CDS Education

Introduction to Machine Learning for Python

Logistic Regression and Decision Trees

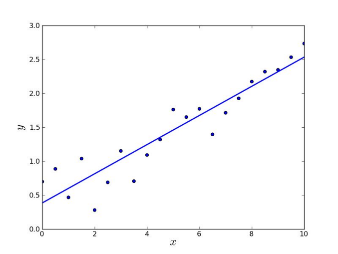
Reminders

- Project Part B was due **yesterday**
- Project Part C will be released tonight
- Mid-Semester Evaluations
  - Helpful whether you really like the class or really hate it
- Get Pollo - code JYHDQR

Review: Supervised Learning

Regression

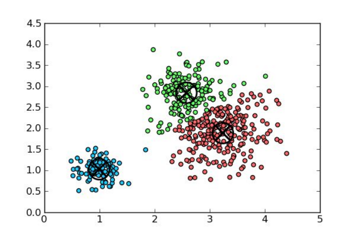
"How much?"
Used for *continuous* predictions



[Source](#)

Classification

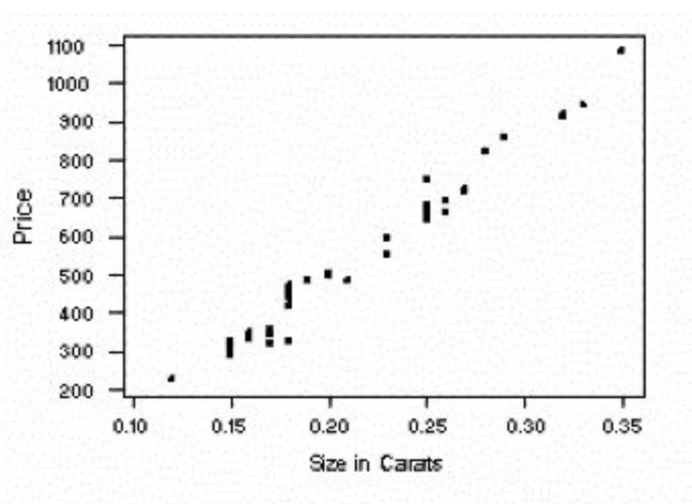
"What kind?"
Used for *discrete* predictions



[Source](#)

Review: Regression

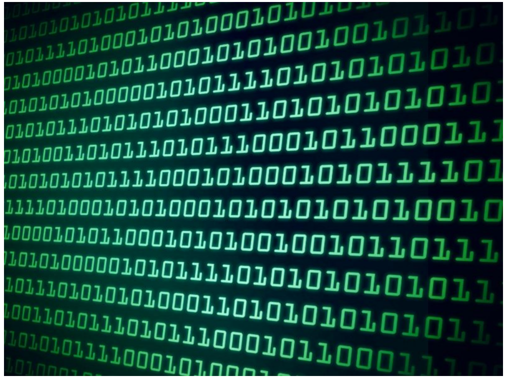
We want to find a **hypothesis** that explains the behavior of a **continuous** *y*.



$$y = B_0 + B_1x_1 + \ldots + B_px_p + \varepsilon$$

Source

Regression for binary outcomes



Regression can be used to **classify**:

- Likelihood of heart disease
- Accept/reject applicants to Cornell Data Science based on affinity to memes

Estimate **likelihood** using regression, convert to **binary** results

Source

Conditional Probability

The probability that an event (A) will occur given that some condition (B) is true

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Conditional Probability

The probability that:

- You have a heart disease given you have x blood pressure, you have diabetes, and you are y years old.
- You are accepted to Cornell Data Science given that you spend x hours a day in the meme fb group

Logistic Regression

1) Fits a linear relationship between the variables
2) Transforms the linear relationship to an estimate function of the **probability** that the outcome is 1.

Basic formula:

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k)}}$$

(Recognize this?)

$$\ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k$$

Pollo Question

What is the output of the logistic regression function?

