Binary Classification

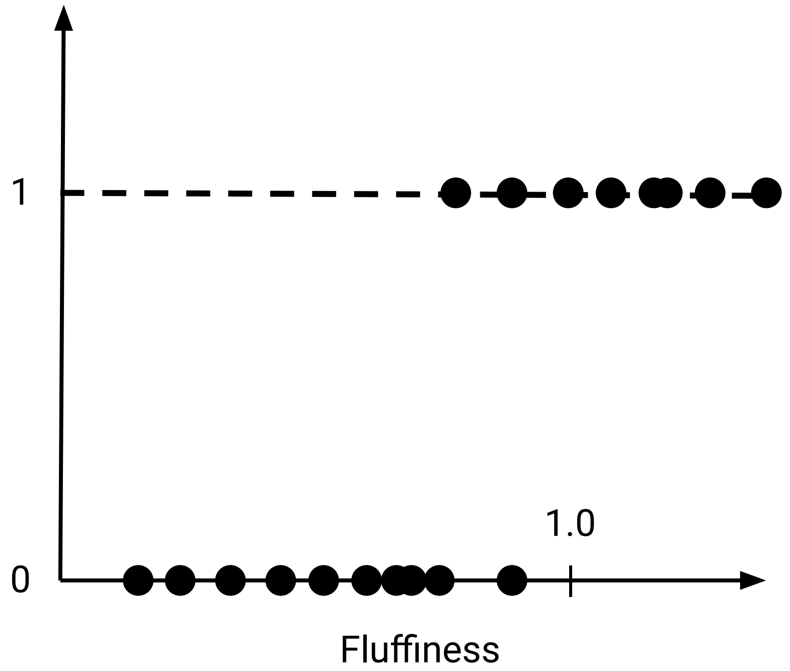
BSDS 100, Spring 2021
Michael Ruddy

Binary Classification

- Binary Classification is the task of classifying inputs in two groups.
- Examples
    - Classify images as either Cat or Dog
    - Emails: Spam or Not Spam
    - Medical Diagnosis: Positive or Negative Diagnosis

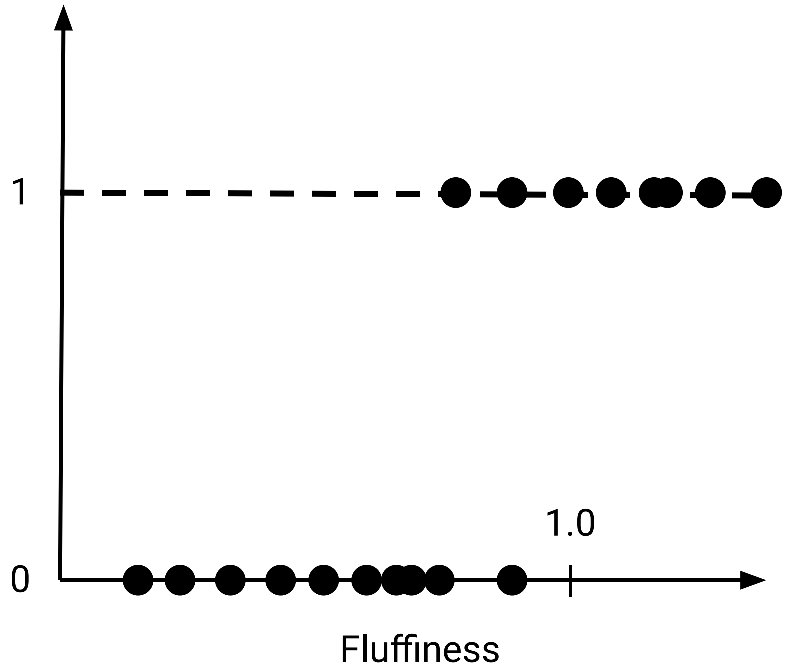
Univariate Binary Classification

- Binary classification with one variable as an input *feature*.
- Example: Classify as “Cat” or “Not Cat” based on Fluffiness
- “Not Cat” is encoded as 0 and “Cat” is encoded as 1



