

Chapter4 Fitting a Model to Data

Data Science For Business

Professor: C.H. Lai

Student: 11204604 Tse Wen Hong

Nov. 14, 2023 [[gDoc](#)]

Chapter.4 Fitting a Model to Data

Fundamental concepts:

- *Finding “optimal” model parameters based on data; Choosing the goal for data mining; Objective functions; Loss functions.*

Exemplary techniques:

- *Linear regression; Logistic regression; Support-vector machines.*

Chapter.4 Fitting a Model to Data

A common case is where the structure of the model:

- *a parameterized mathematical function, or*
- *equation of a set of numeric attributes.*

Parameter learning or Parametric modeling:

- *specifies the form of the model and the attributes*
- *to tune the parameters so that the model fits the data as well as possible*

Index

1. Classification via Mathematical Functions

- 1 Linear Discriminant Functions
- 2 Optimizing an Objective Function
- 3 An Example of Mining a Linear Discriminant from Data
- 4 Linear Discriminant Functions for Scoring and Ranking Instances
- 5 Support Vector Machines, Briefly

2. Regression via Mathematical Functions

3. Class Probability Estimation and Logistic “Regression”

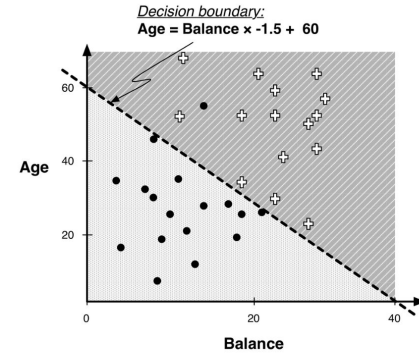
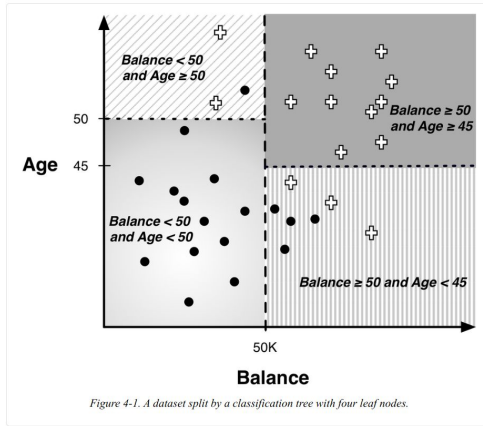
- * Logistic Regression: Some Technical Details

4. Example: Logistic Regression versus Tree Induction

5. Nonlinear Functions, Support Vector Machines, and Neural Networks

6. Summary

1-1 Linear Discriminant Functions

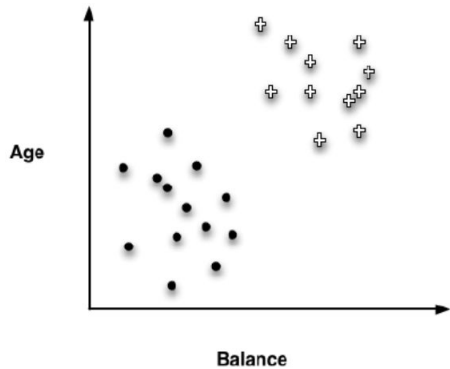


Entropy helps us to set up the boundaries that partition the instance space into similar regions. and the homogeneous regions helps to predict the target variable of a new, unseen instance by determining which segment it falls into.

however, there are other, possibly better, ways to partition the space.

