CMPE 297, INSTRUCTOR: JORJETA JETCHEVA

MODEL EVALUATION

ENSURING GENERALIZABILITY

ABILITY TO GENERALIZE

- Supervised machine learning algorithms need to be able to generalize from examples they have seen to examples they haven't seen
- To simulate a model's ability to generalize, it is typically trained on a training set & tested on a test set
- There is an assumption that the training data & test set data come from the same underlying probability distribution

TRAINING VS. TEST DATA – REVIEW FROM PREVIOUS LECTURE

Usually we split the data into training, dev(elopment/validation) and test sets

- Training set is used to train the model
- Dev set is used to evaluate candidate models & model parameters, and choose the best one
- Test set is only used to test the model once the algorithm has been fine-tuned

Object ID	Attribute 1	…	Attribute N	Target Var
1		…		
2		…		
3		…		
4		…		
5		…		
6		…		
7		…		
8		…		
9		…		
10		…		

Note that we have ground truth (labels for all of the data samples)

CROSS VALIDATION

- Cross validation is used to arrive at a more reliable estimate of model performance
  - K-Fold Cross Validation: Divide data into k folds, and use one as dev set and the rest for training

Experiment 1, Accuracy = 0.9

	Object ID	Attr. 1	…	Attr. N	Target Var
Fold 1	1		…		
	2		…		
Fold 2	3		…		
	4		…		
Fold 3	5		…		
	6		…		
Fold 4	7		…		
	8		…		
Test Data	9		…		
	10		…		

Experiment 2, Accuracy = 0.96

Object ID	Attr. 1	…	Attr. N	Target Var
1		…		
2		…		
3		…		
4		…		
5		…		
6		…		
7		…		
8		…		
9		…		
10		…		

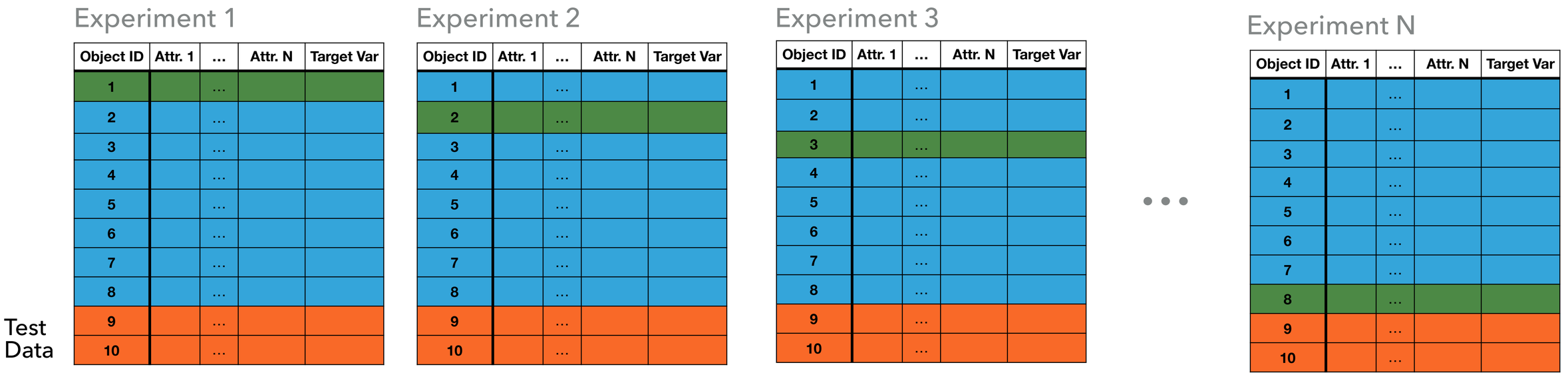
Experiment 3, Accuracy = 0.94

Object ID	Attr. 1	…	Attr. N	Target Var
1		…		
2		…		
3		…		
4		…		
5		…		
6		…		
7		…		
8		…		
9		…		
10		…		

Overall Accuracy is reported as the average of the accuracy of Experiments 1-3: 0.93 = (0.9+0.96+0.94)/3.

CROSS VALIDATION (CONT.)

