QBUS6810: Statistical Learning and Data Mining

Lecture 9: Classification II

Semester 1, 2019

Discipline of Business Analytics, The University of Sydney Business School

Lecture 9: Classification II

1. Model evaluation for binary classification
2. Logistic regression
3. Gaussian discriminant analysis
4. Empirical example
5. Comparison of classification methods

Model evaluation for binary classification

Confusion matrix

A **confusion matrix** counts the number of true negatives, false positives, false negatives, and true positives for the test data.

		Classification		
		$\widehat{y} = 0$	$\widehat{y} = 1$	Total
Actual	$Y = 0$	True negatives (TN)	False positives (FP)	N
	$Y = 1$	False negatives (FN)	True positives (TP)	P
	Total	Negative predictions	Positive predictions	

True positive and true negative rates

The **True Positive Rate** (a.k.a. *sensitivity* or *recall*) is:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{True positives}}{\text{Actual positives}} \approx P(\widehat{y} = 1 | Y = 1)$$

The **True Negative Rate** (a.k.a. *specificity*) is:

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{True negatives}}{\text{Actual negatives}} \approx P(\widehat{y} = 0 | Y = 0)$$

False positive and false negative rates

The **False Positive Rate** (FPR) is

$$\frac{\mathsf{FP}}{\mathsf{TN}+\mathsf{FP}} = \frac{\text{False positives}}{\text{Actual negatives}} = 1-\mathsf{TNR} \approx P(\widehat{y}=1|Y=0)$$

The **False Negative Rate** (FNR) is

$$\frac{\mathsf{FN}}{\mathsf{TP}+\mathsf{FN}} = \frac{\text{False negatives}}{\text{Actual positives}} = 1-\mathsf{TPR} \approx P(\widehat{y}=0|Y=1)$$

Decision rule

Recall that the decision to classify a subject as positive or negative is based on the following decision rule:

$$\widehat{y}(\boldsymbol{x}) = \begin{cases} 1 \text{ if } \widehat{P}(Y = 1 | X = \boldsymbol{x}) > \tau. \\ 0 \text{ if } \widehat{P}(Y = 1 | X = \boldsymbol{x}) \leq \tau. \end{cases}$$

Trade-off between true positive and true negative rates

- There is a trade-off between the true positive and true negative rates, since a classifier can always obtain the maximum true positive (negative) rate by setting $\tau = 0$ ($\tau = 1$) and automatically returning all positives (negatives).

- Equivalently, there is a trade-off between achieving a higher true positive rate and achieving a lower false positive rate.

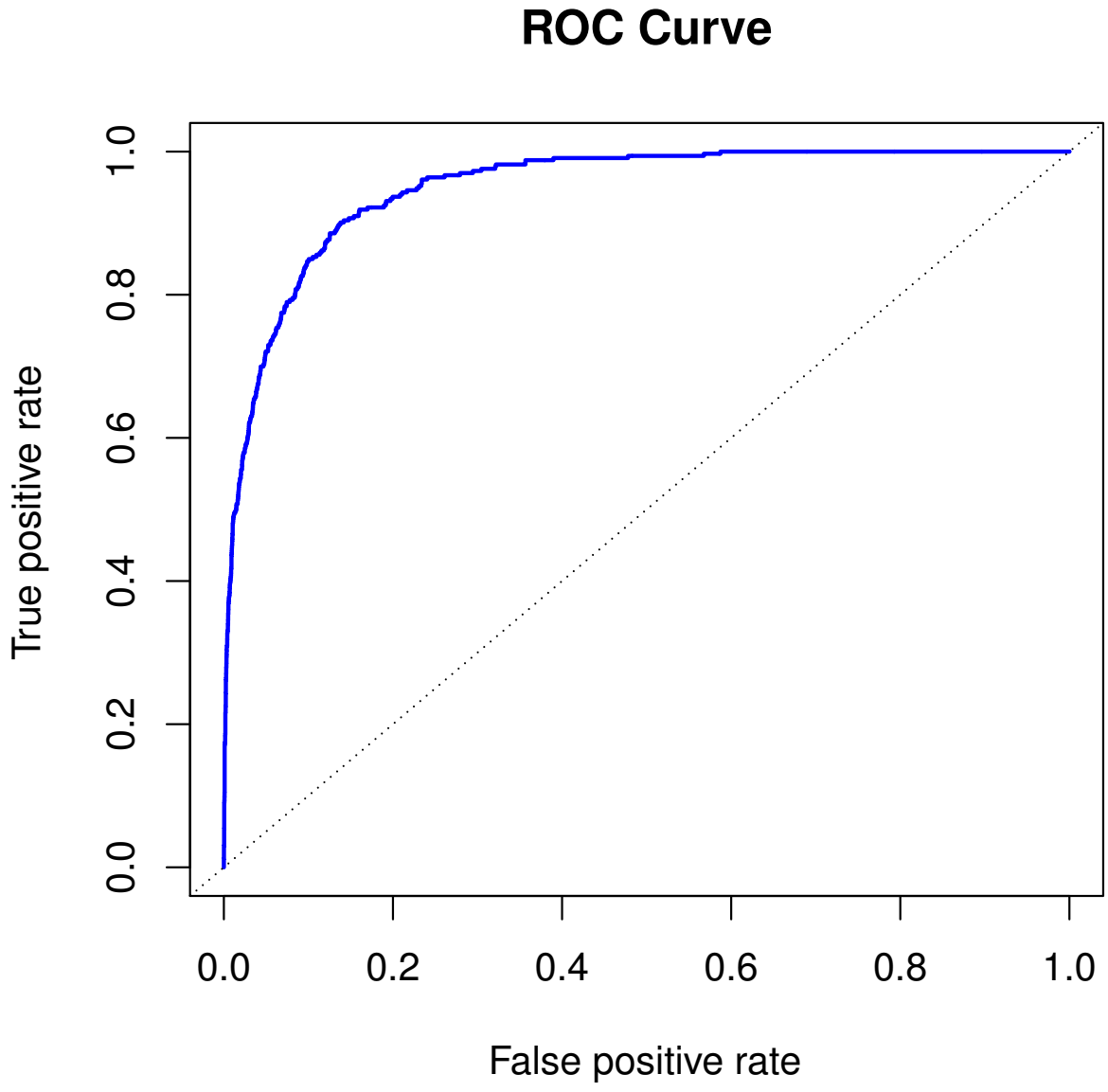
ROC curve

A **receiver operating characteristic** or **ROC** curve plots the true positive rate against the false positive rate for a range of threshold values τ.

