Binary Classification and Logistic Regression

Binary Classification - Problem Motivation

- Common examples of classification problems:
  - identifying spam emails to prevent people from receiving spam
  - predicting if borrowers will default on their loans
  - determining whether someone has a disease to guide treatment decisions
- All of these are <u>binary</u> classification problems

Binary Classification - Mathematical Description

- A classifier model is a mapping between your feature space and a finite set
- A binary classifier maps onto {0, 1}
- Example
  - Features: GPA [0, 4], SAT score [600, 2400]
  - Target: Not admitted {0}, Admitted {1}
  - $F : [0, 4] \times [600, 2400] \mapsto \{0,\ 1\}$
- Binary classifiers can generalize to multiple classes

Logistic Regression - Introduction

- Very popular binary classifier
- Estimates probability that an observation is in a given category based on the observation’s features
  - Regression step estimates the probability
  - Classification step rounds the probability to 0 or 1

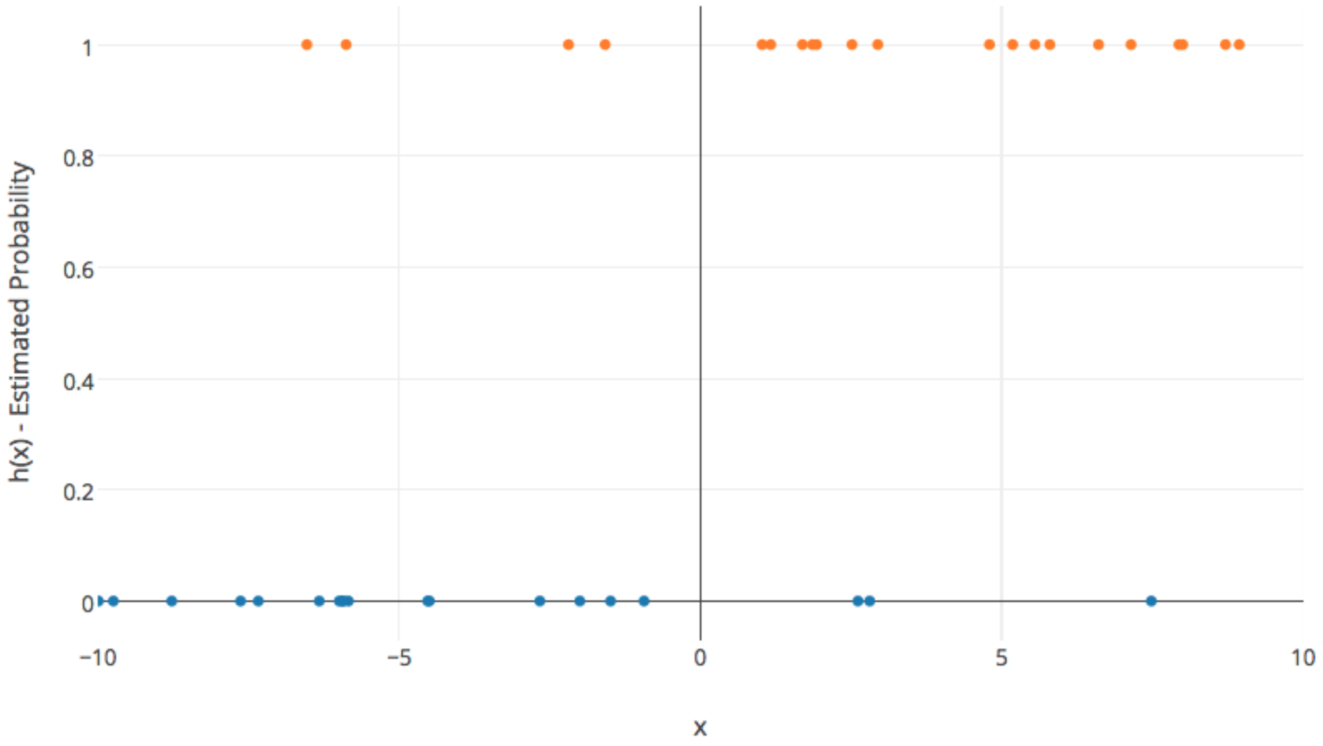
Logistic Regression - Model Framework

- Model assumes each observation is an independent Bernoulli random variable
- Recall that a Bernoulli random variable takes value 1 with probability *p* and value 0 with probability *1-p*

