Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Business

MSc Applied Information and Data Science
Applied Machine Learning and Predictive Modelling 2

Classification

Prof. Dr. Fabio Sigrist
Institute of Financial Services Zug IFZ
Lucerne University of Applied Sciences and Arts

Autumn Semester 2020

Outline of this chapter

- Introduction
- Classification approaches
  - Logistic regression
  - Linear and quadratic discriminant analysis (LDA & QDA)
  - Naive Bayes
- Evaluating and comparing classifiers

- *Literature:*
  - *Chapters 4 and 5.1 of James et al. (2017)*

INTRODUCTION

Goal of classification

- **Given**: data for variables X and Y.
  - **Response variable $\boldsymbol{Y}$. $\boldsymbol{Y}$ is categorical: $\boldsymbol{Y} \in \{\mathbf{1}, \dots, \boldsymbol{k}\}$.** There are k classes / groups.
  - p predictor variables $X = \left(\mathrm{X}_1, \dots, X_p\right)^T$. Can be both quantitative or categorical.

- **Goal of classification**:
  - Predict to which group a new observation belongs (i.e., predict the class j for $X = x_{new}$).

- Often, the response variable Y is **binary**, i.e., it takes only two values.

Example use cases of classification

- **Example use cases:**
  - Churn prediction: will a customer churn or not?
  - Cross selling: will a customer buy a certain product or not?
  - Marketing: will a customer respond to a marketing action or not?
  - Credit risk modeling: will a company default or not?
  - Fraud detection: is a financial transaction fraudulent or not?
  - Is an e-mail spam or not?
  - Medicine: does a patient have a certain disease or not?

Example: Oscar winning movies[1]

Oscar	BoxOffice	Budget	Country	Critics	Length
0	20.91	21.73	Other	77.4	112
0	37.8	33.56	Europe	68.2	124
1	43.61	46.16	UK	38.5	108
1	53.53	18.67	Other	68.6	127
0	19.95	29.34	India	45.2	153
...	...	...	...	...	...

dependent variable y

predictor variables x

[1] Fictional data.

Example: Oscar winning movies

- **Dependent variable:** Oscar win (Y=1/N=0).

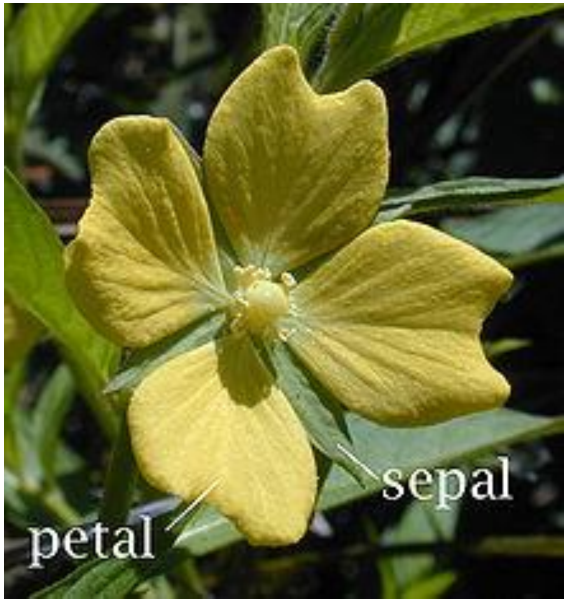
- **Predictor variables:**
  - Box office intake in millions of dollars.
  - Budget in millions of dollars.
  - Country of origin: US, UK, Europe, India, other.
  - Critical reception (average score 0-100).
  - Length of the movie in minutes.

Example: classification of Iris flowers

Iris setosa

Iris versicolor

Iris virginica



Goal: classify species based on data about sepal/petal length/width.

Example: Iris data

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5	3.4	1.5	0.2	setosa
6.8	3.2	5.9	2.3	virginica
6.8	2.8	4.8	1.4	versicolor
4.9	2.4	3.3	1	versicolor
5.1	3.3	1.7	0.5	setosa
4.6	3.4	1.4	0.3	setosa
…	…	…	…	…

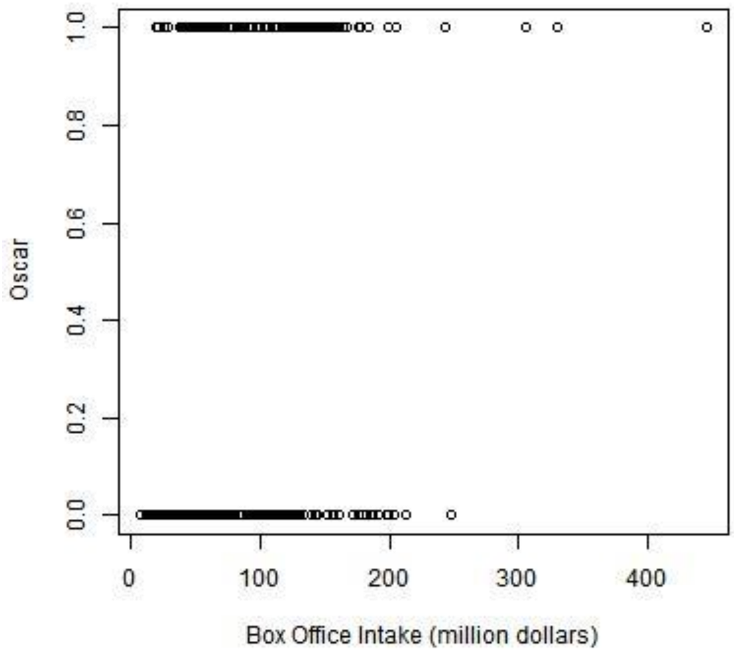
X Y

3 classes (species): setosa, versicolor, and virginica.

