Introduction to Naive Bayes Classifier

THE KNOWLEDGE HOUSE

Agenda - Schedule

1. Kahoot
2. Supervised Learning
3. Classification Tasks
4. Bayes Theorem
5. Break
6. Probability Lab

It's stats all the way down

Agenda - Goals

- ...

Announcements

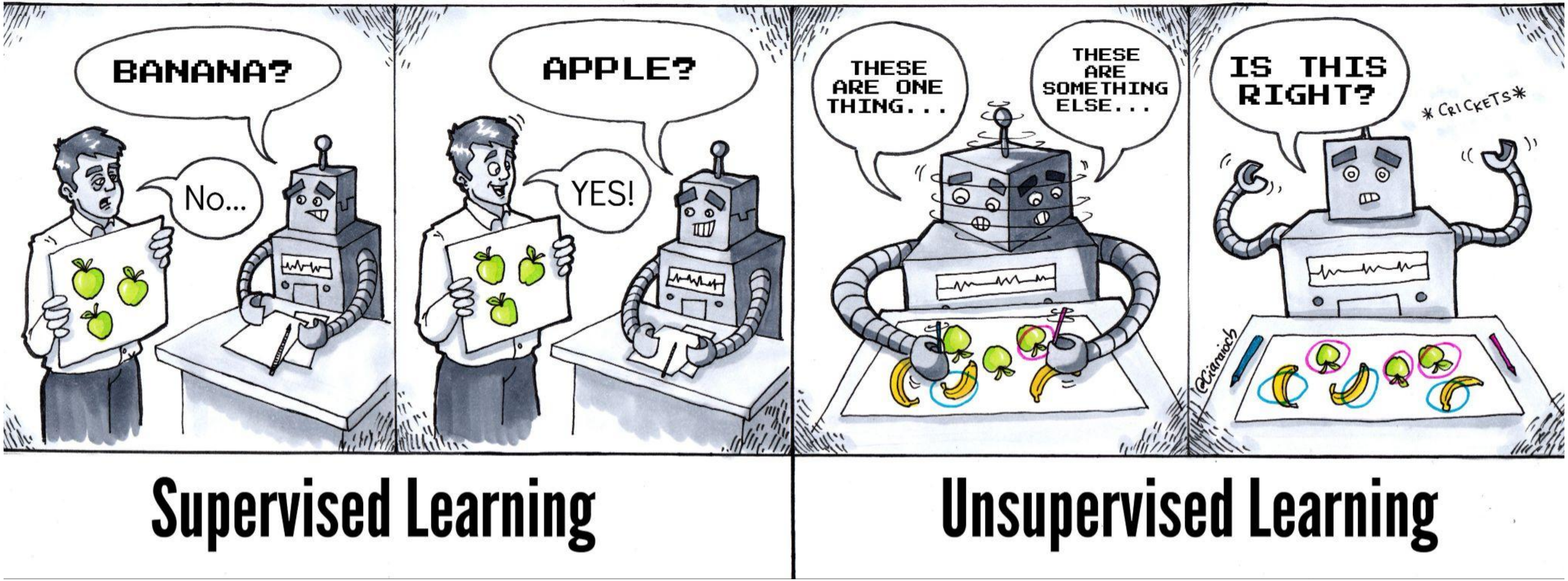
- …

Warm Up

Kahoot!

Start Week 4 Kahoot - Logistic Regression

Supervised Learning Review

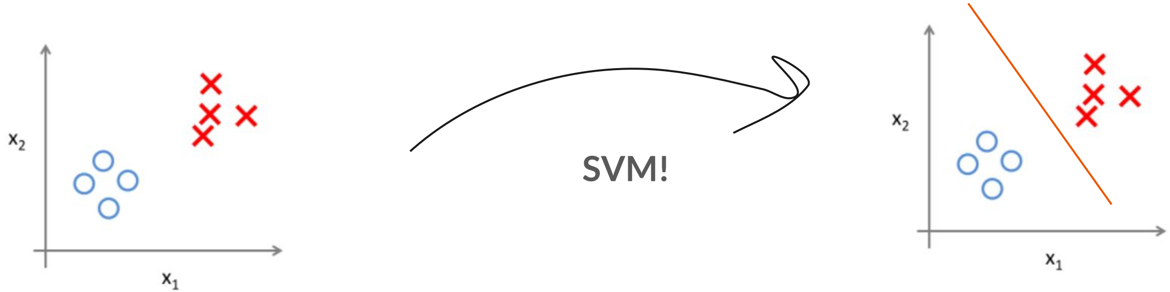


Now that we are thoroughly in Machine Learning territory, let's go over the differences between 2 types of supervised learning.

Supervised Learning

Supervised learning: output is **known** (a target **is** provided), model needs to **recreate relationships.** We “learn” which weights create a function predicts

- **Evaluation**: Tested against the known data. How well did it identify what we know already?



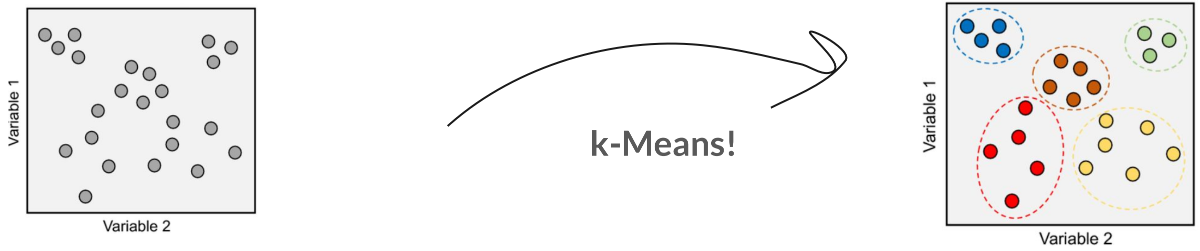
Here are my classes. What differentiates class 1 from class 2?

Wow! 100% accuracy!

Unsupervised Learning

Unsupervised learning: output is **unknown** (target **is not** provided), model needs to **discover structure**

- **Evaluation**: Tested against relative performance criteria. How well did it identify sensible structure?



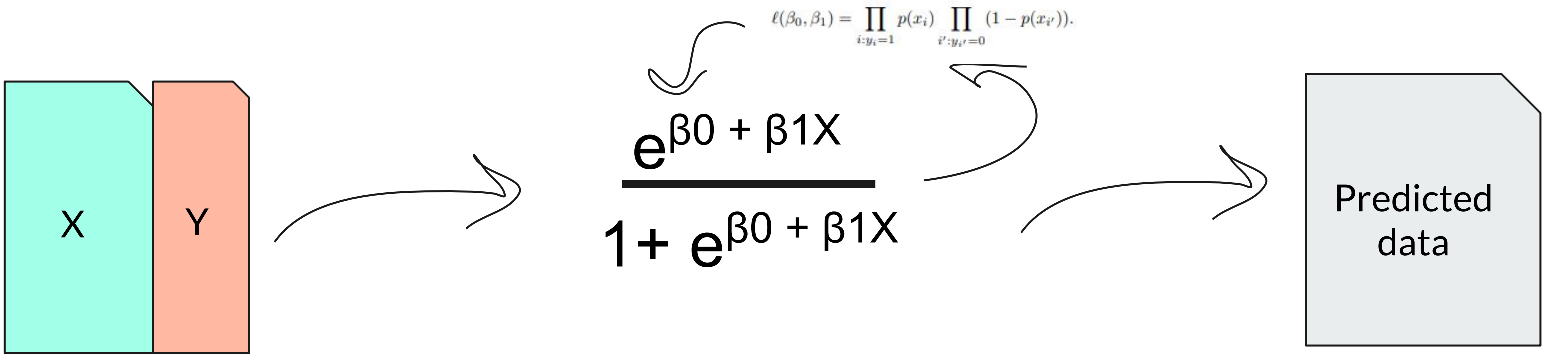
I have data, but I have no clue as to which classes I'm seeing.

Wow! Classes (I think)!

Supervised Learning - Training Methods

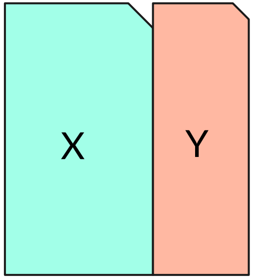
Until week 8, we will be focusing on **supervised learning methods** and various case studies.

Let’s look at a high-level version of the supervised learning logistic regression algorithm.



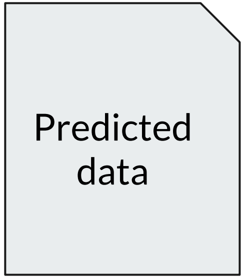
Supervised Learning - Training Methods

We start with our predictors (X) and our known targets (y)

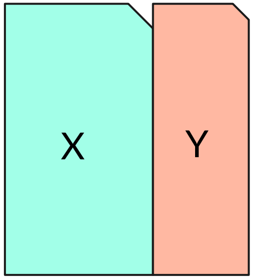


$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

$$\frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$



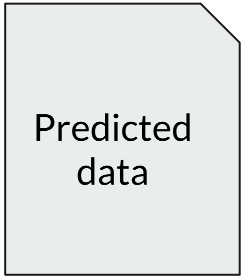
Supervised Learning - Training Methods



$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

$$\frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

We assume that this function is sufficient to model the decision boundary!



Supervised Learning - Training Methods

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

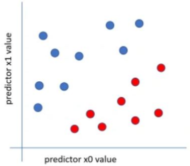
X

Y

$$\frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

Predicted data

The model that we use has to be a competent decision made by a **human**



Supervised Learning - Training Methods

Let’s say that we start with a guess for our weights (or coefficients). We’ll assume B0=0 and B1 = 1

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

X

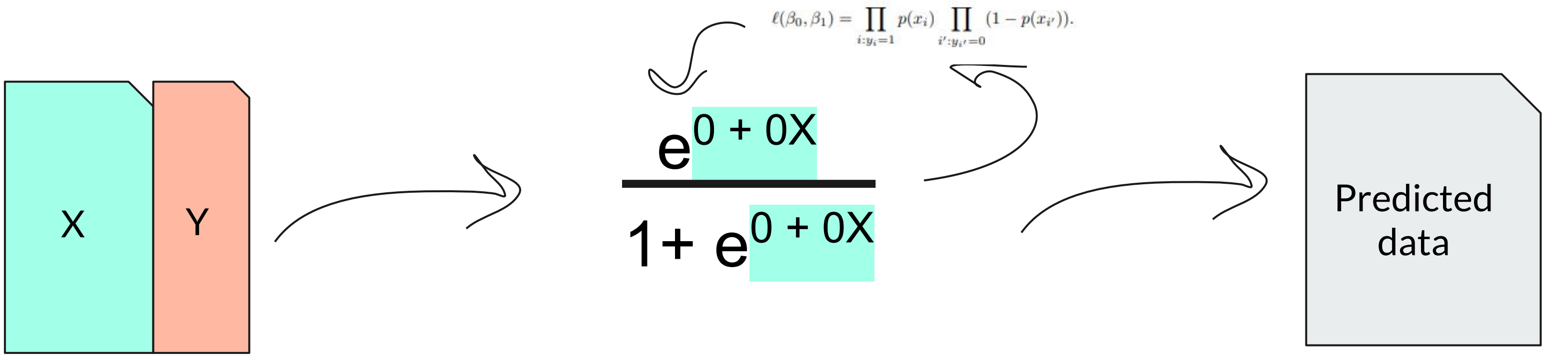
Y

$$\frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

Predicted data

Supervised Learning - Training Methods

Of course, this will result in
poor predictive performance.
**However the model does
not know that yet!**



