Machine Learning

Mohamad Ghassany

2016-12-14

Contents

Welcome

Welcome to this course. It is only a little introduction to Machine Learning.

The aim of Machine Learning is to build computer systems that can adapt to their environments and learn form experience. Learning techniques and methods from this field are successfully applied to a variety of learning tasks in a broad range of areas, including, for example, spam recognition, text classification, gene discovery, financial forecasting. The course will give an overview of many concepts, techniques, and algorithms in machine learning, beginning with topics such as linear regression and classification and ending up with topics such as kmeans and Expectation Maximization. The course will give the student the basic ideas and intuition behind these methods, as well as a more formal statistical and computational understanding. Students will have an opportunity to experiment with machine learning techniques in R and apply them to a selected problem.

Introduction

What is Machine Learning ?

What is Machine Learning?

Two definitions of Machine Learning are offered. Arthur Samuel described it as: “the field of study that gives computers the ability to learn without being explicitly programmed.” This is an older, informal definition.

Tom Mitchell provides a more modern definition: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

Machine Learning is also called Statistical Learning.

Example: playing checkers.

E = the experience of playing many games of checkers

T = the task of playing checkers.

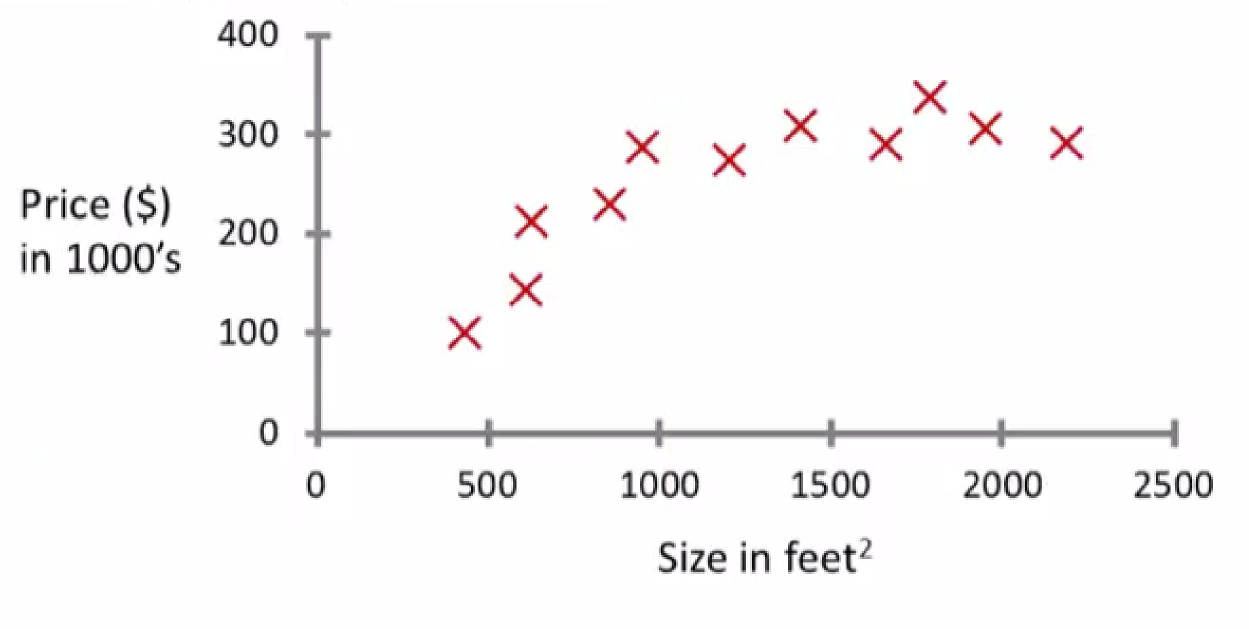
P = the probability that the program will win the next game.

In general, any machine learning problem can be assigned to one of two broad classifications:

Supervised learning and Unsupervised learning.

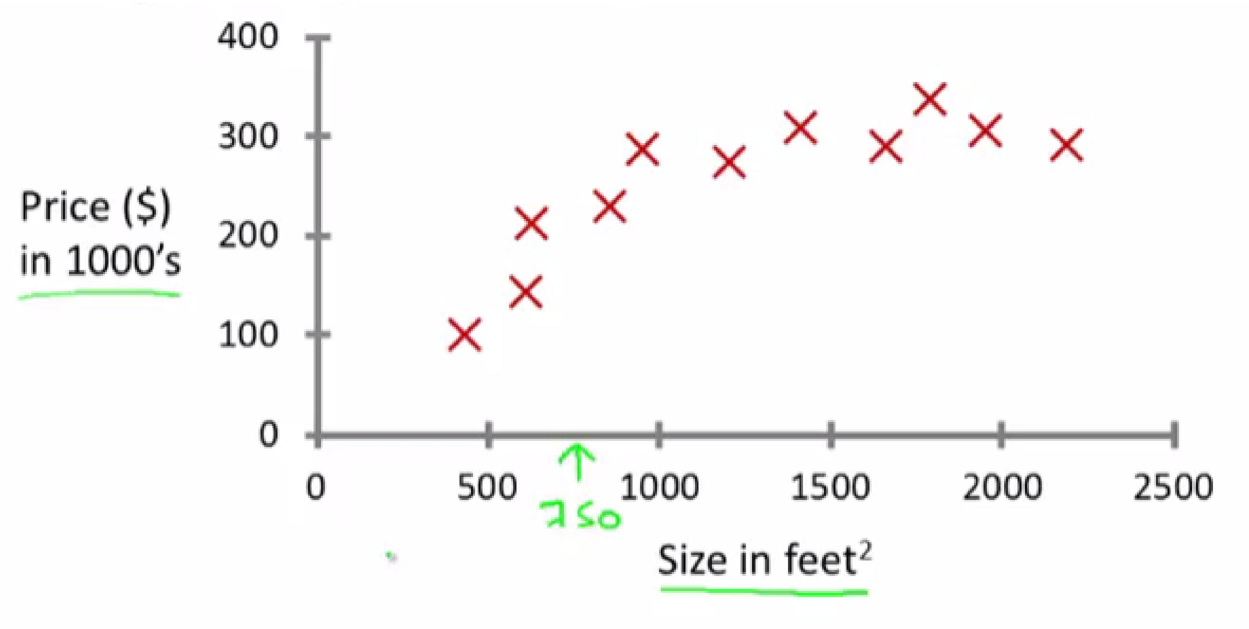
Supervised Learning

Supervised Learning is probably the most common type of machine learning problem. Let’s start with an example of what is it. Let’s say we want to predict housing prices. We plot a data set and it looks like this.



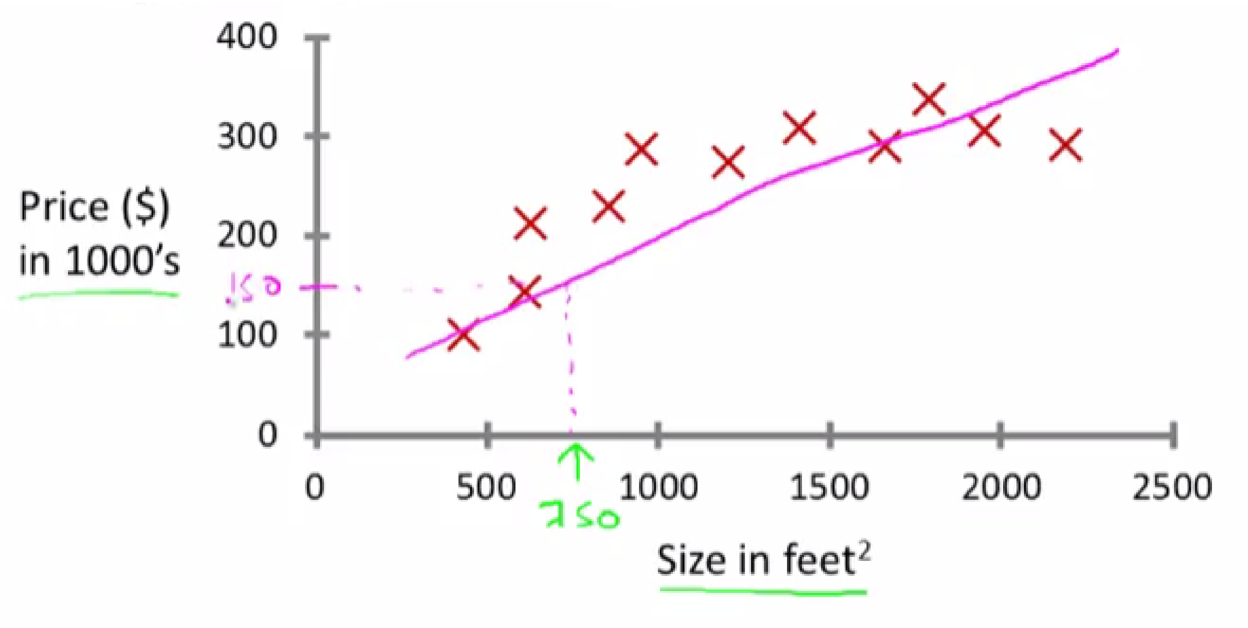
Here on the horizontal axis, the size of different houses in square feet, and on the vertical axis, the price of different houses in thousands of dollars.

