Naïve Bayes Classifier

Probability Review

Joint Probability

- Joint probability: $P(AB)$ means the probability of both A *and* B occurring at the same time
  - We calculate this using $P(AB) = P(A|B)P(B)$.
- Question: How would you estimate each of the following given the observations of the binary features to the right?
  - $P(B)$
  - $P(A|B)$
  - $P(AB)$

A	B	C
1	1	0
1	1	0
1	0	1
0	0	0
0	0	1
0	0	0
0	1	1
1	1	0
0	1	0
0	1	1

Joint Probability

- Joint probability: $P(AB)$ means the probability of both A *and* B occurring at the same time
  - We calculate this using $P(AB) = P(A|B)P(B)$
- Question: How would you estimate each of the following given the observations of the binary features to the right?
- If we use the Maximum Likelihood approach:
  - $P(B) = \frac{\text{count}(B=1)}{\text{count}(B=1)+\text{count}(B=0)} = \frac{6}{10} = .6$
  - $P(A|B) = \frac{\text{count}(A=1,B=1)}{\text{count}(A=0,B=1)+\text{count}(A=1,B=1)} = \frac{3}{6} = .5$
  - $P(AB) = .5 * .6 = .3$
- Question: Which estimation, $P(B)$ or $P(A|B)$, do you feel more confident in? Why?
  - You might feel slightly less confident in $P(A|B)$ simply because we have fewer observations.

A	B	C
1	1	0
1	1	0
1	0	1
0	0	0
0	0	1
0	0	0
0	1	1
1	1	0
0	1	0
0	1	1

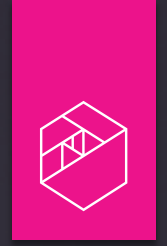
Expanding Joint Probabilities

- Joint probabilities are expanded by the chain rule.
  - 2 variables: $P(AB) = P(A|B)P(B)$
  - 3 variables: $P(ABC) = P(A|BC)P(B|C)P(C)$
  - 4 variables: $P(ABCD) = P(A|BCD)P(B|CD)P(C|D)P(D)$
  - Etc.

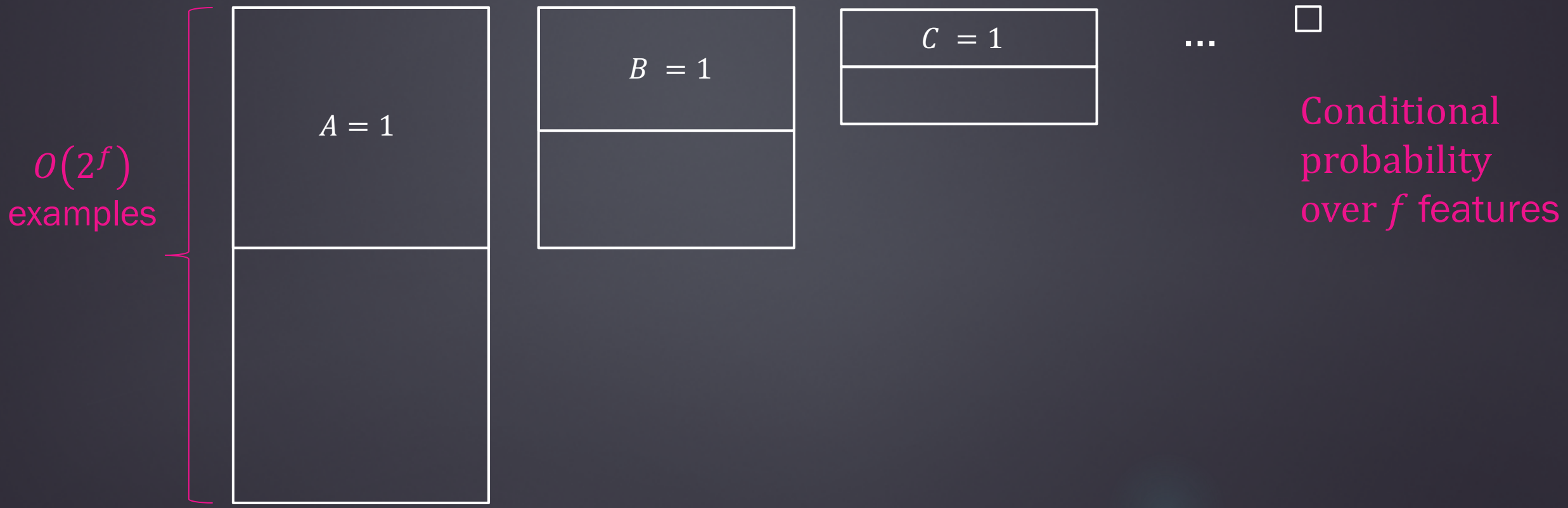
Scaling Conditional Probability

- If we have n total observations how does the number used change for different estimations?
  - Unconditional probability
    - $P(A)$: we can use all n observations.
  - Conditional probability
    - $P(A|B)$: we can use $nP(B)$ observations
    - $P(A|BC)$: we can use $nP(B)P(C)$ observations
    - Etc.

