Classification

web.stanford.edu/class/stats202

Sergio Bacallado, Jonathan Taylor

Autumn 2022

Basic approach

- Supervised learning with a **qualitative or categorical** response.
- Just as common, if not more common than regression:

1. *Medical diagnosis:* Given the symptoms a patient shows, predict which of 3 conditions they are attributed to.

2. *Online banking*: Determine whether a transaction is fraudulent or not, on the basis of the IP address, client's history, etc.

3. *Web searching:* Based on a user's history, location, and the string of a web search, predict which link a person is likely to click.

4. *Online advertising:* Predict whether a user will click on an ad or not.

Bayes classifier

- Suppose $P(Y \mid X)$ is known. Then, given an input x_0, we predict the response

$$\hat{y}_0 = \operatorname{argmax}_y P(Y = y \mid X = x_0).$$

- The Bayes classifier minimizes the expected 0-1 loss:

$$E\left[\frac{1}{m}\sum_{i=1}^{m} \mathbf{1}(\hat{y}_i \neq y_i)\right]$$

- This minimum 0-1 loss (the best we can hope for) is the **Bayes error rate**.

Basic strategy: estimate $P(Y \mid X)$

- If we have a good estimate for the conditional probability $\hat{P}(Y \mid X)$, we can use the classifier:

$$\hat{y}_0 = \text{argmax}_{\ y}\ \hat{P}(Y = y \mid X = x_0).$$

- Suppose Y is a binary variable. Could we use a linear model?

$$P(Y = 1|X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_1 X_p$$

- Problems:
  - *This would allow probabilities* < 0 *and* > 1.
  - *Difficult to extend to more than 2 categories.*

Logistic regression

- We model the joint probability as:

$$P(Y = 1 \mid X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$
$$P(Y = 0 \mid X) = \frac{1}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

This is the same as using a linear model for the log odds:

$$\log\left[\frac{P(Y = 1 \mid X)}{P(Y = 0 \mid X)}\right] = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p.$$

Fitting logistic regression

- The training data is a list of pairs $(y_1, x_1), (y_2, x_2), \ldots, (y_n, x_n)$.
- We don't observe the left hand side in the model

$$\log\left[\frac{P(Y = 1 \mid X)}{P(Y = 0 \mid X)}\right] = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p,$$

- $\Longrightarrow$ We cannot use a least squares fit.

Likelihood

- **Solution:** The likelihood is the probability of the training data, for a fixed set of coefficients $\beta_0, \ldots, \beta_p$:

$$\prod_{i=1}^{n} P(Y = y_i \mid X = x_i)$$

- We can rewrite as

$$\prod_{i=1}^{n} \left(\frac{e^{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}}}{1 + e^{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}}} \right)^{y_i} \left(\frac{1}{1 + e^{\beta_0 + \beta_1 x_{j1} + \cdots + \beta_p x_{jp}}} \right)^{1 - y_i}$$

- Choose estimates $\hat{\beta}_0, \ldots, \hat{\beta}_p$ which maximize the likelihood.

- Solved with numerical methods (e.g. Newton's algorithm).

Logistic regression in R

```
library(ISLR2)
glm.fit = glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
            family=binomial, data=Smarket)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Smarket)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.446  -1.203   1.065   1.145   1.326
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.126000   0.240736  -0.523    0.601
## Lag1        -0.073074   0.050167  -1.457    0.145
## Lag2        -0.042301   0.050086  -0.845    0.398
## Lag3         0.011085   0.049939   0.222    0.824
## Lag4         0.009359   0.049974   0.187    0.851
## Lag5         0.010313   0.049511   0.208    0.835
## Volume       0.135441   0.158360   0.855    0.392
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.2  on 1249  degrees of freedom
## Residual deviance: 1727.6  on 1243  degrees of freedom
## AIC: 1741.6
##
## Number of Fisher Scoring iterations: 3
```

Inference for logistic regression

1. We can estimate the Standard Error of each coefficient.

2. The z-statistic is the equivalent of the t-statistic in linear regression:

$$z = \frac{\hat{\beta}_j}{\mathrm{SE}(\hat{\beta}_j)}.$$

3. The p-values are test of the null hypothesis $\beta_j = 0$ (Wald's test).

4. Other possible hypothesis tests: likelihood ratio test (chi-square distribution).

Example: Predicting credit card `default`

Predictors:

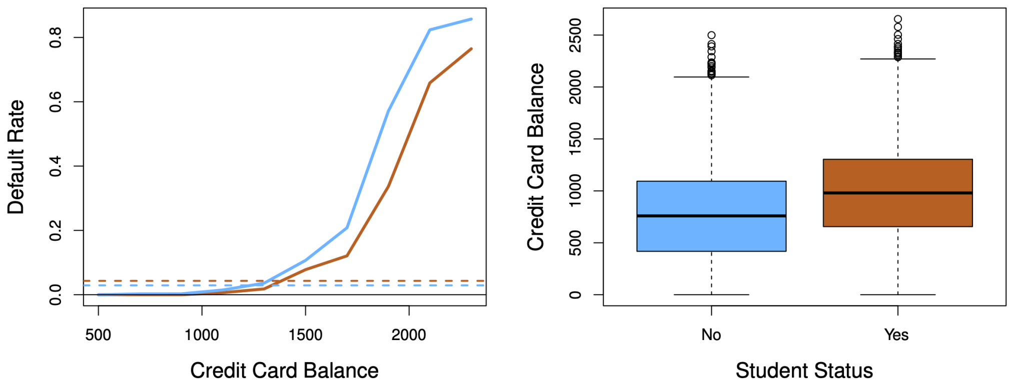
- `student`: 1 if student, 0 otherwise
- `balance`: credit card balance
- `income`: person's income.

Confounding

In this dataset, there is *confounding*, but little collinearity.

- Students tend to have higher balances. So, `balance` is explained by `student`, but not very well.
- People with a high `balance` are more likely to default.
- Among people with a given `balance`, students are less likely to default.