Supervised Learning - Training Methods

It will calculate the MLE
of these coefficients…

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

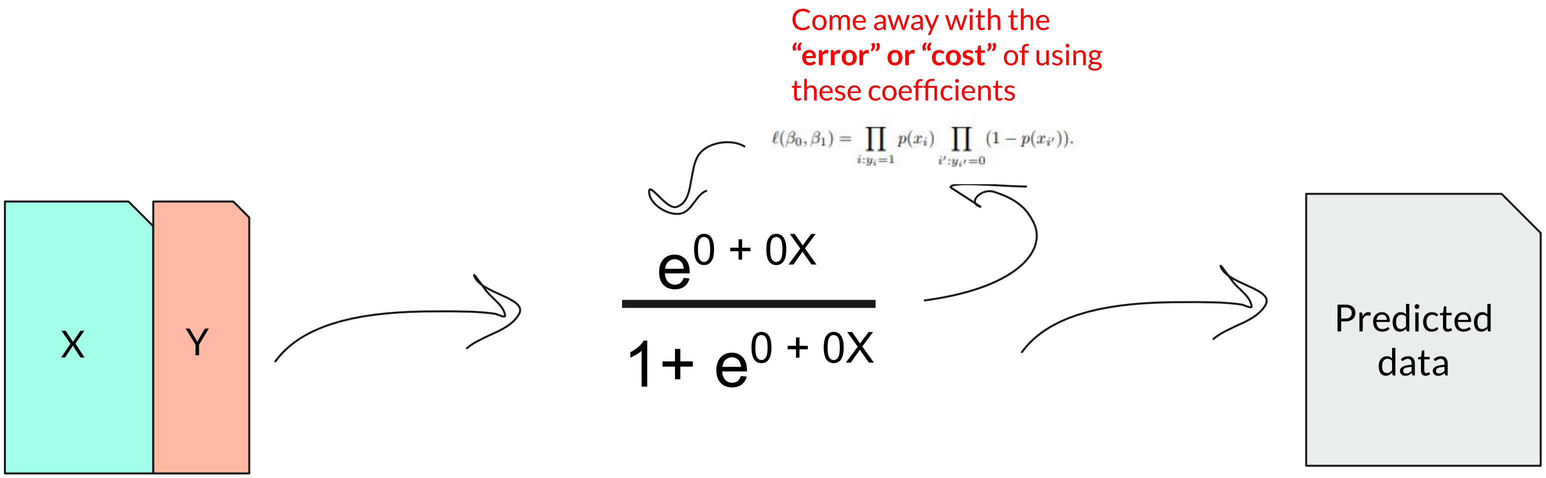
X

Y

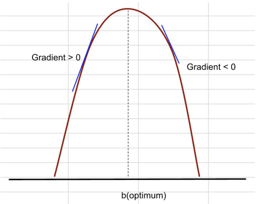
$$\frac{e^{0 + 0X}}{1 + e^{0 + 0X}}$$

Predicted data

Supervised Learning - Training Methods



Supervised Learning - Training Methods



And utilize calculus to **push these** weights to a higher MLE value!*

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

X

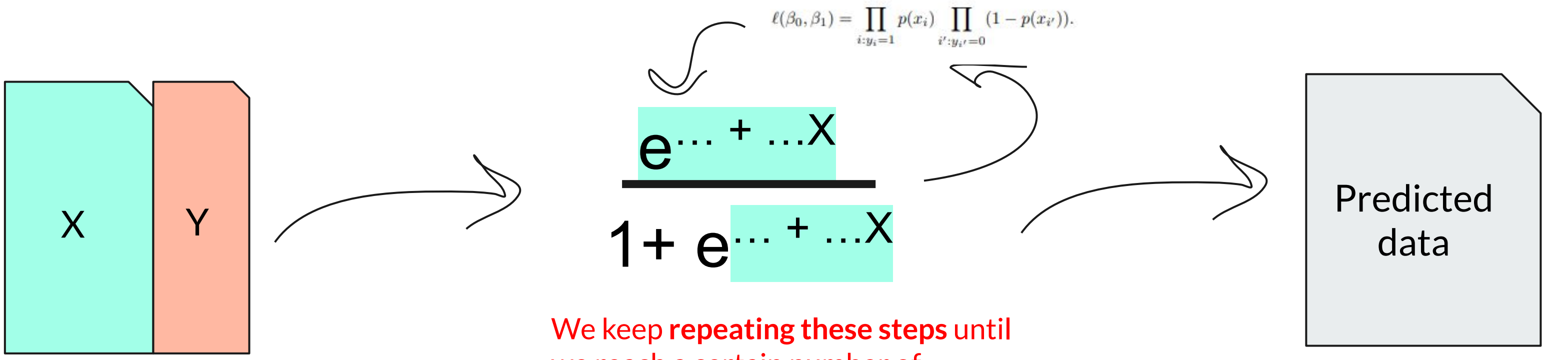
Y

$$\frac{e^{1 + 0.5X}}{1 + e^{1 + 0.5X}}$$

Predicted data

*Keep in mind that this is an extreme over-simplification of the algorithm. You **must** be **familiar** with the maths behind this to be a competent data-scientist. This high-level overview is just a refresher.

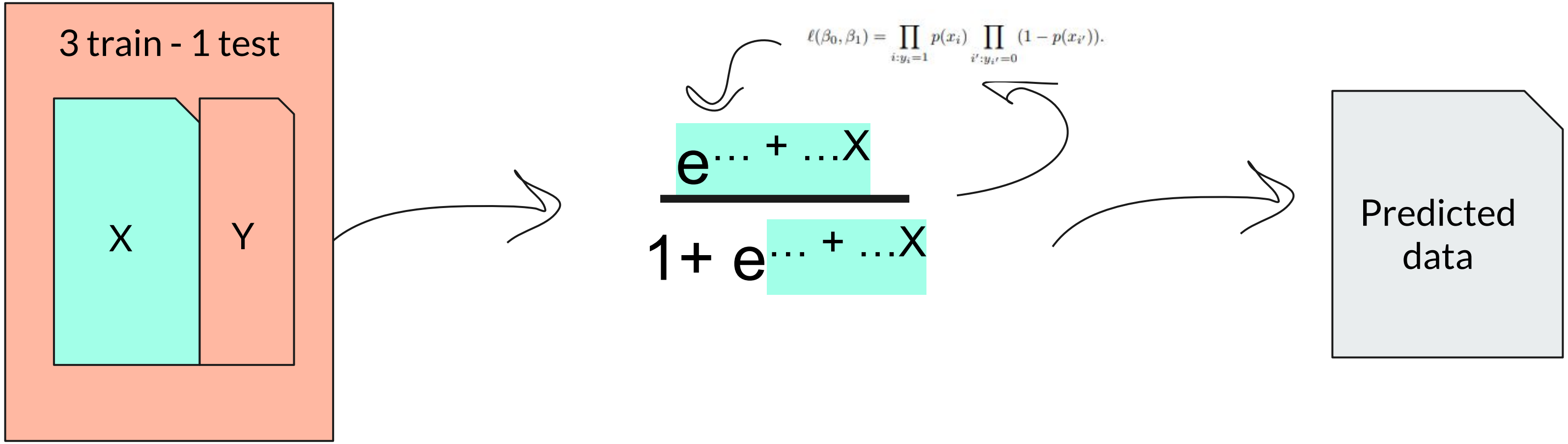
Supervised Learning - Training Methods



We keep **repeating these steps** until we reach a certain number of iterations, or we **simply cannot improve our cost further**

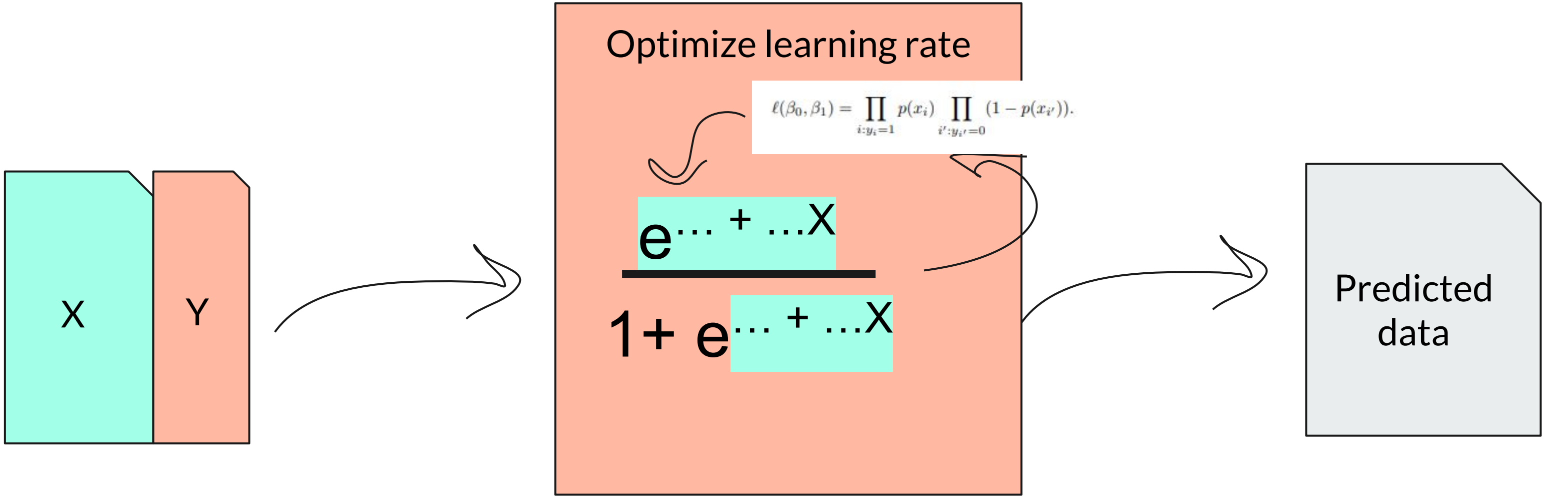
Supervised Learning - Training Methods

Beyond this, we have methods for cross-validation…



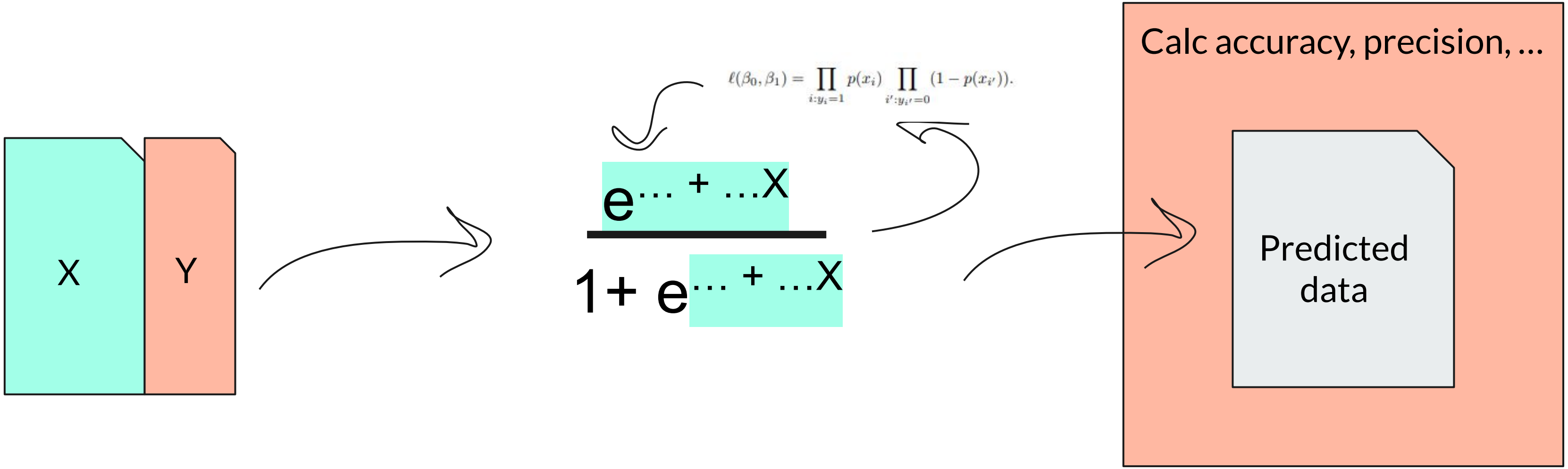
Supervised Learning - Training Methods

Hyperparameter tuning…



Supervised Learning - Training Methods

And lastly, model evaluation



Supervised Learning - Applications

We want to keep in mind that both types of ML have their utility in practical business and research applications! **No one methodology is superior.** When deciding which type of algorithm to apply, consider the following practical questions:

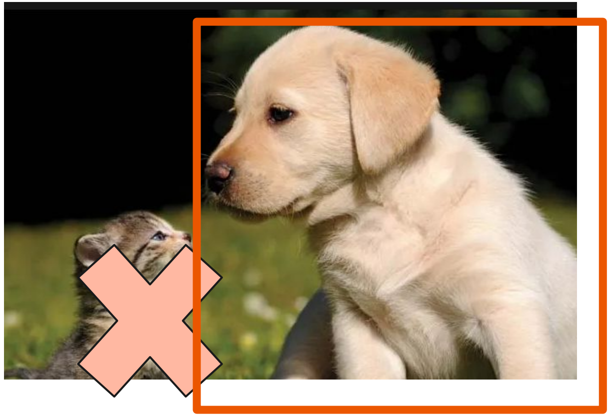
- Is my data **labeled**?
- Are my goals **predictive** or **exploratory**?
  - When we have something to **predict**, we usually want to opt for **supervised learning**. However, if your aims are to **explore structure and patterns**, then perhaps unsupervised learning should be your goto.
- Do I have enough **supplemented or domain knowledge** on my **target variable**?
  - Was this target selected because it was available, or was this target selected for intentional reasons?

