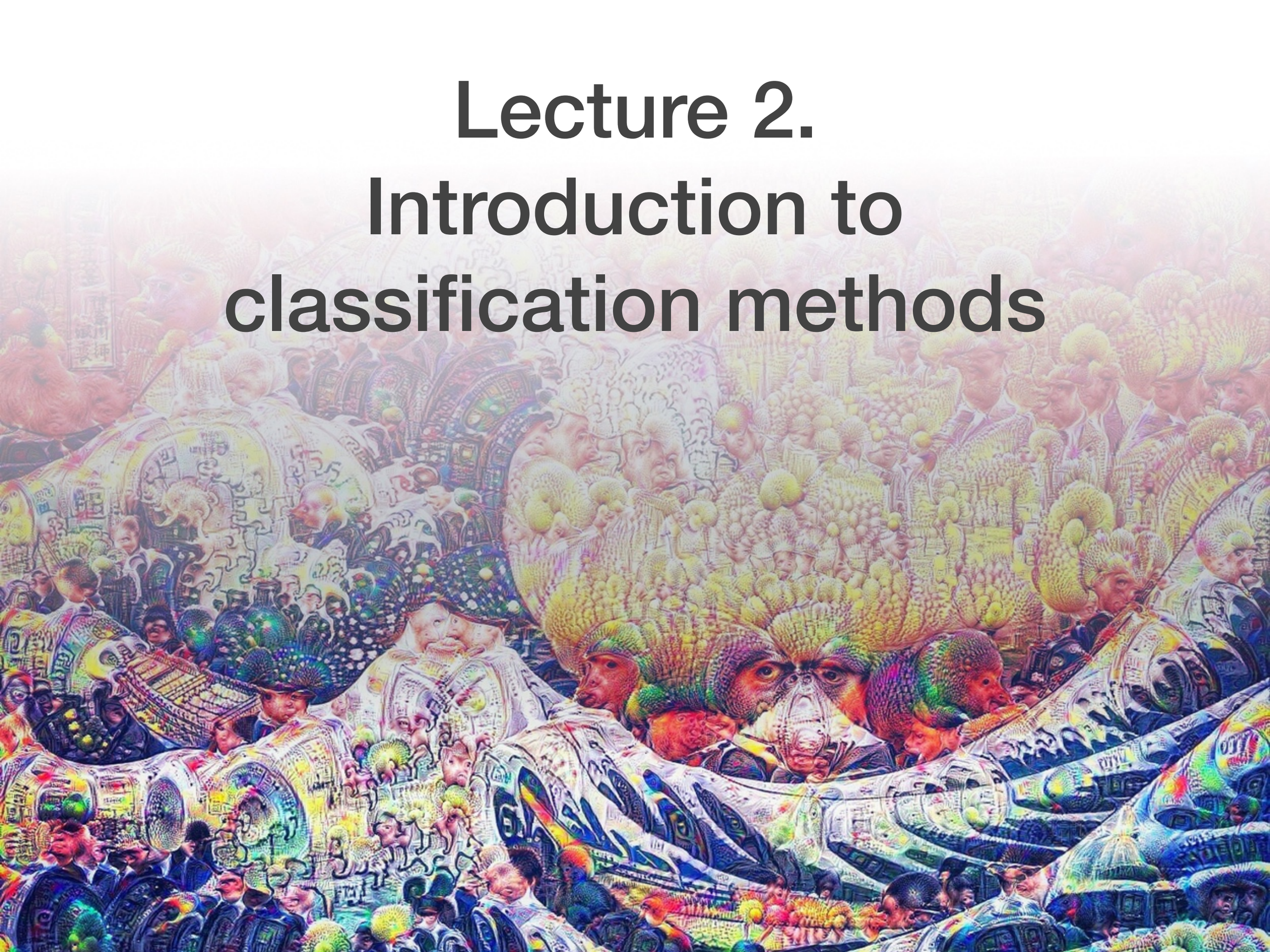
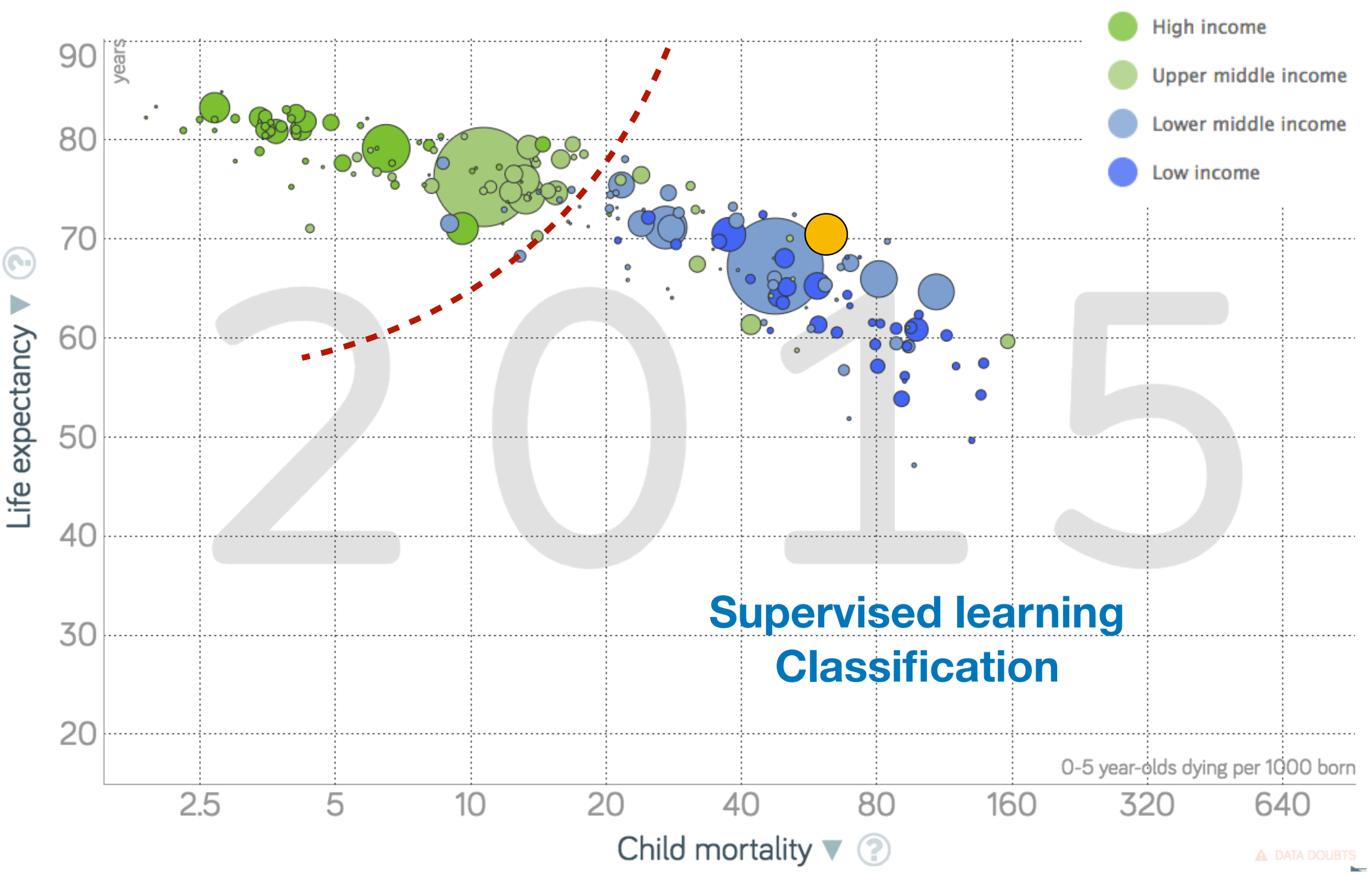
Lecture 2.
Introduction to classification methods



Taxonomy through examples



Supervised learning
Classification

Maximum likelihood method and logistic regression

Logistic regression

$y_i = 0$

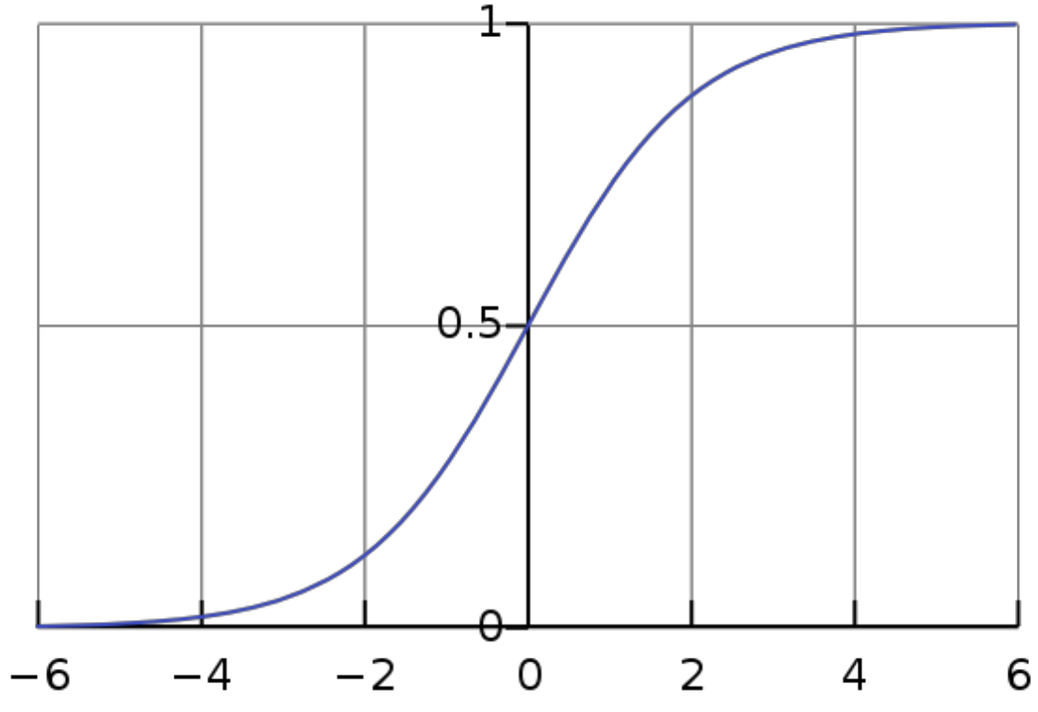
$y_j = 1$

Learn the probability

based on categorical and

continuous features

$$\hat{P}\left(y_i = 1 | x_{i,1}, x_{i,2}, \ldots\right)$$



$$\hat{P}_i \equiv \frac{1}{1 + \exp\left(-\beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2} - \ldots\right)}$$

Last lecture -logistic regression

(Multiple) Logistic regression - a classification problem

$$y_j = 1$$

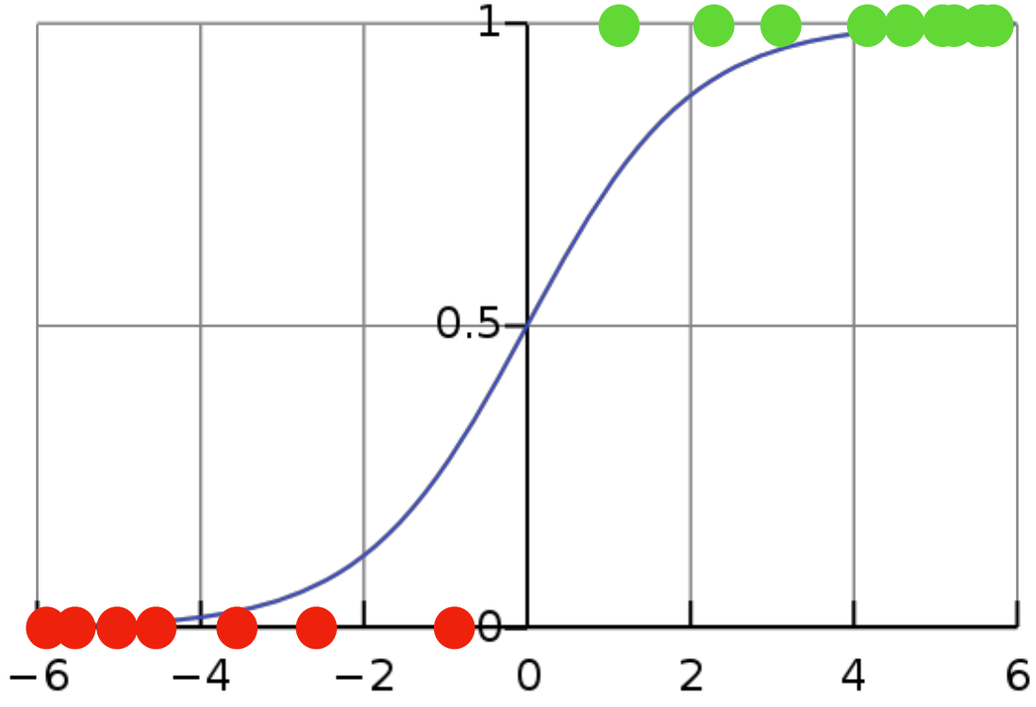
$$y_i = 0$$

Learn the probability

based on categorical and

continuous features

$$\hat{P}\left(y_i = 1 | x_{i,1}, x_{i,2}, \ldots\right)$$



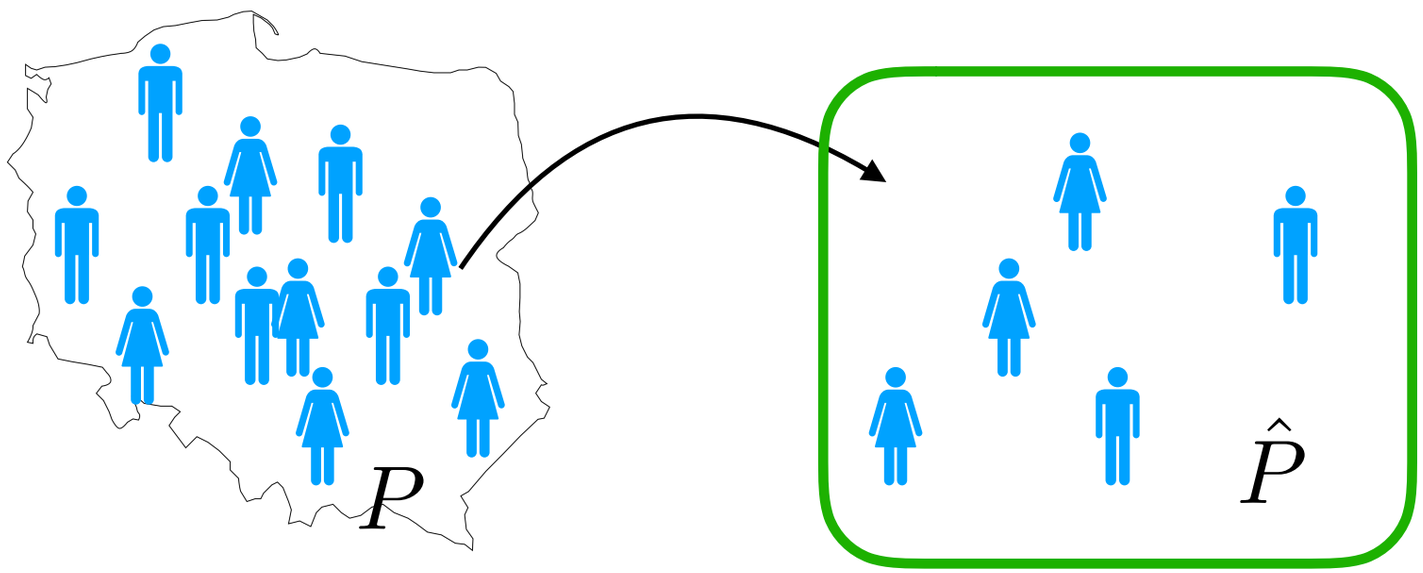
$$\log \frac{P(x)}{1 - P(x)} = \beta_0 + \beta_1 x_1 + \ldots$$

log-odds or logit

$$\hat{P}_i \equiv \frac{1}{1 - \exp\left(-\beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2} - \ldots\right)}$$