*This is called a **linear classifier** and is essentially a weighted sum of the values for the various attributes.*

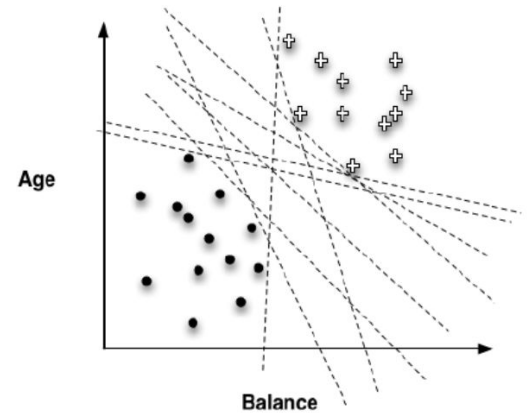
1-1 Linear Discriminant Functions



$$\text{class}(\mathbf{x}) = \begin{cases} + & \text{if } 1.0 \times \text{Age} - 1.5 \times \text{Balance} + 60 > 0 \\ \bullet & \text{if } 1.0 \times \text{Age} - 1.5 \times \text{Balance} + 60 \leq 0 \end{cases}$$

$$f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \dots$$

$$f(\mathbf{x}) = 60 + 1.0 \times \text{Age} - 1.5 \times \text{Balance}$$



Can use format $y=mx+b$ (m is the slope and b is the y intercept, when $x=0$) to describe the line as: $\text{Age}=(-1.5) \times \text{Balance}+60$, then we can:

- *Get classification function :*
- *and a general linear model*

But there are many different possible linear boundaries can separate the two groups of points, How to find out the BEST ONE, the Optimal Fitting one?

1-2 Optimizing an Objective Function

What to optimize ?

- *what should be our goal or objective in choosing the parameters?*
- *what weights should we choose?*

If choose by experience, to creating an objective function that matches the true goal of the data mining:

Several remarkably effective choices would be:

- *SVM - Support Vector Machine*
- *Linear Models for Regression*
- *Logistic Regression*

Note that logistic regression is not actually used for traditional regression tasks. Instead, logistic regression is utilized for estimating the probability of class membership in classification tasks, which makes it highly useful for many applications.

1-3 An Example of Mining a Linear Discriminant from Data

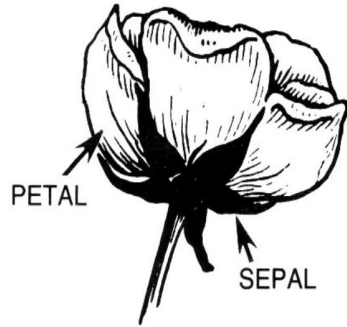




Figure 4-6. Two parts of a flower. Width measurements of these are used in the Iris dataset.

5.1	3.5	1.4	0.2	<i>I. setosa</i>	
4.9	3.0	1.4	0.2	<i>I. setosa</i>	
4.7	3.2	1.3	0.2	<i>I. setosa</i>	
4.6	3.1	1.5	0.2	<i>I. setosa</i>	
5.0	3.6	1.4	0.3	<i>I. setosa</i>	
5.4	3.9	1.7	0.4	<i>I. setosa</i>	
4.6	3.4	1.4	0.3	<i>I. setosa</i>	
5.0	3.4	1.5	0.2	<i>I. setosa</i>	
4.4	2.9	1.4	0.2	<i>I. setosa</i>	
4.9	3.1	1.5	0.1	<i>I. setosa</i>	
5.4	3.7	1.5	0.2	<i>I. setosa</i>	
4.8	3.4	1.6	0.2	<i>I. setosa</i>	
4.8	3.0	1.4	0.1	<i>I. setosa</i>	
4.3	3.0	1.1	0.1	<i>I. setosa</i>	
5.8	4.0	1.2	0.2	<i>I. setosa</i>	
5.7	4.4	1.5	0.4	<i>I. setosa</i>	

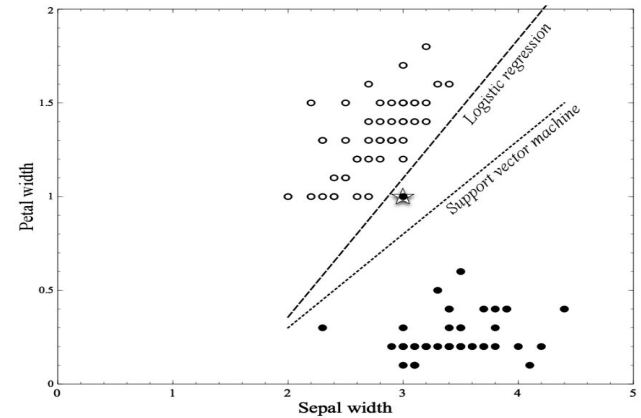


Figure 4-7. A dataset and two learned linear classifiers.

