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 $P(AB) = P(A|B)P(B)$
    - This is just the definition.
- 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:
  - $P(B) = \frac{\text{count}(B=1)}{\text{count}(B=1)+\text{count}(B=0)} = \frac{6}{10} = .6$
  - $P(A|B) = \frac{\text{count}(A=1,B=1)}{\text{count}(A=1,B=1)+\text{count}(A=1,B=0)} = \frac{3}{6} = .5$
  - $P(AB) = .5 * .6 = .3$

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:
  - $P(B) = \frac{\text{count}(B=1)}{\text{count}(B=1)+\text{count}(B=0)} = \frac{6}{10} = .6$
  - $P(A|B) = \frac{\text{count}(A=1,B=1)}{\text{count}(A=1,B=1)+\text{count}(A=1,B=0)} = \frac{3}{6} = .5$
  - $P(AB) = .5 * .6 = .3$
- Question: Which estimation, $P(B)$ or $P(A|B)$, do you feel more confident in? Why?

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:
  - $P(B) = \frac{\text{count}(B=1)}{\text{count}(B=1)+\text{count}(B=0)} = \frac{6}{10} = .6$
  - $P(A|B) = \frac{\text{count}(A=1,B=1)}{\text{count}(A=1,B=1)+\text{count}(A=1,B=0)} = \frac{3}{6} = .5$
  - $P(AB) = .5 * .6 = .3$
- 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.

Scaling Conditional Probability

- Estimating a conditional probability over f 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.

Independence to the Rescue!

- 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

Can Bayes' Formula Help?

- In many machine learning models, we are interested in finding the probability of some response Y given a set of features X.
- In other words, we would like to know: $P(Y|X)$

Can Bayes' Formula Help?

- In many machine learning models, we are interested in finding the probability of some response Y given a set of features X.
- In other words, we would like to know: $P(Y|X)$
- Recall our friend, Bayes' Theorem: $P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)}$
  - We'll keep things simple at first by looking only at $P(X|Y)$.
    - Recall, this approach is Maximium Likelihood Estimation.

Can Bayes' Formula Help?

- In many machine learning models, we are interested in finding the probability of some response Y given a set of features X.
- In other words, we would like to know: $P(Y|X)$
- Recall our friend, Bayes' Theorem: $P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)}$
- The problems with scaling conditional probability apply here!
  - Imagine we are trying to predict whether someone in the USA is rich or poor by looking at 30 Boolean features. We would need $2^{30} \sim 1 \text{ billion}$ observations. There aren't even that many people in the USA!

Why is it “Naïve”?

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

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

How does this help?

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

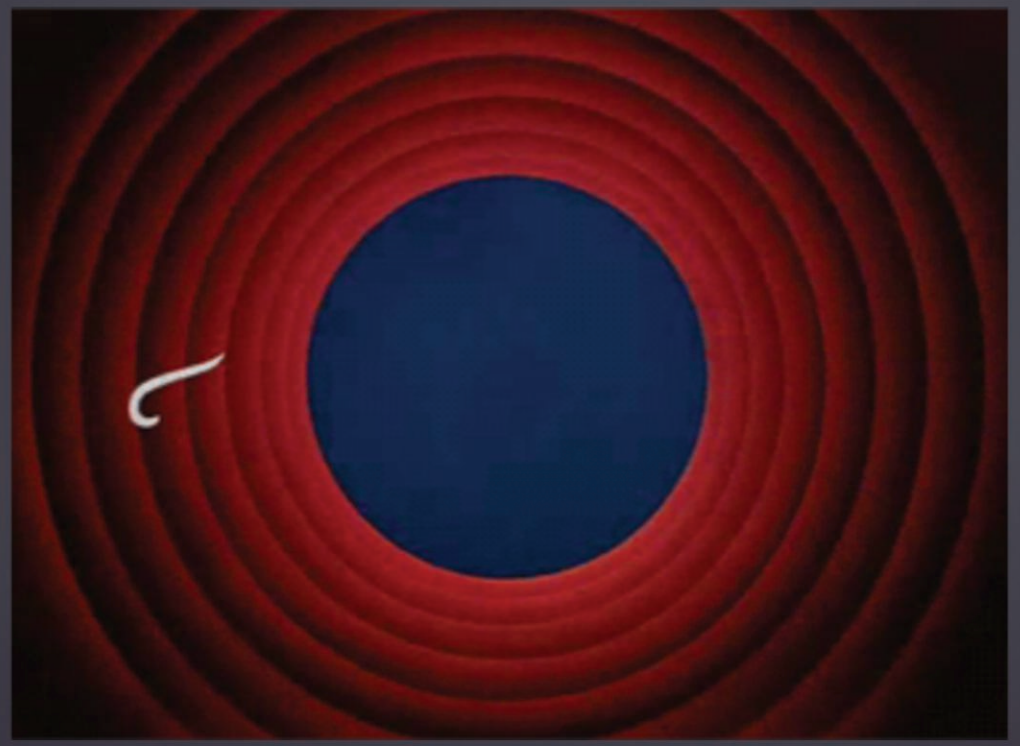
With conditional independence, number of parameters needed for $P(X|Y)P(Y)$ is $2n + 1$ (Linear!)