Logistic regression with maximum likelihood method

Maximum Likelihood method



Estimate the probability of randomly choosing a male from the population of PL

$y_i = 0$

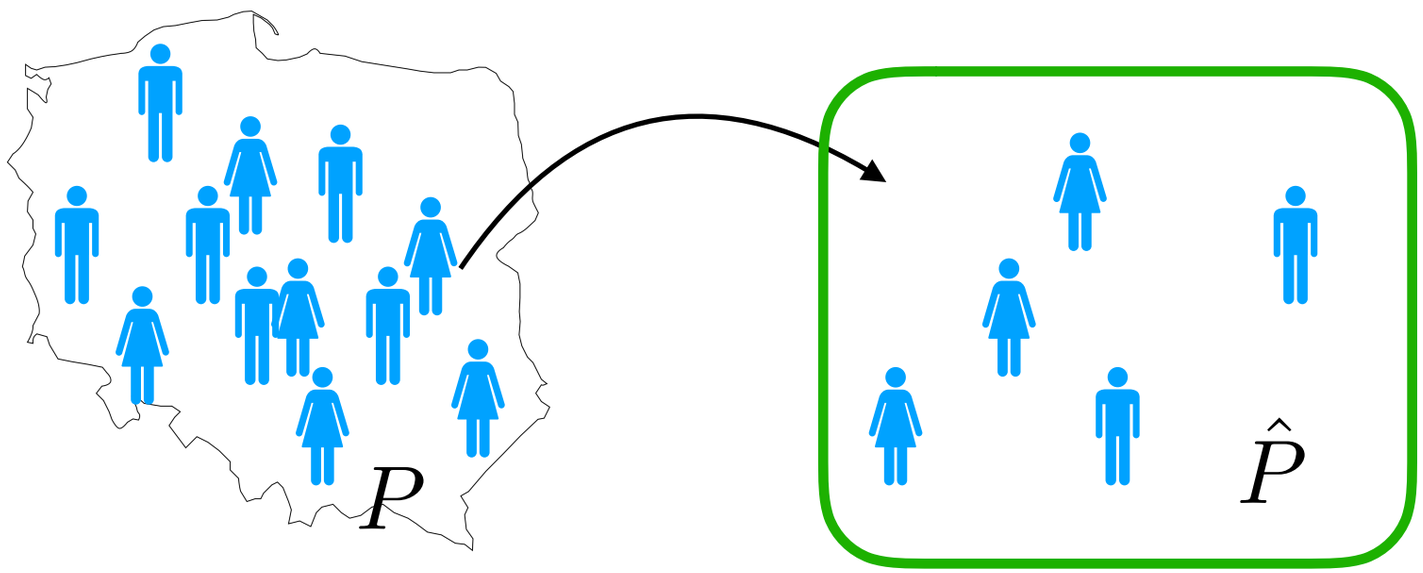
$y_j = 1$

$$L = \prod_i P^{y_i} (1 - P)^{1 - y_i}$$

$$\frac{\partial L}{\partial P}|_{P=\hat{P}} = 0$$

Logistic regression with maximum likelihood method

Maximum Likelihood method



Estimate the probability of randomly choosing a male from the population of PL, given some some features of the person

Sample

$y_i = 0$

$y_j = 1$

$$\hat{P}_i \equiv \frac{1}{1 + \exp\left(-\beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2} - \ldots\right)}$$

$$L = \prod_i P_i^{y_i} \left(1 - P_i\right)^{1 - y_i}$$

For each j:

$$\frac{\partial L}{\partial \beta_j}|_{\beta=\hat{\beta}} = 0$$

Last lecture -logistic regression

(Multiple) Logistic regression - a classification problem

$$y_j = 1$$

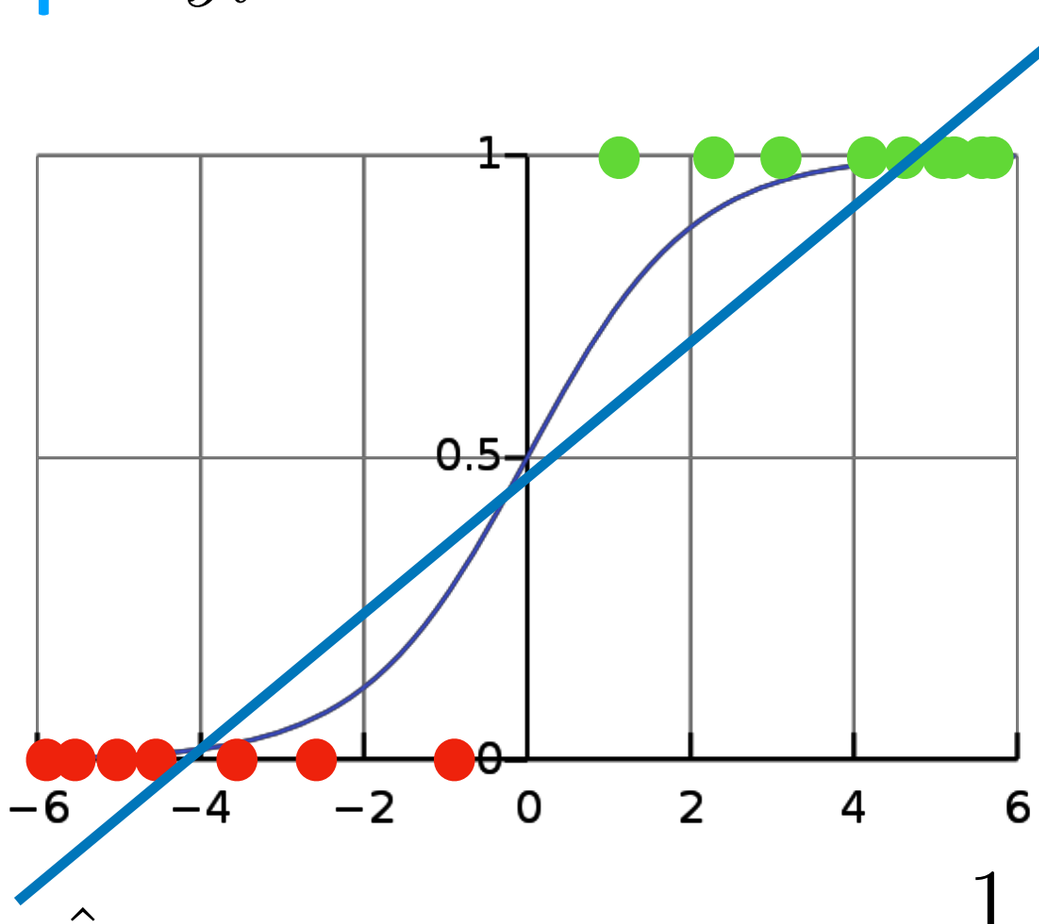
$$y_i = 0$$

Learn the probability

based on categorical and

continuous features

$$\hat{P}\left(y_i = 1 | x_{i,1}, x_{i,2}, \ldots\right)$$



$$\log \frac{P(x)}{1 - P(x)} = \beta_0 + \beta_1 x_1 + \ldots$$

log-odds or logit

$$y(x) = \beta_0 + \beta_1 x_1 + \ldots$$

$$\hat{P}_i \equiv \frac{1}{1 - \exp\left(-\beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2} - \ldots\right)}$$

Another reason not to use linear regression for classification

Predict medical condition of a patient in the emergency room

1 if stroke

2 if drug overdose

3 if epileptic seizure

- But this imposes arbitrary order
- and suggests that the differences are the same between some pairs

Another reason not to use linear regression for classification

Predict medical condition of a patient in the emergency room

1 if stroke

2 if drug overdose

3 if epileptic seizure

- But this imposes arbitrary order
- and suggests that the differences are the same between some pairs

1 if epileptic seizure

2 if stroke

3 if drug overdose

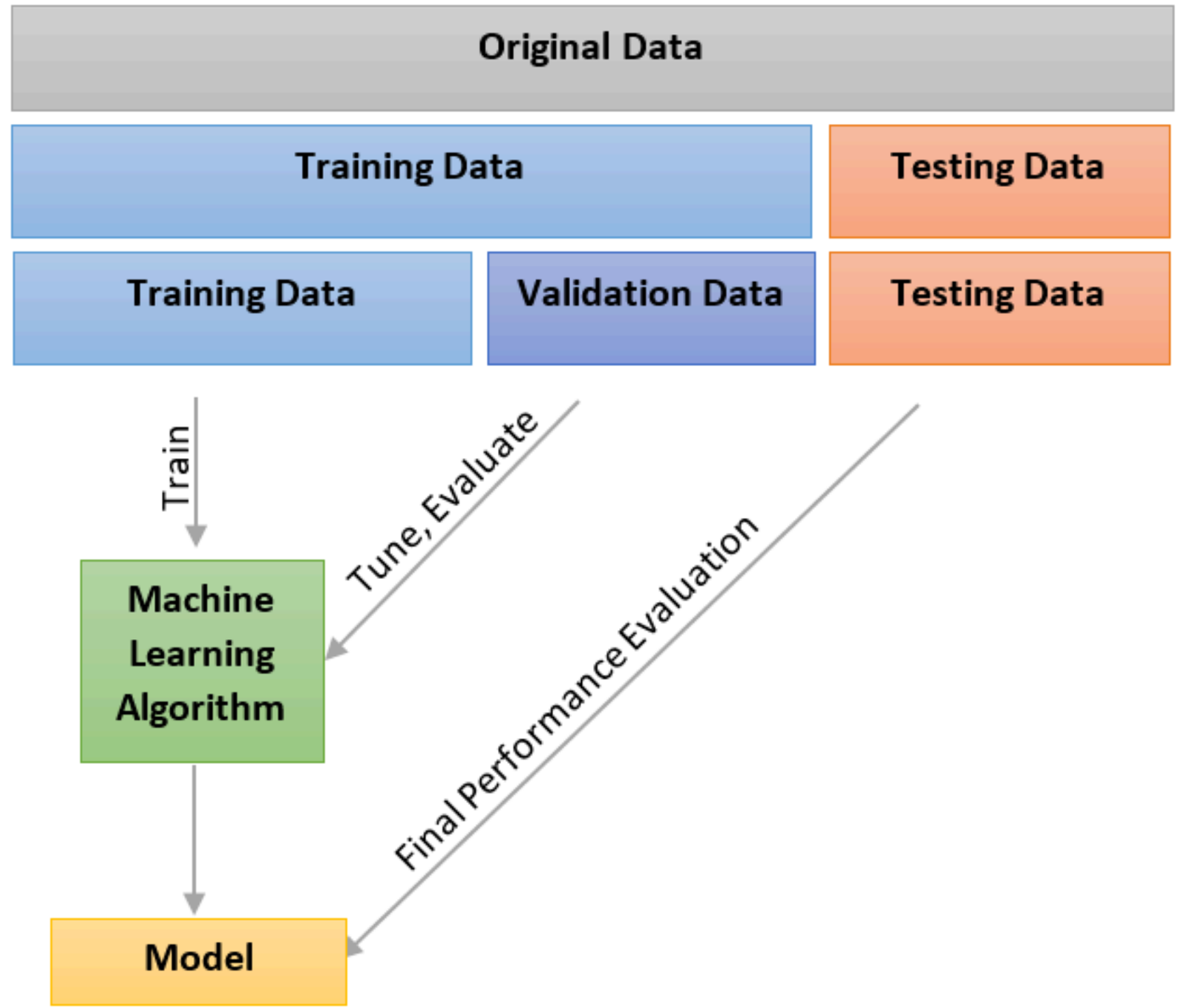
With linear regression, this would lead to three different models with different predictions