Problem: to classify each instance as belonging species based on the attributes.

- original Iris dataset includes 3 species x 4 attributes.
- here just use a simplified dataset, 2 species x 2 attributes.
- species: Setosa and Versicolor; attributes: Petal width and the Sepal width.

Which separator do you think is better?

1-4 Linear Discriminant Functions for Scoring and Ranking Instances

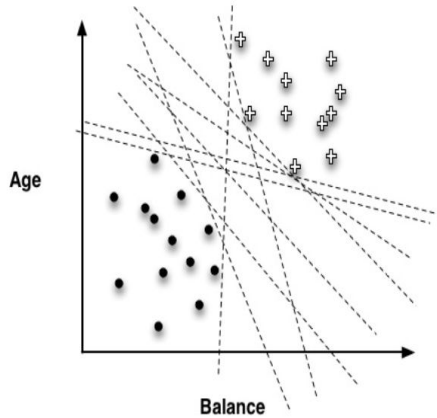


Figure 4-5. Many different possible linear boundaries can separate the two groups of points of Figure 4-4.

Except TI we can do this with Linear Models as well.
If we want to know: which consumers are most likely to respond to this offer? or, Which customers are most likely to leave when their contracts expire? (Not just separated into "yes" or "no")

Where would we be most certain that x would not respond? Where would we be most uncertain?
to the decision boundary: far away=certain; near=uncertain.

$$f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \dots$$

$f(x)=0$ when x is sitting on the decision boundary.
 $f(x)=\text{smaller}$ when x is near the boundary
 $f(x)=\text{larger}$ when x is far from the boundary

'smaller-0-larger' > 'margin'

1-5 Support Vector Machines, Briefly

(SVM) support vector machines are linear discriminants

Within the infinitude of different possible linear discriminants that would separate the classes. The simple, elegant idea of SVMs is: fit the fattest bar between the classes.

A wider bar is better. And once the widest bar is found, the linear discriminant will be the center line through the bar.

*The **distance between the dashed parallel lines** is called the **margin** around the linear discriminant, and thus the objective is to maximize.*

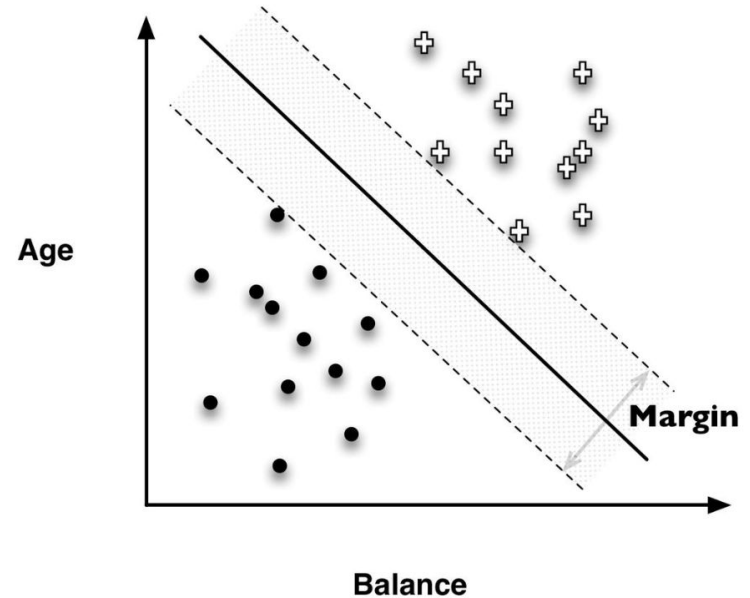


Figure 4-8. The points of Figure 4-2 and the maximal margin classifier.