Results: predicting credit card `default`



Confounding in `Default` data

Using only `balance`

```r
summary(glm(default ~ balance,
        family=binomial, data=Default))
```

```
##
## Call:
## glm(formula = default ~ balance, family = binomial, data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2697  -0.1465  -0.0589  -0.0221   3.7589
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
## balance      5.499e-03  2.204e-04   24.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1596.5  on 9998  degrees of freedom
## AIC: 1600.5
##
## Number of Fisher Scoring iterations: 8
```

Using only `student`

```
summary(glm(default ~ student,
       family=binomial, data=Default))
```

```
##
## Call:
## glm(formula = default ~ student, family = binomial, data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.2970  -0.2970  -0.2434  -0.2434   2.6585
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
## studentYes   0.40489    0.11502    3.52 0.000431 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 2908.7  on 9998  degrees of freedom
## AIC: 2912.7
##
## Number of Fisher Scoring iterations: 6
```

Using both `balance` and `student`

```r
summary(glm(default ~ balance + student,
        family=binomial, data=Default))
```

```
##
## Call:
## glm(formula = default ~ balance + student, family = binomial,
##     data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4578  -0.1422  -0.0559  -0.0203   3.7435
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.075e+01  3.692e-01 -29.116  < 2e-16 ***
## balance      5.738e-03  2.318e-04  24.750  < 2e-16 ***
## studentYes  -7.149e-01  1.475e-01  -4.846 1.26e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.7  on 9997  degrees of freedom
## AIC: 1577.7
##
## Number of Fisher Scoring iterations: 8
```

Using all 3 predictors

```
summary(glm(default ~ balance + income + student,
       family=binomial, data=Default))
```

```
##
## Call:
## glm(formula = default ~ balance + income + student, family = binomial,
##     data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4691  -0.1418  -0.0557  -0.0203   3.7383
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
##
## Number of Fisher Scoring iterations: 8
```

Multinomial logistic regression

- Extension of logistic regression to more than 2 categories

- Suppose Y takes values in $\{1, 2, \ldots, K\}$, then we can use a linear model for the log odds against a baseline category (e.g. 1): for $j \neq 1$

$$\log\left[\frac{P(Y = j \mid X)}{P(Y = 1 \mid X)}\right] = \beta_{0,j} + \beta_{1,j}X_1 + \cdots + \beta_{p,j}X_p$$

- In this case $\beta \in \mathbb{R}^{p\times(K-1)}$ is a *matrix* of coefficients.

Some potential problems

- The coefficients become unstable when there is collinearity. Furthermore, this affects the convergence of the fitting algorithm.

- When the classes are well separated, the coefficients become unstable. This is always the case when $p \geq n - 1$. In this case, prediction error is low, but $\hat{\beta}$ is very variable.

Linear Discriminant Analysis (LDA)

- **Strategy:** Instead of estimating $P(Y \mid X)$ directly, we could estimate:

  1. $\hat{P}(X \mid Y)$: *Given the response, what is the distribution of the inputs.*
  2. $\hat{P}(Y)$: *How likely are each of the categories.*

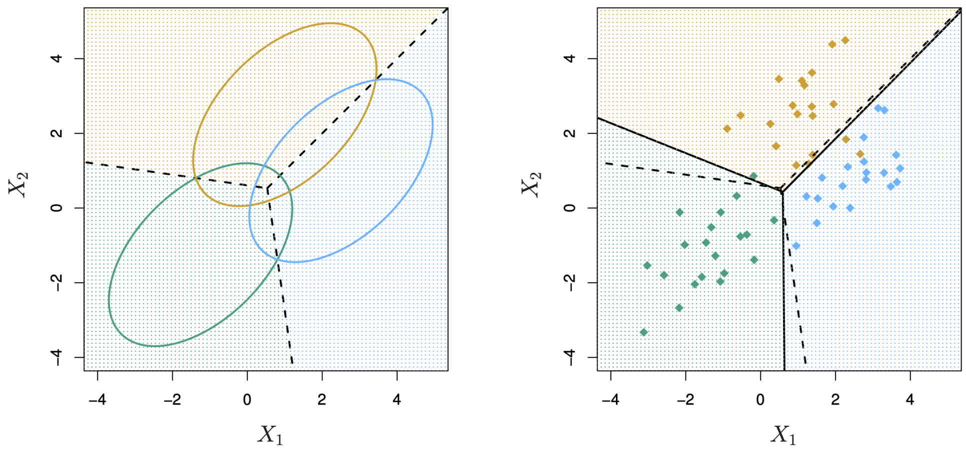
- Then, we use *Bayes rule* to obtain the estimate:

$$\hat{P}(Y = k \mid X = x) = \frac{\hat{P}(X = x \mid Y = k)\hat{P}(Y = k)}{\hat{P}(X = x)}$$
$$= \frac{\hat{P}(X = x \mid Y = k)\hat{P}(Y = k)}{\sum_{j=1}^{K} \hat{P}(X = x \mid Y = j)\hat{P}(Y = j)}$$

LDA: multivariate normal with equal covariance

- LDA is the special case of the above strategy when $P(X \mid Y = k) = N(\mu_k, \mathbf{\Sigma})$.
- That is, within each class the features have multivariate normal distribution with center depending on the class and **common covariance $\mathbf{\Sigma}$.**
- The probabilities $P(Y = k)$ are estimated by the fraction of training samples of class k.

Decision boundaries



Density contours and decision boundaries for LDA with three classes.

LDA has (piecewise) linear decision boundaries

Suppose that:

1. We know $P(Y = k) = \pi_k$ exactly.

2. $P(X = x | Y = k)$ is Mutivariate Normal with density:

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \boldsymbol{\Sigma}^{-1}(x-\mu_k)}$$

3. Above: μ_k : Mean of the inputs for category k and $\boldsymbol{\Sigma}$: covariance matrix (common to all categories)

Then, what is the Bayes classifier?

- By Bayes rule, the probability of category k, given the input x is:

$$P(Y = k \mid X = x) = \frac{f_k(x)\pi_k}{P(X = x)}$$

- The denominator does not depend on the response k, so we can write it as a constant:

$$P(Y = k \mid X = x) = C \times f_k(x)\pi_k$$

- Now, expanding $f_k(x)$:

$$P(Y = k \mid X = x) = \frac{C\pi_k}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)}$$

- Let's absorb everything that does not depend on k into a constant C':

$$P(Y = k \mid X = x) = C'\pi_k e^{-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)}$$

- Take the logarithm of both sides:

$$\log P(Y = k \mid X = x) = \log C' + \log \pi_k - \frac{1}{2}(x - \mu_k)^T \boldsymbol{\Sigma}^{-1}(x - \mu_k).$$

- This is the same for every category, k.

