# Ch. 4. Classification

The linear regression model discussed in Chap. 3 assumes that the response variable Y is quantitative. But in many situations, the response variable is instead qualitative.
Often qualitative variables are referred to as categorical.

Predicting qualitative responses → **Classification**.

Predicting a qualitative response for an obs can be referred to as classifying that obs, since it involves assigning the obs to a category or class.
Often the methods used for classification first predict the probability of each of the categories of a qualitative variable, as the basis for marking the classification. In this sense they also behave like regression methods.

## An Overview of Classification

Examples of classification problems:

1. A person arrives at the ER with a set of symptoms(predictors) that could possibly be attributed to one of three medical conditions(categories). Which of the 3 conditions does he have?
2. An online banking service must be able to determine whether or not a transaction being performed on the site is fraudulent(category), on the basis of the user's IP address, past transaction history, and so forth(predictors).
3. On the basis of DNA sequence data for a number of patients with and without a given disease(predictors: DNA sequence and presence of disease), a biologist would like to figure out which DNA mutations are deleterious and which are not(categories).

We have a set of training obs that we can use to build a classifier. We want our classifier to perform well not only on the training data, but also on test observations that were not used to train the classifier.

## Logistic Regression

Rather than modeling the response Y directly, logistic regression models the probability that Y belongs to a particular category k, given the predictor X.
$Pr(Y=k \mid X = x)$
The value will vary between 0 and 1

## The Logistic Model

How should we model the relationship between $p(X) = Pr(Y = 1 \mid X = x)$ and X? (for convenience we are using the generic 0/1 coding for the response).

We could use a linear regression model to represent these probabilities:
$p(X) = \beta_0 + \beta_1 X$

If we use this model to predict a two-class classification problem, we will obtain a model that is a straight line(duh!). The problem with this approach is that, for some values of X we could have a probability lower than 0 or higher than 1 → not a probability.
To avoid this problem, we must model p(X) using a function that gives outputs between 0 and 1 for all values of X.
Many functions meet this description (e.g. step function). In logistic regression we use the logistic function:
$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$
To fit the model, we use a method called maximum likelihood(see next section).
The logistic function will always produce an S-shaped curve with y-values between 0 and 1 and so, regardless of the value of X, we will obtain a sensible prediction.
The logistic model is better able to capture the range of probabilities than the linear regression.
After a bit of manipulation we find that

$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}$
the quantity p(X)/[1-p(X)] is called the odds, and can take on any value between 0 and plus infinity.
Values of the odds close to 0 and plus infinity indicate very low and very high probabilities of the category associated with 1(for example disease, not disease associated with 0).
example: probability of being in category disease = ⅕ → odds for disease are ¼ .
Interpretation: if I take a total of 5 people, 1 will have the disease while 4 will be healthy.

odds for a certain event are $(1,\infty)$ → odds are in favour
odds for a certain event are in [0,1) → odds are against

to make the two intervals symmetric we take the log of both sides: log odds or logits
$log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X$
The left-hand side is called the log odds or logits. The logistic regression model has a logit that is linear in X.
In the linear regression model, $\beta_1$ gives the average change in Y associated with a one-unit increase in X.
In contrast, in a logistic regression model, increasing X by one unit changes the log odds by $\beta_1$, or equivalently it multiplies the odds by $e^{\beta_1}$.
However, because the relationship between p(X) and X is not a straight line, $\beta_1$ does not correspond to the change in p(X) associated with a one-unit increase in X. The amount that p(X) changes due to a one-unit change in X will depend on the current value of X. But regardless of the value of X, if $\beta_1$ is positive then increasing X will be associated with increasing p(X), and if $\beta_1$ is negative then increasing X will be associated with decreasing p(X).

## Estimating the Regression Coefficients

The coefficients $\beta_0$ and $\beta_1$ are unknown and must be estimated based on the available training data. In chapter 3 we used the least squares approach to estimate the unknown linear regression coefficients. Although we could use (non-linear) least squares to fit the model, the more general method of maximum likelihood is preferred, since it has better statistical properties. The basic intuition behind using maximum likelihood to fit a logistic regression model is as follows:
we seek estimates for $\beta_0$ and $\beta_1$ such that the predicted probability for *disease*(example) corresponds as closely as possible to the observed *disease* status. In other words, we try to find such that plugging these estimates in the logistic function, yields a number close to 1 for all individuals who have the disease and close to zero for all individuals who are healthy.
This intuition can be formalized using a mathematical equation called a likelihood function(below the likelihood for a two-class classifier):

$$\ell(\beta_0, \beta_1) = \prod_{i:y_1=1} p(x_i) \prod_{i':y_{i'}=1} (1 - p(x_{i'}))$$