- Leave-1-out cross validation iterates over each data point using it as a dev/validation set with the rest of the dataset being the training set
  - Useful when we don't have enough data – each training example can serve as both training & validation input just not at the same time



Overall Accuracy is reported as the average of the accuracy of Experiments 1-N (we have assumed there are N training examples)

IMPACT OF CLASS IMBALANCE ON PERFORMANCE METRICS

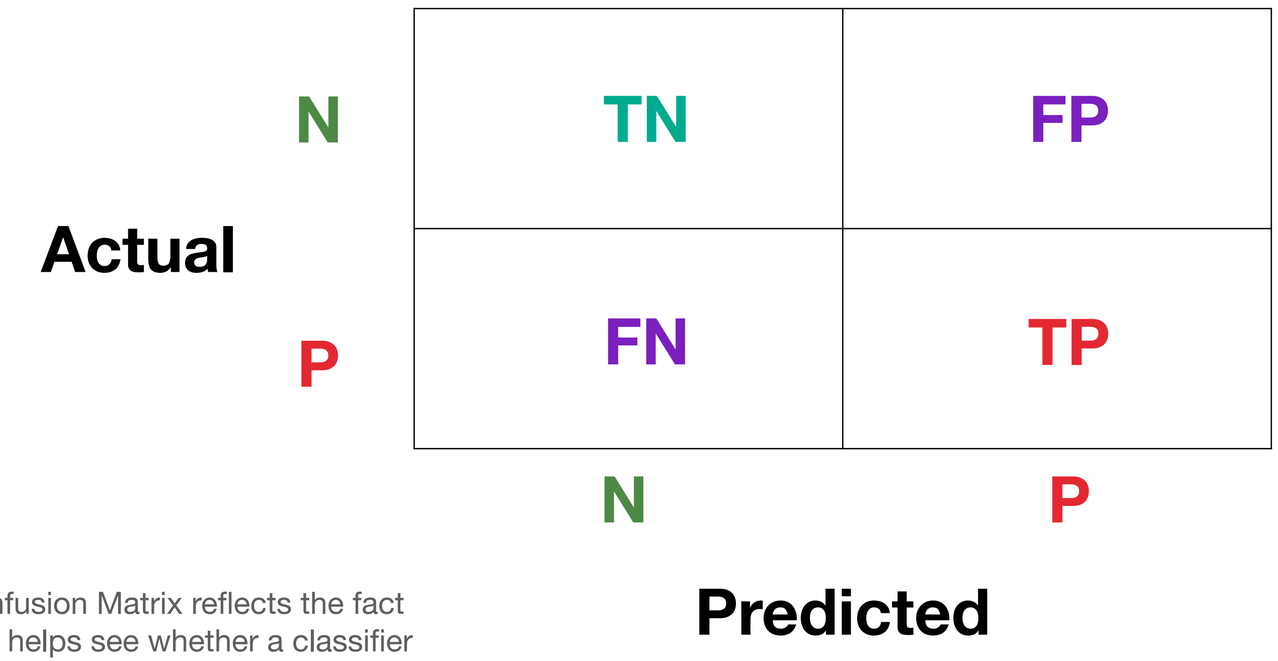
Classification Example

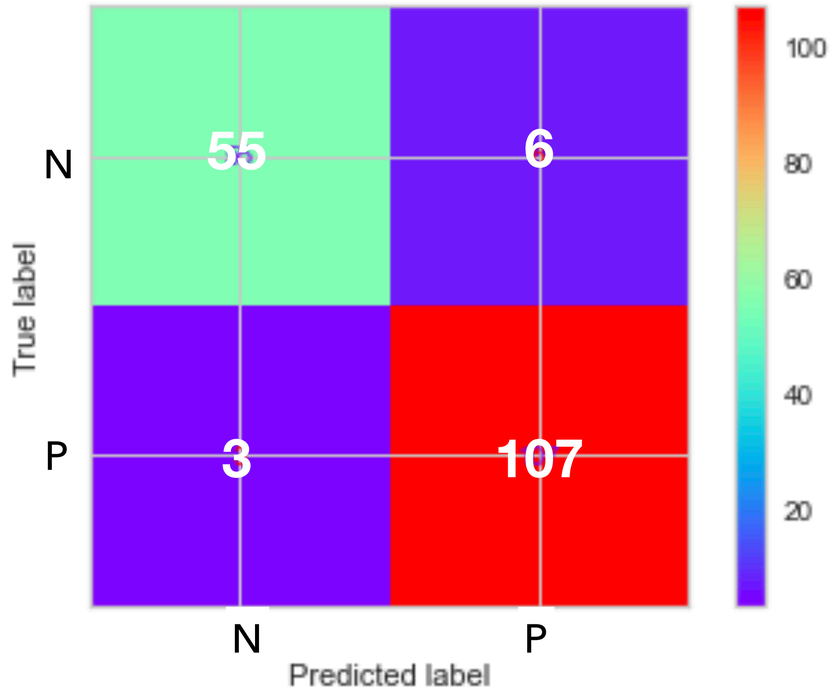
- Let's assume that we have a dataset which has a large number of instances of Class A (e.g., 95%), and a very small number of instances of Class B (e.g., 5%)
- And that our classifier learns to predict the majority class in all cases
  - Assume test set of 100 samples, 95 of them Class A, and 5 of them in Class B
  - Our classifier would predict Class A for all samples and we will get a 95% accuracy result
- Classifier appears to have good performance but in fact never predicts Class B
  - What is Class B are the records of patients that have cancer?

