

CLASSIFICATION – NAÏVE BAYES

BAYES CLASSIFIER

- In statistical classification, the **Bayes Classifier** minimizes the probability of misclassification
- It is possible to show that the test error rate is **minimized** (on average) by a classifier that *assigns each observation to the most likely class given its predictor values*

BAYES CLASSIFIER

- In statistical classification, the **Bayes Classifier** minimizes the probability of misclassification
- It is possible to show that the test error rate is **minimized** (on average) by a classifier that *assigns each observation to the most likely class given its predictor values*
 - Assign Y to the class j for which the conditional probability that $Y=j$ given the values of all the predictors is the highest
 - $$\hat{y}_0 = \underset{j \in \{1, \dots, J\}}{\operatorname{argmax}} P(Y = j | X = x_0)$$

Where x_0 is the vector of predictor values for a new instance

BAYES CLASSIFIER

- **For Example:**
- In a two-class problem (classes 0 and 1):
 - Assign:

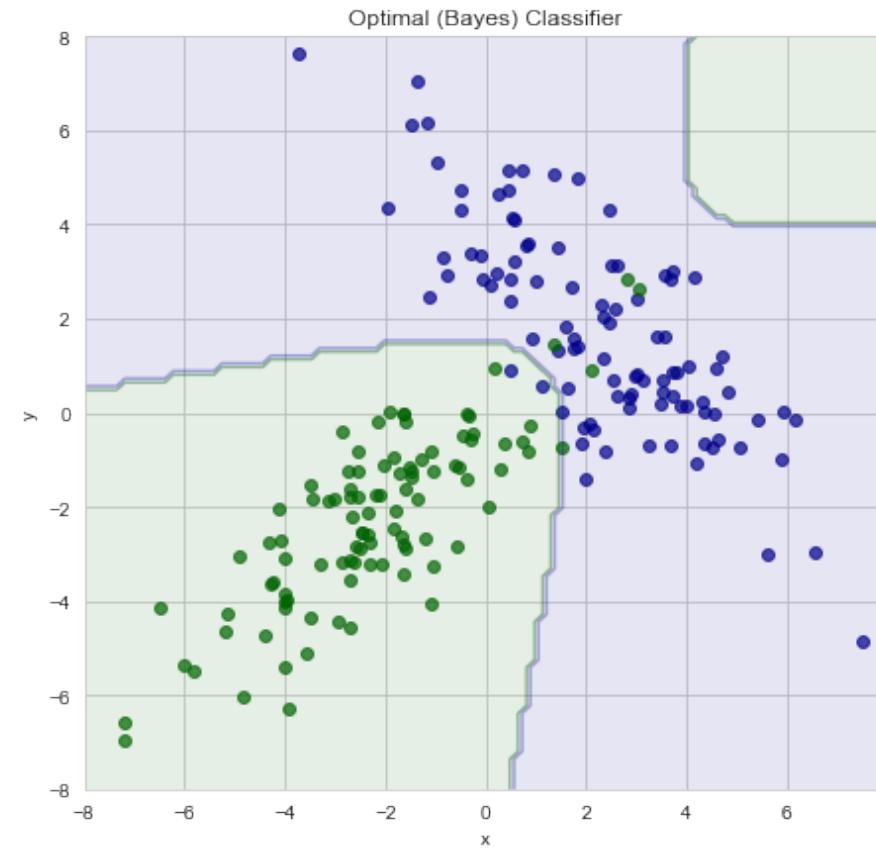
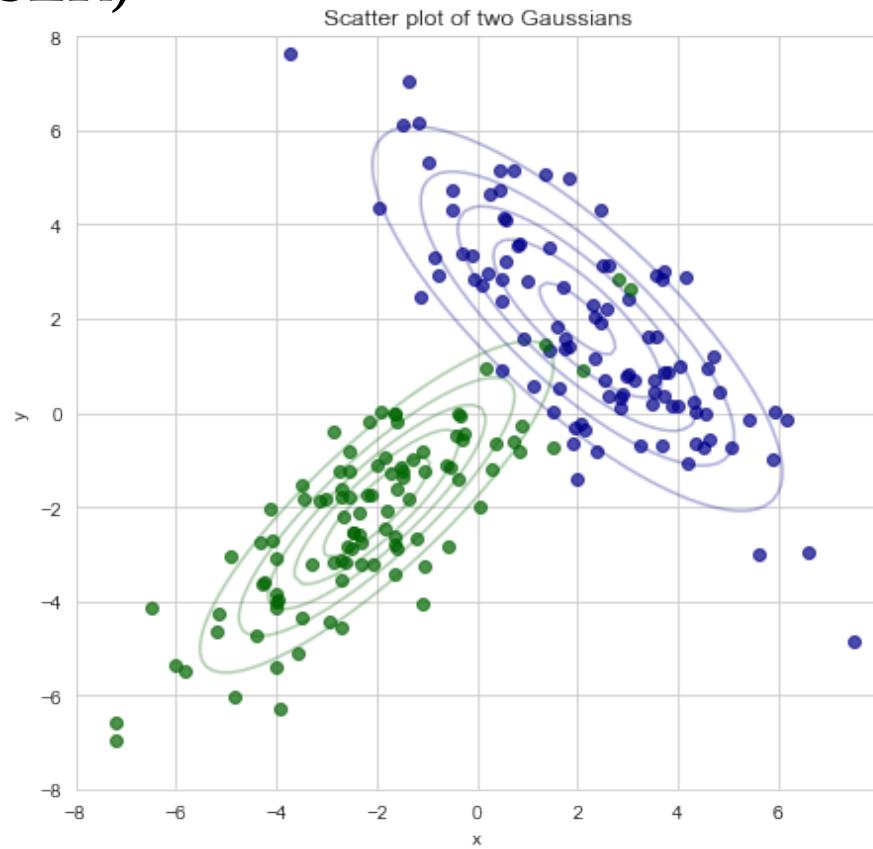
$$\hat{y}_0 = 1 \text{ if } P(y = 1 \mid X = x_0) > 0.5$$

