Hyperparameter optimization

Grid-search, random-search and other methods for hyperparameter-search

see Ch. 05 in book "Python Machine Learning" by Raschka & Mirjalili

Overview

- Binary classifier example
- Confusion matrix
- Different evaluation metrics
- Receiver Operator Curve (ROC) and Area Under Curve (ROC-AUC)
- Performance evaluation for multiclass problems
  - Micro-averaging
  - Macro-averaging
- Other metrics worth knowing about

Binary classifier example

- Rare type of asthma occurs in approximately 1% of the population
- Can be deadly, but is a condition that is easy to treat
- Assume that you have two binary classifiers:
  - C1 with an accuracy of 99%
  - C2 with an accuracy of 98%
- Which one would you prefer?

Binary classifier example

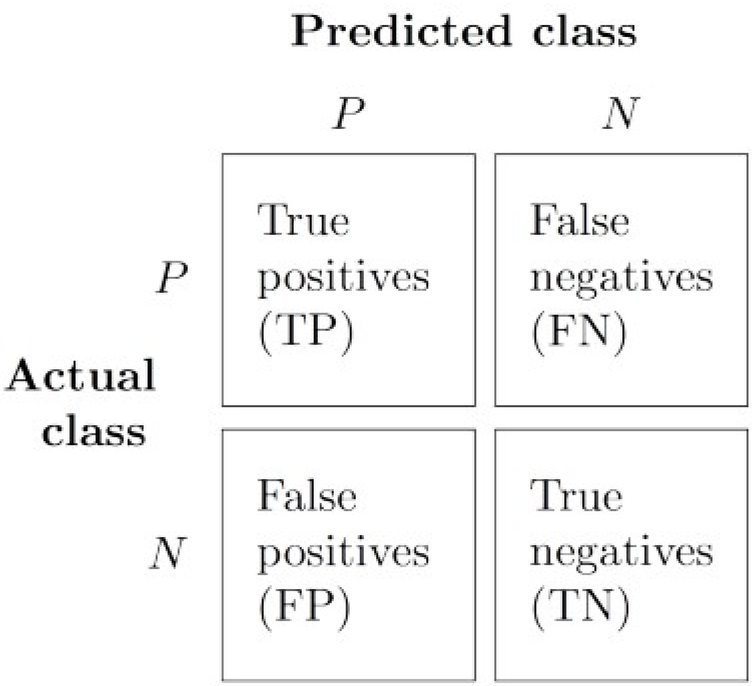
- Let us take a close look on how the two classifiers perform on the test set
- 1000 samples in total 990 healthy and 10 with the disease
- C1 makes 10 mistakes, C2 makes 18 mistakes
- Does anyone see any issue with C1, despite it being more accurate than C2?

IDX	1	2	3	...	973	974	975	976	977	978	979	980	981	982	983	984		
GT	0	0	0	...	0	0	0	0	0	0	0	0	0	0	0	0	No disease	Disease
C1	0	0	0	...	0	0	0	0	0	0	0	0	0	0	0	0	Correct	Incorrect
C2	0	0	0	...	0	1	1	1	1	1	1	1	1	1	1	1	Correct	Incorrect

IDX	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000		
GT	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	No disease	Disease
C1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	Correct	Incorrect
C2	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	Correct	Incorrect

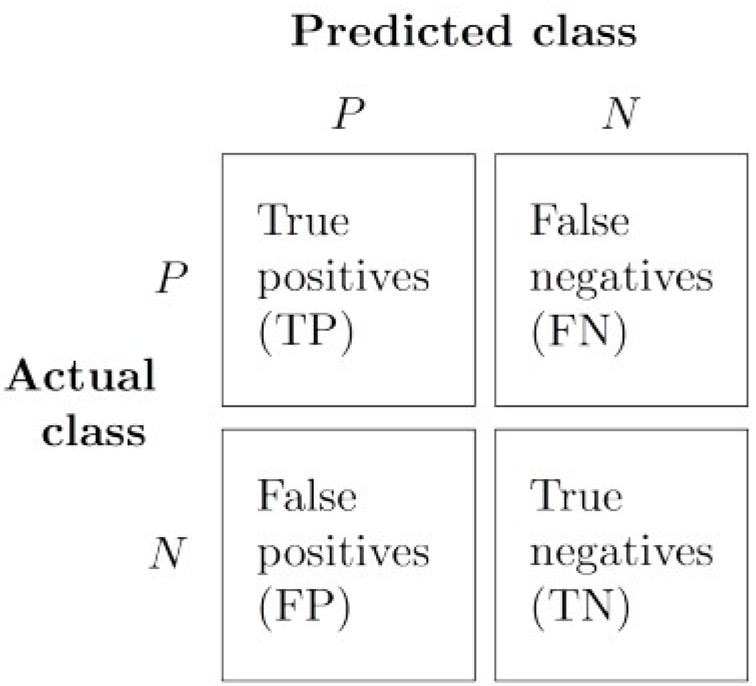
Confusion matrix

- Often we want to get a better idea of how well a classifier detects specific classes, and an overall accuracy is not enough
- To get a more thorough understanding of a models predicted performance we look at a models **confusion matrix**
- For binary classifiers it looks like to matrix to the right



