12 | Logistic Regression

Ivan Corneillet

Data Scientist

Learning Objectives

After this lesson, you should be able to:

- Build a logistic regression classification model using *sklearn*
- Describe the logit and sigmoid functions, odds and odds ratios, and how they relate to logistic regression
- Evaluate a model using metrics such as classification accuracy/error
- Evaluate a binary classification model using advanced metrics such as confusion matrix, ROC, and AUC curves
- Explain the trade-offs between false positives and false negatives

Logistic Regression

Logistic Regression is a binary classifier. But what's binary classification?

- Binary classification is the simplest form of classification
  - I.e., the response is a *boolean* value (true/false)
- Many classification problems are binary in nature
  - E.g., we may be using patient data (medical history) to predict whether a patient smokes or not

- At first, many problems don't appear to be binary; however, you can usually transform them into binary problems
  - E.g., what if you are predicting whether an image is of a *"human"*, *"dog"*, or *"cat"*?
  - You can transform this non-binary problem into three binary problems
    - 1. Will it be *"human"* or *"not human"*?
    - 2. Will it be *"dog"* or *"not dog"*?
    - 3. Will it be *"cat"* or *"not cat"*?
- This is similar to the concept of binary variables

Why is logistic regression so valuable to know?

- It addresses many commercially valuable classification problems, such as:
  - Fraud detection (e.g., payments, e-commerce)
  - Churn prediction (marketing)
  - Medical diagnoses (e.g., is the test positive or negative?)
  - and many, many others…

Logistic Regression

"Retrofitting" linear regression into logistic regression

Logistic Regression

- By putting together $\hat{y} = X \cdot \hat{\beta}$ and $\hat{p} = \pi(\hat{y}) = \frac{1}{1+e^{-\hat{y}}}$, we get

$$\hat{p} = \frac{1}{1 + e^{-X \cdot \hat{\beta}}}$$

or

$$log\left(\frac{\hat{p}}{1 - \hat{p}}\right) = X \cdot \hat{\beta}$$

- Finally, probabilities are “snapped” to class labels (e.g., by thresholding at the 50% level)

Logistic Regression

Interpreting the logistic regression coefficients

Interpreting the logistic regression coefficients

- With linear regressions, $\hat{\beta}_j$ represents the change in y for a change in unit of x_j

$$ln\left(\frac{\hat{p}}{1-\hat{p}}\right) = X \cdot \hat{\beta} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \cdots + \hat{\beta}_k \cdot x_k$$

- With logistic regressions, $\hat{\beta}_j$ represents the **log-odds** change in c for a change in unit of x_j

- This also means that $e^{\hat{\beta}_j}$ represents the multiplier change in **odds** in c for a change in unit of x_j

$$\frac{\widehat{odds}(x_j+1)}{\widehat{odds}(x_j)} = \frac{e^{\hat{y}(x_j+1)}}{e^{\hat{y}(x_j)}} = e^{\hat{y}(x_j+1)-\hat{y}(x_j)} = e^{(\cancel{\boxtimes}+\hat{\beta}_j \cancel{\cdot x_j}+\cancel{\otimes})-(\cancel{\boxtimes}+\hat{\beta}_j\cdot(\cancel{x_j}+1)+\cancel{\otimes})} = e^{\hat{\beta}_j}$$

Logistic Regression

Pros and Cons

Logistic Regression | Pros and cons

- Pros
  - Fit is fast
  - Output is a (posterior) probability which is easy to interpret

- Cons
  - Limited to binary classification (but *sklearn* provides a multiclass implementation; use ensemble under the hood)

Confusion Matrix

Confusion Matrix (a.k.a., Contingency Table or Error Matrix)

		True Class	
		1	0
Hypothesized Class	1	**True Positives** (TP)	**False Positives** (FP) *(type I error)*
	0	**False Negatives** (FN) *(type II error)*	**True Negatives** (TN)
Total Columns		$P = TP + FN$	$N = FP + TN$

- A confusion matrix is a specific table layout that allows visualization of the performance of a supervised learning algorithm
- Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class
- The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e., commonly mislabeling one as another)