A. Value from $-\infty$ to ∞
B. Classification
C. Numerical value from 0 to 1
D. Binary value

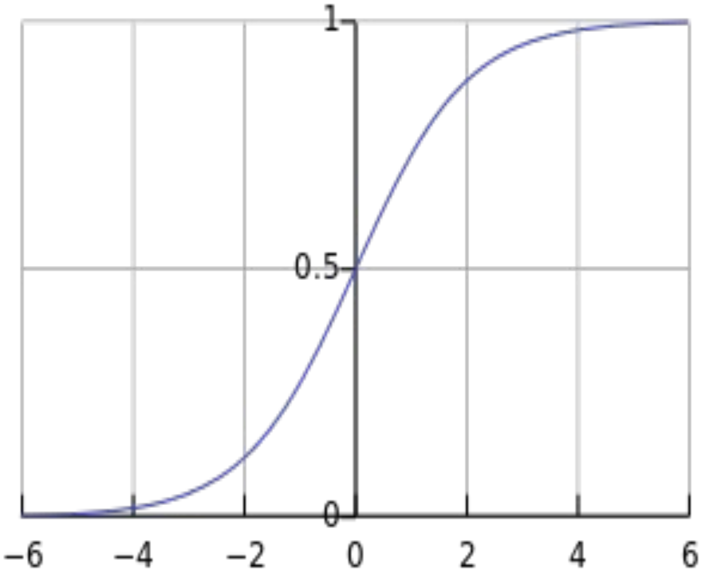
Pollo Question

What is the output of the logistic regression function?

A. Value from -∞ to ∞
B. Classification
C. Numerical value from 0 to 1
D. Binary value

Sigmoid Function

$$P(x) = \frac{1}{1+e^{-x}}$$



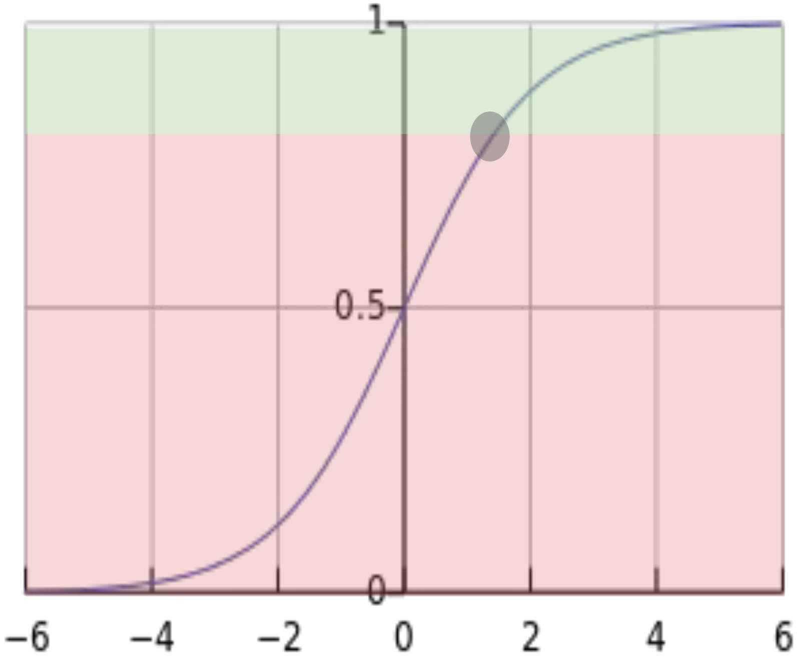
Depending on the regression formula value, P(x) can be between 0 and 1 as x goes from -∞ to ∞.

Source

Threshold

Where between 0 and 1 do we draw the line?