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



How do we classify a new point?

- Given a new observation

$$X_{new} =< X_1, \dots, X_n >$$

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

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

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 has an advanced degree.

The questions:

1) Were you born outside of the USA?
2) Do you work more than 45 hours per week (on average)?*
3) Have you ever been to Burning Man, Outside Lands or Hardly Strictly Bluegrass Music Festival?**

Do you have an advanced degree?

Some notation:

D = 1 if and only if you have an advanced degree

I = 1 if and only if you were born outside of the US

W = 1 if and only if you work more than 45 hours per week (on average)

F = 1 if and only if you have been to Burning Man, Outside Lands, or Hardly Strictly Bluegrass

Do you have an advanced degree?

Advanced Degree	No Advanced Degree
10/19 = 0.526	9/19 = 0.474

Determine the priors

Advanced Degree	No Advanced Degree
P(I=1|D=1) = 5/10 = 0.5	P(I=1|D=0) = 4/9 = 0.444
P(I=0|D=1) = 5/10 = 0.5	P(I=0|D=0) = 5/9 = 0.555
P(W=1|D=1) = 7/10 = 0.7	P(W=1|D=0) = 3/9 = 0.333
P(W=0|D=1) = 3/10 = 0.3	P(W=0|D=0) = 6/9 = 0.666
P(F=1|D=1) = 1/10 = 0.1	P(F=1|D=0) = 2/9 = 0.222
P(F=0|D=1) = 9/10 = 0.9	P(F=0|D=0) = 7/9 = 0.777

Predict the most likely class

Using the Naive Bayes formula, we can calculate whether our model is generalizing well in the test set.

Let's look at my vector: X = <1, 1, 0> , Y = 1

So to calculate which class I belong in, we take my X vector and multiply the priors for both having an advanced degree and no advanced degree and pick the label with the largest result.

Predict the most likely class

Advanced Degree	No Advanced Degree
P(y=1) * P(I=1|D=1) * P(W=1|D=1) * P(F=0|D=1) = 0.526 * 0.5 * 0.7 * 0.9 = 0.166	P(y=0) * P(I=1|D=0) * P(W=1|D=0) * P(F=0|D=0) = 0.474 * 0.444 * 0.333 * 0.777 = 0.054

Prediction → Advanced Degree!!

We did it!

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!

