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, as well as how they relate to logistic regression
- Evaluate a model using metrics such as classification accuracy/error

Here’s what’s happening today:

- Logistic Regression
  - How logistic regression relates to linear regression
  - “Retrofitting” linear regression into logistic regression
  - Interpreting the logistic regression coefficients

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”*?
    - 2. 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

How logistic regression relates to linear regression

Logistic regression is a generalization of the linear regression model to classification problems

- The name is somewhat misleading
  - "Regression" comes from fact that we fit a linear model to the feature space
  - But it is really a technique for classification, not regression

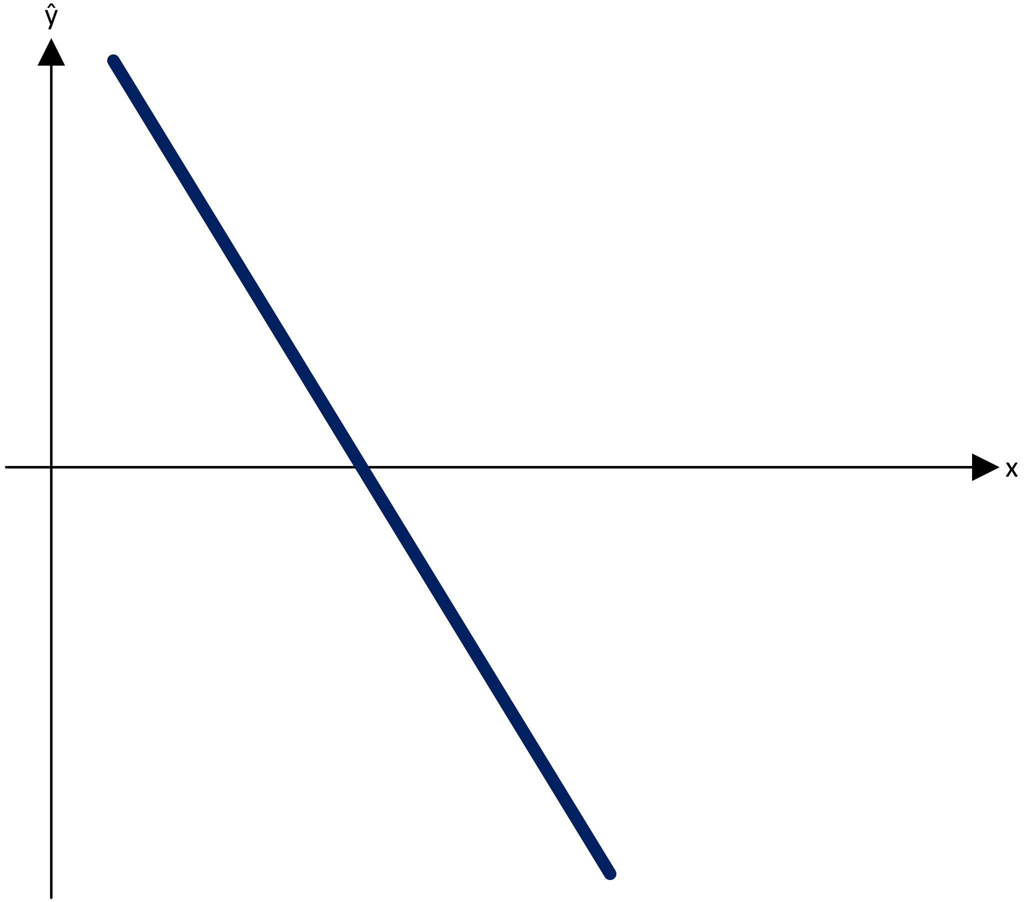
- We use a linear model, similar to linear regression, in order to solve if an item *belongs* or *does not* belong to a class model
  - It is a binary classification technique: $y = \{0, 1\}$
  - Our goal is to classify correctly two types of examples:
    - Class 0, labeled as 0, e.g., *"belongs"*
    - Class 1, labeled as 1, e.g., *"does not belong"*

With linear regression, $\hat{y}$ is in $]-\infty;\ +\infty[$, not $[0; 1]$. How do we fix this for logistic regression?