1-5 Support Vector Machines, Briefly

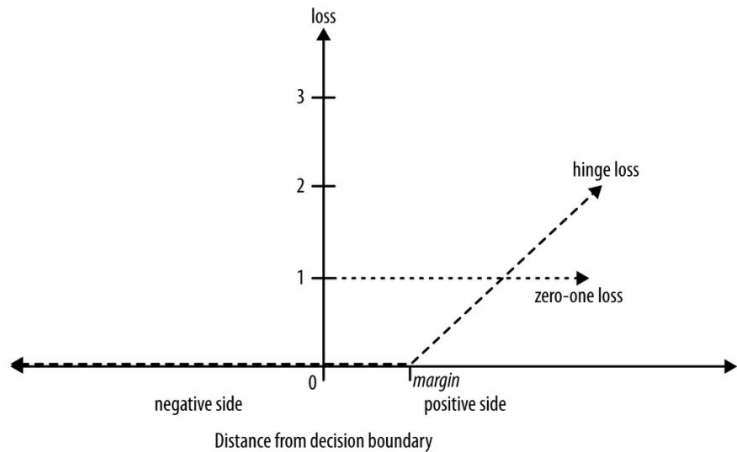


Figure 4-9. Two loss functions illustrated. The x axis shows the distance from the decision boundary. The y axis shows the loss incurred by a negative instance as a function of its distance from the decision boundary. (The case of a positive instance is symmetric.) If the negative instance is on the negative side of the boundary, there is no loss. If it is on the positive (wrong) side of the boundary, the different loss functions penalize it differently. (See Sidebar: Loss functions.)

No perfect choice, only better choice, and try to optimize it.

The margin-maximizing boundary gives the maximal lee-way for classifying such points.

In the objective function that measures how well a particular model fits the training points, we will simply penalize a training point for being on the wrong side of the decision boundary.

If the data are not linearly separable, the best fit is some balance between a fat margin and a low total error penalty.

SVM will make only “small” errors. Technically, this error function is known as hinge loss.

1-5 Support Vector Machines, Briefly

The Maxima margin classifiers are super sensitive to outliers in the training data and that makes them pretty lame. If we want to make it not so sensitive to outliers we must allow misclassifications.

Zero-one loss function: *assigns a loss of zero for a correct decision and one for an incorrect decision.*

SVM use Hinge loss function: *(because the loss graph looks like a hinge) Hinge loss incurs no penalty for an example that is not on the wrong side of the margin. The hinge loss only becomes positive when an example is on the wrong side of the boundary and beyond the margin.*

Loss then increases linearly with the example's distance from the margin, thereby penalizing points more the farther they are from the separating boundary.

*As to **Squared loss function**, usually use for numeric value prediction (regression), rather than classification. Using squared error in classification wrongly penalizes correctly classified points far from the decision boundary, which often misaligns with business objectives, leading to alternatives like Hinge-Adjusted squared error.*

2. Regression via Mathematical Functions

$$f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \dots$$

How far away are the estimated values from the true values on the training data? In other words, how big is the error of the fitted model? Presumably we'd like to minimize this error.

Least Squares Regression: The most convenient and popular linear regression procedure.

Choosing an objective function like least squares regression, which is data-sensitive and skewed by outliers, may not suit automated systems or applications with limited data-cleaning resources. Instead, a robust method like absolute error may be preferred, always considering the business context.

3. Class Probability Estimation and Logistic “Regression”

We often need to estimate the probability of an instance belonging to a class, factoring in decision-making elements like costs and benefits. In fraud detection used by banks, telecoms, and e-commerce, the goal is to focus on high-stakes cases, not just likely frauds, requiring actual fraud probability estimates.

If simply using our basic linear model to estimate the class probability.

The output of the linear function, $f(x)$, gives the distance from the separating boundary, the $f(x)$ will ranges from $-\infty$ to ∞ , however a **probability should range from zero to one.**

Probability	Corresponding odds
0.5	50:50 or 1
0.9	90:10 or 9
0.999	999:1 or 999
0.01	1:99 or 0.0101
0.001	1:999 or 0.001001

One very useful notion of the likelihood of an event is the **odds**. The odds of an event is the ratio of the probability of the event occurring to the probability of the event not occurring.