Conditional probabilities get out of hand quickly



- Estimating a conditional probability over f binary features requires $O(2^f)$ examples



Independence to the Rescue!

- Joint probabilities are expanded by the chain rule.
  - 2 variables: $P(AB) = P(A|B)P(B)$
  - 3 variables: $P(ABC) = P(A|BC)P(B|C)P(C)$
  - 4 variables: $P(ABCD) = P(A|BCD)P(B|CD)P(C|D)P(D)$
  - Etc.
- If we assume ABCD are independent, this becomes a little easier
  - 2 variables: $P(AB) = P(A)P(B)$
  - 3 variables: $P(ABC) = P(A)P(B)P(C)$
  - 4 variables: $P(ABCD) = P(A)P(B)P(C)P(D)$
  - Etc.
- If we assume our features are independent, we don't need an increase in the number of observations to estimate joint probabilities to a similar level of confidence.

Naïve Bayes

Classification Problem: Predict Party Membership

- Goal: Predict senators' party alignment based on how they voted on bills

Senator	Bill 1 (X_1)	Bill 2 (X_2)	Bill 3 (X_3)	Party (Y)
Senator A	1	1	0	D
Senator B	1	1	0	D
Senator C	1	0	1	D
Senator D	0	0	0	R
Senator E	0	0	1	R
Senator F	0	0	0	R
Senator G	0	1	1	R
Senator H	1	1	0	D
Senator I	0	1	0	D
Senator J	0	1	1	R

Classification Review – An Ideal Classifier?

- In many classification models, we are interested in finding the probability of some response Y given a set of features $\mathbf{X}$
  - In other words, we want to know the conditional distribution: $P(Y|\mathbf{X})$
  - Ex: Probability of party membership given vote history
- Imagine you knew $P(Y|\boldsymbol{X})$ exactly
  - How would you make predictions on a new data point x?
  - Predict the most probable class!

$$Y_{new} \leftarrow \underset{y_k}{\operatorname{argmax}}\, P(Y = y_k|\boldsymbol{X} = x)$$

- Unfortunately, we never know $P(Y|\boldsymbol{X})$ in real life, so this classifier is impossible ☹

How can Bayes' Theorem Help?

- Recall our friend, Bayes' Theorem:
  - $P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)}$
  - This is was derived from rules of probability
- We can try to learn the right-hand side from training data
- First, let's focus on $P(\boldsymbol{X}|Y)$
  - $\boldsymbol{X}$ is a data matrix with f features
  - From party membership example: $\boldsymbol{X}$ consists of one feature for each bill, i.e. X_1, X_2, and X_3
- The problems with scaling conditional probability apply here!

Apply that independence assumption

- If we assume $X's$ are conditionally independent given Y, things becomes easier
- $P(\boldsymbol{X}|Y) = P(X_1\ X_2\ X_3|Y)$

  $= \cancel{P(X_1|X_2\ X_3 Y)P(X_2|X_3 Y)P(X_3|Y)}$ Assume conditional independence of X's

  $= P(X_1|Y)P(X_2|Y)P(X_3|Y)$
- Ex: Assume votes on bills are independent of one another once you know party membership

Why is it “Naïve”?

- Naïve Bayes assumes our features are conditionally independent given Y
  - The term naïve refers to this assumption

$$P(X|Y) = P(< X_1, \dots, X_f > |Y) = \prod_{i=1}^{f} P(X_i|Y)$$

How does this help?

With conditional independence, we're able to reduce the amount of parameters we need to estimate from Bayes' Formula.

