K Nearest Neighbours

In this lecture. . .

- Coloured dots in space
- Introduce a new dot
- What colour should it be? Just look at its neighbours

Introduction

K nearest neighbours is a supervised-learning technique. We take classified data represented by a vector of features. Think of coloured dots in space, the dimension of that space being the number of features. We want to classify a new dot, i.e. figure out its colour, given where it is in space.

We measure distances between the new data point and the nearest K of our already-classified data and thus conclude to which class our new data point belongs by majority voting.

Instead of having classes we can associate a number with each data point and so use the method for regression.

What Is It Used For?

K nearest neighbours is used for

- Classifying samples according to their numerical features
- Performing a regression of numerical values based on the features of the sample
- Example: Predict whether a spouse will like a Christmas present based on price, size, — of the present, not of the spouse — Amazon rating, etc.

- Example: Forecast the sales of a mathematics text book based on features such as price, length, number of university courses in the area, etc.

- Example: Credit scoring, classify as good or bad risk, or on a scale, based on features such as income, value of assets, etc.

- Example: Predict movie rating (number of 'stars') based on features such as amount of action sequences, quantity of romance, budget, etc.

- Example: Classify risk of epilepsy by analysing EEG signals

How It Works

K nearest neighbours (KNN) is perhaps the easiest machine-learning technique to grasp conceptually. Although really there is no learning at all.

It can be used for classification, determining into which group an individual belongs. Or it can be used for regression, predicting for a new individual a numerical value of a variable based on the values for similar individuals.

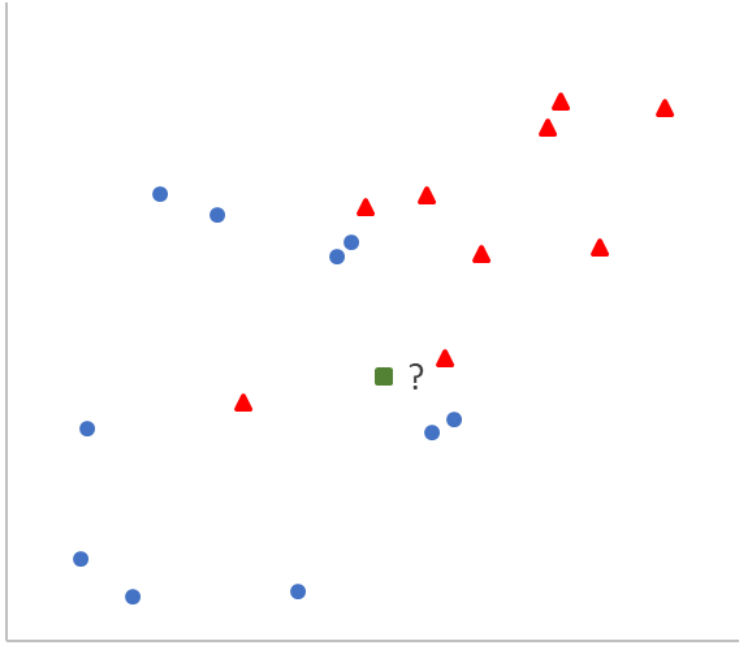
- We start off with N points of already-classified data, making this a supervised-learning technique

- Each data point has M features. Each point is represented by a vector with M entries. Each entry represents a feature. The first entry in the vector might be the number of words in an email, the second entry might be the number of exclamation points, the third might be number of spelling mistakes, and so