- The key variable in any regression problem is the outcome variable $\hat{y}$ given the covariate x

$$\hat{y} = X \cdot \hat{\beta}$$

- With linear regression, $\hat{y}$ takes values in $]-\infty;\ +\infty[$
- However, with logistic regression, $\hat{y}$ takes values in the unit interval $[0; 1]$



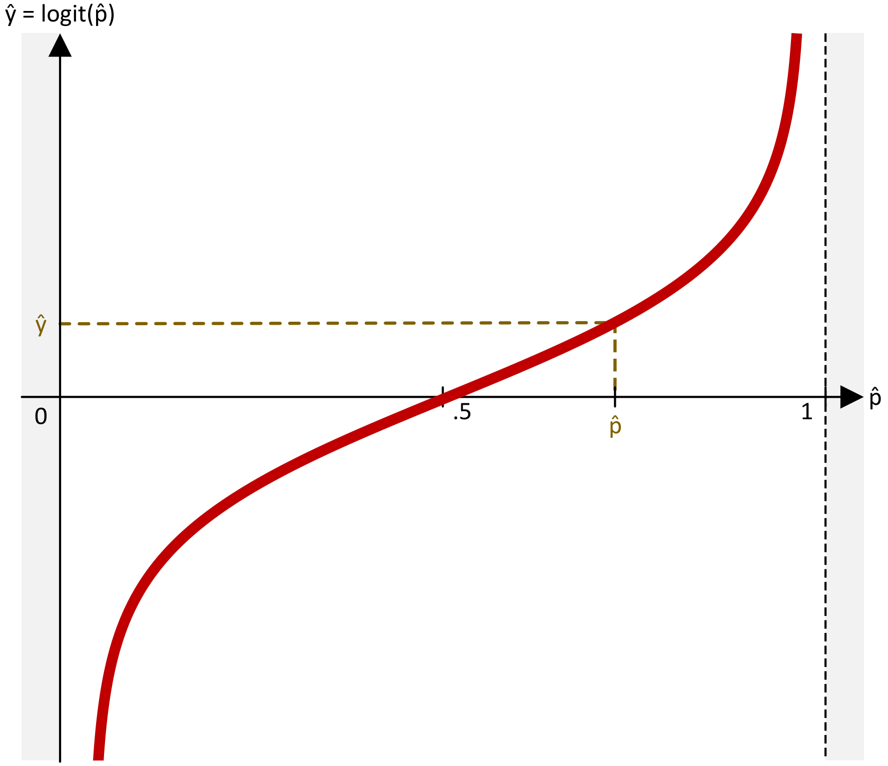
Logistic Regression

“Retrofitting” linear regression into logistic regression

We “retrofit” linear regression in logistic regression with a transformation called the *logit* function (a.k.a., the *log-odds* function) and its inverse, the *logistic* function (a.k.a., *sigmoid* function)

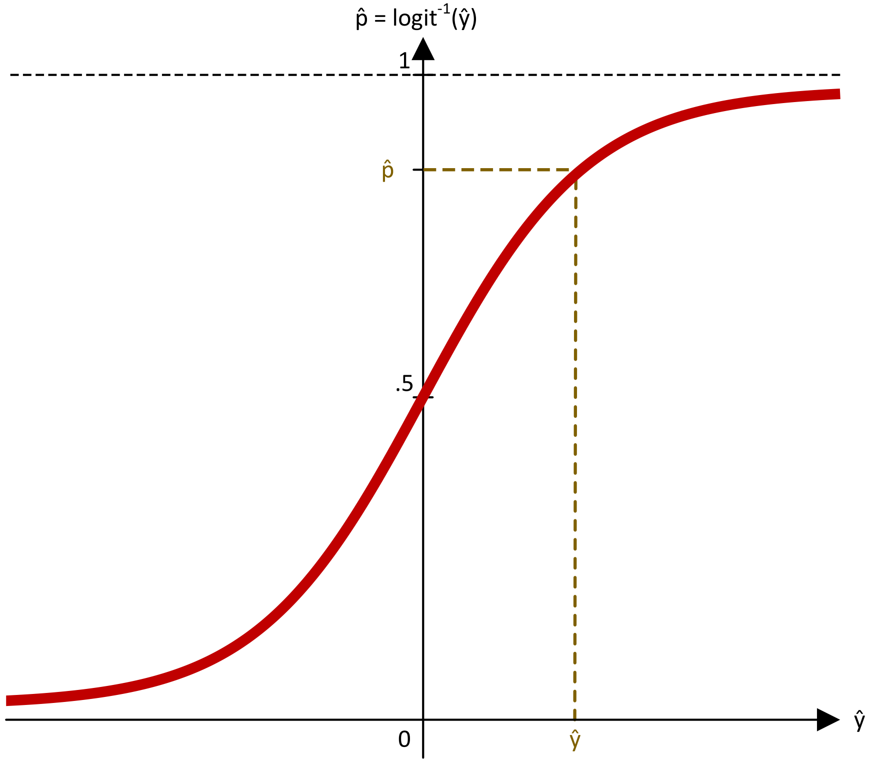
$logit$ maps $\hat{p}$ ($[0;1]$) to $\hat{y}$ ($]-\infty;\ +\infty[$)