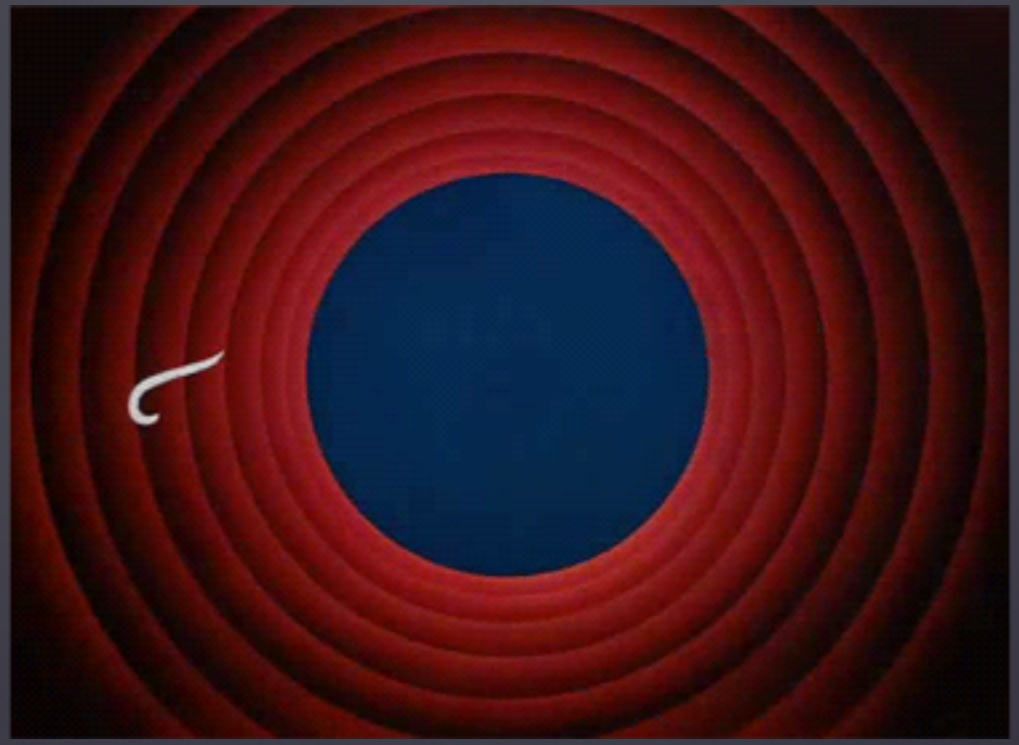
With conditional independence, number of parameters needed for $P(X|Y)P(Y)$ is $2f + 1$ (Linear!)

What does this mean for us? We reduce the amount of data needed to achieve a good estimate for our model!

Naïve Bayes in a Nutshell

The Naïve Bayes algorithm is quite simple:

- Take Bayes' Formula and add conditional independence → use it as a classifier



How do we train the model?

- Estimate $P(X_i|Y)$ for each feature X_i
  - $P(X_1|Y = D) = \frac{4}{5} = 0.8$
  - $P(X_1|Y = R) = \frac{0}{5} = 0$
  - $P(X_2|Y = D) = \frac{4}{5} = 0.8$
  - etc..
- Estimate $P(Y = y_k)$ for each target y_k
  - $P(Y = D) = \frac{5}{10} = 0.5$
  - $P(Y = R) = \frac{5}{10} = 0.5$

Senator	Bill 1 (X_1)	Bill 2 (X_2)	Bill 3 (X_3)	Party (Y)
Senator A	1	1	0	D
Senator B	1	1	0	D
Senator C	1	0	1	D
Senator D	0	0	0	R
Senator E	0	0	1	R
Senator F	0	0	0	R
Senator G	0	1	1	R
Senator H	1	1	0	D
Senator I	0	1	0	D
Senator J	0	1	1	R

How do we classify a new point?

- Given a new observation $X_{new} =< X_1, \ldots, X_f >$
  - Ex: Someone voted "Yes" for all three bills, i.e. $X_{new} =< 1, 1, 1 >$

- Calculate the product on the right for each response type and pick the label with the largest value

$$Y_{new} \leftarrow \underset{y_k}{\operatorname{argmax}} P(Y = y_k) \prod_{i}^{f} P(X_i^{new} | Y = y_k)$$

  - Democrat: $P(Y = D)P(X_1 = 1|Y = D)P(X_2 = 1|Y = D)\ P(X_3 = 1|Y = D)$
  - Republican: $P(Y = R)P(X_1 = 1|Y = R)P(X_2 = 1|Y = R)\ P(X_3 = 1|Y = R)$
  - Prediction = whichever of these calculations is larger

Let’s see it in action!

Game Time!

We will try to determine, by asking a few simple questions to our training set, whether someone in our test set is from California.

The questions:

1) Are you a Lakers/Clippers/Warriors fan?
2) Do you feel cold when it drops below 60°F (15.5°C)?
3) Have you ever experienced an earthquake?

Worst case scenario #1

- We're assuming our features are independent, but what if they're not?
  - Worst case: have two copies of the same feature? i.e.

$$X_i = X_j, i \neq j$$

- In this case, the feature gets weighted twice! (i.e. it appears as a square term in our product)
- Remember, this is the worst case scenario where we explicitly break the conditional independence. While this may change our final prediction, it does not break the algorithm.

