Machine Learning (Lecture 6)

UEM/IEM Summer 2018

Logistic Regression: Model Training

- Loss function for Logistic Regression
  - The loss function for linear regression is squared loss.
  - The loss function for logistic regression is Log Loss, which is defined as follows:

$$\text{Log Loss} = \sum_{(x,y)\in D} -y\log(y') - (1-y)\log(1-y')$$

  - where:
    - $(x, y) \in D$ is the data set containing many labeled examples, which (x, y) are pairs.
    - y is the label in a labeled example. Since this is logistic regression, every value of y must either be 0 or 1.
    - y' is the predicted value (somewhere between 0 and 1), given the set of features in x.

Logistic Regression: Model Training

- The equation for Log Loss is closely related to Shannon's Entropy measure from Information Theory.
- It is also the negative logarithm of the likelihood function, assuming a Bernoulli distribution of y.
- Indeed, minimizing the loss function yields a maximum likelihood estimate.

Logistic Regression: Model Training

- Regularization in Logistic Regression
  - Regularization is extremely important in logistic regression modeling.
  - Without regularization, the asymptotic nature of logistic regression would keep driving loss towards 0 in high dimensions.
  - Consequently, most logistic regression models use one of the following two strategies to dampen model complexity:
    - L_2 regularization.
    - Early stopping, that is, limiting the number of training steps or the learning rate.

Logistic Regression: Model Training

- Imagine that you assign a unique id to each example, and map each id to its own feature.
- If you don't specify a regularization function, the model will become completely overfit.
- That's because the model would try to drive loss to zero on all examples and never get there, driving the weights for each indicator feature to +infinity or -infinity.
- This can happen in high dimensional data with feature crosses, when there’s a huge mass of rare crosses that happen only on one example each.
- Fortunately, using L_2 or early stopping will prevent this problem.

Classification: Thresholding

- We will learn how logistic regression can be used for classification tasks, and explores how to evaluate the effectiveness of classification models.
- Logistic regression returns a probability. You can use the returned probability "as is" (for example, the probability that the user will click on this ad is 0.00023) or convert the returned probability to a binary value (for example, this email is spam).

Classification: Thresholding

- A logistic regression model that returns 0.9995 for a particular email message is predicting that it is very likely to be spam.
- Conversely, another email message with a prediction score of 0.0003 on that same logistic regression model is very likely not spam.
- However, what about an email message with a prediction score of 0.6?

Classification: Thresholding

- In order to map a logistic regression value to a binary category, you must define a **classification threshold** (also called the **decision threshold**).
- A value above that threshold indicates "spam"; a value below indicates "not spam."
- It is tempting to assume that the classification threshold should always be 0.5, but thresholds are problem-dependent, and are therefore values that you must tune.

Classification: True vs. False and Positive vs. Negative

- **An Aesop's Fable: The Boy Who Cried Wolf (*compressed*)**

  A shepherd boy gets bored tending the town's flock. To have some fun, he cries out, "Wolf!" even though no wolf is in sight. The villagers run to protect the flock, but then get really mad when they realize the boy was playing a joke on them.

  [Iterate previous paragraph *N* times.]

  One night, the shepherd boy sees a real wolf approaching the flock and calls out, "Wolf!" The villagers refuse to be fooled again and stay in their houses. The hungry wolf turns the flock into lamb chops. The town goes hungry. Panic ensues.

Classification: True vs. False and Positive vs. Negative

- Let's make the following definitions:
  - "Wolf" is a **positive class**.
  - "No wolf" is a **negative class**.
- We can summarize our "wolf-prediction" model using a 2x2 confusion matrix that depicts all four possible outcomes:

True Positive (TP): • Reality: A wolf threatened. • Shepherd said: "Wolf." • Outcome: Shepherd is a hero.	**False Positive (FP):** • Reality: No wolf threatened. • Shepherd said: "Wolf." • Outcome: Villagers are angry at shepherd for waking them up.
False Negative (FN): • Reality: A wolf threatened. • Shepherd said: "No wolf." • Outcome: The wolf ate all the sheep.	**True Negative (TN):** • Reality: No wolf threatened. • Shepherd said: "No wolf." • Outcome: Everyone is fine.