$$\hat{y}_0 = 0 \text{ if } P(y = 1 \mid X = x_0) < 0.5$$

Where \mathcal{X}_0 is the vector of predictor values
for a new instance

ILLUSTRATION OF OPTIMAL CLASSIFIER

(From ISLR)



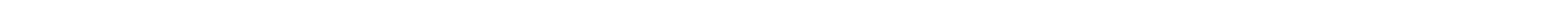
ONE SMALL PROBLEM...

- It is impossible to know the probability of Y given X , so computing the Bayes classifier is impossible for real problems
- Many approaches attempt to **estimate the conditional distribution of Y given X** and then classify a given observation to the class with the highest estimated probability

KNN AGAIN

- We've already seen how the KNN Classifier estimates this probability
- $P(Y = j|X = x_0) = \frac{1}{k} \sum_{x_i \in \mathcal{N}_0} \mathcal{I}(y_i = j)$

NAIVE BAYES



NAIVE BAYES CLASSIFIER

- ▶ Naive Bayes is a simple classifier that tries to estimate the conditional probability of $Y=j$ given the values of the Xs :

$$P(y = j|X = x)$$

- ▶ It is called "Naive" because it makes several “naive” simplifying assumptions in order to compute this conditional probability

BAYES RULE

$$P(Y = j|X = x) =$$

BAYES RULE

$$P(Y = j|X = x) = \frac{P(X = x|Y = j)P(Y = j)}{P(X = x)}$$

Likelihood Prior
Normalizing constant

Proportional to

Likelihood Prior

Joint pdf of X given Y=j

The diagram illustrates the Bayes Rule formula with several annotations:

- The formula is $P(Y = j|X = x) = \frac{P(X = x|Y = j)P(Y = j)}{P(X = x)}$.
- The term $P(X = x|Y = j)$ is labeled "Likelihood".
- The term $P(Y = j)$ is labeled "Prior".
- The term $P(X = x)$ is labeled "Normalizing constant".
- A yellow arrow points from the denominator $P(X = x)$ to the numerator $P(X = x|Y = j)P(Y = j)$, with the text "Proportional to" above it.
- A blue arrow points from the numerator $P(Y = j)$ to the denominator $P(X = x|Y = j)P(Y = j)$, with the text "Joint pdf of X given Y=j" below it.