3. Class Probability Estimation and Logistic “Regression”

The distance from the boundary, ranging from $-\infty$ to ∞ , translates to odds ranging from 0 to ∞ . By taking the log-odds, which span from $-\infty$ to ∞ , we align with the distance measures, as shown in [Table ->].

This is exactly a logistic regression model: the same linear function $f(x)$ that we’ve examined throughout the chapter is used as a measure of the log-odds of the “event” of interest.

Probability	Odds	Log-odds
0.5	50:50 or 1	0
0.9	90:10 or 9	2.19
0.999	999:1 or 999	6.9
0.01	1:99 or 0.0101	-4.6
0.001	1:999 or 0.001001	-6.9

- *For probability estimation, logistic regression uses the same linear model as do our linear discriminants for classification and linear regression for estimating numerical target values.*
- *The output of the logistic regression model is interpreted as the log-odds of class membership.*
- *Log-odds from logistic regression directly indicate class membership probabilities, widely used to estimate default, response, fraud probabilities, and document relevance.*

3. Class Probability Estimation and Logistic “Regression”

$$\log \left(\frac{p_+(\mathbf{x})}{1 - p_+(\mathbf{x})} \right) = f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \dots$$

$$p_+(\mathbf{x}) = \frac{1}{1 + e^{-f(\mathbf{x})}}$$

Thus, Equation (left-top) specifies that for a particular data item, described by feature-vector x , the log-odds of the class is equal to our linear function, $f(x)$.

Since often we actually want the estimated probability of class membership, not the log-odds, we can solve $p_+(x)$ in Equation (left) This yields the quantity in Equation 4-4

Now we may use the $f(x)$ and estimated probability to draw the plot (left).

This curve is called a “sigmoid” curve because of its “S” shape, which squeezes the probabilities into their correct range (between zero and one)

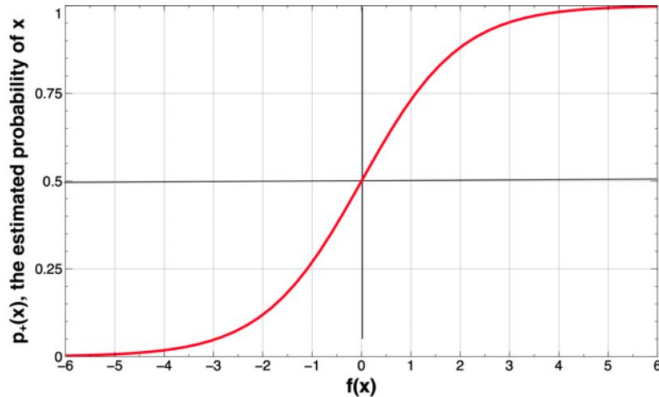


Figure 4-10. Logistic regression's estimate of class probability as a function of $f(x)$, (i.e., the distance from the separating boundary). This curve is called a “sigmoid” curve because of its “S” shape, which squeezes the probabilities into their correct range (between zero and one).

3. * Logistic Regression: Some Technical Details

The figure shows that at the decision boundary (at distance $x = 0$), the probability is 0.5 (a coin toss). The probability varies approximately linearly near to the decision boundary, but then approaches certainty farther away. Part of the “fitting” of the model to the data includes determining the slope of the almost-linear part, and thereby how quickly we are certain of the classes we move away from the boundary.

What is the objective function we use to fit the logistic regression model to the data?

$$g(\mathbf{x}, \mathbf{w}) = \begin{cases} p_+(\mathbf{x}) & \text{if } \mathbf{x} \text{ is a } + \\ 1 - p_+(\mathbf{x}) & \text{if } \mathbf{x} \text{ is a } \bullet \end{cases}$$

The g function gives the model’s estimated probability of seeing x ’s actual class given x ’s features.

The model (set of weights) that gives the highest sum is the model that gives the highest “likelihood” to the data—the “maximum likelihood” model. The maximum likelihood model “on average” gives the highest probabilities to the positive examples.