Worst case scenario #2

- What if for some feature, we have zero observations fall into a given category?

$$P(X_i|Y) = 0$$

- We would introduce a zero into a product and we would never predict this class!

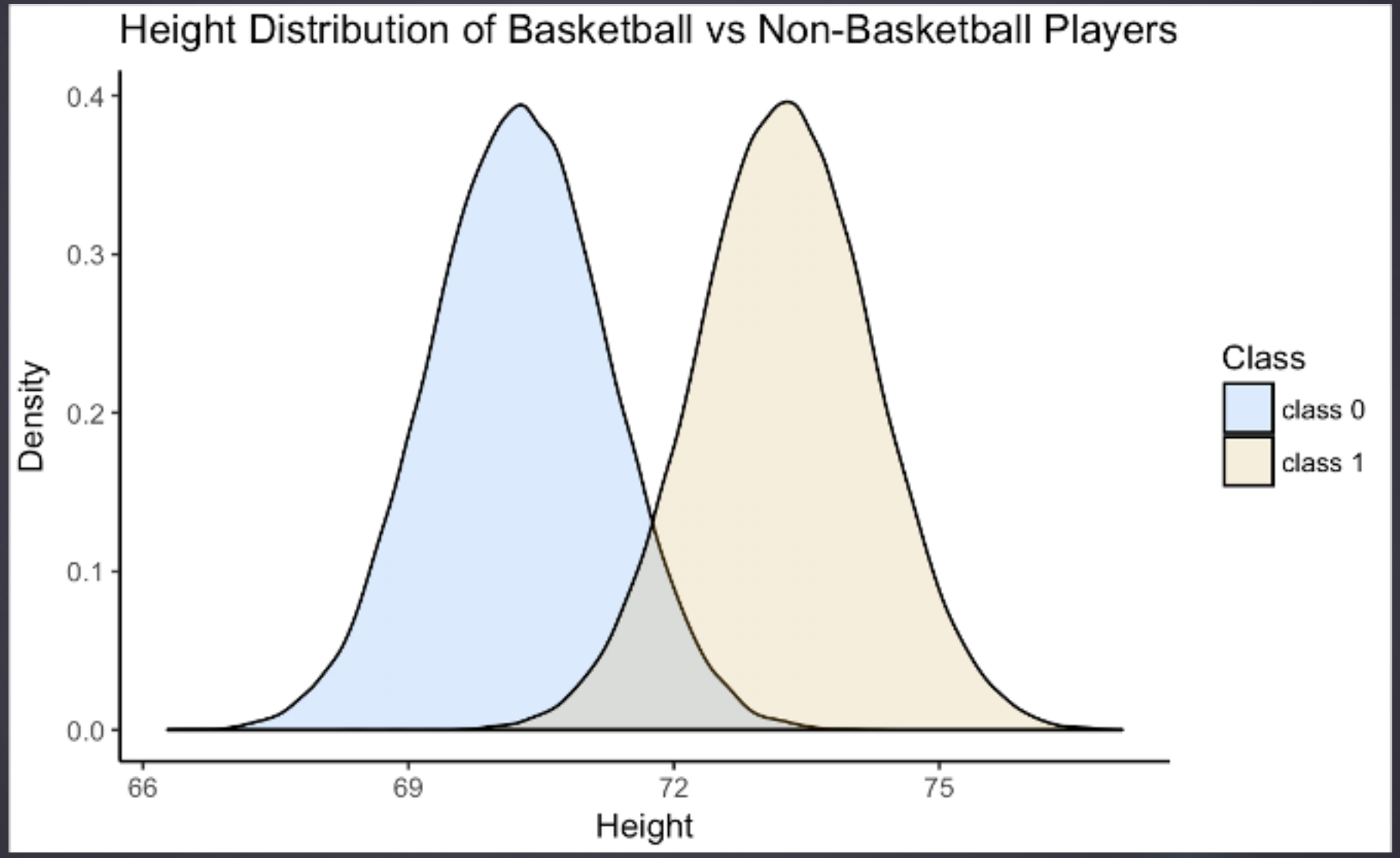
- If this happens, we could put a prior on our estimates of X_i.
  - (don't worry, sklearn handles this for us)