Classification: True vs. False and Positive vs. Negative

- A **true positive** is an outcome where the model *correctly* predicts the *positive* class.
- Similarly, a **true negative** is an outcome where the model *correctly* predicts the *negative* class.
- A **false positive** is an outcome where the model *incorrectly* predicts the *positive* class.
- And a **false negative** is an outcome where the model *incorrectly* predicts the *negative* class.

Classification: Accuracy

- Accuracy is one metric for evaluating classification models.
- Informally, **accuracy** is the fraction of predictions our model got right. Formally, accuracy has the following definition:

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

- For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

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

  - Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

Classification: Accuracy

- Let's try calculating accuracy for the following model that classified 100 tumors as malignant (the positive class) or benign (the negative class):

True Positive (TP): • Reality: Malignant • ML model predicted: Malignant • **Number of TP results: 1**	**False Positive (FP):** • Reality: Benign • ML model predicted: Malignant • **Number of FP results: 1**
False Negative (FN): • Reality: Malignant • ML model predicted: Benign • **Number of FN results: 8**	**True Negative (TN):** • Reality: Benign • ML model predicted: Benign • **Number of TN results: 90**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1 + 90}{1 + 90 + 1 + 8} = 0.91$$

- Accuracy comes out to 0.91, or 91% (91 correct predictions out of 100 total examples). That means our tumor classifier is doing a great job of identifying malignancies, right?

Classification: Accuracy

- Let's do a closer analysis of positives and negatives to gain more insight into our model's performance.
- Of the 100 tumor examples, 91 are benign (90 TNs and 1 FP) and 9 are malignant (1 TP and 8 FNs).
- Of the 91 benign tumors, the model correctly identifies 90 as benign. That's good.
- However, of the 9 malignant tumors, the model only correctly identifies 1 as malignant—a terrible outcome, as 8 out of 9 malignancies go undiagnosed!

Classification: Accuracy

- While 91% accuracy may seem good at first glance, another tumor-classifier model that always predicts benign would achieve the exact same accuracy (91/100 correct predictions) on our examples.
- In other words, our model is no better than one that has zero predictive ability to distinguish malignant tumors from benign tumors.
- Accuracy alone doesn't tell the full story when you're working with a **class-imbalanced data set**, like this one, where there is a significant disparity between the number of positive and negative labels.

Classification: Precision and Recall

- Precision
  - **Precision** attempts to answer the following question:

    What proportion of positive identifications was actually correct?
  - Precision is defined as follows:

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

  - Let's calculate precision for our ML model from the previous example that analyzes tumors:

True Positives (TPs): 1	False Positives (FPs): 1
False Negatives (FNs): 8	True Negatives (TNs): 90

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{1}{1 + 1} = 0.5$$

    - Our model has a precision of 0.5—in other words, when it predicts a tumor is malignant, it is correct 50% of the time.

Classification: Precision and Recall

- Recall
  - Recall attempts to answer the following question:

    What proportion of actual positives was identified correctly?
  - Mathematically, recall is defined as follows:

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

  - Let's calculate recall for our tumor classifier:

True Positives (TPs): 1	False Positives (FPs): 1
False Negatives (FNs): 8	True Negatives (TNs): 90

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{1}{1 + 8} = 0.11$$

  - Our model has a recall of 0.11—in other words, it correctly identifies 11% of all malignant tumors.

Classification: A Tug of War

- To fully evaluate the effectiveness of a model, you must examine **both** precision and recall.
- Unfortunately, precision and recall are often in tension. That is, improving precision typically reduces recall and vice versa.

Classification: A Tug of War

- The following figure shows 30 predictions made by an email classification model. Those to the right of the classification threshold are classified as "spam", while those to the left are classified as "not spam."