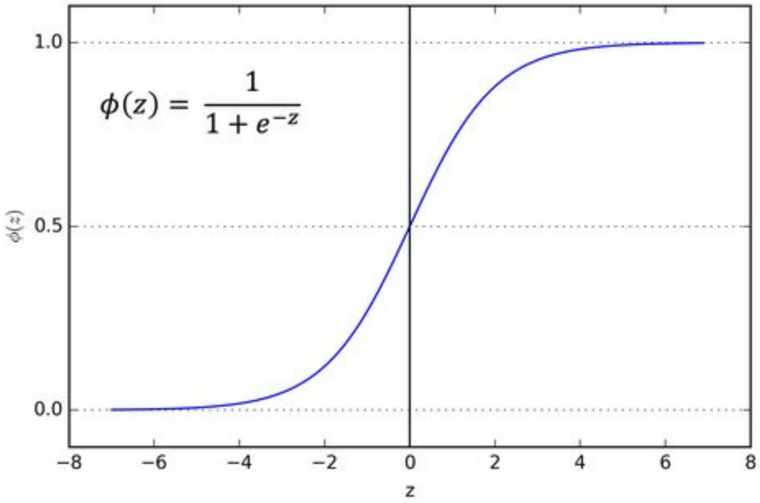
- $P(x)$ below threshold: predict 0
- $P(x)$ above threshold: predict 1



Thresholds matter (a lot!)

What happens to the specificity when you have a

- Low threshold?
  - Sensitivity increases
- High threshold?
  - Specificity increases

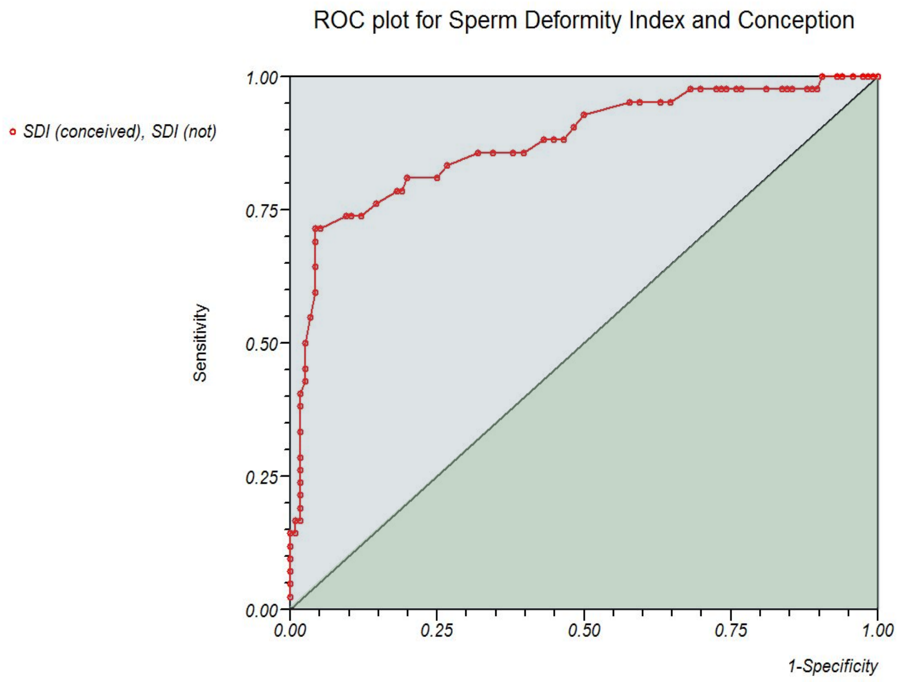


Source

ROC Curve

Receiver **O**perating **C**haracteristic

- Visualization of trade-off
- Each point corresponds to a specific threshold value