ROC plots tell us the false positive rate that we need to accept if we want to obtain a particular level of the true positive rate.

We often summarise the quality of ROC curve as a single number using the **area under the curve** or **AUC**. Higher AUC scores are better, and the highest possible AUC value is one.

ROC curve


ROC Curve
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

Imbalanced classes

Many classification scenarios (such as fraud detection) concern rare events, leading to a very large proportion of negatives in the data.

In this situation we say that the classes are highly **imbalanced**.

TNR and FPR are not very informative for these problems, as it TNR will tend to be high (thus, FPR will be low) regardless of the quality of the classifier.

Precision

In the imbalanced scenario, we are usually more interested in the proportion of detections that are actually positive. We define the **precision** as

$$\frac{\mathsf{TP}}{\mathsf{TP}+\mathsf{FP}} = \frac{\text{True positives}}{\text{Positive classifications}} \approx P(Y=1|\widehat{y}=1)$$

Business application: customer churn

We now move on to the next topic of logistic regression.

First, consider the following application example.

Business application: customer churn

Customer churn (or attrition) occurs when a customer leaves the current service provider and switches to a different one.

- What are the drivers of customer attrition?
- When is a current customer likely to end the relationship?
- What strategies improve the retention of profitable customers?
- Is it more profitable to invest additional resources into customer acquisition or retention?

Customer attrition can be very costly for companies that do not have good answers to these questions.

Issues addressed in customer retention modelling

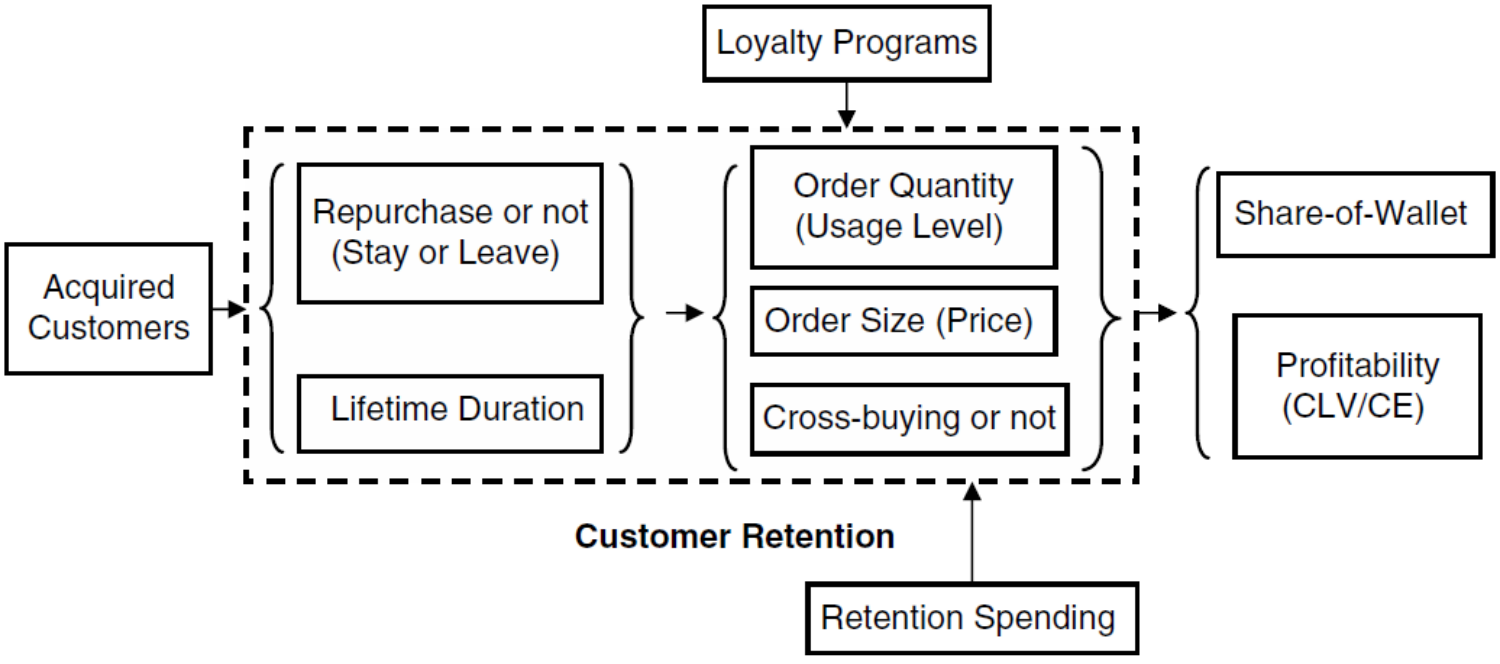


Figure from Kumar and Petersen, 2012.