The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to maximize this likelihood function.
Actually, it's convenient and equivalent to maximize the log-likelihood function.
Maximum-likelihood is a very general approach that is used to fit many of the non-linear models that we examine throughout this book.
Many aspects of the output of logistic regression are similar to the output of a linear regression.
The z-statistic plays the same role as the t-statistic in the linear regression output.
For instance, the z-statistic associated with $\beta_1$ is equal to $\hat{\beta}_1 / SE(\hat{\beta}_1)$ , and so a large(absolute) value of the z-statistic indicates evidence against the null hypothesis $H_0$: $\beta_1 = 0$. This null hypothesis implies that $p(X) = \frac{e^{\beta_0}}{1 + e^{\beta_0}}$ , in other words that the probability of belonging to the class(e.g. disease) does not depend on the predictor.
If the associated p-values are tiny (we usually use an alpha-level dependent on the setting as a threshold), we conclude that there is indeed an association between the predictor and the class. The estimated intercept is typically not of interest; its main purpose is to adjust the average fitted probabilities to the proportion of ones in the data.

## Making Predictions

once the coefficients have been estimated, it is a simple matter to compute the probability of belonging to a class for any given observation.
We just plug the values of the predictor into the fitted logistic function $\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}}$ and we get the estimated probability. One can use qualitative predictors using the dummy variable approach.
After having estimated the probabilities of belonging to one class(and by consequence, also the probability of belonging to the other) we can actually classify the observations using a probability threshold. For example if we use 0.5 as threshold we classify an observation as belonging to the most probable class. But, given additional information about the problem,

we could want to assign an obs to a class only if we are super confident of its probability of belonging to that class. In such a case we could choose to use a higher threshold, such as 0.9.

## Multiple Logistic Regression

Just add more terms to the logistic function. The results obtained using one predictor may be quite different from those obtained using multiple predictors, especially when there is correlation among the predictors. Beware of *confounding*.

## Linear Discriminant Analysis

Logistic regression involves directly modeling Pr(Y = k | X = x) using the logistic function. In statistical jargon, we model the conditional distribution of the response Y, given the predictor(s) X.
We now consider an alternative and less direct approach to estimating these probabilities.

We a) model the distribution of the predictors X separately in each of the response classes (i.e. given Y), and then b) use Bayes' theorem to flip these around into estimates for Pr(Y = k | X = x).

a) estimate the distribution of P(X = x | Y = k) for each class
b) use Bayes' to estimate the posterior probabilities P(Y = k | X = x)

Reasons for LDA over logistic regression:

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. LDA does not suffer from this problem.
- If n is small and the distribution of the predictors X is approximately normal in each of the classes, LDA is again more stable than the logistic regression model.
- LDA is popular

## Using Bayes' Theorem for Classification

Suppose that we wish to classify an obs into one of K classes, where K ≥ 2 (i.e. the qualitative response variable Y can take K possible distinct and unordered values.

- Let $\pi_k$ represent the overall or prior probability that a randomly chosen observation comes from the kth class; this is the probability that a given observation is associated with the kth category of the response variable Y.
- Let f(x) = Pr(X =x | Y = k) (technically correct only for discrete RVs) denote the density function of X for an observation that comes from the kth class.

Then **Bayes' Theorem** states that:

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

which is P(Y = k | X = x) = [P(Y = k) P(X = x | Y = y)] / P(X = x)

Instead of computing directly $p_k(X) = Pr(Y = k \mid X = x)$ we can simply plug estimates of $\pi_k$ and $f_k(x)$ into Bayes' Theorem.

In general, computing $\pi_k$ (the prior) is easy if we have a random sample of Ys from the population: we simply compute the fraction of obs that belong to the kth class.

However, estimating $f_k(x)$ tends to be more challenging, unless we assume some simple forms for these densities.

We refer to $p_k(x)$ as the posterior probability that an observation X = x belongs to the kth class. That is, it is the probability that the observation belongs to the kth class, given the predictor value for that observation.

We know from Chapter 2 that the Bayes classifier, which classifies an obs to the class for which is $p_k(x)$ the largest, has the lowest possible error rate out of all classifiers (this is of course only true if the terms in the Bayes' Theorem equation are all correctly specified).

Therefore, if we can find a way to estimate $f_k(x)$, then we can develop a classifier that approximates the Bayes classifier.