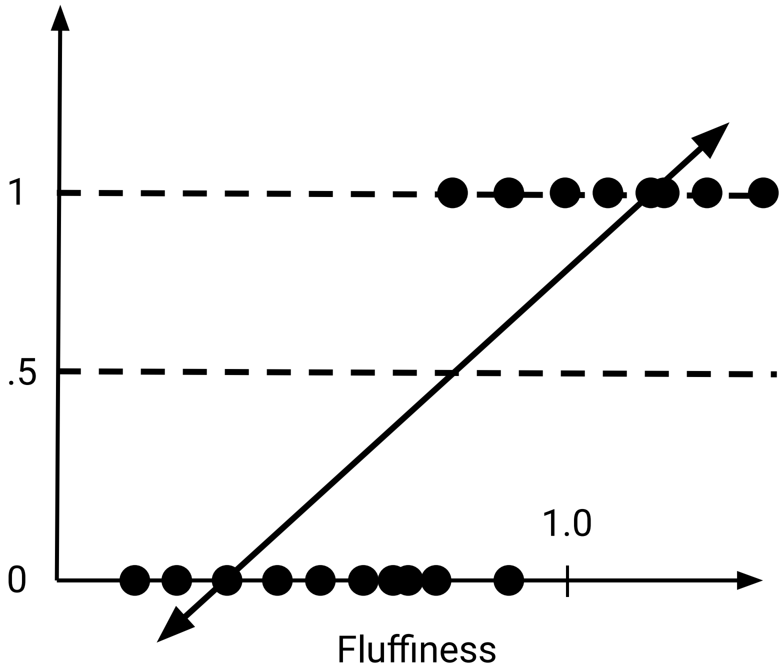
Univariate Binary Classification

- Goal: Construct a model that outputs the probability of being a Cat given its Fluffiness score.



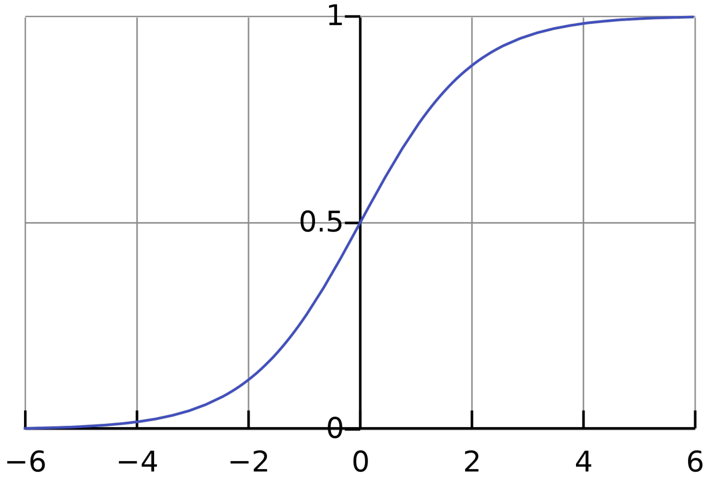
Univariate Binary Classification

- Goal: Construct a model that outputs the probability of being a Cat, given its Fluffiness score.
- Linear Model doesn't do a great here....