BAYES RULE

Likelihood

Prior

$$P(Y = j|X = x) \propto P(X = x|Y = j)P(Y = j)$$

NAIVE BAYES SIMPLIFYING ASSUMPTIONS

$$P(Y = j|X = x) \propto \underset{\text{Likelihood}}{P(X = x|Y = j)} \underset{\text{Prior}}{P(Y = j)}$$

Simplifying assumption #1:

- In order to compute the likelihood, we assume **independence** between predictors
 - That is, we assume that X_1, X_2, \dots, X_p are mutually independent
-

NAIVE BAYES SIMPLIFYING ASSUMPTIONS

$$P(Y = j|X = x) \propto P(X = x|Y = j) \underset{\text{Likelihood}}{P(X = x)} \underset{\text{Prior}}{P(Y = j)}$$

Simplifying assumption #2 (usually):

- Prior distribution is computed using the data
- That is, $P(Y = j) = \frac{\# \text{instances of class } j \text{ in data}}{\text{total instances in data}}$

NAIVE BAYES SIMPLIFYING ASSUMPTIONS

$$P(Y = j|X = x) \propto \underbrace{P(X = x|Y = j)}_{\text{Likelihood}} P(Y = j) \underbrace{\qquad}_{\text{Prior}}$$

Simplifying assumption #3 (actually, this is always required in modeling data):

- We know the distribution of the X s, that is, the distribution of $P(X = x|Y = j)$
- And it is the same for every feature
- Popular choices in Scikit Learn
 - Normal distribution (`GaussianNB`)
 - Multinomial distribution (`MultinomialNB`)
 - Bernoulli distribution (`BernoulliNB`)
 - *IMPORTANT - Remember that these distributions refer to the pdf/pmf of the FEATURES*

NAIVE BAYES SIMPLIFYING ASSUMPTIONS

$$P(Y = j|X = x) \propto P(Y = j) \prod_{i=1}^p P(X_i = x_i|Y = j)$$

Estimated with frequencies from the training data

Assume to be Normal when Xs are numerical

Assumed to be Multinomial or Bernoulli otherwise

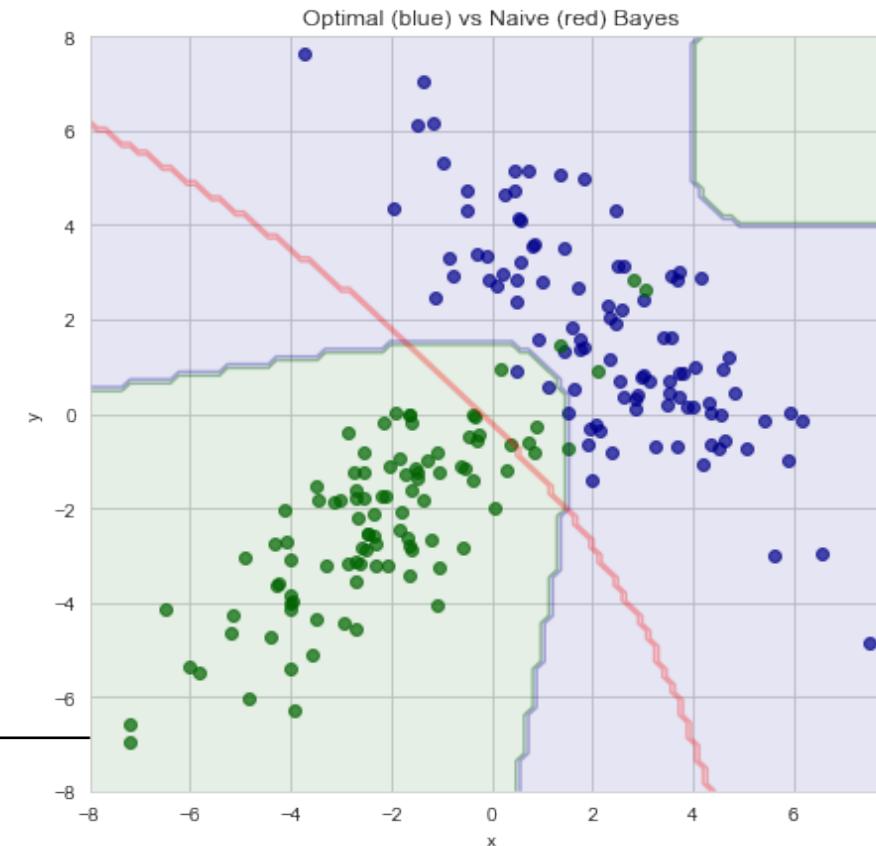
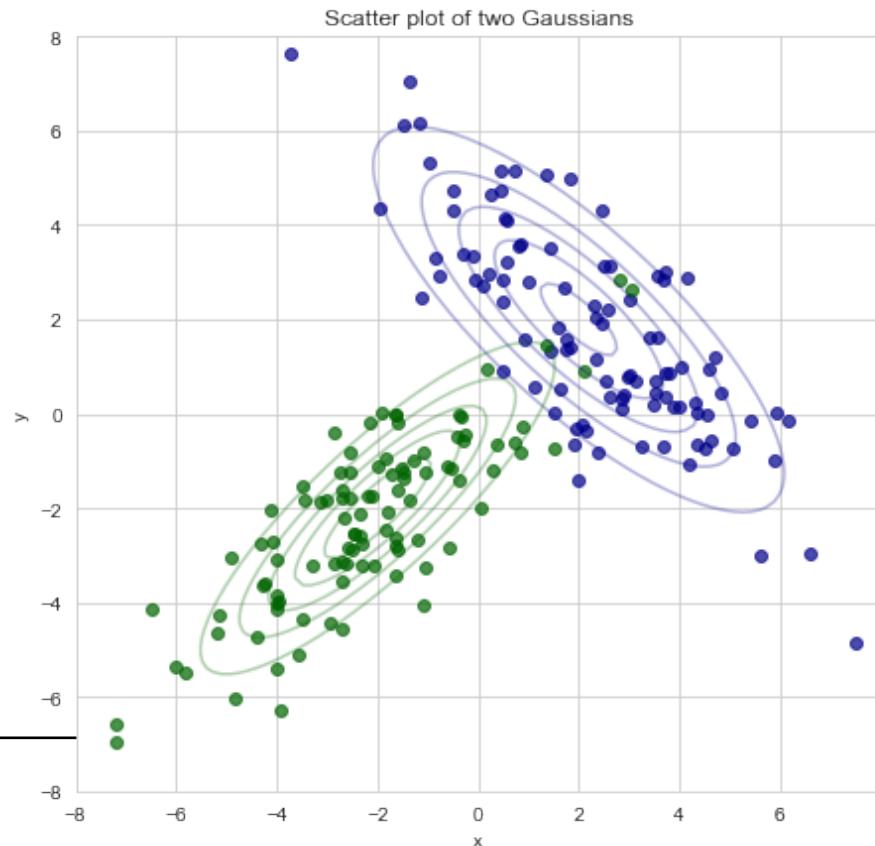
Main naive assumptions: Xs are all independent, so the joint distribution is the product of the marginals

NAIVE BAYES CLASSIFIER

$$\hat{y}_0 = \arg \max_{j \in \{1, \dots, J\}} P(Y = j) \left(\prod_{i=1}^p P(X = x_i \mid Y = j) \right)$$

BAYES VS NAIVE BAYES

(From ISLR)



WHEN TO USE NB

- Because NB classifiers make such stringent assumptions, they are often not as good as more complicated models
- *NB classifiers are a good choice for an initial baseline classification*
- Some advantages of NB classifiers:
 - They are very fast for both training and prediction
 - They are easily interpretable
 - There are few (if any) tunable parameters

WHEN TO USE NB

- NB classifiers tend to work in the following situations:
 - The naive assumptions actually match the data (rare)
 - For well-separated categories, when model complexity is less important
 - For very high-dimensional data, when model complexity is less important

COMMON SCIKIT-LEARN NB COMMANDS

Module: `sklearn.naive_bayes`

Estimators: `GaussianNB`, `MultinomialNB`, `BernoulliNB`
(use the model that corresponds to the *features*)

Fitting: call `.fit(Xtrain,ytrain)` method off the estimator

Prediction: Two possible methods (true for most classification estimators)

- ▶ # predicted class - 1d array
`.predict(Xtest)`
 - ▶ # predicted $P(Y=j|X)$ - 2d array ($rows=n$,
 $columns=\#classes$)
`.predict_proba(Xtest)`
-