So. Given this data, let's say we own a house that is, say 750 square feet and hoping to sell the house and they want to know how much they can get for the house.

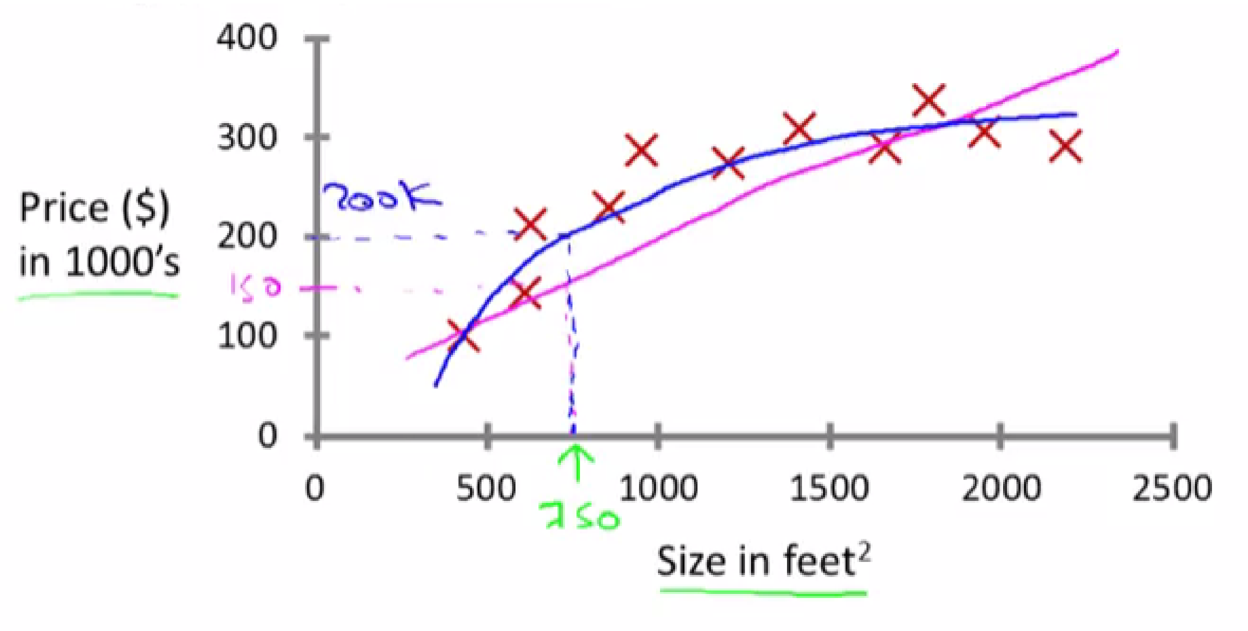


So how can the learning algorithm help?

One thing a learning algorithm might be able to do is put a straight line through the data or to **"fit"** a straight line to the data and, based on that, it looks like maybe the house can be sold for maybe about $150,000.



But maybe this isn't the only learning algorithm we can use. There might be a better one. For example, instead of sending a straight line to the data, we might decide that it's better to fit a *quadratic function* or a *second-order polynomial* to this data.



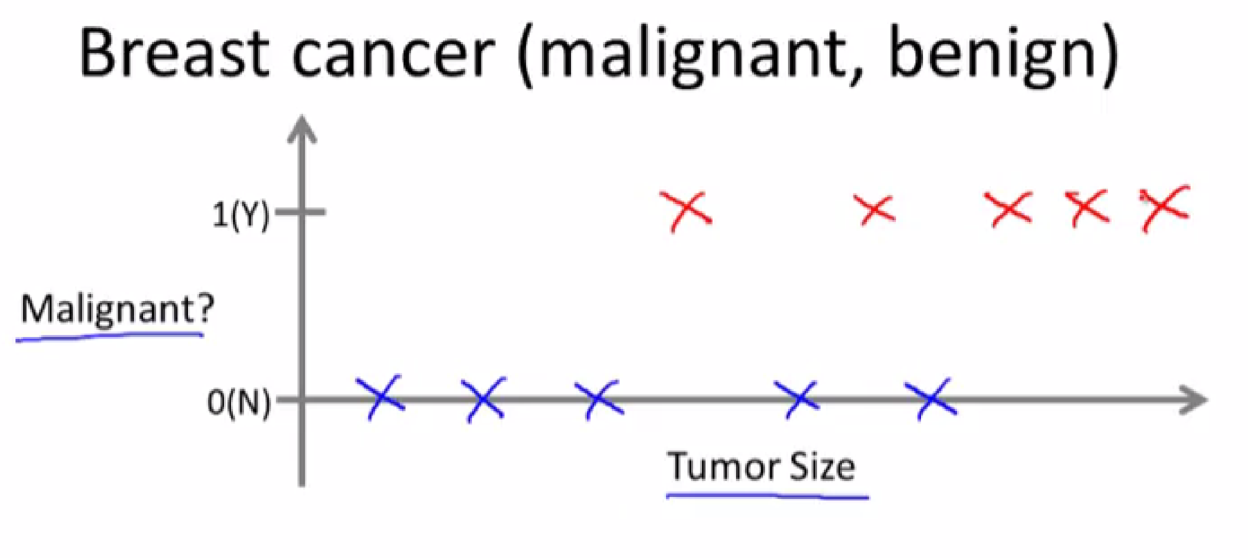
If we do that, and make a prediction here, then it looks like, well, maybe we can sell the house for closer to $200,000.

This is an example of a supervised learning algorithm.

The term supervised learning refers to the fact that we gave the algorithm a data set in which the **"right answers"** were given.

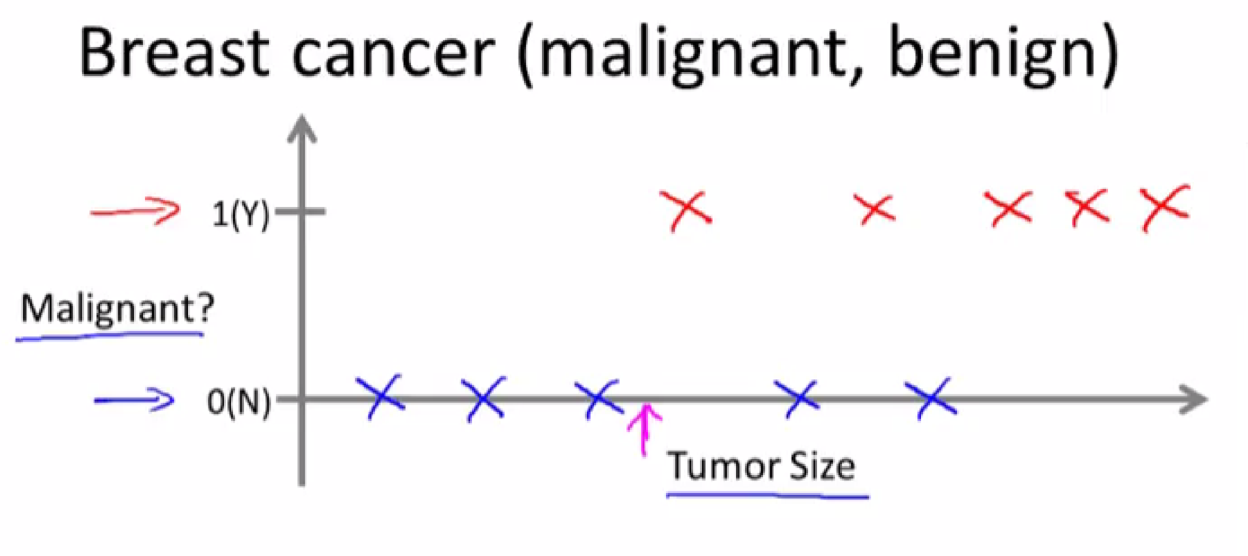
The example above is also called a regression problem. A regression problem is when we try to predict a **continuous** value output. Namely the price in the example.

Here's another supervised learning example. Let's say we want to look at medical records and try to predict of a breast cancer as malignant or benign. If someone discovers a breast tumor, a lump in their breast, a malignant tumor is a tumor that is harmful and dangerous and a benign tumor is a tumor that is harmless. Let's see a collected data set and suppose in the data set we have the size of the tumor on the horizontal axis and on the vertical axis we plot one or zero, yes or no, whether or not these are examples of tumors we've seen before are malignant (which is one) or zero if not malignant or benign.



In this data set we have five examples of benign tumors, and five examples of malignant tumors.

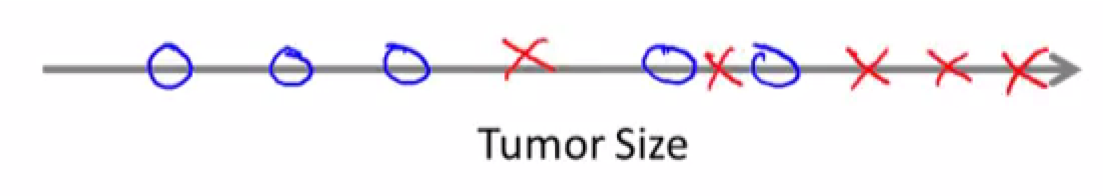
Let's say a person who tragically has a breast tumor, and let's say her breast tumor size is known (rose arrow in the following figure).



