What is a Model?

Kate Becker

2024-08-26

Intro Packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

A Model is any sort of function that has predictive power

Exploring mtcars package

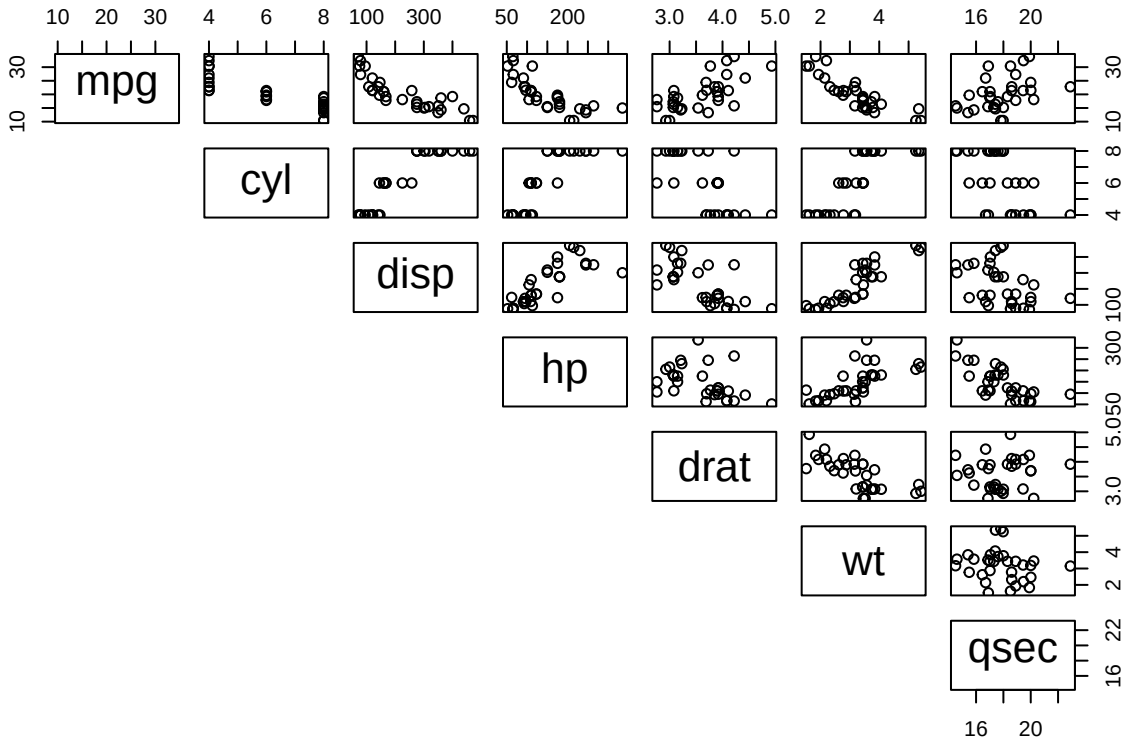
```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

- Columns of data are called features

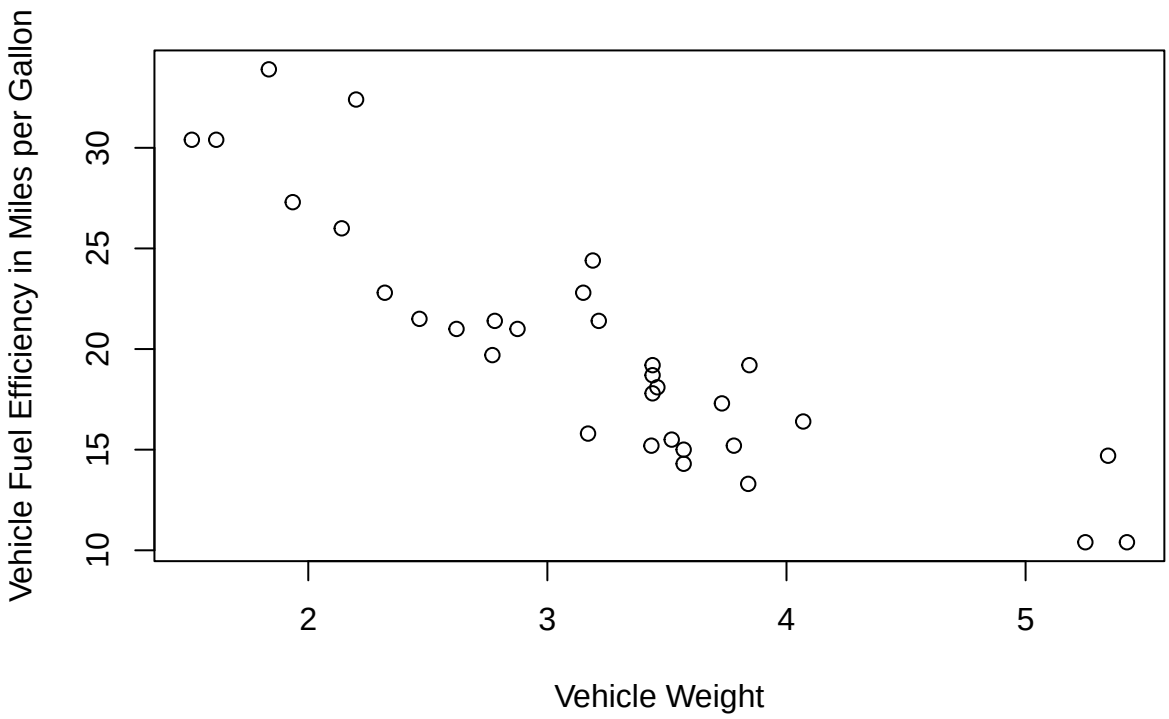
Examining if theres a relationship between the cars fuel efficiency and any of the features