Classification Review

Classification: Binary Categories

In the context of **logistic regression (and naive bayes!)**, we always consider a binary classifier to be an "on and off" switch. **Is something present or is it not?**

- Fraud vs **not fraud**
- **Malignant** vs **nonmalignant**
- **Human** vs **not human**
- **Dog** vs **not dog**



Is this a dog or not a dog?

<- A *not*-dog

$$Y_{pred\_dog} = \begin{cases} <=0.5 \rightarrow Not\ dog \\ >0.5 \rightarrow Dog \end{cases}$$

kibble_grams	noise_dB	dog
200	40	0
250	60	1
115	45	0
300	80	1

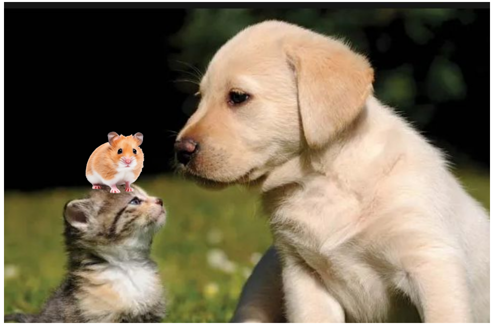
In binary classification, we round the calculated probability to either a 1 or 0 using a cutoff of 0.5. For example, **0.6 will be rounded to 1, therefore belongs to the "dog" class**. Anything **below 0.5 will belong to the "not dog" class.**

Classification: Multiple Categories

Let's say we wanted to create a classification system that can identify if an image contains a **dog, cat, or hamster (K=3)**. Let's assume that we can only use binary classifiers.

How can we use binary classifiers to identify **multiple classes**?

Feel free to mention different ideas until something sticks.

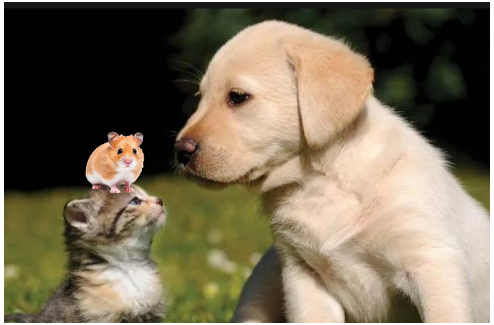


Classification: Multiple Categories

Let's say we wanted to create a classification system that can identify if an image contains a **dog, cat, or hamster (K=3)**. Let's assume that we can only use binary classifiers.

How can we use binary classifiers to identify **multiple classes**?

Feel free to mention different ideas until something sticks.

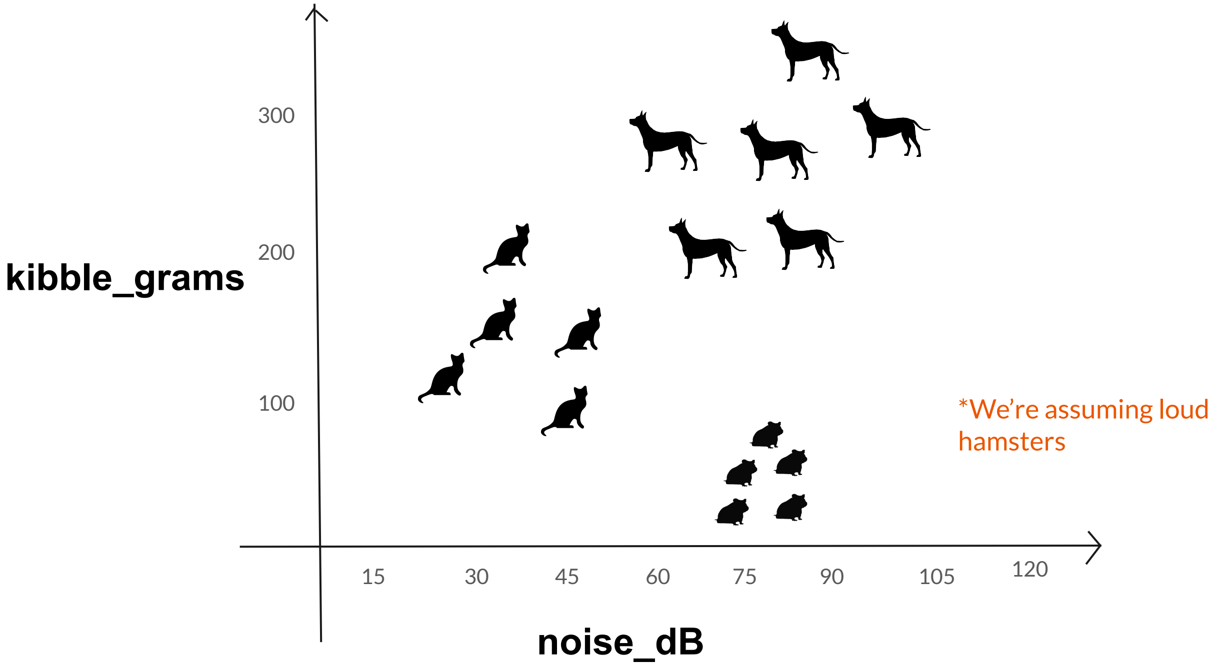


Maybe try multiple classifiers???

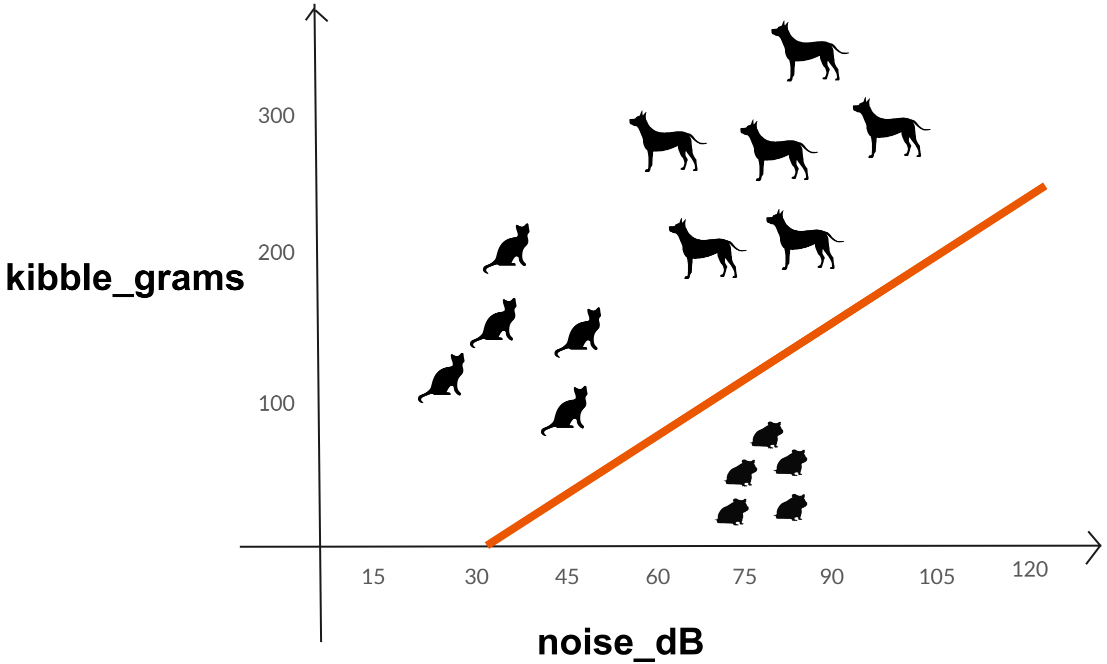
$$Y = \{ ????$$

kibble_grams	noise_dB	dog	cat	hamster
200	40	0	1	0
250	60	1	0	0
115	45	0	1	0
300	80	1	0	0
50	75	0	0	1

Let's assume the following dataset. **Note** : some classifiers require this vector of 1's and 0's.



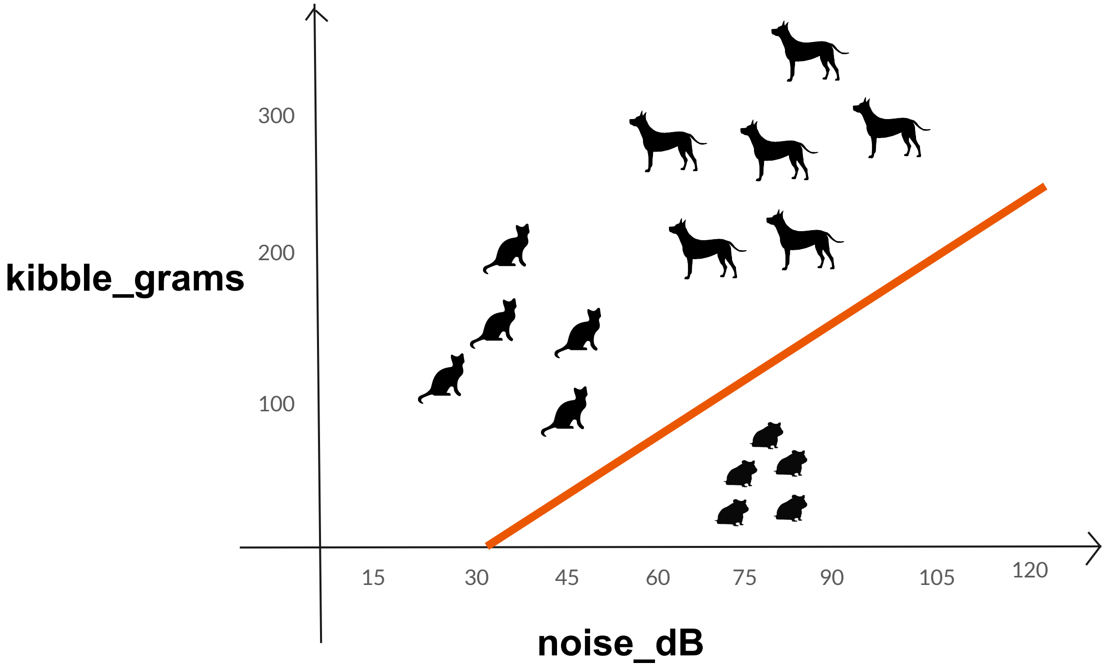
Which we then plot to this scatter-plot.