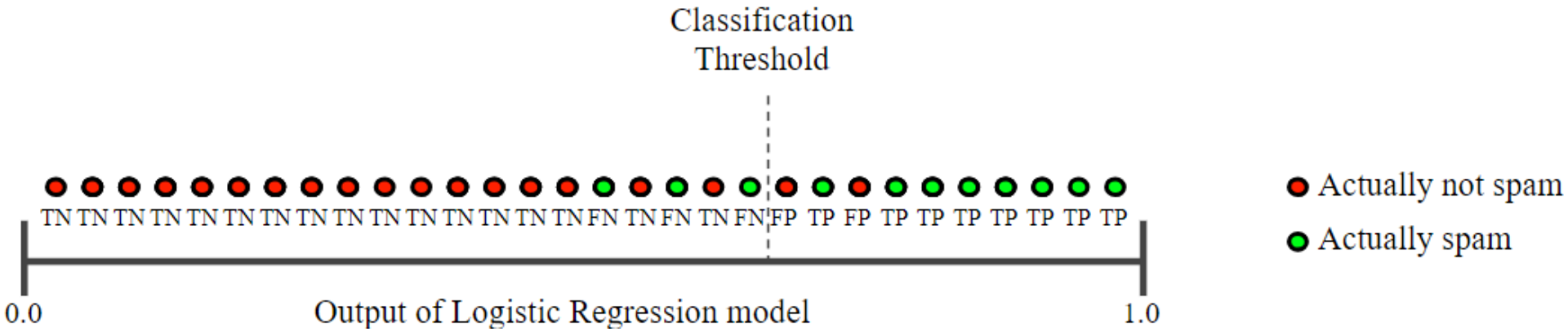

Figure 1. Classifying email messages as spam or not spam.

Classification: A Tug of War

- Let's calculate precision and recall based on the results shown in Figure 1:

True Positives (TP): 8	False Positives (FP): 2
False Negatives (FN): 3	True Negatives (TN): 17

- Precision measures the percentage of **emails flagged as spam** that were correctly classified—that is, the percentage of dots to the right of the threshold line that are green in Figure 1:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{8}{8 + 2} = 0.8$$

Classification: A Tug of War

- Recall measures the percentage of **actual spam emails** that were correctly classified—that is, the percentage of green dots that are to the right of the threshold line in Figure 1:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{8}{8 + 3} = 0.73$$

Classification: A Tug of War

- Figure 2 illustrates the effect of increasing the classification threshold.

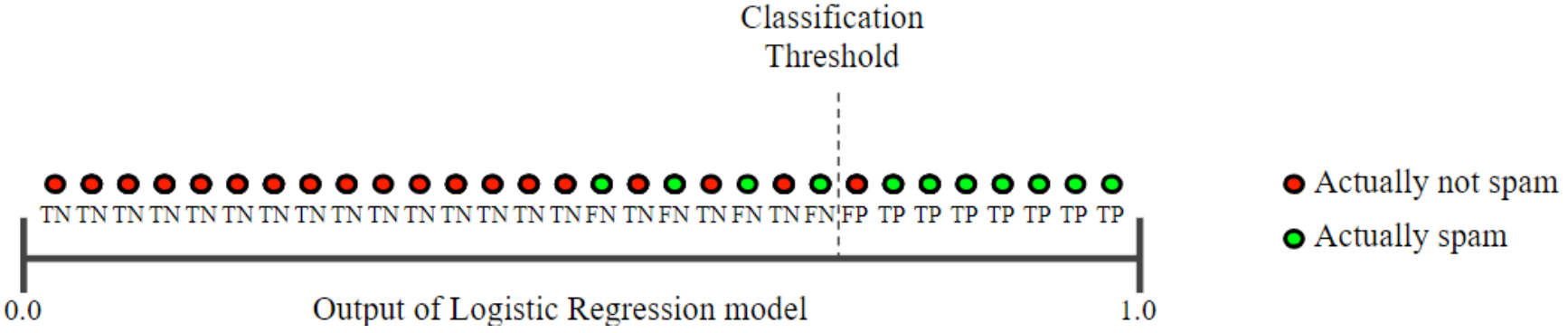

Figure 2. Increasing classification threshold.

- The number of false positives decreases, but false negatives increase. As a result, precision increases, while recall decreases:

True Positives (TP): 7	False Positives (FP): 1
False Negatives (FN): 4	True Negatives (TN): 18

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{7}{7+1} = 0.88$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{7}{7+4} = 0.64$$

Classification: A Tug of War

- Conversely, Figure 3 illustrates the effect of decreasing the classification threshold (from its original position in Figure 1).