Balancing acquisition and retention

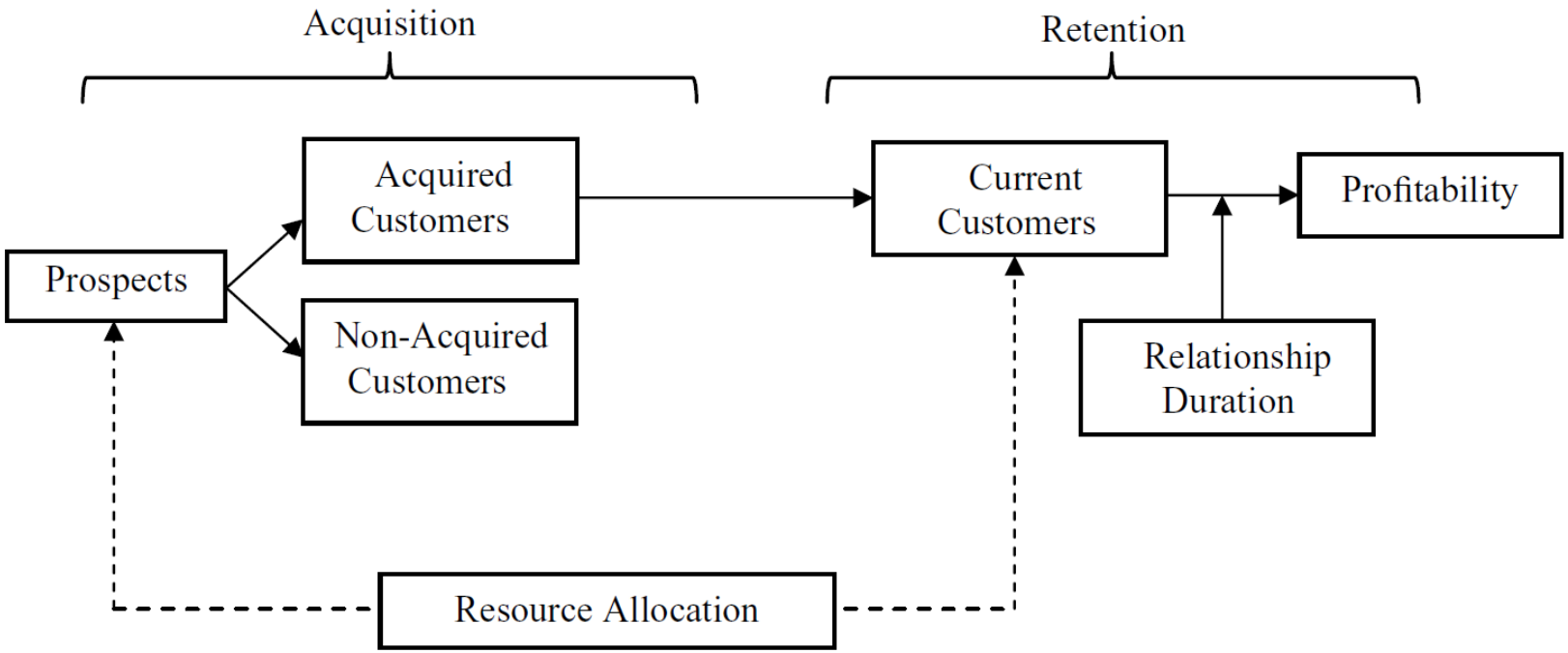


Figure from Kumar and Petersen, 2012.

Customer churn data

Response: whether the customer had churned by the end of the observation period.

Predictors

1. Average number of dollars spent on marketing efforts to try and retain the customer per month.
2. Total number of categories the customer has purchased from.
3. Number of purchase occasions.
4. Industry: 1 if the prospect is in the B2B industry, 0 otherwise.
5. Revenue: annual revenue of the prospect's firm.
6. Employees: number of employees in the prospect's firm.

Observations: 500.

Source: Kumar and Petersen (2012).

Logistic regression

Regression models for classification

Suppose that we want to specify a model for binary classification. The response Y follows the Bernoulli distribution:

$$Y = \begin{cases} 1 \text{ with probability } p(\boldsymbol{x}) = P(Y = 1 | X = \boldsymbol{x}) \\ 0 \text{ with probability } 1 - p(\boldsymbol{x}) \end{cases}$$

How can we model the conditional probability $P(Y = 1 | X = \boldsymbol{x})$ as a function of the predictors?

Regression models for classification

Since $P(Y = 1|X = \boldsymbol{x}) = E(Y|X = \boldsymbol{x})$, one option is to specify a linear regression model

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \varepsilon.$$

This is called the **linear probability model**. However, there are several reasons why this framework is unsatisfactory.

Why not the linear probability model?

1. There is no guarantee that the linear probability model will generate probabilities between zero and one, since the regression function $\beta_0 + \sum_{j=1}^{p} \beta_j x_j$ is unconstrained.

2. The Bernoulli distribution has variance $p(\boldsymbol{x})(1 - p(\boldsymbol{x}))$. Hence, the linear probability model violates the classical assumption of constant error variance.