1 if epileptic seizure

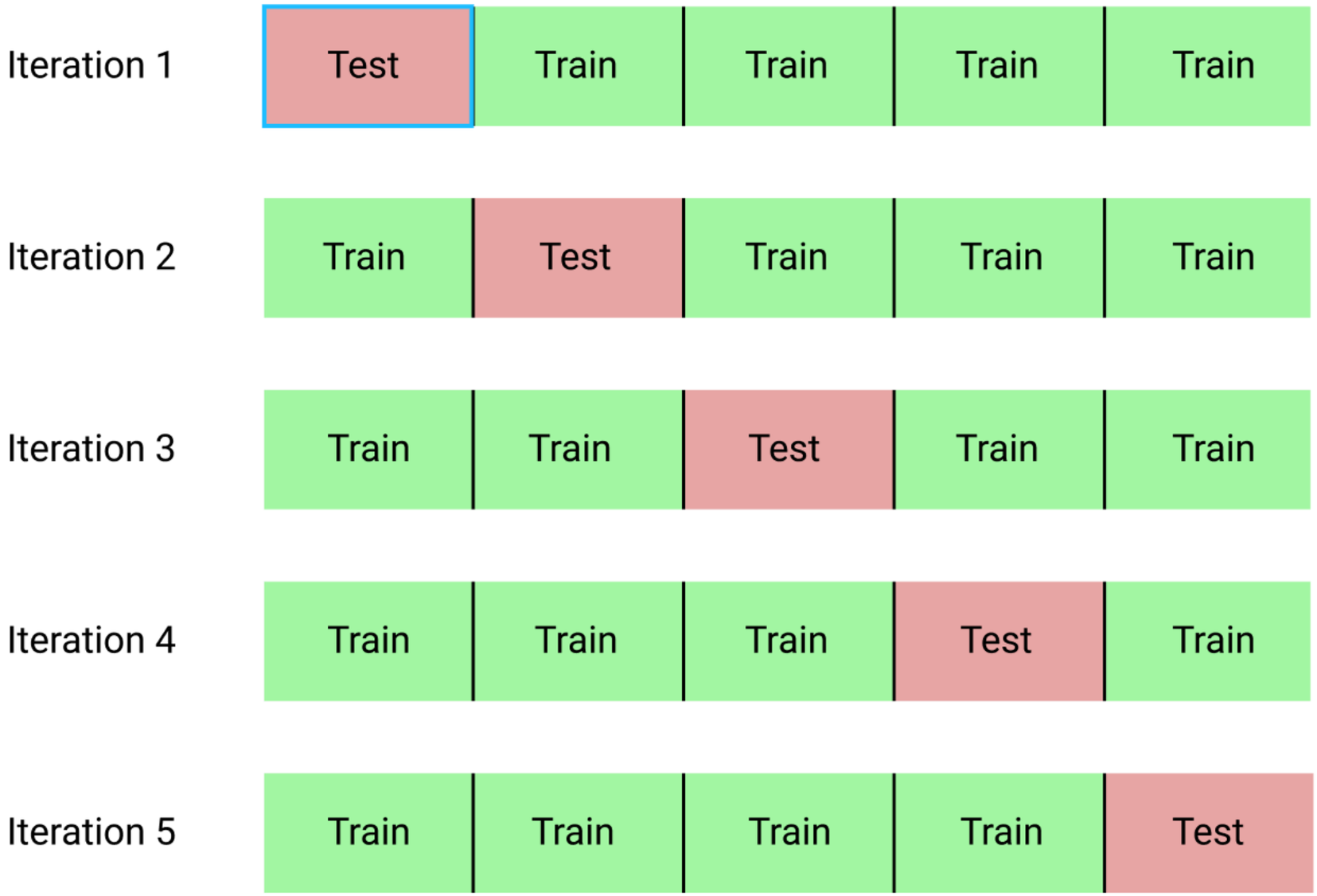
3 if stroke

9 if drug overdose

How to measure the quality of predictions (in binary Classification)



How to measure the quality of predictions (in binary Classification)



k-fold cross validation (for example when little data)

- watch out for bias and stationarity (this would look different for time series prediction)

Quality of predictions - Bayes rule

Seek estimate of f (call it $\hat{f}$) on the basis of observations (from the training ds): $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

- **Training error rate** $= \frac{1}{n} \sum_{i=1}^{n} I(y_i, \hat{y}_i)$

$$I(y_i \neq \hat{y}_i) = 1$$

$$I(y_i = \hat{y}_i) = 0$$

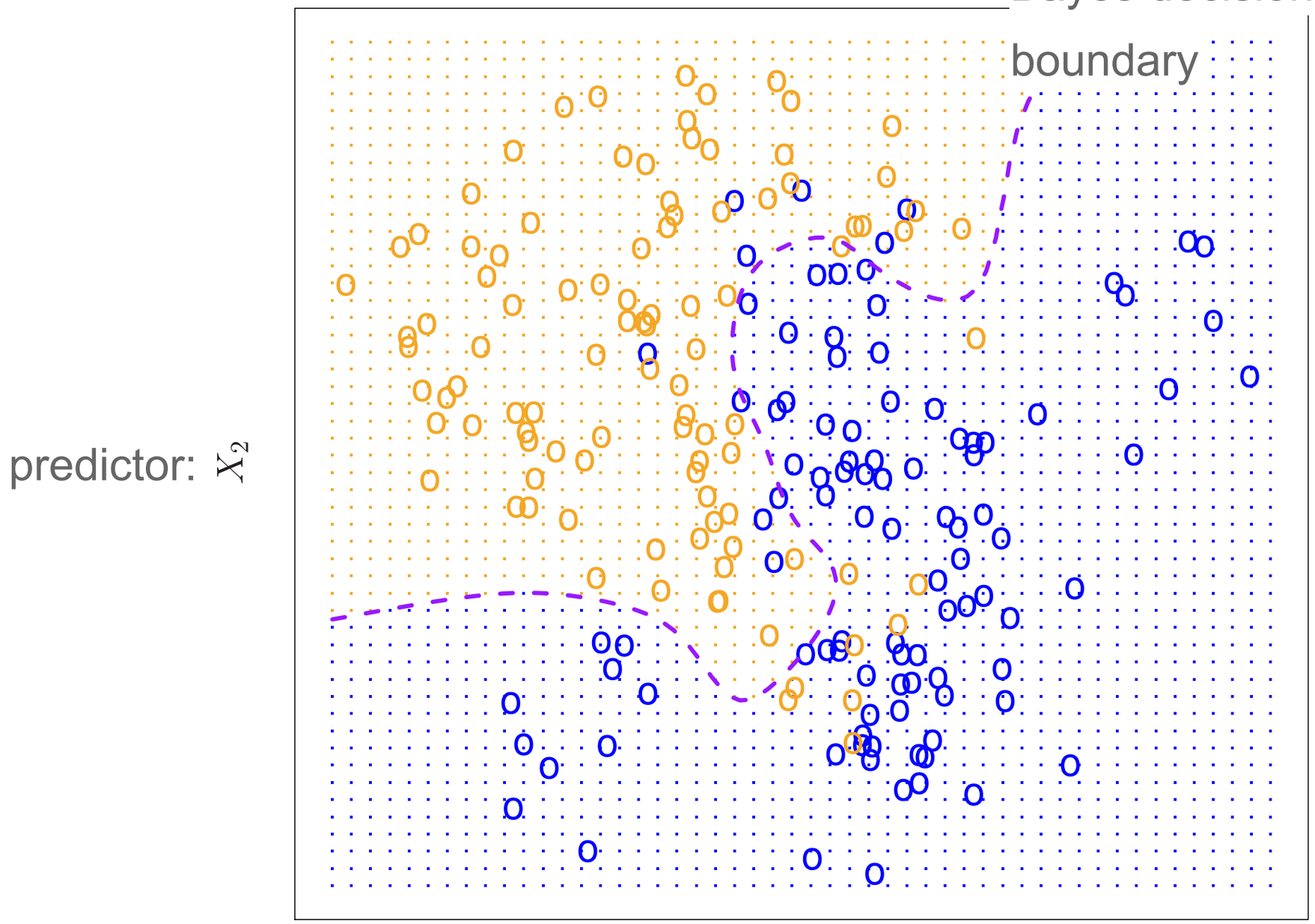
$\{(x_{n+1}, y_{n+1}), \ldots, (x_{n+m}, y_{n+m})\}$ (test ds)

- **Test error rate** $= \frac{1}{m-n} \sum_{i=n}^{m} I(y_i, \hat{y}_i)$

Theorem: Test error is minimized, on average, by the so-called Bayes classifier, that assigns the observations to the most likely class, given the predictor (x) values. In other words: to minimize test error, assign class to maximize the conditional probability: $P(Y = j | X = x)$

Quality of predictions - Bayes rule

In binary classification (simulated data)



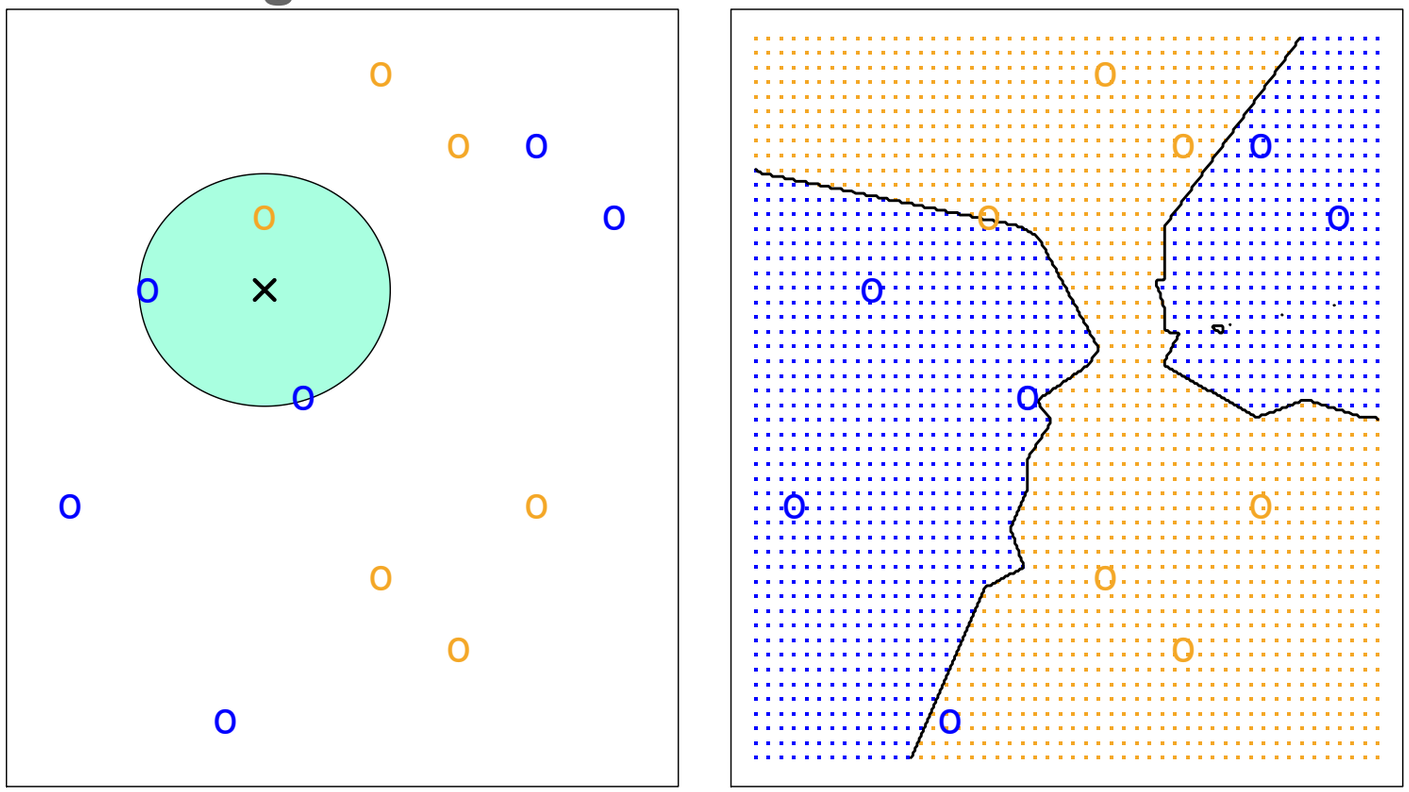
$P(Y = 1|X = x) > 0.5$ $P(Y = 0|X = x) > 0.5$

In reality, we never know the conditional probability distribution of Y given X!

Quality of predictions - K-Nearest Neighbor example

Thus some methods attempt to estimate the conditional probability distribution of Y given X.

Example: K-Nearest Neighbor classifier



Given an integer K, and test observation x_0, the KNN classifier identifies the points in the training data that are closest to x_0 (represented by N_0), and estimates the conditional probability for class j as the fraction of points in N_0 whose response values equal j:

$$P(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

Then applies the Bayes rule (our theorem) to classify x_0 to the class with highest probability

Quality of predictions - K-Nearest Neighbor example

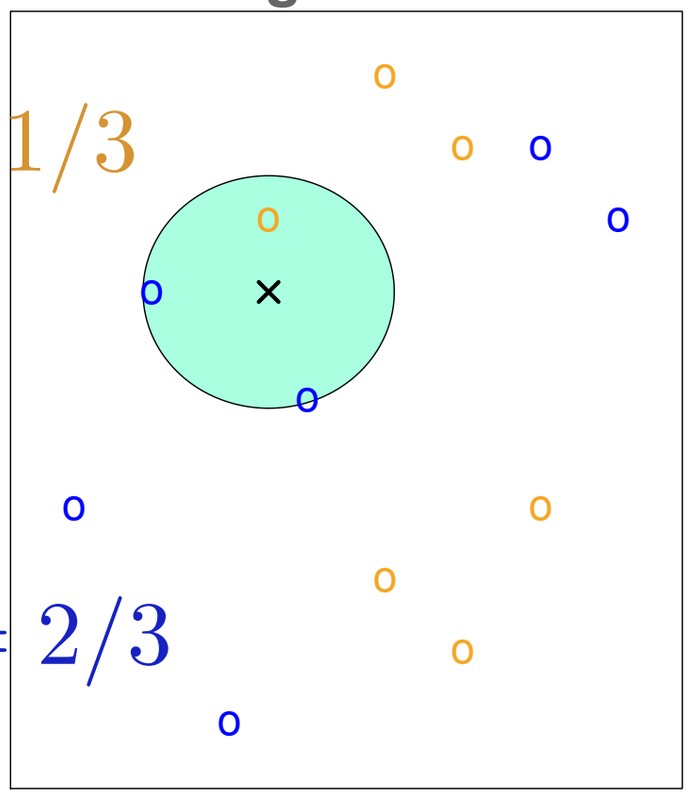
Thus some methods attempt to estimate the conditional probability distribution of Y given X.