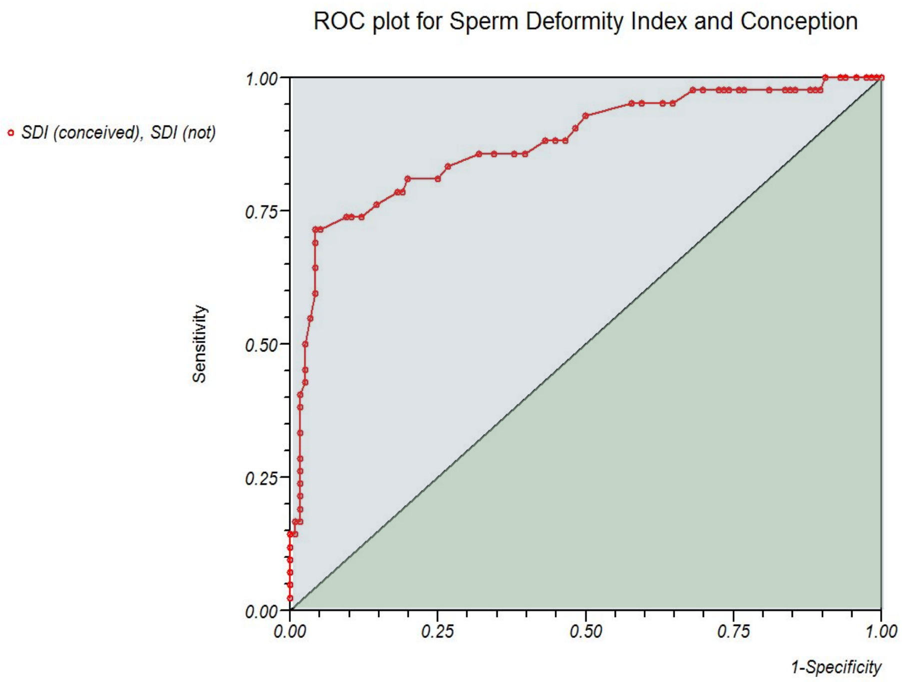
Area Under Curve

$$\text{AUC} = \int ROC\text{-}curve$$

Always between 0.5 and 1.

Interpretation:

- 0.5: Worst possible model
- 1: Perfect model



Why Change the Threshold?

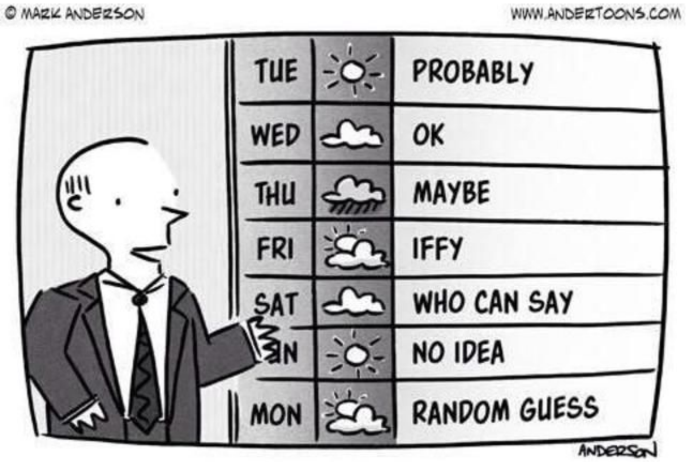
- Want to increase either sensitivity or specificity
- Imbalanced class sizes
  - Having very few of one classification skews the probabilities
  - Can also fix with rebalancing classes
- Just a very bad AUC

Changing Thresholds in the Code

- Sklearn uses a default of 0.5
    - This will be fine a majority of the time
- Have to change the threshold "manually"
    - If the accuracy is low, check the **auc**
    - If high auc, then use **predict_proba**
        - Map the probabilities for each class to the label

Is Logistic Regression Classification?

- Partly classification, partly prediction
- Value in logistic regression is the probabilities
    - Have confidence value for each prediction
    - Can act differently based on confidence



"And now the 7-day forecast..."