Univariate Binary Classification

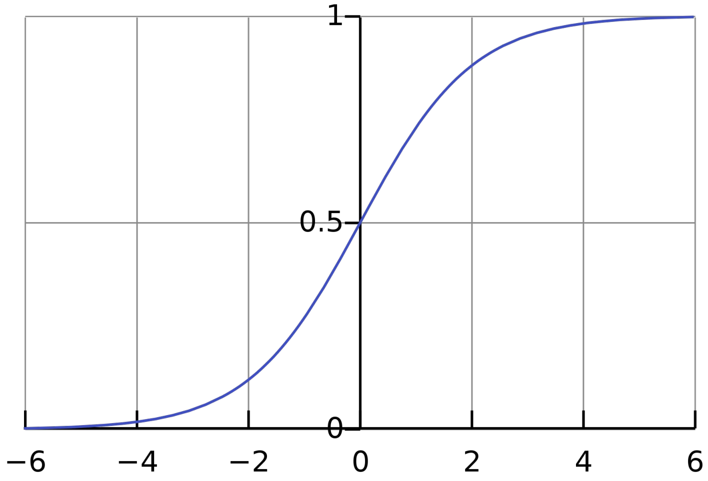
- Instead we often use what is called a **logistic model**.
- As *x* becomes more negative, the function *p(x)* approaches zero.
- As *x* becomes larger, the function *p(x)* approaches 1.
- *Sigmoid*



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

Univariate Binary Classification

- To fit a logistic model to data, we can introduce a more general *model family* with parameters.
- Equivalent to applying *p(x)* to a linear function of *x*.

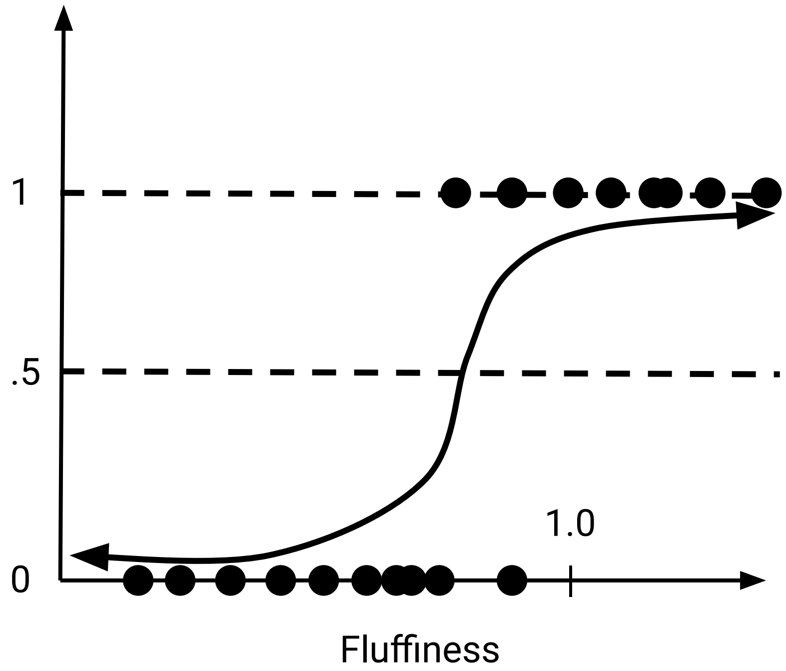


$$p(x) = \frac{1}{1+e^{-(a_1+a_2x)}}$$

Univariate Binary Classification

- Fitting a logistic model means choosing “good” values for parameters *a1* and *a2*.