4.Example: Logistic Regression versus Tree Induction

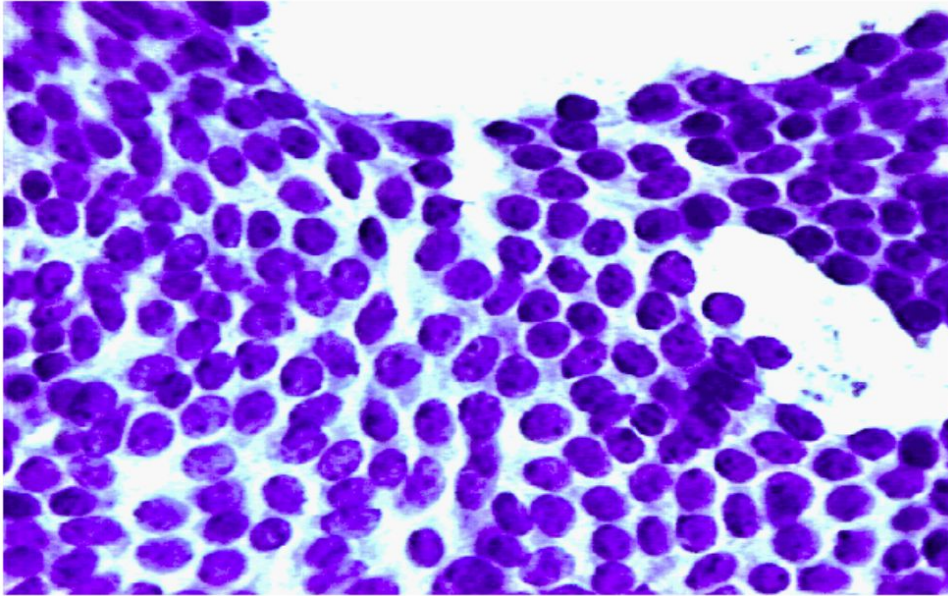
Two important differences between classification trees and linear classifiers.

- 1. A classification tree's boundaries are axis-perpendicular, as it selects one attribute at a time, unlike linear classifiers that use all attributes, allowing boundaries in any direction.*
- 2. A classification tree recursively segments the instance space finely using divide-and-conquer, while a linear classifier uses one decision surface for a single, two-segment division, due to its single equation utilizing all variables.*

Determining which characteristics best fit a dataset is challenging without knowing the optimal decision boundary, leading to practical implications from these differences.

In business, model comprehensibility varies among stakeholders; logistic regression is clear to statisticians but not to others, while a modest-sized decision tree is more universally understandable.

4.Example: Logistic Regression versus Tree Induction



*The **Wisconsin Breast Cancer Dataset**, this is popular dataset from the the machine learning dataset repository at the University of California at Irvine.*

Each example describes characteristics of a cell nuclei image, which has been labeled as either benign or malignant (cancerous), based on an expert's diagnosis of the cells. A sample cell image is shown in left Figure.

4.Example: Logistic Regression versus Tree Induction

Attribute name	Description
RADIUS	Mean of distances from center to points on the perimeter
TEXTURE	Standard deviation of grayscale values
PERIMETER	Perimeter of the mass
AREA	Area of the mass
SMOOTHNESS	Local variation in radius lengths
COMPACTNESS	Computed as: $\text{perimeter}^2 / \text{area} - 1.0$
CONCAVITY	Severity of concave portions of the contour
CONCAVE POINTS	Number of concave portions of the contour
SYMMETRY	A measure of the nuclei's symmetry
FRACTAL DIMENSION	'Coastline approximation' - 1.0
DIAGNOSIS (Target)	Diagnosis of cell sample: malignant or benign

These were “computed from a digitized image of a fine needle aspirate (FNA) of a breastmass. They describe characteristics of the cell nuclei present in the image.”

From each of these basic characteristics, three values were computed: the mean (`_mean`), standard error (`_SE`), and “worst” or largest (mean of the three largest values, `_worst`).

This resulted in 30 measured attributes in the dataset.

There are 357 benign images and 212 malignant images.