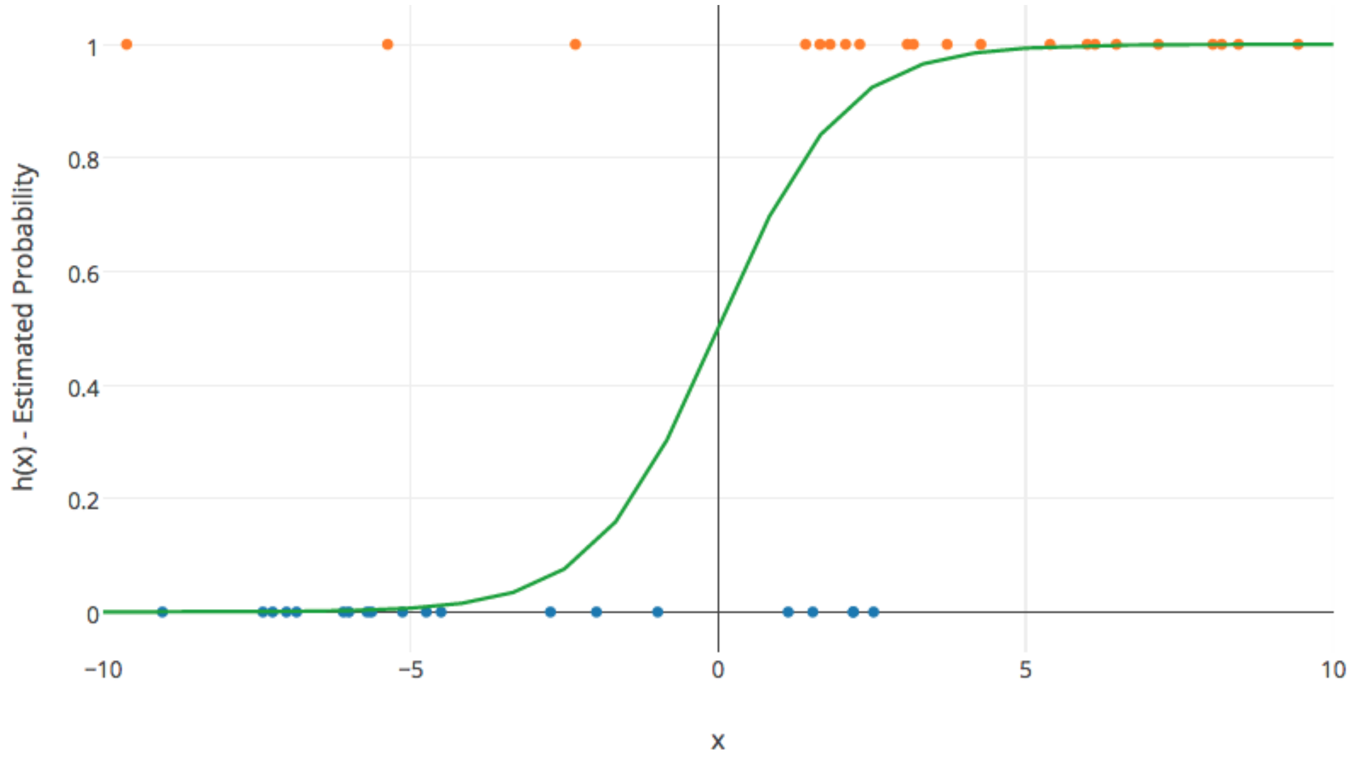
$$f(k;p) = p^k(1-p)^{1-k} \quad \text{for } k \in \{0,1\}.$$