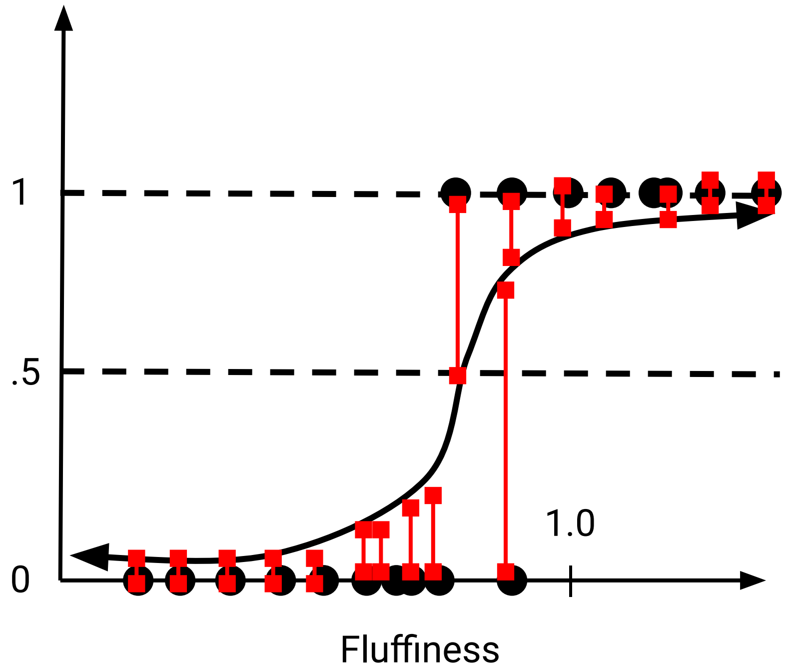
$$p(x) = \frac{1}{1+e^{-(a_1+a_2x)}}$$



Binary Cross-Entropy Loss

- As before we can estimate the error of our prediction.
- Error for a single point:

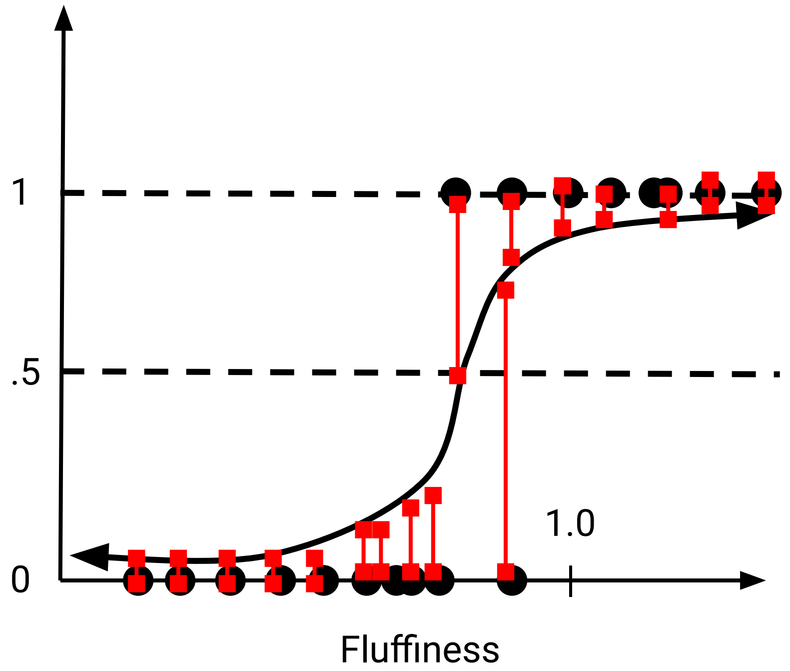
$$y(1 - p(x)) + (1 - y)p(x)$$



Binary Cross-Entropy Loss

- But for BCE Loss we measure how *well* the model does (and take the negative).