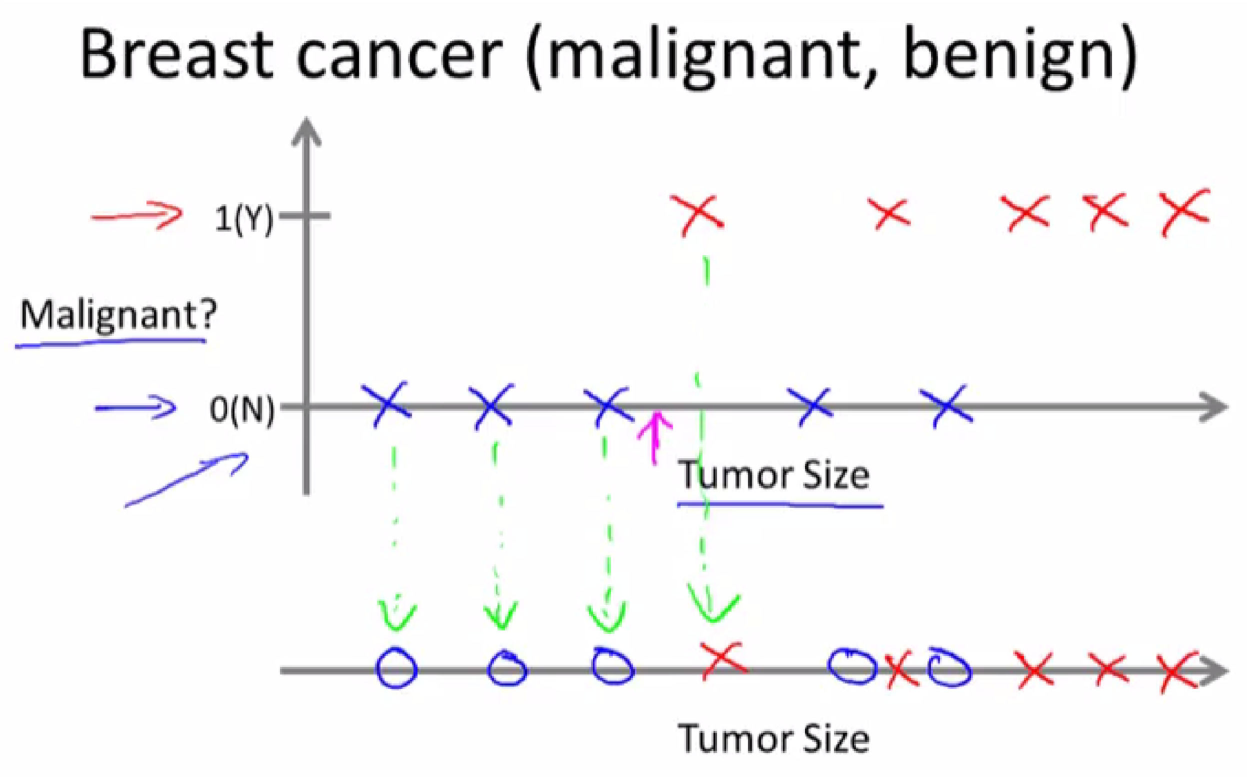
The machine learning question is, can you estimate what is the probability that a tumor is malignant versus benign? To introduce a bit more terminology this is an example of a classification problem.

The term classification refers to the fact that here we're trying to predict a **discrete** value output: zero or one, malignant or benign. And it turns out that in classification problems sometimes you can have more than two values for the two possible values for the output.

In classification problems there is another way to plot this data. Let's use a slightly different set of symbols to plot this data. So if tumor size is going to be the attribute that we are going to use to predict malignancy or benignness, we can also draw the data like this.



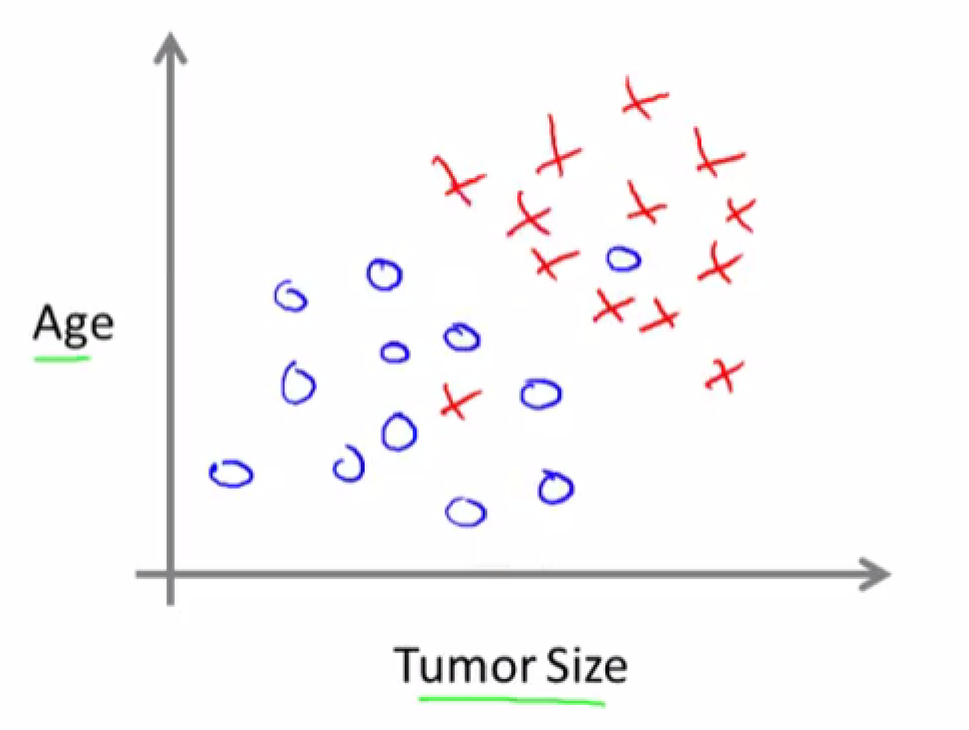
All we did was we took the data set on top and just mapped it down using different symbols. So instead of drawing crosses, we are now going to draw `O`'s for the benign tumors.



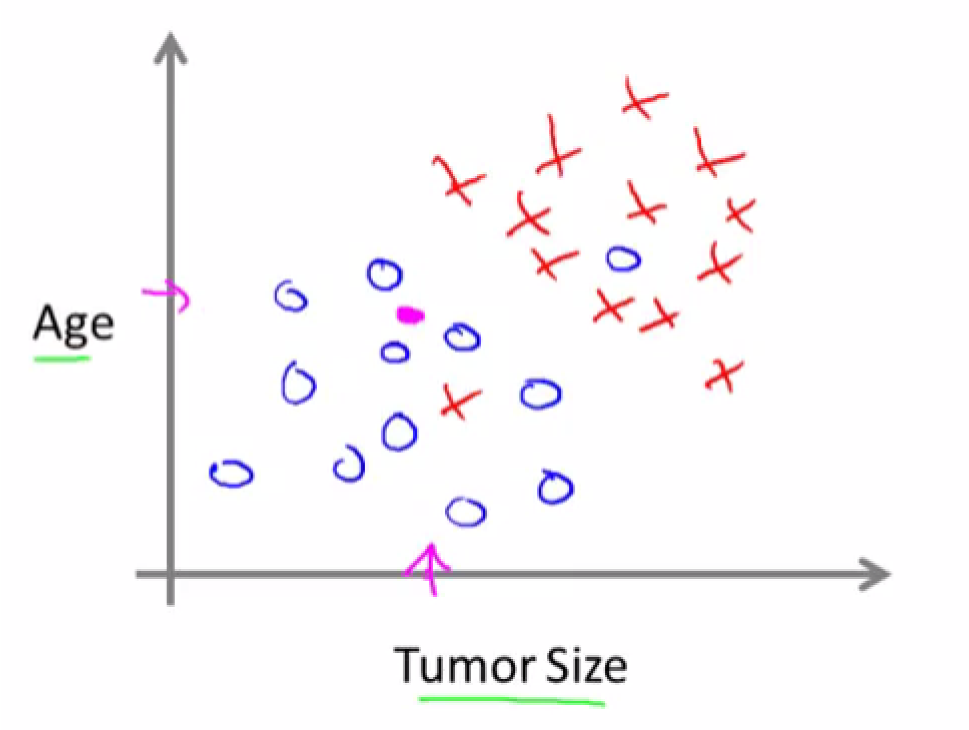
Now, in this example we use only one **feature** or one attribute, mainly, the *tumor size* in order to predict whether the tumor is malignant or benign.

In other machine learning problems we may have more than one feature.