```
# matrix of scatterplots if producted with pairs()
# lower.panel separate panel functions to be used above diaganol
# clearly nothing in the cyl row look like they help for simple regression modelling
pairs(mtcars[1:7], lower.panel = NULL)
```



- MPG as a function of cycl looks very different from mpg as a function of wt

```
plot(y = mtcars$mpg, x = mtcars$wt, xlab = "Vehicle Weight", ylab = "Vehicle Fuel Efficiency in Miles p
```



From this kind of data format, we can extract a best fit to all the data points and turn this plot into an equation.

Using a function to model the value were interested in, called a response, against other features in the dataset

```
mt.model <- lm(formula = mpg ~ wt, data = mtcars)
mt.model
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Coefficients:
## (Intercept)           wt
##      37.285       -5.344
```

```
coef(mt.model)[2]
```

```
##        wt
## -5.344472
```

```
coef(mt.model[1])
```

```
## (Intercept)          wt
##   37.285126   -5.344472
```

Now we have predictive power given some kind of input that can give us a value for any number we put in but there may be error in data or observation.

Algorithms

An algorithm is a set of steps performed in order - Simplest: linear regression, involves putting two points on a plot and then drawing a line between them you get the important parts of the question (slope and intercept) by taking the difference in the coordinates of those points with respect to some origin - this gets more complicated with more than two points though so need more complex systems that humans cant do to do this - Algorithsm are the underpinnings of simple R code

3 major types of models:

- Regression models
- Classification models
- Mixed models (Combo of both)

Classificaiton models

- Take input data and arrange it according to a type, class, group, or other discrete output

Mixed models

- Start with a regression model and then use output from that to help to classify other types of data and the reverse can be true
- lm(y~x) "give me the lienar model for the variable y as a function of the feature x"

Function: object that has some processing power, sits inside model

Model: complex object that takes an input parameter and gives an output

Equation: mathematical representation of a funciton

Algorithm: set of steps passed into a model for calculating or processing

- the lm() is a function itself, but also a linear model
  - calls a series of algorithms to find best values that are then output as a slope and an intercept

  All models are wrong like the mtcars data isnt comparable to data we have now

  Coefficient of determination: measure of how close the data fits the model fitted line with values ranging from 0 to 1

Statistics that define model accuracy

Coefficient of determination: Sometimes listed as R^2, this is how well the data fits to the modelling line for regression analyses

P-values: Measures of statistical validity where if your p-value is below 0.05 the values are examining is likely to be statistically valid

Confidence intervals: Two values between wich we expect a parameter to be

*Can use these to understand difference between a model that fits the data well and one that fits poorly, can assess which features are useful for us to use in our model and can determine accuracy of the answers produced by the model

Data Training

We have a model for which we have an input that generates an output, we have data for which we want a prediction so we pass it through the model, then evaluate the results and see if the associated errors in the model go down or not - if they do, we are tuning the model in the right direction - if the errors continue to build we need to tweak our model further *Must not train our ML model on data that we then pass back into it in order to test validity* The model has already seen the data so it will be very accurate - So we should split the data into testing and training! - Be careful with limited data points - Black box model will be trainined on data thats the same form as the test set but the black box model hasnt seen the exact data points in the test set yet (wont be biased) *** Methods of splitting the data is. known as sampling techniques

Cross Validation

- Tuning a machine learning model is when you have a bunch of inputs whose values we can change slightly without changing the underlying data Ex: Model with three parameters that you can tune (A - 1, B = 2, C = FALSE) but if model doesnt turn out right you can tweak i tby changing the values (A = 1.5, B = 2.5, C = "FALSE"), and so forth with various permutations
- Many models have built in ways to ingest data, perform some operations on it, and save the tuned operations to a data structure that is used on the test data
- During training phase might want to try another statistical technique like cross- validation
  - like another mini step of splitting data into training and test sets and running the model Ex: split data 50 data points into 80% training and rest testing. Left with 40 rows to train data model, can split these 40 rows further into a 32 row training set and an 8 row test set. By doing so and going throiugh a simialr training and test procedure you can get an ensemble of errors out of your model and use those to help refine its tuning further

Cross val techniques:

- Bootstrap cross validation
- Bootstrap 632 cross validation
- k fold cross validation
- Repeated cross validation
- leave one out cross validation

- leave group out cross validation
- out of bag.cross validation
- adaptive cross validaiton
- adaptive bootstrap cross validaiton
- adaptive leave group out cross validation
- Use of these techniques is highly dependent on structure of the data itself, typical gold standard is k-fold cross validation where you pick k = 10 folds against which to validate which is the best balance between efficient data useage and avoding splits in the data that may be poor choices

Use swirl for having fun!

We have three different vector types: one numeric, one a vector of factors, and one a vector of logical values

```
v1 = c(1,2,3)
v2 = c("Jerry", "George", "Elaine")
v3 = c(TRUE, FALSE, TRUE)

data_frame = data.frame(v1,v2,v3)

str(data_frame)
```

```
## 'data.frame':    3 obs. of  3 variables:
##  $ v1: num  1 2 3
##  $ v2: chr  "Jerry" "George" "Elaine"
##  $ v3: logi  TRUE FALSE TRUE
```

Packages! dplyr: reshape and manipulate data in a verbiage that makes intuitive sense lubridate: manipulate data on tricky datetime formatted data

Tips: the function operator and acts as an equals sign, lm(mtcars$mpg $mtcars$wt), mathematically this would be y = f(x) and y ~ f(x)

ex: y = f(x1, x2, x3, . . .) in R y ~ x_1 + x_2 + x_3 and here were saying our model output y is not only a function of x_1 but many other variabels as well