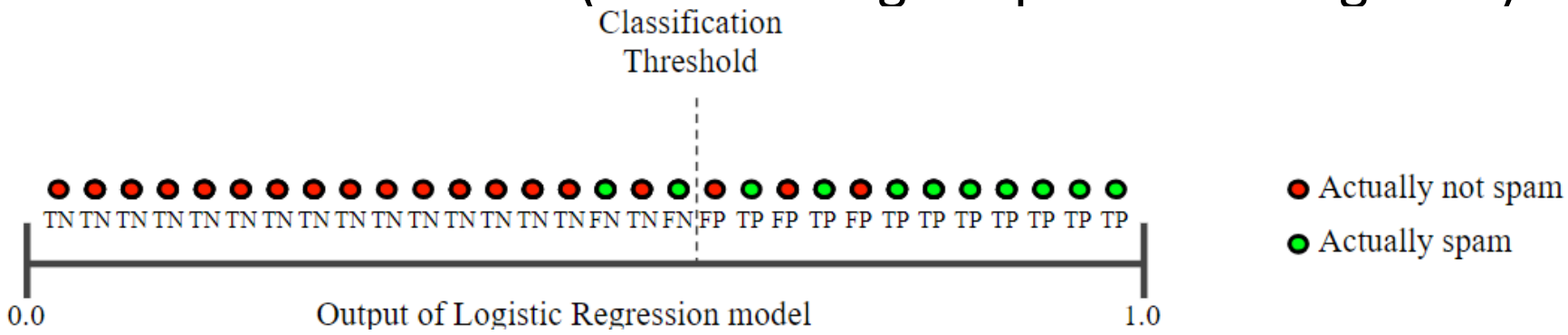

Figure 3. Decreasing classification threshold.

- False positives increase, and false negatives decrease. As a result, this time, precision decreases and recall increases:

True Positives (TP): 9	False Positives (FP): 3
False Negatives (FN): 2	True Negatives (TN): 16

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{9}{9 + 3} = 0.75$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{9}{9 + 2} = 0.82$$

Quiz: Accuracy

- In which of the following scenarios would a high accuracy value suggest that the ML model is doing a good job?

  - An expensive robotic chicken crosses a very busy road a thousand times per day. An ML model evaluates traffic patterns and predicts when this chicken can safely cross the street with an accuracy of 99.99%.
  - A deadly, but curable, medical condition afflicts .01% of the population. An ML model uses symptoms as features and predicts this affliction with an accuracy of 99.99%.
  - In the game of roulette, a ball is dropped on a spinning wheel and eventually lands in one of 38 slots. Using visual features (the spin of the ball, the position of the wheel when the ball was dropped, the height of the ball over the wheel), an ML model can predict the slot that the ball will land in with an accuracy of 4%.

Quiz: Accuracy

- In which of the following scenarios would a high accuracy value suggest that the ML model is doing a good job?

  - An expensive robotic chicken crosses a very busy road a thousand times per day. An ML model evaluates traffic patterns and predicts when this chicken can safely cross the street with an accuracy of 99.99%.
  - A deadly, but curable, medical condition afflicts .01% of the population. An ML model uses symptoms as features and predicts this affliction with an accuracy of 99.99%.
  - In the game of roulette, a ball is dropped on a spinning wheel and eventually lands in one of 38 slots. Using visual features (the spin of the ball, the position of the wheel when the ball was dropped, the height of the ball over the wheel), an ML model can predict the slot that the ball will land in with an accuracy of 4%.

Quiz: Precision

- Consider a classification model that separates email into two categories: "spam" or "not spam." If you raise the classification threshold, what will happen to precision?

  - Definitely decrease.
  - Probably increase.
  - Probably decrease.
  - Definitely increase.

Quiz: Precision

- Consider a classification model that separates email into two categories: "spam" or "not spam." If you raise the classification threshold, what will happen to precision?

  - Definitely decrease.
  - Probably increase.
  - Probably decrease.
  - Definitely increase.