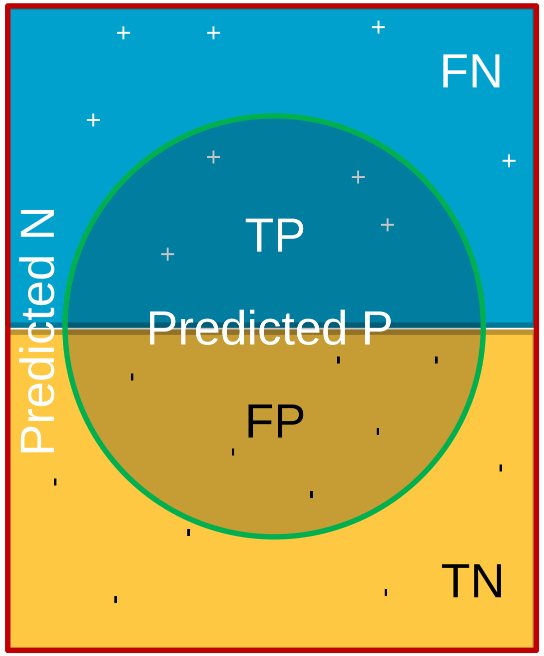
Confusion matrix

- It is a square matrix showing the classes that a model predicts versus the classes of the ground truth
- Let 1 be positive and 0 be negative
- **TP**: Positive sample model predicts positive
- **FP**: Negative sample model predicts positive
- **FN**: Positive sample model predicts negative
- **TN**: Negative sample model predicts negative