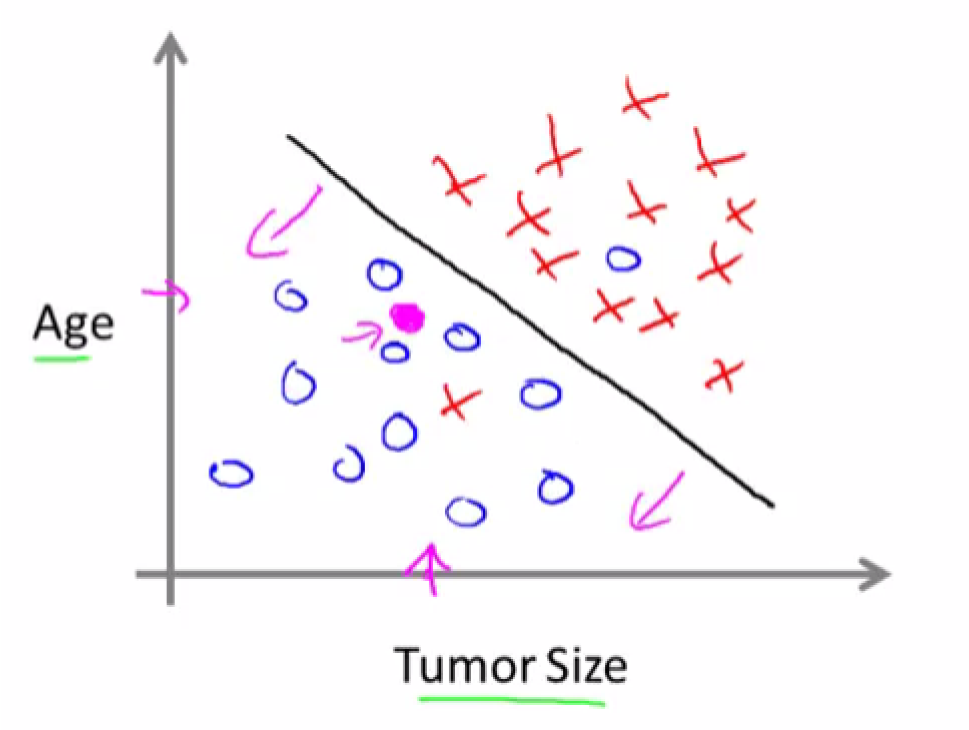
Here's an example. Let's say that instead of just knowing the tumor size, we know both the age of the patients and the tumor size. In that case maybe the data set will look like this.



So, let's say a person who tragically has a tumor. And maybe, their tumor size and age falls around there (rose point):



So given a data set like this, what the learning algorithm might do is throw a straight line through the data to try to separate out the malignant tumors from the benign ones. And with this, hopefully we can decide that the person's tumor falls on this benign side and is therefore more likely to be benign than malignant.



In this example we had **two features**, namely, the age of the patient and the size of the tumor. In other machine learning problems we will often have more features.

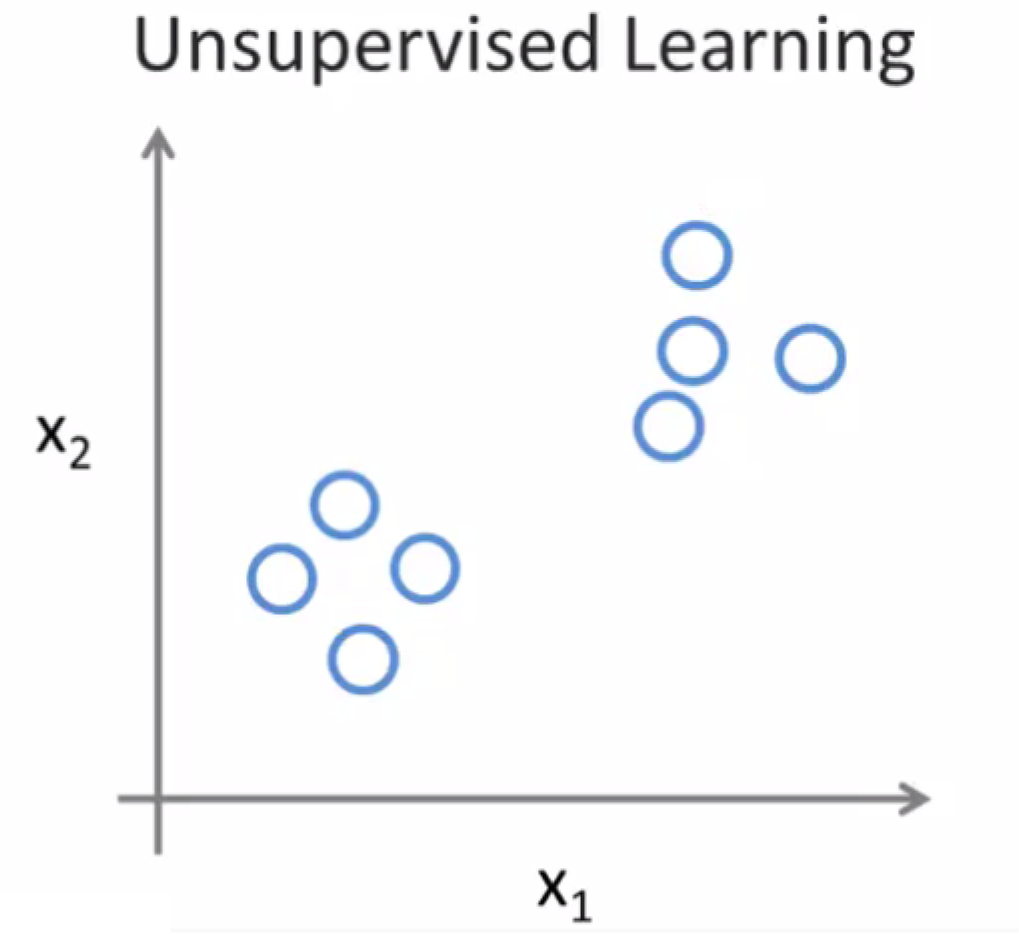
Most interesting learning algorithms is a learning algorithm that can deal with, not just two or three or five features, but an **infinite number of features**. So how do you deal with an infinite number of features. How do you even store an infinite number of things on the computer when your computer is gonna run out of memory.

Unsupervised Learning

The second major type of machine learning problem is called Unsupervised Learning.

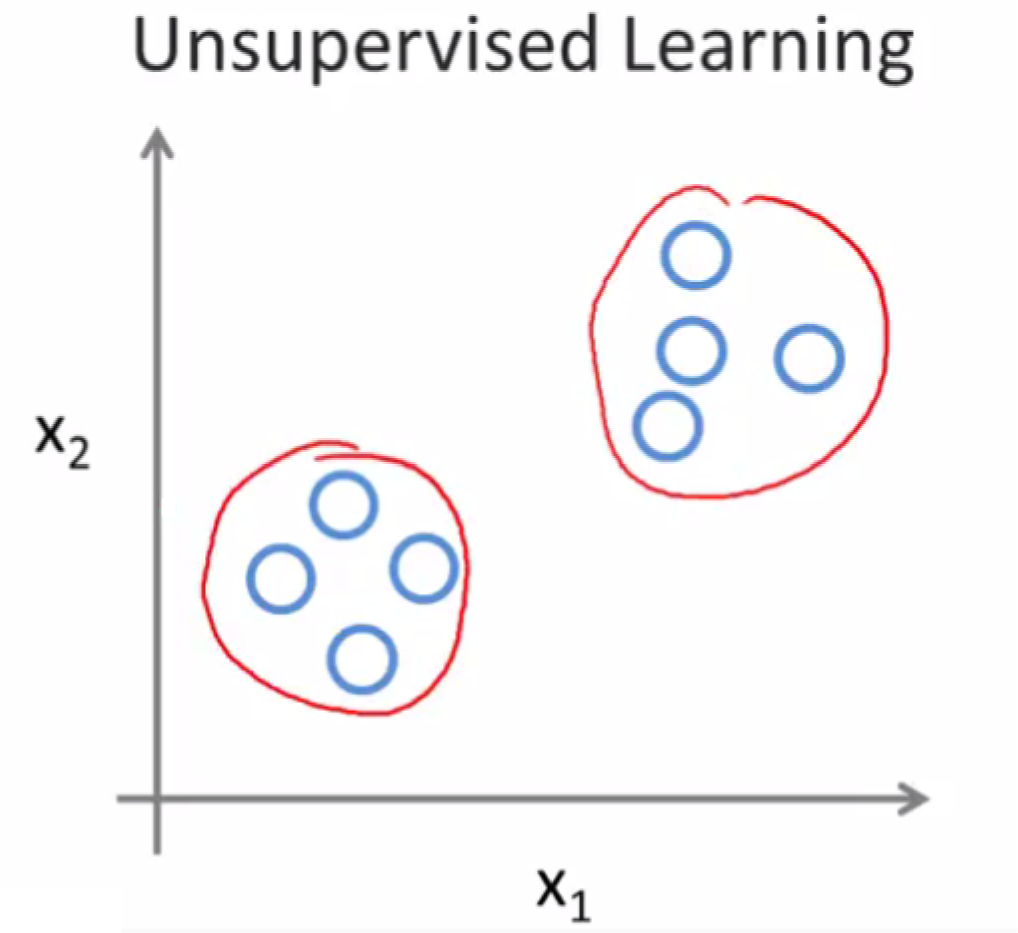
The difference between Unsupervised Learning and Supervised Learning is that in Supervised Learning we are told explicitly what is the so-called right answers (data are labeled).

In Unsupervised Learning, we're given data that doesn't have any labels or that all has the same label or really no labels. Like in this example:



So we're given the data set and we're not told what to do with it and we're not told what each data point is. Instead we're just told, here is a data set. Can you find some structure in the data?