Interpreting the Confusion Matrix

		True Class	
		1	0
Hypothesized Class	1 ●	● **True Positives** (TP)	● **False Positives** (FP) *(type I error)*
	0 ■	■ **False Negatives** (FN) *(type II error)*	■ **True Negatives** (TN)
Total Columns		$P = TP + FN$	$N = FP + TN$

Overall, how often is the classifier correct? $\frac{TP+TN}{P+N}$ *(accuracy)*	*Overall, how often is the classifier wrong?* $\frac{FP+FN}{P+N}$ *(misclassification rate)*
When the classifier predicts yes, how often is it correct? $\frac{TP}{TP+FP}$ *(precision)*	*When it's actually yes, how often does the classifier predict yes?* $\frac{TP}{P}$ *(TPR, sensitivity, recall)*
When it's actually no, how often does the classifier predict yes? $\frac{FP}{N}$ *(FPR, fall-out)*	*When it's actually no, how often does it predict no?* $\frac{TN}{N}$ *(specificity)*
How often does the yes condition actually occur in our sample? $\frac{P}{P+N}$ *(prevalence)*	

True and False Positive Rates

True Positive Rate, $TPR = \frac{TP}{P}$

		True Class 1	True Class 0
Hypothesized Class	**1**	**True Positives** (TP)	**False Positives** (FP) *(type I error)*
	0	**False Negatives** (FN) *(type II error)*	**True Negatives** (TN)
Total Columns		$P = TP + FN$	$N = FP + TN$

- When it's actually yes, how often does the classifier predict yes?
- A.k.a., "Sensitivity"

  - E.g., given a medical exam that tests for cancer, how often does it correctly identify patients with cancer?
  - Likewise, this can be inverted: how often does a test *correctly* identify patients without cancer

False Positive Rate, $FPR = \frac{FP}{N}$

		True Class 1	True Class 0
Hypothesized Class	1	True Positives (TP)	False Positives (FP) *(type I error)*
	0	False Negatives (FN) *(type II error)*	True Negatives (TN)
Total Columns		$P = TP + FN$	$N = FP + TN$

- When it's actually no, how often does the classifier predict yes?
- A.k.a., "Fall-out"

  - E.g., given a medical exam that tests for cancer, how often does it trigger a "false alarm" by saying a patient has cancer when they actually don't?
  - Likewise, this can be also inverted: how often does a test *incorrectly* identify patients as being cancer-free when they might actually have cancer!

True Positive and False Positive Rates

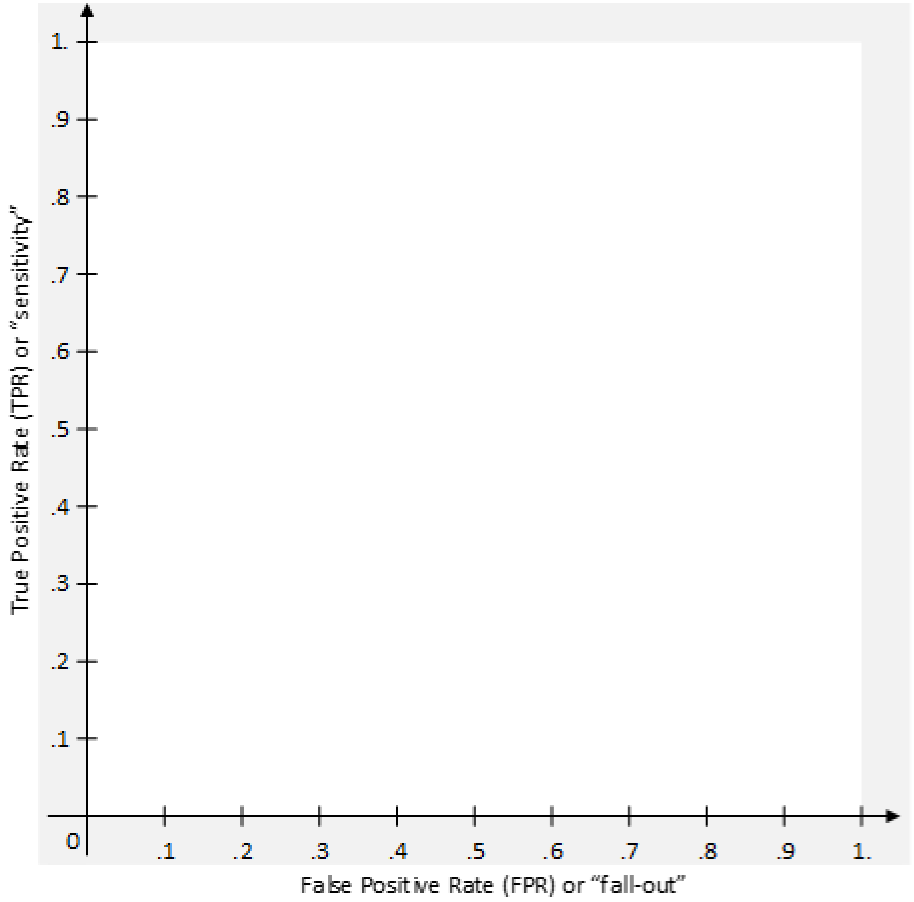
- We can split up the accuracy of each label by using true positive and false positive rates. Using them, we can get a much clearer picture of where predictions begin to fall apart
- A good classifier would have a true positive rate approaching 1, and a false positive rate approaching 0. In a binary problem (say, predicting if someone smokes or not), it would accurately predict all of the smokers as smokers, and not accidentally predict any of the non-smokers as smokers