3. The linear probability approach does not easily generalise to categorical responses with more than two classes.

Logistic regression

The **logistic regression model** is

$$Y|X = \boldsymbol{x} \;\sim\; \text{Bernoulli}\Big(p(\boldsymbol{x})\Big)$$

where

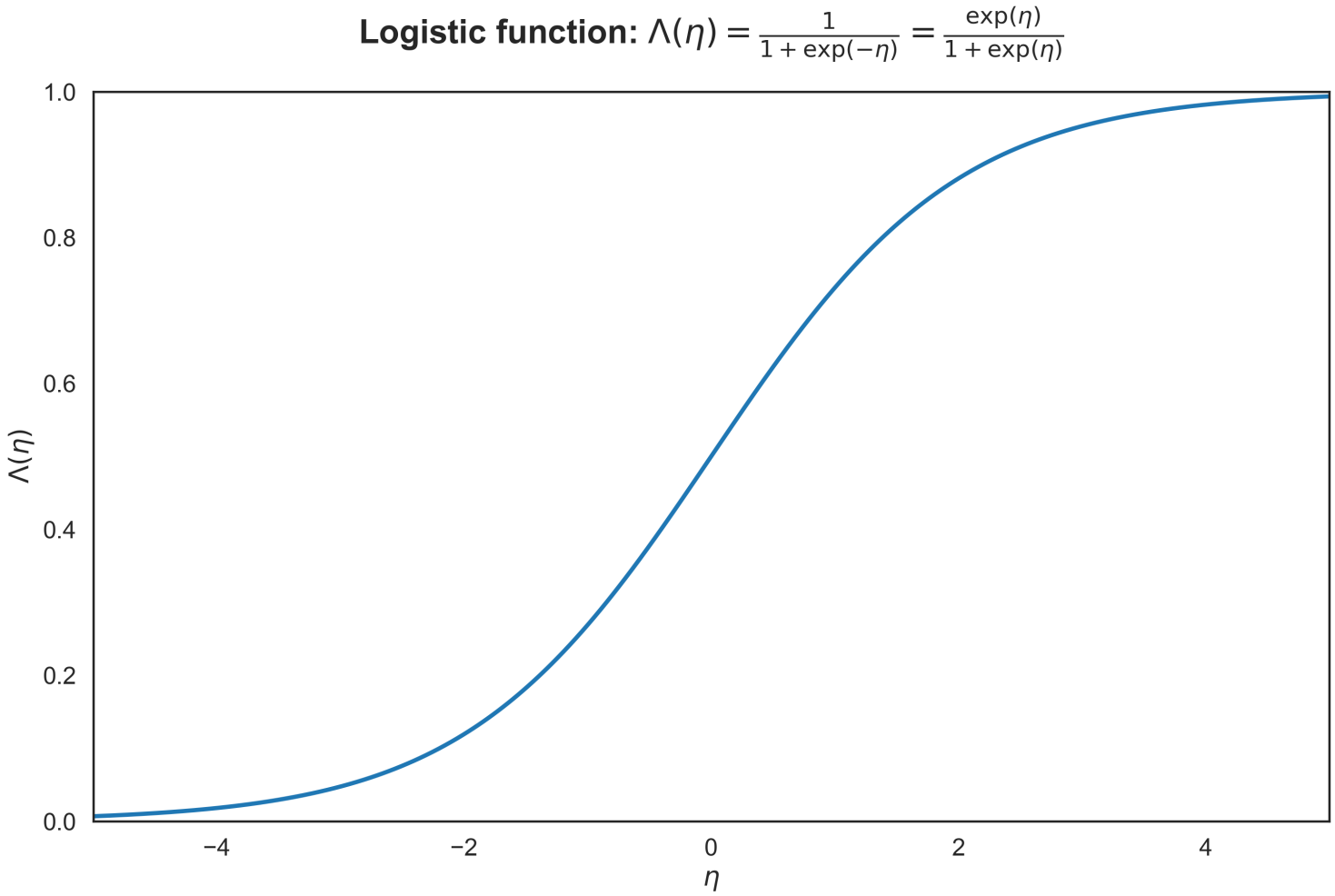
$$p(\boldsymbol{x}) = \frac{\exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)}$$

The **logistic function** (given below as a function of η):

$$\frac{\exp(\eta)}{1 + \exp(\eta)} = \frac{1}{1 + \exp(-\eta)}$$

constrains the probability to lie between zero and one.

Logistic function



Logistic Regression

The **odds** are defined as:

$$\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}$$

It follows from the logistic regression model that:

$$\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})} = \exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right)$$

Thus, logistic regression specifies a linear model for the log-odds:

$$\log\left(\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

The left-hand side is called the logit transformation of $p(\boldsymbol{x})$

Maximum likelihood estimation

We estimate the logistic regression model by maximum likelihood. Recall that a Bernoulli random variable Y takes values in $\{0, 1\}$ and has probability mass function

$$p(y; \pi) = \pi^y (1 - \pi)^{1-y}.$$

In the context of the logistic regression model, the probability mass function for a training case i is therefore

$$p(y_i | \boldsymbol{x}_i) = p(\boldsymbol{x}_i)^{y_i} (1 - p(\boldsymbol{x}_i))^{1-y_i}.$$

Maximum likelihood estimation

The likelihood function is

$$\begin{aligned}
\ell(\boldsymbol{\beta}) &= p(y_1|\boldsymbol{x}_1)\, p(y_2|\boldsymbol{x}_2)\, \ldots\, p(y_n|\boldsymbol{x}_n) \\
&= \prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i) \\
&= \prod_{i=1}^{n} p(\boldsymbol{x}_i)^{y_i} (1 - p(\boldsymbol{x}_i))^{1-y_i} \\
&= \prod_{i=1}^{n} \left(\frac{p(\boldsymbol{x}_i)}{1 - p(\boldsymbol{x}_i)} \right)^{y_i} (1 - p(\boldsymbol{x}_i))
\end{aligned}$$

This slide and the next one contain some mathematical derivations, which you will **not** need to reproduce on the exam

Maximum likelihood estimation

The log-likelihood is

$$
\begin{aligned}
L(\boldsymbol{\beta}) &= \log\left(\prod_{i=1}^{n}\left(\frac{p(\boldsymbol{x}_i)}{1-p(\boldsymbol{x}_i)}\right)^{y_i}(1-p(\boldsymbol{x}_i))\right) \\
&= \sum_{i=1}^{n}\left[y_i \log\left(\frac{p(\boldsymbol{x}_i)}{1-p(\boldsymbol{x}_i)}\right) + \log\left(1-p(\boldsymbol{x}_i)\right)\right] \\
&= \sum_{i=1}^{n}\left[y_i\left(\beta_0 + \sum_{j=1}^{p}\beta_j x_{ij}\right) - \log\left(1+\exp\left(\beta_0+\sum_{j=1}^{p}\beta_j x_{ij}\right)\right)\right]
\end{aligned}
$$

The negative log-likelihood, $-L(\boldsymbol{\beta})$, is known as the **cross-entropy loss function** or log-loss in machine learning. Minimising this loss function is equivalent to maximizing the likelihood.

Maximum likelihood estimation

The MLE for the logistic regression model is

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}}\ L(\boldsymbol{\beta})$$

where

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) - \log \left(1 + \exp \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right) \right]$$

Maximum likelihood estimation

Optimisation. Setting the partial derivatives of the log-likelihood to zero leads to estimation equations that are nonlinear in the coefficients. Numerical optimisation routines are used to obtain the estimates.

Statistical inference. Statistical inference for the logistic regression model can be conducted using the large sample theory for maximum-likelihood estimation.

