Lecture 4: Introduction to Machine Learning and Performance Metrics

References

- Some material based on
  - *Vijay Pande, Patrick Walters, Peter Eastman, Bharath Ramsundar. Deep Learning For The Life Sciences, 2019.*
- Some figures are based on
  - Ubershmekel's Uberpython Pythonlog, Link

Overall Goal

- In most cases, our goal is to create a mathematical function

$$y = f(\boldsymbol{x})$$

$$x \longrightarrow \boxed{f(x)} \longrightarrow y$$

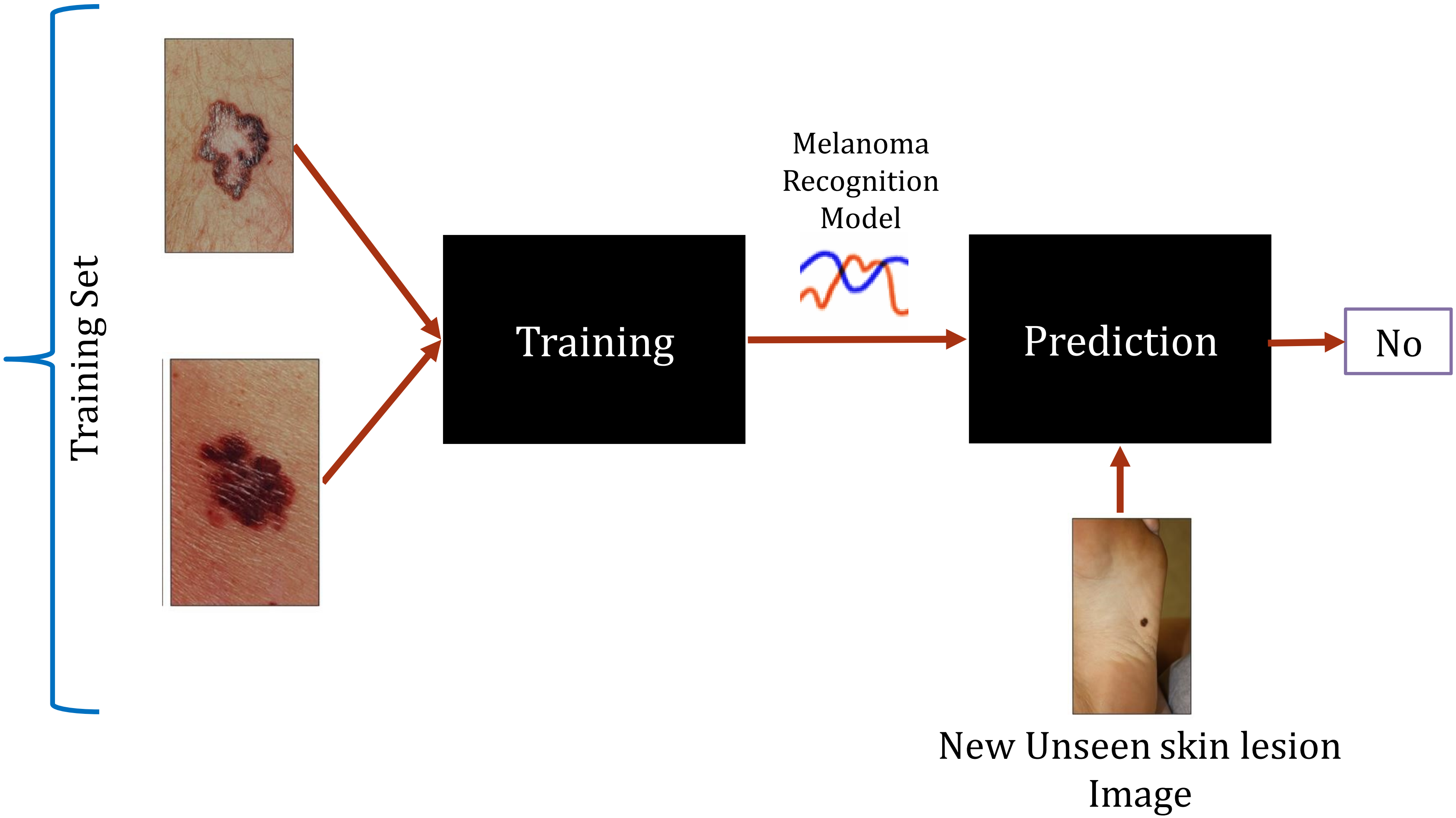
- x can be all the pixels in an image,
- y can be {0= cat, 1= dog}

- x can be the DNA sequence of a chromosome
- y can be {0=non-coding, 1=protein-coding}
- ...