ROC and AUC

ROC (receiver operating characteristic or relative operating characteristic) and AUC (Area Under the Curve)

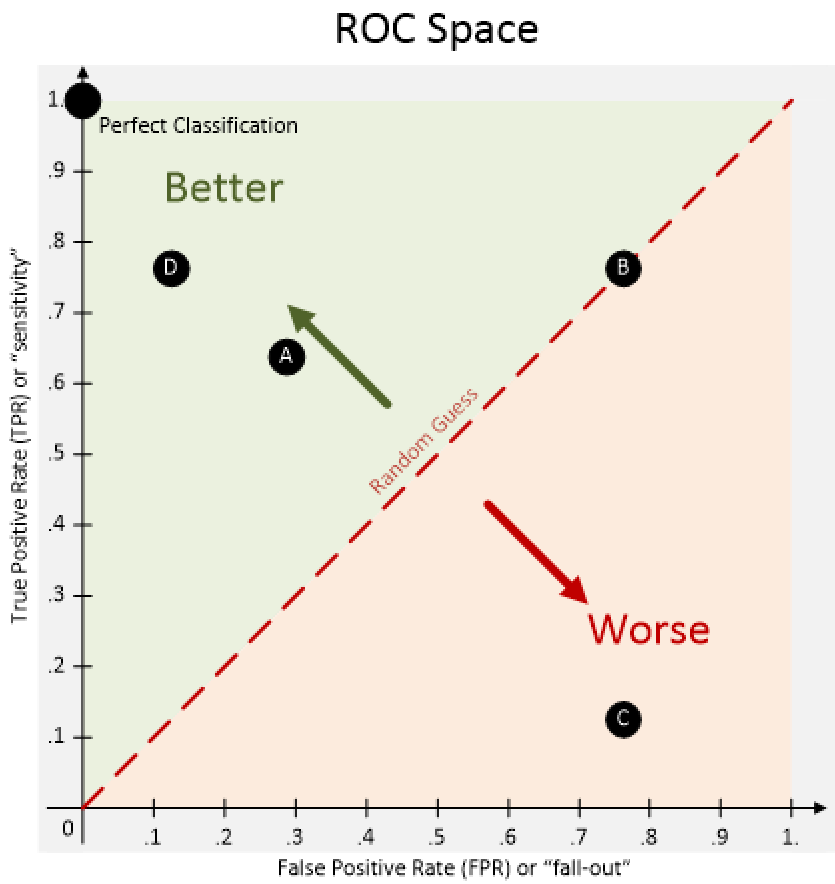
ROC (receiver operating characteristic) curve (a.k.a., relative operating characteristic curve)

- An ROC curve plots the true positive rate (TPR) (or “sensitivity”) against the false positive rate (FPR) (or “fall-out”) at various threshold settings to illustrate the performance of a binary classifier system
- The ROC curve is thus the sensitivity as a function of fall-out