- We want to find the maximum of this expression over k.

- Goal is to maximize the following over k:

$$
\begin{aligned}
&\log \pi_k - \frac{1}{2}(x-\mu_k)^T \mathbf{\Sigma}^{-1}(x-\mu_k). \\
=&\log \pi_k - \frac{1}{2}\left[x^T \mathbf{\Sigma}^{-1} x + \mu_k^T \mathbf{\Sigma}^{-1} \mu_k\right] + x^T \mathbf{\Sigma}^{-1} \mu_k \\
=&C'' + \log \pi_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1} \mu_k + x^T \mathbf{\Sigma}^{-1} \mu_k
\end{aligned}
$$

- We define the objectives (called *discriminant functions*):

$$
\delta_k(x) = \log \pi_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1} \mu_k + x^T \mathbf{\Sigma}^{-1} \mu_k
$$

At an input x, we predict the response with the highest $\delta_k(x)$.

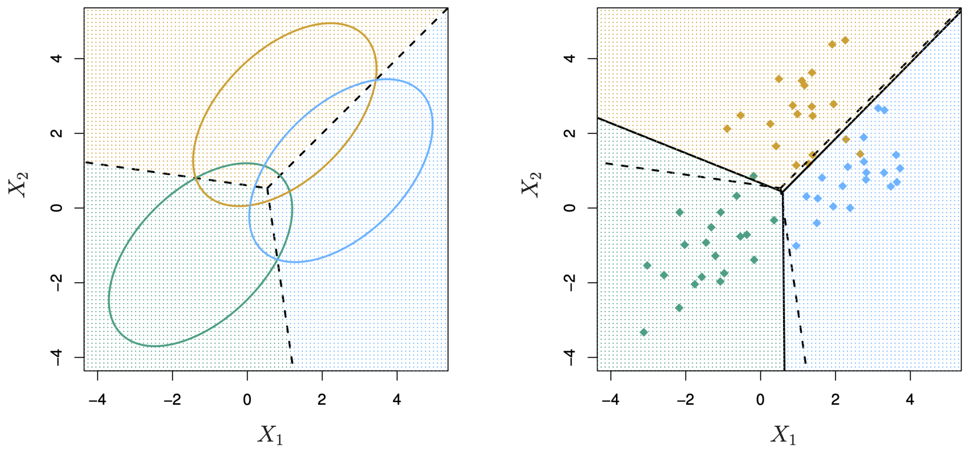
Decision boundaries

- What are the decision boundaries? It is the set of points x in which 2 classes do just as well (i.e. the discriminant functions of the two classes agree at x):

$$
\begin{aligned}
\delta_k(x) &= \delta_\ell(x) \\
\log \pi_k - \frac{1}{2}\mu_k^T \boldsymbol{\Sigma}^{-1} \mu_k + x^T \boldsymbol{\Sigma}^{-1} \mu_k &= \log \pi_\ell - \frac{1}{2}\mu_\ell^T \boldsymbol{\Sigma}^{-1} \mu_\ell + x^T \boldsymbol{\Sigma}^{-1} \mu_\ell
\end{aligned}
$$

- This is a linear equation in x.

Decision boundaries revisited



Density contours and decision boundaries for LDA with three classes.

Estimating π_k

$$\hat{\pi}_k = \frac{\#\{i \;;\; y_i = k\}}{n}$$

- In English: the fraction of training samples of class k.

Estimating the parameters of $f_k(x)$

Estimate the center of each class μ_k:

$$\hat{\mu}_k = \frac{1}{\#\{i \;;\; y_i = k\}} \sum_{i \;;\; y_i = k} x_i$$

- Estimate the common covariance matrix $\mathbf{\Sigma}$:

- One predictor ($p = 1$):

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

- Many predictors ($p > 1$): Compute the vectors of deviations $(x_1 - \hat{\mu}_{y_1}), (x_2 - \hat{\mu}_{y_2}), \ldots, (x_n - \hat{\mu}_{y_n})$ and use an unbiased estimate of its covariance matrix, $\mathbf{\Sigma}$.

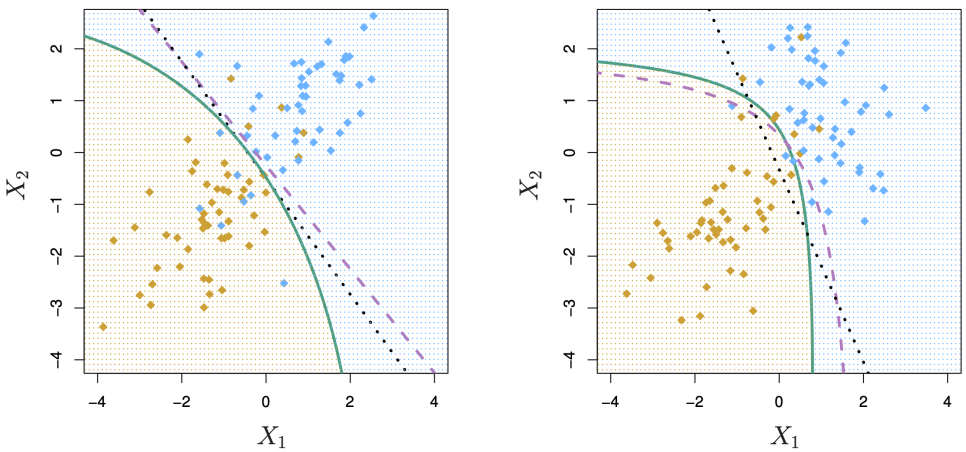
Final decision rule

- For an input x, predict the class with the largest:

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

- The decision boundaries are defined by $\{x : \delta_k(x) = \delta_\ell(x)\}$, $1 \leq j, \ell \leq K$.

Quadratic discriminant analysis (QDA)



Comparison of LDA and QDA boundaries

- The assumption that the inputs of every class have the same covariance $\mathbf{\Sigma}$ can be quite restrictive.
- Bayes boundary (– – – – – ––), LDA (· · ·), QDA (– – – – – – – ––).

QDA: multivariate normal with differing covariance

- In **quadratic discriminant analysis** we estimate a mean $\hat{\mu}_k$ and a covariance matrix $\hat{\mathbf{\Sigma}}_k$ for each class separately.