Creating $f(x)$

- How can we design $f(x)$? This can be a very complicated function.
- Approach 1: We could analyze the problem and design a function by hand.
  - E.g. what pixel patterns in a skin lesion image tend to indicate melanoma?
  - Slow, time-consuming, subjective

Approach 2: Machine Learning Approach



Terminology Refresher: Feature

- Features = the set of attributes associated with an example
- (aka Independent variable in statistics)

Feature

Label

Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class Label
2	5	1	1	1	2	1	3	-1
2	5	4	4	5	7	10	3	+1
3	2	1	1	1	2	5	4	?

Model Evaluation

- In some models such as neural networks, we define a loss function $L(y, \acute{y})$.
  - It tells us whether the model output $\acute{y}$ is close to the provided ground truth y.
  - More specifically, most of the time we want to reduce the average loss (lower is better)

$$\bar{L} = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \acute{y}_i)$$

  - We need to choose an appropriate loss function for each problem, e.g. cross entropy (more on that later).

Model Evaluation

- Typically emphasis is on the predictive capability of a model
  - Rather than how fast it takes to classify or build models

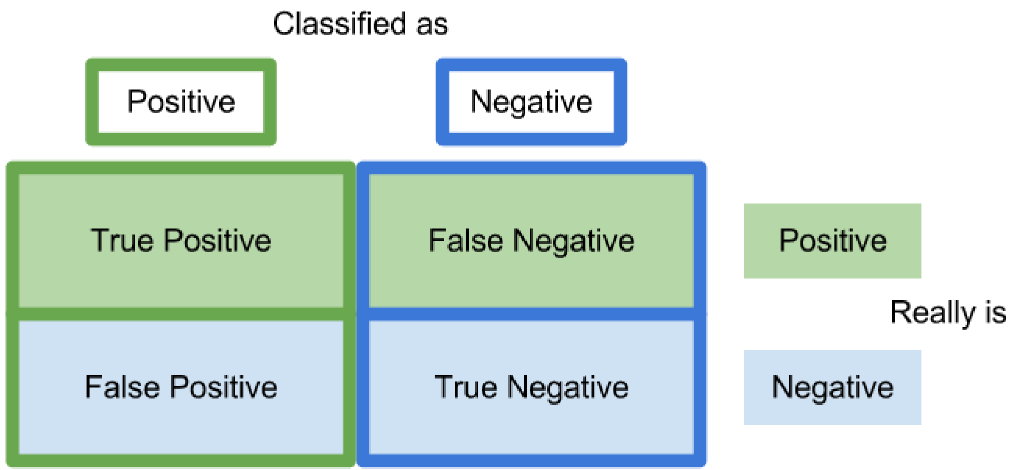
Positive/Negative

- True positive (TP) – a person we predicted to have sepsis who really had sepsis.
- True negative (TN)– a person we predicted not to have sepsis who really didn't have sepsis.
- False negative (FN) – a person we said doesn't have sepsis, though they really had.
- False positive (FP) – a person we said has sepsis, though they didn't.

http://www.info.univ-angers.fr/~gh/Predipath/classification.php

Confusion Matrix

- Confusion Matrix



Metrics for Performance Evaluation…

- Most widely-used metric is accuracy

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

n = 165	Predicted: No	Predicted: Yes
Actual: No	50	10
Actual: Yes	5	100

Limitation of Accuracy

- Consider a 2-class problem in a skewed dataset
  - Number of negative examples = 9990
  - Number of positive examples = 10

- If model predicts everything to be negative, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any positive condition

Performance Metrics

$$sensitivity\ (recall\ or\ TP\ rate) = \frac{TP}{TP+FN}$$

$$specificity = \frac{TN}{TN + FP}$$

$$precision\ (PPV) = \frac{TP}{TP + FP}$$

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Metrics Explained

- Sensitivity/recall: how good a test is at detecting a medical condition.
- Specificity: how good a test is at avoiding false alarms for healthy subjects.
- Precision: how many of the positively classified were relevant.

Ubershmekel's Uberpython Pythonlog, [Link]()

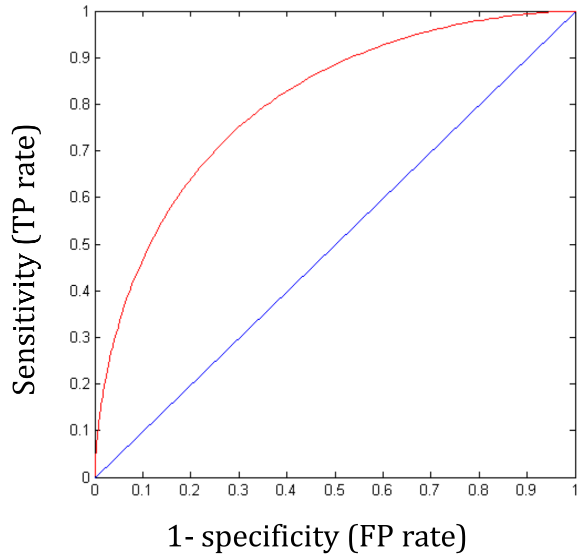
Metrics – Cheating?