$$logit(\hat{p}) = ln\left(\frac{\hat{p}}{1-\hat{p}}\right) = \hat{y}$$



$\pi = logit^{-1}$ maps $\hat{y}$ ($]-\infty;\ +\infty[$) to $\hat{p}$ ($[0;1]$)

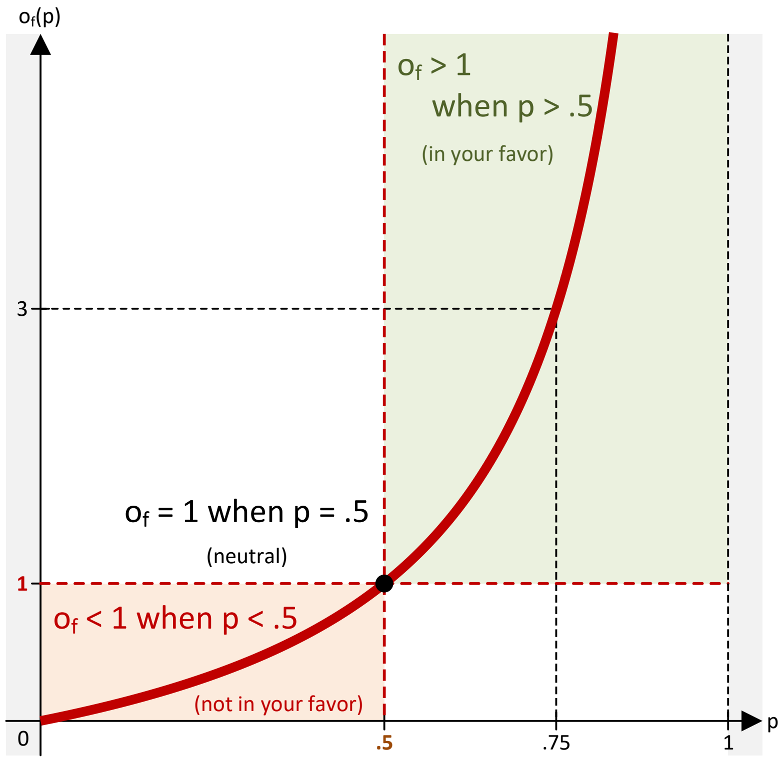
$$\pi(\hat{y}) = \frac{e^{\hat{y}}}{e^{\hat{y}}+1} = \hat{p}$$