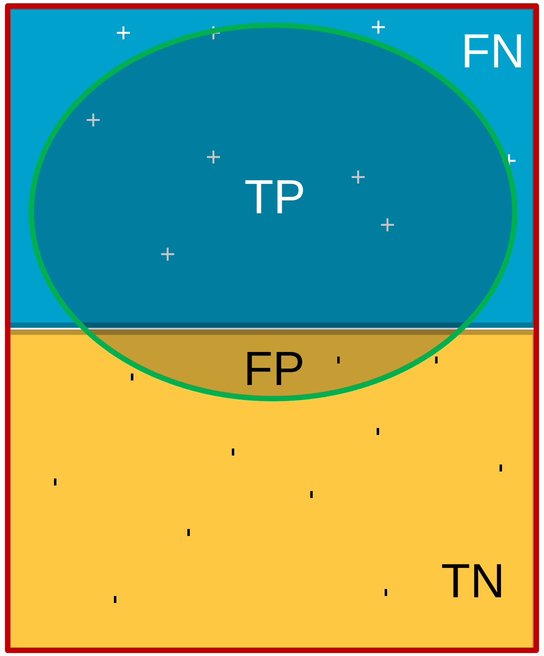


Confusion matrix

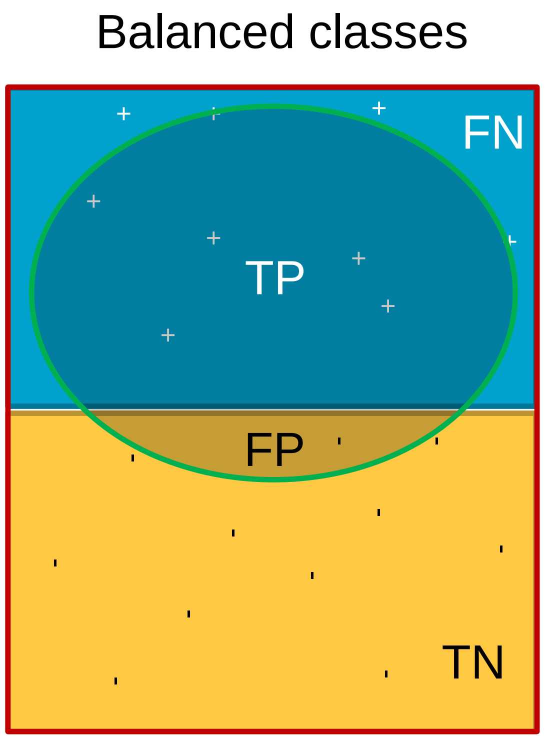
Bad model

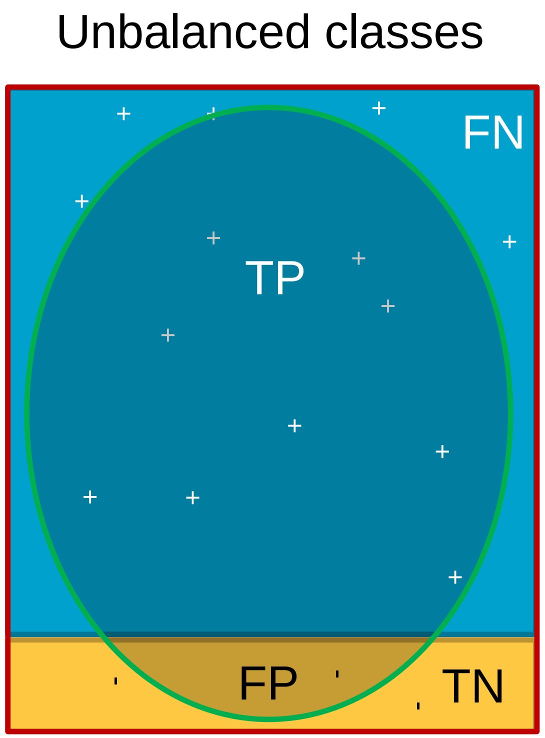


Better model



Confusion matrix





Predict everything positive?

Confusion matrix

- Lets compute the confusion matrix of CA1 and CA2 from the rare asthma condition example

		Predicted class	
		P	N
Actual class	P	True positives (TP)	False negatives (FN)
	N	False positives (FP)	True negatives (TN)

IDX	1	2	3	...	973	974	975	976	977	978	979	980	981	982	983	984		
GT	0	0	0	...	0	0	0	0	0	0	0	0	0	0	0	0	No disease	Disease
C1	0	0	0	...	0	0	0	0	0	0	0	0	0	0	0	0	Correct	Incorrect
C2	0	0	0	...	0	1	1	1	1	1	1	1	1	1	1	1	Correct	Incorrect

IDX	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000		
GT	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	No disease	Disease
C1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	Correct	Incorrect
C2	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	Correct	Incorrect

Confusion matrix

- Lets compute the confusion matrix of CA1 and CA2 from the rare asthma condition example

	PP	*PN*
P	TP	FN
N	FP	TN

C1

	PP	*PN*
P	1	9
N	1	989

C2

	PP	*PN*
P	9	1
N	17	973

Different evaluation metrics

- As of now you should be familiar with two evaluation metrics
  - **Accuracy (ACC)**
  - **Prediction error (ERR)**
- **ACC**: Number of **correct** predictions divided by number of total predictions
- **ERR**: Number of **incorrect** predictions divided by number of total predictions

Different evaluation metrics

- As of now you should be familiar with two evaluation metrics
  - **Accuracy (ACC)**
  - **Prediction error (ERR)**

$$\mathrm{ACC} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\mathrm{ERR} = \frac{FP + FN}{TP + TN + FP + FN} = 1 - \mathrm{ACC}$$

Different evaluation metrics

- Different evalutation metrics are important for different classification problems
- True Positive Rate (**TPR**), False Positive Rate (**FPR**), False Negative Rate (**FNR**) and True Negative Rate (**TNR**) are especially useful for imbalanced datasets
- In some problems it will be important to minimize the FNR, in other problems it will not be as important

$$\mathbf{TPR} = \frac{TP}{P} = \frac{TP}{TP + FN}$$

$$\mathbf{FPR} = \frac{FP}{N} = \frac{FP}{FP + TN}$$

$$\mathbf{FNR} = \frac{FN}{P} = \frac{FN}{FN + TP}$$

$$\mathbf{TNR} = \frac{TN}{N} = \frac{TP}{TN + FP}$$

Different evaluation metrics

- **Precision** (PRE), **Recall** (REC) and **F1**-score are central metrics in this course
- Precision seeks to measure the amount of TP’s in relation to FP’s
- Recall (also known as *sensitivity* in medicine) is equivilant to TPR

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

- Often, optimizing for recall might come at the cost of lowering precision
- F1-score is a metric that seeks to combine precision and recall