- Given an input, it is easy to derive an objective function:

$$\delta_k(x) = \log \pi_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}_k^{-1} \mu_k + x^T \mathbf{\Sigma}_k^{-1} \mu_k - \frac{1}{2} x^T \mathbf{\Sigma}_k^{-1} x - \frac{1}{2} \log |\mathbf{\Sigma}_k|$$

- This objective is now quadratic in x and so the decision boundaries are 0s of quadratic functions.

Evaluating a classification method

- We have talked about the 0-1 loss:

$$\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}(y_i \neq \hat{y}_i).$$

- It is possible to make the wrong prediction for some classes more often than others. The 0-1 loss doesn't tell you anything about this.

- A much more informative summary of the error is a **confusion matrix**:

		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*	

Confusion matrix for a 2 class problem

Confusion matrix for `Default` example

```
library(MASS) # where the `lda` function lives
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
lda.fit = predict(lda(default ~ balance + student, data=Default))
table(lda.fit$class, Default$default)
```

```
##
##          No  Yes
##     No 9644  252
##     Yes  23   81
```

1. The error rate among people who do **not** default (false positive rate) is very low.
2. However, the rate of false negatives is 76%.
3. It is possible that false negatives are a bigger source of concern!
4. One possible solution: Change the threshold

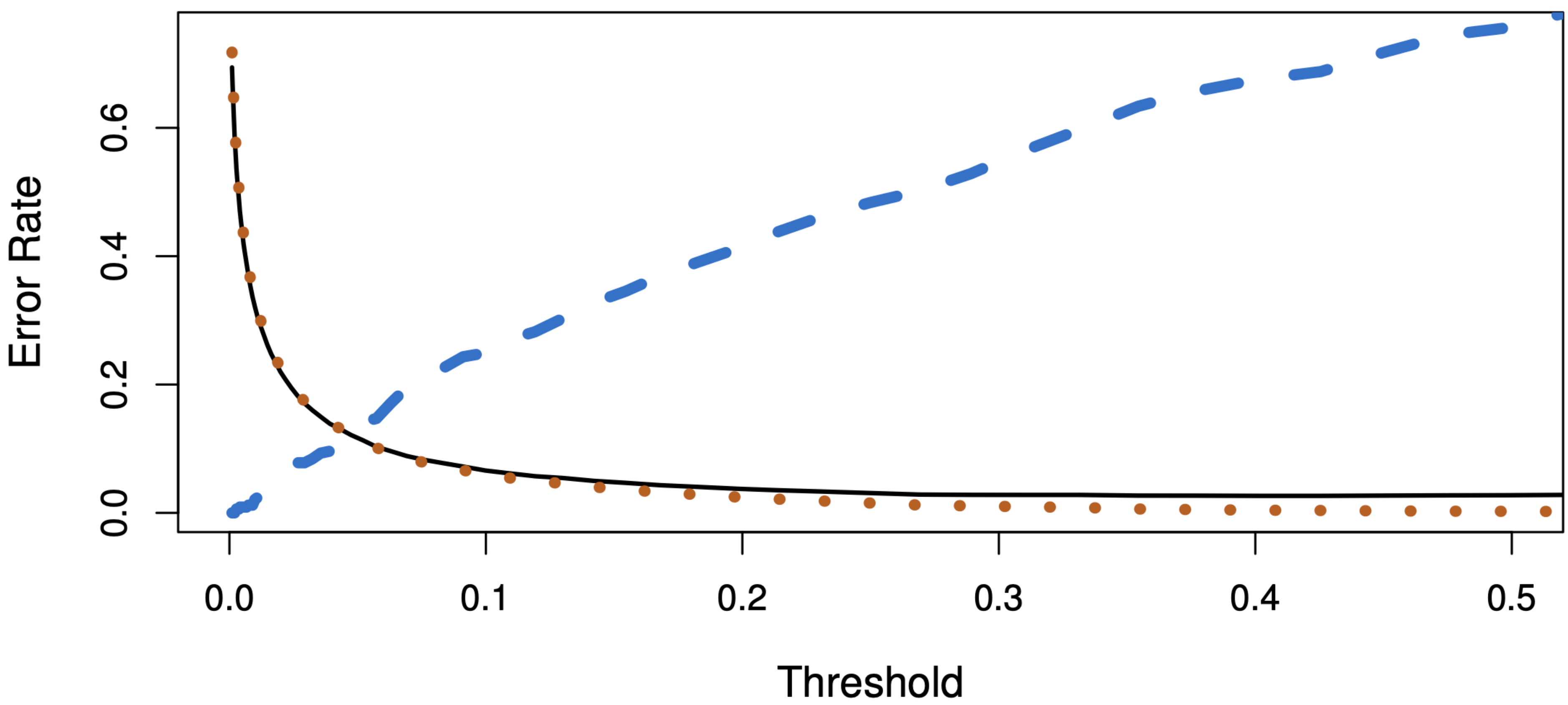
Changing decision rule

```
new.class = rep("No", length(Default$default))
new.class[lda.fit$posterior[,"Yes"] > 0.2] = "Yes"
table(new.class, Default$default)
```

```
##
## new.class   No  Yes
##       No  9432  138
##       Yes  235  195
```

- Predicted `Yes` if $P(\texttt{default} = \text{yes}|X) > 0.2$.
- Changing the threshold to 0.2 makes it easier to classify to `Yes`.
- Note that the rate of false positives became higher! That is the price to pay for fewer false negatives.