4.Example: Logistic Regression versus Tree Induction

Attribute	Weight (learned parameter)
SMOOTHNESS_worst	22.3
CONCAVE_mean	19.47
CONCAVE_worst	11.68
SYMMETRY_worst	4.99
CONCAVITY_worst	2.86
CONCAVITY_mean	2.34
RADIUS_worst	0.25
TEXTURE_worst	0.13
AREA_SE	0.06
TEXTURE_mean	0.03
TEXTURE_SE	-0.29
COMPACTNESS_mean	-7.1
COMPACTNESS_SE	-27.87
w_0 (intercept)	-17.7

*left-Table shows the linear model learned by logistic regression to predict **benign** versus **malignant** for this dataset. Specifically, it shows the nonzero weights ordered from highest to lowest.*

The model achieves 98.9% accuracy with six errors, while a 25-node classification tree reaches 99.1% accuracy, slightly outperforming logistic regression.

4.Example: Logistic Regression versus Tree Induction

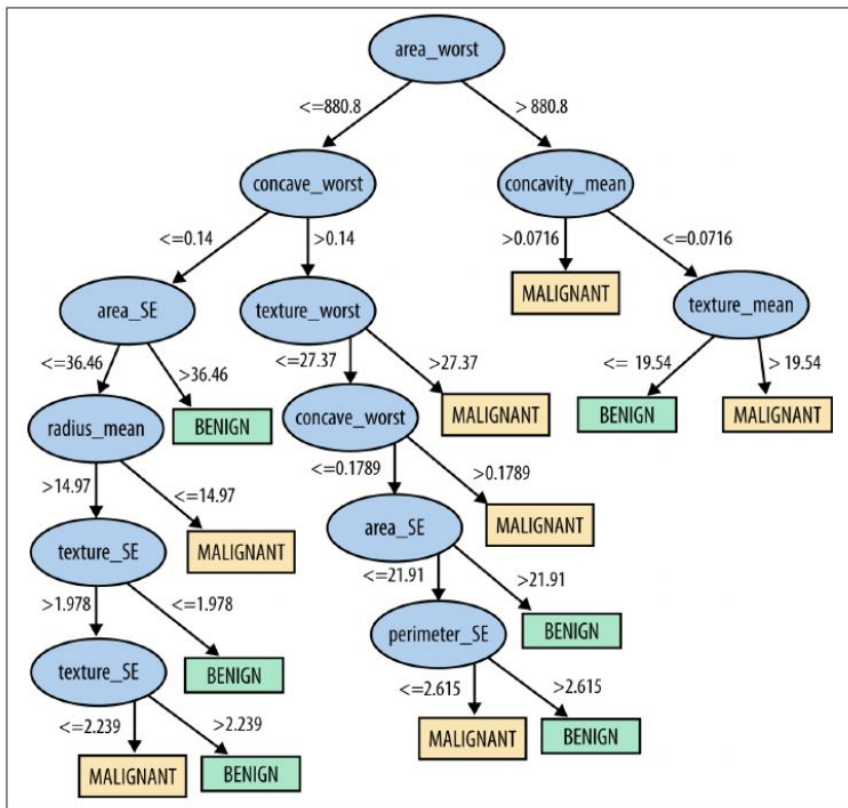


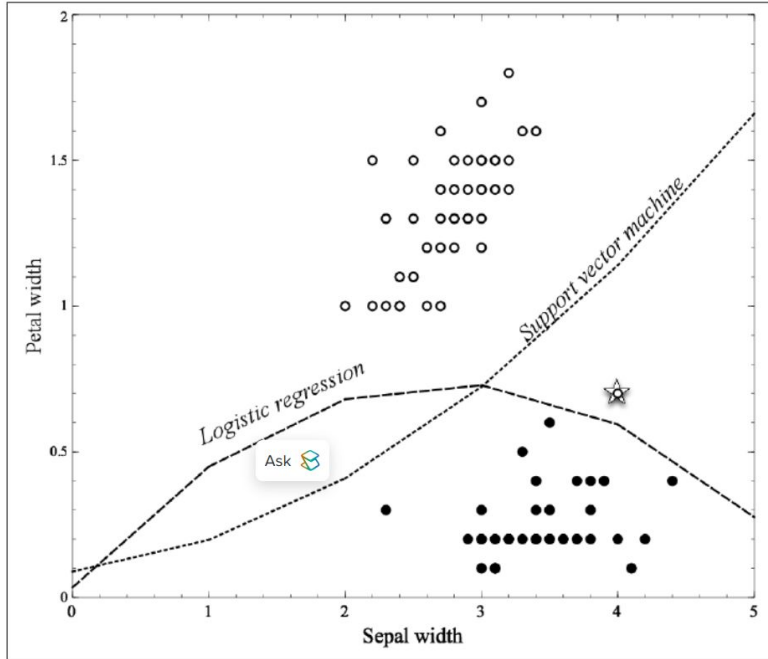
Figure 4-13. Decision tree learned from the Wisconsin Breast Cancer dataset.

Evaluating classifiers can be tricky, and although the tree is slightly more accurate, the difference is minimal, based on one less error in 569 examples.

How confident should we be in this evaluation?

The later content will discuss guidelines and pitfalls of model evaluation.

5. Nonlinear Functions, Support Vector Machines, and Neural Networks



The Iris dataset with a nonlinear feature. In this figure, logistic regression and support vector machine—both linear models—are provided an additional feature, Sepal width^2 , which allows both the freedom to create more complex, nonlinear models (boundaries), as shown.

The core idea extends beyond fitting linear functions, allowing for complex numeric functions like nonlinear support-vector machines and neural networks, where parameters are fitted to data.

Nonlinear support vector machines use a **kernel function** to map features into a new space, where a linear model is fit, similar to adding complex terms in our example. This includes polynomial kernels for higher-order feature combinations, with data scientists exploring various kernel options.

5. Nonlinear Functions, Support Vector Machines, and Neural Networks

Neural networks, based on this chapter's concepts, are like model "stacks." The bottom layer uses original features for simple models, like logistic regressions, and each layer above applies a model to the previous layer's outputs. In a two-layer network, the second layer's logistic regression uses the first layer's outputs, akin to weighing opinions from "experts" on the problem.