Quiz: Recall

- Consider a classification model that separates email into two categories: "spam" or "not spam." If you raise the classification threshold, what will happen to recall?

  - Always decrease or stay the same.
  - Always stay constant.
  - Always increase.

Quiz: Recall

- Consider a classification model that separates email into two categories: "spam" or "not spam." If you raise the classification threshold, what will happen to recall?

  - Always decrease or stay the same.
  - Always stay constant.
  - Always increase.

Classification: ROC and AUC

- ROC curve
  - An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:
    - True Positive Rate
    - False Positive Rate
  - **True Positive Rate** (**TPR**) is a synonym for recall and is therefore defined as follows:

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

Classification: ROC and AUC

- **False Positive Rate** (**FPR**) is defined as follows:

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

- An ROC curve plots TPR vs. FPR at different classification thresholds.
- Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

Classification: ROC and AUC

- The following figure shows a typical ROC curve.

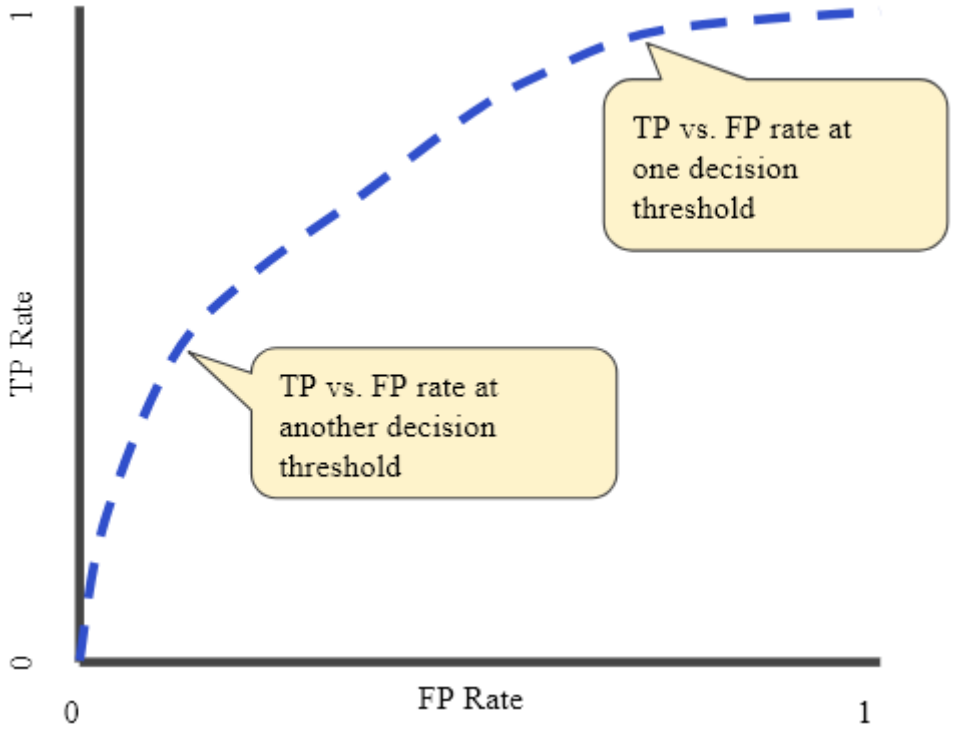

Figure 4. TP vs. FP rate at different classification thresholds.

- To compute the points in an ROC curve, we could evaluate a logistic regression model many times with different classification thresholds, but this would be inefficient. Fortunately, there's an efficient, sorting-based algorithm that can provide this information for us, called AUC.

Classification: ROC and AUC

- AUC: Area Under the ROC Curve
  - **AUC** stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).