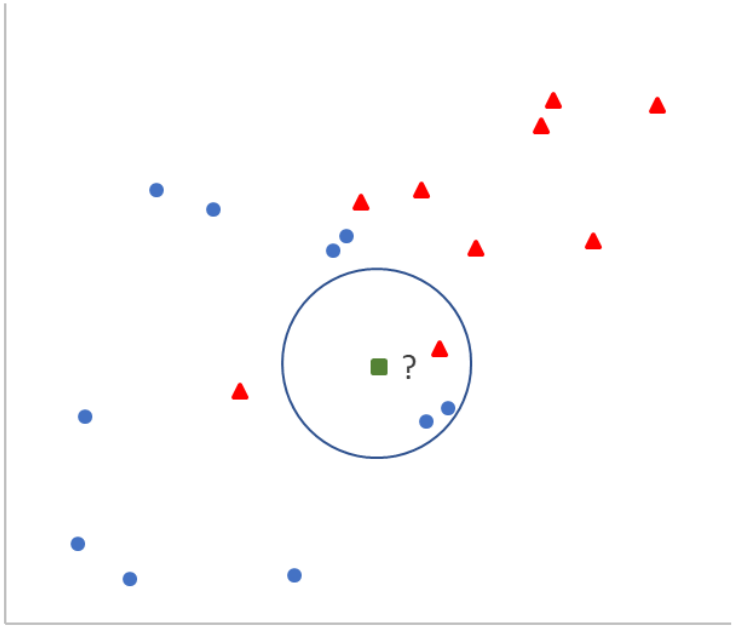
- And then each vector is classified as 'spam' or 'not spam.' The number of classifications does not have to be two

When we have a new data point to classify we simply look at the K original data points nearest to our new point and those tell us how to classify our new point.

In the figure below should the new point (temporarily marked a square) be classified as a circle or a triangle?



Suppose we decide that $K = 3$, that we will look at the three nearest neighbours, then from the figure we see that two of the nearest neighbours are circles and only one is a triangle. Thus we classify our square as being a circle (if you see what I mean!). This is simple 'majority voting.'



The K Nearest Neighbours Algorithm

Step 0: Scaling

The first thing we do is to scale all features so that they are of comparable sizes. Perhaps scale so that the mean number of words in an email is zero with a standard deviation of one. Do this for all features.

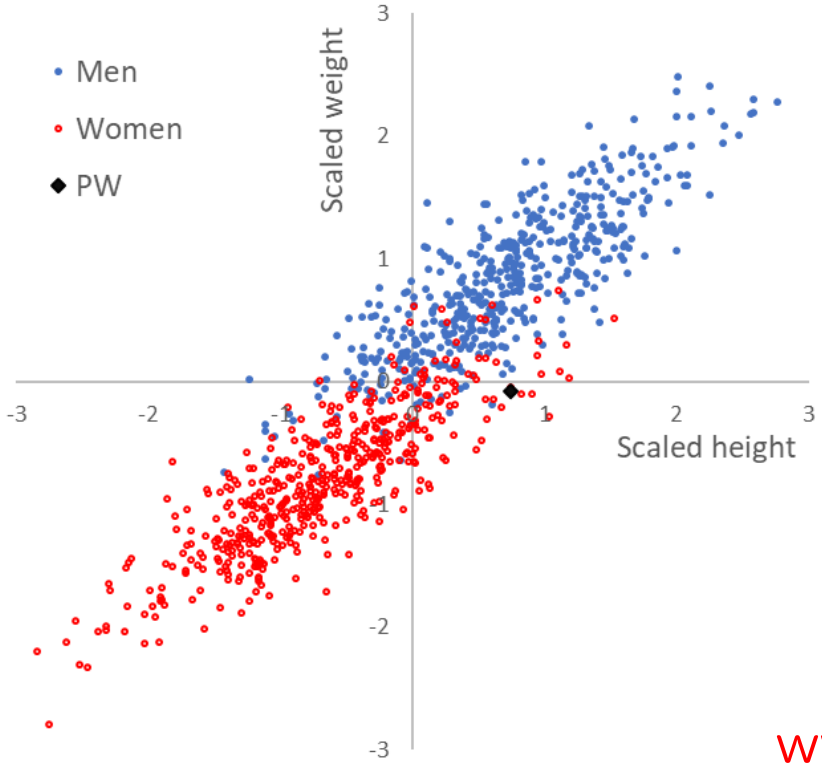
Step 1: A new data point

We now take our new, unclassified data point, and measure the distance from it to all of the other data points. We look at the K points with the smallest distance. From those K points the classification with the most number of appearances determines the classification of our new point.