Neural networks learn lower-layer logistic regressions, or "experts," without specific targets, unlike stacked models. They're trained as part of a big function whose parameters are optimized to fit the training data. This process simultaneously determines the best "experts" and how to combine them.

Increasing a model's flexibility to fit data can lead to overfitting, where it captures noise rather than general patterns applicable to new data. This concern, critical in data science, applies broadly, not just to neural networks, and is the focus of the next chapter.

Increasing a model's flexibility to fit data can lead to overfitting, where it captures noise rather than general patterns applicable to new data. This concern, critical in data science, applies broadly, not just to neural networks.

6. Summary

*This chapter presents **function fitting** or **parametric modeling**, where models are equations with undefined parameters, and data mining finds the "best" parameter set.*

***Function fitting techniques often use a linear model, weighting attribute values.** The difference between techniques like **SVMs**, **logistic regression**, and **linear regression** lies in their definition of 'best fit', determined by varying objective functions, leading to distinct methods.*

*We've explored **tree induction** and **function fitting** and compared them, assessing models on predictive performance and intelligibility. Building various models from data is beneficial for insight.*

If you look hard enough, you will find structure in a dataset, even if it's just there by chance.

*This chapter discusses optimizing model fit to data, which can result in **overfitting—finding patterns by chance**. The next chapter is dedicated to recognizing and avoiding overfitting, a key issue in data science.*

What I learned:

- **Linear regression** is a technique that tries to find the best linear relationship between a dependent variable and one or more independent variables.
- **Logistic regression** is a technique that tries to find the best logistic function that can separate the data into two or more classes.
- **Support vector machines** are a technique that tries to find the best hyperplane or curve that can separate the data into two or more classes with the maximum margin.
- **Neural networks** are a technique that are modeled after the human brain, and consist of a network of interconnected nodes, or neurons, that can learn from data and make predictions.