LOGISTIC REGRESSION

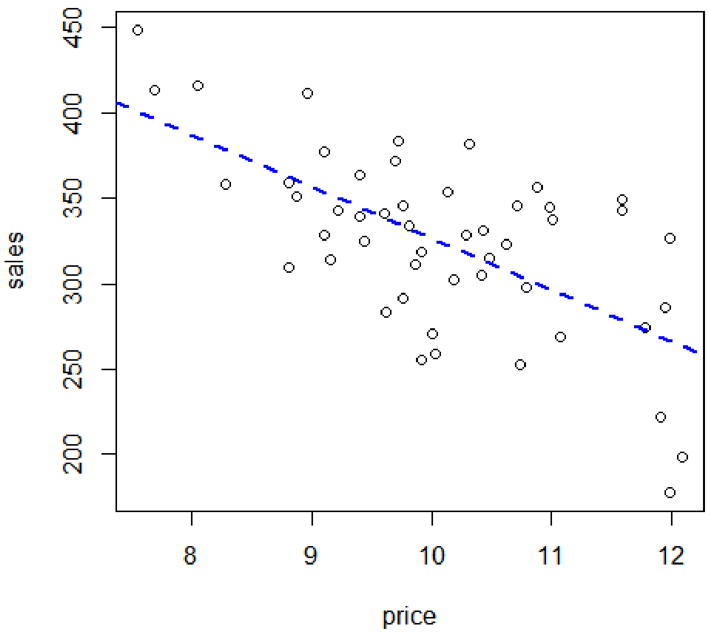
Example: Oscar winning movies

- At first, we only consider box office intake in millions of dollars as predictor variable.
- Will a movie with high a **box office intake** win the Oscar?



Recap: linear regression

- $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_K x_K + \epsilon$
  - y: dependent variable
  - $x_1, \dots, x_K$: predictor variables
  - $\beta_0, \beta_1, \dots, \beta_K$: coefficients
  - ϵ: random error

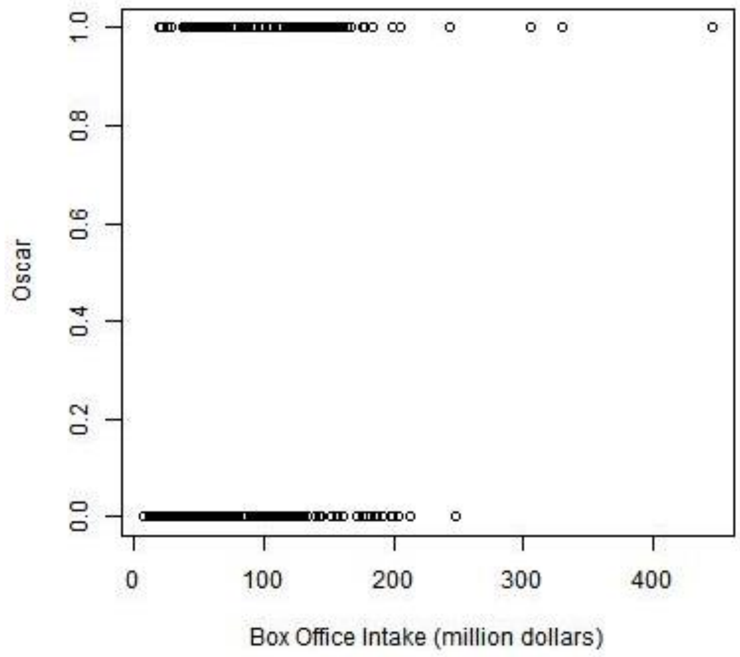


- In linear regression, the dependent variable y does **not only take two values but any number** on the real line.

- One **should not use linear regression** for modeling a binary variable.

Logistic regression

- How can we model binary data?



→ use a “two step” approach.