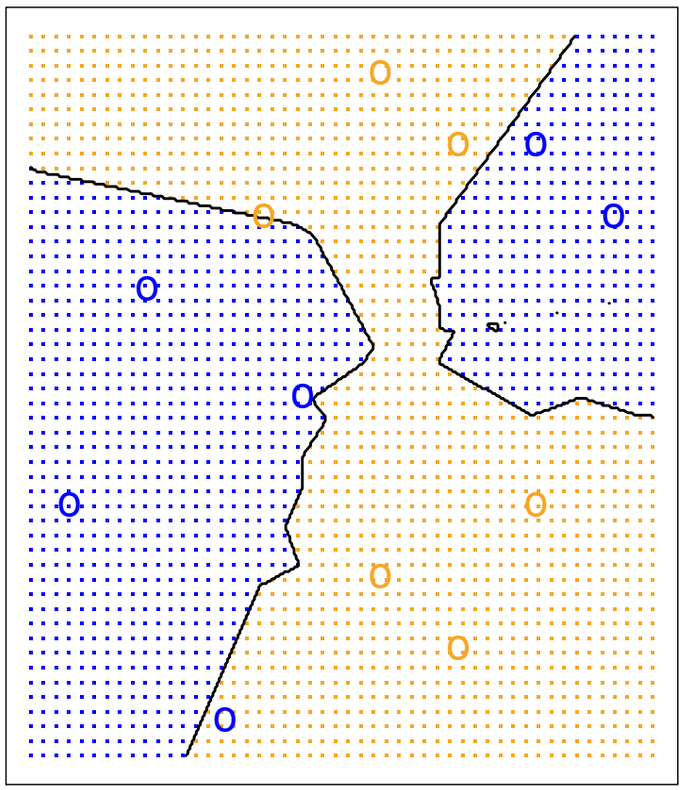
Example: K-Nearest Neighbor classifier

$P = 1/3$

$K = 3$

$P = 2/3$



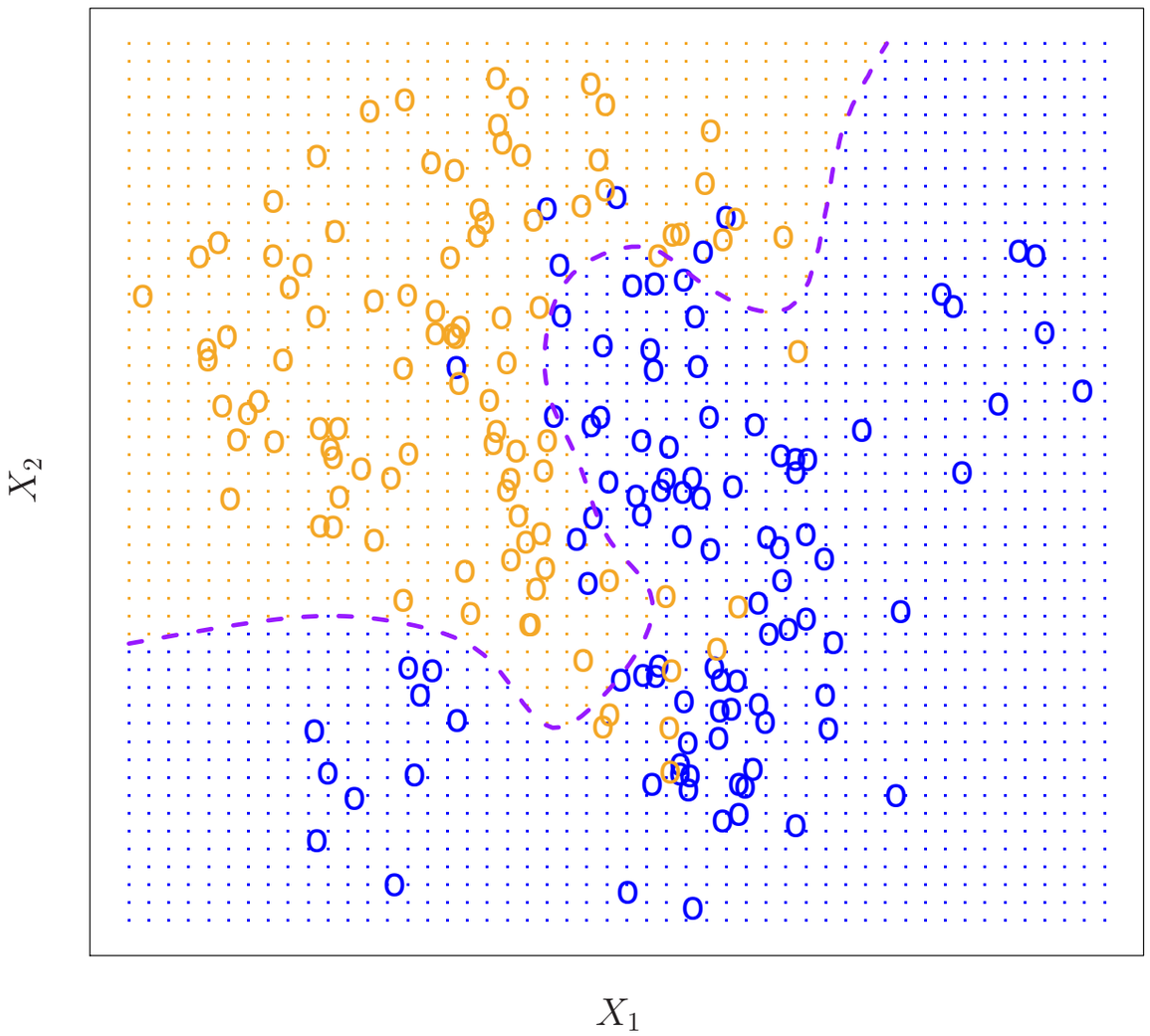


Given an integer K, and test observation x_0, the KNN classifier identifies the points in the training data that are closest to x_0 (represented by N_0), and estimates the conditional probability for class j as the fraction of points in N_0 whose response values equal j:

$$P(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

Then applies the Bayes rule (our theorem) to classify x_0 to the class with highest probability

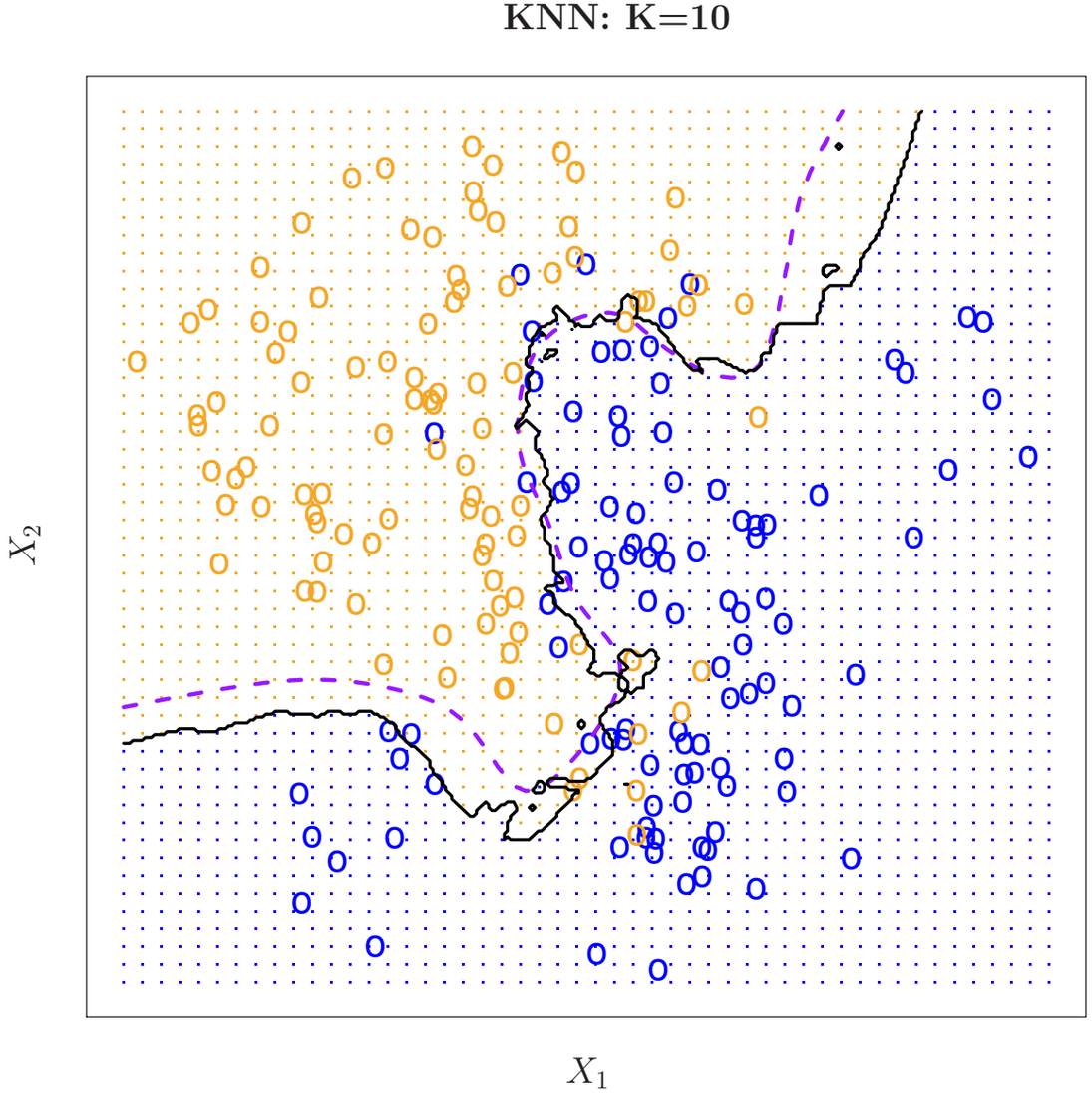
Quality of predictions - K-Nearest Neighbor example



Bayes error rate: 0.1304

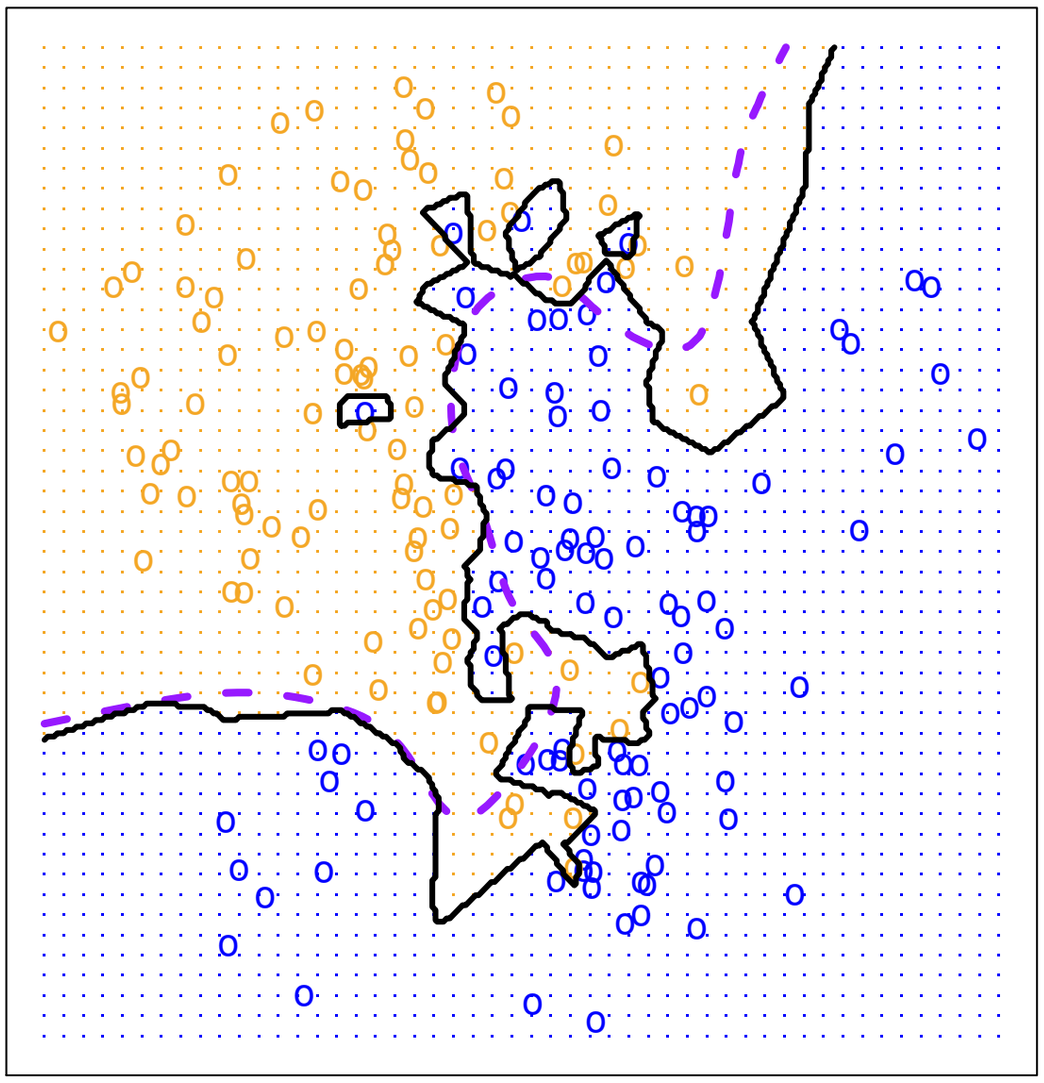
(>0 due to class overlap

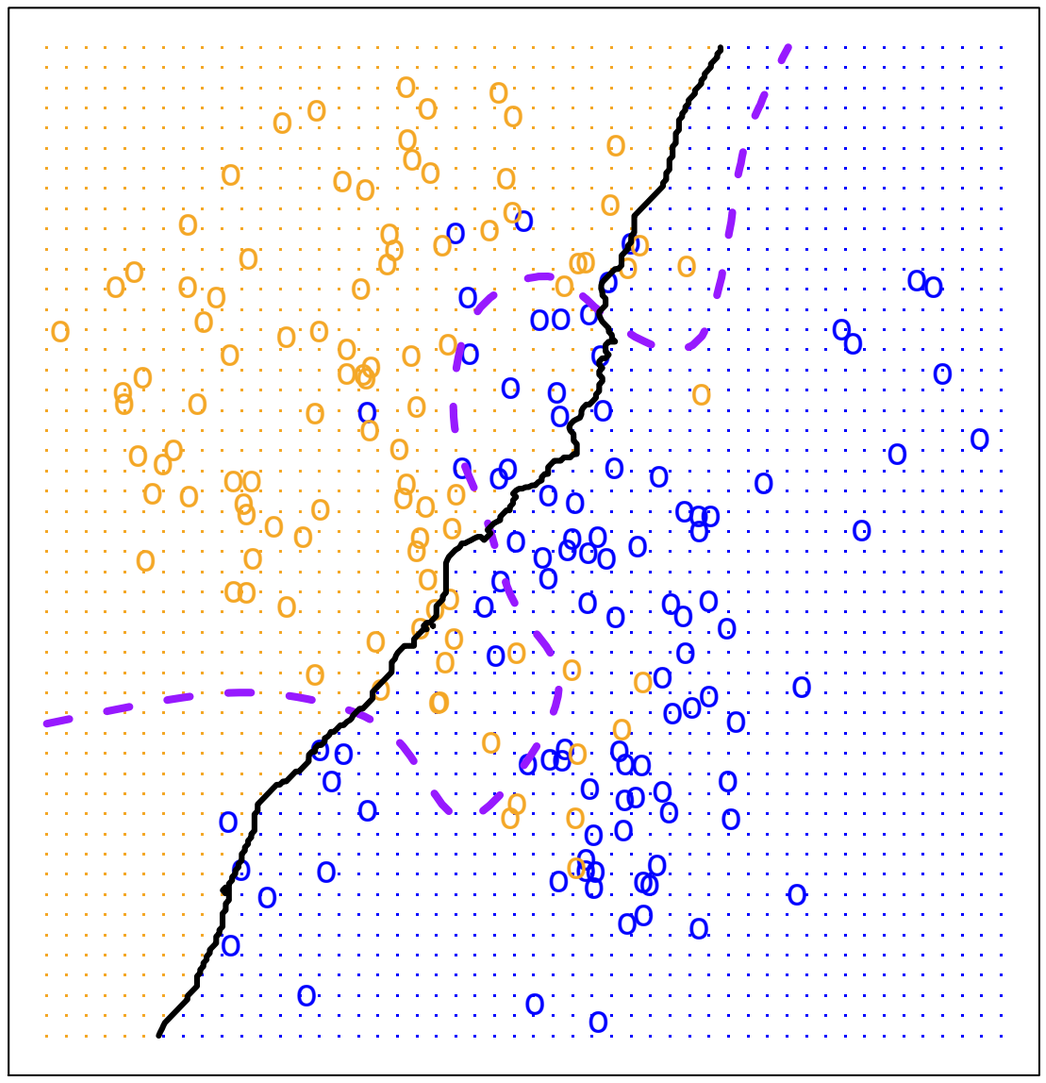
in real population)