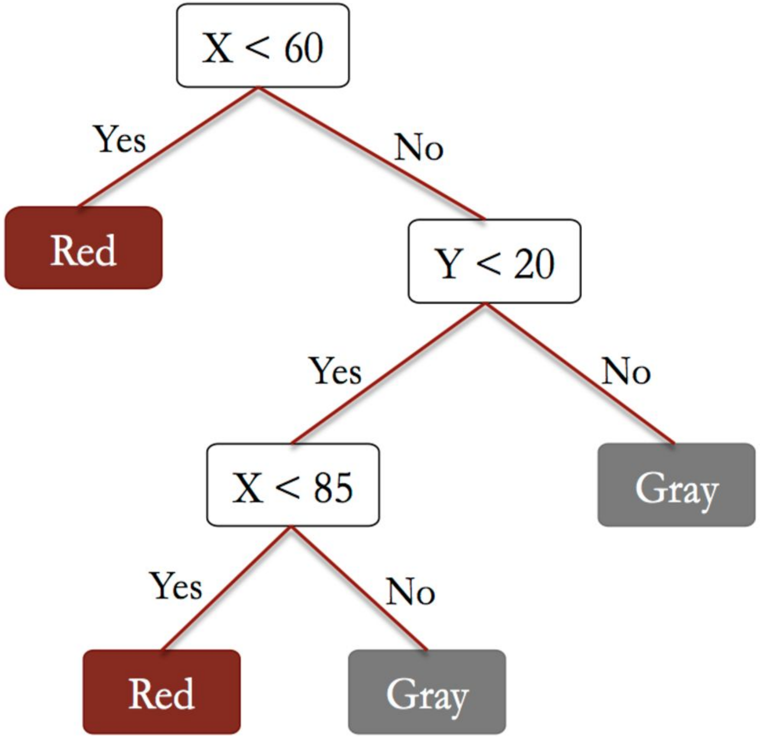
[Source]

When to Use Regression

- Works well on (roughly) linearly separable problems
  - Remember SVM kernels for non-linearly separable
- Outputs probabilities for outcomes
- Can lack **interpretability**, which is an important part of any useful model

CART (Classification and Regression Trees)

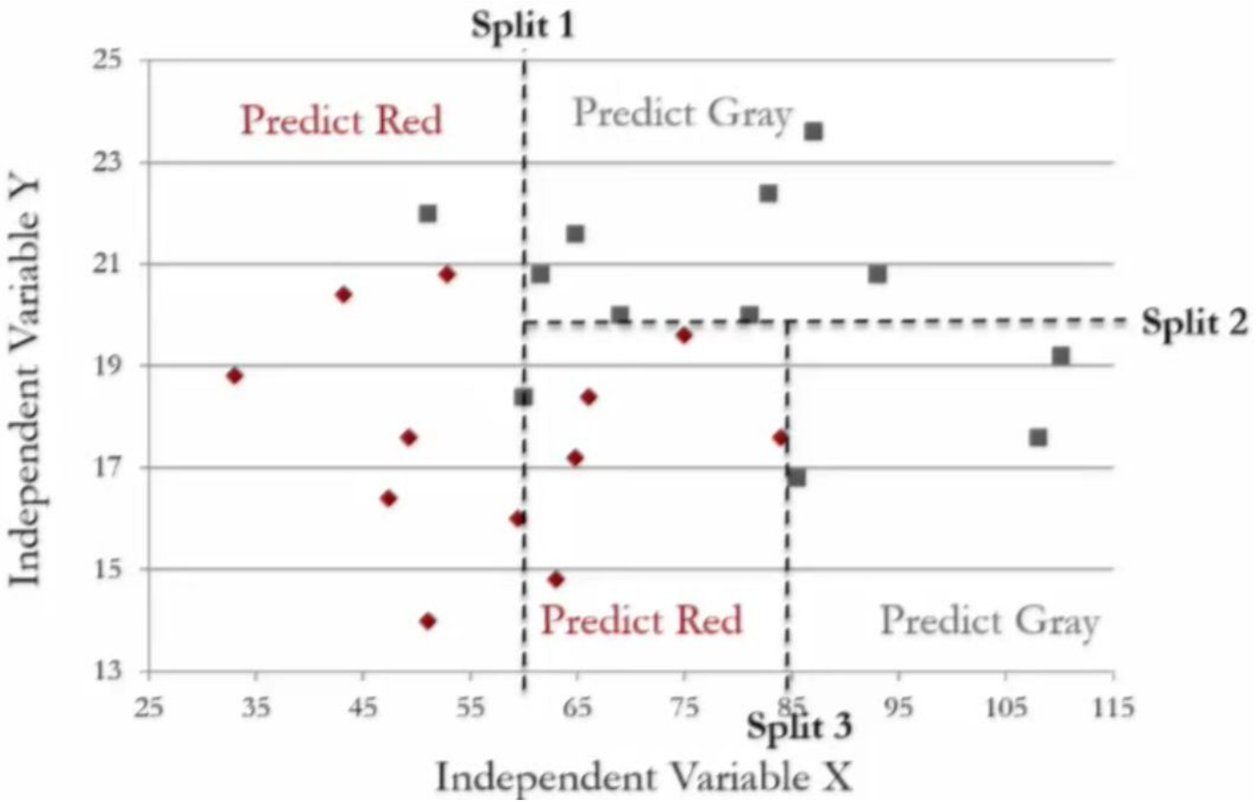
- At each node, split on variables
- Each split minimizes error function
- Very interpretable
- Models a non-linear relationship!