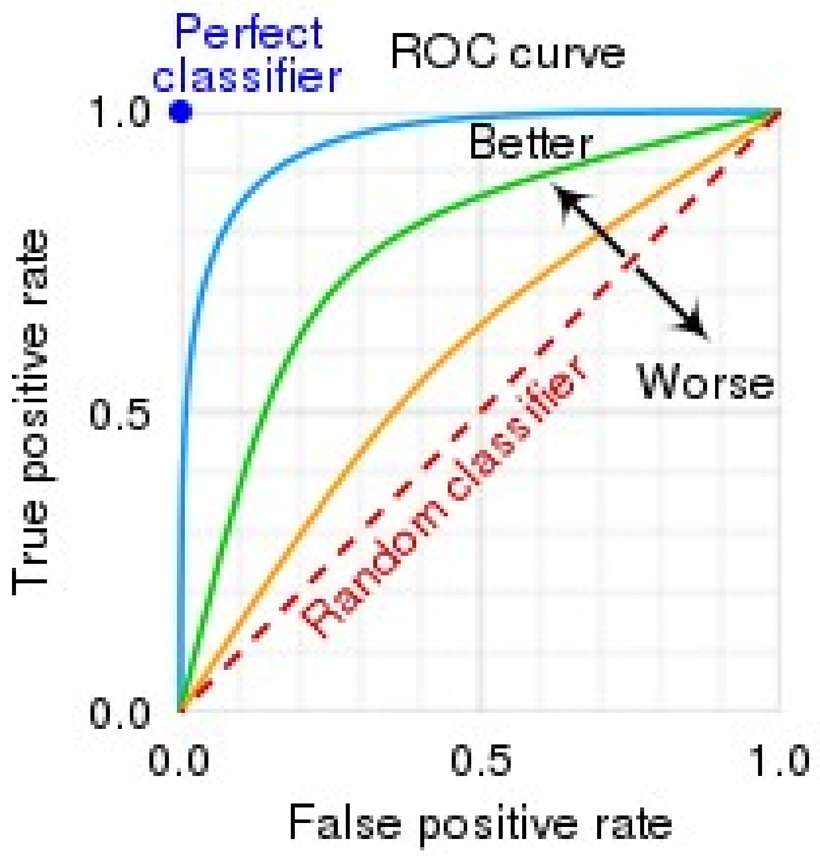
$$\textbf{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Examples precision recall and F1

- Let us compute the precision, recall and F1-scores of our C1 and C2 classifiers for the Asthma example

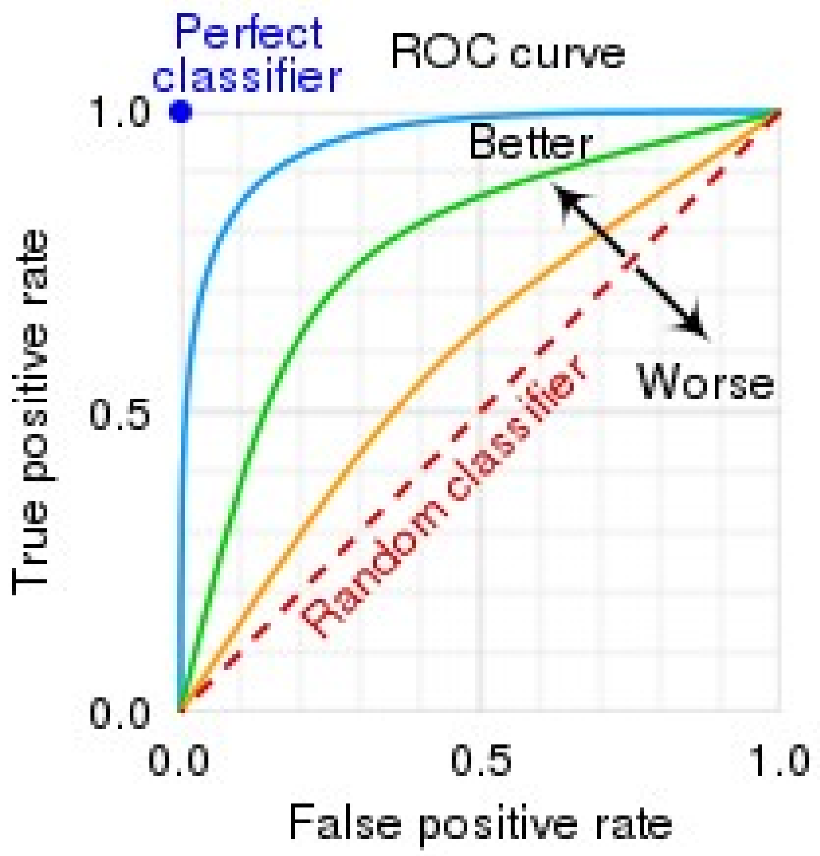
Receiver Operator Curve and Area Under Curve