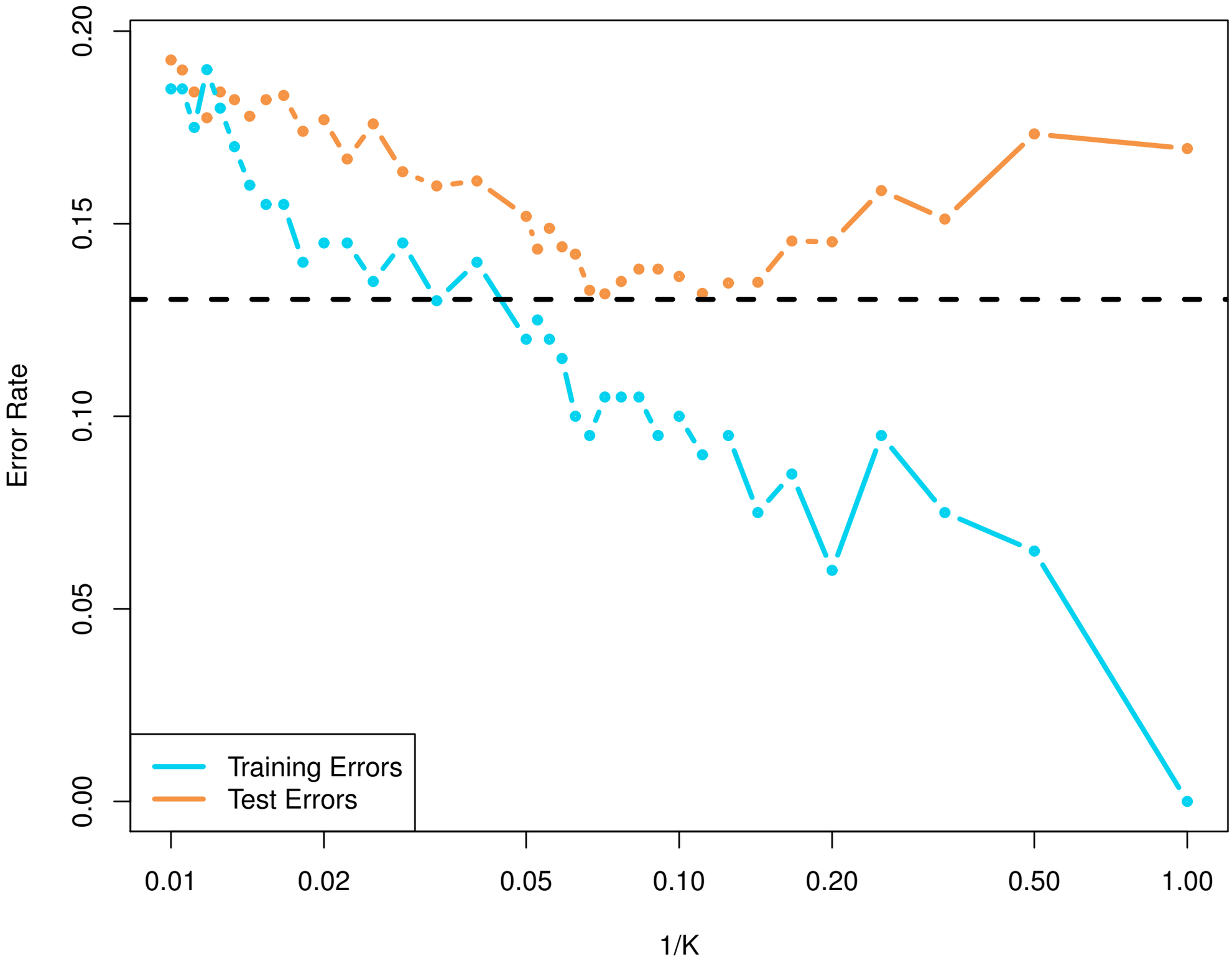
K=10-NN error rate 0.1363

Quality of predictions - K-Nearest Neighbor example





Quality of predictions - K-Nearest Neighbor example



Approximating the Bayes classifier

- KNN is a method for approximating the conditional probability and using the Bayes classifier. $P\left(Y=j|X=x_0\right)=\frac{1}{K}\sum_{i\in N_0}I(y_i=j)$

- **another way to this is Bayes theorem:** $P(A|B)P(B)=P(B|A)P(A)$

$$\pi_k \equiv P(Y=k) \quad \textbf{- prior}$$

$$f_k(X) \equiv P(X=x|Y=k) \quad \textbf{- likelihood}$$

$$p_k(X) \equiv P(Y=k|X=x) \quad \textbf{- posterior}$$

$$p_k(X)=\frac{\pi_k f_k(x)}{\sum_{l=1}^{K}\pi_l f_l(x)}$$

Approximating the Bayes classifier

- KNN is a method for approximating the conditional probability and using the Bayes classifier. $P\left(Y=j|X=x_0\right)=\frac{1}{K}\sum_{i\in N_0}I(y_i=j)$

- **another way to this is Bayes theorem:** $P(A|B)P(B)=P(B|A)P(A)$

$\pi_k \equiv P(Y=k)$ **- prior - easy to estimate**

$f_k(X)\equiv P(X=x|Y=k)$

$p_k(X)\equiv P(Y=k|X=x)$ **- posterior - we need some assumptions on the form**

$$p_k(X)=\frac{\pi_k f_k(x)}{\sum_{l=1}^{K}\pi_l f_l(x)}$$

Approximating the Bayes classifier - Linear Discriminant Analysis

- assume only one predictor (p=1) and $f_k(X)$ is a Gaussian.

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

- let (for simplicity): $\sigma \equiv \sigma_1 = \sigma_2 = \ldots = \sigma_K$

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

Independent <- of k

- our rule is that, given predictor x we assign class k with highest $p_k(x)$. This is the one with largest:

$$\delta_k(x) = x\,\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{(2\sigma^2)} + \log \pi_k$$

Linear Discriminant Analysis - example 1

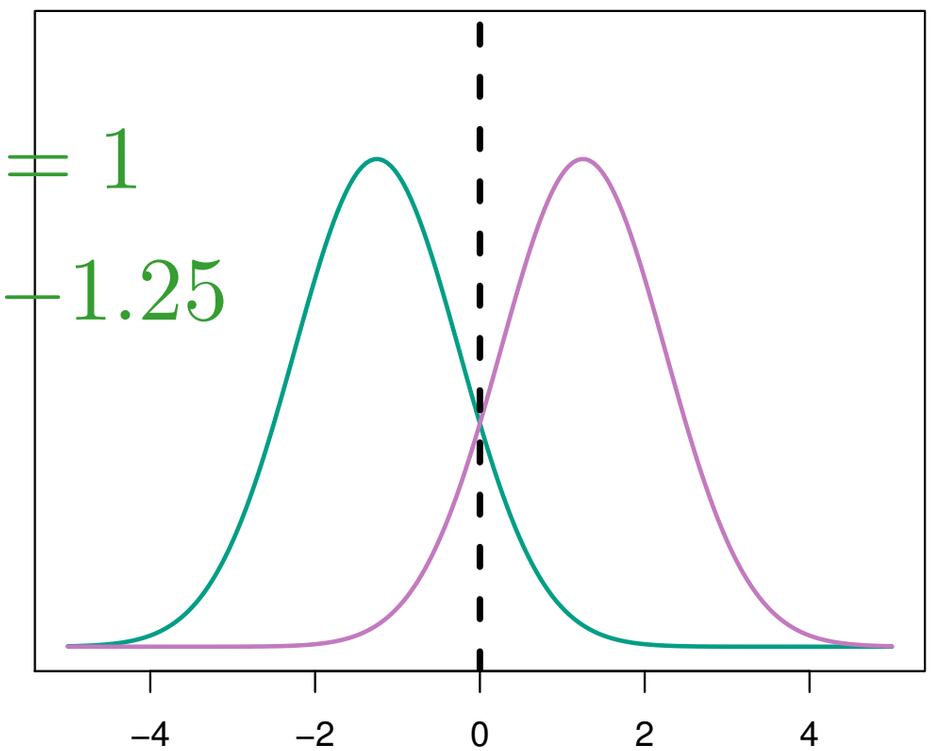
- assume 2 classes - $K = 2$
- represented in the same amount in the sample - $\pi_1 = \pi_2$ = 0.5
- Then the Bayes classifier assigns an observation to class 1 if:

$$2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2$$

- and the decision boundary is the average of the means $\frac{\mu_1 + \mu_2}{2}$

$$\sigma = 1$$

$$\mu_1 = -\mu_2 = -1.25$$



But, usually, we don't know the parameters of the distribution!

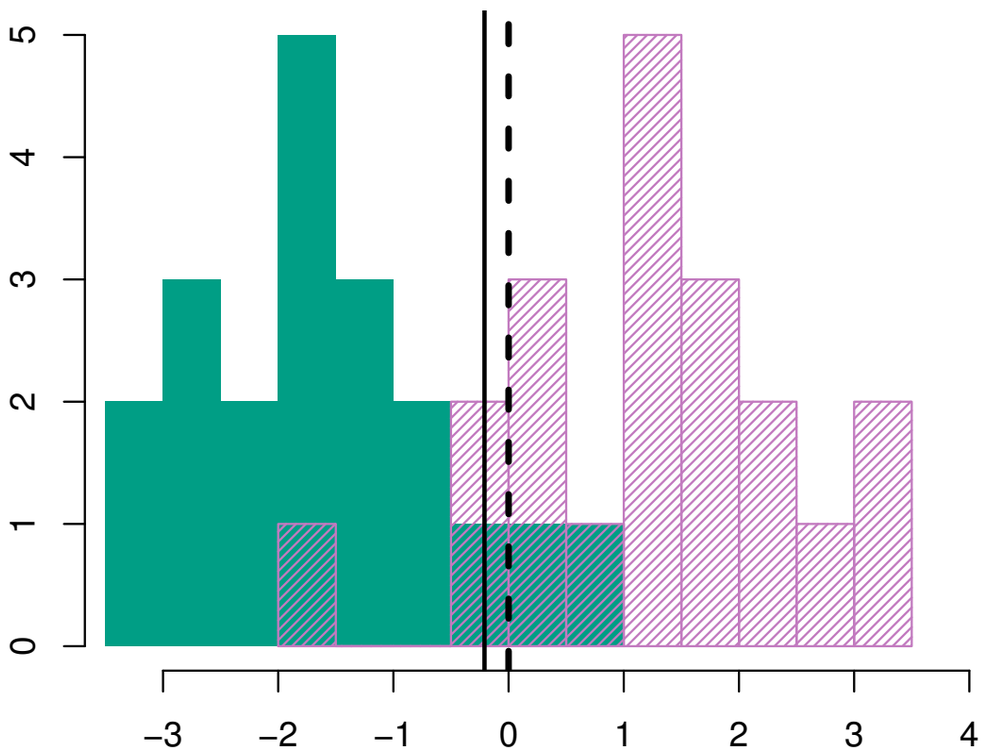
Linear Discriminant Analysis - example 2

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i \qquad \hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

- compute estimates of the parameters of the distribution

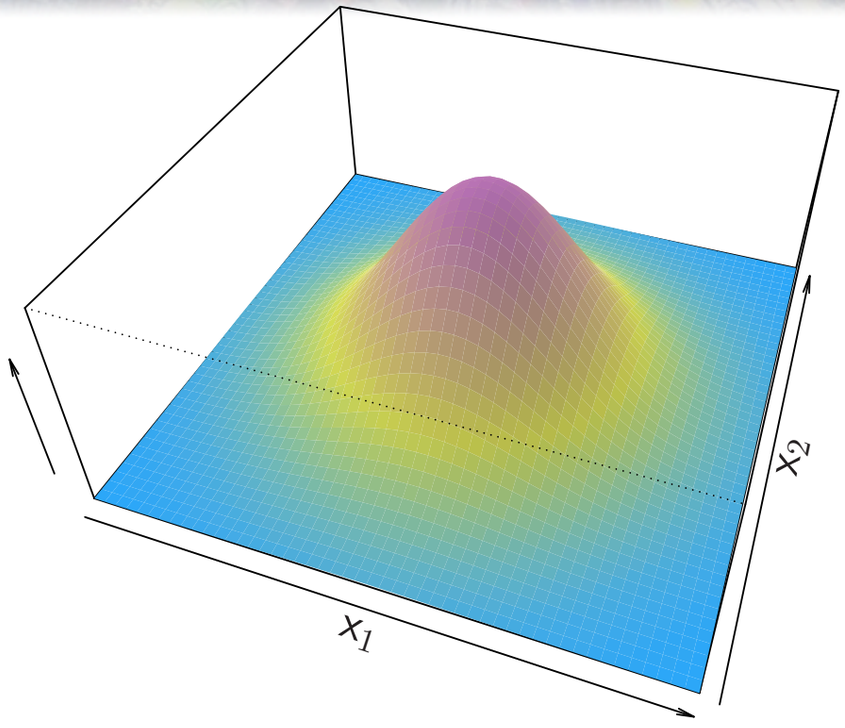
$$\hat{\delta}_k(x) = x \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{(2\hat{\sigma}^2)} + \log \hat{\pi}_k$$