- Logistic regression estimates parameter *p* of the Bernoulli

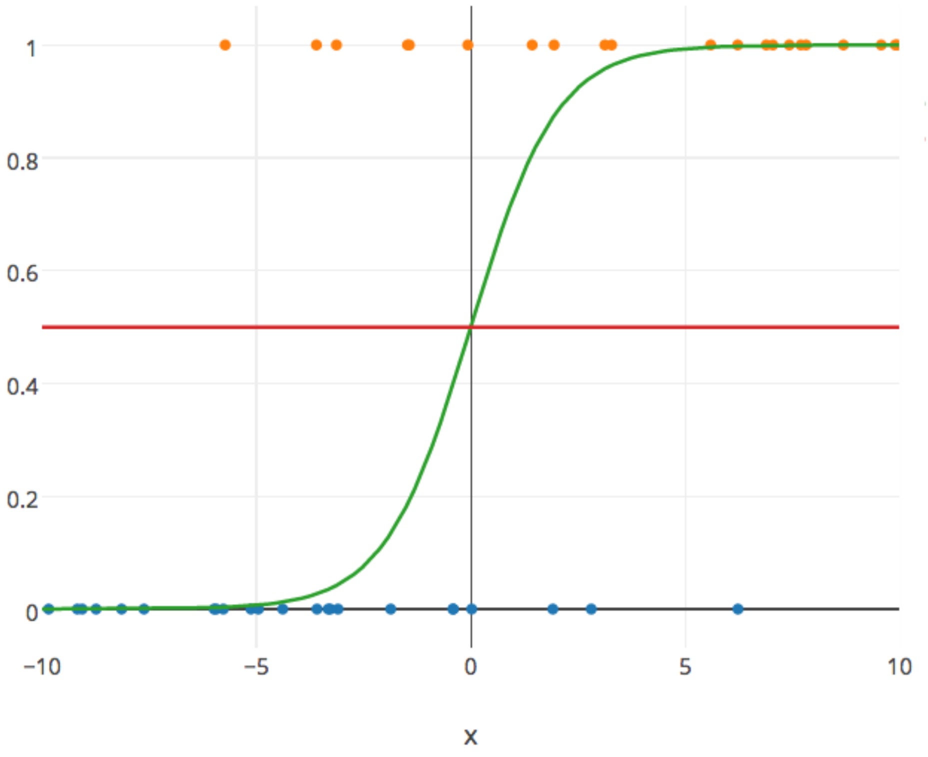
Logistic Regression - Graph



Logistic Regression - Graph



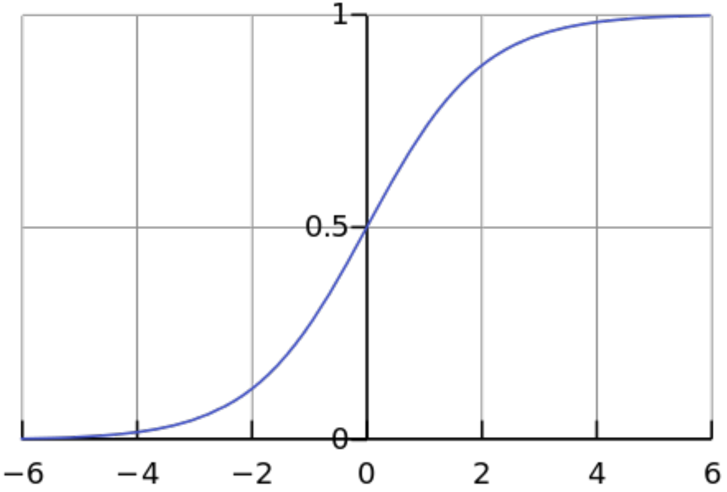
Logistic Regression - Graph



Mapping Feature Space onto Probabilities

- Modeling probabilities requires a functional form that maps onto interval [0,1]
- Typical choice is the logistic function*

$$h_\theta(x) = \frac{1}{(1 + e^{-\theta^T x})}$$



*Other less common choices include the inverse Gaussian ("probit") and the hyperbolic tangent functions.