- The Receiver Operator Curve (ROC) is a graphical representation of a classifier performance.
- Plots TPR versus FPR for a binary classifier at **different decision thresholds**
- Can be understood as trying to visualize how much better a model is than random guessing
- Can be used with any classifier that applies a decision boundary (the majority of classifiers)
- Online illustration



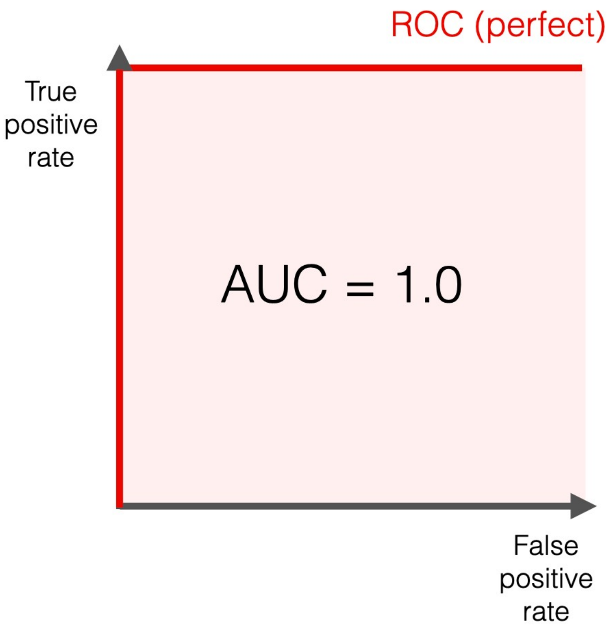
Receiver Operator Curve and Area Under Curve

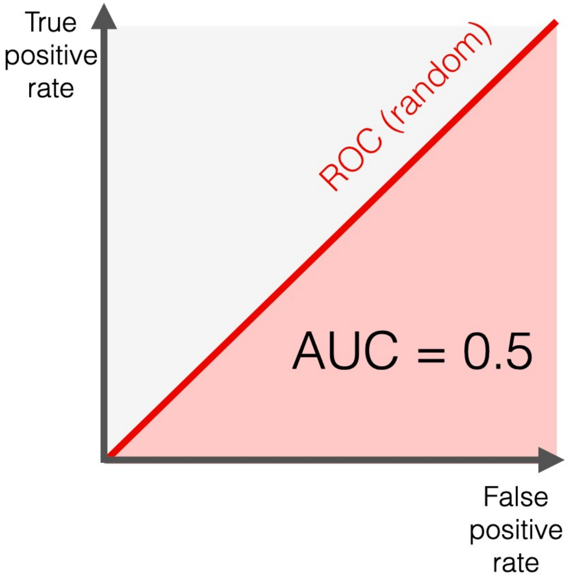
- Illustrates the trade-off between FPR and TPR
- To compute the ROC classifiers must output a probabilistic output/softmax output which can be thresholded
- Diagonal curve is the performance of random guessing (worst possible score)
- Blue curve is better



Receiver Operator Curve and Area Under Curve

- ROC-AUC: Area Under the Receiver Operating Characteristic Curve
- Quantitative measure of overall classifier performance at all possible thresholds