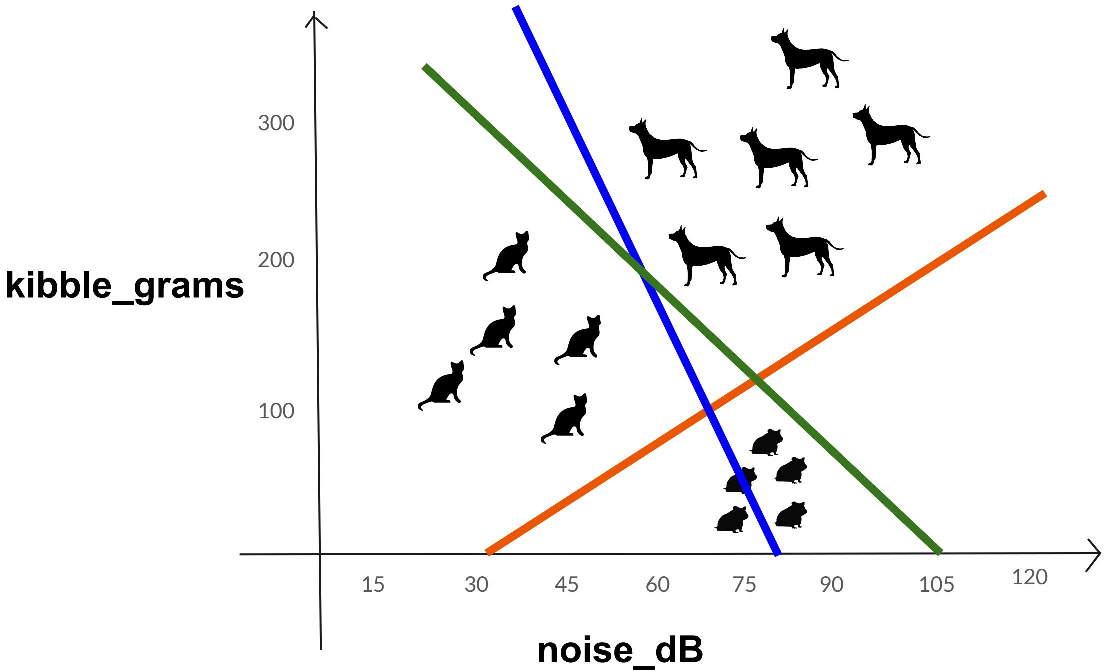
$$\text{Logistic}_{\text{Hamster}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

Let's consider the following, we create a logistic regressor that only looks for hamsters. **All other data-points will just be non-hamsters.** This gives us **one decision boundary**.

$$\text{Logistic}_{\text{Hamster}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$



Considering we made one decision boundary, then how many logistic regressors should we create for all 3 classes?

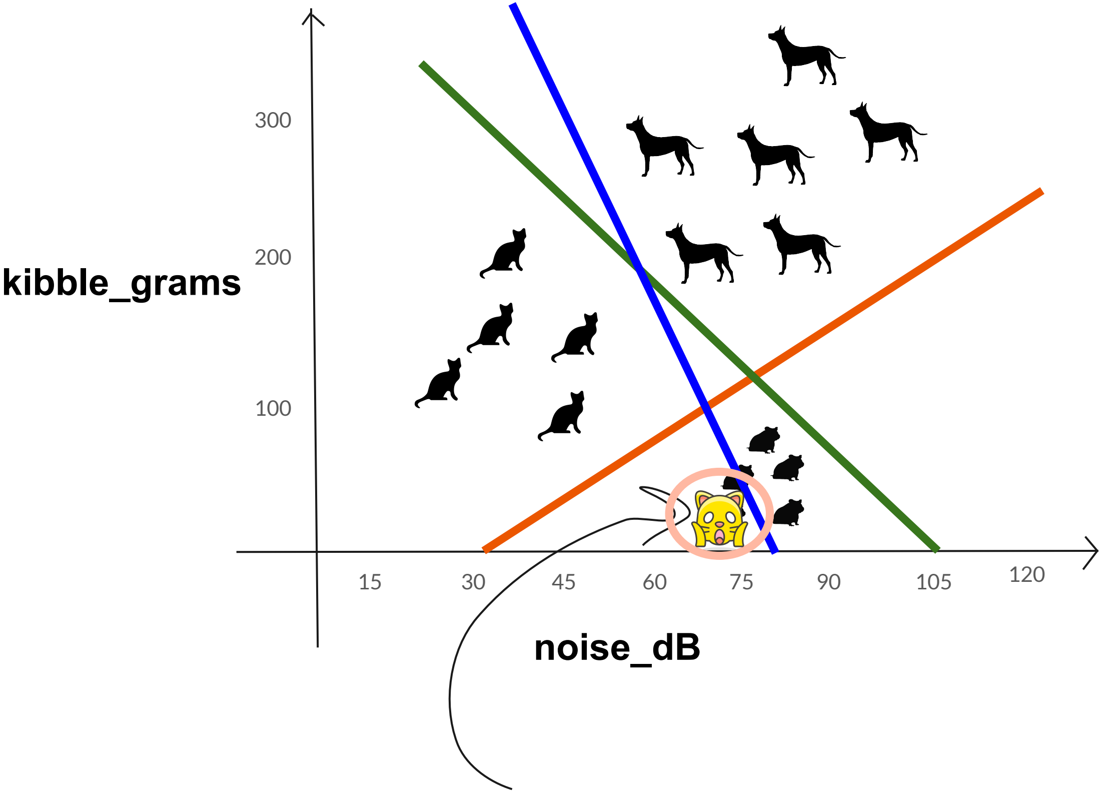


$$\text{Logistic}_{\text{Hamster}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

$$\text{Logistic}_{\text{Cat}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

$$\text{Logistic}_{\text{Dog}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

If we want to classify 3 classes, then we should create 3 binary classifiers! That is, for *K classes we make K binary classifiers*! This methodology is called **one-versus-all**. There is another technique called **one-versus-one classification which we will explore later** (but generally applies the same idea).

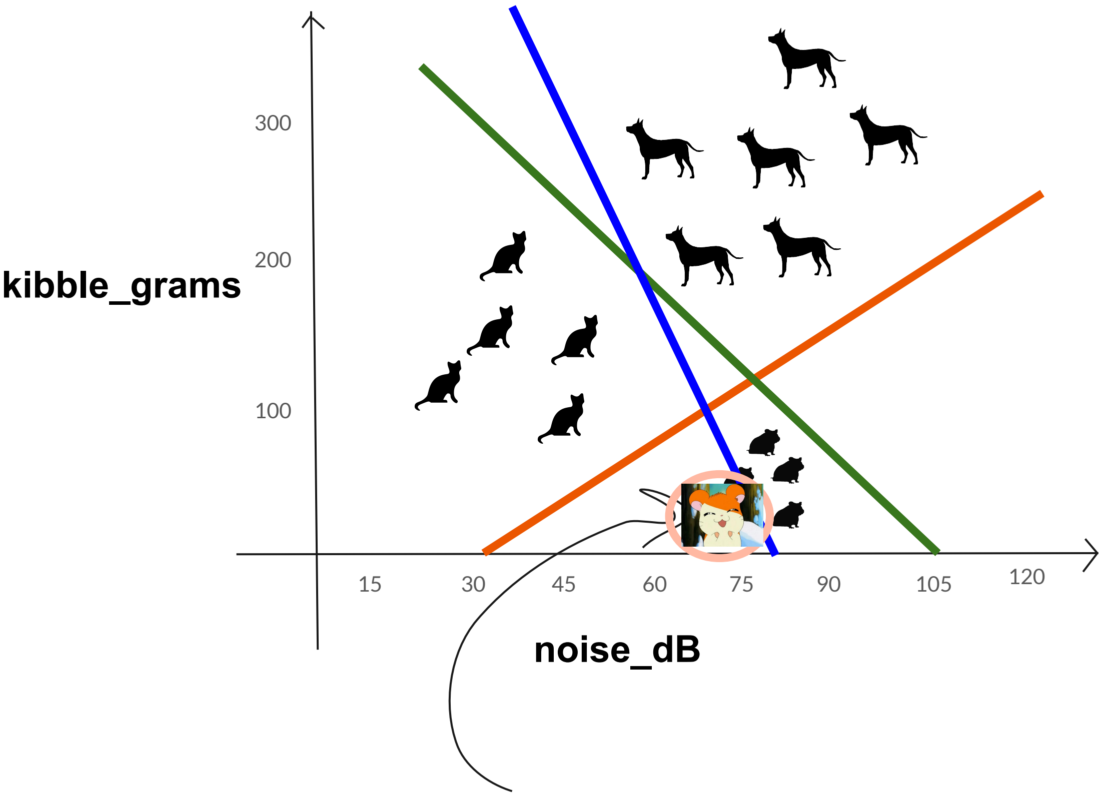


$$\text{Logistic}_{\text{Hamster}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

$$\text{Logistic}_{\text{Cat}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

$$\text{Logistic}_{\text{Dog}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

You might be thinking, “Hey (Mickal/Farukh), what about our poor misclassified hamster friend? **Does it become a cat?”**

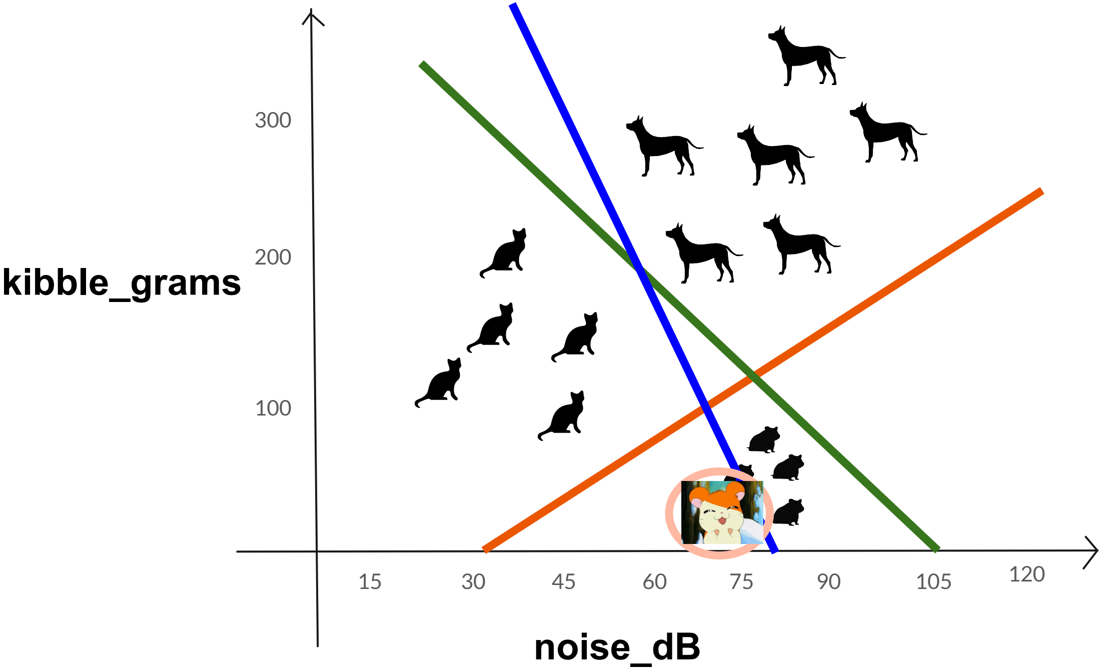


$$\text{Logistic}_{\text{Hamster}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

$$\text{Logistic}_{\text{Cat}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

$$\text{Logistic}_{\text{Dog}} = \frac{e^{\beta 0 + \beta 1X}}{1 + e^{\beta 0 + \beta 1X}}$$

No! The beauty of this method is that we assign classes based on the probabilities. Therefore if we have a more confident prediction for hamster, then we will likewise classify it as a hamster!



$$\text{Logistic}_{\text{Hamster}} = \frac{e^{\beta 0 + \beta 1 X}}{1 + e^{\beta 0 + \beta 1 X}}$$

$$\text{Logistic}_{\text{Cat}} = \frac{e^{\beta 0 + \beta 1 X}}{1 + e^{\beta 0 + \beta 1 X}}$$

$$\text{Logistic}_{\text{Dog}} = \frac{e^{\beta 0 + \beta 1 X}}{1 + e^{\beta 0 + \beta 1 X}}$$

This could be interpreted two ways:

- We assign classes based on how many times a model "voted" for a specific classification
- We assign classes based on "highest probability" that something belongs to a class.