Log-Odds Ratio

- Logistic model of probability is equivalent to a linear model of the log-odds ratio

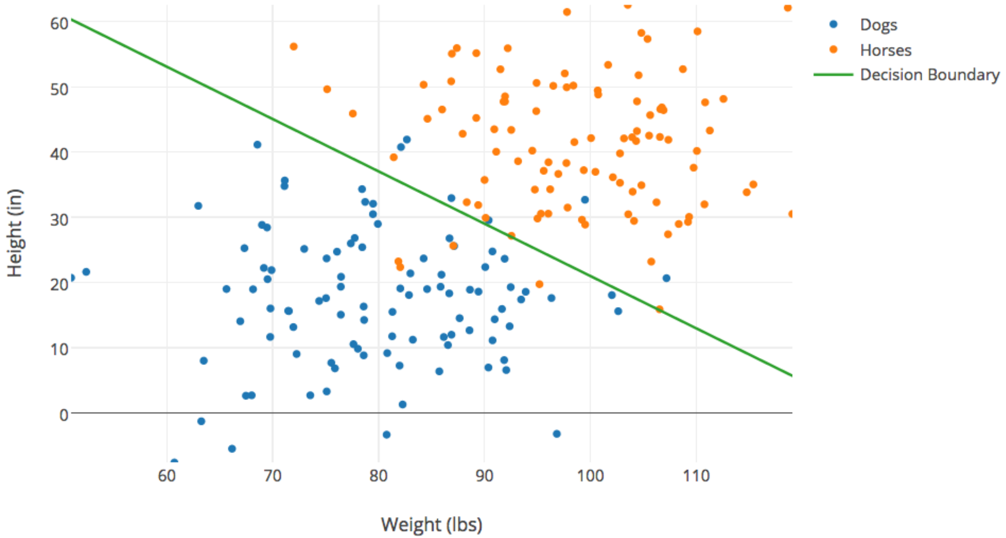
$$h_\theta(x) = \frac{1}{(1 + e^{-\theta^T x})} \rightarrow \ln\left(\frac{p}{1-p}\right) = \theta^T x$$

Example Model

See IPython notebook

Decision Boundary

- The category favored by the hypothesis function flips from 0 to 1 in a certain region of the feature space
- That region is called the “decision boundary”
- Occurs when estimated probability = .5



Decision Boundary

- decision boundary is the surface defined by

$$h_\theta(x) = .5$$

$$\rightarrow \frac{1}{1+e^{-\theta^T x}} = .5$$

$$\rightarrow 1 = e^{-\theta^T x}$$

$$\rightarrow \theta^T x = 0$$

Note: can use threshold values other than .5

Finding Coefficients

- Coefficients for logistic regression are found using Maximum Likelihood Estimation (MLE)
- Likelihood of an observation given the model:

$$p(y_i|x_i;\theta) = h_\theta(x_i)^{y_i}(1-h_\theta(x_i))^{1-y_i}$$

- Assuming each observation is independent:

$$p(\vec{y}|X;\theta) = \prod_{i=1}^{n} h_\theta(x_i)^{y_i}(1-h_\theta(x_i))^{1-y_i}$$

- Choose the coefficients that maximize this expression

Finding Coefficients

- In practice, we maximize the log likelihood:

$$\ln p(\vec{y}|X;\theta) = \sum_{i=1}^{n} (y_i \ln h_\theta(x_i) + (1-y_i)\ln(1-h_\theta(x_i)))$$

- Observe how the value of each term varies:

$$y_i = 0 \Rightarrow \lim_{h_\theta(x)\to 0} (y_i \ln h_\theta(x_i) + (1-y_i)\ln(1-h_\theta(x_i))) = 0$$

$$y_i = 0 \Rightarrow \lim_{h_\theta(x)\to 1} (y_i \ln h_\theta(x_i) + (1-y_i)\ln(1-h_\theta(x_i))) = -\infty$$

$$y_i = 1 \Rightarrow \lim_{h_\theta(x)\to 1} (y_i \ln h_\theta(x_i) + (1-y_i)\ln(1-h_\theta(x_i))) = 0$$

$$y_i = 1 \Rightarrow \lim_{h_\theta(x)\to 0} (y_i \ln h_\theta(x_i) + (1-y_i)\ln(1-h_\theta(x_i))) = -\infty$$

Interpreting Coefficients

- Recall that logistic regression implies a linear relationship between the features and the logit odds:

$$\ln \frac{p}{1-p} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_k x_k$$

- Increasing feature value by 1 increases logit odds by *θ* and odds by *e^θ*

Evaluating a Binary Classifier

Evaluating a Binary Classifier

Accuracy

- Percent of observations correctly classified: $\frac{TP+TN}{n}$
- Most intuitively understandable metric
- Unfortunately, accuracy is a problematic metric
  - Imbalanced classes will inflate accuracy
    - Ex. If 90% of the population is in one category, then naive model has 90% accuracy
  - Doesn’t reveal what kind of errors are being made

Evaluating a Binary Classifier

Evaluating a Binary Classifier

Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actually Positive	True Positives	False Negatives
Actually Negative	False Positives	True Negatives

Classifier Metrics

Accuracy

$$\frac{TP + TN}{n}$$

True Positive Rate (Sensitivity/Recall)

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

True Negative Rate (Specificity)

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

Precision

$$\frac{TP}{TP + FP}$$

Evaluating a Binary Classifier

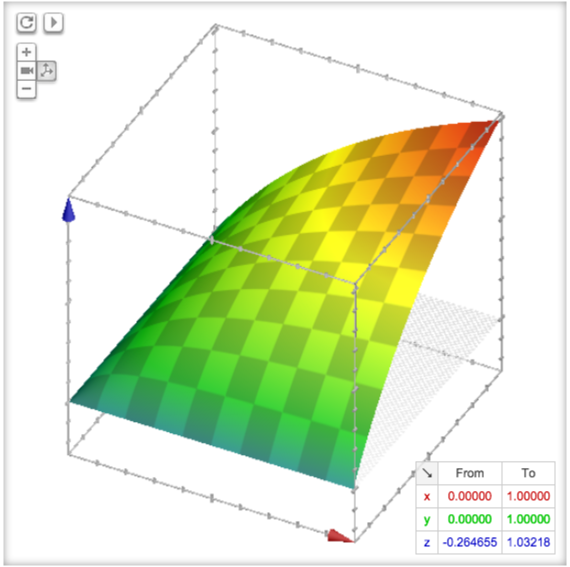
F1 Score

- harmonic mean of precision and recall

$$F_1 = \frac{2 * precision * recall}{precision + recall} = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

F_β Score

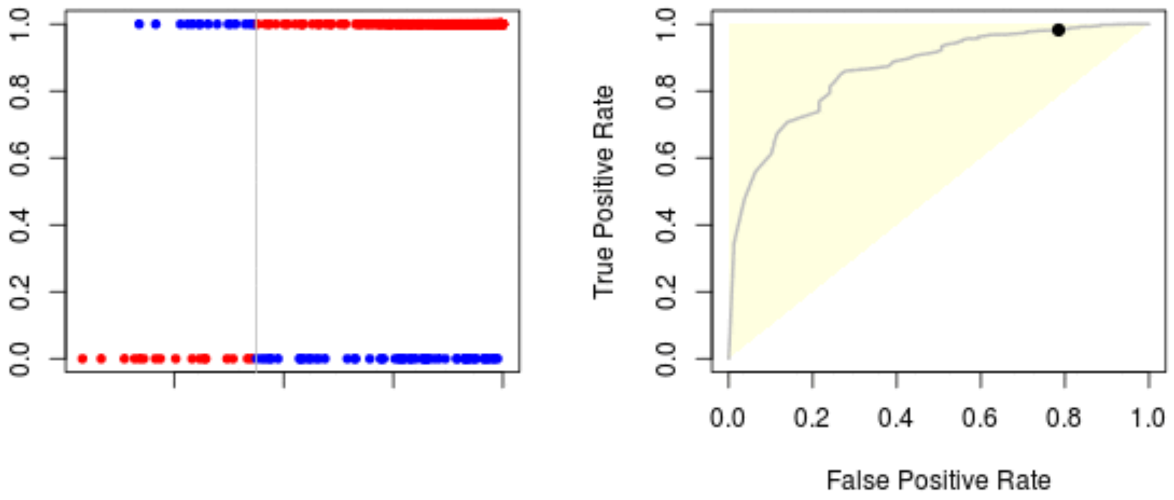
$$F_\beta = (1 + \beta^2)\frac{precision * recall}{\beta^2 precision + recall}$$