- Sensitivity/recall: maximize by always returning +
- Specificity: maximize by always returning -
- Precision: maximize by only returning + on one sample we are most confident in.
- The cheating can be resolved by looking at several metrics instead of just one.
  - E.g. the cheating 100% sensitivity that always has 0% specificity.

Ubershmekel's Uberpython Pythonlog, Link

ROC (Receiver Operating Characteristics) curve

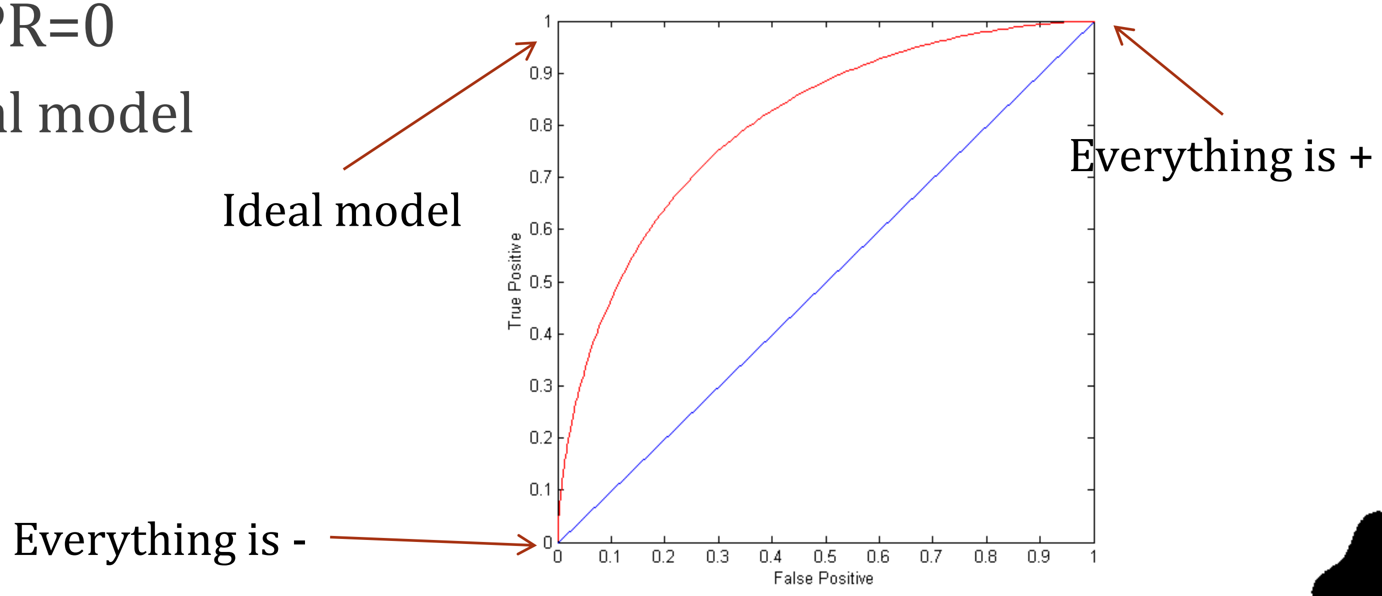
- A performance measurement for classification problem at different thresholds settings
- AUC (Area Under The Curve)
  - degree or measure of separability (0-1)
  - Higher the AUC, better the model is at distinguishing between patients with disease and no disease
- Changing the threshold of the algorithm, data sample, or cost matrix changes the location of the point