Predicted probabilities

The predicted probability given observed input values $x_1, \ldots, x_p$ is

$$\widehat{p}(\boldsymbol{x}) = \frac{\exp(\widehat{\beta}_0 + \sum_{j=1}^{p} \widehat{\beta}_j x_j)}{1 + \exp(\widehat{\beta}_0 + \sum_{j=1}^{p} \widehat{\beta}_j x_j)},$$

where $\widehat{\beta}_0$, $\widehat{\beta}_1, \ldots,$ $\widehat{\beta}_p$ are the maximum likelihood estimates.

Example: customer churn data

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	350
Model:	Logit	Df Residuals:	348
Method:	MLE	Df Model:	1
Date:		Pseudo R-squ.:	0.1319
Time:		Log-Likelihood:	-208.99
converged:	True	LL-Null:	-240.75
		LLR p-value:	1.599e-15

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-1.2959	0.189	-6.866	0.000	-1.666	-0.926
Avg_Ret_Exp	0.0308	0.004	7.079	0.000	0.022	0.039

Example: customer churn

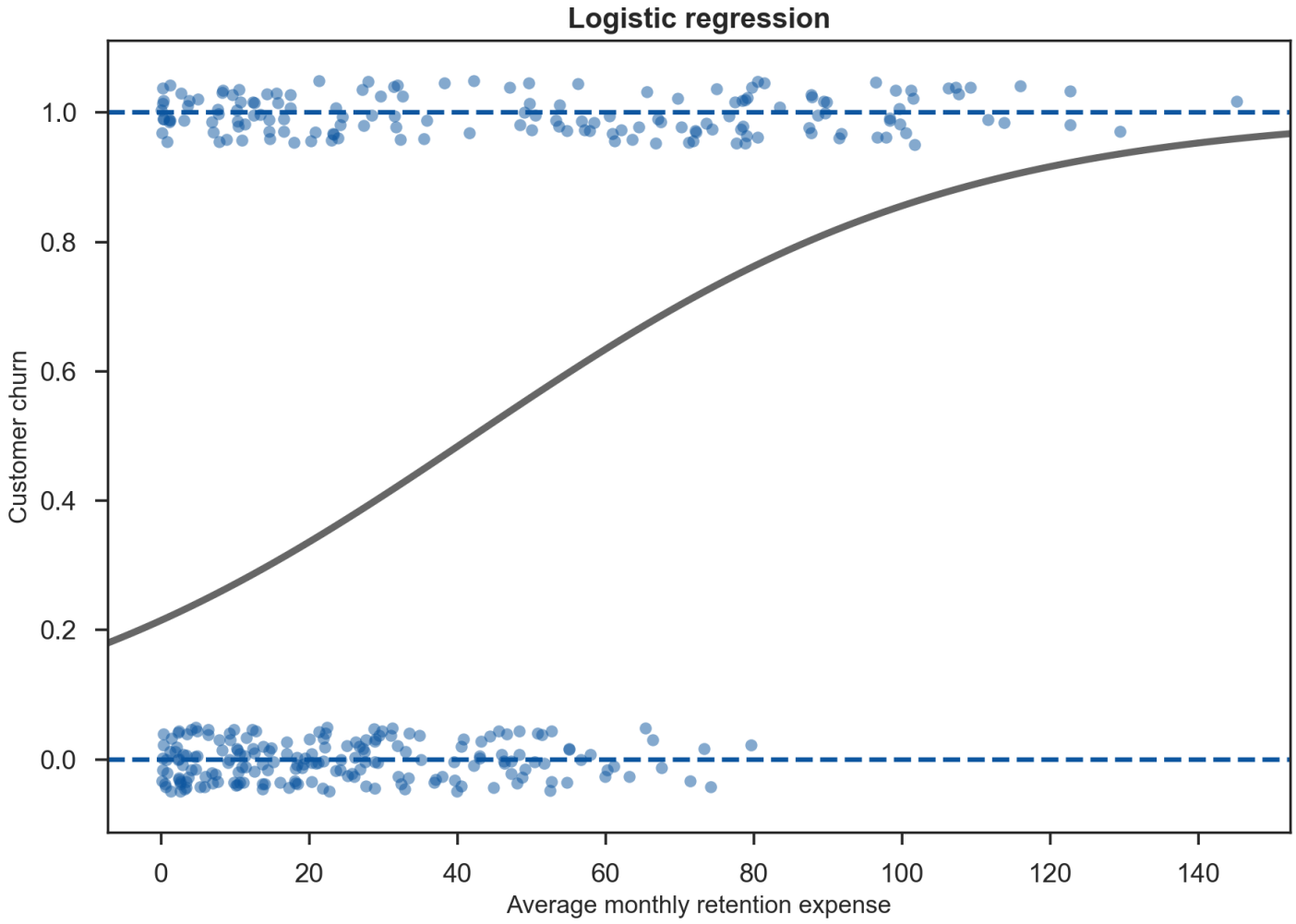
Suppose that we want to predict the probability that a customer with average retention expenses of 100 will churn.

$$\widehat{p} = \frac{\exp(\widehat{\beta}_0 + \widehat{\beta}_1 \times 100)}{1 + \exp(\widehat{\beta}_0 + \widehat{\beta}_1 \times 100)}$$

$$= \frac{\exp(-1.296 + 0.031 \times 100)}{1 + \exp(-1.296 + 0.031 \times 100)}$$

$$= 0.856$$

Example: customer churn data



Regularised logistic regression

We can apply regularised empirical risk minimisation to logistic regression. Using an ℓ_1 penalty as in the Lasso, we solve the following problem:

$$\min_{\boldsymbol{\beta}} \; -L(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|$$

where

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) - \log \left(1 + \exp \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right) \right]$$

Subset selection methods also naturally extend to logistic regression.

Extension: Multinomial logistic regression

The **multinomial logistic regression** is a generalisation of logistic regression beyond the binary response setting to responses with multiple classes: $Y = 1, 2, \ldots, C$.