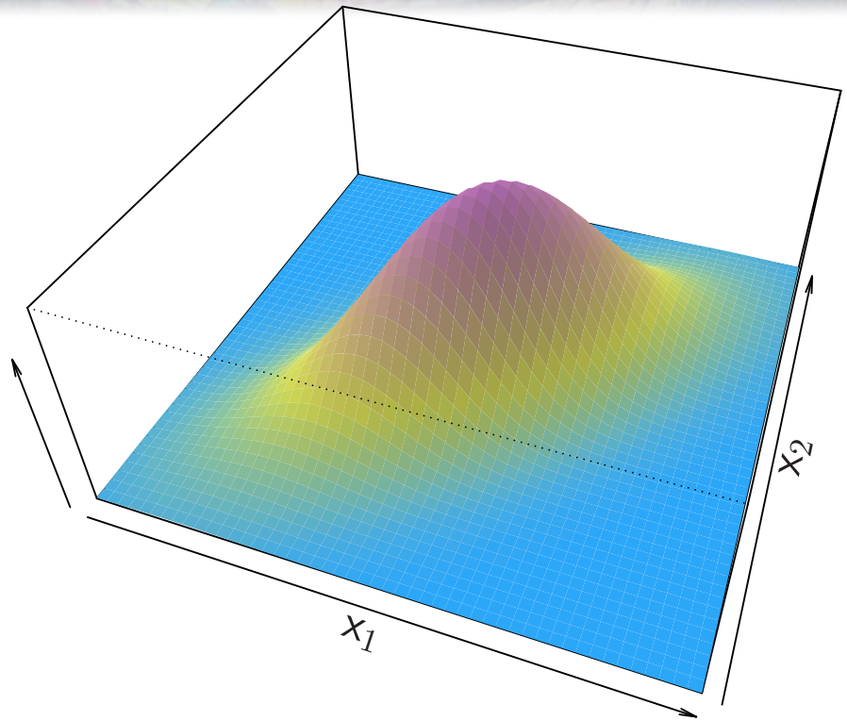


example 2 - based on 20 data point for each of the distribution from example 1

Bayes error rate = 0.106
LDA error rate = 0.111

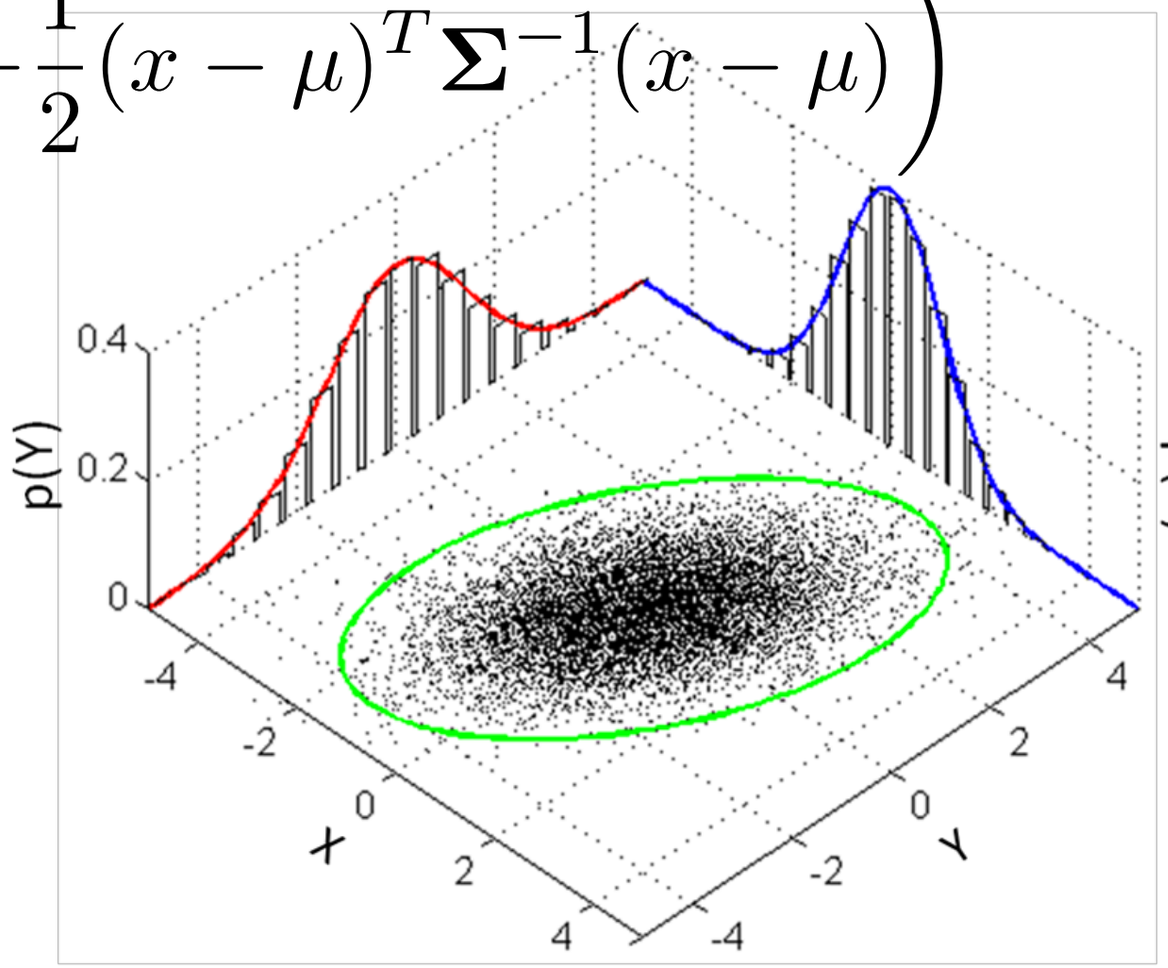
Linear Discriminant Analysis - multiple predictors - p>1



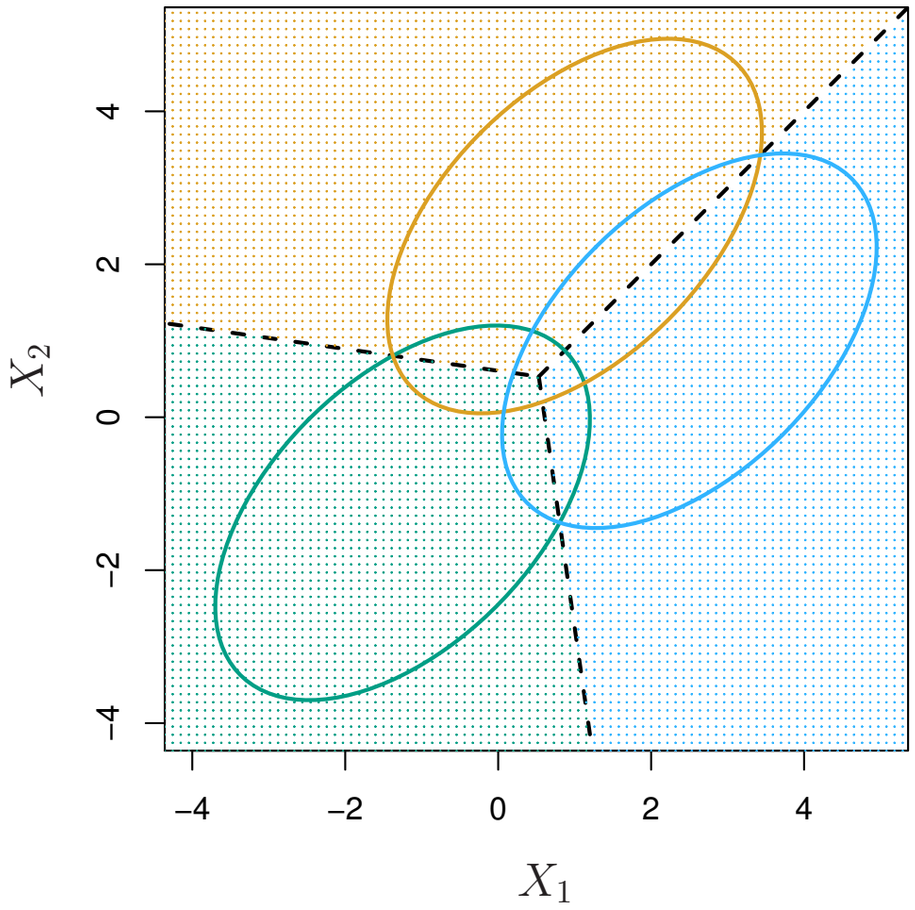


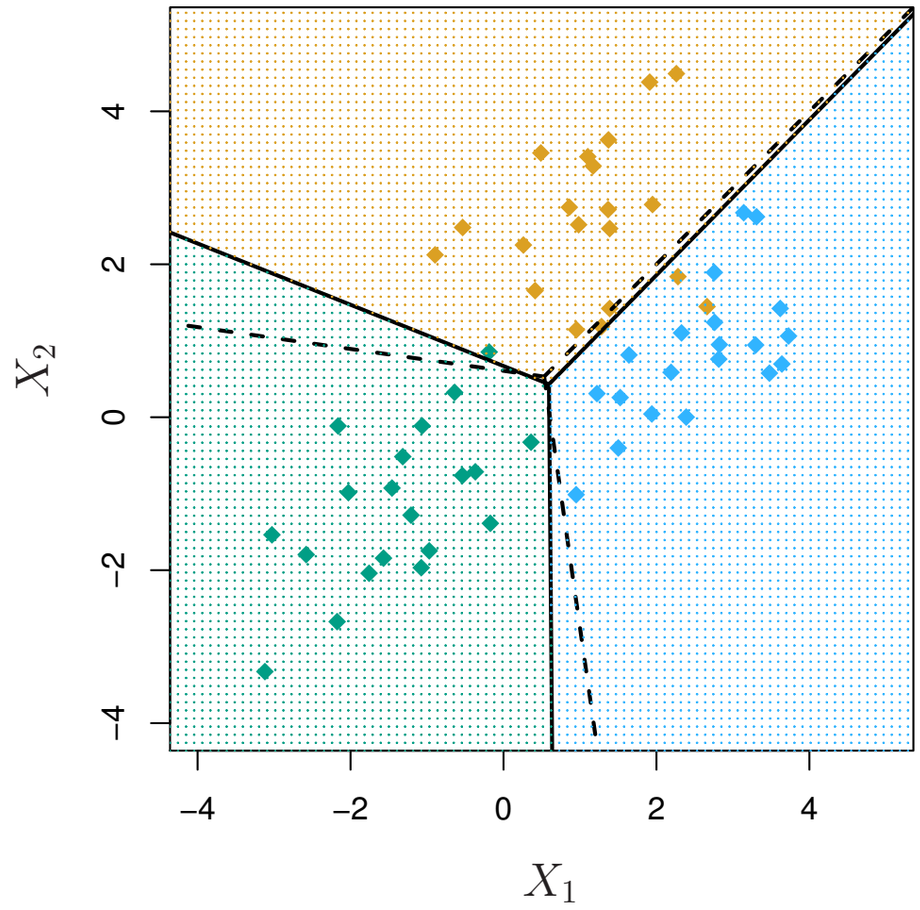
$$f_k(x) = \frac{1}{\sqrt{(2\pi)^p \det \mathbf{\Sigma}}} \exp\left(-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)\right)$$

- multivariate Gaussian distribution



Linear Discriminant Analysis - multiple predictors - p>1



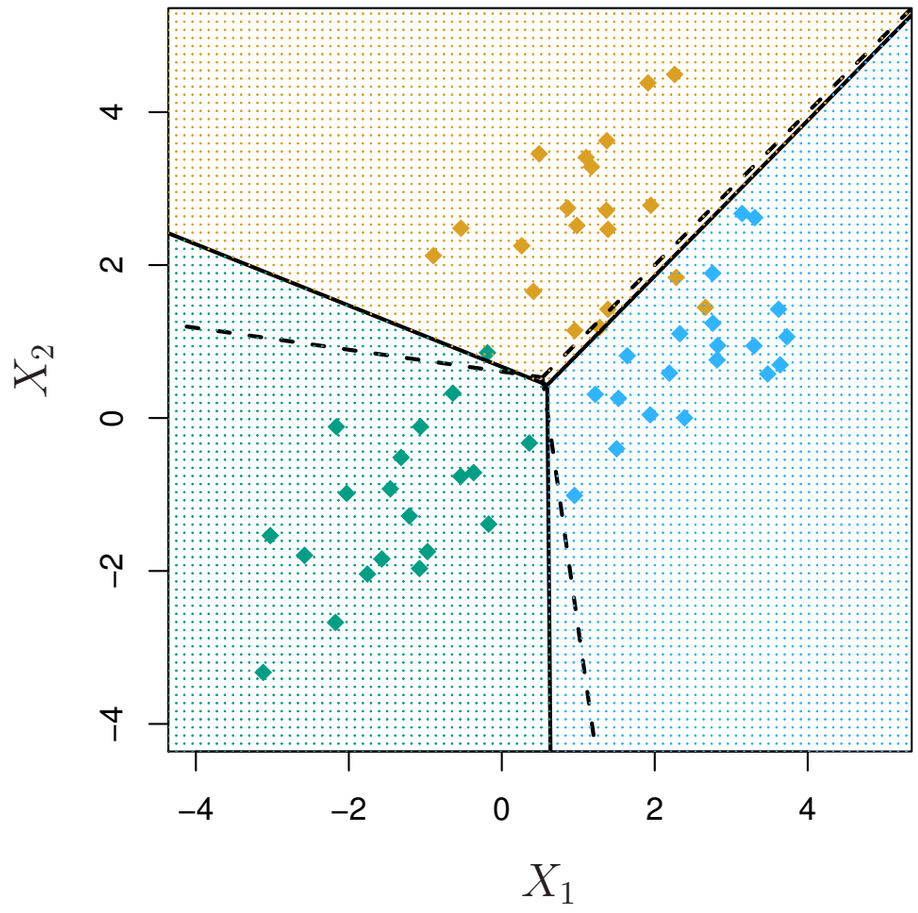


- our rule is that, given predictor x we assign class k with highest $p_k(x)$. This is the one with largest:

$$\hat{\delta}_k(x) = x^T \hat{\mathbf{\Sigma}}^{-1} \mu_k - \frac{1}{2} \mu_k^T \hat{\mathbf{\Sigma}}^{-1} \mu_k + \log \hat{\pi}_k$$

and the decision boundaries for: $\hat{\delta}_k(x) = \hat{\delta}_l(x)$

Linear Discriminant Analysis - multiple predictors - p>1



Bayes error rate = 0.0746
LDA error rate = 0.0770

- our rule is that, given predictor x we assign class k with highest $p_k(x)$. This is the one with largest:

$$\hat{\delta}_k(x) = x^T \hat{\mathbf{\Sigma}}^{-1} \mu_k - \frac{1}{2} \mu_k^T \hat{\mathbf{\Sigma}}^{-1} \mu_k + \log \hat{\pi}_k$$

and the decision boundaries for: $\hat{\delta}_k(x) = \hat{\delta}_l(x)$

What if the classes are not balanced?