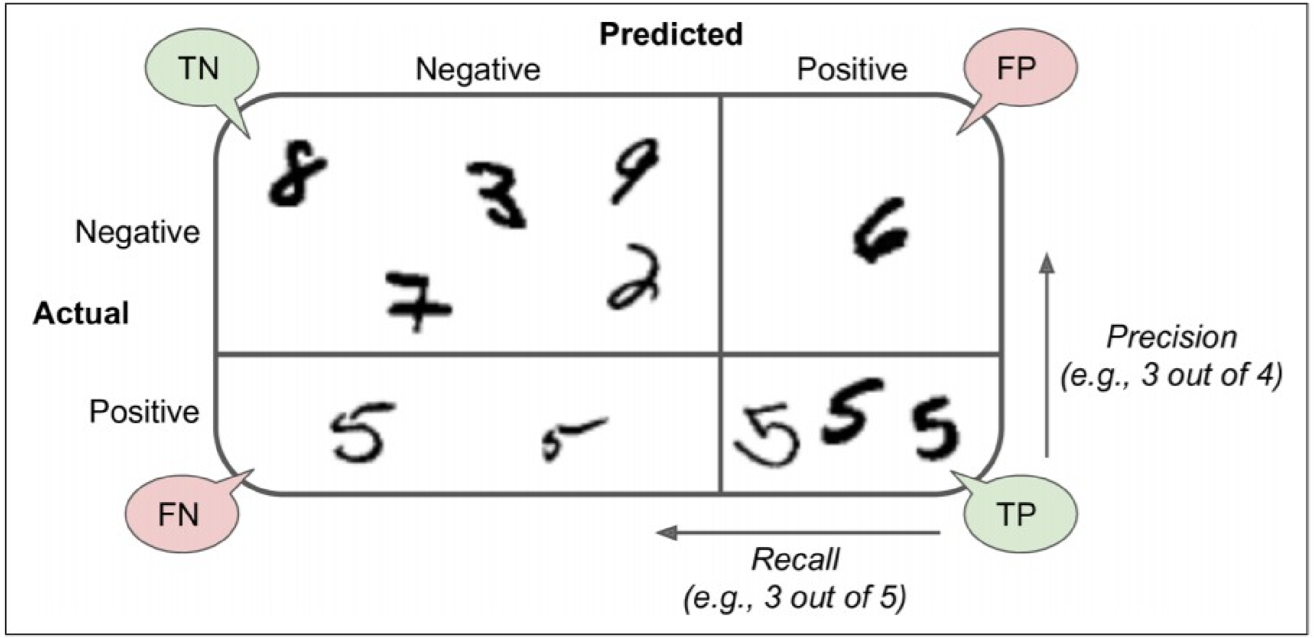


Examples ROC-AUC

- `ROC_AUC_with_LR.ipynb`
  - Example of how visualize the ROC and compute the AUC

Performance evaluation for multiclass problems

- When evalutating multiclass models: compute multiple “one-vs-all” assessments
- This is what one one-vs-all assessment looks like for handwritten digits: 5 vs. “not 5”

