0.76 | 0.77 |

| Relationship | TrIP | TrWP | TrCP | TrAP | TrNAP | TeRP | TeCP | PIP | Total Relations | No Relation |
|---|---|---|---|---|---|---|---|---|---|---|
| Total Count | 203 | 133 | 526 | 2617 | 174 | 3053 | 504 | 2203 | 9413 | 10,308 |
| Unique Sentence Count | 140 | 86 | 370 | 1643 | 119 | 1858 | 345 | 1447 | 6008 | 10,308 |

# Dealing with Imbalanced Data

- Learning on severely imbalanced datasets can be difficult
  - Classifiers can learn to just predict everything as the majority class
- This can be hard to overcome, however there are a few simple methods:
  1. Add class weights
  2. Artificially balance the dataset
     - Under-sample majority class
     - Over-sample minority class
     - Generate synthetic data for the minority class (SMOTE)

# Option 1: Add Class Weights

- We can add class weights to make samples of different classes contribute more or less to loss

Scale the loss by the class weight

$$Training\_Loss = J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w_{c,i}} * loss(\hat{y}_i, y_i)$$

where $w_{c,i}$ is the class weight of the class of class i

- By **default**, the class weight balances the contribution of each class

$$\boldsymbol{w_c} = 1 - \frac{count\ of\ samples\ of\ class\ c}{total\ samples\ in\ the\ dataset}$$

You don’t have to use the default. You can select any class weights you want. Determining optimum class weights is a hyperparameter

# Class Weights Example

| | | Predicted Class | | |
|---|---|---|---|---|
| | | Class 1 | Class 2 | Class 3 |
| True Class | Class 1 | 4 | 50 | 1 |
| | Class 2 | 10 | 290 | 65 |
| | Class 3 | 0 | 10 | 2 |

$$\boldsymbol{w_c} = 1 - \frac{count\ of\ samples\ of\ class\ c}{total\ samples\ in\ the\ dataset}$$

55 samples of class 1
365 samples of class 2
12 samples of class 3

432 Samples total

$$Training\_Loss = J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} w_{c,i} * loss(\hat{y}_i, y_i)$$

where $w_{c,i}$ is the class weight of the class of class i

$$w_1 = 1 - \frac{55}{432} = 1 - 0.127 = 0.873$$

$$w_2 = 1 - \frac{365}{432} = 1 - 0.845 = 0.155$$

$$w_3 = 1 - \frac{12}{432} = 1 - 0.028 = 0.972$$

So, each time we incorrectly classify a sample of class 3, it adds about 6 times more loss than missing a sample from class 2

This changes the loss surface, and simply classifying everything as class 2 may no longer be a good option (therefore removing lots of local minimum)

# Option 2: Artificially Balance the Dataset

1. **Under-sample** the majority class
   - Randomly select a fixed number of samples from each class such that the resulting dataset is balanced
   - Problem: you are removing data which may be informative to the system
2. **Over-sample** the minority class
   - Randomly repeat a fixed number of samples from each minority class such that the resulting dataset is balanced
   - Problem: you are repeating data which makes those points artificially more important. What if the points are outliers?

$$Training\_Loss = J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} loss(\hat{y}_i, y_i)$$

Each of these methods make it so that each class contributes equally to loss

# Option 2: Artificially Balance the Dataset

3. **Generate synthetic data** for the minority class
    - This is typically a better option than over-sampling, but could be problematic if your synthetic data is not representative of your real data.
    - A popular data generation method is "Synthetic Minority Oversampling Technique" (SMOTE)
    - The authors recommend a combination of under-sampling the majority class and using SMOTE to over-sample the minority class. Stating that it outperforms doing only under-sampling or doing only over-sampling.

SMOTE: Synthetic Minority Over-sampling Technique by Chawla, et al., 2002 https://arxiv.org/pdf/1106.1813.pdf

# “Synthetic Minority Oversampling Technique” (SMOTE)

- SMOTE works by:
    1. For a single point, find the k-nearest neighbors and randomly select one
    2. For each feature:
        1. Find the difference between the sample and the its neighbor
        2. Randomly generate a number between 0 and 1 and multiply it by the difference
        3. Add the scaled difference to the feature value of that original point
- This effectively creates new feature values somewhere on a line between the point and its nearest neighbor.
- “This approach effectively forces the decision region of the minority class to become more general.”

SMOTE: Synthetic Minority Over-sampling Technique by Chawla, et al., 2002 https://arxiv.org/pdf/1106.1813.pdf

# Notes on Class Weights and Artificially Balancing the Dataset

- It is best, if possible to learn with the original data distribution
- **Potential Drawbacks:**
  - Class weights/oversampling can give excessive importance to outliers
    - So, for severely imbalanced datasets, you may not want to reweight/sample so that all classes are equally weighted in the loss calculation
  - Under-sampling can eliminate import datapoints
    - Particularly if the samples for a class are already sparse.
  - When you modify the dataset, you learning something that is not representative of the real world.
    - Weighting or balancing the dataset biases the system
      - If you are unsure of a sample, maybe you should classify it as the majority class rather than giving equal “weight”

# Implementation Detail: Stratified Sampling

- When you have severely imbalanced data, it is important to use stratified sampling for train/validation/test splits and cross-validation
- Stratified sampling ensures a similar class distribution per split
  - E.g. if the dataset is 95% negative, and 5% positive, then each fold of cross-validation will have a 95%-5% class balance
  - This is done by sampling each class independently
- Stratified sampling is more complex for multi-label problems, and the exact solution is unclear

Discuss

(I believe there is no implementation of this in scikit-learn)

# Section Summary

We learned how to deal with complex problems

1. How to evaluate performance for problems with class imbalances
   - Precision, recall, F1
2. How to evaluate performance for multiclass and multilabel problems
   - Macro vs. micro performance
3. Practical considerations for problems with severe class imbalance