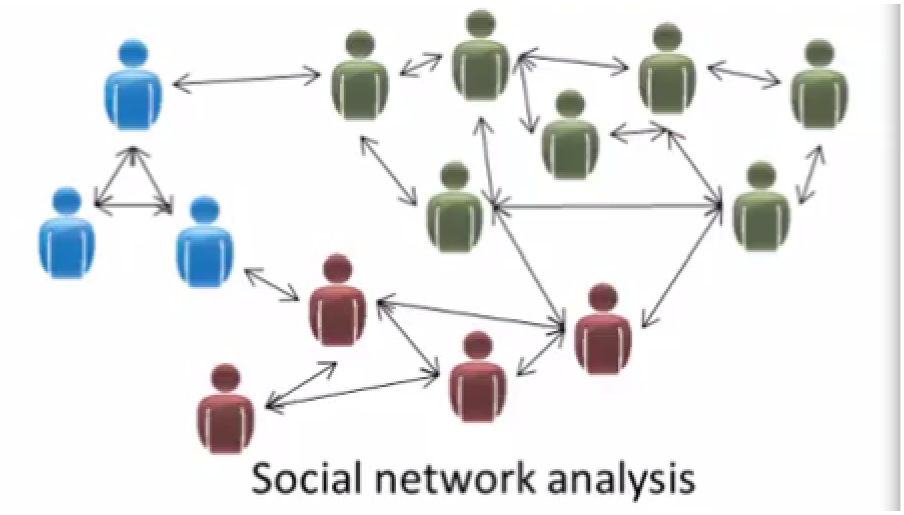
Given this data set, an Unsupervised Learning algorithm might decide that the data lives in two different clusters.



This is called a **clustering** algorithm.

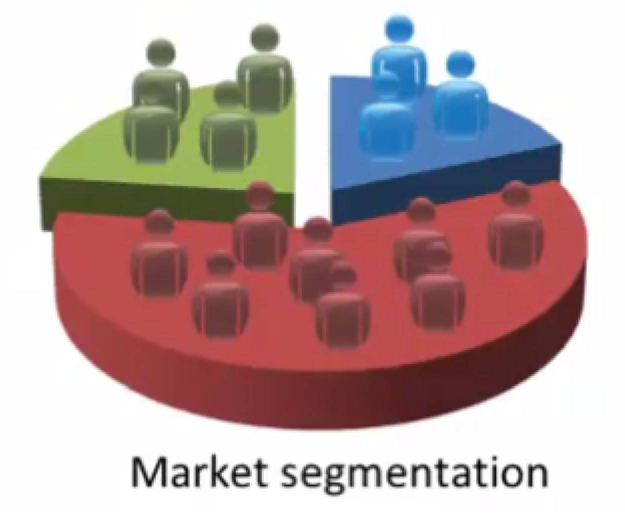
Here are two examples where Unsupervised Learning or clustering is used.

Social network analysis:



So given knowledge about which friends you email the most or given your Facebook friends or your Google+ circles, can we automatically identify which are cohesive groups of friends, also which are groups of people that all know each other?

Market segmentation:



Many companies have huge databases of customer information. So, can you look at this customer data set and automatically discover market segments and automatically group your customers into different market segments so that you can automatically and more efficiently sell or market your different market segments together?

This is Unsupervised Learning because we have all this customer data, but we don't know in advance what are the market segments and for the customers in our data set, we don't know in advance who is in market segment one, who is in market segment two, and so on. But we have to let the algorithm discover all this just from the data.

Part I

Supervised Learning

Part II

Regression

Chapter 1

Linear Regression

1.1 Notation

In general, we will let x_{ij} represent the value of the jth variable for the ith observation, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$. We will use i to index the samples or observations (from 1 tp n) and j will be used to index the variables (or features) (from 1 to p). We let $\mathbf{X}$ denote a $n \times p$ matrix whose (i, j)th element is x_{ij}. That is,

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \ldots & x_{1p} \\ x_{21} & x_{22} & x_{23} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \ldots & x_{np} \end{pmatrix}$$

Note that it is useful to visualize $\mathbf{X}$ as a spreadsheet of numbers with n rows and p columns. We will write the rows of $\mathbf{X}$ as $x_1, x_2, \ldots, x_n$. Here x_i is a vector of length p, containing the p variable measurements for the ith observation. That is,

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix}$$

(Vectors are by default represented as columns.)

We will write the columns of $\mathbf{X}$ as $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$. Each is a vector of length n. That is,

$$\mathbf{x}_j = \begin{pmatrix} \mathbf{x}_{1j} \\ \mathbf{x}_{2j} \\ \vdots \\ \mathbf{x}_{nj} \end{pmatrix}$$

Using this notation, the matrix $\mathbf{X}$ can be written as

$$\mathbf{X} = (\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_p)$$

or

$$\mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix}$$

The ^{T} notation denotes the transpose of a matrix or vector.

We use y_i to denote the ith observation of the variable on which we wish to make predictions. We write the set of all n observations in vector form as

$$\mathbf{y} = \begin{pmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{pmatrix}$$

Then the observed data consists of $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where each x_i is a vector of length p. (If $p = 1$, then x_i is simply a scalar).

1.2 Model Representation

Let's consider the example about predicting housing prices. We're going to use this data set as an example,

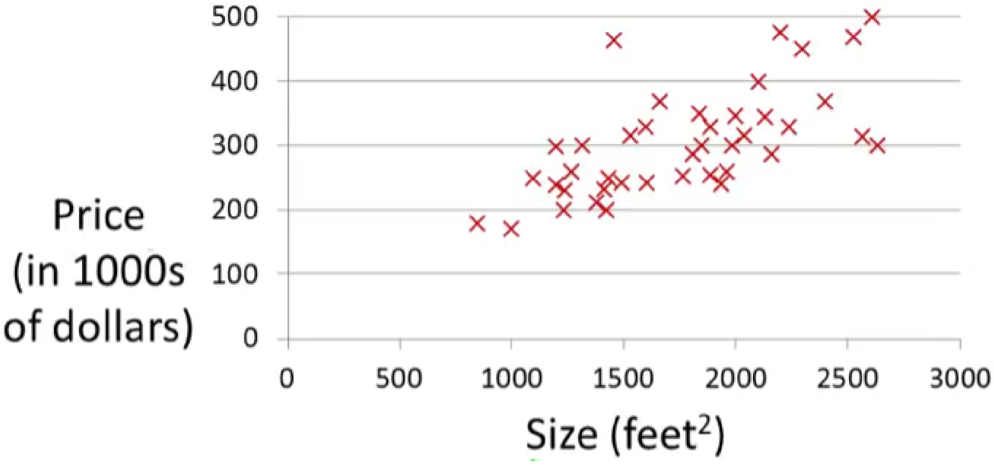


Figure 1.1:

Suppose that there is a person trying to sell a house of size 1250 square feet and he wants to know how much he might be able to sell the house for. One thing we could do is fit a model. Maybe fit a straight line to this data. Looks something like this,

and based on that, maybe he can sell the house for around $220,000. Recall that this is an example of a supervised learning algorithm. And it's supervised learning because we're given the "right answer" for each of our examples. More precisely, this is an example of a regression problem where the term regression refers to the fact that we are predicting a real-valued output namely the price.

More formally, in supervised learning, we have a data set and this data set is called a **training set**. So for housing prices example, we have a training set of different housing prices and our job is to learn from this data how to predict prices of the houses.

Let's define some notation from this data set:

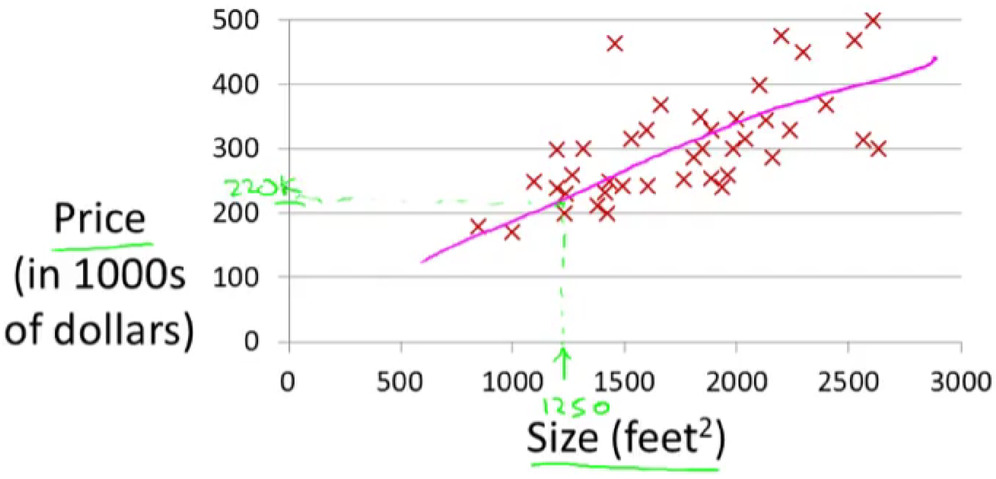
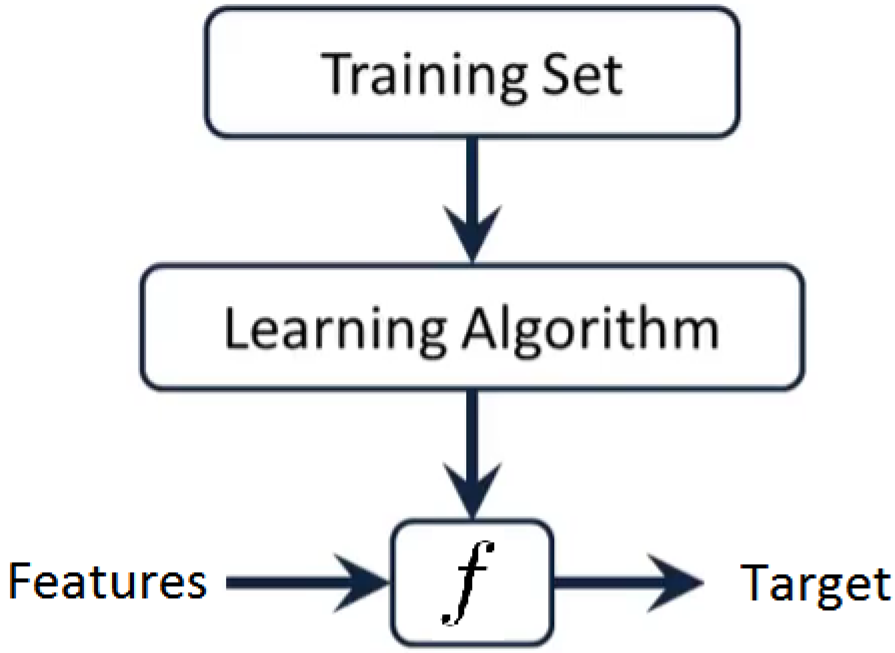