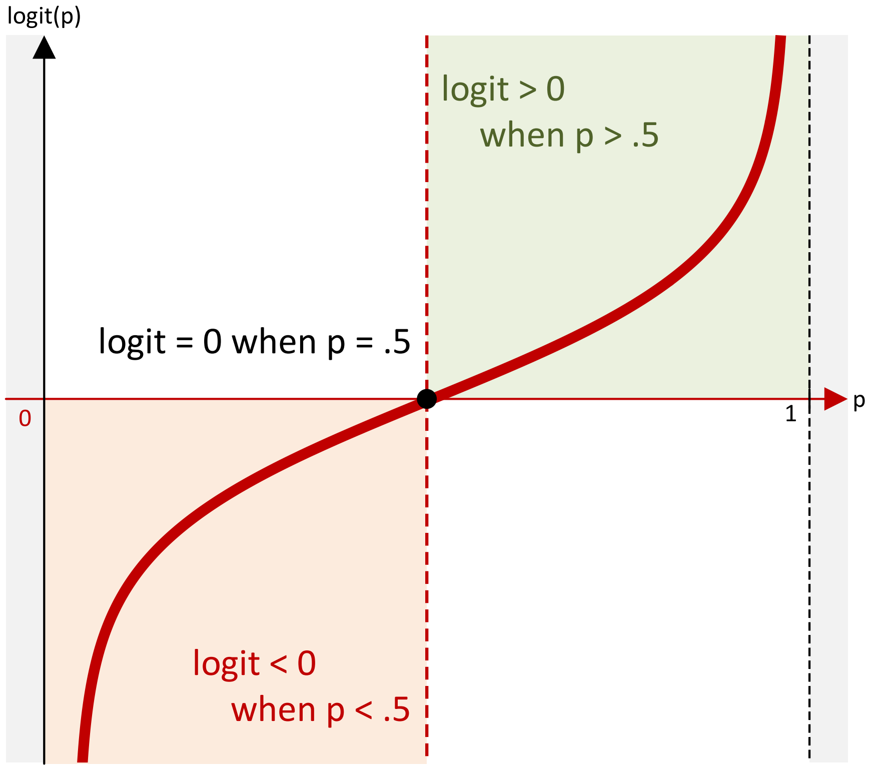


Why is the *logit* function also called the *log-odds* function?

$$\underbrace{o_f}_{odds\ (in\ favor)} = \frac{\overbrace{p}^{probability\ that\ the\ event\ (with\ probability\ p)\ happens}}{\underbrace{1-p}_{probability\ that\ the\ event\ \textbf{does not happen}}}$$