Evaluating a Binary Classifier

ROC Plot

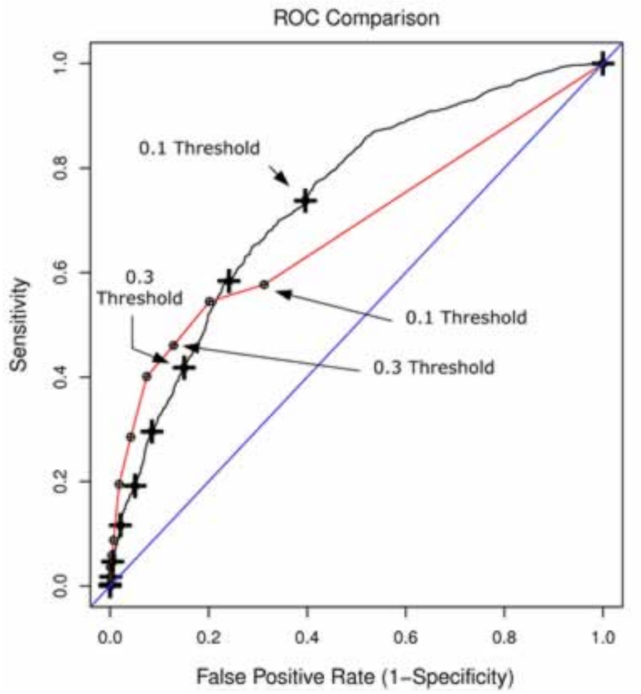
- Shows how true and false positive rates vary as the decision boundary is moved (animation)



Evaluating a Binary Classifier

ROC Plot

- If classifier A's ROC curve is strictly greater than classifier B's, then classifier A is always preferred
- If two classifier's ROC curves intersect, then the choice depends on relative importance of sensitivity and specificity



Evaluating a Binary Classifier

ROC - Area Under Curve (AUC)

- equals the probability that the model will rank a randomly chosen positive observation higher than a randomly chosen negative observation
- useful for comparing different classes of models in general setting

Imbalanced Class Problem

- What happen if your sample is imbalanced?
  - E.g. 99% not spam, 99% healthy, 99% no default
- This is a problem when interested in minority class
  - i.e. False positive not equal in cost to false negative

Solutions to Imbalanced Class Problem

- Under/oversampling
- Cost-sensitive learning

Over- and Under-sampling for Imbalanced Classes

- Populations often do not have equal proportions of each class
- Over- and under-sampling can simulate balanced classes
- Oversampling
  - replicate samples in smaller class
  - can cause overfitting because noise is replicated
  - can generate new examples in neighborhood of observations
- Undersampling
  - subsample from larger class repeatedly and ensemble classifiers
- Can combine over- and under-sampling

Evaluating Logistic Regression

Likelihood Ratio Test

- A hypothesis test that compares one model with a null hypothesis model
- Given 2 models, where one model's parameters is a subset of the other, compute the likelihood ratio:

$$G^2 = 2\ln\left(\frac{L}{L_0}\right) \sim \chi^2$$

- degrees of freedom equals difference in number of parameters between the two models

Evaluating Logistic Regression

Likelihood Ratio

- Common choice of null hypothesis is model with only intercept term (i.e. the sample mean of y)

$$H_0 : \ln\left(\frac{p}{1-p}\right) = \frac{1}{1+e^{-\theta_0}}$$

$$H_1 : \ln\left(\frac{p}{1-p}\right) = \frac{1}{1+e^{-\theta^T x}}$$

- Note that this has the same caveats as any frequentist hypothesis testing method