Figure 1.2:

- The size of the house is the input variable.
- The house price is the output variable.
- The input variables are typically denoted using the variable symbol X,
- The inputs go by different names, such as *predictors*, *independent variables*, *features*, *predictor* or sometimes just *variables*.
- The output variable is often called the *response*, *dependent variable* or *target*, and is typically denoted using the symbol Y.
- (x_i, y_i) is the ith training example.
- The set of $\{(x_i, y_i)\}$ is the training set.
- n is the number of training examples.

So here's how this supervised learning algorithm works. Suppose that we observe a quantitative response Y and p different predictors, $X_1, X_2, \ldots, X_p$. We assume that there is some relationship between Y and $X = (X_1, X_2, \ldots, X_p)$, which can be written in the very general form

$$Y = f(X) + \epsilon$$



Here f is some fixed but unknown function of $X_1, X_2, \ldots, X_p$, and ϵ is a random error term, which is independent of X and has mean zero. The f function is also called *hypothesis* in Machine Learning. In general, the function f may involve more than one input variable. In essence, Supervised Learning refers to a set of approaches for estimating f.

1.3 Why Estimate f ?

There are two main reasons that we may wish to estimate f: *prediction* and *inference.*

Prediction

In many situations, a set of inputs X are readily available, but the output Y cannot be easily obtained. In this setting, since the error term averages to zero, we can predict Y using

$$\hat{Y} = \hat{f}(X)$$

where $\hat{f}$ represents our estimate for f, and $\hat{Y}$ represents the resulting prediction for Y. Like in the example above about predicting housing prices.

We can measure the accuracy of $\hat{Y}$ by using a **cost function**. In the regression models, the most commonly-used measure is the *mean squared error* (MSE), given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

Inference

We are often interested in understanding the way that Y is affected as $X_1, X_2, \ldots, X_p$ change. In this situation we wish to estimate f , but our goal is not necessarily to make predictions for Y. We instead want to understand the relationship between X and Y, or more specifically, to understand how Y changes as a function of $X_1, X_2, \ldots, X_p$. In this case, one may be interested in answering the following questions:

- Which predictors are associated with the response?
- What is the relationship between the response and each predictor?
- Can the relationship between Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?

1.4 Simple Linear Regression

1.4.1 Model

Simple linear regression is a very straightforward approach for predicting a quantitative response Y on the basis of a single predictor variable X. It assumes that there is approximately a linear relationship between X and Y. Mathematically, we can write this linear relationship as

$$Y = \beta_0 + \beta_1 X + \epsilon$$
$$Y \approx \beta_0 + \beta_1 X$$

where β_0 and β_1 are two unknown constants that represent the *intercept* and *slope*, also known as ***coefficients*** or *parameters*, and ϵ is the error term.

Given some estimates $\hat{\beta_0}$ and $\hat{\beta_1}$ for the model coefficients, we predict future inputs x using

$$\hat{y} = \hat{\beta_0} + \hat{\beta_1} x$$

where $\hat{y}$ indicates a prediction of Y on the basis of $X = x$. The *hat* symbol, $\hat{}$, denotes an estimated value.

1.4.2 Estimating the Coefficients

Let $\hat{y_i} = \hat{\beta_0} + \hat{\beta_1} x_i$ be the prediction for Y based on the ith value of X. Then $e_i = y_i - \hat{y_i}$ represents the ith ***residual***.

We define the ***residual sum of squares*** (**RSS**) as

$$\begin{aligned} RSS &= e_1^2 + e_2^2 + \ldots + e_n^2 \\ &= \sum_{i=1}^{n} e_i^2 \end{aligned}$$

or equivantly as

$$\begin{aligned} RSS &= (y_1 - \hat{\beta_0} - \hat{\beta_1} x_1)^2 + (y_2 - \hat{\beta_0} - \hat{\beta_1} x_2)^2 + \ldots + (y_n - \hat{\beta_0} - \hat{\beta_1} x_n)^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{\beta_0} - \hat{\beta_1} x_i)^2 \end{aligned}$$

The *least squares* approach chooses $\hat{\beta_0}$ and $\hat{\beta_1}$ to minimize the RSS. The minimizing values can be show to be

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

and

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ are the sample means.

1.4.3 Assessing the Accuracy of the Coefficient Estimates

The standard error of an estimator reflects how it varies under repeated sampling. We have

$$\text{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\text{SE}(\hat{\beta}_0)^2 = \sigma^2\left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]$$

where $\sigma^2 = Var(\epsilon)$

In general, σ^2 is know known, but can be estimated from the data. The estimate of σ is known as the *residual standard error*, and is given by

$$\text{RSE} = \sqrt{\frac{\text{RSS}}{(n-2)}}$$

These standard errors can be used to compute *confidence intervals.* A 95% confidence interval is defined as a range of values such that with 95% probability, the range will contain the true unknown value of the parameter. It has the form

$$\hat{\beta}_1 \pm 2 \cdot \text{SE}(\hat{\beta}_1)$$

That is, there is approximately a 95% chance that the interval

$$\left[\hat{\beta}_1 - 2 \cdot \text{SE}(\hat{\beta}_1), \hat{\beta}_1 + 2 \cdot \text{SE}(\hat{\beta}_1)\right]$$

will contain the true value of β_1. Similarly, a confidence interval for β_0 approximately takes the form

$$\hat{\beta}_0 \pm 2 \cdot \text{SE}(\hat{\beta}_0)$$

Hypothesis testing

Standard errors can also be used to perform *hypothesis tests* on the coefficients. The most common hypothesis test involves testing the *null hypothesis* of

$$H_0 : \text{There is no relationship between}\, X \,\text{and}\, Y$$

versus the *alternative hypothesis*

$$H_1 : \text{There is some relationship between} \, X \, \text{and} \, Y$$

Mathematically, this corresponds to testing

$$H_0 : \beta_1 = 0$$

versus

$$H_1 : \beta_1 \neq 0$$

since if $\beta_1 = 0$ then the simple linear regression model reduces to $Y = \beta_0 + \epsilon$, and X is not associated with Y.

To test the null hypothesis H_0, we compute a ***t-statistic***, given by

$$t = \frac{\hat{\beta}_1 - 0}{\text{SE}(\hat{\beta}_1)}$$

This will have a t-distribution (*Student*) with $n - 2$ degrees of freedom, assuming $\beta_1 = 0$.

Using statistical software, it is easy to compute the probability of observing any value equal to $|t|$ or larger. We call this probability the ***p-value***.

If p-value is small enough (typically under 0.01 (1% error) or 0.05 (5% error)) we reject the null hypothesis, that is we declare a relationship to exist between X and Y.

1.4.4 Assessing the Accuracy of the Model

Once we have rejected the null hypothesis in favor of the alternative hypothesis, it is natural to want to quantify the extent to which the model fits the data. The quality of a linear regression fit is typically assessed using two related quantities: the *residual standard error* (RSE) and the R^2 statistic.

Residual Standard Error (RSE)

The Residual Standard Error (RSE) is computed using the formula

$$\text{RSE} = \sqrt{\frac{\text{RSS}}{(n-2)}} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{(n-2)}}$$

recall that RSS is the *residual sum-of-squares*.

The RSE is considered a measure of the lack of fit of the model to the data. If the predictions obtained using the model are very close to the true outcome values (if $\hat{y}_i = y_i$ for $i = 1, \ldots, n$) then RSE will be small, and we can conclude that the model fits the data very well. On the other hand, if $\hat{y}_i$ is very far from y_i for one or more observations, then the RSE may be quite large, indicating that the model doesn't fit the data well.

R^2 Statistic

To calculate R^2, we use the formula

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

where $\text{TSS} = \sum(y_i - \bar{y})^2$ is the *total sum of squared.*

R^2 measures the *proportion of variability in Y that can be explained using X.* An R^2 statistic that is close to 1 indicates that a large proportion of the variability in the response has been explained by the regression. A number near 0 indicates that the regression did not explain much of the variability in the response; this might occur because the linear model is wrong, or the inherent error σ^2 is high, or both.

