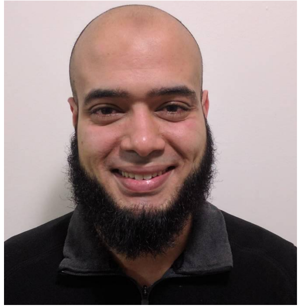
Machine Learning

Evaluation Metrics 1

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / MSc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



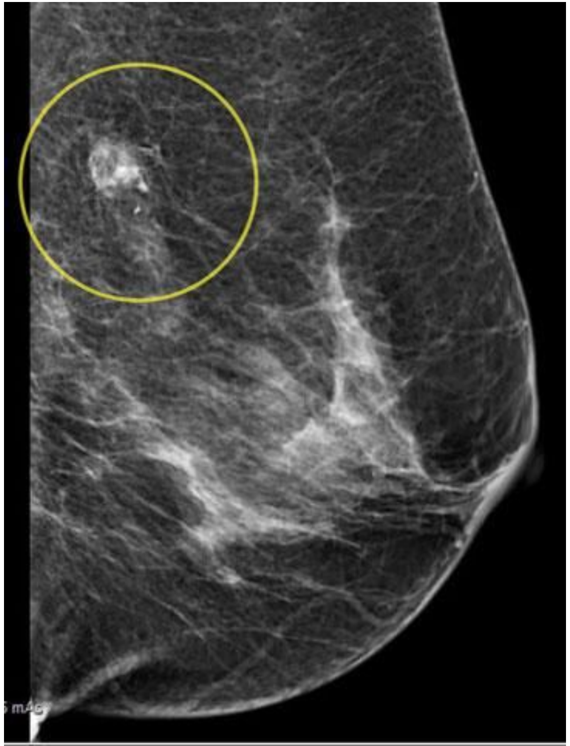
Evaluation Metrics

- Similar to regression, when you train your model (optimize cost function), you may want to evaluate your model against other practical metrics
- This is very critical (and sometimes tricky) in classification
- Let’s see some metrics
- It is very critical to decide what is the **positive class**
  - This depends on the problem
    - Ask yourself: Which case is more critical to report
  - For example an email binary classifier could have spam=True as the positive class
  - Highly recommended to mark the positive class with label 1
  - Being consistent is critical

Question!

- Given a medical image for a woman breast, we would like to know if there is a cancer or not
- Most probably, What is the positive class? Why?
  - 1) has cancer
  - 2) no cancer

- It is critical to report cancer cases
- My opinion, let *has-cancer* be the positive class
- Call the model: Breast Cancer Classifier



Question!

- Given an image for the steering wheel, we would like to analyze if any of the hands on the wheel or not
- Most probably, What is the positive class? Why?
  - 1) hands on wheels
  - 2) hands off wheels

- It depends, but probably from safety perspective to know when the user has his hands off the wheel (to get control)
- So hands-off is the positive class
  - Call the feature: Hands-off wheel detection