Logistic regression

1. As in linear regression, we start with
$$\eta(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p.$$
   - This is called the **linear predictor.**

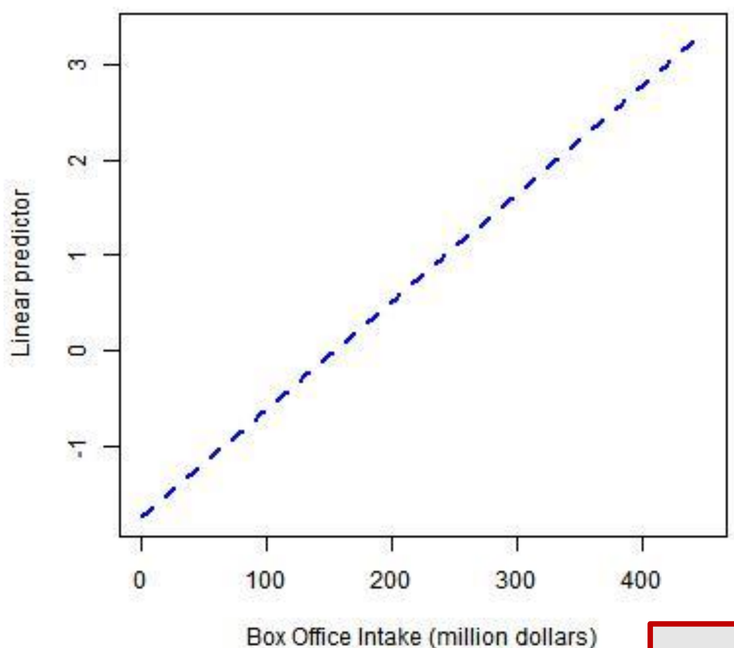
2. We then use a function to **transform** this, such that the resulting value is between 0 and 1:
$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}} \qquad \left(= \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}} \right)$$
   - This function is called the **logistic function**.

- This value $p(x)$ is interpreted as the **probability** that y equals one: $P(y = 1 | X = x)$.

Logistic regression

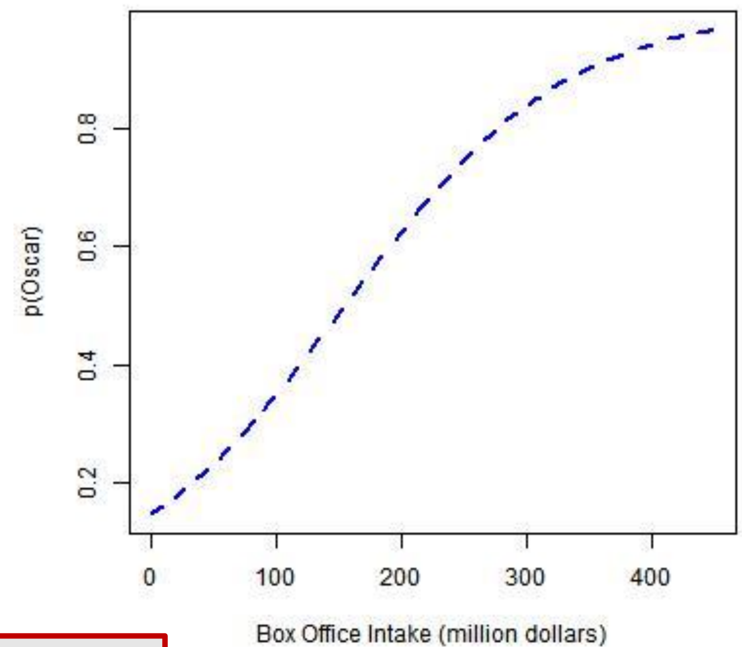
- $x_1, \dots, x_p$: predictor variables
- $\beta_0, \beta_1, \dots, \beta_p$: coefficients

linear predictor



transform to [0,1]

Probability of winning the Oscar

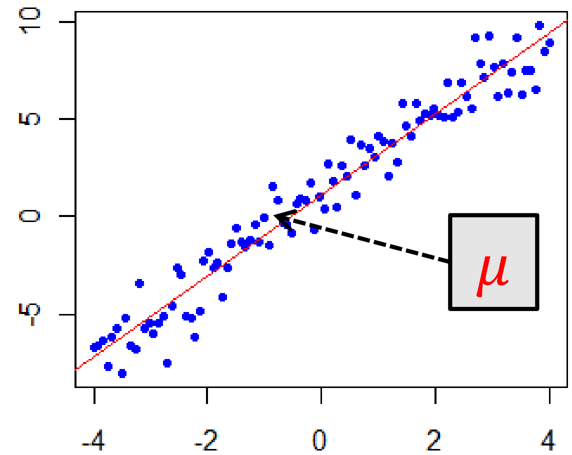


Comparison of linear and logistic regression

- **Linear regression:**

$$Y \sim N(\mu, \sigma^2)$$
$$\mu = \beta_0 + \beta_1 x_1$$

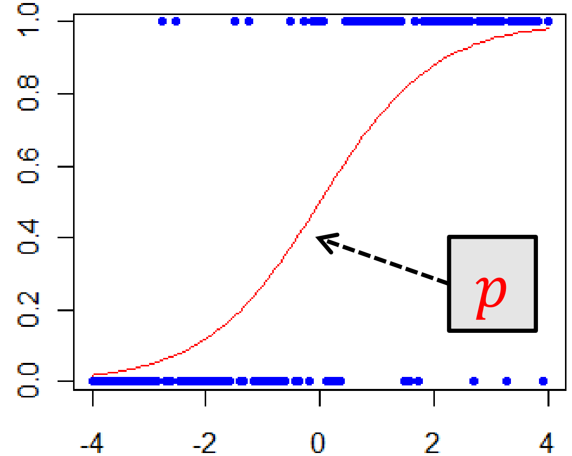
  - Example: distance and travel time in tram



- **Logistic regression:**

$$Y \sim \text{Bernoulli}(p)$$
$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1)}}$$

  - Example: failure probability and lifetime



Logistic regression in R

- Build the logistic regression model using the `glm` function:

```
boxOfficeModel <- glm(Oscar~BoxOffice,
                               family=binomial(link="logit"),
                               data=movieData)
```

- Comments:
  - `glm` stands for generalized linear model.