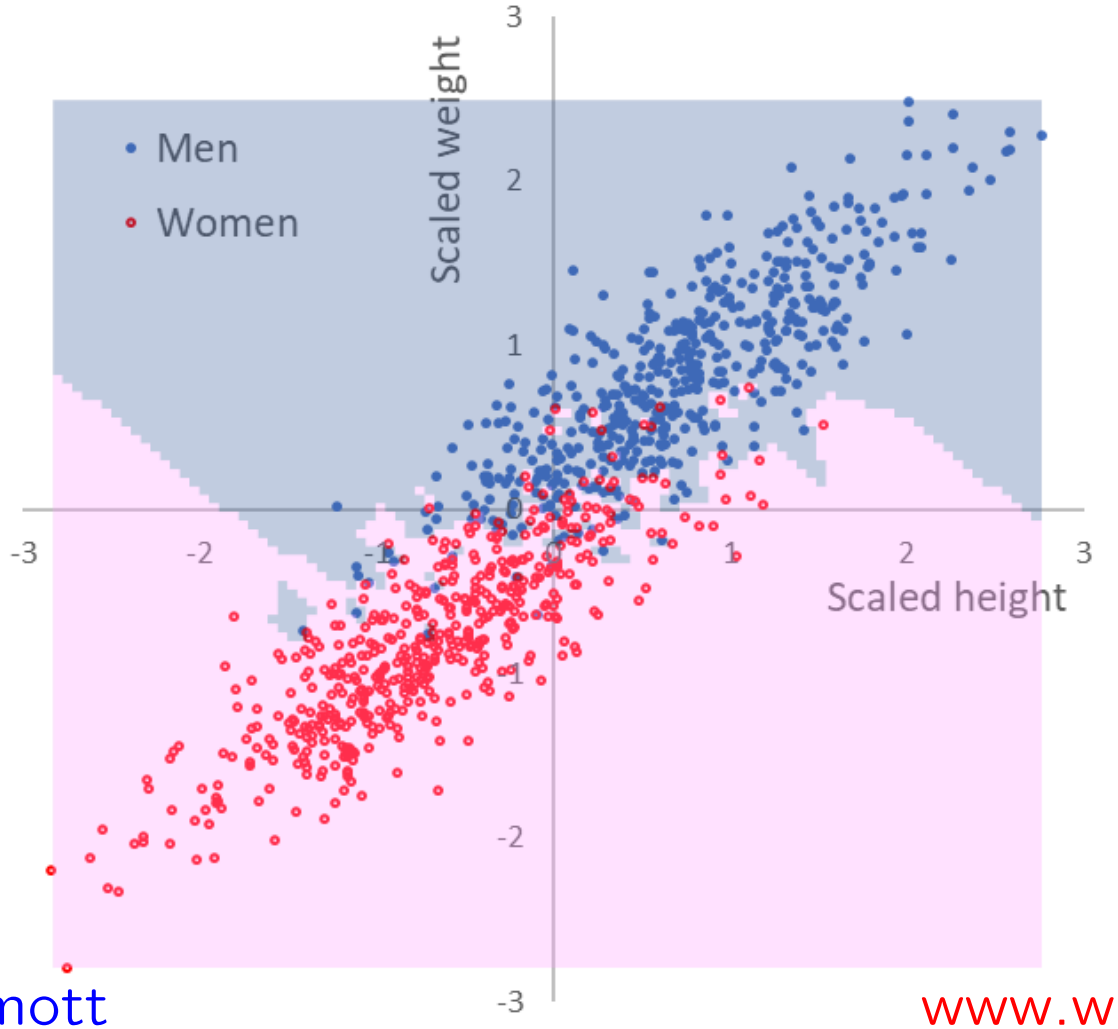
There is no real learning, as such, done in KNN. Algorithms with this feature are often called 'lazy.'