Classification: One vs One; One vs Rest

This describes a very **specific strategy** to classify your dataset when you have multiple classes. There are two ways we can implement this.

One vs Rest

- Create one classifier **per unique class**, with samples of that class being considered positive cases, and all other classes considered negative.

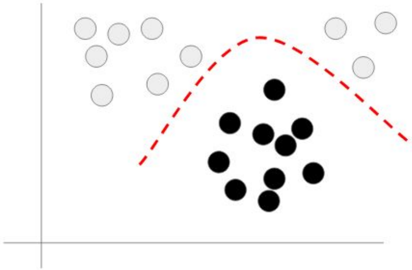
One vs One

- Create one classifier for **every pair of classes**. If there are "n" classes, you will train ($n*(n-1)/2$) classifiers. We decide on tie-breakers by seeing how many times a sample is classified as a specific class. The class with the most "votes" wins.

Classification: One vs One; One vs Rest

Pros

- Simple implementation. More classes → More models!
- Can model a **non-linear decision boundary**. This is a term that we will explore more tomorrow.



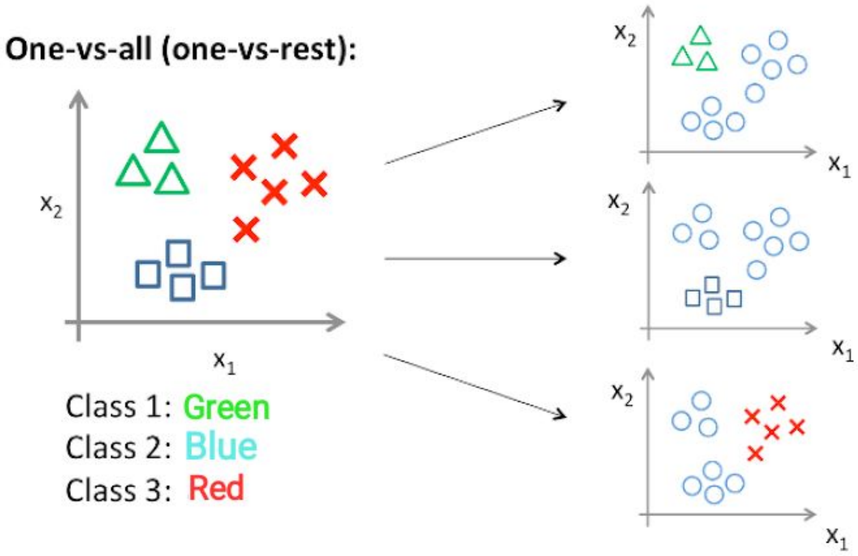
However, there are negative consequences to this as well…

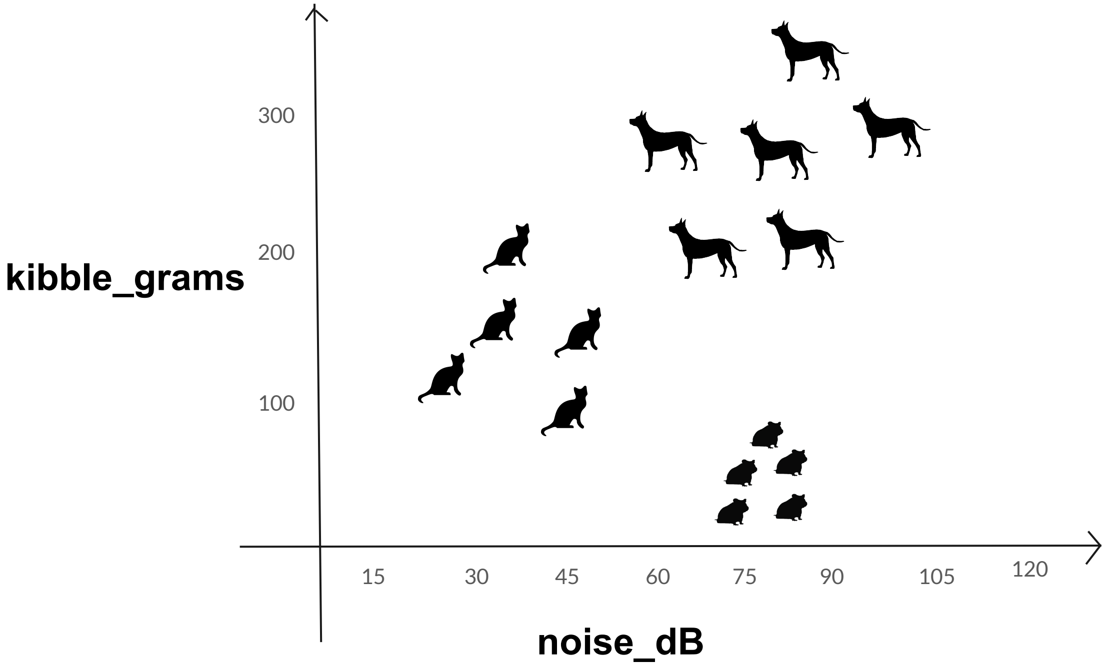
Classification: One vs. All Binary Classifiers

In the **one-vs-rest** approach, **we train *K* binary classifiers,** **each time comparing one of the *K* classes to the remaining *K-1* classes.**

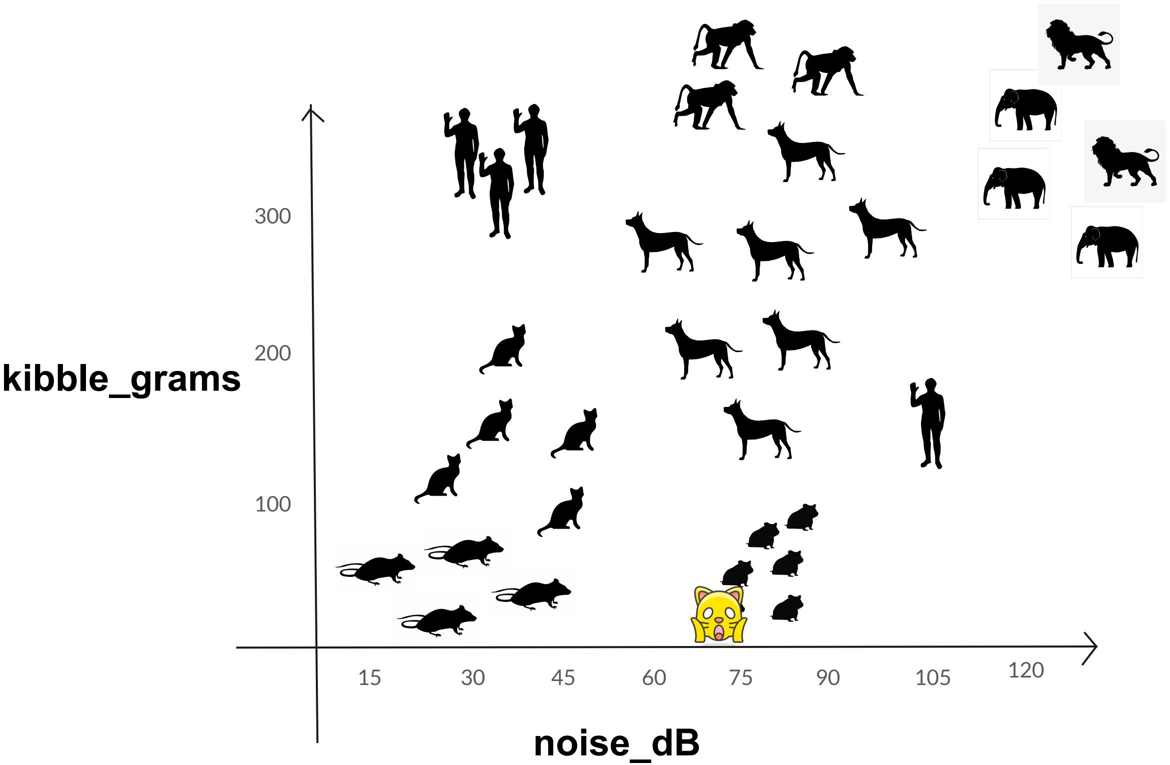
We assign observations to class for which has the highest probability, as "this **amounts to the highest level of confidence that the test observation belongs to the kth class.**"

But what are the down-sides to this?

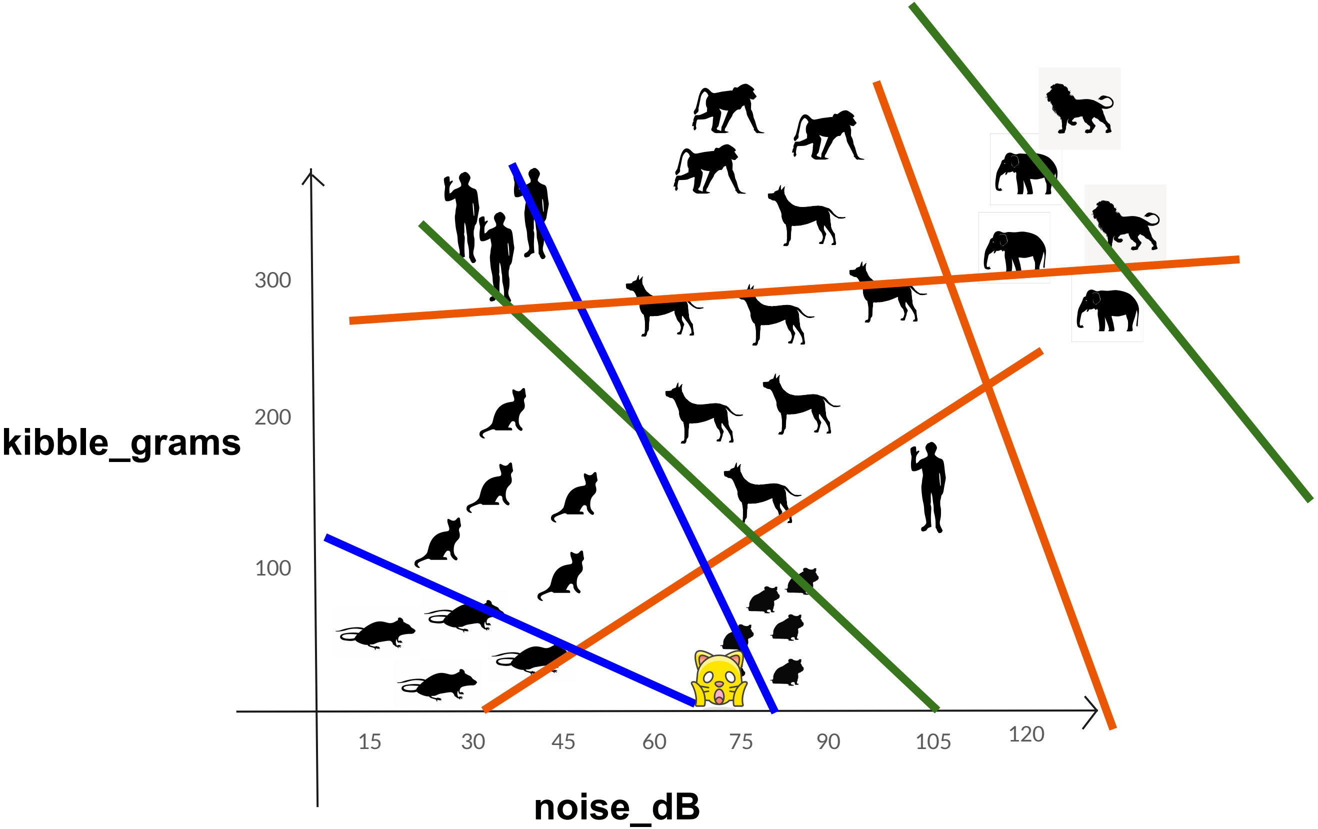




What if I don’t have 3 classes, but instead…



What if I don’t have 3 classes, but instead…100 CLASSES



This will necessitate that we create 100 models. Keep in mind that creating machine learning models is **expensive in terms of time and resources. This becomes more applicable as we move to enterprise environments.** We would much rather prefer a cheaper alternative.

