kNN Regression

Bookmark this page

Predicting a Variable

Let's imagine a scenario in which we would like to predict one variable using another variable, or a set of other variables. An example is predicting the **number of views** a TikTok video will get next week based on video length, the date it was posted, the previous number of views, etc. Or predicting **which movies** a Netflix user will rate highly based on their previous movie ratings, demographic data, etc.

Working Example

Throughout the course, we will try to use working examples - sets of real-world data that we can use as a class. The advertising data set (below) consists of the sales of a particular product in 200 different markets, and advertising budgets for the product in each of those markets for three different media: TV, radio, and newspaper. We have 4 columns and 200 rows where budgets are given in units of $1,000 and sales in 1,000 sales.

Skip to below table.

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

Some of the figures are taken from "An Introduction to Statistical Learning, with Applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T.Hastie and R. Tibshirani

Response vs. Predictor Variables

There is an **asymmetry** in many of these problems: The variable we would like to predict may be more difficult to measure, is more important than the other(s), or maybe directly or indirectly influenced by the other variable(s). Thus, we like to define two categories of variables:

- Variables whose values we **use** to make our prediction. These are known as *predictors*, *features*, or *covariates*.
- Variables whose values we want to **predict**. These are known as *outcomes*, *response variables*, or *dependent variables*.

In this course, the terms predictors and response variables will be used. The first three columns in the advertising data (TV, radio, newspaper) are the predictors denoted by p. Each predictor on its own is a vector; when put together, they are denoted by X which is also known as **The Data Matrix** or **Design Matrix**. These variables are used to predict sales, the response variable, which is denoted by y.

Every row in the data denotes one observation and in general we have n observations and p predictors in a data set.

In general, the nomenclature that we will use throughout the course is as follows:

- n observations
- p predictors $X = X_1, \ldots, X_p$
- Each predictor denoted by $X_j = x_{1j}, \ldots, x_{ij}, \ldots, x_{nj}$
- Response Variables $Y_j = y_1, \ldots, y_n$

Connecting with Pandas

If you're practicing on your home computer, make sure you have pandas installed on your machine. It's already available in our programming environment within the course, so if you are practicing here you won't need to install anything.

Let's connect with Pandas, taking the data matrix and checking the shape that gives the dimensions of the data matrix $\boldsymbol{X}$.

```
X.shape
```

results in (n, p) where n is the number of observations and p is the number of predictors.

For the response variable,

```
y.shape
```

results in $(n, 1)$ or $(n,)$.

For simplification for the rest of this section, we will consider one predictor (TV budget) where the shape of the data matrix would be $(n, 1)$ or $(n,)$ and the shape of the response variable sales is the same as before, $(n, 1)$ or $(n,)$. In later sections we will consider more than one predictor.

Connecting to Pandas, (n,1) is defined as a Pandas `DataFrame` with one column and (n,) is defined as a Pandas `Series`. When we later use `sklearn` to do modeling, `sklearn` expects the predictors to be an array with at least one column and not a data series.

When we read in our data into a data frame 'df' with a column named 'x', there is an important difference between the two operations `df[['x']]` versus `df['x']`.

- `df[['x']]` returns a Pandas data frame object that is an array we can use in `sklearn`
- `df['x']` returns a Pandas series that will give an error when using sklearn

> **Data Object Difficulties**
>
> Sometimes you know you have the data stored, but you're getting an error about the type of object you stored it in. Not sure how you might solve that kind of problem? Patrick ran into the same problem. Here's how he solved it.

Video

Video

Descargar archivo de video

Transcripciones

Descargar SubRip (.srt) file

Descargar Text (.txt) file

Calculator

True vs. Statistical Model

Now we have decided how we are going the names for our data, let us move to how we are going to model. We will assume that the response variable, Y , relates to the predictors, X, through some **unknown function** generally expressed as:

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

Here f is the unknown function expressing an underlying rule to relate Y to X, ϵ is the `random` amount (not related to X) that describes the difference of Y from rule $f(X)$.

A **statistical model** is any algorithm that estimates f. We denote the estimated function as $\hat{f}$.

Prediction vs. Estimation

For some problems, what is important is obtaining $\hat{f}$, our estimate of f. These are called **inference** problems.

When we use a set of measurements, $(x_{1p}, \ldots, x_{ip})$ to predict a value for the response variable, we denote the predicted value by:

$$\hat{y_i} = \hat{f}\,(x_{i1}, \ldots, x_{ip})$$

In other cases, we do not care about the specific form of $\hat{f}$, we just want to make our predictions $\hat{y}$ as close to the observed values y's as possible. These are called **prediction** problems.

Discussion (External resource)

Haga clic en Aceptar para que su nombre de usuario y dirección de correo electrónico se envíen a una aplicación de terceros.

Aceptar

‹ Previous Next ›

Calculator