Object	Correct Label	Predicted Label
1	A	A
...	A	A
93	A	A
94	A	A
95	A	A
96	B	A
97	B	A
98	B	A
99	B	A
100	B	A

CONFUSION MATRIX FOR BINARY CLASSIFICATION

- We assume a binary classification problem and we refer to one class as the Positive class P, and to other as the Negative class N.
- The confusion matrix for binary classification is structured as follows:





The name Confusion Matrix reflects the fact that this matrix helps see whether a classifier often confuses one class with the other.

CONFUSION MATRIX EXERCISES

Object	Attribute 1	…	Attribute N	Ground Truth (Target) Label	Predicted Label
1	…	…	…	A	A
2	…	…	…	A	A
3	…	…	…	A	B
4	…	…	…	A	A
5	…	…	…	A	A
6	…	…	…	B	A
7	…	…	…	B	B
8	…	…	…	B	A
9	…	…	…	B	A
10	…	…	…	B	B

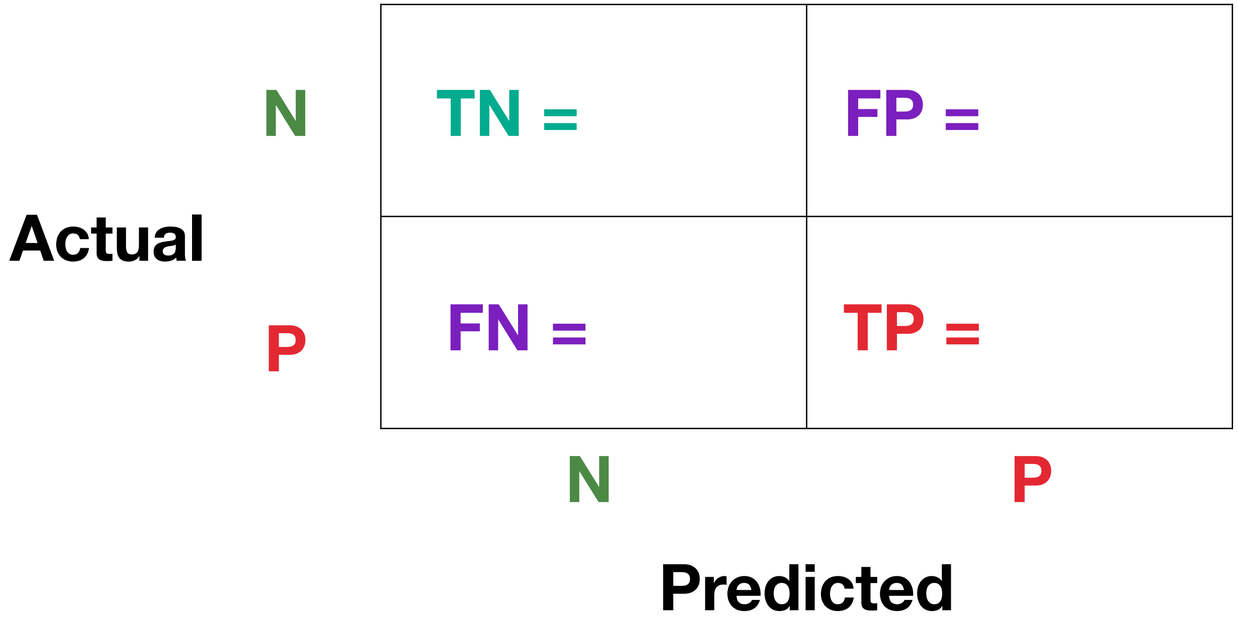
Let's call A the positive class, and B, the negative class