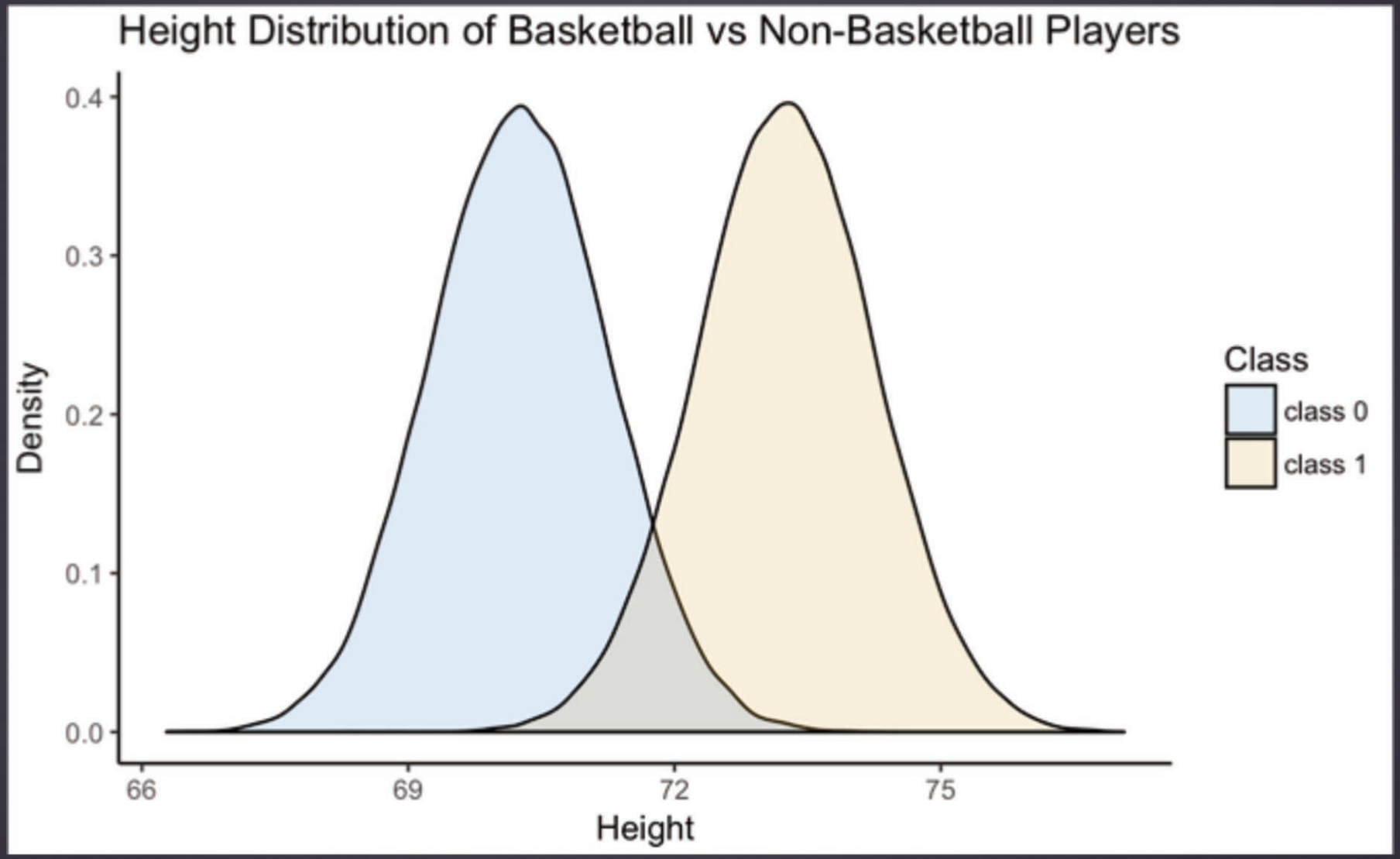
- If this happens, we could put a prior on our estimates of X_i.
  - (don't worry, sklearn handles this for us)