## Linear Discriminant Analysis for p = 1

For now, assume we only have one predictor, p = 1.
We would like to obtain an estimate for $f_k(x)$ to plug into our equation for the Bayes' Theorem in order to estimate $p_k(x)$. We will then classify an observation to the class for which $p_k(x)$ is greatest.

In order to estimate $f_k(x)$ we will first make some assumptions about its form:

- $f_k(x)$ is *normal* or *gaussian* with parameters $\mu_k$ and $\sigma_k^2$.
- There is a shared variance term $\sigma^2$ across all K classes: $\sigma_1^2 = \ldots = \sigma_k^2 = \sigma^2$.

We can now plug this variable in Bayes and get and estimate of $p_k(x)$, and assign X = x to the class k for which it is the largest.

Taking the log of Bayes (with our estimates plugged in) and rearranging the terms, this (the classification process) is equivalent to assigning the observation to the class for which

$$\delta_k(x) = x \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + log(\pi_k) \quad \leftarrow \text{discriminant function}$$

is largest.
LDA approximates the Bayes classifier by plugging estimates for the above equation:

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

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

where n is the total number of training obs, and $n_k$ is the number of training observations in the kth class.

The estimate for $\mu_k$ is simply the average of all the training obs from the kth class.
The estimate for $\sigma^2$ can be seen as a weighted average of the sample variances for each of the K classes.

Sometimes we have knowledge of the class membership probabilities $\pi_1, \ldots, \pi_k$ which can be used directly. In the absence of any additional information, LDA estimates using the proportions of the training obs that belong to the kth class:
$\hat{\pi}_k = n_k/n$
LDA plugs the estimates into the discriminant function and assigns an obs X = x to the class for which

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

is largest. The word linear stems from the fact that the discriminant functions are linear functions of x.

## Linear Discriminant Analysis for p > 1

We now extend the LDA classifier to the case of multiple predictors. To do this we will **assume** that X = (X, … , X) is drawn from a *multivariate Gaussian* distribution, with a class-specific mean vector and a common covariance matrix.

X ~ N(μ, $\boldsymbol{\Sigma}$)

The multivariate Gaussian distribution assumes that each individual predictor follows a one-dimensional normal distribution with some correlation between each pair of predictors.

Suppose 2 predictors, $X_1$ and $X_2$: if $Cor(X_1, X_2) = 0$ then their surface will have a normal *bell shape*.
If $Cor(X_1, X_2)$ is different from 0 or they have uneven variance, the bell shape will be distorted.

The model is the same, just in higher dimensions (vector/matrix version of discriminant functions) pg. 143.

pg. 144-148: type of errors,confusion matrix, sensitivity/specificity/ROC curve (not specific for LDA)

## Quadratic Discriminant Analysis

LDA assumes that the obs within each class are drawn from a multivariate Gaussian distribution with a class-specific mean vector and a covariance matrix that is common to all K classes.

*Quadratic discriminant analysis* (QDA) provides an alternative approach.
Like LDA, it assumes that the obs from each class are drawn from a Gaussian distribution and plugging estimates for the parameters into Bayes in order to perform prediction.
However, unlike LDA, QDA assumes that each class has its own covariance matrix.
Thas is, it assumes that an obs from the kth class if of the form $X \sim N(\mu, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\Sigma}_k$ is a covariance matrix for the kth class.

So the QDA classifier involves plugging estimates for into the discriminant functions and then assigning an obs X = x to the class for which this quantity is the largest. Unlike in LDA, the quantity x appears as a quadratic function in the discriminant functions.

Why does it matter whether or not we assume that the K classes share a common covariance matrix?
In other words, why would one prefer LDA to QDA, or vice-versa?
The answer lies in the bias-variance trade-off.
Where there are p predictors, then estimating a covariance matrix requires estimating p(p+1)/2 parameters. QDA estimates a separate covariance matrix for each class, for a total of Kp(p+1)/2 parameters. With 50 predictors this is some multiple of 1275 which is a lot of parameters. By assuming that the K classes share a common covariance matrix, the LDA model becomes linear in x, which means there are Kp linear coefficients to estimate.

Consequently, LDA is a much less flexible classifier than QDA, and so has substantially lower variance.

But there is a trade-off: if LDA's assumption that the K classes share a common covariance matrix is badly off, then LDA can suffer from high bias.

Roughly speaking, LDA tends to be a better bet than QDA if there are relatively few training obs and so reducing variance is crucial.
In contrast, QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix for the K classes is clearly untenable.