TP =

TN =

FP =

FN =



CLASSIFICATION ACCURACY & CLASSIFICATION ERROR

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}$$

Range: 0-1

Actual \ Predicted	N	P
N	TN	FP
P	FN	TP

$$Classification\ Error = 1 - Accuracy$$

Range: 0-1

$$Classification\ Error = \frac{FP + FN}{TN + TP + FN + FP}$$

PRECISION

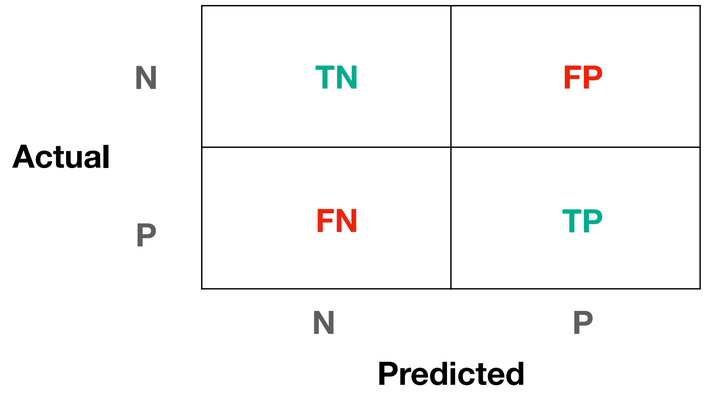
What fraction of the instances classified as positive are actually positive?

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

Improve precision by reducing FP

Reducing the number of false positives improves precision.

Example: In a user facing application, precision is very important since users have very low patience for false recommendations, poor search results, etc.



RECALL

Fraction of positive instances the classifier identifies correctly as positive is the True Positive Rate or Recall

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

Total number of positive instances

Improve recall by reducing FN

Example: In a health application, recall is more important than precision, e.g., the ability of the classifier to correctly identify that a patient has cancer is more important than having high overall accuracy or precision.

Actual \ Predicted	N	P
N	TN	FP
P	FN	TP

Recall is also sometimes referred to as sensitivity or probability of detection

PRECISION-RECALL TRADE OFF

- Often if you increase precision, you reduce recall because reducing FP tends to increase FN
  - You reduce FP by being more conservative about labeling an instance as positive, which means you are more liberal with labeling it as negative

- Alternatively, if you want to improve recall, e.g., if you want to improve cancer detection, you may decide that even slightly suspicious cases are positive (and reduce FN but increase FP)

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

F1 SCORE

- F1 score combines precision & recall into a single number
  - F1 goes up when either precision or recall goes up

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

F1 is the harmonic mean of precision & recall

Example:

	precision	recall	f1-score	support
not 1	0.96	0.98	0.97	407
1	0.79	0.60	0.68	43
avg / total	0.94	0.95	0.94	450

Single metric makes it easier to analyze model performance

F-SCORE

More generally, we have F-score

- To weigh precision more heavily, use $\beta < 1$
  - FP affects F-score more than FN
  - Improvements in FP, improve the F-score more
- To weigh recall more heavily, use $\beta > 1$
  - FN affects F-score more than FP
  - Improvements in FN, improve the F-score more

$$F_\beta = (1+\beta^2) \cdot \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}$$

$$= \frac{(1+\beta^2) \cdot TP}{(1+\beta^2) \cdot TP + \beta \cdot FN + FP}$$

MULTI-CLASS CONFUSION MATRIX

Object ID	Attribute 1	...	Attribute N	Class Label	Predicted Label
1	...	...	...	Red	Red
2	...	...	...	Blue	Red
3	...	...	...	Green	Blue
4	...	...	...	Green	Green
5	...	...	...	Blue	Blue
6	...	...	...	Blue	Green
7	...	...	...	Red	Red
8	...	...	...	Green	Green
9	...	...	...	Yellow	Yellow
10	...	...	...	Yellow	Yellow
11	...	...	...	Yellow	Yellow
12	...	...	...	Red	Red
13	...	...	...	Blue	Red
14	...	...	...	Green	Blue
15	...	...	...	Green	Green
16	...	...	...	Blue	Blue
17	...	...	...	Red	Red
18	...	...	...	Green	Green
19	...	...	...	Yellow	Yellow
20	...	...	...	Green	Yellow

Complete the multi-class confusion matrix for the dataset shown on the left.

