Logistic Regression

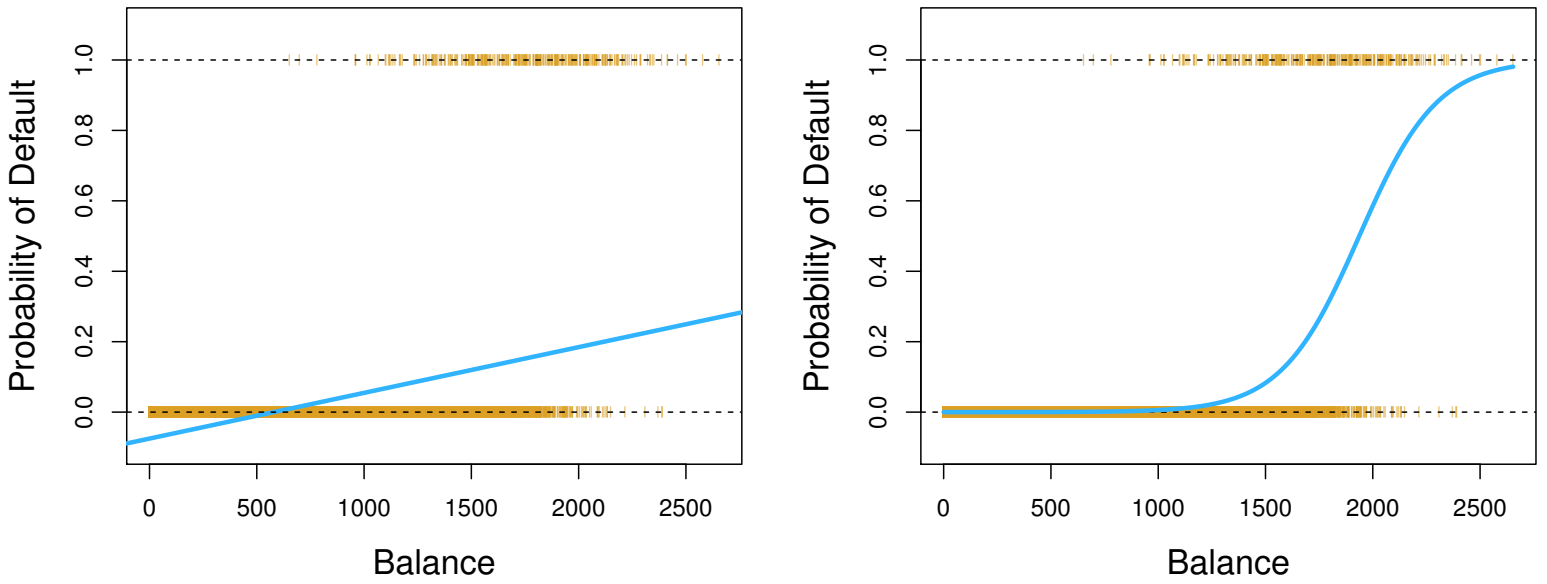
Ramesh Srinivasan

October 8, 2024

Classification

- Qualitative variables take values in an unordered set C, such as: eye color $\in$ {brown, blue, green}
- The classification task is to build a function C(X) that takes as input the feature vector X and predicts its value for Y, i.e., C(X) $\in$ C
- Often we are more interested in estimating the probabilities that X belongs to each category in C

Linear vs Logistic Regression: Credit Default Example



- The orange dots are the the labels of the two classes which we arbitrary mark as 0 and 1
- Linear Regression is not bounded correctly.

Logistic Function

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

$$ln(\frac{p(x)}{1-p(x)}) = \beta_0 + \beta_1 x$$

- The first expression is a 1 variable logistic regression model and the lower expression gives the log odds.
- The log odds gives us an intutition on how to interpret the logistic regression model - increasing X by one unit changes the odds by e^{β_1}

Likelihood

$$L(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{j:y_j=1} (1 - p(x_j))$$

with the i and j combining to all the training data.
In practice, we would try to maximize this function by minimizing the negative log Likelihood.
The negative log Likelihood is an example of a loss function (cost function) that has to be optimized numerically.