Example: Heights and weights

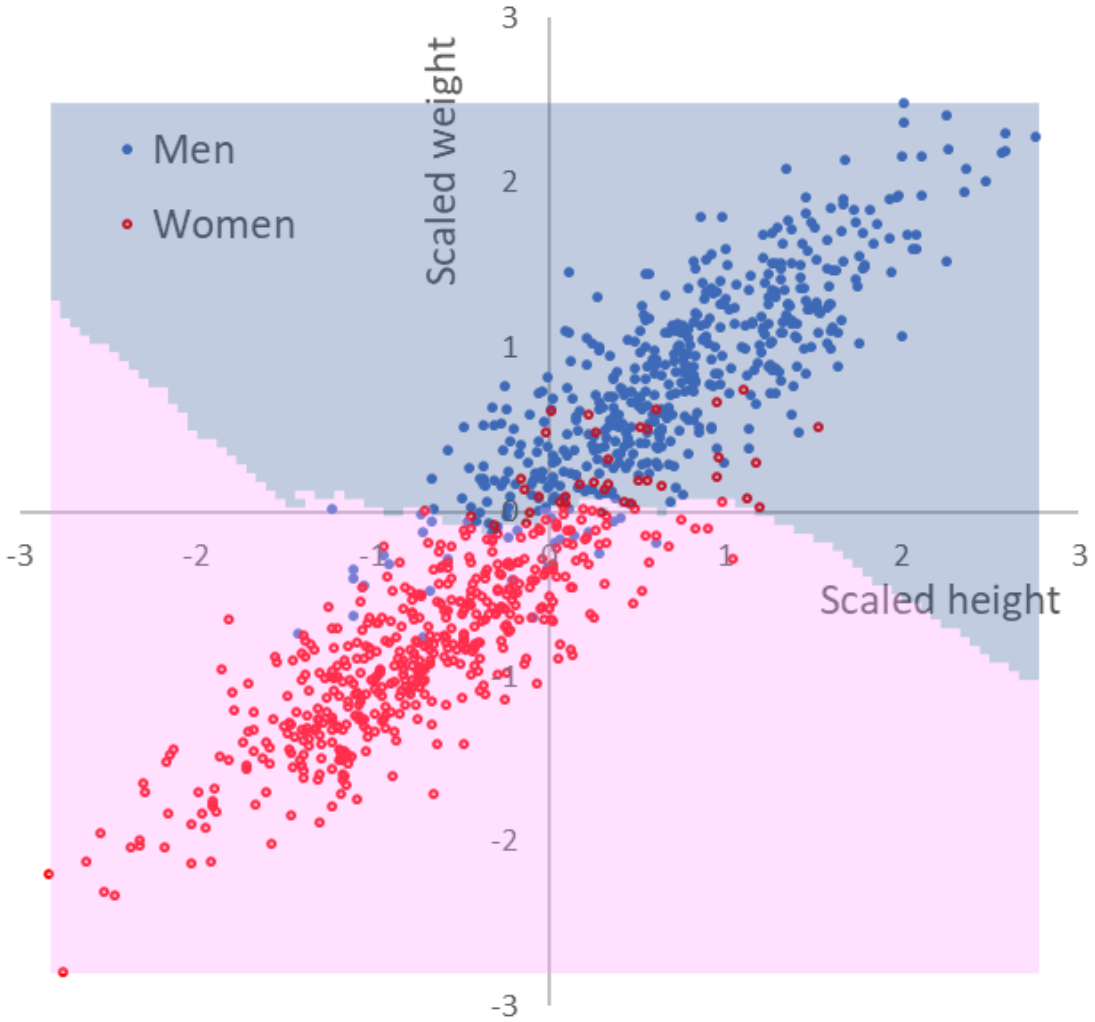
You can easily find online data for men's and women's heights and weights. In the figure I show sample data, just 500 points for each gender. Note that the data has been shifted and scaled so as to have a mean of zero and a standard deviation of one. Obviously I've used the same transformation for both genders.