It can be shown that in this simple linear linear regression setting that $R^2 = r^2$, where r is the correlation between X and Y:

$$r = \frac{cov(X, Y)}{\sigma_X \sigma_Y}$$

PW 1

How to use R

Rstudio

intro to r from ~macro **NO** intro R Julien jacques **Maybe?** or intro to R from tibshirani's book **Maybe?** A (very) short introduction to R **Maybe?**

Data preprocessing from udemy

Markdown Rmarkdown Html notebook

Get familiar with R

Notions

Data preprocessing from udemy

Linear Regression

Application on the salary example from udemy or Abass el sharif ready pdf

Chapter 2

Multiple Linear Regression

Simple linear regressionis a useful approach for predicting a response on the basis of a single predictor variable. However, in practice we often have more than one predictor. In the previous chapter, we took for example the prediction of housing prices considering we had the size of each house. We had a single feature X, the size of the house. But now imagine if we had not only the size of the house as a feature but we also knew the number of bedrooms, the number of flours and the age of the home in years. It seems like this would give us a lot more information with which to predict the price.

2.1 The Model

In general, suppose that we have p distinct predictors. Then the multiple linear regression model takes the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

where X_j represents the jth predictor and β_j quantifies the association between that variable and the response. We interpret β_j as the average effect on Y of a one unit increase in X_j, *holding all other predictors fixed.*

In matrix terms, supposing we have n observations and p variables, we need to define the following matrices:

$$\mathbf{Y}_{n\times 1} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \qquad \mathbf{X}_{n\times(p+1)} = \begin{pmatrix} 1 & X_{11} & X_{12} & \dots & X_{1p} \\ 1 & X_{21} & X_{22} & \dots & X_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \dots & X_{np} \end{pmatrix} \tag{2.1}$$

$$\beta_{(p+1)\times 1} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \qquad \epsilon_{n\times 1} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix} \tag{2.2}$$

In matrix terms, the general linear regression model is

$$\mathbf{Y}_{n\times 1} = \mathbf{X}_{n\times(p+1)}\beta_{(p+1)\times 1} + \epsilon_{n\times 1}$$

where,

- **Y** is a vector of responses.
- β is a vector of parameters.
- **X** is a matrix of constants.
- ϵ is a vector of independent normal random variables.

2.2 Estimating the Regression Coefficients

As was the case in the simple linear regression setting, the regression coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are unknown, and must be estimated. Given estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$, we can make predictions using the formula

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots, \hat{\beta}_p x_p$$

We choose $\beta_0, \beta_1, \ldots, \beta_p$ to minimize the sum of squared residuals

$$\begin{aligned} RSS &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} (y_1 - \hat{\beta}_0 - \hat{\beta}_1 \hat{x}_{i1} - \hat{\beta}_2 \hat{x}_{i2} - \ldots - \hat{\beta}_p \hat{x}_{ip})^2 \end{aligned}$$

The values $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ that minimize the RSS are the multiple least squares regression coefficient estimates, they are calculated using this formula (in matrix terms):

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

Note 1:

> It is a remarkable property of matrix algebra that the results for the general linear regression model in matrix notation appear exactly as those for the simple linear regression model. Only the degrees of freedom and other constants related to the number of X variables and the dimensions of some matrices are different.

Note 2:

> If $\mathbf{X}^T \mathbf{X}$ is noninvertible, the common causes might be having:
>
> - Redundant features, where two features are very closely related (i.e. they are linearly dependent)
> - Too many features (e.g. $p \geq n$). In this case, we delete some features or we use “regularization” (to be explained in a later lesson).

2.3 Some important questions

When we perform multiple linear regression, we usually are interested in answering a few important questions.

1. Is at least one of the predictors $X_1, X_2, \ldots, X_p$ useful in predicting the response?
2. Do all the predictors help to explain Y, or is only a subset of the predictors useful?
3. How well does the model fit the data?
4. Given a set of predictor values, what response value should we predict, and how accurate is our prediction?

Relationship Between the Response and Predictors?

F-Statistic

Recall that in the simple linear regression setting, in order to determine whether there is a relationship between the response and the predictor we can simply check whether $\beta_1 = 0$. In the multiple regression setting with p predictors, we need to ask whether all of the regression coefficients are zero, i.e. whether $\beta_1 = \beta_2 = \ldots = \beta_p = 0$. As in the simple linear regression setting, we use a hypothesis test to answer this question. We test the null hypothesis,

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$$

versus the alternative hypothesis

$$H_1 : \text{at least one}\, \beta_j \text{ is non-zero}$$

This hypothesis test is performed by computing the F-statistic (*Fisher*):

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)} \sim F_{p,n-p-1}$$

where, as with simple linear regression, $\text{TSS} = \sum(y_i - \bar{y})^2$ and $\text{RSS} = \sum(y_i - \hat{y}_i)^2$.

When the F-statistic value is close to 1, then H_0 is true, which means there is no relationship between the response and predictors. On the other hand, if H_1 is true, so we expect F to be greater than 1.

So the question we ask here: *Is the whole regression explaining anything at all?* The answer comes from the F-test in the ANOVA (ANalysis Of VAriance) table. This is what we get in an ANOVA table:

Source	df	SS	MS	F	p-value
Factor (Explained)	$p-1$	SST	SST/$(k-1)$	MST/MSE	p-value
Error (Unexplained)	$n-p$	SSE	SSE/$(n-k)$		
Total	$n-1$	SS			

The ANOVA table has many pieces of information. What we care about is the F Ratio and the corresponding p-value. We compare the F Ratio with $F_{(p-1,n-p)}$ and a corresponding α value (error).

p-values

The p-values provide information about whether each individual predictor is related to the response, after adjusting for the other predictors. Let's look at the following table we obtain in general using a statistical software for example

	Coefficient	Std. error	t-statistic	p-value
Constant	2.939	0.3119	9.42	<0.0001
X_1	0.046	0.0014	32.81	<0.0001
X_2	0.189	0.0086	21.89	<0.0001
X_3	-0.001	0.0059	-0.18	0.8599

In this tablewe the following model

$$Y = 2.939 + 0.046X_1 + 0.189X_2 - 0.001X_3$$

Note that for each individual predictor a t-statistic and a p-value were reported. These p-values indicate that X_1 and X_2 are related to Y, but that there is no evidence that X_3 is associated with Y, in the presence of these two.

Deciding on Important Variables

The most direct approach is called *all subsets* or *best subsets* regression: we compute the least squares fit for all possible subsets and then choose between them based on some criterion that balances training error with model size.

However we often can't examine all possible models, since they are 2^p of them; for example when $p = 40$ there are over a billion models! Instead we need an automated approach that searches through a subset of them. Here are two commonly use approaches:

Forward selection:

- Begin with the *null model* — a model that contains an intercept (constant) but no predictors.
- Fit p simple linear regressions and add to the null model the variable that results in the lowest RSS.
- Add to that model the variable that results in the lowest RSS amongst all two-variable models.
- Continue until some stopping rule is satisfied, for example when all remaining variables have a p-value above some threshold.

Backward selection:

- Start with all variables in the model.
- Remove the variable with the largest p-value — that is, the variable that is the least statistically significant.
- The new (p1)-variable model is fit, and the variable with the largest p-value is removed.
- Continue until a stopping rule is reached. For instance, we may stop when all remaining variables have a significant p-value defined by some significance threshold.

  There are more systematic criteria for choosing an "optimal" member in the path of models produced by forward or backward stepwise selection. These include *Mallow's* C_p , *Akaike information criterion (AIC)*, *Bayesian information criterion (BIC)*, *adjusted* R^2 and *Cross-validation (CV).*

Model Fit

Two of the most common numerical measures of model fit are the RSE and R^2, the fraction of variance explained. These quantities are computed and interpreted in the same fashion as for simple linear regression. Recall that in simple regression, R^2 is the square of the correlation of the response and the variable. In multiple linear regression, it turns out that it equals $Cor(Y, \hat{Y})^2$, the square of the correlation between the response and the fitted linear model; in fact one property of the fitted linear model is that it maximizes this correlation among all possible linear models. An R^2 value close to 1 indicates that the model explains a large portion of the variance in the response variable.

In general RSE is defined as

$$\text{RSE} = \sqrt{\frac{1}{n-p-1}\text{RSS}}$$

2.3.1 Other Considerations in Regression Model

Qualitative Predictors

- If we have a categorial (qualitative) variable (feature), how do we fit into a regression equation?
- For example, if X_1 is the gender (male or female).
- We can code, for example, male = 0 and female = 1.
- Suppose X_2 is a quantitative variable, the regression equation becomes:

$$Y_i \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 = \begin{cases} \beta_0 + \beta_2 X_2 & \text{if male} \\ \beta_0 + \beta_1 X_1 + \beta_2 X_2 & \text{if female} \end{cases}$$

- Another possible coding scheme is to let male = -1 and female = 1, the regression equation is then:

$$Y_i \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 = \begin{cases} \beta_0 - \beta_1 X_1 + \beta_2 X_2 & \text{if male} \\ \beta_0 + \beta_1 X_1 + \beta_2 X_2 & \text{if female} \end{cases}$$

Interaction Terms

- When the effect on Y of increasing X_1 depends on another X_2.
- We may in this case try the model

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$$

- $X_1 X_2$ is the Interaction term.