Classification: Multiple Categories

This “cheaper alternative” are inherently **multiclass classifiers** (kNN, decision trees). This allows us to have **one model for identifying multiple classes.** Some examples include:

- **Digits:** (0,1,2,3,4,5,6,7,8,9)
- **Flower-type:** (rose, tulip, petunia)
- **Dog, cat, or hamster**

Is this a dog, a cat, or a hamster?

Y = [prob_dog, prob_cat, prob_ham]

kibble_grams	noise_dB	animal
200	40	cat
250	60	dog
115	45	cat
300	80	dog
50	75	hamster

We want to use multiclass classifiers when we have K>2 classes, and we assume that **each sample can only have one class**. Sometimes, multiclass classifiers output probabilities.

kibble_grams	noise_dB	animal
200	40	cat
250	60	dog
115	45	cat
300	80	dog
50	75	hamster

$$Y = \begin{cases} \textit{dog} \\ \textit{cat} \\ \textit{hamster} \end{cases}$$

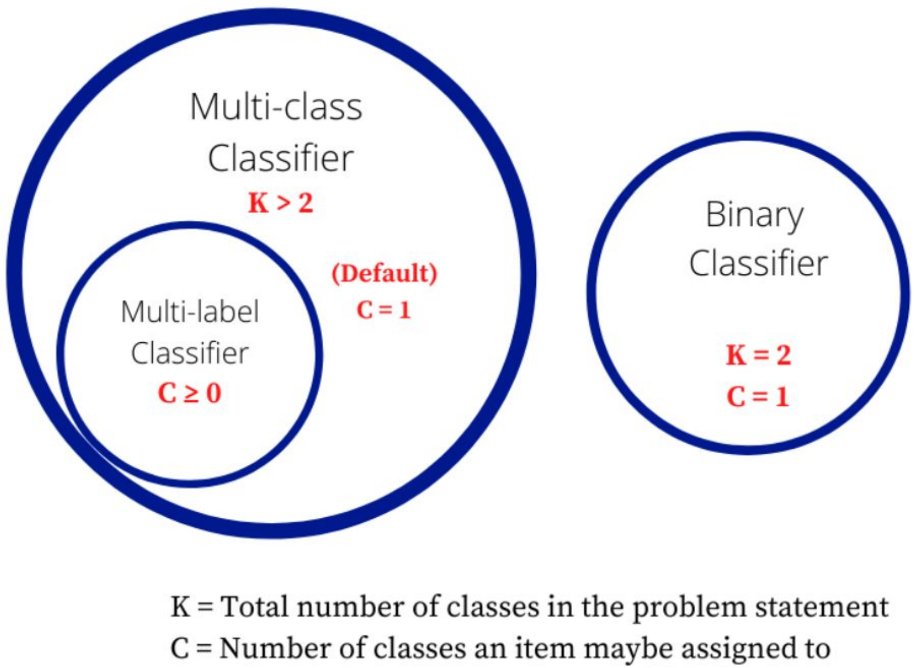
Other times they output discrete classes. Keep in mind however that most models **support both outputs.**

Classification: Multilabel Classification

Finally, there is one more type of classification, called **multi-label classification**. Once again, we also calculate probabilities of a sample belonging to a class, **however this time we assume that samples can belong to multiple classes!**

- **Movie genre:** *horror, horror & comedy, …*
- **Computer Vision:** *dog, dog & cat, …*
- **Document Org:** *law, law & financial, …*

Which animals does this photo contain?



A good way to internalize this, is the fact that multi-label classifiers are a variant of multi-class classifiers, except that we permit **0 or more classes** to belong to one sample!

Bayes Theorem

Bayes Theorem - Frequentist World-View

So far, in our discussion of statistics and probability, we have assumed a **frequentist world-view**.

That is, **we define probability as simply a long-run frequency.** For example, do you want to figure out the probability of flipping heads on a quarter? Simply run an experiment!

number of flips	11	12	13	14	15	16	17	18	19	20
number of heads	8	8	9	10	10	10	10	10	10	11
proportion	.73	.67	.69	.71	.67	.63	.59	.56	.53	.55

Bayes Theorem - Frequentist Pros

We like this world-view because it is:

- **Objective:** *we run experiments to figure out the world*
- **Unambiguous:** *you and I will calculate the same probability from one experiment*

Bayes Theorem - Frequentist Cons

But, there are also negatives to this world-view:

- **No use of prior information:** *when running our experiment, we close ourselves off from any prior information.*
- **Inflexible:** *Frequentists rely on large amounts of data to make statements on probability.*

Bayes Theorem - Frequentist Cons

To further explain the limitations of the frequentist perspective, let's say I give you a coin. **I tell you that there is a 30% chance that this coin lands on heads.**

From our previous perspective, how do we use this information? Well we basically chuck it in the trash.

Bayes Theorem - Bayesian Statistics

Enter **Thomas Bayes**.

He introduced probability as a **degree of belief** to which a “rational agent” assigns a truth to an event…

…as opposed to a ratio of experiments.

This **degree of belief is updated with new information**. Let’s see an example…

Full-time minister, part-time statistician

*MidJourney Image of Hamster Flipping a Coin

You're approached by a hamster with a quarter and a proposition. **Every time the coin lands on heads, you have to give him a carrot. Every time the coin lands on tails, he gives you a carrot.**

First off, let's understand what assumptions we've made about this game. What is your "assumption" about the probability of this being a fair game?

Let’s assume we don’t believe this hamster is nefarious, but since we don’t personally know this hamster, we have an inkling of suspicion.

We assume 98% chance of fairness
We’ll keep this “belief” in mind.

P(Hyp) = 0.98

An Aside - A Fair Coin

Let’s rehash what a “fair coin” means.

We **assume that the chance of a coin landing on the heads side**, is **equal** to the chance of it **landing on the tails side.**

Since there are only 2 possible outcomes. What do we say is the probability of landing on heads and tails???

P(Heads) = ? P(Tails) = ?

An Aside - A Fair Coin

Let’s rehash what a “fair coin” means.

We **assume that the chance of a coin landing on the heads side**, is **equal** to the chance of it **landing on the tails side.**

Since there are only 2 possible outcomes. What do we say is the probability of landing on heads and tails???

P(Heads) = 0.5 P(Tails) = 0.5

An Aside - A Fair Coin

This probability of 98% that we are using is a bit of a *meta-probability.*

From a *frequentist* point of view, we are basically stating that out of 100 times that this hamster will ask us to play this game, 98 times will be played with a fair coin, whereas 2 times will be played with a biased coin.

It’s a stretch of imagination, but bear with us.

P(Heads) = 0.5 **P(Tails) = 0.5**

An Aside - A Fair Coin

Figuring out what this biased coin will behave like is out of scope of this thought experiment,

so let's just assume that if this hamster is tricking us, it is using a coin that is biased towards heads.

$$P(\text{Heads}) = 0.8 \qquad P(\text{Tails}) = 0.2$$

P(Hyp) = 0.98

First round the hamster flips heads.

Do we have reason to doubt our initial assumption yet?

Probably not.

P(Hyp) = 0.98

After 4 rounds the hamster keeps flipping heads.

Do we have reason to doubt our initial assumption yet?

*MidJourney Image of Nefarious Hamster Flipping a Coin

Most likely yes! But which new probability do we use for the probability of heads?

The frequentist framework does not give us an easy answer.

Bayes, however, **provides a formula for updating beliefs with evidence**!

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(E)}$$

Here's the formula, let's **break down its components** before going through our hamster example again

Remember, the "|" is the symbol for conditional probability statements.

"Hypothesis given Event"

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(E)}$$

The probability the **hypothesis** is true given the **event** is equal to

Probability of the hypothesis occurring (also known as the **prior belief**!)

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(E)}$$

The probability the **hypothesis** is true given the **event** is equal to

Probability of the hypothesis occurring (also known as the **prior belief**!)

Probability of seeing **event** given that the **hypothesis** is true (also known as the **likelihood***)