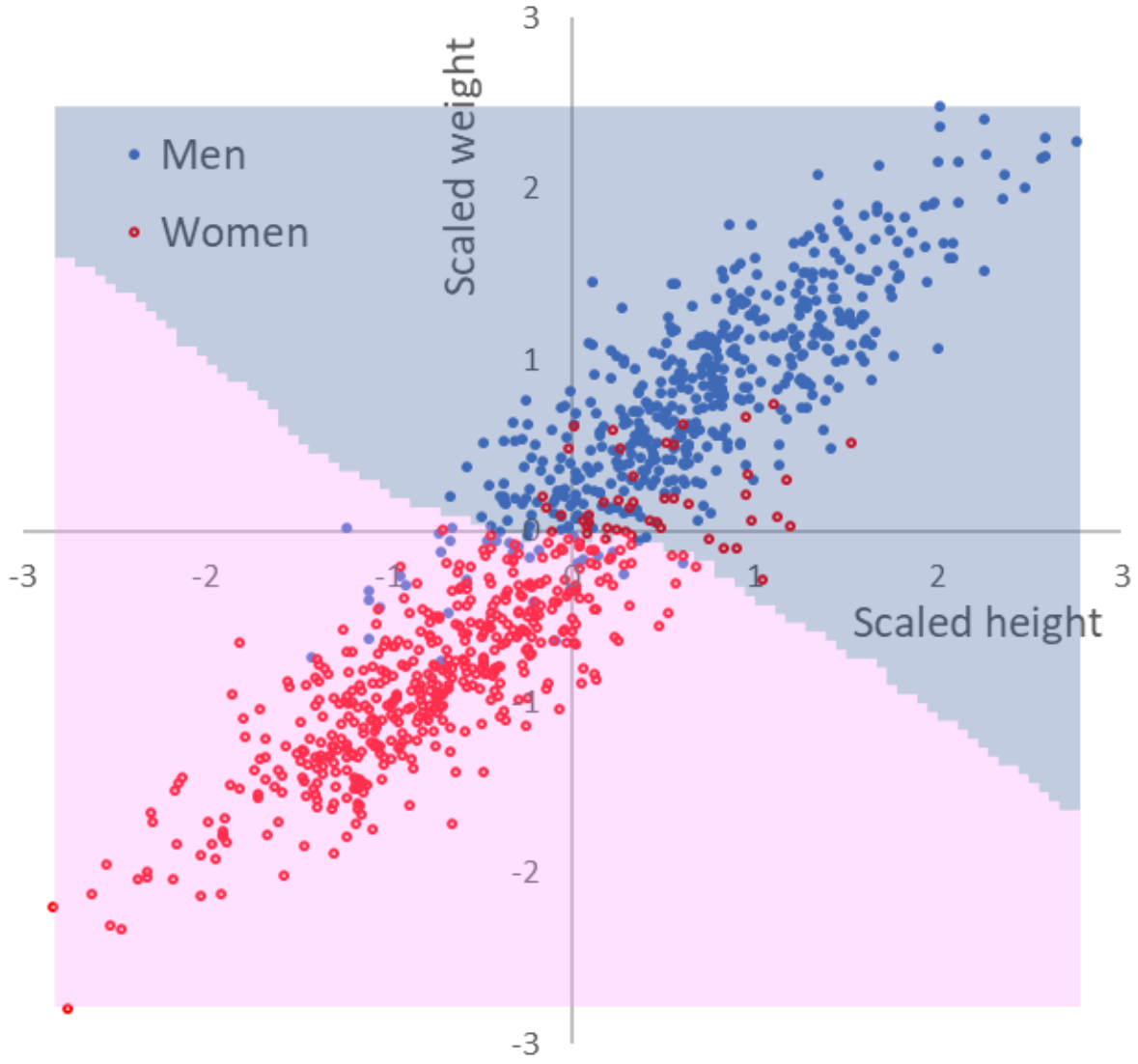
When $K = 1$ the male/female regions are as shown below. You can see several islands, pockets of males within what looks like otherwise female zones and vice versa.



When $K = 21$ the boundary has been smoothed out and there is just a single solitary island left.



As an extreme case take $K = 101$.