  - **Generalized linear models** are an extension of the linear regression model that allow for the dependent variable to have distributions other than the normal distribution.

  - In the case of the logistic regression model, this is the so called **binomial distribution**.

Logistic regression in R

`summary(boxOfficeModel)`

```
Call:
glm(formula = Oscar ~ BoxOffice, family = binomial(link = "logit"),
    data = moviedata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6432  -0.8316  -0.6997   1.2380   1.8546

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.750349   0.256883  -6.814 9.50e-12 ***
BoxOffice    0.011306   0.002507   4.510 6.48e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 374.60  on 299  degrees of freedom
Residual deviance: 350.82  on 298  degrees of freedom
AIC: 354.82

Number of Fisher Scoring iterations: 4
```

Learned / estimated coefficients

Significance of predictor variables

AIC: goodness of fit (lower = better)

Logistic regression in R

- The output is very similar to linear regression.

- The **interpretation** of the **magnitudes** of the coefficients is somewhat more complicated (no details here).

- The **interpretation** of the **signs** of the coefficients is the same as for linear regression models.

- The **interpretation** of the **p-values** is the same as for linear regression models.

Prediction with logistic regression

Assume we want to predict the probability that a movie with a $50 million box office intake wins the Oscar.

1. We first **calculate the linear predictor**

$$-1.75 + 0.011 \cdot 50 = -1.2$$

2. We then **transform** this to obtain a **probability**

$$p = \frac{1}{1 + e^{-(-1.2)}} = 0.231$$

- Thus, our model says **that the movie has a 23.1% chance of winning an Oscar**.
- If we have to make a **point prediction**, we would say that the movie **does not win** the Oscar, since 23.1%<50%.

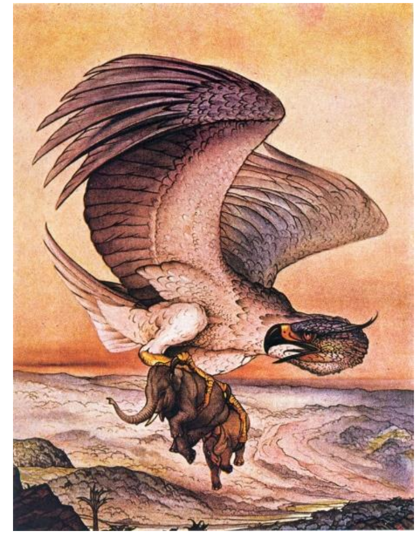
Kahoot question

Prediction in R

- In R, predictions are obtained as follows:

```
p=predict(boxOfficeModel,
              newdata=data.frame(BoxOffice=50),
              type = "response")
p # Probability of winning the Oscar
p>0.5 # Will the movie win the Oscar?
```

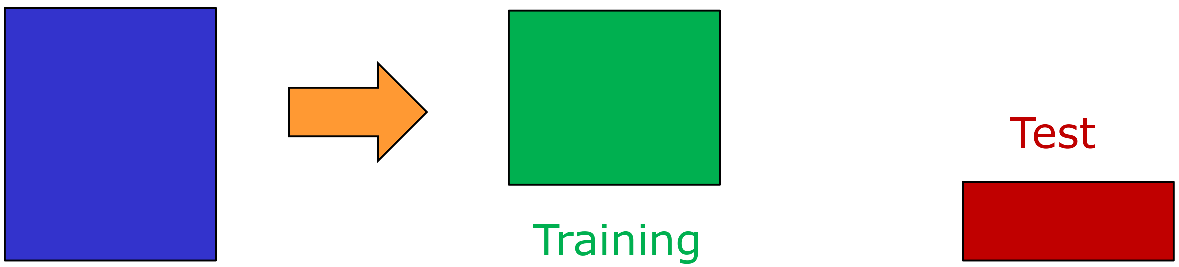
```
> p=predict(boxOfficeModel, newdata = data.frame(BoxOffice=50), type = "response")
> p
        1
0.2341426
> p>0.5
    1
FALSE
```



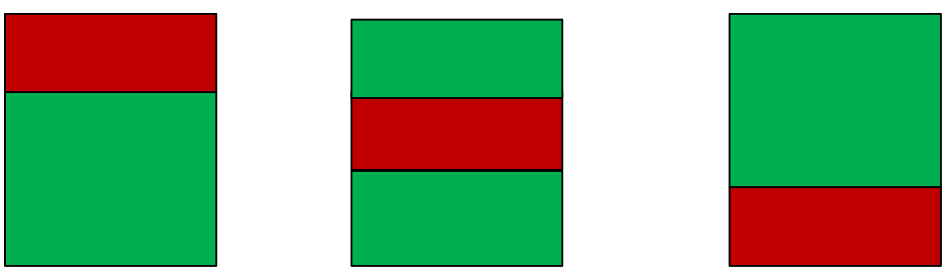
EVALUATION

Quality of classification

- **Problem**: if we use the same data for model fitting and evaluation, there is the danger of **overfitting**: too optimistic for error on new data.
- **Solution**: separate the data into training and test data.