PW 2

RMarkdown

Markdown Rmarkdown Html notebook

Multiple Linear Regression

Text

Part III

Classification

Chapter 3

Logistic Regression + K-NN

3.1 Introduction

The linear regression model discussed in the previous two chapters assumes that the response variable Y is quantitative. But in many situations, the response variable is instead qualitative (categorical). For example, eye color is qualitative, taking on values blue, brown, or green.

The process for predicting qualitative responses is known as ***classification***.

Given a feature vector X and a qualitative response Y taking values in the set $\mathcal{C}$, 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. $C(X) \in \mathcal{C}$. We are often more interested in estimating the probabilities that X belongs to each category in $\mathcal{C}$.

If c is a category ($c \in \mathcal{C}$), by the probability that X belongs to c we mean $p(X \in c) = \Pr(Y = c|X)$.

Now take the example of Andrew

3.2 Logistic Regression

Consider a data set where the response falls into one of two categories, Yes or No. Rather than modeling the response Y directly, logistic regression models the *probability* that Y belongs to a particular category.

3.2.1 The Logistic Model

Let us suppose the response has two categories and we use the generic 0/1 coding for the response. How should we model the relationship between $p(X) = \Pr(Y = 1|X)$ and X?

Why not linear regression? Any time a straight line is fit to a binary response that is coded as 0 or 1, in principle we can always predict $p(X) < 0$ for some values of X and $p(X) > 1$ for others (unless the range of X is limited).

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. In logistic regression, we use the *logistic function*,

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

No matter what values β_0, β_1 or X take, $p(X)$ will have values between 0 and 1.

The logistic function will always produce an *S-shaped* curve.

After a bit of manipulation of the previous equation, we find that

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

The quantity $p(X)/[1p(X)]$ is called the *odds*, and can take on any value between 0 and ∞.

By taking the logarithm of both sides of the equation, we arrive at

$$\log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X$$

The left-hand side is called the *log-odds* or *logit.* We see that the logistic regression model has a logit that is linear in X.

3.2.2 Estimating the Regression Coefficients

We estimate β_0 and β_1 using the *maximum likelihood* method. The basic intuition behind using maximum likelihood to fit a logistic regression model is as follows: we seek estimates for β_0 and β_1 such that the predicted probability $\hat{p}(x_i)$ of the response for each individual, corresponds as closely as possible to the individual's observed response status (recall that the response Y is categorical). The *likelihood function* is

$$l(\beta_0, \beta_1) = \prod_{i:y=1} p(x_i) \prod_{i':y=0} (1 - p(x_{i'}))$$

This likelihood gives the probability of the observed zeros and ones in the data. The estimates $\hat{\beta_0}$ and $\hat{\beta_1}$ are chosen to *maximize* this likelihood function.

Some remarks: * $p(x_i)$ is the probability of x_i is 1. And $1-p(x_i)$ is the probability of x_i is 0. * The likelihood expression written in the equation above is the joint probability of the observed sequence of 0's and 1's. * We suppose that the x_i's are independent.

In the linear regression setting, the least squares approach is a special case of maximum likelihood.

We will not give mathematical details about the maximum likelihood and how to estimate the parameters. We will use R to fit the logistic regression models (using `glm` function).

Exapmle

	Coefficient	Std. error	Z-statistic	p-value
Constant	-10.6513	0.3612	-29.5	<0.0001
X	0.0055	0.0002	24.9	<0.0001

In this example, $\hat{\beta_0} = -10.6513$ and $\hat{\beta_1} = 0.0055$. It produces the blue curve that separates that data in the following figure,

As for prediction, we use the model built with the estimated parameters to predict probabilities. For example,

If $X = 1000$,

$$\hat{p}(X) = \frac{e^{\hat{\beta_0}+\hat{\beta_1}X}}{1+e^{\hat{\beta_0}+\hat{\beta_1}X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

If $X = 2000$,

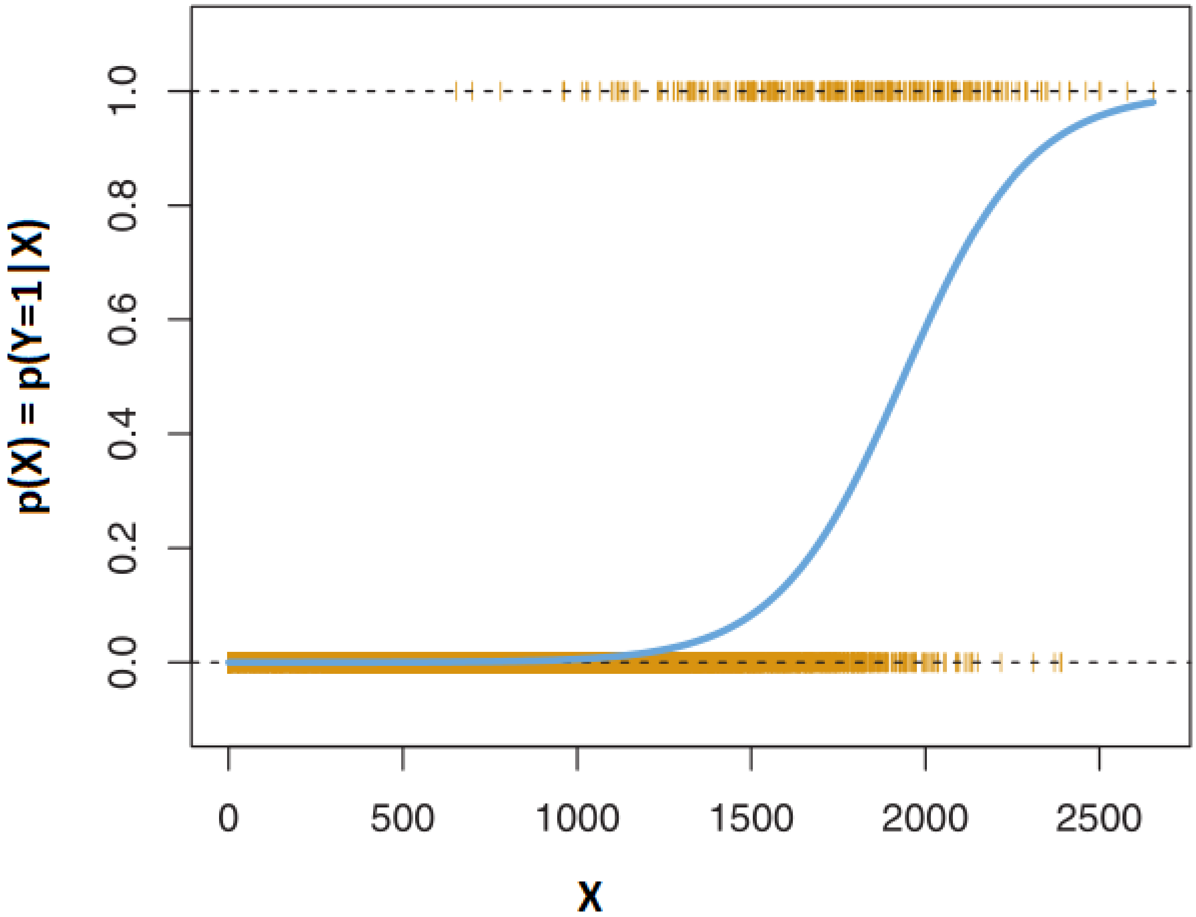


Figure 3.1:

$$\hat{p}(X) = \frac{e^{\hat{\beta_0}+\hat{\beta_1}X}}{1+e^{\hat{\beta_0}+\hat{\beta_1}X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

3.3 Multiple Logistic Regression

We now consider the problem of predicting a binary response using multiple predictors. By analogy with the extension from simple to multiple linear regression in the previous chapters, we can generalize the simple logistic regression equation as follows:

$$\log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

where $X = (X_1, \ldots, X_p)$ are p predictors. The equation above can be rewritten as

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}$$

Just as in the simple logistic regression we use the maximum likelihood method to estimate $\beta_0, \beta_1, \ldots, \beta_p$.

PW 3

Subsection

Text

Subsection

Text

Chapter 4

Discriminant Analysis

4.1 Subsection 1

Text

4.2 Subsection 2

Text

4.2.1 Subsection 3

Text

PW 4

Subsection

Text

Subsection

Text

Chapter 5

Support Vector Machines

5.1 Subsection 1

Text

5.2 Subsection 2

Text

5.2.1 Subsection 3

Text

PW 5

Subsection

Text

Subsection

Text

Part IV

Unsupervised Learning

Chapter 6

Dimensionality Reduction

6.1 Subsection 1

Text

6.2 Subsection 2

Text

6.2.1 Subsection 3

Text

PW 6

Subsection

Text

Subsection

Text

Chapter 7

Clustering: Kmeans

7.1 Subsection 1

Text

7.2 Subsection 2

Text

7.2.1 Subsection 3

Text

PW 7

Subsection

Text

Subsection

Text

Chapter 8

Expectation Maximisation

8.1 Subsection 1

Text

8.2 Subsection 2

Text

8.2.1 Subsection 3

Text

PW 8

Subsection

Text

Subsection

Text

References & Resources

Readings:

1. Tibshirani
2. Andrew Ng
3. Applied Linear Statistical Models
4. Bishop