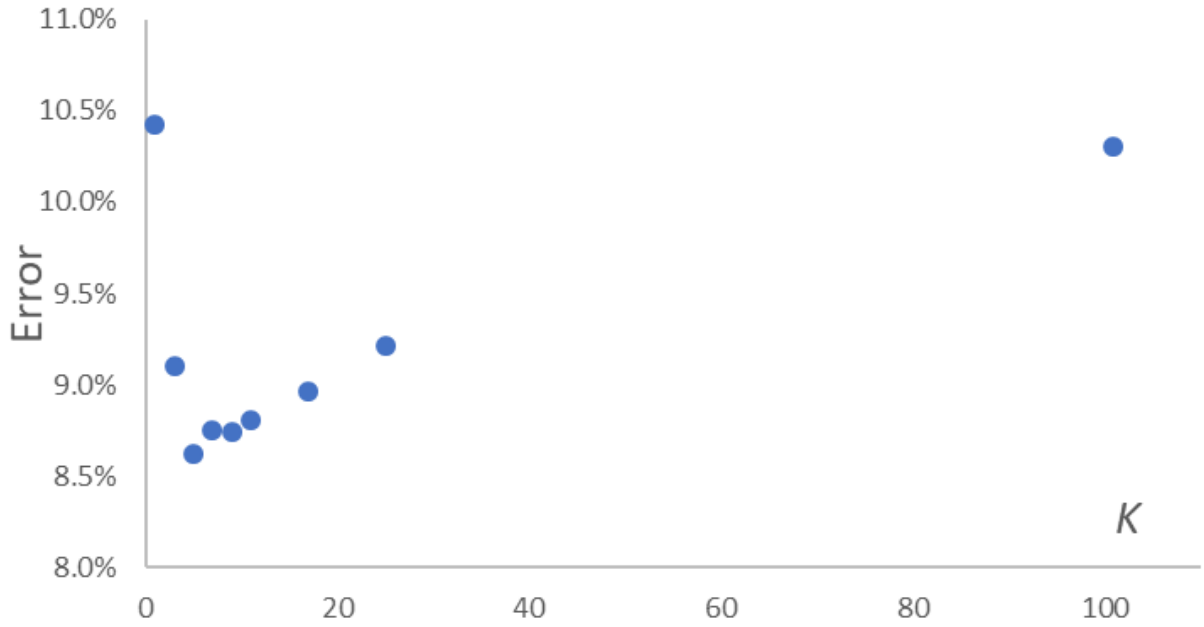


These figures nicely show the difference between bias and variance and the trade off between them.

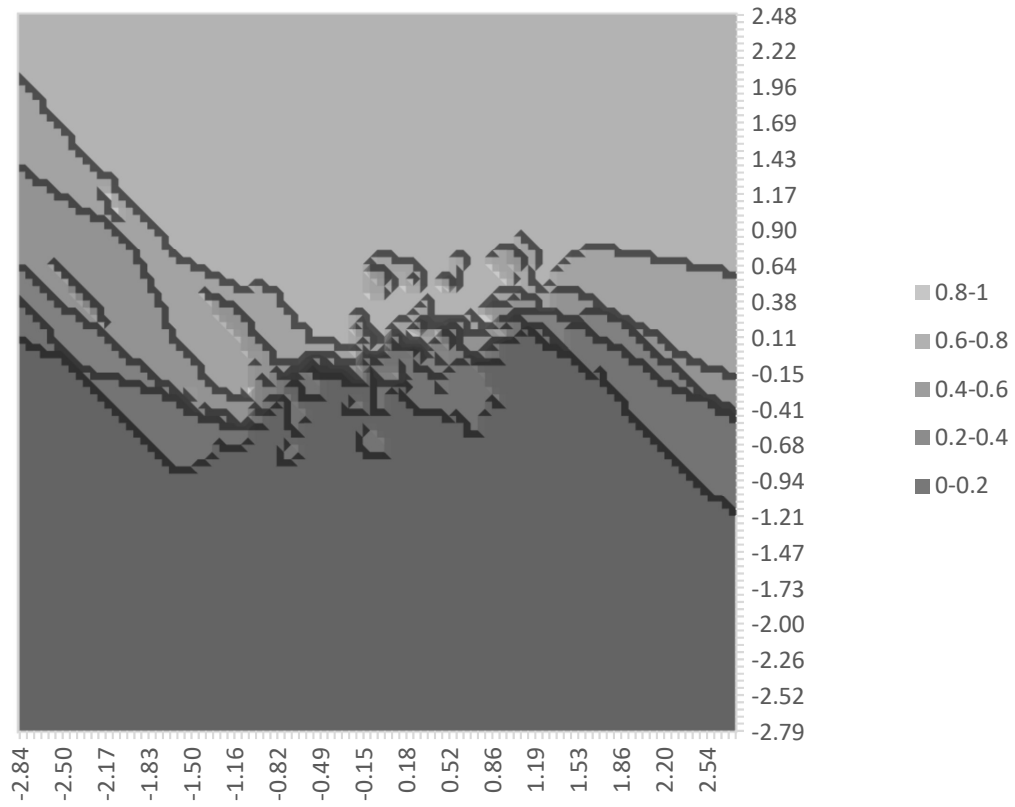
One figure shows high variance but low bias. It's a complicated model because it only looks at the nearest neighbour. Yes, I know that sounds like it's a simpler model but it's really not. The model changes a lot as we move around because the nearest neighbours change a lot.