Gaussian NB

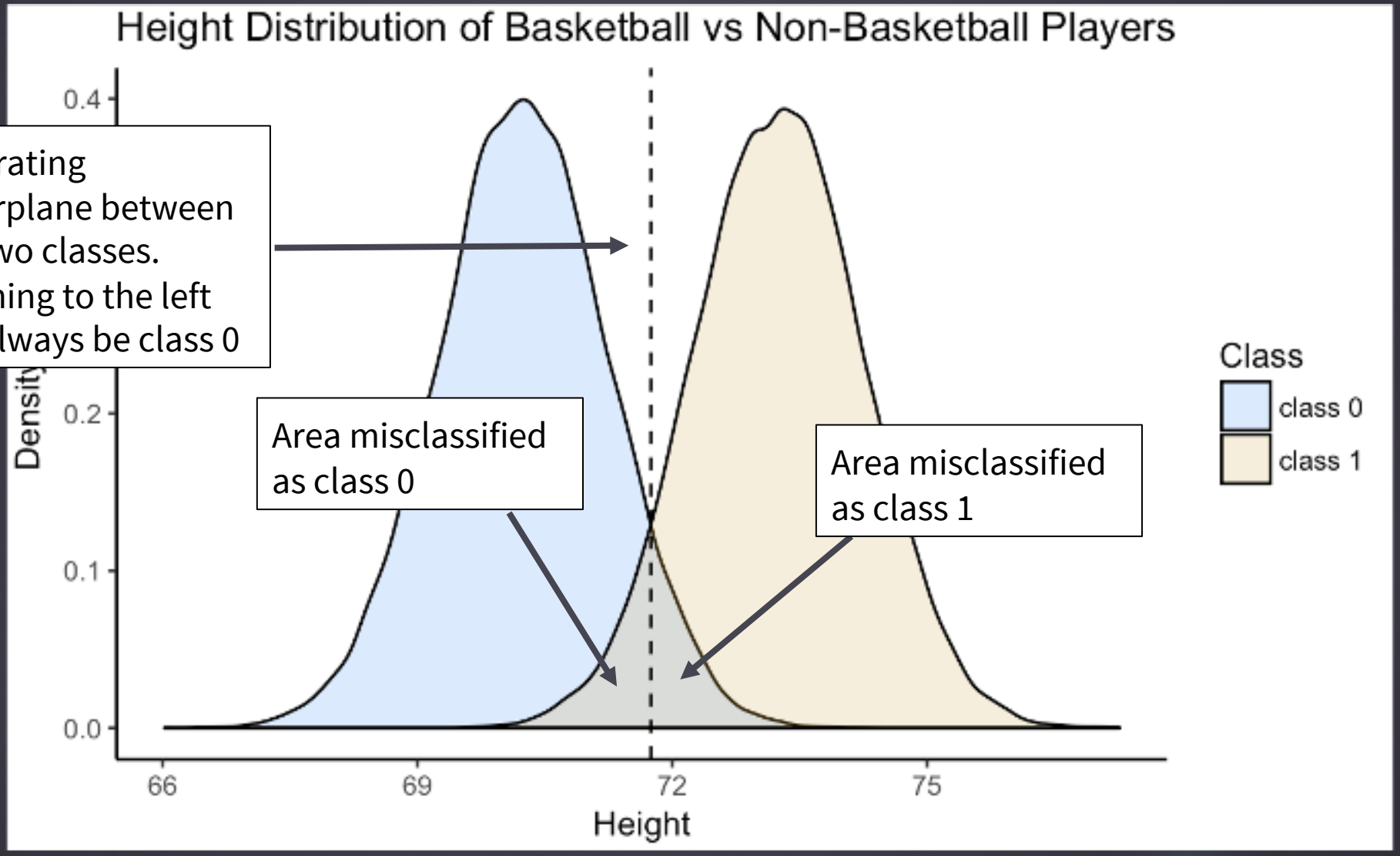
- So far, we've been doing Bernoulli Naïve Bayes.
- Not all real-world data is Boolean. Fortunately, NB can also handle continuous features.
- Gaussian Naïve Bayes assumes each X_i is normally distributed given Y.
- Note that our NB prediction does not change. We are still trying to find the class label that maximizes the product of conditional probabilities.

$$Y_{new} \leftarrow \underset{y_k}{\operatorname{argmax}} P(Y = y_k) \prod_{i}^{f} P(X_i^{new} | Y = y_k)$$

How does NB classify in this scenario?



Separating hyperplane



Naïve Bayes Summary

Naïve Bayes Classifier

- Works by applying Bayes' formula along with an assumption that our features are independent.
  - This often works well, even when we know our features are not truly independent.
- Application:
  - Problems with a very high number of features that all follow **Bernoulli (binary), Gaussian (Bell Curve), or Multinomial (n values) distribution.**
- **Complexity with n observations and m features**
  - **Train:** $O(n)$
  - Predict: $O(m)$