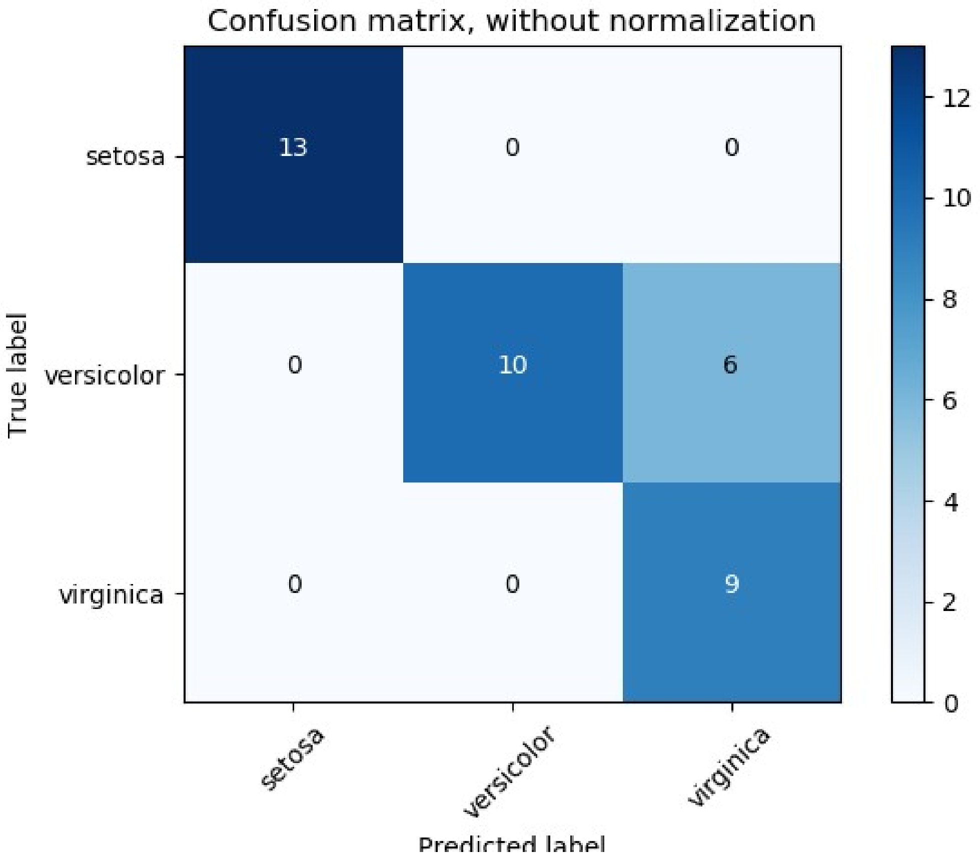


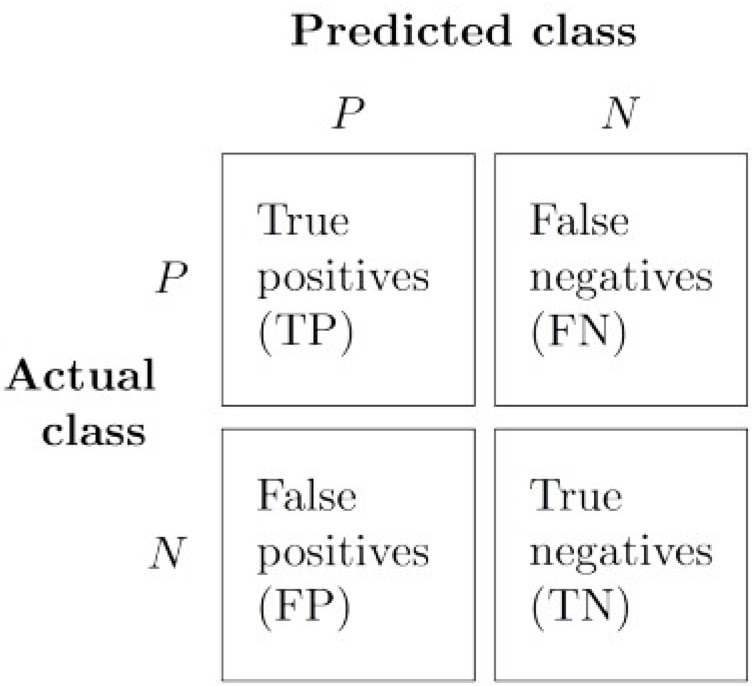
Performance evaluation for multiclass problems

- Can choose between **micro** and **macro** averaging:
  - Micro gives **equal weight** to every **sample**
  - Macro gives **equal weight** to every **class** (sample/class_size)
- Micro/macro averaging can be done for any performance metric
- Can even be used for binary classification problems

Performance evaluation for multiclass problems

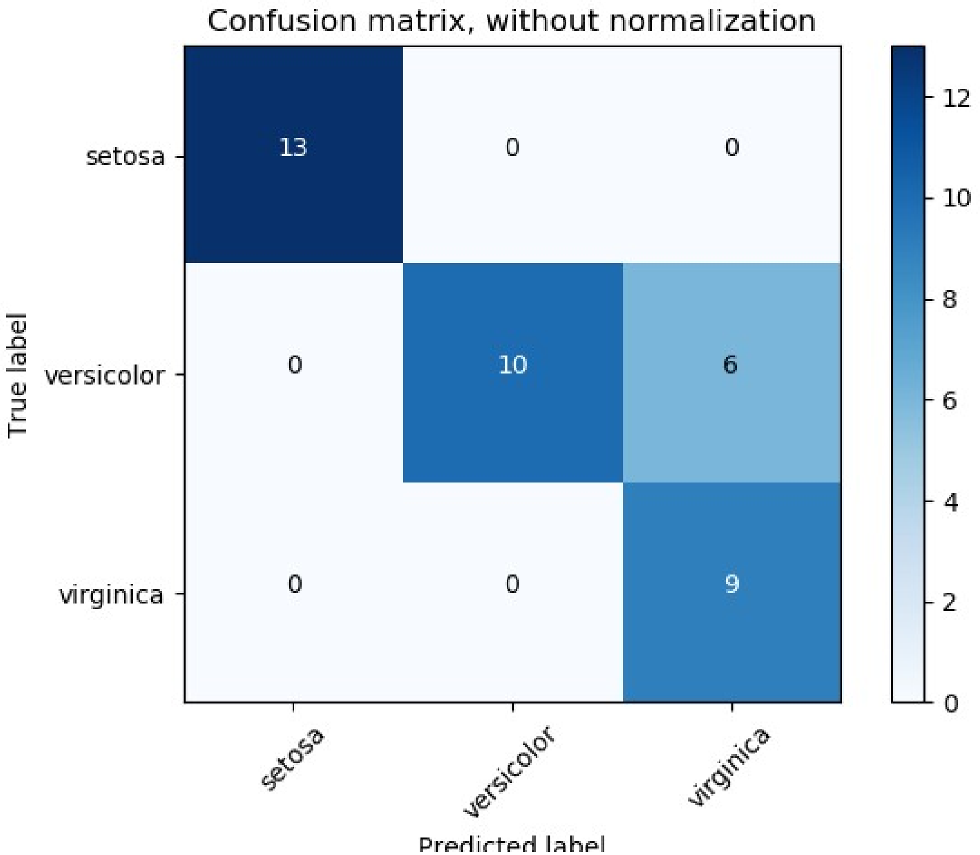
- Confusion matrices for *n*-class multiclass problems have *n* x *n* dimensions





Performance evaluation for multiclass problems

- Let us use this confusion matrix to illustrate difference between micro/macro averaging