Multiple Logistic Regression

$$\hat{p} = \frac{e^{w^T x}}{1 + e^{w^T x}}$$

$$\hat{p} = \frac{1}{1 + e^{-w^T x}}$$

- The logistic regression model can be written in vectorized form where w are the weights that we can return and x are out feature vector.
- The decision boundaries are exactly at the position where the two classes are equiprobable. The boundary decision probability is exactly 0.5. Solving our sigmoid function for $p = 0.5$:

$$\hat{p} = \frac{1}{1 + e^{-w^T x}} = 0.5 = \frac{1}{1 + 1}$$

$$e^{-w^T x} = 1$$

$$w^T x = 0$$

Bayes Classifier

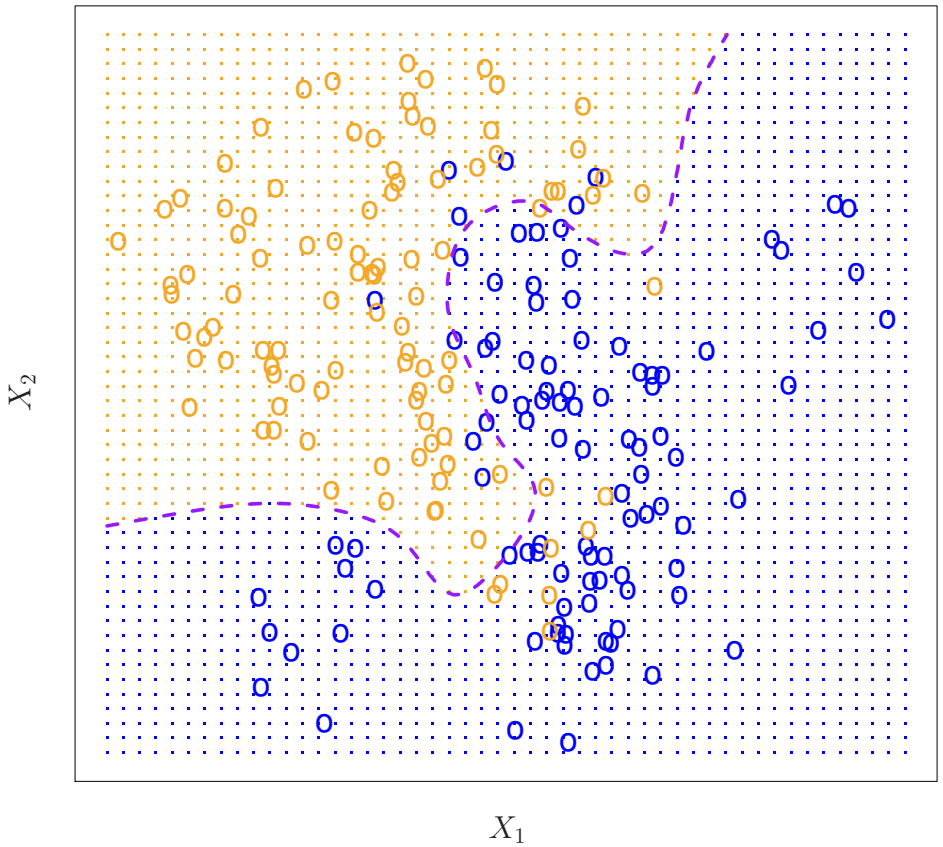
- Suppose we fit a classifier model $\hat{f}(x)$ to some **training** data $Tr = \{x_i, y_i\}$ of size N. Our metric of performance is of course the performance of test data that we have set aside.
- a very simple classifier that assigns each observation to the most likely class, given its predictor values will minimize the test error rate **on average**. In other words, we should simply assign a test observation with predictor vector x_i to the class j for which

$$p(Y = j|X = x_i)$$

  is largest.
- This simple classifier is called a Bayes classifier.
- In a 2-class (0/1) classifier, if