- **Cross-validation (CV)**
  Example: “leave-one-out” cross-validation. Every row / observation is the test case once, the rest in the training data.



Confusion matrix

- **Confusion matrix** (e.g. 300 movies):

	Truth = 0	Truth = 1
Prediction = 0	190	80
Prediction = 1	15	15

Wrongly classified.

Correctly classified.

All movies that **did not win** an Oscar.

All movies that **did win** an Oscar.

- **Error rate**:
  (80+15)/300=0.32
  (wrongly classified) / (number of samples)
- We expect that our classifier predicts 32% of new observations *incorrectly.*

$$\text{Error rate} = \frac{\text{FN+FP}}{\text{TN+FN+FP+TP}}$$

Example confusion matrix for more than 2 categories

- Confusion matrix (e.g. 100 samples):

	Truth=0	Truth=1	Truth=2
Pred = 0	23	7	6
Pred = 1	3	27	4
Pred = 2	3	1	26

- **Error (misclassification) rate**:
  1 – sum(diagonal entries) / (number of samples) =
  = 1 – 76/100 = 0.24.
- We expect that our classifier predicts approx. 24% of new observations incorrectly.

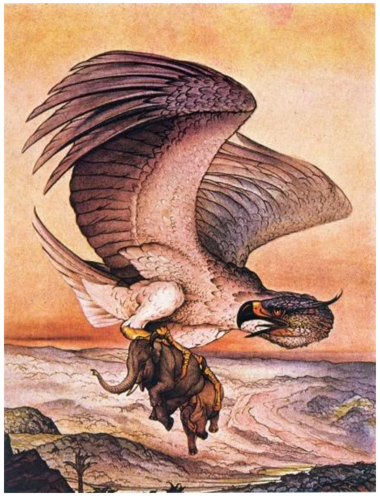
Comparing binary classifiers

- Binary classifiers: we usually predict $Y = 1$ if $\hat{P}(Y = 1|X) > \delta$ where $\delta = 0.5$
- The threshold $\delta = 0.5$ can be arbitrary and does not always produce the best results
- To avoid choosing one single threshold, one can **compare classifiers for various choices of thresholds**
  → choose classifier which is best for "many thresholds".

Comparing binary classifiers using the ROC curve

- Recall confusion matrix for binary classification

	Truth = 0	Truth = 1
Pred = 0	True negative (TN)	False negative (FN)
Pred = 1	False positive (FP)	True positive (TP)



https://en.wikipedia.org/wiki/Roc_(mythology)

- **Receiver operating characteristic (ROC)** plots
  - true positive (TP) rate vs.
  - false positive (FP) rate

  for various thresholds.

The higher the better

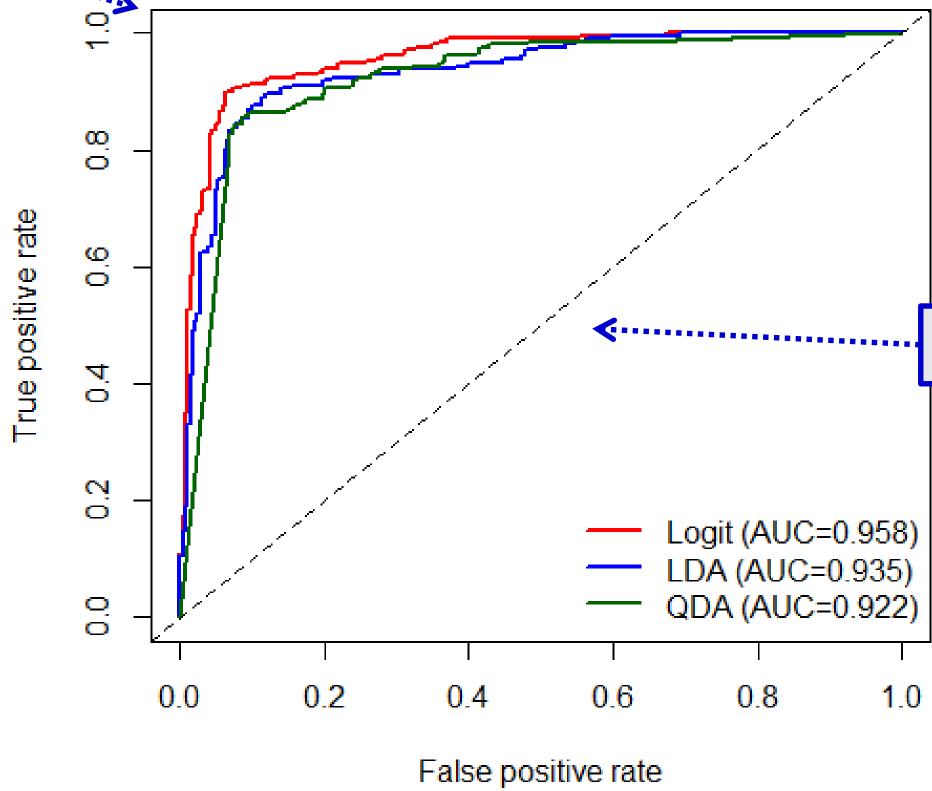
The lower the better

- Summary measure**: area under the receiver operating characteristic (AUC)**

The higher the better

Example: SPAM detection

The closer to this corner the better



Random guessing

See R examples.