$$-[yp(x) + (1-y)(1-p(x))]$$

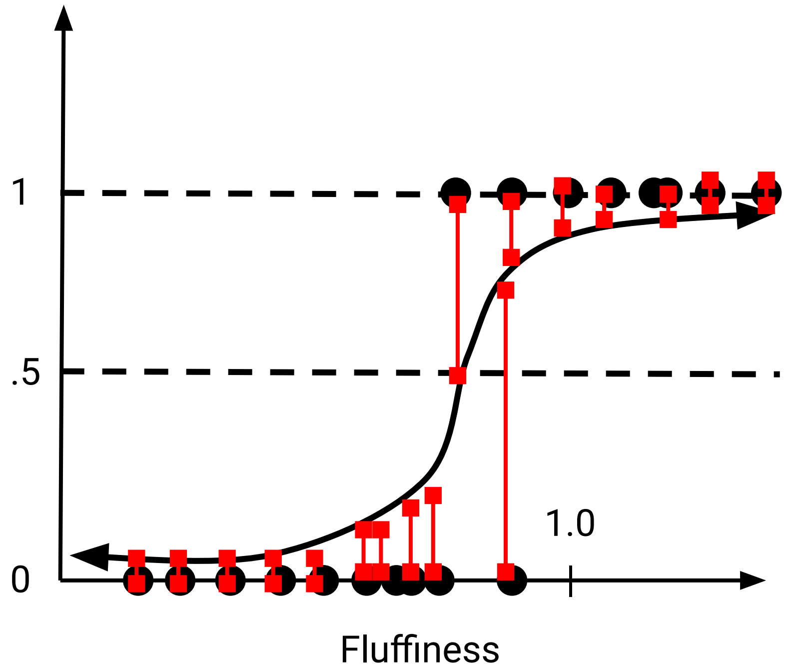


Binary Cross-Entropy Loss

- Instead of measuring probabilities we measure the *logarithm* of the probabilities.

$$-[y\log(p(x)) + (1-y)\log(1-p(x))]$$

- Based on a dissimilarity measure between prob. distributions.



Binary Cross-Entropy Loss

- Finally we average this score over all points in our training data.

$$\frac{\sum_{i=1}^{N} -[y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i))]}{N}$$

- Optimization: Find values of *a1* and *a2* that minimize above.