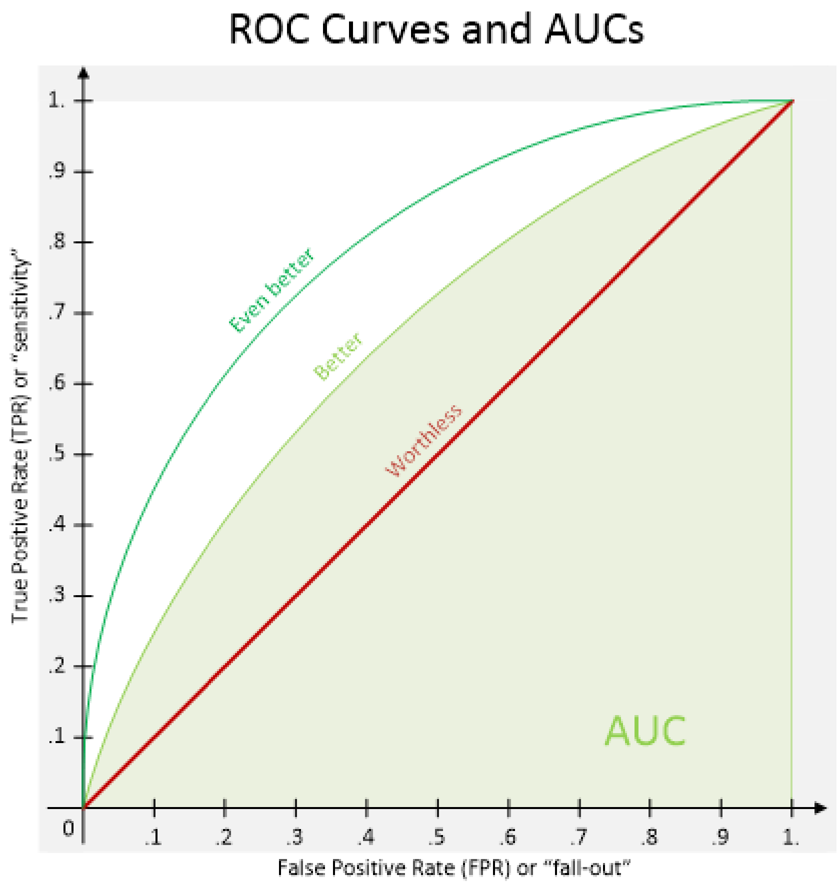
ROC curves demonstrate several things:

- It shows the tradeoff between sensitivity and fall-out (any increase in sensitivity will be accompanied by an increase in fallout)
  - The closer the **points** are in the left-hand border and then the top border of the ROC space, the more accurate the classifier is
  - The closer the **points** come to the 45-degree diagonal of the ROC space, the less accurate the classifier is



ROC curves demonstrate several things: (cont.)

- The area under the curve (AUC) is a measure of classifier accuracy
  - The closer the **curve** follows the left-hand border and then the top border of the ROC space, the more accurate the classifier is
  - The closer the **curve** comes to the 45-degree diagonal of the ROC space, the less accurate the classifier is



Plotting an ROC curve

- ❶ Discard $\hat{c}$ (hypothesized class) and whether it is a true/false positive/negative
- ❷ Order the trained sample by their decreasing hypothesized probabilities $\hat{p}$ (from more confident to have a '1' down to less confident to have a '1')
- ❸ Discard the original ranking from the dataset as well as $\hat{p}$
- ❹ Start at $(0, 0)$
- ❺ For each training sample in the sorted order
  - If $c = 1$, move up by $^1/_P$
  - If $c = 0$, move up by $^1/_N$
- ❻ If not already at $(1, 1)$, go all the way to the right, then up all the way to $(1, 1)$

Let’s plot the ROC for the following trained binary classifier

EXAMPLE

#	$\hat{p}$	$\hat{c}$	c	*True/False Positive/Negative*
1	.44	0	1	FN
2	.29	0	0	TN
3	.98	1	1	TP
4	.69	1	0	FP
5	.07	0	1	FN

Plotting an ROC curve (cont.)

- Notes
  - We don't rely on a threshold (e.g., .5) for plotting ROC curves. Indeed, moving up or right is independent of $\hat{p}$ (we discarded it in step ❸) and only relies on a decreasing ranking of $\hat{p}$ and then c
  - As a matter of fact, you can use ROC curves to select the best threshold