ACTUAL \ PREDICTED	Red	Blue	Green	Yellow
Red				
Blue	2	2		
Green				
Yellow				

MACRO VS. MICRO AVERAGING INTUITION

- Micro-average accuracy in example is 95% (fraction of correct predictions regardless of class)
  - Even though accuracy for Class B is 0% and for Class A is 100%
- Macro-average accuracy is the average of the accuracies of the different classes
  - E.g.,: 0% (for class B) + 100% (for class A) = 50%

Note: Macro vs. Micro Averaging applies to other metrics like precision and recall, and to both binary & multi-class classification.

Object	Correct Label	Predicted Label
1	A	A
...	A	A
93	A	A
94	A	A
95	A	A
96	B	A
97	B	A
98	B	A
99	B	A
100	B	A

MULTI-CLASS CLASSIFICATION

- Multi-class classification is where we classify objects into more than 2 classes
- We compute TN, FN, FP, TP by viewing each class as the positive class and relative to it, all other classes are the negative class (macro average)

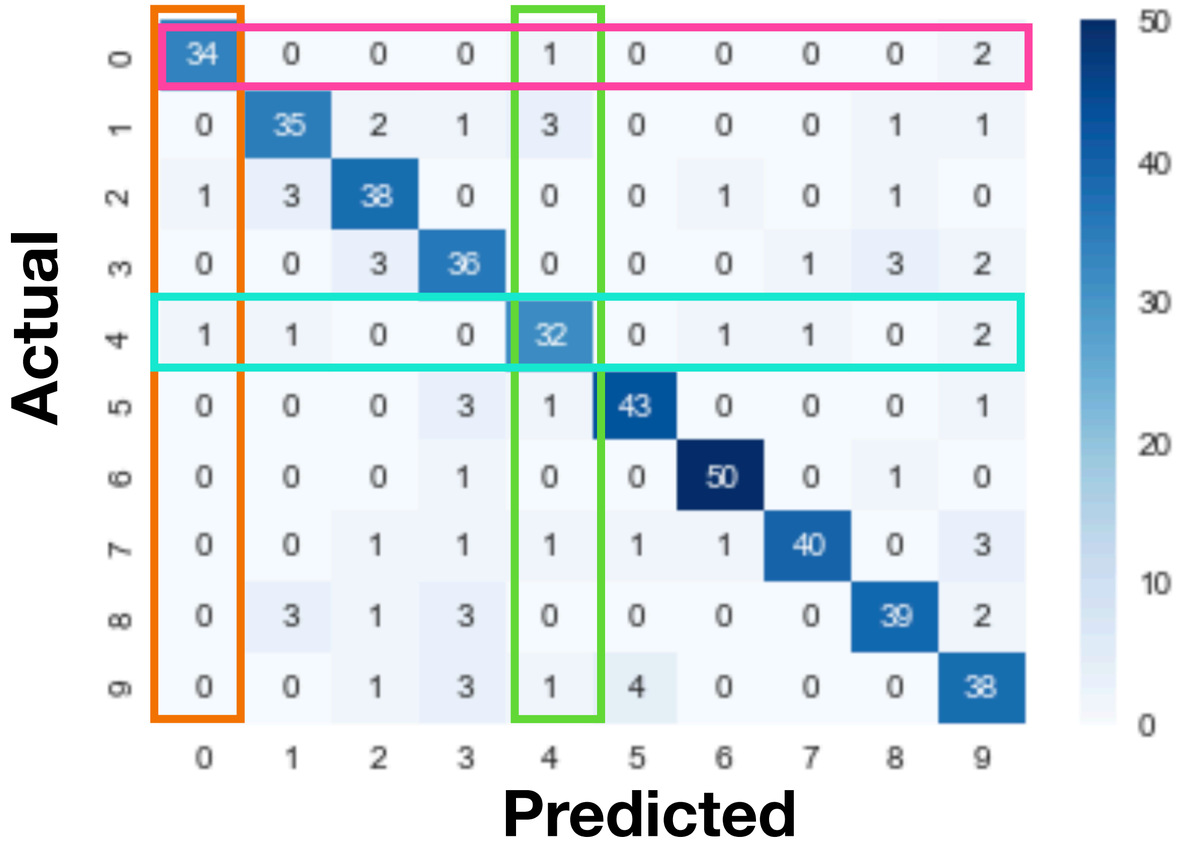
Precision for class 0: $\frac{34}{34+1+1} = 0.94$