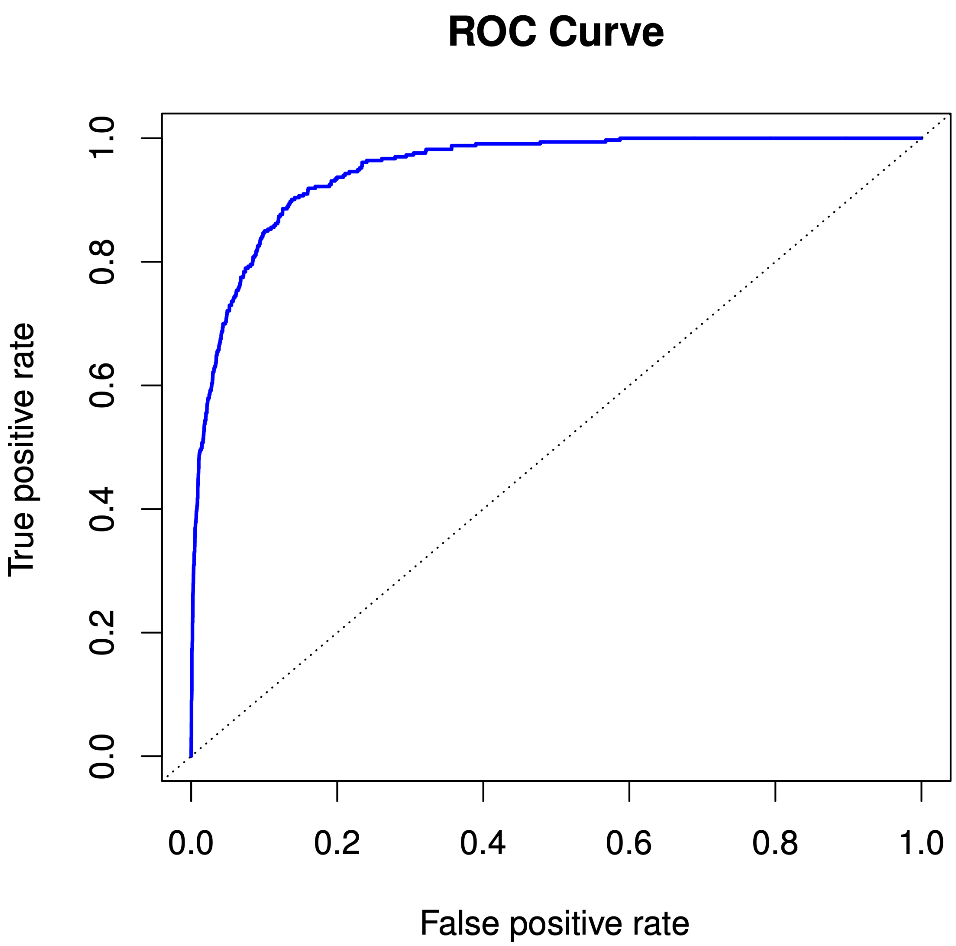
Let's visualize the dependence of the error on the threshold:



Error rates for LDA classifier on `Default` dataset

– – – – –– False negative rate (error for defaulting customers), · · · False positive rate (error for non-defaulting customers), – – – – – – – –– Overall error rate.

The ROC curve



ROC curve for LDA classifier on `Default` dataset.

- Displays the performance of the method for any choice of threshold.
- The area under the curve (AUC) measures the quality of the classifier:
    1. *0.5 is the AUC for a random classifier*

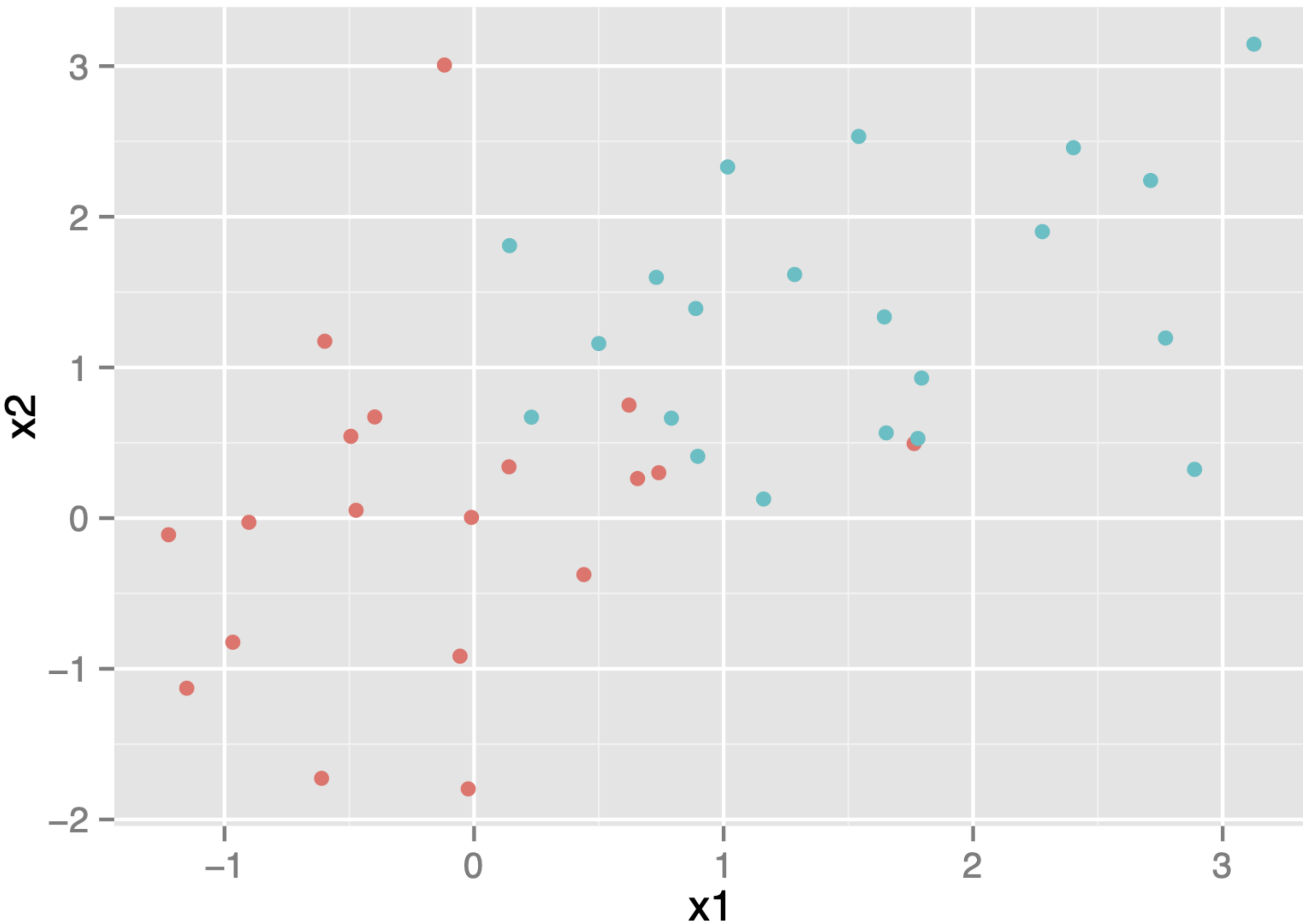
2. The closer the AUC is to 1, the better.

Comparing classification methods through simulation

- Simulate data from several different known distributions with 2 predictors and a binary response variable.
- Compare the test error (0-1 loss) for the following methods:

    1. *KNN-1*
    2. *KNN-CV ("optimally tuned" KNN)*
    3. *Logistic regression*
    4. *Linear discriminant analysis (LDA)*
    5. *Quadratic discriminant analysis (QDA)*

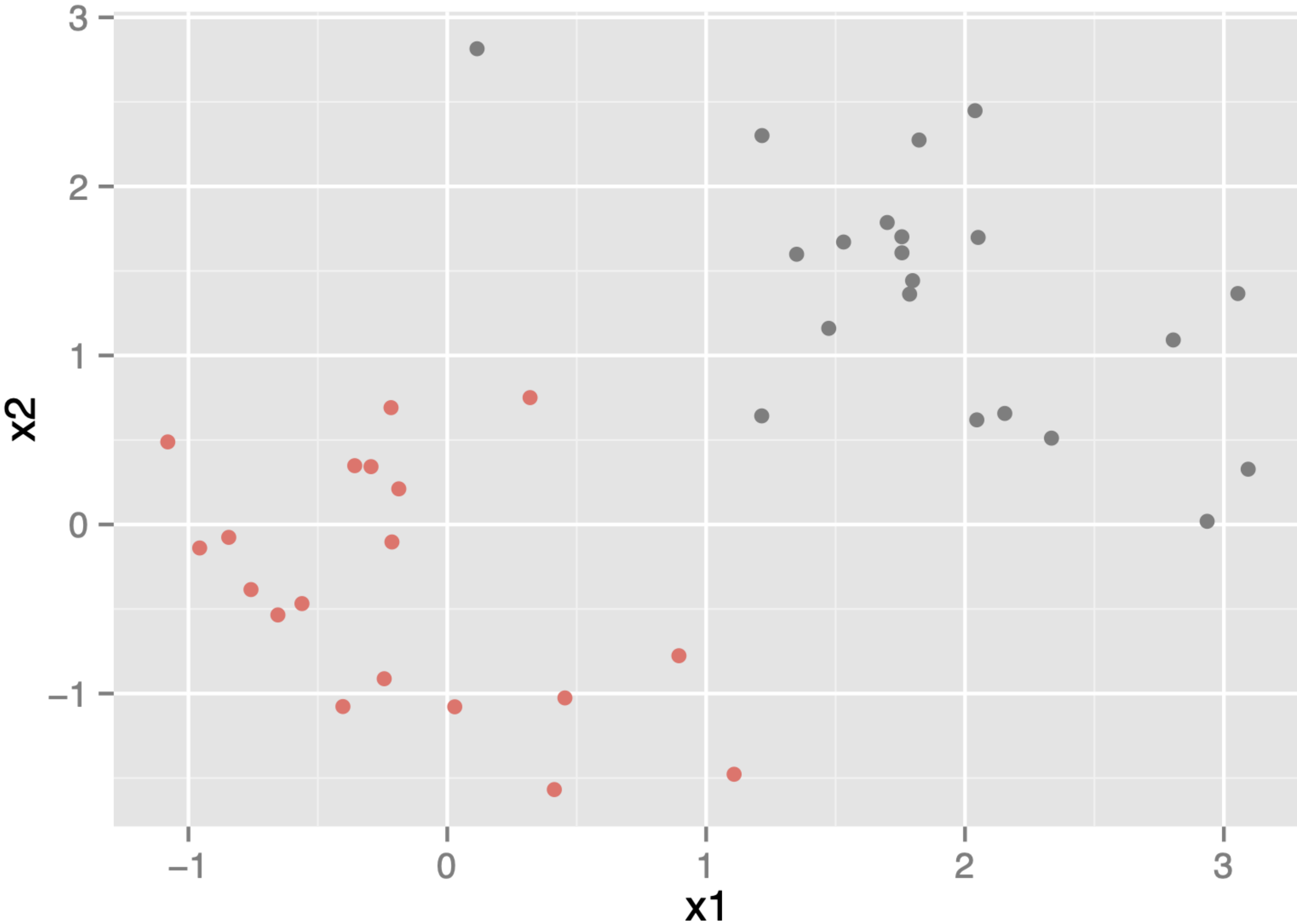
Scenario 1



Instance for simulation scenario #1.

- X_1, X_2 normal with identical variance.
- No correlation in either class.