Splitting the data

= red

= gray



How to Grow Trees

Greedy Splitting (recursive binary splitting)

- Check all possible splits using a cost function
  - Categorical: try every category
  - Numerical: bin the data
- Pick the one that minimizes the cost
- Recurse until reached the stopping criterion
- Prune to prevent overfitting

Source

How to Grow Trees - Cost Function

- Classification and Regression Trees
  - Can be for either classification or regression
- Cost function for regression is the minimizing sum of squared errors
  - Same function

How to Grow Trees - Cost Function

Gini Impurity

- 1 - probability that guess i is correct
- Lower is better

$$1 - \sum p_i^2$$

Entropy (Information Gain)

- Homogeneity of a group
- Lower is better

$$-\sum p_i \log p_i$$

Source

Gini Impurity Example - Good Split

Healthy?	
Yes	No
9	1

- Probability(Yes) = 0.9
- Probability(No) = 0.1
- Impurity

  = 1 - (0.9^2 + 0.1^2)

  = **0.18**

Gini Impurity Example - Bad Split

Healthy?	
Yes	No
5	5

- Probability(Yes) = 0.5
- Probability(No) = 0.5
- Impurity
  = 1 - (0.5^2 + 0.5^2)
  = **0.5**

Entropy Example - Good Split

Healthy?	
Yes	No
9	1

- Probability(Yes) = 0.9
- Probability(No) = 0.1
- Entropy

  = -0.9*log 0.9 - 0.1*log 0.1

  = **0.14**

Entropy Example - Bad Split

Healthy?	
Yes	No
5	5

- Probability(Yes) = 0.5
- Probability(No) = 0.5
- Entropy

  = -0.5*log 0.5 - 0.5*log 0.5

  = **0.3**

How to Grow Trees - Stopping Criterion & Pruning

Used to **control overfitting** of the tree

- Stopping Criterion
  - max_depth, max_leaf_nodes
  - **min_samples_split**
    - Minimum number of cases needed for a split
- Pruning
  - Compare overall cost with and without each leaf
  - Not currently supported

How to Grow Trees

- Start at the top of the tree
- Split attributes one by one
    - Based on cost function
- Assign the values to the leaf nodes
- Repeat
- Prune for overfitting

When to Use Decision Trees

- Easy to interpret
  - Can be visualized
- Requires little data preparation
- Can use a lot of features
- Prone to overfitting

Coming Up

Your problem set: Project Part C released

Next week: Unsupervised Learning

See you then!

Cornell Data Science