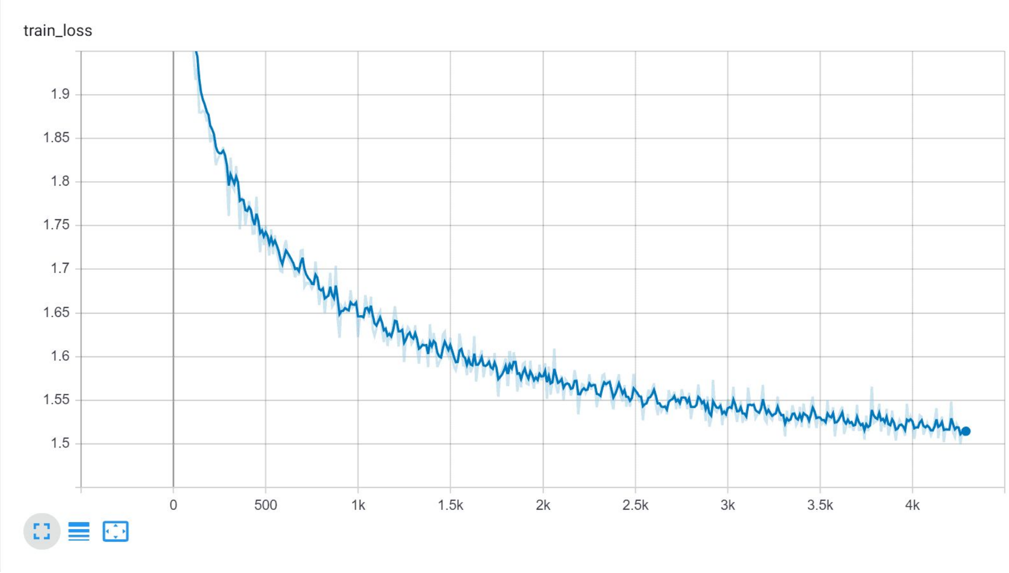
Log-Loss Metric

- This is the metric we used to train the classifier
- It tries to learn the parameters to match the output probability distribution for the examples
- This is a must to monitor metric. The smaller the better

$$logloss(p, t) = -tlog(p) - (1 - t)log(1 - p)$$

Log-Loss Curve

- In long training (like deep learning), it is important to monitor your training
  - Iterations (epoch) vs your loss (Whatever loss type)
- Observe, in real sets, the curve can be zigzag
  - So don’t make decisions based on narrow window. See the big picture



Log-Loss Drawbacks

- Lack of Intuitive Interpretability
  - What is the **meaning** of 0.2235 loss?
  - We just know it is the *average logarithmic* error between the predicted probabilities and the true labels
- Outlier Sensitivity: An extreme outlier with a predicted probability close to zero or one can significantly impact the log loss value
  - log(0) = undefined
- Business alignment: maybe the business metrics have interests that is not aligned (e.g. business target lower FPS [later])
- Sensitivity to Class Imbalance [soon]

Accuracy Metric

- We simply count the total number of accurate predictions out of the total predictions
- As the output is probability for logistic classifier, we need a threshold to convert to 0 or 1
- A higher the threshold will keep reducing positive corrections
- Common threshold is 0.5
  - A high accuracy model with high threshold (e.g. 0.8), gives more confidence in correctness

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Log-Loss vs Accuracy

- Although the 2 models have the **same accuracy**, the model on the right has lower loss value, hence better
  - With a higher **threshold** (e.g. 0.7), the accuracy of the first model drops

Ground Truth	Prob	Logloss -ln(p)	Prediction >= 0.5
1	0.6	0.5	1
1	0.65	0.4	1
1	0.2	1.6	0
0	0.3	0.36	0
0	0.8	1.6	1
		4.46	3/5

Ground Truth	Prob	Logloss	Prediction >= 0.5
1	0.9	0.1	1
1	0.95	0.05	1
1	0.48	0.7	0
0	0.1	0.12	0
0	0.6	0.92	1
		1.89	3/5

Question!

- Our model in a complex problem has 100% accuracy
    - What do you think about the model?
- So weird. Due to noise / complexity of the problems, we don’t get such 100% accuracy!
- Our model has 98% accuracy. Find a case that this results is misleading?
- Imbalance dataset

Imbalanced dataset

- When one or more of the classes has too many labels and while some are very little examples, we call it imbalanced dataset
- Credit card transactions: 99.9% of legitimate transactions and only 0.1% of fraud
  - In 1000,000 examples, only 1000 are fraud
- If the classifier decided just predict EVERYthing as proper, we get accuracy of 99.9%, but this is very **biased** classifier
- Tip: In imbalanced datasets, both **accuracy and log-loss** might be used as indicator if the model is wrong, but shouldn't be used for the opposite
  - 55% accuracy ⇒ bad mode. 99% accuracy ⇒ probably majority focused model

Balanced Accuracy

- Balanced accuracy is better indicator on imbalanced datasets.
    - It the average accuracy for each class
    - So in best case it is ½ (1 + 1) = 1

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right)$$

$$\text{Overall Accuracy} = \frac{(TP + TN)}{(P + N)}$$

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”