Gaussian NB

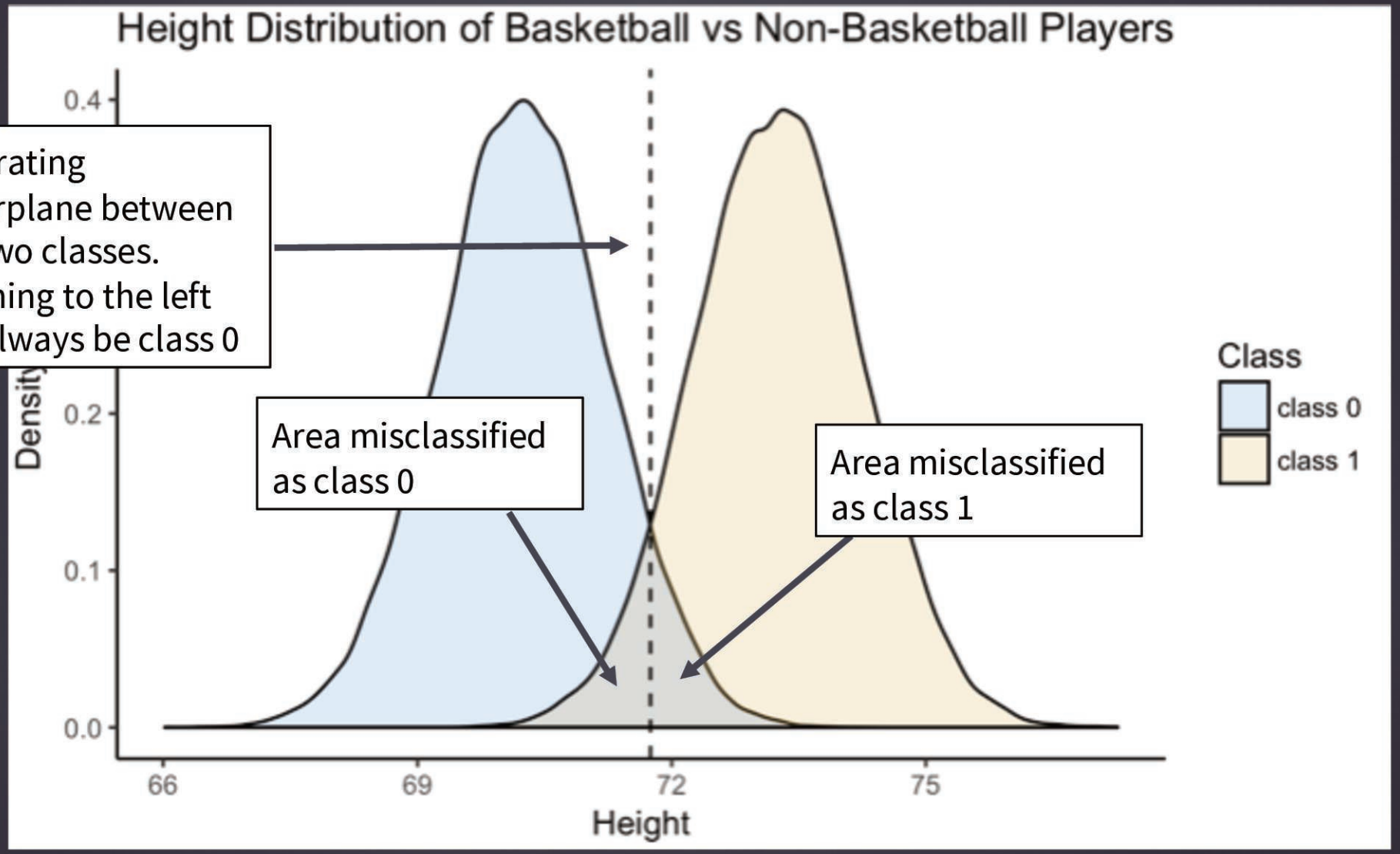
- Not all real-world data is Boolean. Fortunately, NB can also handle continuous data.
- Consider Boolean Y, X normally distributed. For now, also assume that P(Y=1) = ½.
- Note that our NB prediction does not change. We are still trying to find the class label that maximizes the product of conditional probabilities.