For $y = 1, \ldots, C$ the model specifies:

$$P(Y = y|X = \boldsymbol{x}) = \frac{\exp\left(\beta_{y0} + \beta_{y1}x_1 + ... + \beta_{yp}x_p\right)}{\sum_{y'=1}^{C} \exp\left(\beta_{y'0} + \beta_{y'1}x_1 + ... + \beta_{y'p}x_p\right)}$$

where $\beta_{y0}, ..., \beta_{yp}$ are coefficients for the class y

and $\beta_{y'0}, ..., \beta_{y'p}$ are coefficients for the class $y' = 1, ..., C$.

Multinomial logistic regression

To avoid redundancy in the parameters, we choose one class, typically C, as the baseline, and set $\beta_{C0} = \beta_{C1} = ... = \beta_{Cp} = 0$.

Then, for $y = 1, \ldots, C-1$:

$$P(Y = y|X = \boldsymbol{x}) = \frac{\exp\left(\beta_{y0} + \beta_{y1}x_1 + ... + \beta_{yp}x_p\right)}{1 + \sum_{y'=1}^{C-1} \exp\left(\beta_{y'0} + \beta_{y'1}x_1 + ... + \beta_{y'p}x_p\right)}$$

and for class C:

$$P(Y = C|X = \boldsymbol{x}) = \frac{1}{1 + \sum_{y=1}^{C-1} \exp\left(\beta_{y'0} + \beta_{y'1}x_1 + ... + \beta_{y'p}x_p\right)}$$

Gaussian discriminant analysis

Gaussian discriminant analysis

In **Gaussian discriminant analysis**, we assume that the predictors are normally distributed conditional on the class

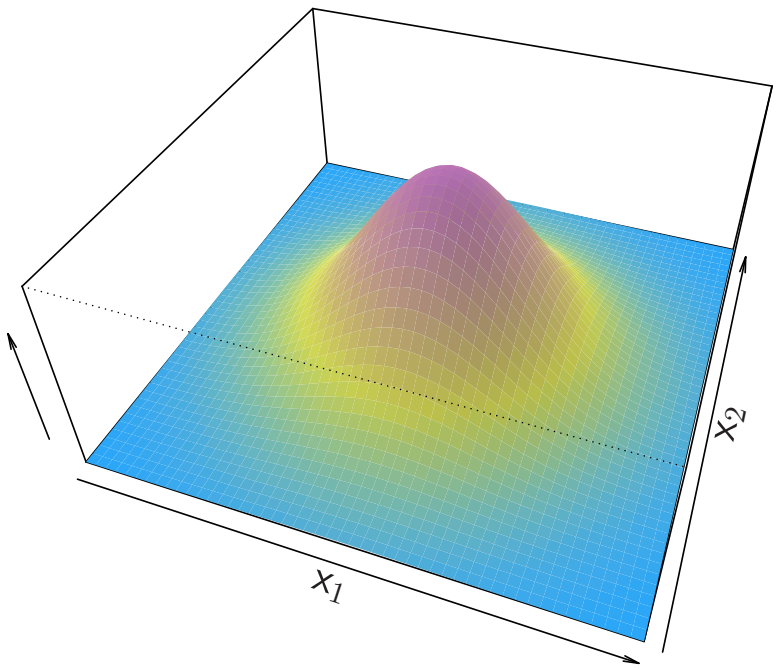
$$X|Y = y \sim N(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$$

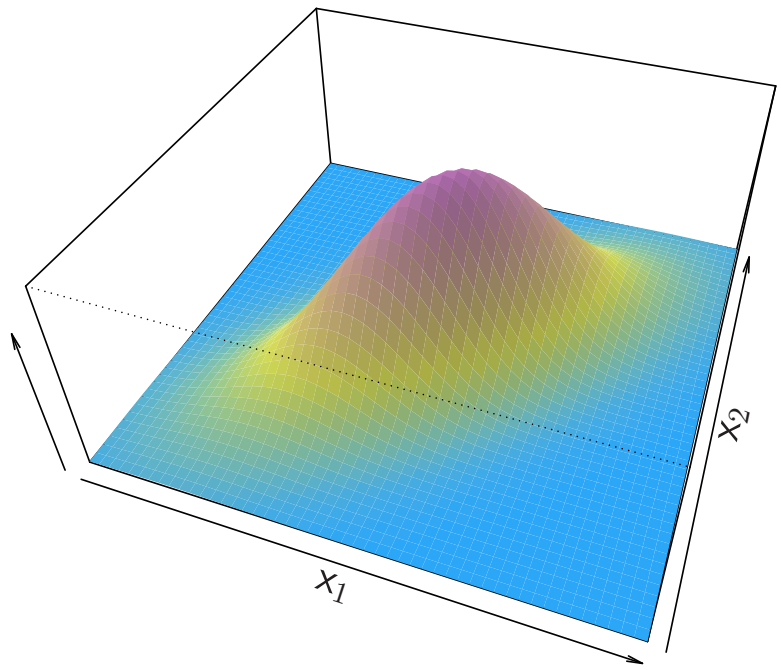
where $\boldsymbol{\mu}_y$ is the mean of the predictors and $\boldsymbol{\Sigma}_y$ is the covariance matrix of the predictors for the observations corresponding to the class y.

If no additional assumptions are placed on $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$, we call this approach **Quadratic Discriminant Analysis** (QDA).

Note that if we assume that $\boldsymbol{\Sigma}_y$ is diagonal, the approach becomes Naive Bayes with the Gaussian assumption on the class conditional predictor densities (because diagonal covariance matrix in the Gaussian case means independence).

Multivariate normal distribution





Quadratic discriminant analysis

Using Bayes' rule, together with some algebraic manipulations, we can write the conditional probability in the form:

$$P(Y = y|X = \boldsymbol{x}) = \frac{\exp\left[\delta_y(\boldsymbol{x})\right]}{\sum_{y'=1}^{C} \exp\left[\delta_{y'}(\boldsymbol{x})\right]},$$

where $\delta_y(\boldsymbol{x})$ is a quadratic function of $\boldsymbol{x}$ that depends on the parameters $\boldsymbol{\mu}_y$, $\boldsymbol{\Sigma}_y$ and π_y (where $\pi_y = P(Y = y)$, as before).