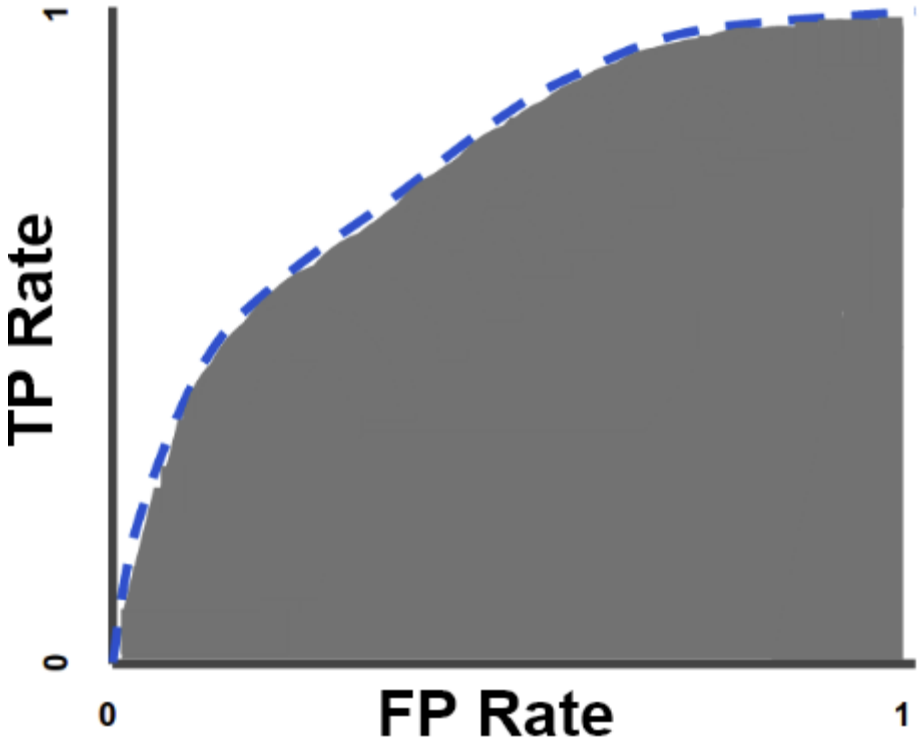

Figure 5. AUC (Area under the ROC Curve).

Classification: ROC and AUC

- AUC provides an aggregate measure of performance across all possible classification thresholds.
- One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example.
- For example, given the following examples, which are arranged from left to right in ascending order of logistic regression predictions:

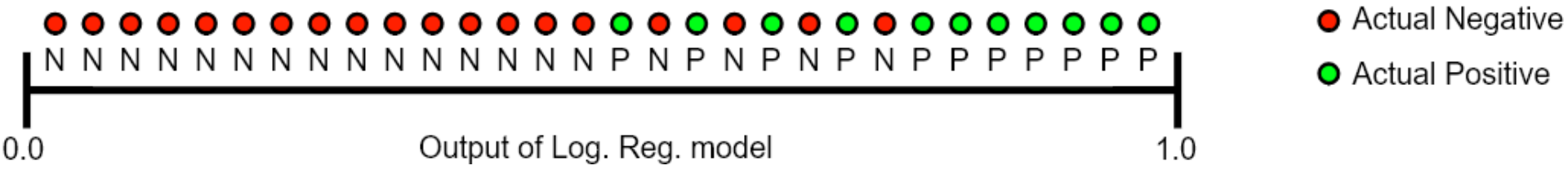

Figure 6. Predictions ranked in ascending order of logistic regression score.

Classification: ROC and AUC

- AUC represents the probability that a random positive (green) example is positioned to the right of a random negative (red) example.
- AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.
- AUC is desirable for the following two reasons:
  - AUC is **scale-invariant**. It measures how well predictions are ranked, rather than their absolute values.
  - AUC is **classification-threshold-invariant**. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

Classification: ROC and AUC

- However, both these reasons come with caveats, which may limit the usefulness of AUC in certain use cases:
  - **Scale invariance is not always desirable.** For example, sometimes we really do need well calibrated probability outputs, and AUC won’t tell us about that.
  - **Classification-threshold invariance is not always desirable.** In cases where there are wide disparities in the cost of false negatives vs. false positives, it may be critical to minimize one type of classification error. For example, when doing email spam detection, you likely want to prioritize minimizing false positives (even if that results in a significant increase of false negatives). AUC isn't a useful metric for this type of optimization.

Reference

- This lecture note has been developed based on the machine learning crash course at Google, which is under *Creative Commons Attribution 3.0 License*.