$$p(Y = 1|X = x_i) > 0.5$$

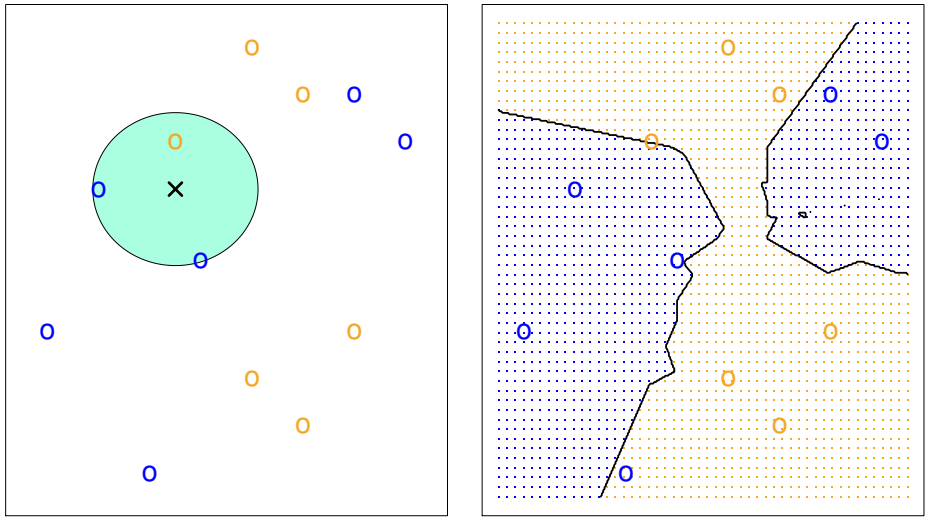
  the sample x_i is of class 1.



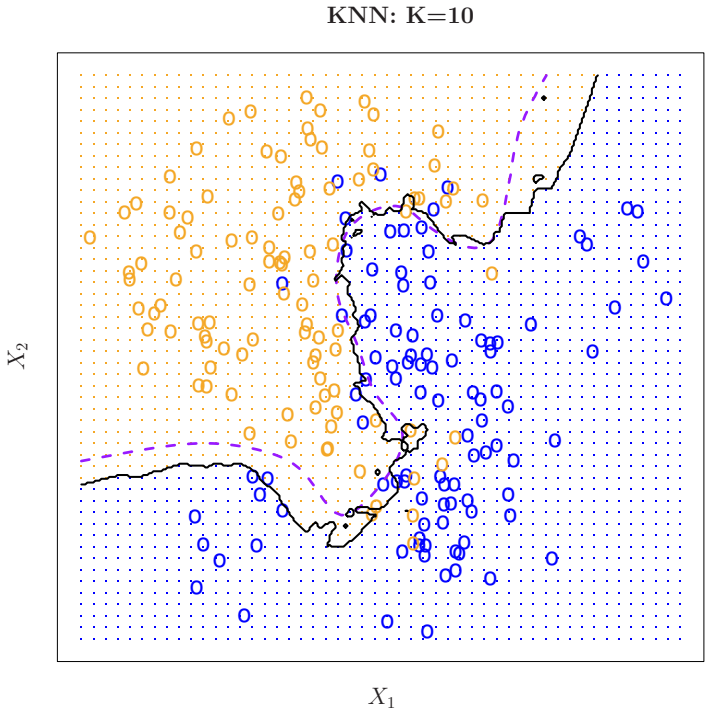
A simulated data set consisting of 100 observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.

K-nearest neighbor classifier

- Since we dont know the probability distribution that generated the real data, we cant even know the Bayes classifer.
- K-NN classifiers attempt to estimate the conditional distribution of Y given X, so that we can classify an observationx_i to the class with the highest estimated probability.
- the KNN classifier identifies K points in the training data closest to x_i. The conditional probability for each class j is just the fraction of the K points where Y = j.
- KNN computes the probability for each j and then chooses the maximum probability.



The KNN approach, using K = 3, is illustrated in a simple situation with six blue observations and six orange observations. Left: a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue. Right: The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.



The black curve indicates the KNN decision boundary on the data from Figure 2.13, using K = 10. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.