$$logit(p) = ln(o_f) = ln\left(\frac{p}{1-p}\right)$$



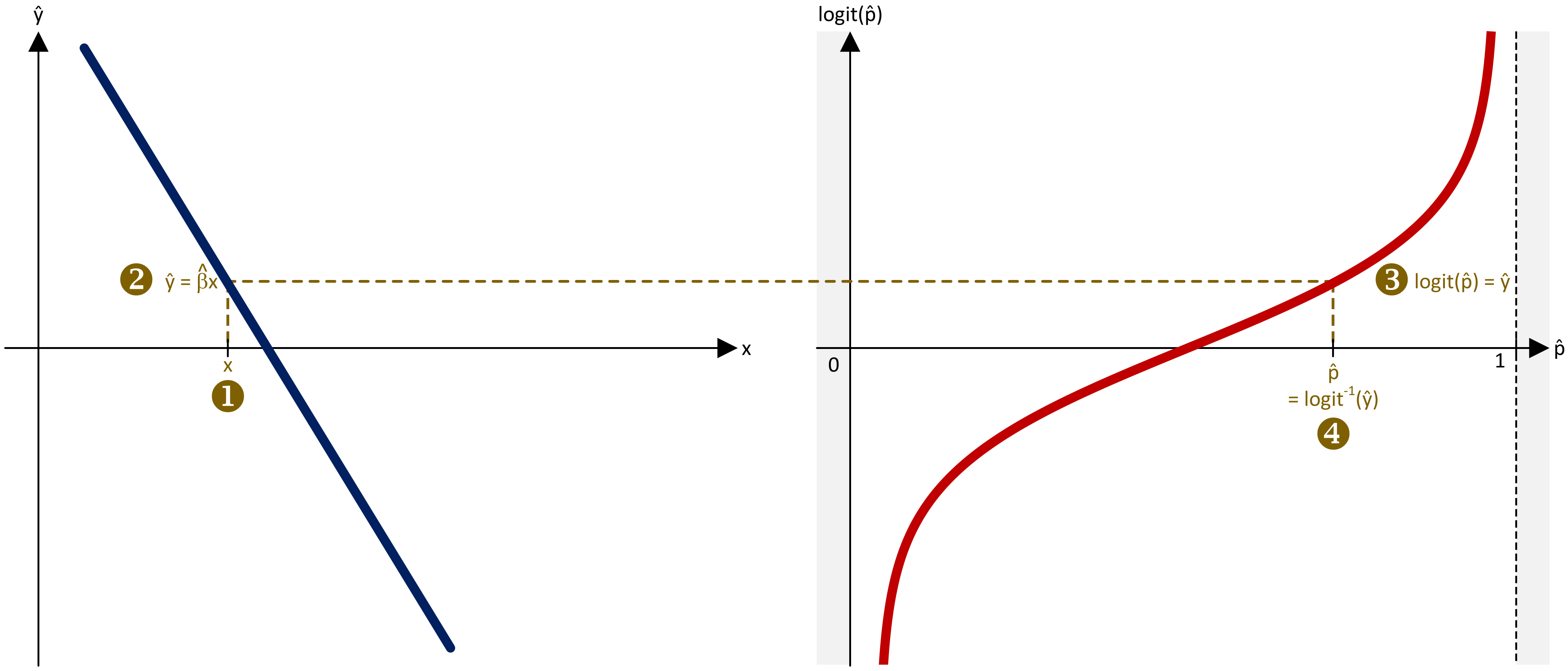


Logistic Regression

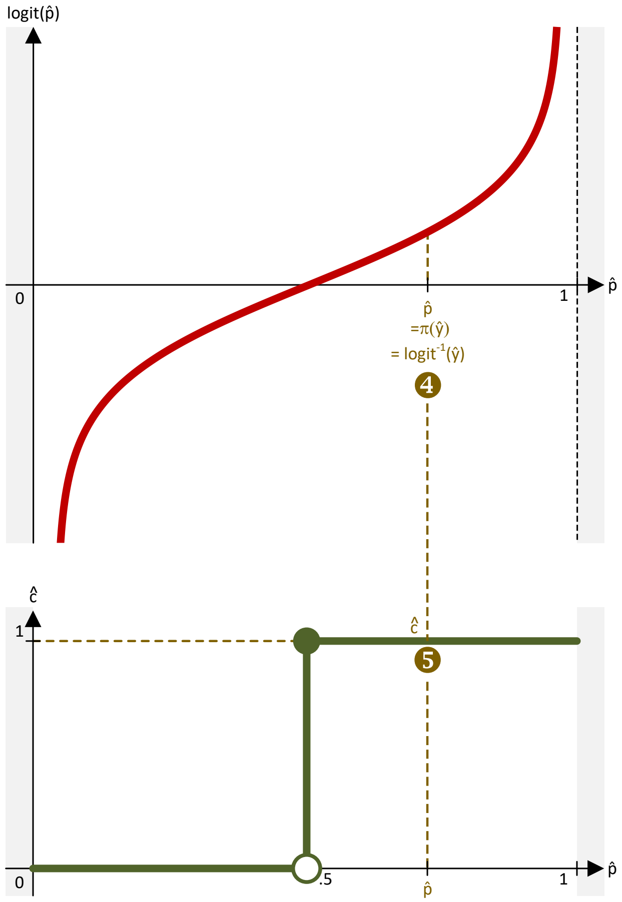
- Putting together $\hat{y} = X \cdot \hat{\beta}$ and $\hat{p} = \pi(\hat{y})$ (really, mapping $\hat{y}$ back to $\hat{p}$), we get

$$\hat{p} = \pi\left(X \cdot \hat{\beta}\right) = \frac{e^{X \cdot \widehat{\beta}}}{e^{\widehat{\beta} x} + 1} = \frac{1}{1 + e^{-X \cdot \widehat{\beta}}}$$

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



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 $e^{\widehat{\beta}_j}$ represents the multiplier change in **odds** in c for a change in unit of x_j

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

Activity | Interpreting the logistic regression coefficients

EXERCISE

DIRECTIONS (5 minutes)

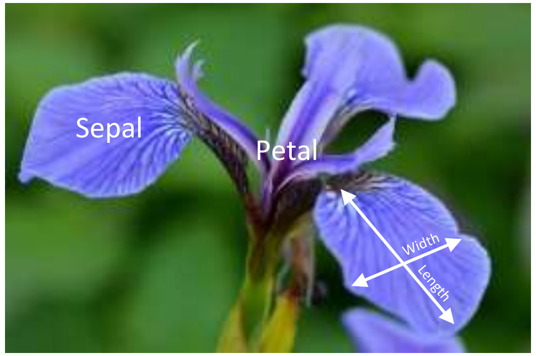
1. Suppose we are interested in mobile purchasing behavior. Let c be the class label denoting purchase/no purchase, and x_1 a feature denoting whether a phone is an iPhone or not. After performing a logistic regression, we get $\hat{\beta}_1 = .693$. What does this mean?
2. When finished, share your answers with your table

DELIVERABLE

Answers to the above question

Review | Iris dataset

Iris Setosa



Iris Versicolor

Iris Virginica

Source: Flickr

- 3 classes of Irises (*Setosa*, *Versicolor*, and *Virginica*)
- 4 attributes
  - Sepal length and width
  - Petal length and width
- 50 instances of each class

DS

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)