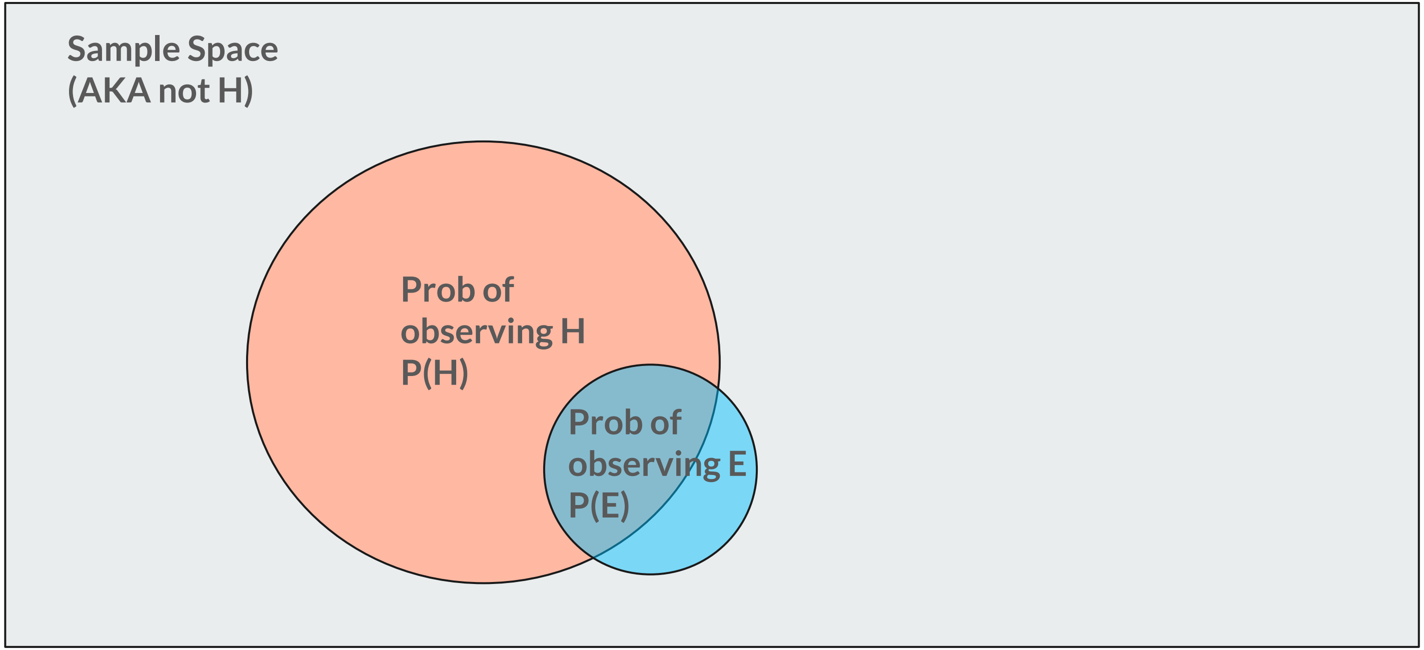
$$P(H|E) = \frac{P(H)\ P(E|H)}{P(E)}$$

The probability the **hypothesis** is true given the **event** is equal to

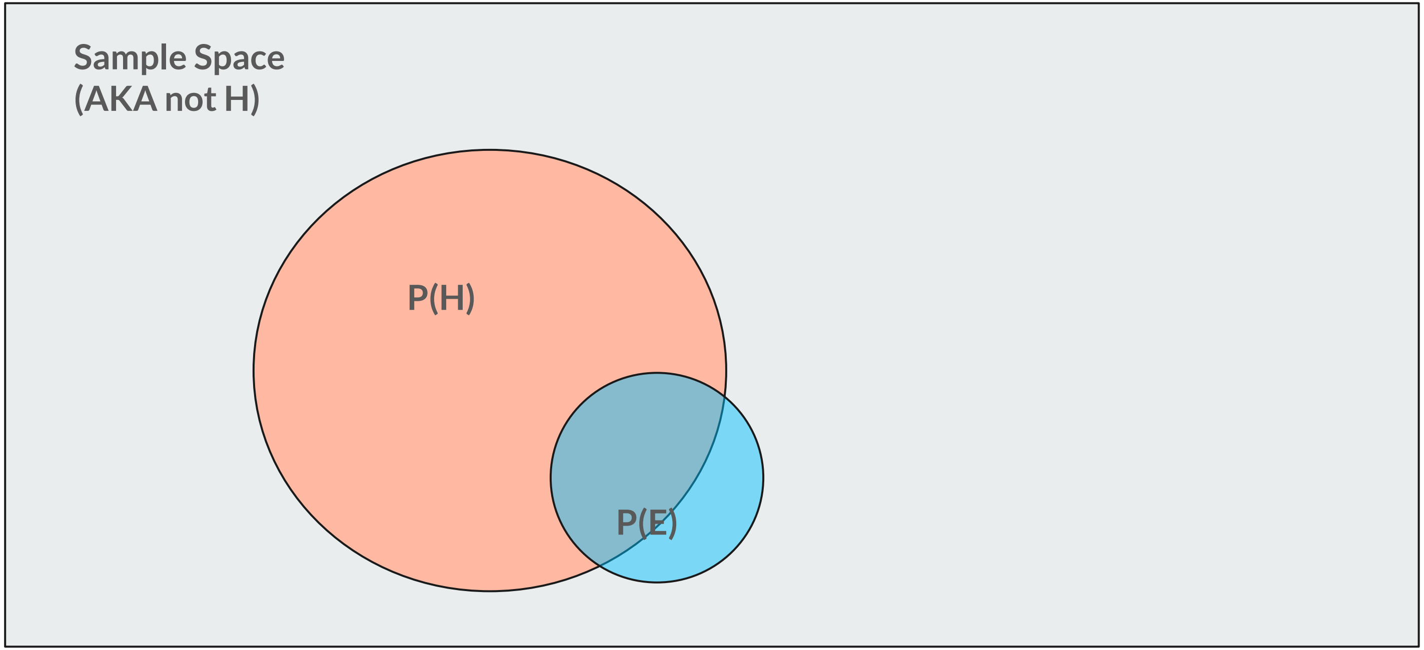
***note that probability and likelihood are not the same thing!**

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(E)}$$

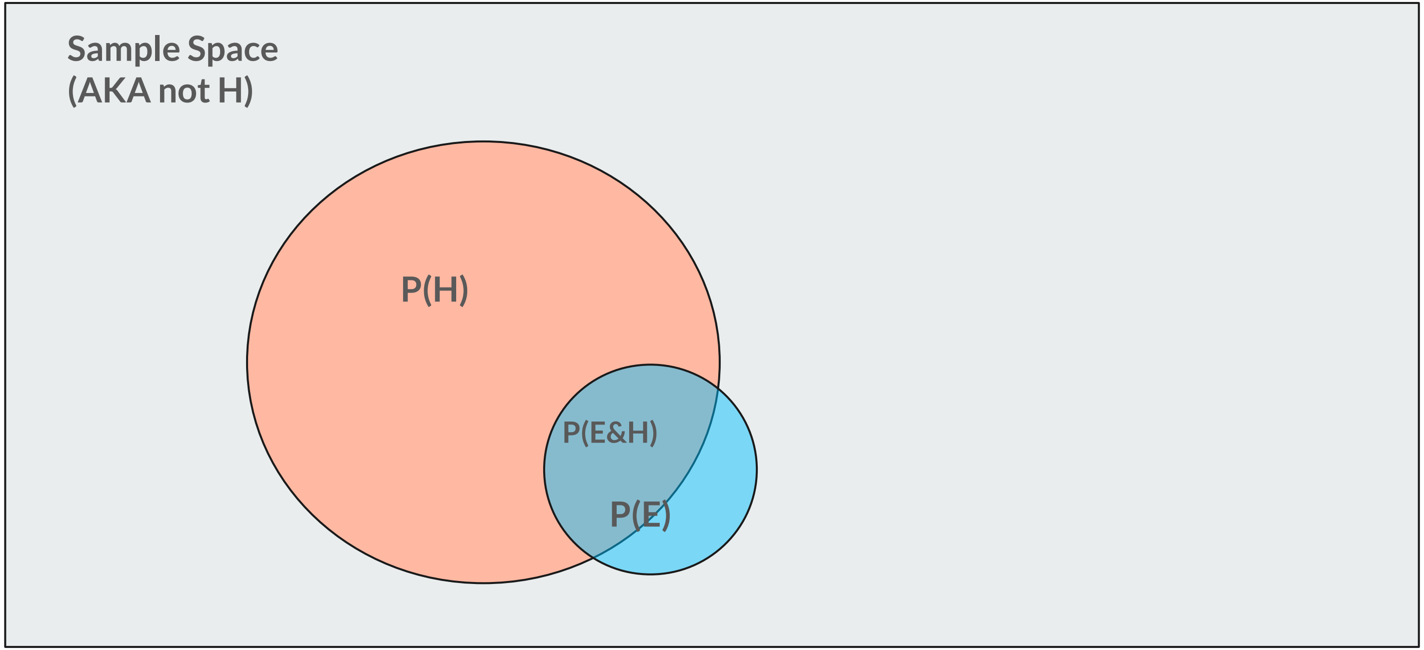
The probability of **event** occurring. This part is sometimes deceptively simple. Keep in mind that we want to consider the **event occurs given the hypothesis is true and given the hypothesis is false**!



Let’s understand how to calculate this. I find diagrams to be the most helpful in visualizing this calculation

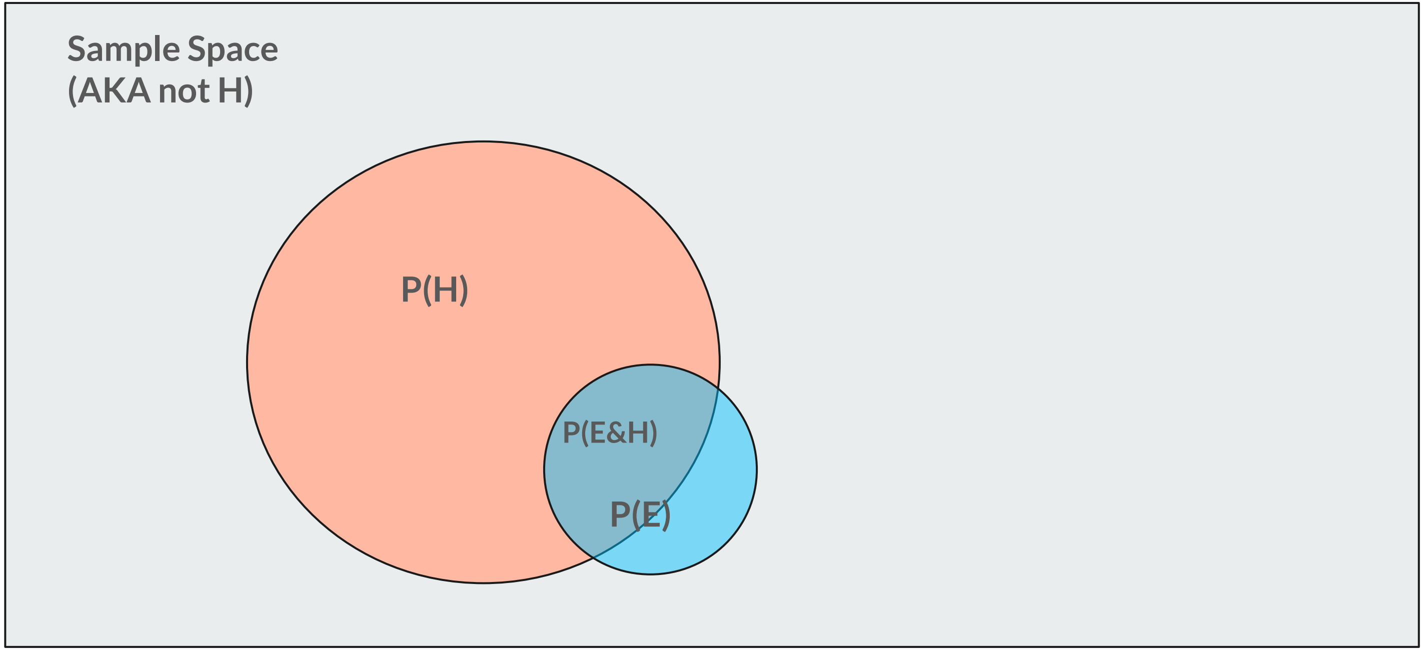


To calculate P(E), we have to consider the intersection of P(E) and P(H), as well as the space where P(E) exists outside of P(H)



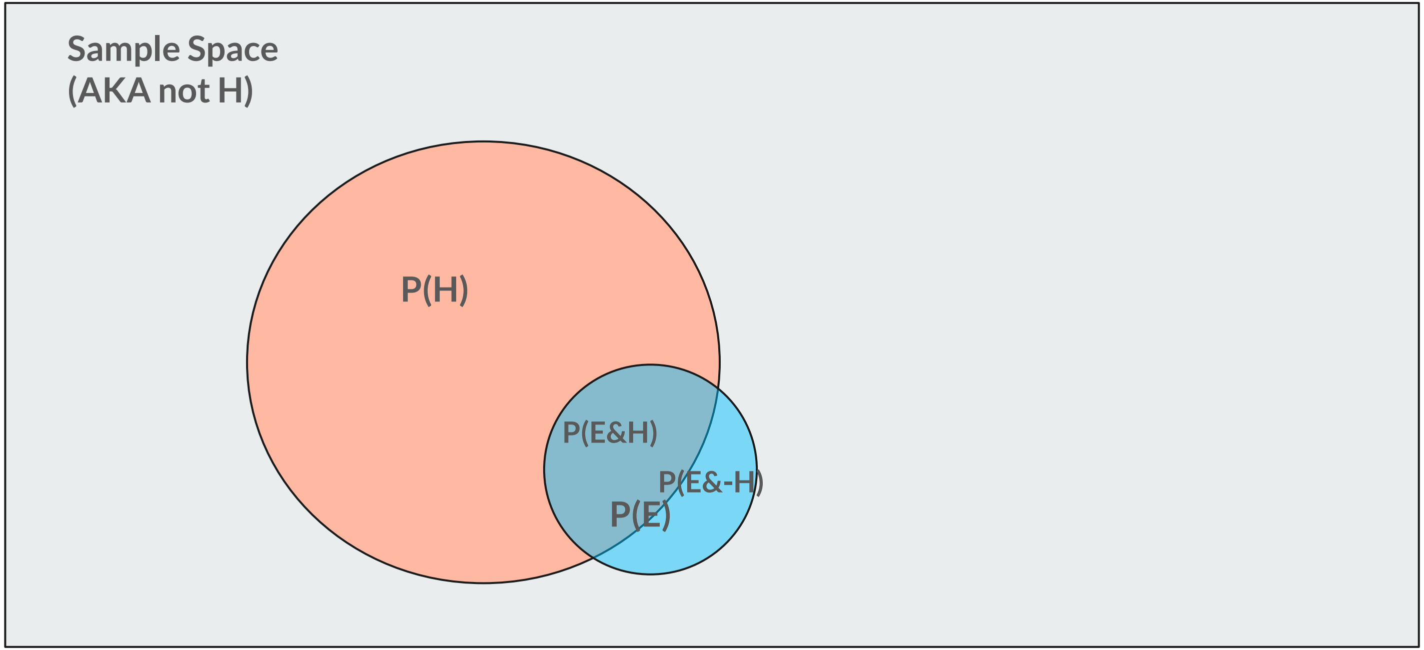
This intersect can be labeled **P(E & H)**