LDA & QDA

Classification

- Most approaches for classification calculate an estimate for the **probability**
$$\boldsymbol{P(Y = j|X = x)}.$$

- In general, $X = x$ is then classified into the group j for which this probability is highest.

- There are two different approaches to obtain $P(Y = j|X = x)$:
  - Direct modelling of $P(Y = j|X = x)$ (e.g., logistic regression).
  - First model $P(X = x|Y = j)$ and then use Bayes' theorem to obtain $P(Y = j|X = x)$ (LDA & QDA).

Idea of LDA and QDA

- Both **linear and quadratic discriminant analysis** (LDA & QDA) start by specifying:
  1. The **prior probability** $\boldsymbol{p_j = P(Y = j)}$ that an observation belongs to class j.
  2. The **distribution of $\boldsymbol{X}$ given that an observation belongs to class j**. This is assumed to be a multivariate normal distribution $\boldsymbol{X|Y = j \sim N(\mu_j, \Sigma_j)}$.

- Bayes' theorem is then used to calculate **posterior probability** $\boldsymbol{P(Y = j|X = x)}$.
  - Bayes' theorem:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}.$$

For LDA and QDA, we apply this with A replaced by "X=x" and B replaced by "Y=j".

LDA & QDA

- It follows that the (unconditional) **distribution of X** is a **Gaussian mixture** with density

$$\sum_{j=1}^{k} p_j g_j(x; \theta_j),$$

Note: Observe the **similarity to model based clustering.** In contrast to clustering, we know the number of groups k and, in particular, to which group an observation in the data belongs to.

where

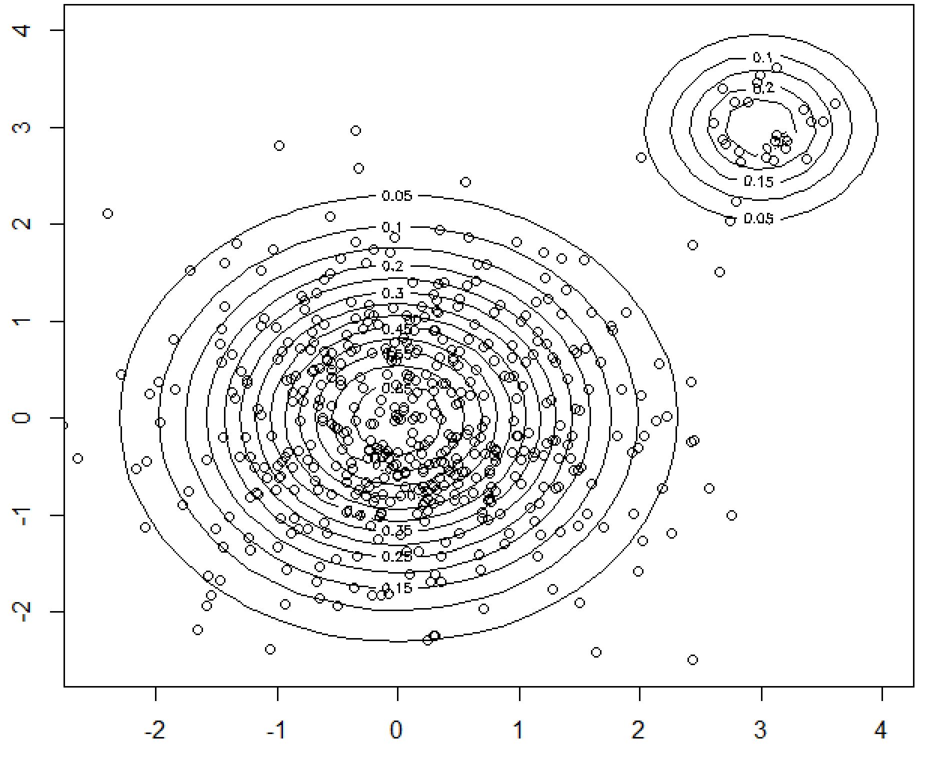
$$g_j(x; \theta_j) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j)\right)$$

is the multivariate normal density and

$$\sum_{j=1}^{k} p_j = 1.$$

Gaussian mixture model example

- 2D example



LDA & QDA

- By applying Bayes' theorem, one obtains the so-called **posterior probability that an observation belongs to class j given $X = x$**

$$P(Y = j|X = x) = \frac{p_j g_j(x;\theta_j)}{\sum_{j'=1}^{k} p_{j'} g_{j'}(x;\theta_{j'})}.$$

- $X = x$ is classified into the class j for which this probability is maximal.