Performance evaluation for multiclass problems

- Lets look at the example from the beginning of the class to show how micro/macro averaging can be used for binary classifiers

	PP	*PN*
P	TP	FN
N	FP	TN

C1

	PP	*PN*
P	1	9
N	1	989

C2

	PP	*PN*
P	9	1
N	17	973

Performance evaluation for multiclass problems

- There are almost as many evaluation metrics as there are problems
- Different people also use different names for the same metric
  - TPR/sensitivity/recall
  - precision/positive predictive value
- Some are application specific
  - Computer vision: mean Average Precision, Intersection over Union, etc.
  - Medicine: Diagnostic Odds Ratio
  - Unsupervised learning: Adjusted Rand Index

Performance evaluation for multiclass problems

- Some of my favourites include:
  - F1-score
  - Matthews Correlation Coefficient (MCC)
  - Balanced Accuracy

Performance evaluation for multiclass problems

Check out the Wikipedia page, or scikit-learn documentation for a good overview

		Predicted condition		Sources: [4][5][6][7][8][9][10][11][12] view · talk · edit	
	Total population $= \mathrm{P} + \mathrm{N}$	**Predicted Positive (PP)**	**Predicted Negative (PN)**	Informedness, bookmaker informedness (BM) $= \mathrm{TPR} + \mathrm{TNR} - 1$	Prevalence threshold (PT) $= \frac{\sqrt{\mathrm{TPR} \times \mathrm{FPR}} - \mathrm{FPR}}{\mathrm{TPR} - \mathrm{FPR}}$
Actual condition	**Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{\mathrm{TP}}{\mathrm{P}} = 1 - \mathrm{FNR}$	False negative rate (FNR), miss rate type II error [c] $= \frac{\mathrm{FN}}{\mathrm{P}} = 1 - \mathrm{TPR}$
Actual condition	**Negative (N)**[d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]	False positive rate (FPR), probability of false alarm, fall-out type I error [f] $= \frac{\mathrm{FP}}{\mathrm{N}} = 1 - \mathrm{TNR}$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{\mathrm{TN}}{\mathrm{N}} = 1 - \mathrm{FPR}$
	Prevalence $= \frac{\mathrm{P}}{\mathrm{P} + \mathrm{N}}$	Positive predictive value (PPV), precision $= \frac{\mathrm{TP}}{\mathrm{PP}} = 1 - \mathrm{FDR}$	False omission rate (FOR) $= \frac{\mathrm{FN}}{\mathrm{PN}} = 1 - \mathrm{NPV}$	Positive likelihood ratio (LR+) $= \frac{\mathrm{TPR}}{\mathrm{FPR}}$	Negative likelihood ratio (LR−) $= \frac{\mathrm{FNR}}{\mathrm{TNR}}$
	Accuracy (ACC) $= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}}$	False discovery rate (FDR) $= \frac{\mathrm{FP}}{\mathrm{PP}} = 1 - \mathrm{PPV}$	Negative predictive value (NPV) $= \frac{\mathrm{TN}}{\mathrm{PN}} = 1 - \mathrm{FOR}$	Markedness (MK), deltaP (Δp) $= \mathrm{PPV} + \mathrm{NPV} - 1$	Diagnostic odds ratio (DOR) $= \frac{\mathrm{LR+}}{\mathrm{LR-}}$
	Balanced accuracy (BA) $= \frac{\mathrm{TPR} + \mathrm{TNR}}{2}$	F_1 score $= \frac{2\,\mathrm{PPV} \times \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$	Fowlkes–Mallows index (FM) $= \sqrt{\mathrm{PPV} \times \mathrm{TPR}}$	Matthews correlation coefficient (MCC) $= \sqrt{\mathrm{TPR} \times \mathrm{TNR} \times \mathrm{PPV} \times \mathrm{NPV}} - \sqrt{\mathrm{FNR} \times \mathrm{FPR} \times \mathrm{FOR} \times \mathrm{FDR}}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$

Other metrics worth knowing about

- You don't need to memorize the formulas for all the evaluation metrics that you have been presented in this lecture for the exam
- Understand the **concepts** of the metrics such that you can use them later
  - Should be able to remember what the purpose of a confusion matrix
  - Assess when it is appropriate to use which metric
  - When working with multiclass, when one should use micro/macro averaging

Other metrics worth knowing about

- That being said, there are a few things you should memorize for the exam
  - Confusion matrix
  - Precision, Recall and F1-score
  - Concept of an ROC curve, and ROC-AUC metric
  - Difference between micro and macro averaging when computing multiclass metrics

Examples metrics with scikit-learn

- `metrics_example.ipynb`
  - Example of how compute evaluation metrics with scikit-learn

Thank you for listening