Function $\delta_y(\boldsymbol{x})$ is called the **discriminant** function.

Quadratic discriminant analysis

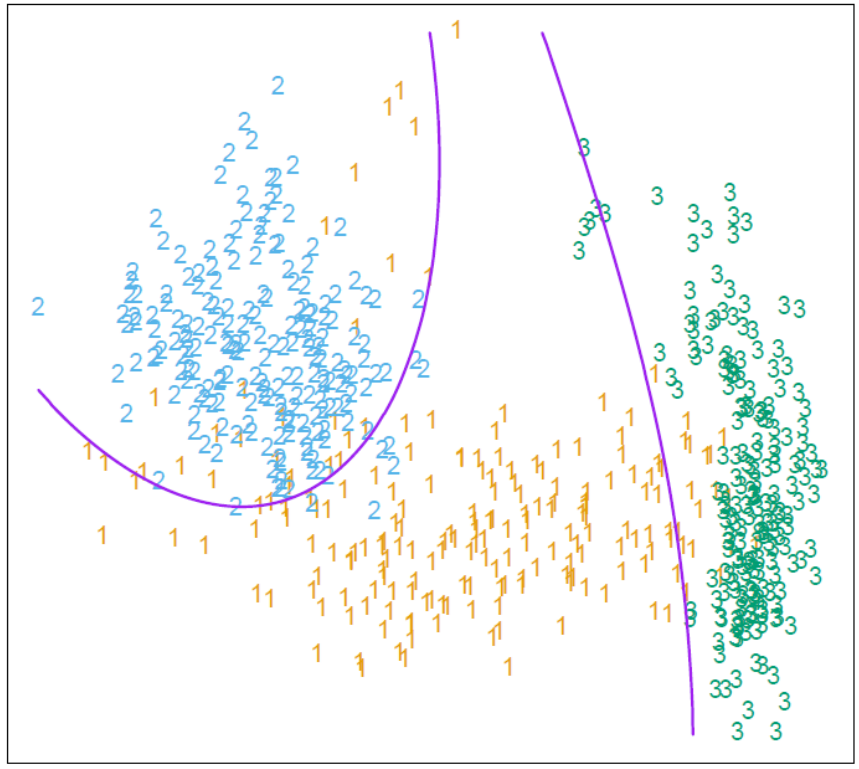
We can describe any decision rule in terms of the discriminant function. For example, under the zero-one loss function, the decision rule is

$$\widehat{y}(\boldsymbol{x}) = \operatorname*{argmax}_y \delta_y(\boldsymbol{x}).$$

Thus, we classify each subject to the class with the highest value of the discriminant function.

The decision boundary between two classes (y_1 and y_2) is given by $\{\boldsymbol{x} : \delta_{y_1}(\boldsymbol{x}) = \delta_{y_2}(\boldsymbol{x})\}$ and is described by a quadratic equation.

Quadratic decision boundary in QDA



Maximum likelihood estimation (MLE)

Let n_y be number of training observations in class y

We use MLE to estimate the model parameters (for $y = 1, \ldots, C$):

$$\widehat{\pi}_y = \frac{n_y}{n}$$

$$\widehat{\boldsymbol{\mu}}_y = \frac{1}{n_y} \sum_{i:\, y_i = y} \boldsymbol{x}_i$$

i.e. the mean of the $\boldsymbol{x}$ values from class y

$$\widehat{\boldsymbol{\Sigma}}_y = \frac{1}{n_y} \sum_{i:\, y_i = y} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_y)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_y)^T$$