- example: default data - consider weather an individual will default on his credit card payment (based on income, credit card balance, student or not) for a sample of 10 000

p=3, K=2

With LDA -> for training data

		True deafault status		
		No	Yes	Total
Predicted default status	No	9 644	252	9 896
	Yes	23	81	104
	Total	9 667	333	10 000

- training error rate 2,75 % (= (252 +23)/10000) low, but:
- only 3,33 % defaulted, so if we always say no, we achieve almost the same result!
- Two types of error here: **False Positive error rate 23 / 9667 ~ 0,24%** and **False Negative 252 / 333 ~ 75,7%** **!**
- The cost of different types of errors needs to be considered
- The Bayes classifier has the lowest **TOTAL** error rate

What if the classes are not balanced?

- Instead of assigning class for which posterior is greatest, for K=2:

$$P(\text{default} = \text{Yes}|X = x) > 0.5$$

- we assign when

$$P(\text{default} = \text{Yes}|X = x) > 0.2$$

Now **False Negative error rate 138 / 333 ~ 41,4%**

		True deafault status		
		No	Yes	Total
Predicted default status	No	9 644	252	9 896
	Yes	23	81	104
	Total	9 667	333	10 000

		True deafault status		
		No	Yes	Total
Predicted default status	No	9 432	138	9 570
	Yes	235	195	430
	Total	9 667	333	10 000

What if the classes are not balanced? -tresholds

- Instead of assigning class for which posterior is greatest, for K=2:

$$P(\text{default} = \text{Yes}|X = x) > 0.5$$

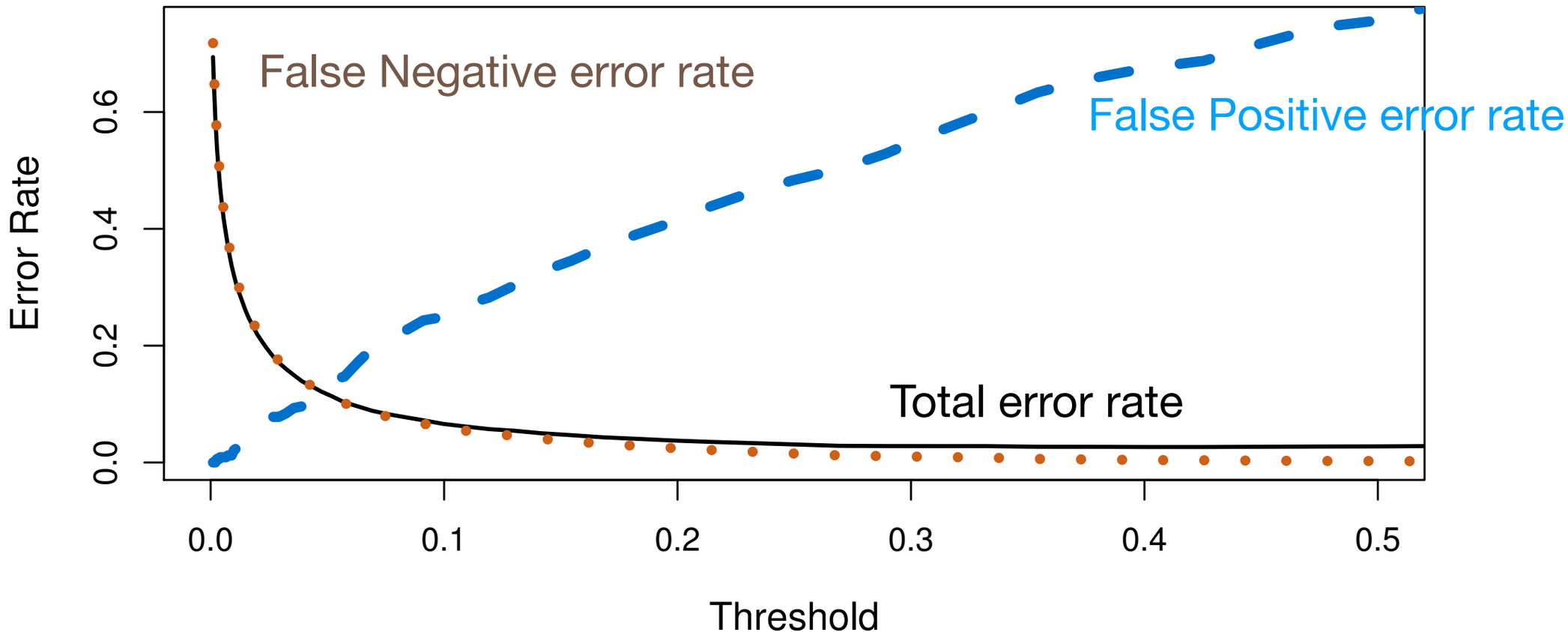
- we assign when

$$P(\text{default} = \text{Yes}|X = x) > 0.2$$

Now **False Negative error rate 138 / 333 ~ 41,4%**

		True deafault status		
		No	Yes	Total
Predicted default status	No	9 644	252	9 896
	Yes	23	81	104
	Total	9 667	333	10 000

		True deafault status		
		No	Yes	Total
Predicted default status	No	9 432	138	9 570
	Yes	235	195	430
	Total	9 667	333	10 000

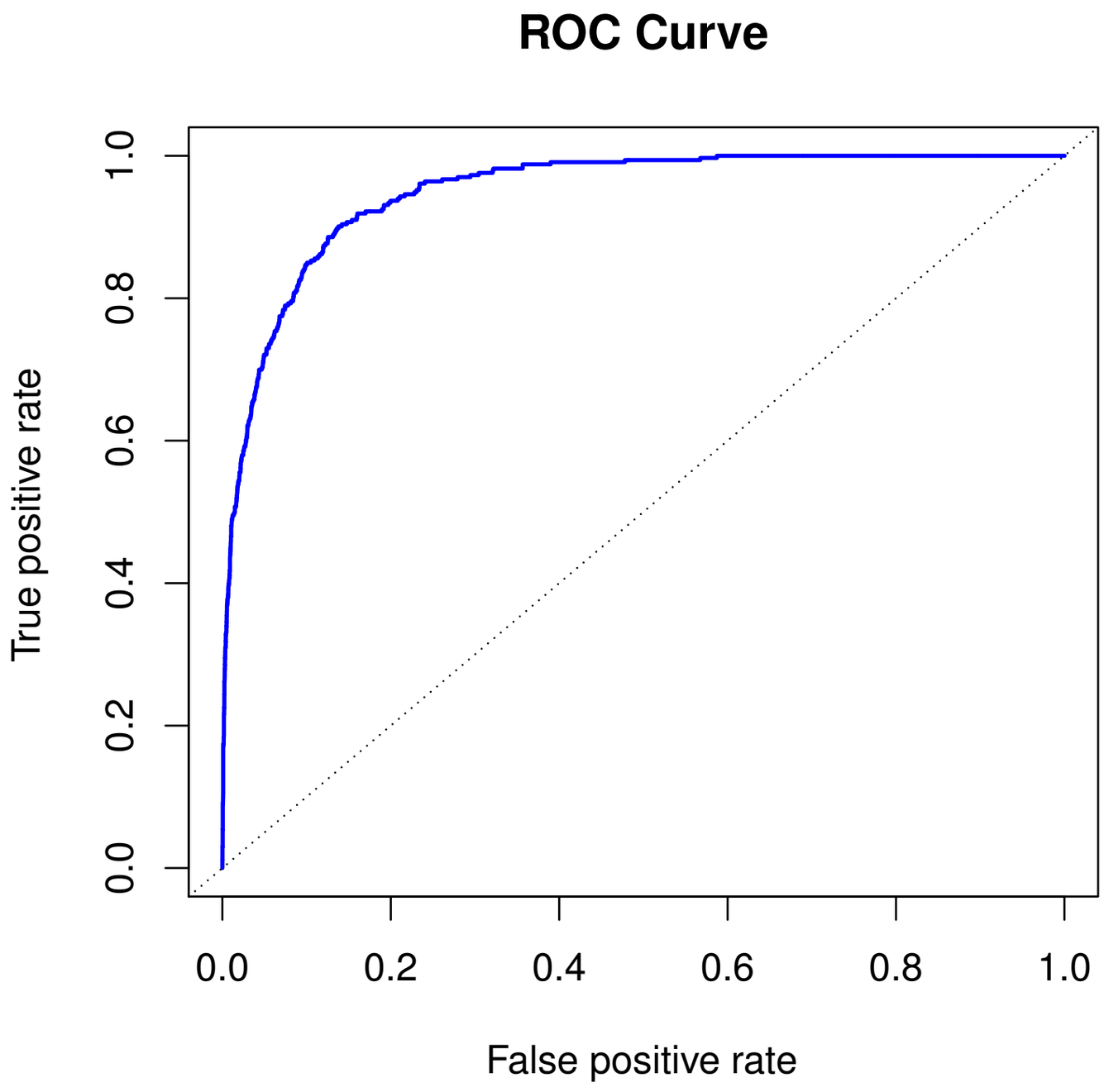


What if the classes are not balanced? - naming and measures

		Predicted class		
		- or Null	+ or Non-null	Total
True class	- or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

Name	Definition	Synonyms
False Pos. Rate	FP/N	Type I error
True Pos. Rate	TP/P	Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision
Neg. Pred. value	TN/N*	
Specificity	1-FP/N	
False discovery proportion	1-TP/P*	
F1 score	2 TP/ (P* + P)	
Accuracy	(TP+TN)/(N+P)	

What if the classes are not balanced? - ROC curve



- ROC curve (receiver operating characteristics -historic name)
- AUC - area under curve, hear 0.95 for LDA, train data. 0.5 for random - measure of accuracy independent of threshold
- Calculate both for training and test data

Quadratic Discriminant Analysis

- LDA - (probabilities of x in specific class) multivariate Gaussians with class specific mean and common covariance matrix

- what if we make covariance class specific?

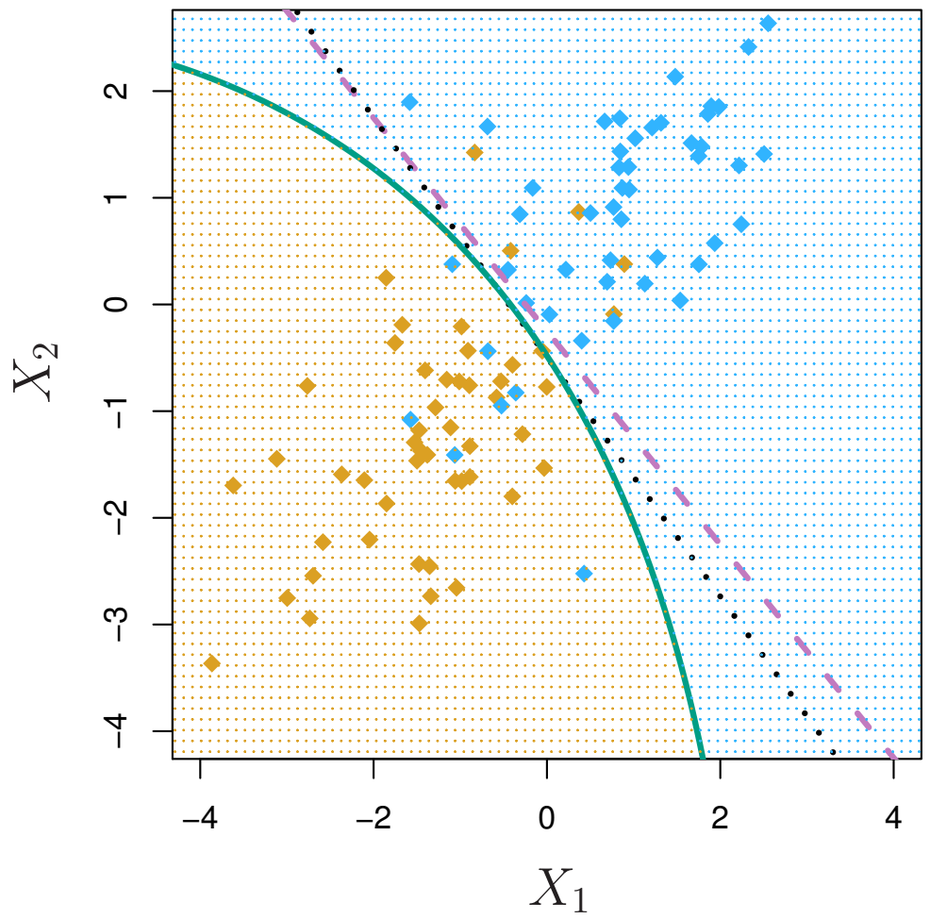
$$\hat{\delta}_k(x) = -\frac{1}{2}x^T\hat{\mathbf{\Sigma}}_k^{-1}x + x^T\hat{\mathbf{\Sigma}}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\hat{\mathbf{\Sigma}}_k^{-1}\mu_k - \frac{1}{2}\log|\det\hat{\mathbf{\Sigma}}_k| + \log\hat{\pi}_k$$

- QDA - Quadratic discriminant analysis

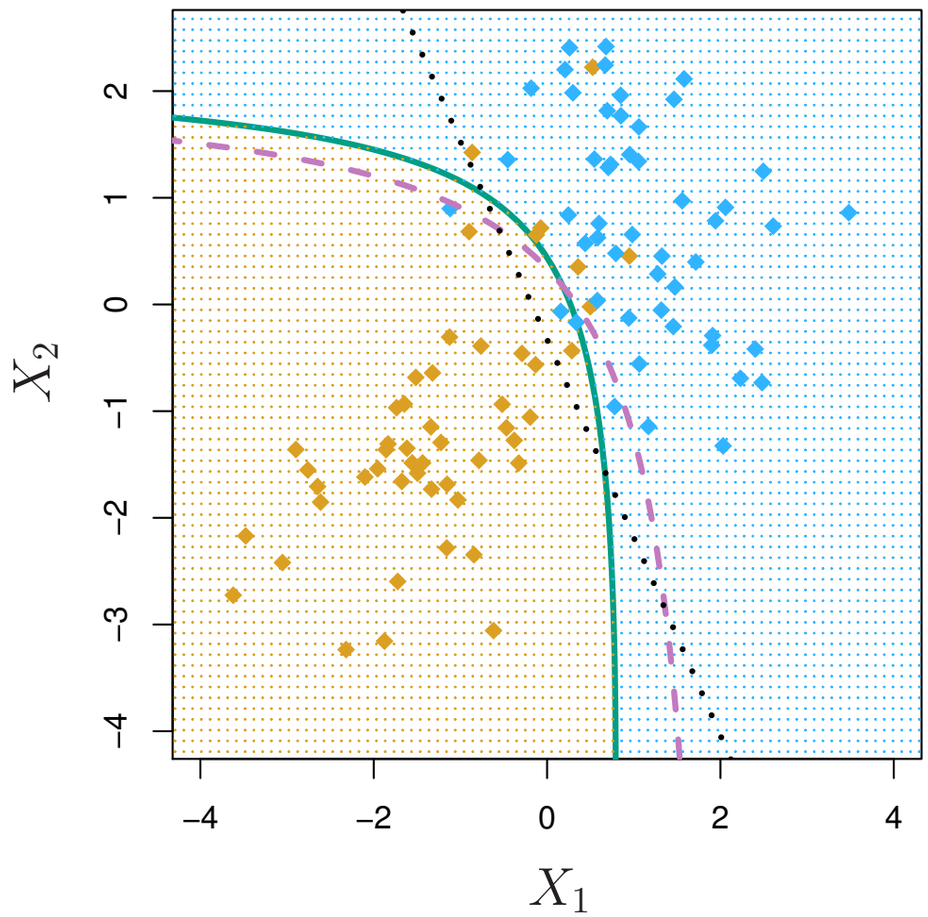
- note, for p predictors, p(p+1)/2 parameters for covariance matrix, times K for classes. LDA, a lot less.