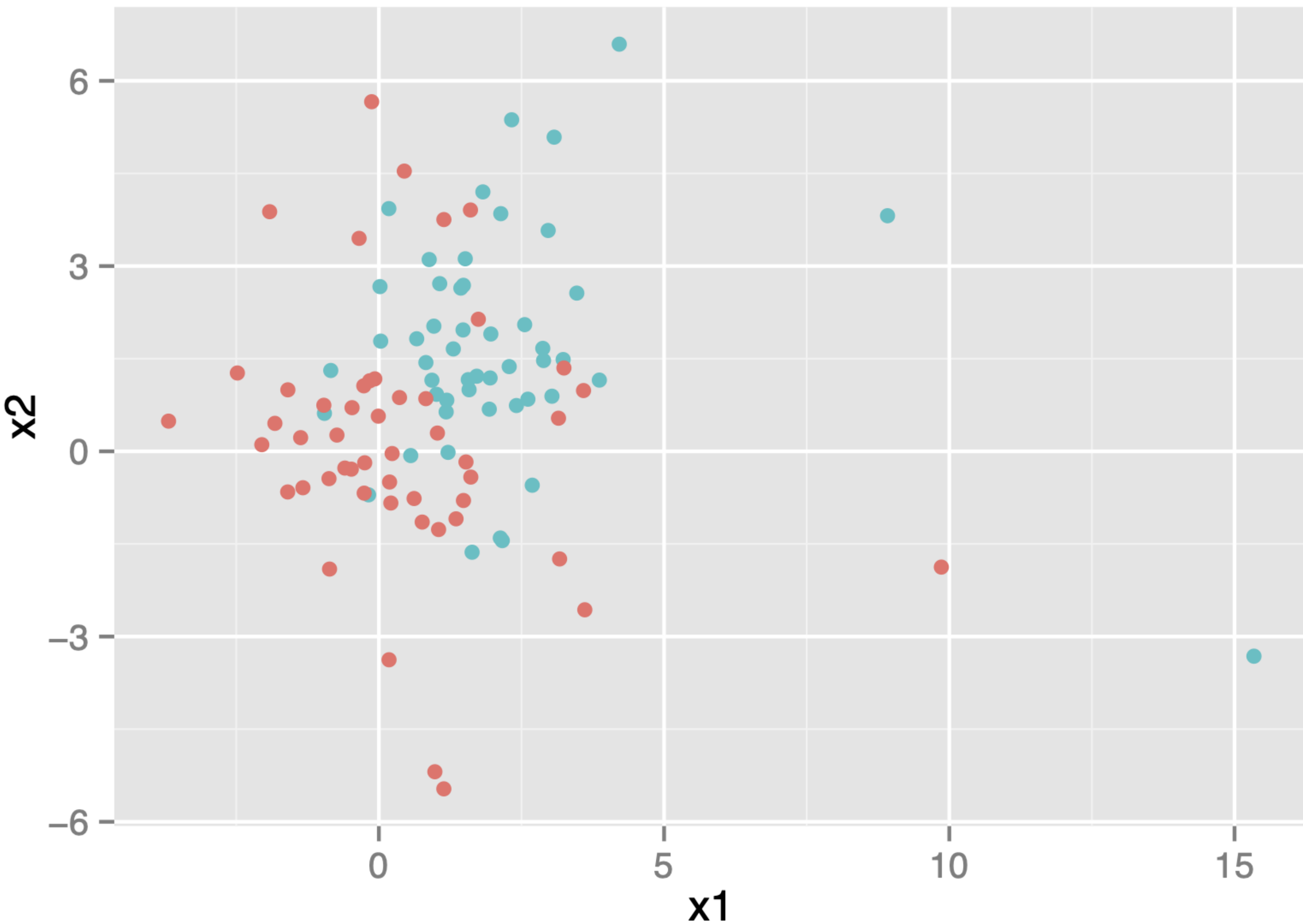
Scenario 2



Instance for simulation scenario #2.

- X_1, X_2 normal with identical variance.
- Correlation is -0.5 in both classes.

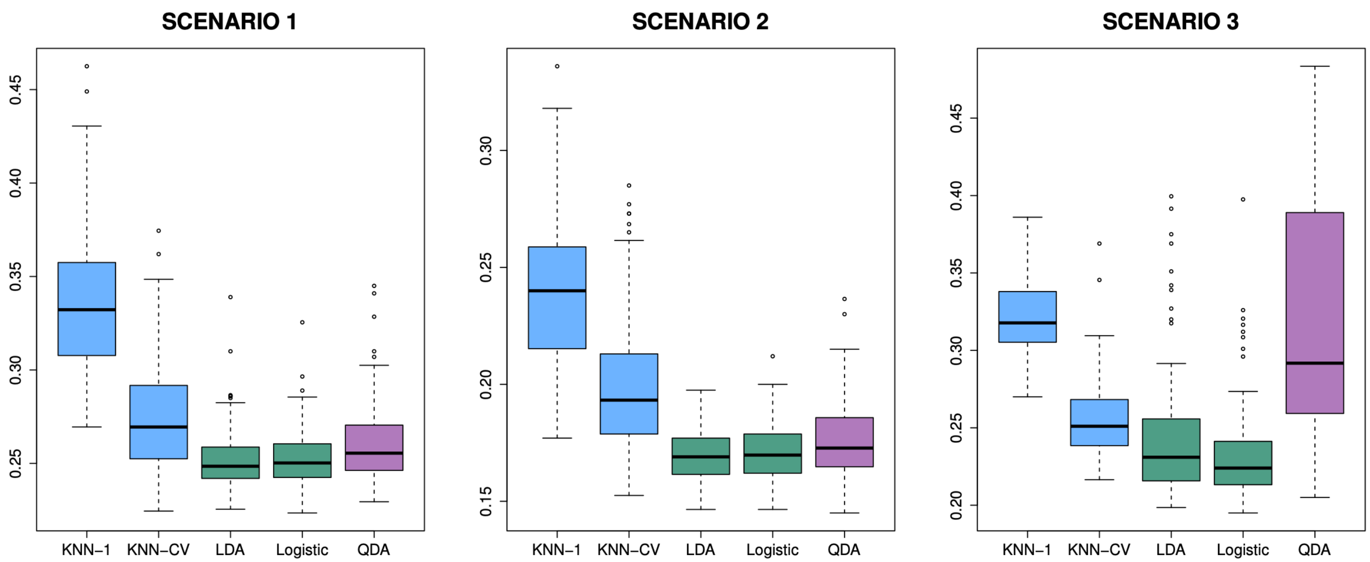
Scenario 3



Instance for simulation scenario #3.