P(H)P(E|H) +



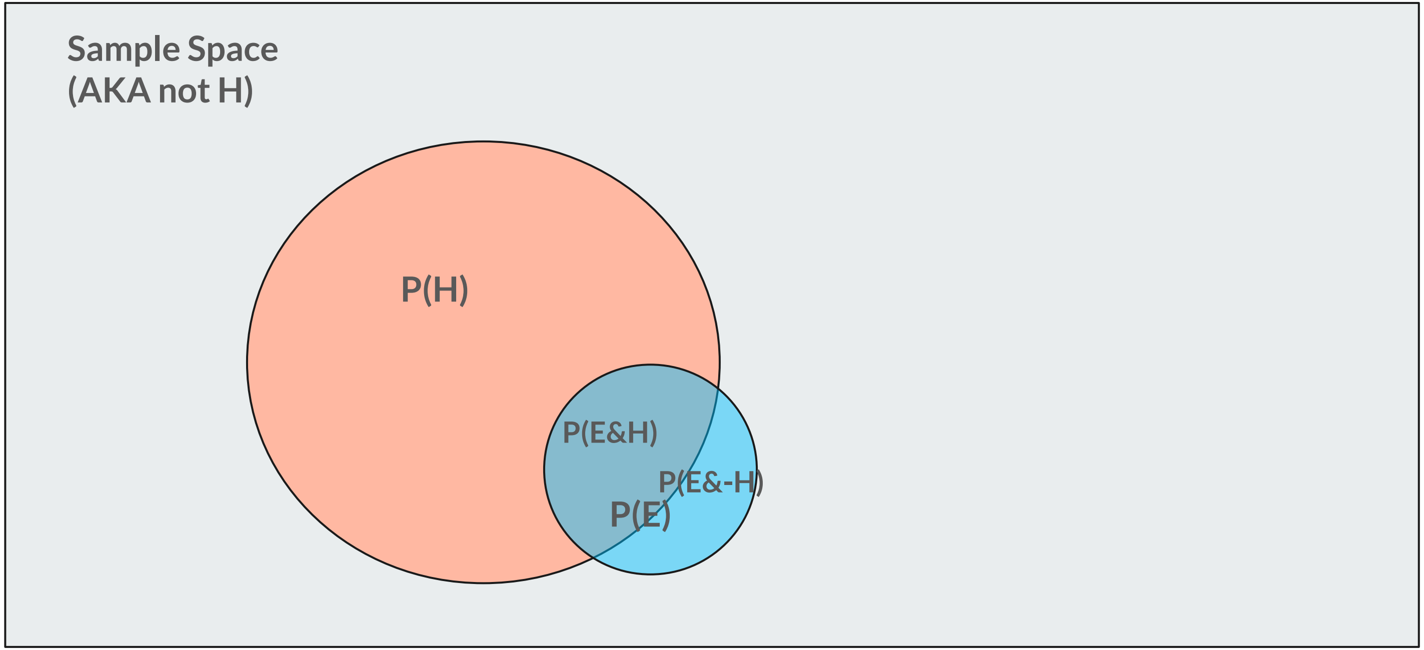
In terms of conditional probability, that is **P(H)P(E|H)**

P(H)P(E|H) +



And lastly we have **P(E & -H).** Can anyone express this in terms of conditional probability as well?

P(H)P(E|H) + P(-H)P(E|-H)



P(-H)P(E|-H). This allows us to express P(E) as P(H)P(E|H) + P(-H)P(E|-H)

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(H)P(E|H) + P(-H)P(E|-H)}$$

This is expressed as the sum of the probability of the **hypothesis is true and the event occurs given the hypothesis is true**, with the **probability the hypothesis is false and the event occurs given the hypothesis is false.**

$$P(H|E) = \quad P(H)\ P(E|H)$$

Let’s summarize this: the probability of a hypothesis given an event is our **previous evidence** multiplied by the **probability of the event occurring ASSUMING OUR HYPOTHESIS IS TRUE**.

$$P(H|E) = \frac{P(H)\ \ P(E|H)}{P(H)P(E|H) + P(-H)P(E|-H)}$$

Divided by **the probability of the event occurring when the hypothesis is true or not true!**

Let's apply this framework to our hamster experiment

P(H|E) = Likelihood that hypothesis (not scam) is true

P(H) = Our prior understanding of the hypothesis

P(E|H) = Likelihood of flipping heads given belief that the coin is fair

P(E|-H) = Likelihood of flipping heads given belief that the coin is unfair

P(-H) = Chance hypothesis is NOT true (1-P(H))

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(H)P(E|H) + P(-H)P(E|-H)}$$

With this new formula in mind, let's see how our probability of us **landing on tails** updates with each additional flip from our scam-hamster. **I switch some letters around**:

H = "Coin is fair"
E = "Outcome of the bet"

One challenge however is, what is our "prior" belief that the coin is fair?

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(H)P(E|H) + P(-H)P(E|-H)}$$

For our “prior” we usually use the most available evidence of our probability. Let’s bring back our prior assumption of 0.98.

$P(H) = 0.98$

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(H)P(E|H) + P(-H)P(E|-H)}$$

$P(H) = 0.98$

Let's play the game again. First round, hammy wins. We reach in our pocket and give it a loose baby carrot.

Let’s recalculate our probabilities. Let’s see if we can calculate these ourselves.

P(H) = 0.98
P(E|H) = Probability of heads given fair coin

P(-H) = Probability of unfairness
P(E|-H) = Probability of heads given unfair coin

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(H)P(E|H) + P(-H)P(E|-H)}$$

Let's recalculate our probabilities. Let's see if we can calculate these ourselves.

$P(H) = 0.98$
$P(E|H) = 0.5$

$P(-H) = 0.02$
$P(E|-H) = 0.8$*

*This one is tricky, let's just say we're looking for evidence of a biased coin towards heads, so we use some arbitrary probability that leans towards heads

$$P(H|E) = \frac{(0.98)(0.5)}{(0.98)(0.5) + (0.02)(0.8)}$$

We get 0.96. And this makes sense, we slightly **decreased** the probability that this coin is fair. **We've updated our beliefs with evidence.**

This probability BECOMES OUR NEW PRIOR P(H)

Let's go for another round.

P(H|E) = 0.9683

P(H) = 0.97

Hammy wins again

Let’s recalculate our probabilities. Let’s see if we can calculate these ourselves.

P(H) = 0.97
P(E|H) = Probability of heads given fair coin

P(-H) = Probability of unfairness
P(E|-H) = 0.8*

$$P(H|E) = \frac{P(H)\ P(E|H)}{P(H)P(E|H) + P(-H)P(E|-H)}$$

Let’s recalculate our probabilities. Let’s see if we can calculate these ourselves.

P(H) = 0.97
P(E|H) = 0.5

P(-H) = 0.03
P(E|-H) = 0.8*

$$P(H|E) = \frac{(0.97)(0.5)}{(0.97)(0.5) + (0.03)(0.8)}$$

Our probability drops again!

Just for fun, one last time.

P(H|E) = 0.9528

P(H) = 0.9528

Hammy wins again

Let’s recalculate our probabilities. Let’s see if we can calculate this ourselves.

$P(H) = 0.95$
$P(E|H) = 0.5$

$P(-H) = 0.05$
$P(E|-H) = 0.8$*

$$P(H|E) = \frac{(0.95)(0.5)}{(0.95)(0.5) + (0.05)(0.8)}$$

But wait! What if hammy flips a tails.

P(H) = 0.92
P(E|H) = 0.5

P(-H) = 0.05
P(E|-H) = Assuming the probability of heads given on unfair coin is 0.8, what would be the probability of tails given the unfair coin?

$$P(H|E) = \frac{(0.95)(0.5)}{(0.95)(0.5) + (0.05)(0.8)}$$

Using these values, let's recalculate our conditional probability.

P(H) = 0.92
P(E|H) = 0.5

P(-H) = 0.05
P(E|-H) = 0.2

$$P(H|E) = \frac{(0.92)(0.5)}{(0.92)(0.5) + (0.08)(0.2)}$$

*MidJourney Image of Nice Hamster

It goes back up!!!!!

Since we received evidence that this coin is fair, the probability of a fair coin goes from 92% to 96%! This should be mind blowing.

P(H|E) = 0.9663

Naive Bayes Classifier

We can utilize this formula in a **multiclass supervised learning classifier** called the **naive bayes classifier**.

The goal of naive bayes classification is to calculate **the probability a new sample belongs to a certain class.**

We can take this a step further and assume shapes of our our probability distribution.

We'll go over the details next week

A Refresher on Classification Metrics

Metrics

True Class

	Actual Default	Actual No Default
Default	30	10
No Default	40	70

Predicted

ALL CLASSIFIERS CAN USE THESE METRICS!

Accuracy

Overall **correctness** of the model.

Precision

Proportion of correct **predicted** positive cases.

Recall

Proportion of correctly **identified** positive cases.

Specificity

Ability to **predict negative cases.**

Metrics

True Class

Predicted

	Actual Default	Actual No Default
Default	30	10
No Default	40	70

- To increase **precision**, we must have fewer **False Positives.**
  - However, we do not have to worry about false negatives.
- To increase **recall**, we must have fewer **False Negatives.**
  - However, we do not have to worry about false positives.

Metrics

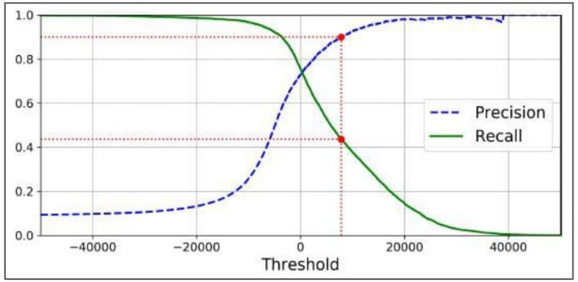
True Class

Predicted

	Actual Default	Actual No Default
Default	30	10
No Default	40	70

For this reason, as we increase precision (proportion of correct predicted), we decrease recall (proportion of correctly identified).

This is known as **precision-recall tradeoff**. We usually want to find the right **balance**.



Metrics

When moving towards multiclass classifiers, our confusion matrix get's a tad bit more complex, but the concept remains the same.

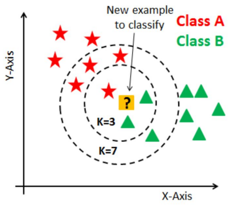
?

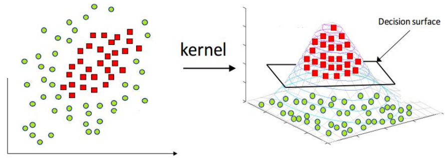
		PREDICTED classification			
	Classes	a	b	c	d
ACTUAL classification	a	TN	FP	TN	TN
	b	FN	TP	FN	FN
	c	TN	FP	TN	TN
	d	TN	FP	TN	TN

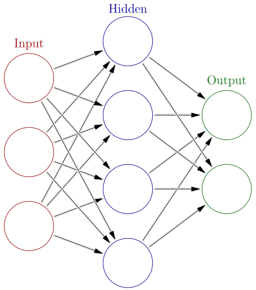
Probability & One vs All Lab

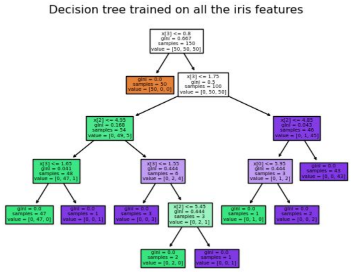
Naive Bayes Lab

As a class, let’s complete the challenge together.









Monday

Next week, we will continue our review of classification algorithms by diving deeper into Naive Bayes & something called K-Nearest-Neighbors.

You are who your neighbors are.

If you understand what you're doing, you're not learning anything. - Anonymous