- How can we find this class j?
  - Use the fact that
    - $\underset{j}{\operatorname{argmax}}\, P(Y = j|X = x) = \underset{j}{\operatorname{argmax}} \log\big(P(Y = j|X = x)\big)$
    - $\log\big(P(Y = j|X = x)\big) = \log(p_j) + \log\Big(g_j(x;\theta_j)\Big) - \log\big(\sum_{j'=1}^{k} p_{j'} g_{j'}(x;\theta_{j'})\big)$

The last term is the same for all j. So we can drop it for finding the maximum.

Quadratic discriminant analysis (QDA)

- **Quadratic discriminant analysis (QDA)** assigns an observation $X = x$ to the class j for which

$$\delta_j(x) = \log\left(p_j\right) - \frac{1}{2}\log(|\Sigma_j|) - \frac{1}{2}\left(x-\mu_j\right)^T \Sigma_j^{-1}(x-\mu_j)$$

is maximal.

This is obtained by plugging in the Gaussian density for $g_j\left(x;\theta_j\right)$ (and dropping the term $-\frac{p}{2}\log(2\pi)$).

- It is called "quadratic", since x appears as a quadratic function in $\delta_j(x)$.

- The x's for which $\delta_j(x) = \delta_{j'}(x)$ are called **decision boundaries.**

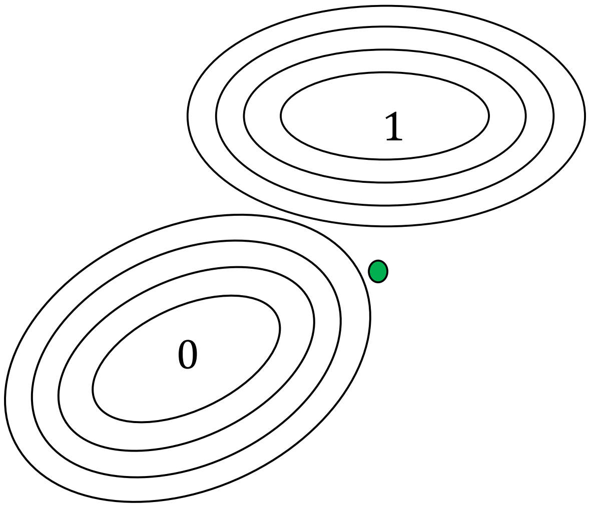
Parameter estimation

- The **parameters** $\mathrm{p_j}, \mu_j$, and Σ_j are **estimated** as follows:
  - p_j: fraction of observations in the data that belong to class j.
  - μ_j: sample mean $\bar{x}_j$ of all observations that belong to class j.
  - Σ_j: sample covariance S_j of all observations that belong to class j.

Intuition for QDA

- $\delta_j(x) = \log(p_j) - \frac{1}{2}\log(|\Sigma_j|) - \frac{1}{2}(x - \mu_j)^T \Sigma_j^{-1}(x - \mu_j)$

- Example:



Classify to which class
(assuming equal p_j and $|\Sigma_j|$)?

Kahoot question

Linear discriminant analysis (LDA)

- Depending on the number of variables p, $X = \left(\mathrm{X}_1, \dots, X_p\right)$, this can lead to a large number of parameters. In particular, the covariance matrices Σ_j can contain a lot of parameters.

- In **linear discriminant analysis (LDA),** we assume that all the **covariance matrices** are **equal** for all classes. I.e., we assume