$$p(x) = \frac{1}{1+e^{-(a_1+a_2x)}}$$

Binary Cross-Entropy Loss

- Find parameters in our model family that minimize the loss function

Loss

$$\frac{\sum_{i=1}^{N} -[y_i \log(p(x_i)) + (1 - y_i)\log(1 - p(x_i))]}{N}$$

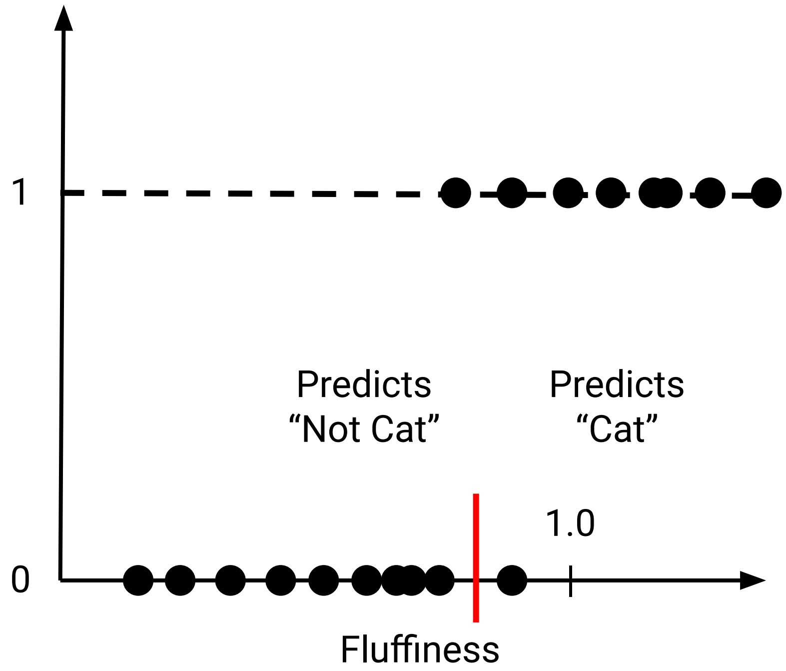
parameters

$$p(x) = \frac{1}{1+e^{-(a_1+a_2x)}}$$

model family

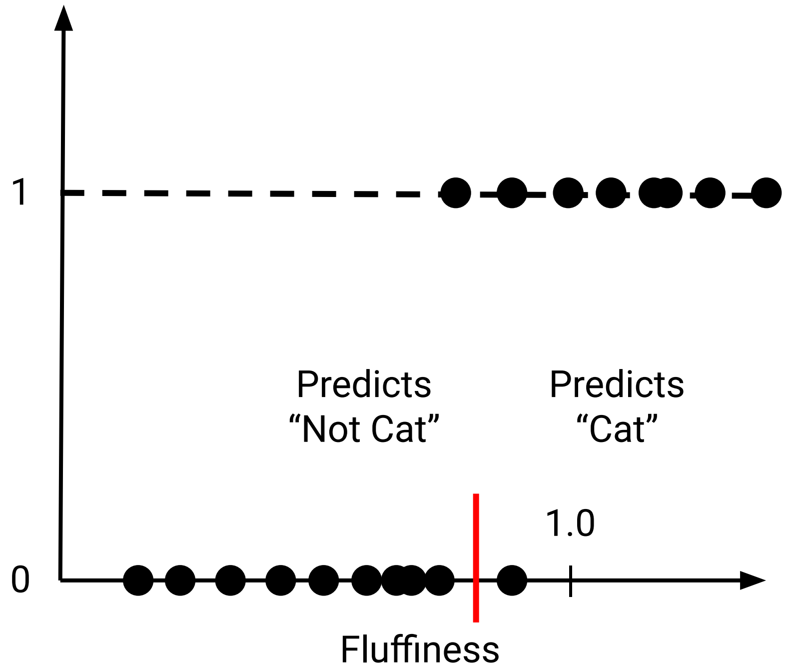
Decision Boundary

- We can also interpret the logistic model as finding a good *decision boundary*.



Decision Boundary

- We can also interpret the logistic model as finding a good *decision boundary*.
- Model: $a_1 + a_2 x$
- Positive -> "Cat"
- Negative -> "Not Cat"

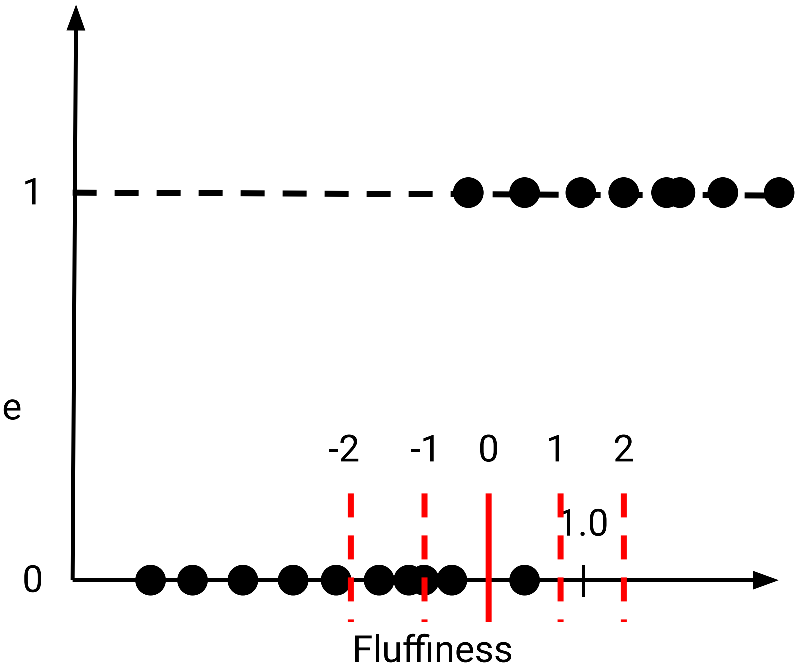


Decision Boundary

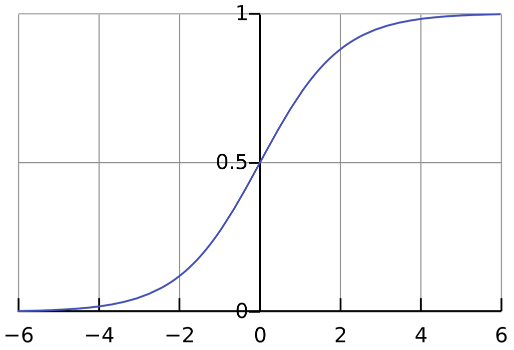
- We can also interpret the logistic model as finding a good *decision boundary*.
- Model: $a_1 + a_2 x$
- Positive -> "Cat"
- Negative -> "Not Cat"