Another figure shows high bias and low variance. It's a simpler model because the nearest neighbours don't change so much as we look at new samples. It has a high bias and low variance. In the extreme case in which we use the same number of neighbours as data points then every prediction would be the same.

We can plot misspecification error as a function of the number of neighbours, K. I have done this below using out-of-sample data. It looks like $K = 5$ is optimal.



We can also get a probabilistic interpretation of the classification. Here is shown the probability of being male. This uses $K = 5$ and the obvious calculation.



Regression

I've explained how KNN can be used for classification but it can also easily be used for 'regression.'

The idea is simple. Instead of samples being labelled with classes, such as type of fruit, the labels are numbers. The features might be age, IQ, and height with the labels being salary. Your goal is to find the salary for a new sample. You tell me your age, IQ and height and I'll tell you how much you earn.

This regression can be done in a variety of ways.

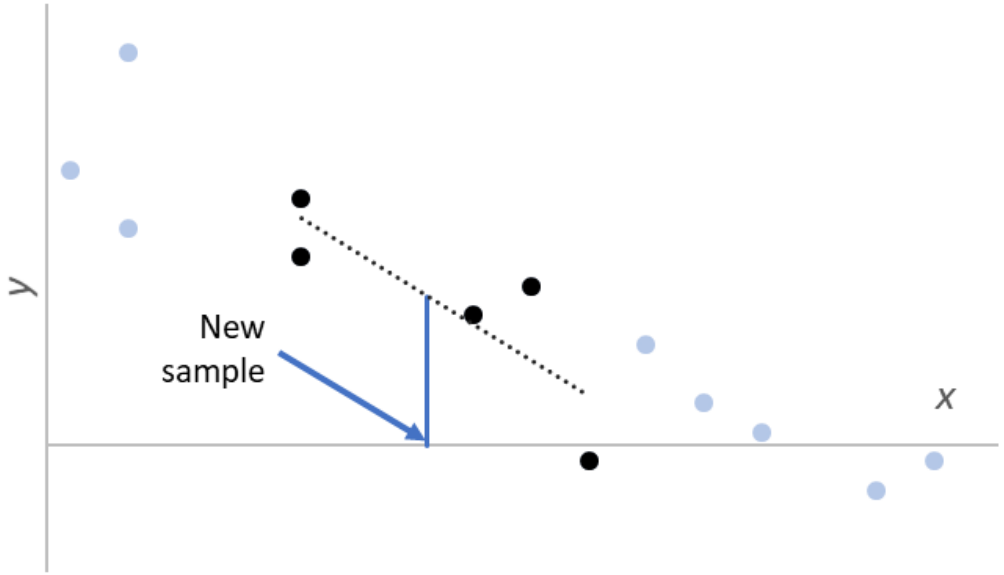
- The simplest method is to just average the numbers (salaries) of the nearest K neighbours

- You could get more sophisticated by weighting the average, perhaps weighted by the inverse of the distances to the new data point.

- Or exponentially weighted by distance

- Or using a Gaussian function of the distance, a Gaussian kernel

- Or by fitting a hyperplane to the K data points. This is a local linear regression. The figure shows an example. Here $K = 5$. The bold dots are the nearest five points to our new sample. The faint dots represent the ones we are ignoring in the linear regression



These methods generally go by the name of 'kernel smoothing.'

Summary

Please take away the following important ideas

- K nearest neighbours is possibly the easiest machine-learning technique
- But there isn't any learning
- It is very easy to understand and can be used for classification or regression