$$X|Y = j \sim N(\mu_j, \Sigma)$$

instead of

$$X|Y = j \sim N(\mu_j, \Sigma_j).$$

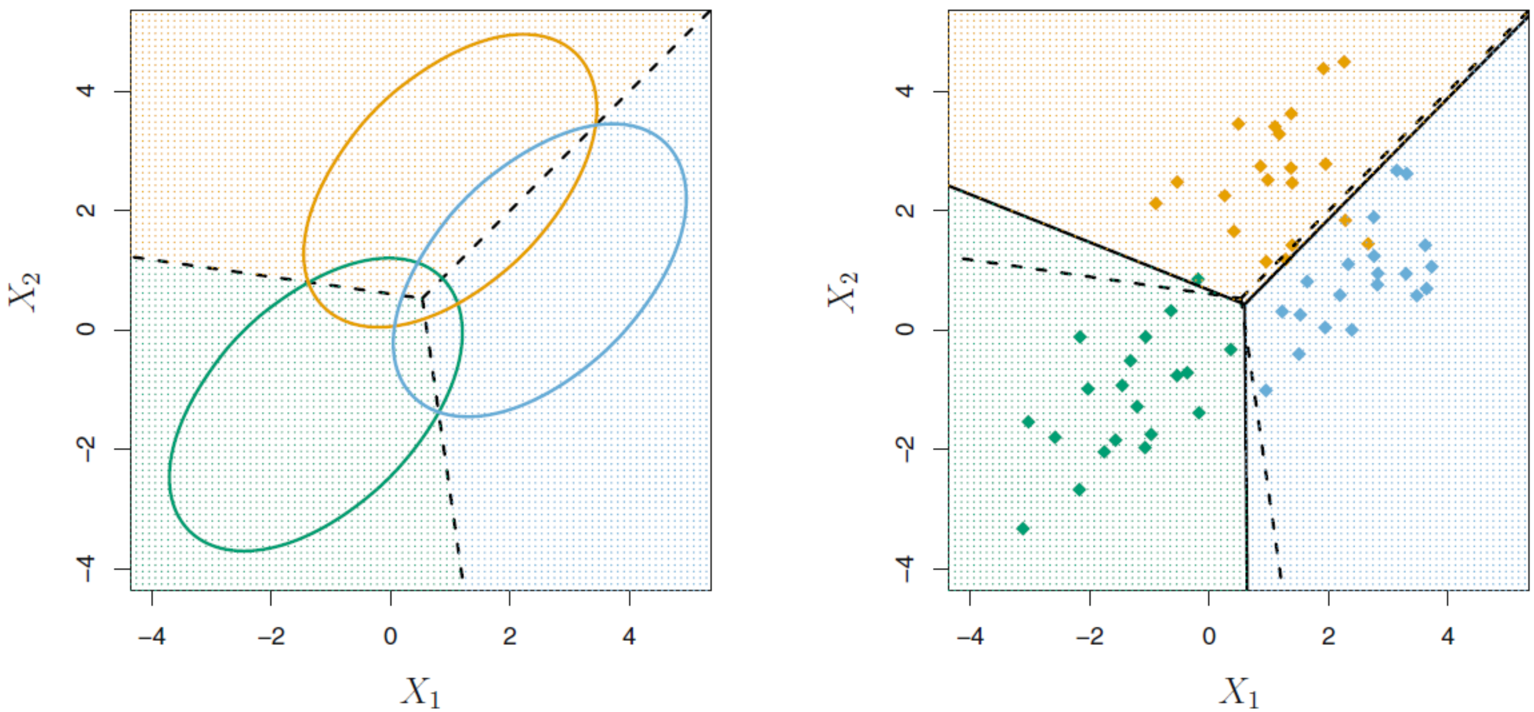
Linear discriminant analysis

- It follows that linear discriminant analysis assigns an observation $X = x$ to the class j for which
$$\delta_j(x) = \log(p_j) + x^T \Sigma^{-1} \mu_j - \frac{1}{2} \mu_j^T \Sigma^{-1} \mu_j$$
is maximal.

  - The function $\delta_j(x)$ is linear in x.
  - In contrast to QDA, the quadratic term $-\frac{1}{2} x^T \Sigma^{-1} x$ has been dropped since it does not depend on j.

Illustration of decision boundaries for LDA

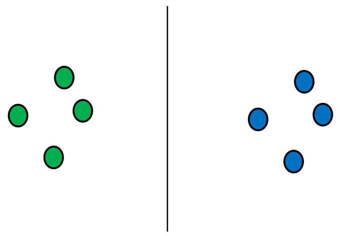
- LDA example with three classes ($k = 3$) and two variables ($p = 2$). The dashed lines are the true decision boundaries ($\delta_j(x) = \delta_{j'}(x)$) and the solid lines the estimated ones.

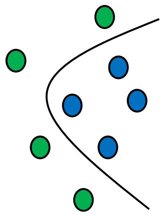


Source: James et al. (2013)

LDA vs. QDA

LDA	QDA
+ Only few parameters to estimate;	- Many parameters to estimate; potentially less accurate estimates
- Less flexible (linear decision boundary)	+ More flexible (quadratic decision boundary)





Naïve Bayes and QDA

- **Naïve Bayes** is an often used technique in applied machine learning.

- Gaussian naïve Bayes is a special case of QDA:
  - Instead of $X|Y = j \sim N(\mu_j, \Sigma_j)$ with Σ_j being a general covariance matrix, it is assumed that Σ_j is diagonal
$$\Sigma_j = \mathrm{diag}\left(\sigma_{1j}^2, \dots, \sigma_{pj}^2\right).$$

- In general, naïve Bayes assumes that conditional on $Y = j$, the $\left(X_1, \dots, X_p\right)$'s are independent.


Thomas Bayes
1702 - 1761

Comparing logistic regression and LDA

- For LDA, we have

$$\log\left(\frac{P(Y=1|X=x)}{P(Y=0|X=x)}\right) =$$

$$= \underbrace{\log\left(\frac{p_0}{p_1}\right) - \frac{1}{2}(\mu_0+\mu_1)^T\Sigma^{-1}(\mu_1-\mu_0)}_{\alpha_0} + x^T \underbrace{\Sigma^{-1}(\mu_1-\mu_0)}_{\alpha}$$

$$= \alpha_0 + x^T\alpha.$$

- For logistic regression, we have

$$\log\left(\frac{P(Y=1|X=x)}{P(Y=0|X=x)}\right) = \beta_0 + x^T\beta.$$

- Logistic regression is thus based on less assumptions and directly finds the “best” β_0 and β → more flexible and often better