Precision for class 4: $\frac{32}{32+1+3+1+1+1} = 0.82$

Recall for class 0: $\frac{34}{34+1+2} = 0.92$

Recall for class 4: $\frac{32}{32+1+1+1+1+2} = 0.84$

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



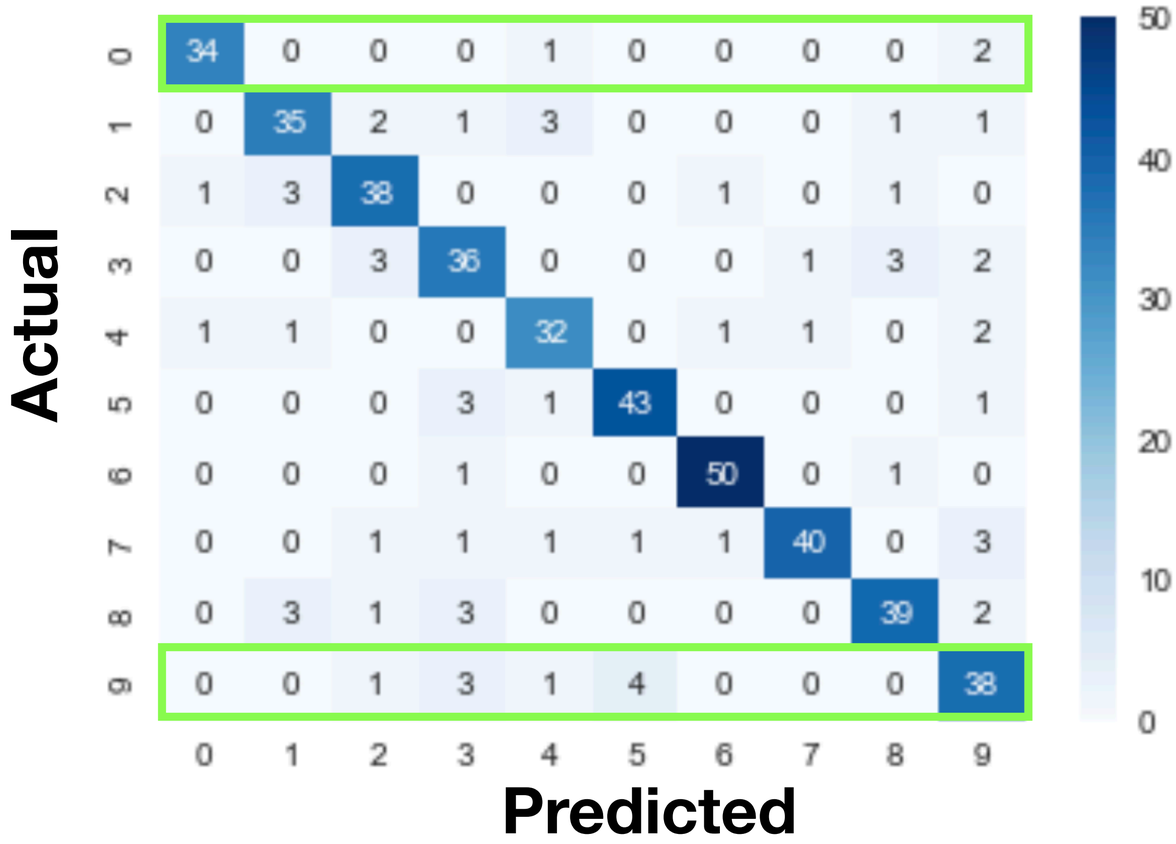
$$Recall = \frac{TP}{TP + FN}$$

MULTI-CLASS CLASSIFICATION METRIC COMPUTATION OPTIONS – RECALL EXAMPLE

- **Macro Average:** Metric is computed for each class, and then the resulting metrics are averaged together to arrive at overall metrics for the dataset (e.g., precision, recall, etc.)
  - Each class has the same impact on classifier performance measures regardless of number of samples of that class that are in the data.