https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

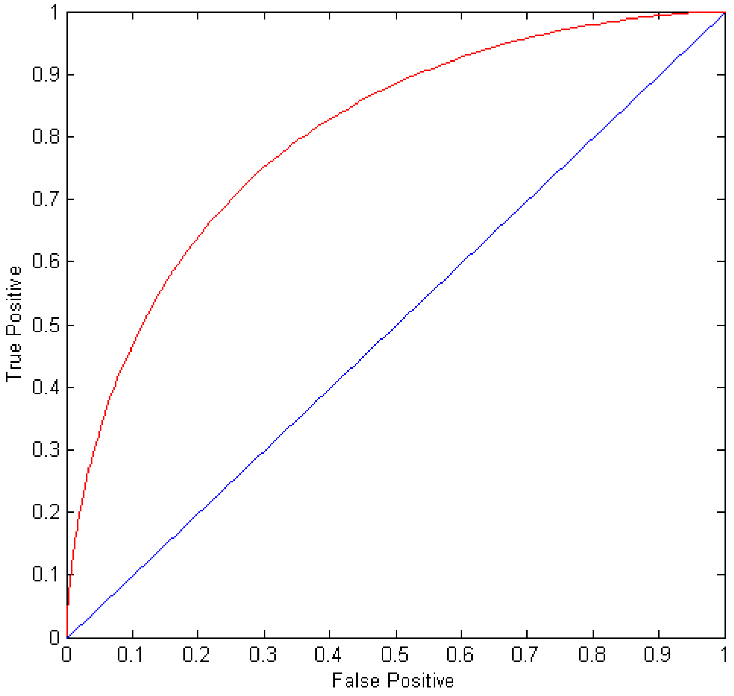
ROC Curves

- TPR=0, FPR=0
  - Model predicts every instance to be a negative class
- TPR=1, FPR=1
  - Model predicts every instance to be a positive class.
- TPR=1, FPR=0
  - The ideal model



ROC Curve

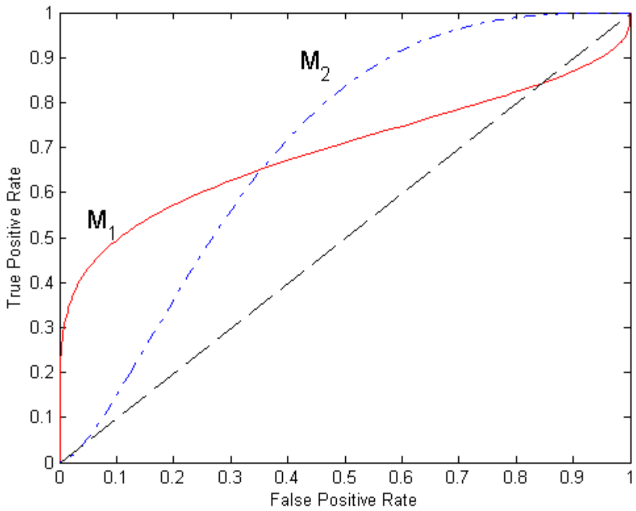
- A good classification model should be as close as possible to the upper left corner.
- Diagonal line:
  - Random guessing
- Good ROC curves: AUC > 0.7
- Diagnostic tests: AUC >0.9

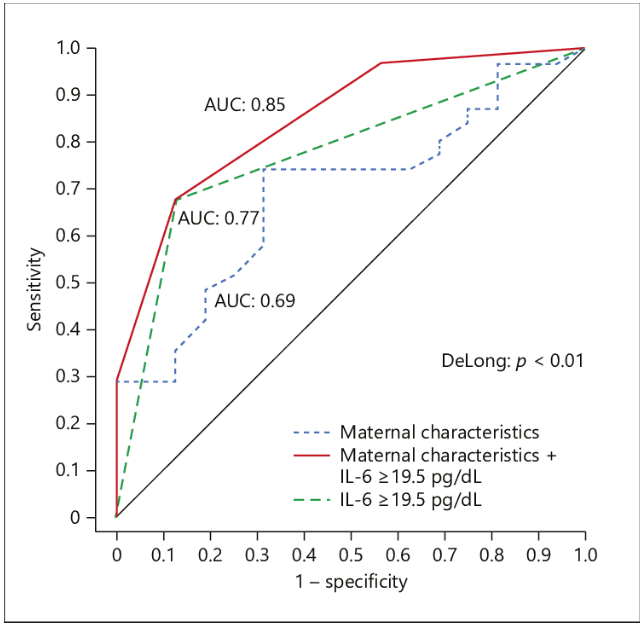


Tradeoff

- Typically, a trend of increasing sensitivity with decreasing specificity
- Choosing the best cut-off point is very important to find a balance between the two
  - Diagnostic tests: sacrifice specificity
  - Reporting purposes: a good balance (e.g. ~75% sensitivity and ~75% specificity)

Using ROC in Model Comparison





- No model consistently outperform the other
  - M1 is better for small FPR
  - M2 is better for large FPR
- Look at Area Under the ROC curve (AUC)
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5

Martinez-Portilla, Raigam Jafet, et al. "Maternal Serum Interleukin-6: A Non-Invasive Predictor of Histological Chorioamnionitis in Women with Preterm-Prelabor Rupture of Membranes." Fetal diagnosis and therapy (2018).