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

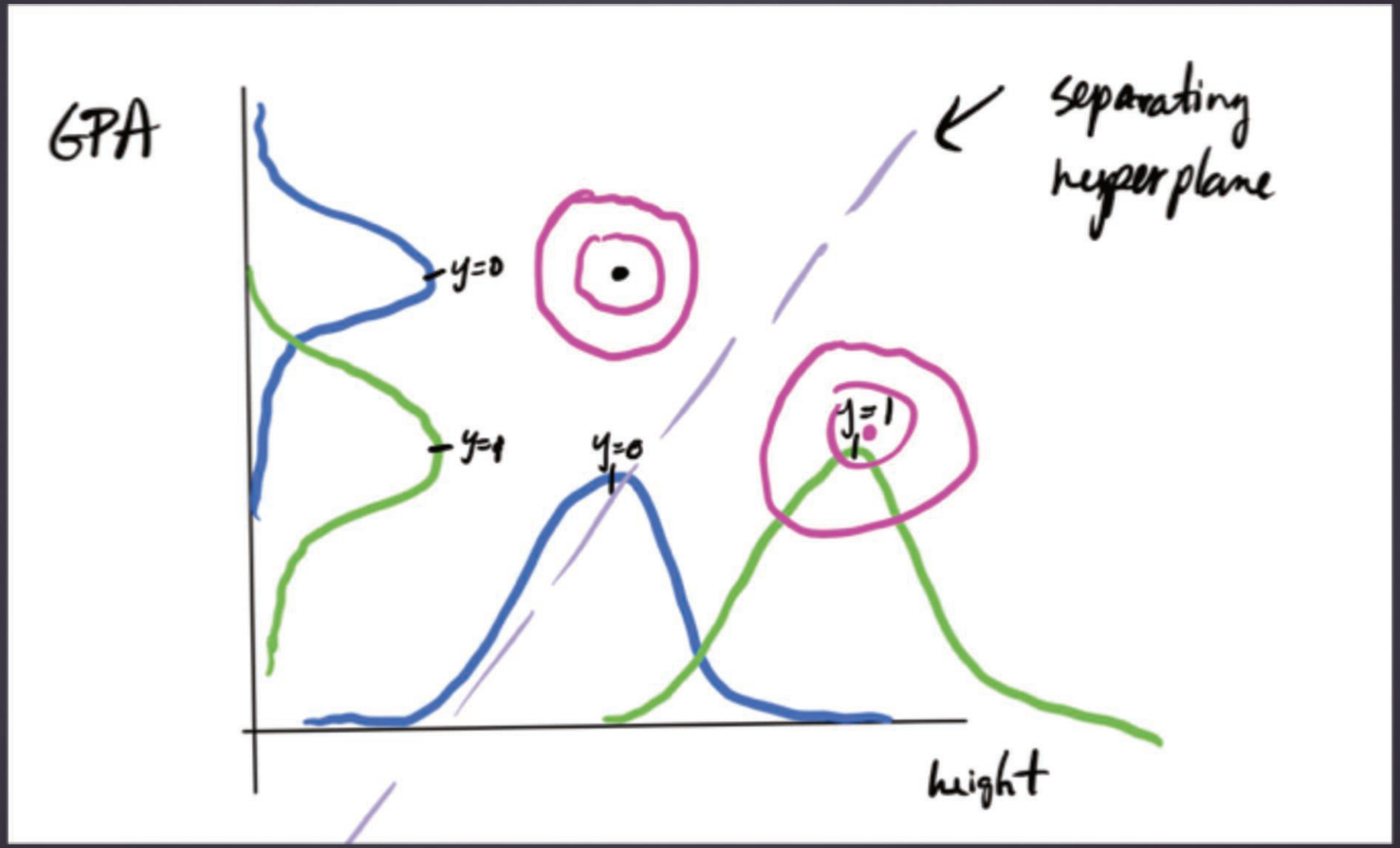
How does NB classify in this scenario?



Separating hyperplane



Separating hyperplane



Naïve Bayes Summary

Naïve Bayes

- 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 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: $O(m)$