Position of boundary

Scaling of magnitude "confidence"



Decision Boundary

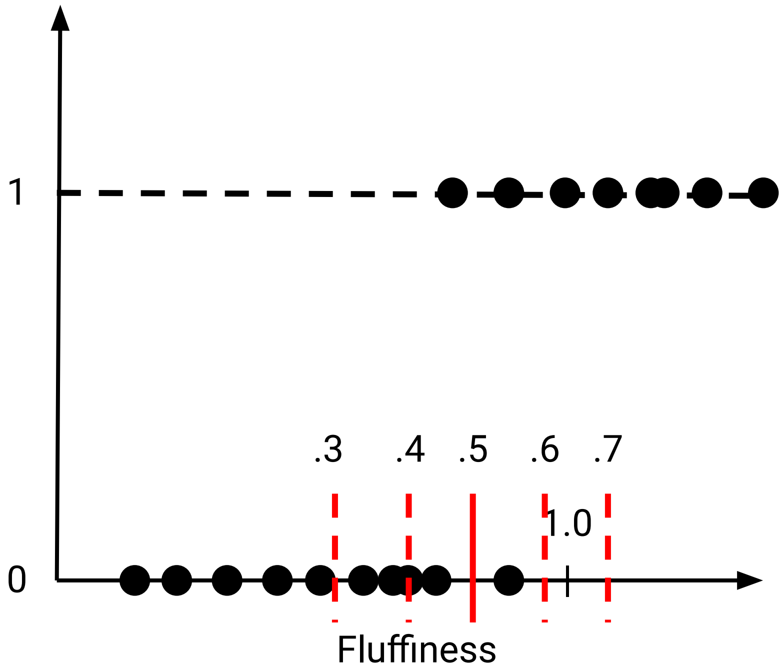


$$\frac{1}{1+e^{-(a_1+a_2 x)}}$$

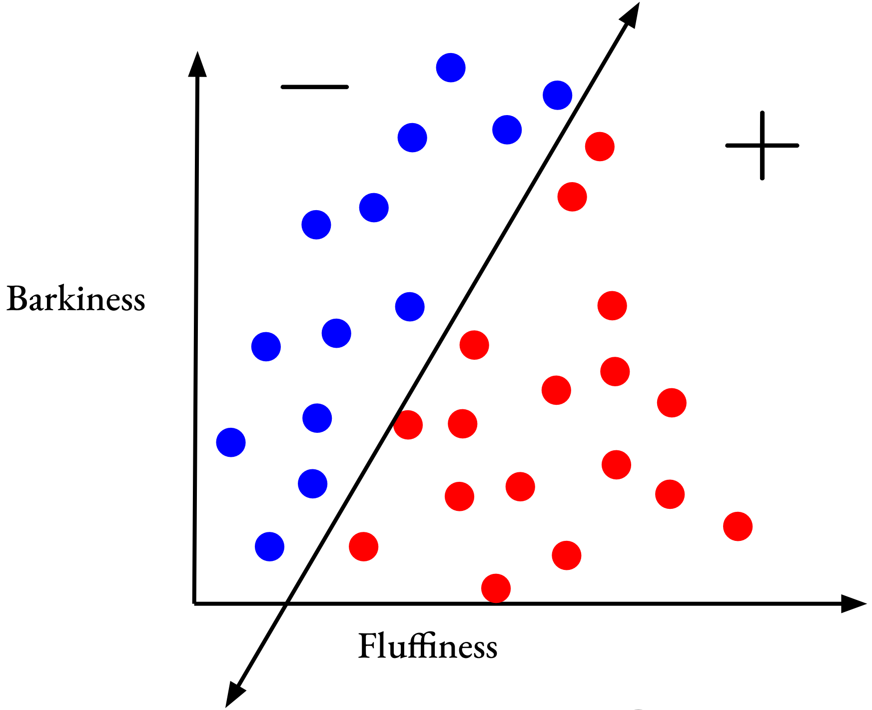
- Model:

  p(x) > .5 -> “Cat”

- p(x) < .5 -> “Not Cat”

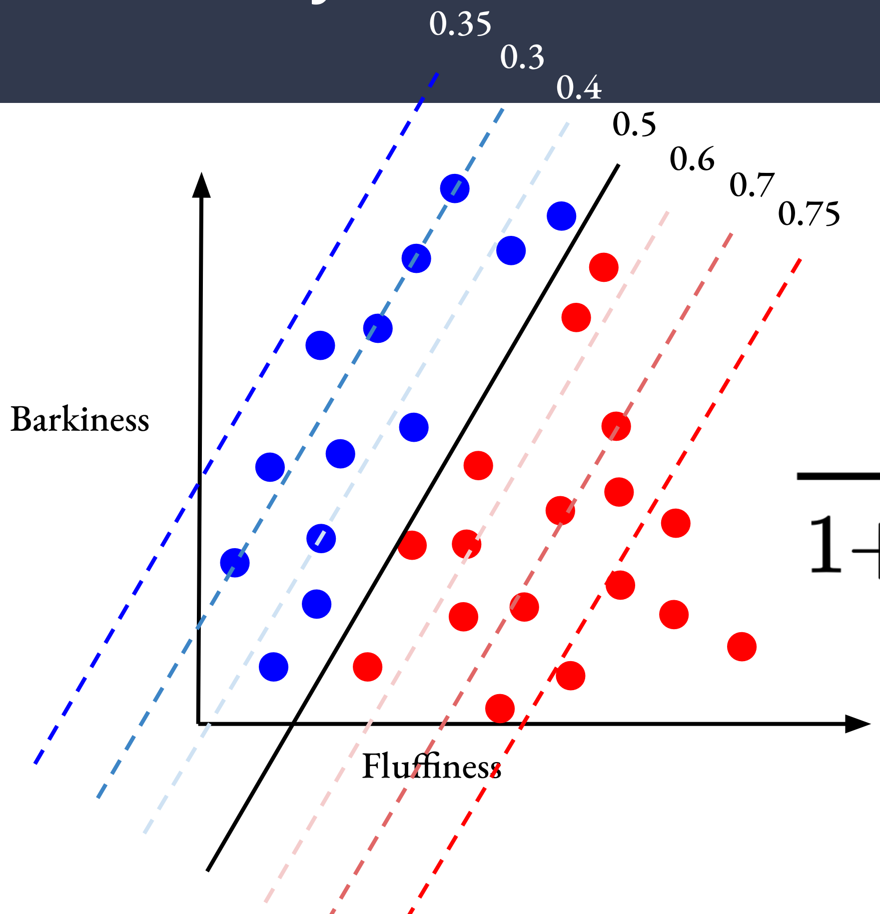


Decision Boundary



$$a_1 + a_2 x + a_3 y = 0$$

Decision Boundary



$$\frac{1}{1+e^{-(a_1+a_2x+a_3y)}}$$

Summary

- The steps for logistic regression are the same as for linear regression
  - Good Model Family and Loss Function
  - Find parameters that minimize Loss
- We can implement it from scratch almost identically to how we implemented linear regression in R
- We will also see that these models are so common that there are built-in methods for linear/logistic regression