i.e. the "sample covariance" using just the $\boldsymbol{x}$ values from class y

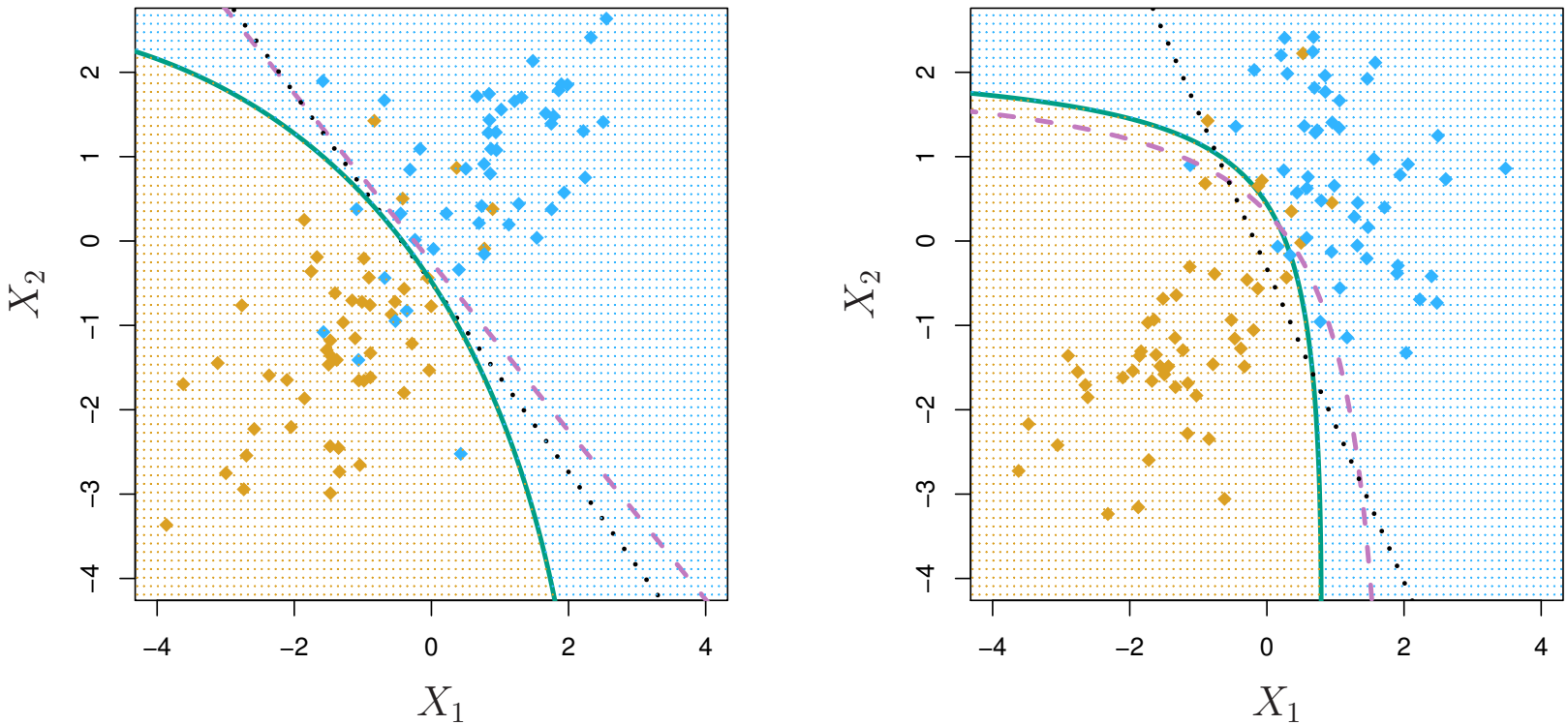
Linear discriminant analysis

Linear Discriminant Analysis (LDA) is a special case of the Gaussian discriminant analysis.

In LDA we assume that the classes have a common covariance matrix. That is, $\boldsymbol{\Sigma}_y = \boldsymbol{\Sigma}$ for $y = 1, \ldots, C$.

This assumption leads to a **linear** discriminant function and therefore results in linear decision boundaries.

Linear and quadratic discriminant analysis



Bayes decision boundary is purple dashed, the LDA one is black dotted, and the QDA one is green solid

Maximum likelihood estimation

LDA estimates the parameters the same way as QDA, except that to estimate $\boldsymbol{\Sigma}$ we compute the so-called pooled covariance matrix:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{y=1}^{C} \sum_{i:\, y_i = y} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_y)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_y)^T,$$

for $y = 1, \ldots, C$.

Comparison to logistic regression

Because the LDA discriminant functions is linear, the conditional probability in LDA has the form:

$$p(Y = y|X = \boldsymbol{x}) = \frac{\exp\left(\alpha_{y0} + \alpha_{y1}x_1 + ... + \alpha_{yp}x_p\right)}{\sum_{y'=1}^{C} \exp\left(\alpha_{y'0} + \alpha_{y'1}x_1 + ... + \alpha_{y'p}x_p\right)}$$

for some coefficients α.

Note that the conditional probability in logistic regression has the same form (but we used β instead of α). In particular, both LDA and logistic regression result in linear decision boundaries.

However, the two methods differ in how they estimate the coefficients.

Regularised Gaussian discriminant analysis

We can add regularisation to the QDA approach by shrinking matrixes $\boldsymbol{\Sigma}_y$ towards diagonal matrices. This achieves a compromise between QDA and the Naive Bayes method.

We can similarly add regularisation to the LDA method by shrinking $\boldsymbol{\Sigma}$ towards a diagonal matrix.

Empirical example

Customer churn data

- We randomly split the data to allocate 70% of customers (350 observations) to the training set.
- The customer attrition rate in the training data is 45%.
- Among the six predictors, three are continuous, two are count variables, and one is binary.

Logistic regression

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	350
Model:	Logit	Df Residuals:	343
Method:	MLE	Df Model:	6
Date:		Pseudo R-squ.:	0.4722
Time:		Log-Likelihood:	-127.06
converged:	True	LL-Null:	-240.75
		LLR p-value:	2.783e-46

	coef	std err	z	P>|z|	[0.025	0.975]
const	4.5768	0.897	5.101	0.000	2.818	6.335
Avg_Ret_Exp	0.0584	0.008	7.596	0.000	0.043	0.073
Revenue	-0.0258	0.010	-2.537	0.011	-0.046	-0.006
Employees	-0.0039	0.000	-7.846	0.000	-0.005	-0.003
Total_Crossbuy	-0.8610	0.128	-6.705	0.000	-1.113	-0.609
Total_Freq	-0.0547	0.027	-2.002	0.045	-0.108	-0.001
Industry	0.2758	0.335	0.823	0.410	-0.381	0.933

Test results

Classification results

	Error rate	True Pos. Rate	True Neg. Rate	AUC	Precision
Logistic regression	0.213	0.747	0.827	0.887	0.812
ℓ_1 regularised	0.213	0.747	0.827	0.888	0.812
ℓ_2 regularised	0.207	0.720	0.867	0.887	0.844
LDA	0.187	0.760	0.867	0.887	0.851
QDA	0.187	0.760	0.867	0.889	0.851
Regularised QDA	0.173	0.760	0.893	0.889	0.877
KNN	0.193	0.653	0.960	0.849	0.942

Comparison of classification methods

Gaussian discriminant analysis vs. logistic regression and KNN

If the assumptions of Gaussian discriminant analysis are correct, the model will need less training data than logistic regression to achieve a given level of performance.

Alternatively, if the assumptions are incorrect logistic regression will generally perform better.

KNN makes no assumptions about the decision boundary, so we can expect KNN to do better than LDA and logistic regression when the boundary is highly nonlinear. On the other hand, LDA and logistic regression will do better than KNN when there are few training observations and reducing the variance is critical.

QDA can be seen as a compromise between a nonparametric approach (like KNN) and one that produces a linear decision boundaries (like LDA or logistic regression).

Review questions

- What is a confusion matrix? Write down what the matrix looks like. What are true positive rate, true negative rate, and precision? Why is there a trade-off between the true positive rate and the true negative rate?
- What is the logistic regression model? What is Gaussian discriminant analysis?
- What are the similarities and the differences between LDA and QDA? What are the similarities and the differences between LDA and logistic regression?
- In what situations does each of the classification methods that we have studied so far tend to be the most useful?