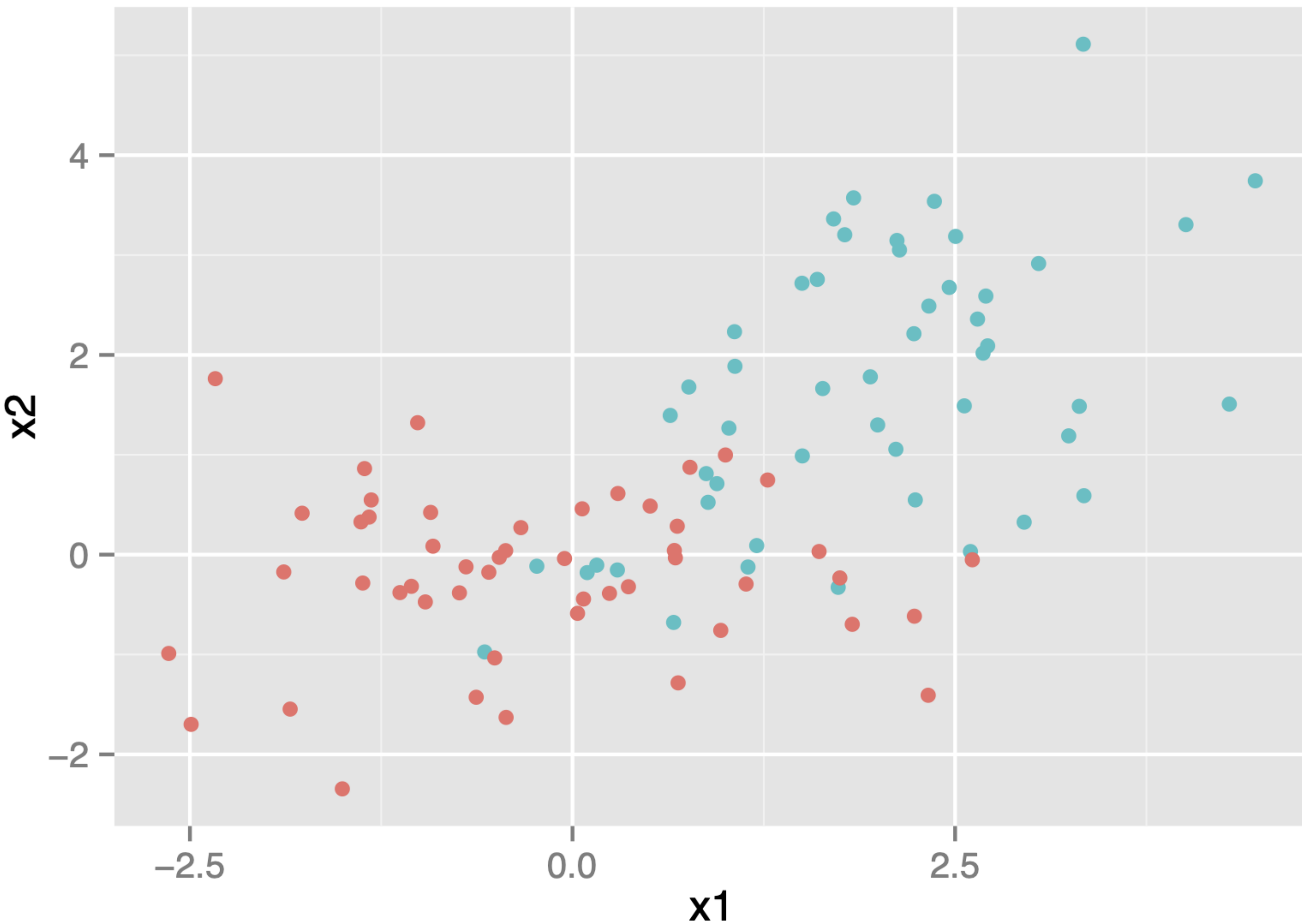
- X_1, X_2 student T.
- No correlation in either class.

Results for first 3 scenarios



Simulation results for linear scenarios #1-3.

Scenario 4



Instance for simulation scenario #4.

- X_1, X_2 normal with identical variance.
- First class has correlation 0.5, second class has correlation -0.5.

Scenario 5

- X_1, X_2 normal with identical variance.
- Response Y was sampled from:

$$P(Y = 1 \mid X) = \frac{e^{\beta_0 + \beta_1 X_1^2 + \beta_2 X_2^2 + \beta_3 X_1 X_2}}{1 + e^{\beta_0 + \beta_1 X_1^2 + \beta_2 X_2^2 + \beta_3 X_1 X_2}}.$$

- The true decision boundary is quadratic but this is not QDA model. (Why?)

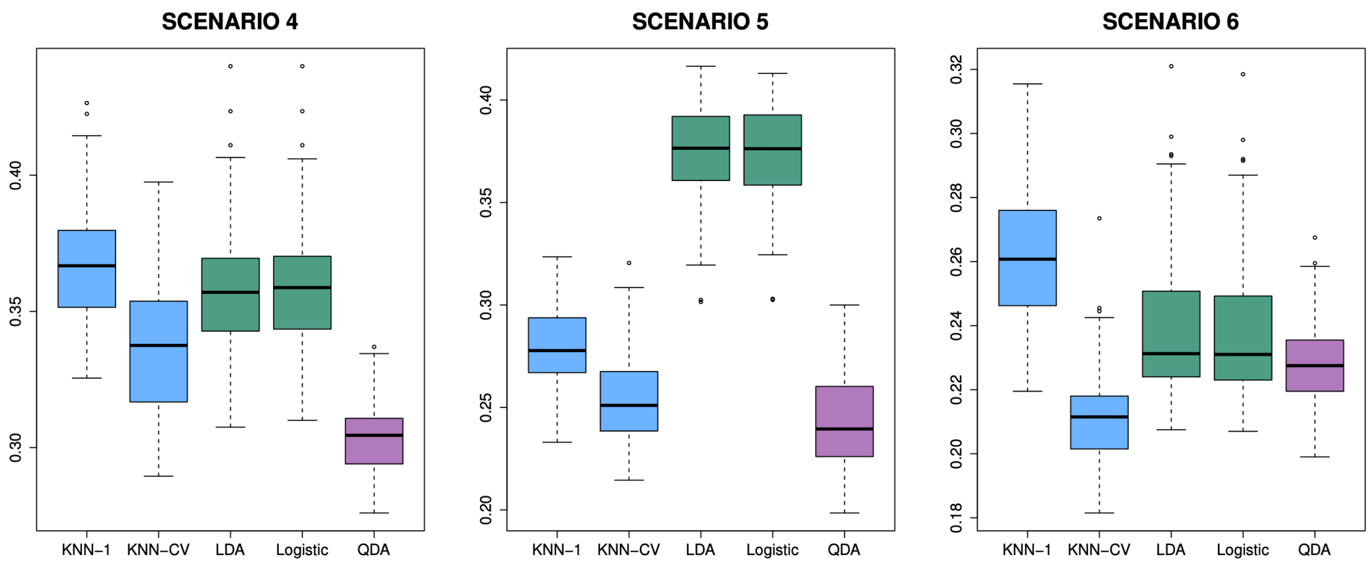
Scenario 6

- X_1, X_2 normal with identical variance.
- Response Y was sampled from:

$$P(Y = 1 \mid X) = \frac{e^{f_{\text{nonlinear}}(X_1, X_2)}}{1 + e^{f_{\text{nonlinear}}(X_1, X_2)}}.$$

- The true decision boundary is very rough.

Results for scenarios 4-6



Simulation results for nonlinear scenarios #4-6.