- This translates into more flexibility but more likely overfitting if the p large and n comparably not so

Quadratic vs. Linear Discriminant Analysis



- same cov. of 0.7
- Bayes dec. boundary linear (in violet)
- QDA inferior



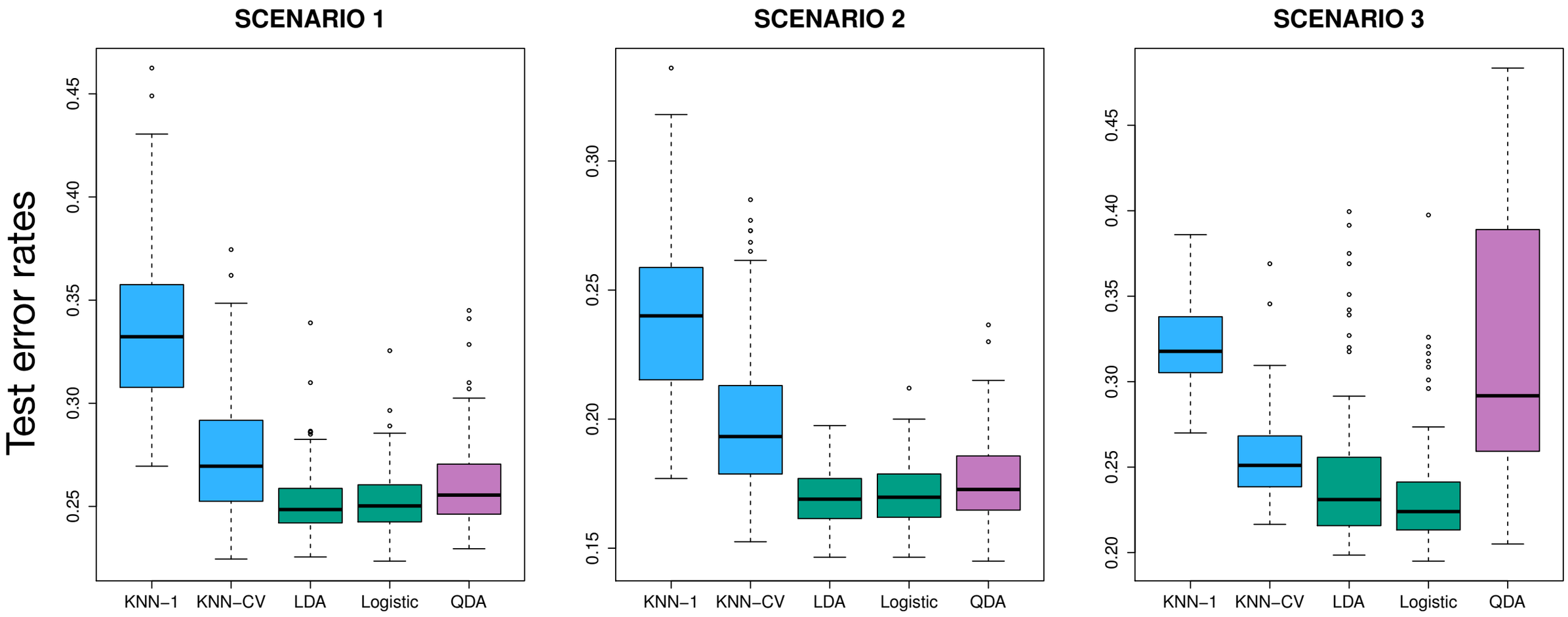
- cov. 0.7 and -0.7
- Bayes dec. boundary quadratic
- QDA superior

Comparison of KNN, Logistic Reg., LDA and QDA

- LDA and Log Reg - linear boundaries, different fitting.
- But, LDA Gaussianity assumption
- KNN completely non-parametric - no assumption about shape of decision boundary - dominates if db highly nonlinear
- but, we don’t now which predictors important like with Log reg.
- QDA a compromise between KNN and linear approaches

- 6 scenarios
- 100 random training data sets each
- 3 linear db
- 3 nonlinear db
- KNN with K=1 and based on a validation approach
- p=2 predictors

Comparison of KNN, Logistic Reg., LDA and QDA

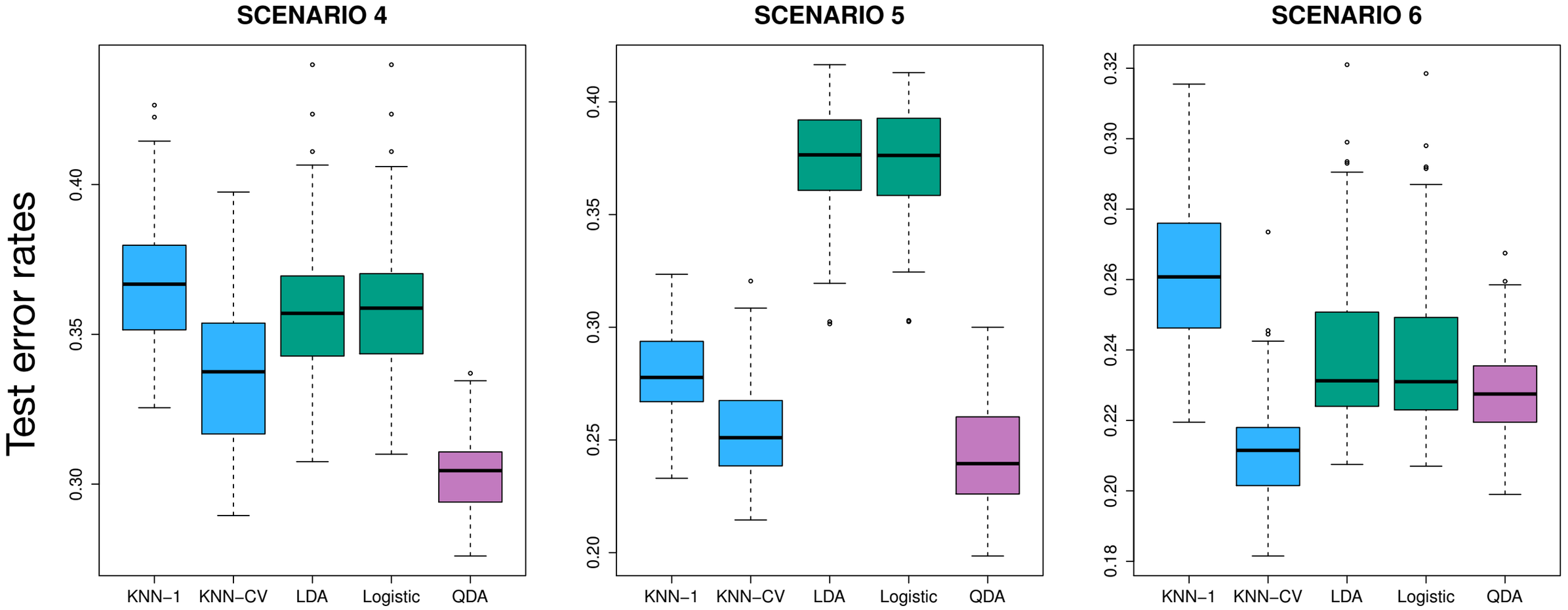


- 20 observation in each class
- uncorrelated normal
- different mean

- same, but
- predictors correlated with -0.5

- 50 observation in each class
- heavier distribution tails (non-normality)

Comparison of KNN, Logistic Reg., LDA and QDA



- normal
- correlations 0.5 and -0.5

- x1 and x2 normal
- uncorrelated
- but predictors x1^2, x2^2 and x1 x2

- highly non-linear db

One final model - Naive (or Simple, Independence) Bayes

- **strong assumption: predictors are uncorrelated (independent)**

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- look for k such that max

$$f_k(X) \equiv P(X = x|Y = k) = \prod_{i=1}^{p} P(x_i|Y = k)$$

discrete case: $$P(x_i|Y = k) = \frac{\sharp D\{X = x_i \wedge Y = k\}}{\sharp D\{Y = k\}}$$

- Fast, also for K>2, scales well to high n, requires small amount of data
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- Good for high dimensional data when model complexity not important
- For continuous predictors use independent continuous distributions (ex.: Gaussian (NB)) (and estimate its parameters)
- in applications, smoothing (Laplace j=1) required to get reasonable results if the training data doesn't include any points from some class(es)

One final model - Naive (or Simple, Independence) Bayes

- **strong assumption: predictors are uncorrelated (independent)**

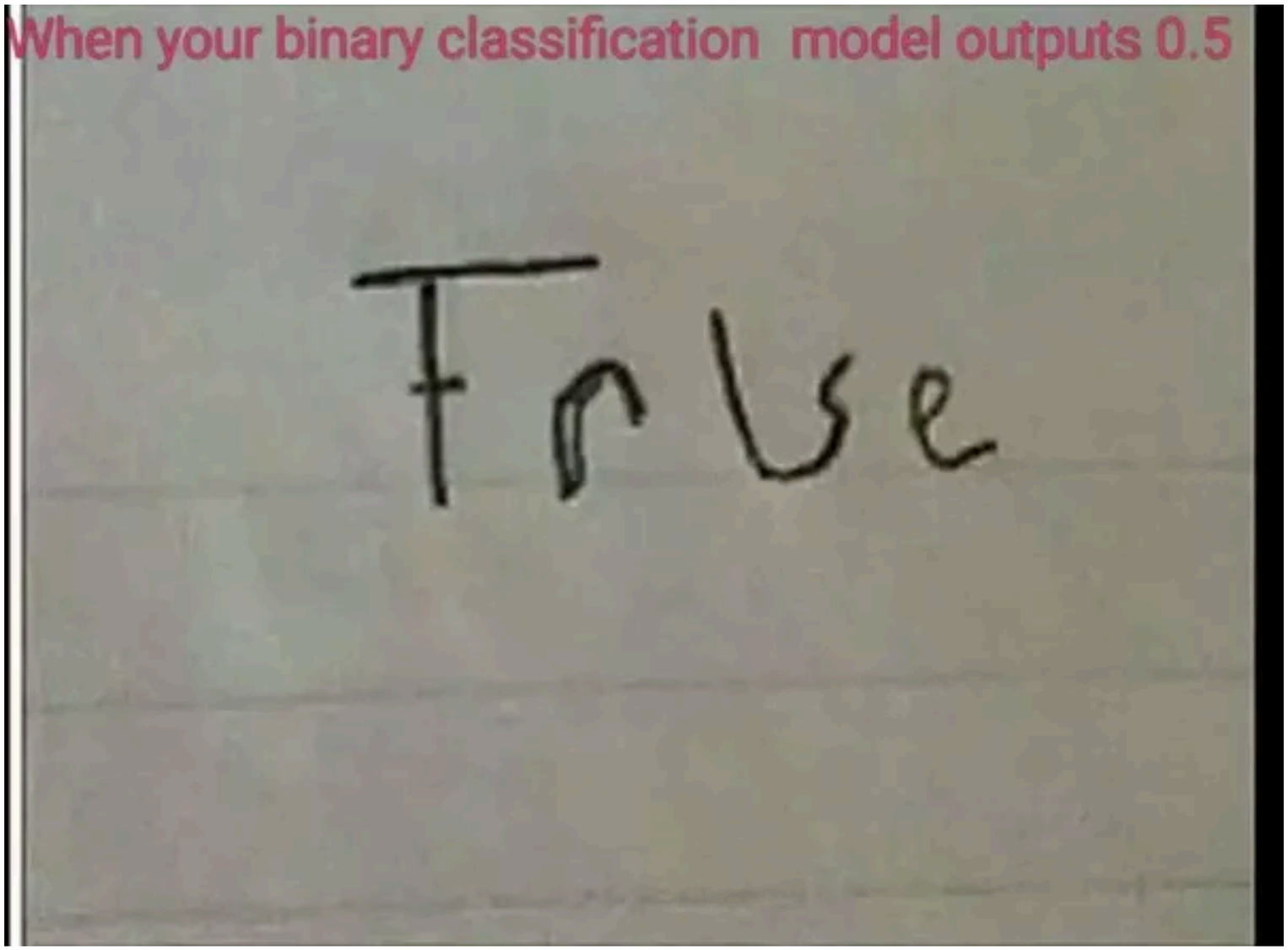
$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- look for k such that max

$$f_k(X) \equiv P(X = x|Y = k) = \prod_{i=1}^{p} P(x_i|Y = k)$$

discrete case: $$P(x_i|Y = k) = \frac{\sharp D\{X_i = x_i \wedge Y = k\} + j}{\sharp D\{Y = k\} + j|X_i|}$$

- Fast, also for K>2, scales well to high n, requires small amount of data
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- Good for high dimensional data when model complexity not important
- For continuous predictors use independent continuous distributions (ex.: Gaussian (NB)) (and estimate its parameters)
- in applications, smoothing (Laplace j=1) required to get reasonable results if the training data doesn't include any points from some class(es)



"Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani "