ITCS 6156/8156 Fall 2023
Machine Learning

Measurements & Multi-class Classification

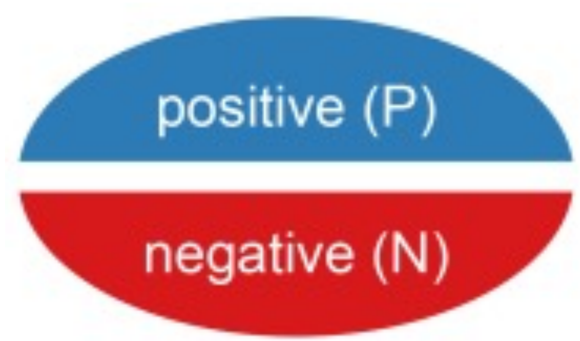
Instructor: Hongfei Xue
Email: hongfei.xue@charlotte.edu
Class Meeting: Mon & Wed, 4:00 PM – 5:15 PM, CHHS 376

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

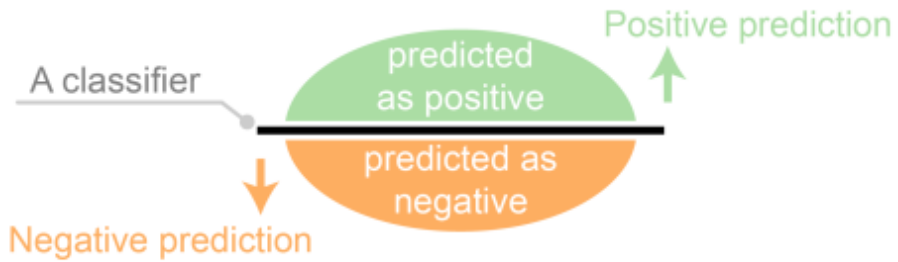
Some content in the slides is based on Dr. Razvan's lecture

Binary Classification

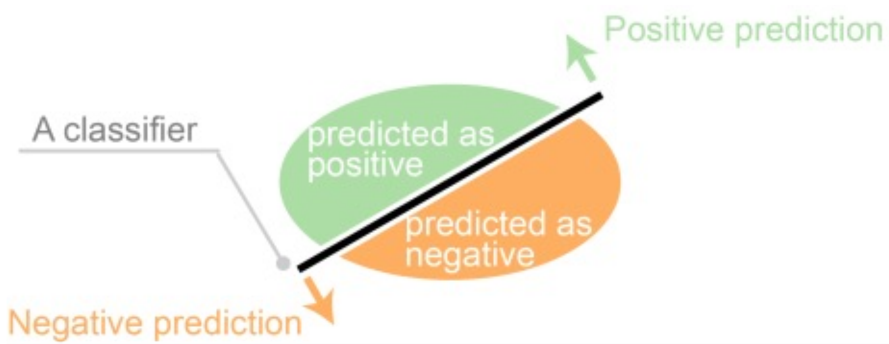
- Test dataset for evaluation:
  - In binary classification dataset, each instance will have its true label (true class): Positive Class (P) vs Negative Class (N).



- Predictions on test dataset:
  - A perfect classifier



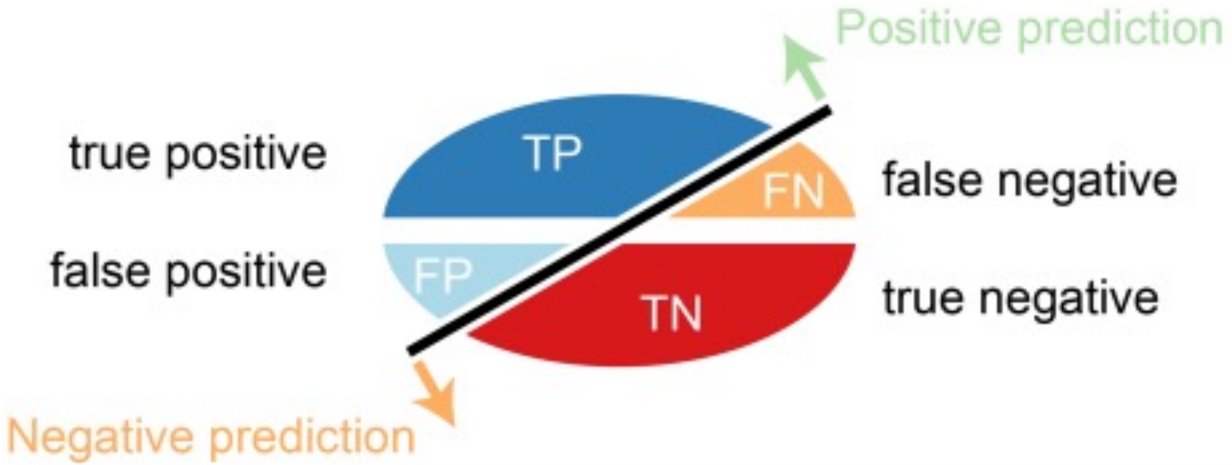
  - A real-world classifier



Images from https://classeval.wordpress.com/introduction/basic-evaluation-measures/

Confusion Matrix

- **Confusion matrix** (a 2x2 table) is composed of four outcomes of classification:
  - True positive (TP): correct positive prediction
  - False positive (FP): incorrect positive prediction
  - True negative (TN): correct negative prediction
  - False negative (FN): incorrect negative prediction



True	Prediction	Positive	Negative
Positive		# of TPs	# of FNs
Negative		# of FPs	# of TNs

Basic Measurements

- **Accuracy** is calculated as the number of all correct predictions divided by the total number of the dataset.

Accuracy: (TP + TN) / (P + N)

- **Recall** (sensitivity, true positive rate) is calculated as the number of correct positive predictions divided by the total number of positives.

Sensitivity: TP / P

- **Precision** is calculated as the number of correct positive predictions divided by the total number of positive predictions.

Precision: TP / (TP + FP)

- **F1 Score** is a harmonic mean of precision and recall.

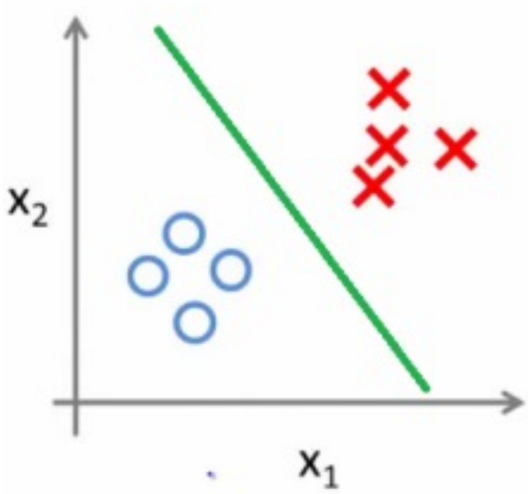
$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{2\mathrm{tp}}{2\mathrm{tp} + \mathrm{fp} + \mathrm{fn}}$$

Multi-class Classification

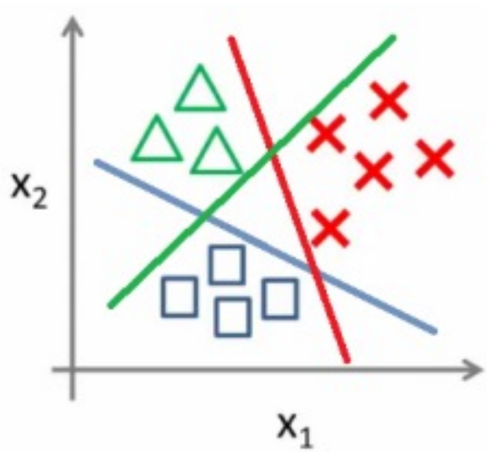
- **Multi-class Classification:**
  - To classify instances into one of more than two classes. (i.e., there are more than two possible categories or labels)

- **Strategies:**
  - One-vs-All (One-vs-Rest)
  - One-vs-One
  - Softmax Regression
  - Decision Trees(Later)

- Binary classification:



- Multi-class classification:



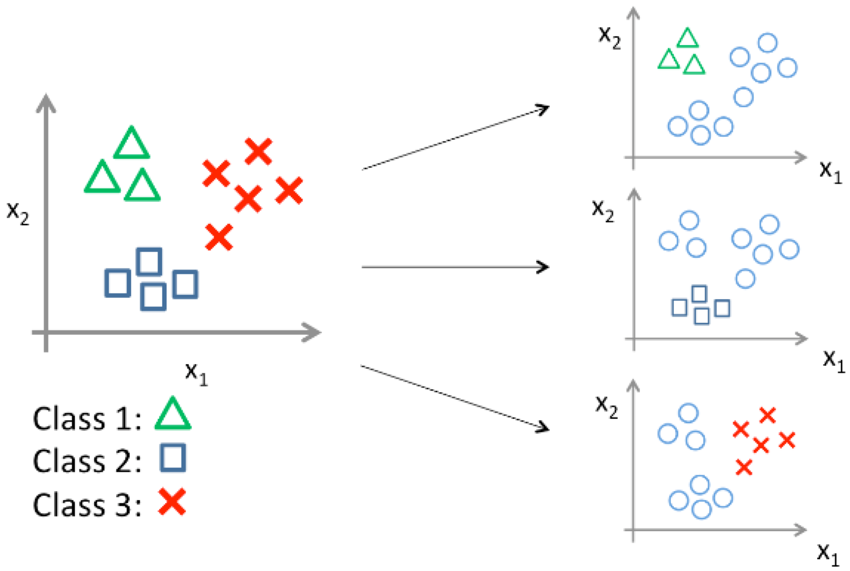
Images from: https://utkuufuk.com/2018/06/03/one-vs-all-classification/

One-vs-All

- **One-vs-all classification** breaks down N classes present in the dataset into N binary classifier models that aims to classify a data point as either part of the current class or not.

- Suppose you have classes 1, 2, and 3.
  - Model A: 1 or 2,3 (1 or not 1)
  - Model B: 2 or 1,3 (2 or not 2)
  - Model C: 3 or 1,2 (3 or not 3)

- At prediction time, the class that corresponds to the classifier with the highest confidence score is the predicted class.
  - Model A: $P(x = 1)$ and $P(x \neq 1)$
  - Model B: $P(x = 2)$ and $P(x \neq 2)$
  - Model C: $P(x = 3)$ and $P(x \neq 3)$
  - Among $P(x = 1)$, $P(x = 2)$, and $P(x = 3)$, which one is the highest?



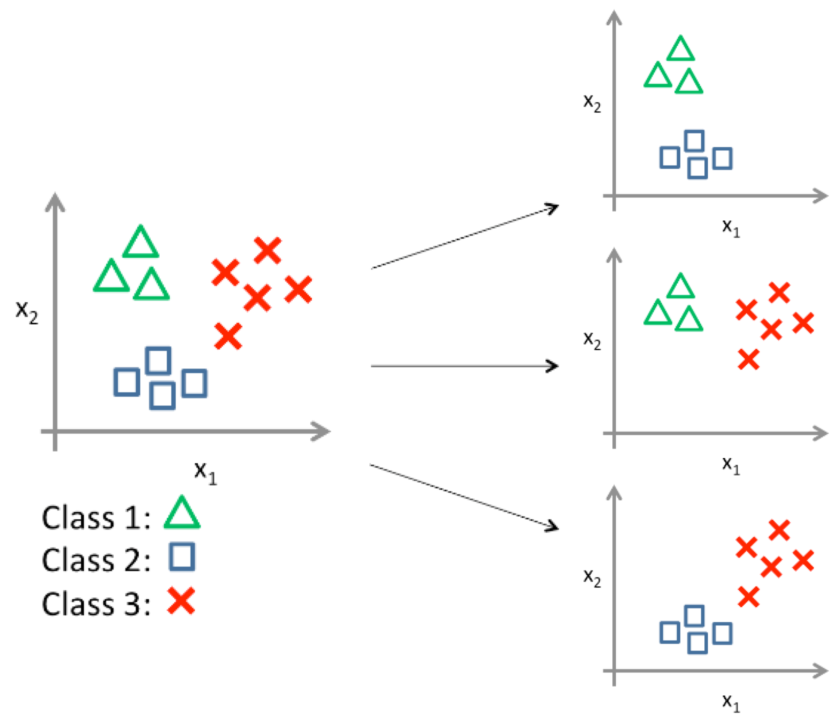
Images from: https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj4/html/jnanda3/index.html

One-vs-one

- **One-vs-one classification** breaks down *N* classes present in the dataset into ***N*(N-1)/2*** binary classifier models – one for each pair of classes.

- Suppose you have classes 1, 2, and 3.
  - Model A: 1 or 2
  - Model B: 1 or 3
  - Model C: 2 or 3

- At prediction time, each classifier votes for a class, and the class with the most votes is the predicted class.
  - Model A: Vote for 1 or 2
  - Model B: Vote for 1 or 3
  - Model C: Vote for 2 or 3
  - Classes 1, 2, and 3, which one has the most votes?



Softmax Regression

- Multiclass classification

$$Y = \{C_1, C_{2,} \cdots, C_K\} = \{1,2,\cdots,K\}.$$

- Training set is $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)$.

$$\mathbf{x} = [1, x_1, x_2, \cdots, x_M]$$
$$y_1, y_2, \cdots, y_n \in \{1,2,\cdots,K\}$$

- One weight vector per class:

$$p(C_k|\mathbf{x}_n) = \frac{\exp(\mathbf{w}_k^T \mathbf{x}_n + b_k)}{\sum_{j=1,\cdots,K} \exp(\mathbf{w}_j^T \mathbf{x}_n + b_j)}$$

Softmax Regression

- Inference:

$$
\begin{aligned}
C_* &= \underset{C_k}{\operatorname{argmax}}\; p(C_k|\mathbf{x}) \\
&= \underset{C_k}{\operatorname{argmax}}\; \frac{\exp(\mathbf{w}_k^T\mathbf{x})}{\sum_j \exp(\mathbf{w}_j^T\mathbf{x})} \\
&= \underset{C_k}{\operatorname{argmax}}\; \exp\left(\mathbf{w}_k^T\mathbf{x}\right) \\
&= \underset{C_k}{\operatorname{argmax}}\; \mathbf{w}_k^T\mathbf{x}
\end{aligned}
$$

$Z(\mathbf{x})$ a normalization constant

- Training using:
  - Maximum Likelihood (ML)
  - Maximum A Posteriori (MAP) with a Gaussian prior on **w**.

Softmax Regression

- The negative log-likelihood error function is:

$$E_D(\mathbf{w}) = -\frac{1}{N}\ln\prod_{n=1}^{N} p(y_n|\mathbf{x}_n) = \boxed{-\frac{1}{N}\sum_{n=1}^{N}\ln\frac{\exp(\mathbf{w}_{y_n}^T\mathbf{x}_n)}{Z(\mathbf{x}_n)}}$$

convex in **w**

- The Maximum Likelihood solution is:

$$\mathbf{w}_{ML} = \underset{\mathbf{w}}{\text{argmin}}\ E_D(\mathbf{w})$$

- The gradient is (prove it):

$$\nabla_{w_k}E_D(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}(\delta_k(t_n) - p(C_k|\mathbf{x}_n))\mathbf{x}_n$$

where$\delta_t(x) = \begin{cases} 1, & x = t \\ 0, & x \neq t \end{cases}$ is the Kronecker delta function

Regularized Softmax Regression

- The new cost function is:

$$E(\mathbf{w}) = E_D(\mathbf{w}) + E_{\mathbf{w}}(\mathbf{w})$$
$$= -\frac{1}{N}\sum_{n=1}^{N} \ln \frac{\exp\left(\mathbf{w}_{t_n}^T \mathbf{x}_n\right)}{Z(\mathbf{x}_n)} + \frac{\alpha}{2}\|\mathbf{w}\|^2$$

- The new gradient is (prove it):

$$grad_k = \nabla_{w_k} E(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}(\delta_k(t_n) - p(C_k|\mathbf{x}_n))\mathbf{x}_n + \alpha\mathbf{w}_k$$

Questions?

UNIVERSITY OF NORTH CAROLINA
CHARLOTTE