Recall for class 0: $\frac{34}{34+1+2} = 0.92$

Recall for class 9: $\frac{38}{38+1+3+1+4} = 0.81$



Macro Average Recall = Ave(0.92, Recall for class 1, …, Recall for class 8, 0.81)

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

MULTI-CLASS CLASSIFICATION METRIC COMPUTATION OPTIONS – RECALL EXAMPLE

- **Micro Average:** All the samples are considered together.
  - Classes with a larger number of instances will have higher impact on classifier performance

$$TP_{total} = TP_0 + TP_1 + \ldots + TP_9 = 34+35\ldots+38$$

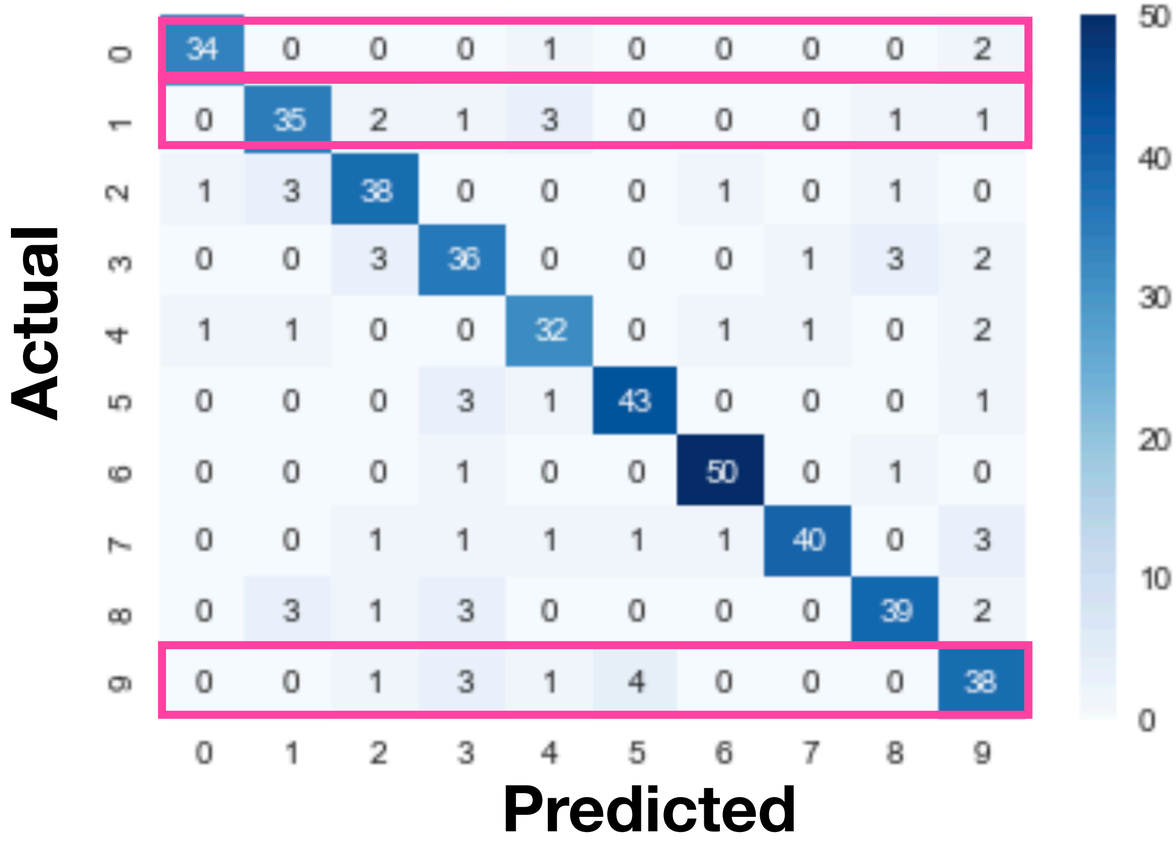
$$FN_{total} = FN_0 + FN_1 + \ldots + FN_9 = 3+8+\ldots+9$$

$FN_0 = 1+2 = 3$

$FN_1 = 2+1+3+1+1 = 8$

$FN_9 = 1+3+1+4 = 9$

$$\textbf{Micro Average } Recall = \frac{TP_{total}}{TP_{total} + FN_{total}}$$



STATISTICAL SIGNIFICANCE TESTING

MOTIVATION

- When comparing the performance of two models, we want to know whether the difference in their performance is truly indicative of one of them being better than the other
  - E.g., is 1% better f1 score indicative that model A is better than model B, or is this an artifact of our test set/experimental setup

- We use statistical hypothesis testing to ascertain what differences in performance are significant

OVERVIEW

Given a difference in performance δ between models A and B on test set x, using an evaluation metric M: $\delta(x) = M(A, x) - M(B, x)$

If we want to evaluate the claim that that A is better than B, we define the following hypotheses

$H_0 : \delta(x) \leq 0$ – the Null Hypothesis; assumes A is not better than B

$H_1 : \delta(x) > 0$ – the alternative hypothesis (if we are able to negate H_0, then H_1 would be true, which supports the claim that A is better than B)

- Approach: estimate across all test sets (X is a random variable over the test sets), how likely is it for the null hypothesis to be true

  In particular, p-value is the probability that we would see the performance difference $\delta(x)$ that we have observed (or even a larger difference), assuming the null hypothesis is true (which is that A is not better than B)

  p-value = $P(\delta(X) \geq \delta(x) \,|\, H_0 \; is \; true)$

  In other words, how likely is it, that under a null hypothesis, we would observe a difference in performance of $\delta(x)$ or larger.

- Small p-value indicates that the difference we observed is very unlikely under the null hypothesis and therefore we can reject the null hypothesis (and thus affirm the alternative hypothesis - this is a proof by contradiction)
  - Usually we use a p-value threshold of 0.01 or 0.05 to indicate statistical significance

COMPUTING STATISTICAL SIGNIFICANCE

- In statistics, there are a number of tests for determining statistical significance (e.g., t-test, z-test) but those make theoretical assumptions about the statistical properties of the data
- Instead, in NLP we use empirical tests — we create multiple test sets and compute the p-value based on on those
  - E.g., Paired Bootstrap Test described next

$H_0 : \delta(x) \leq 0$ – the Null Hypothesis assumes A is not better than B

$H_1 : \delta(x) > 0$ – A is better than B

PAIRED BOOTSTRAP TEST

- Paired - we are using the same test set for both models
- Bootstrapping - sampling with replacement used to generate a large number of smaller samples x^i (virtual test sets in this case) from a larger dataset
  - Assumption: each bootstrap sample is representative of the overall population (of data points)
- Sampling with replacement - pick data from D randomly + same data point can be picked multiple times

Initial dataset	**D**	1	2	3	4	5	6	7	8	9	10
Sampling without replacement	**D_1**	7	8	10	4	2	5	1	3	6	9
Sampling with replacement	**D_2**	10	4	9	10	2	3	2	7	3	2	3	10	10	4	9	10	2

Note that without replacement, you can only generate a dataset that is the same size or smaller than the original, whereas sampling with replacement you can generate any size dataset.

PAIRED BOOTSTRAP TEST (CONT.)

- We create a large number (e.g., b) of test sets $x^{(i)}$
- Due to some statistical assumptions, what we look for is whether $\delta(x^i) \geq 2\delta(x)$ holds true for a given test set $x^{(i)}$

Initially we had: p-value = $P(\delta(X) \geq \delta(x) \,|\, H_0 \; is \; true)$ which can be re-written as

p-value = $P(\delta(X) - \delta(x) \geq 0 \,|\, H_0 \; is \; true)$

Instead now compute the p-value as follows:

$$\text{p-value}(x) = \frac{1}{b}\sum_{i=1}^{b} \mathbb{1}\left(\delta(x^{(i)}) - \delta(x) \geq \delta(x)\right)$$

$$= \frac{1}{b}\sum_{i=1}^{b} \mathbb{1}\left(\delta(x^{(i)}) \geq 2\delta(x)\right)$$

The indicator function returns 1 if its input is True, 0 otherwise

Example: Given 10,000 tests, and a p-value threshold of 0.01, and we find that in 47 tests $\delta(x^i) \geq 2\delta(x)$, this results in a p-value of 0.0047 which is below the p-value threshold of 0.01 and thus we can reject the null hypothesis.

CLASSIFICATION PITFALLS

CLASSIFICATION HAS BEEN SHOWN TO OUTPUT OBJECTIONABLE RESULTS

- Representational harm
  - Classification algorithms that perpetuate negative stereotypes, e.g., associating negative sentiment with text generated by someone who is African American
- Toxicity detection
  - Classifiers labeling text as toxic when it references minority identities, blind or gay people
- Causes
  - ML algorithms replicate and often amplify biases in the input data
  - Human labelers may be biased
- Solutions
  - No universal solution
  - Important to create a Model Card which keeps track of data characteristics and pre-processing & model performance across different groups of people