Introduction to Data Science

Modeling with Multiple Predictors

Joanna Bieri DATA101

Important Information

- Email: joanna_bieri@redlands.edu
- Office Hours: Duke 209 Click Here for Joanna's Schedule

Announcements

Come to Lab! If you need help we are here to help!

Day 17 Assignment - same drill.

1. Make sure you can **Fork** and **Clone** the Day17 repo from Redlands-DATA101
2. Open the file Day17-HW.ipynb and start doing the problems.
   - You can do these problems as you follow along with the lecture notes and video.
3. Get as far as you can before class.
4. Submit what you have so far **Commit** and **Push** to Git.
5. Take the daily check in quiz on **Canvas**.
6. Come to class with lots of questions!

Data: Book weight and volume

The `allbacks` data frame gives measurements on the volume and weight of 15 books, some of which are paperback and some of which are hardback

- Volume - cubic centimetres
- Area - square centimetres
- Weight - grams

Data: Book weight and volume

These books are from the bookshelf of J. H. Maindonald at Australian National University.

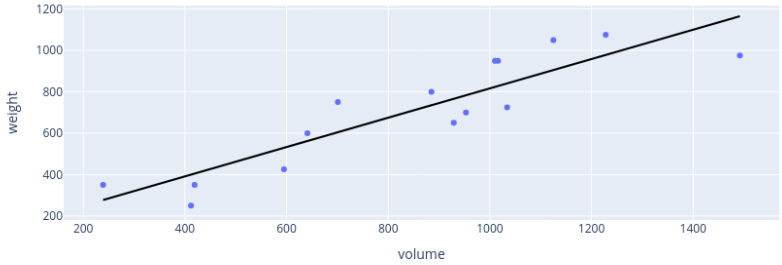
This lab follows the Data Science in a Box units "Models with Multiple Predictors" by Mine Çetinkaya-Rundel. It has been updated for our class and translated to Python by Joanna Bieri.

Data: Book weight and volume

	volume	area	weight	cover
0	885	382	800	hb
1	1016	468	950	hb
2	1125	387	1050	hb
3	239	371	350	hb
4	701	371	750	hb
5	641	367	600	hb
6	1228	396	1075	hb
7	412	0	250	pb
8	953	0	700	pb
9	929	0	650	pb
10	1492	0	975	pb
11	419	0	350	pb
12	1010	0	950	pb
13	595	0	425	pb
14	1034	0	725	pb

Start with a simple linear model

We will try to predict a books weight using the volume. Before we do, what are some things we should consider? Are there variables outside of weight and volume that might affect our results?



Interpret this result

```
X = DF_model['volume'].values.reshape(-1,1)
Coefs:
[0.70863714]
Intercept:
107.679310613766
Score:
0.8026345746312898
```

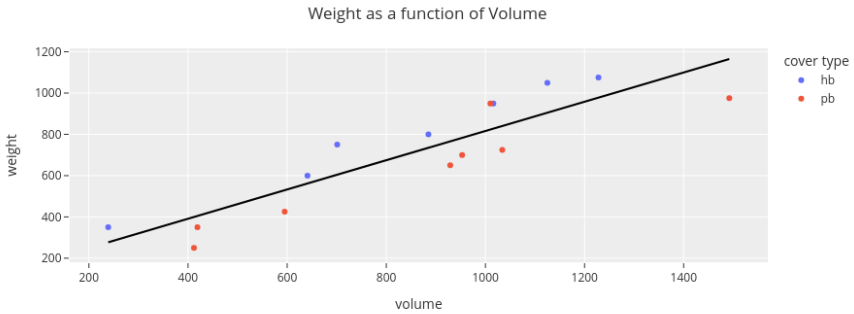
Interpret this result

For every square centimeter the book is larger we expect the weight of the book to increase by 0.7086 grams.

If a book has zero volume then we expect it to have a weight of 107.68 grams. **This does not make sense!** Why not? Because we are extrapolating here. Notice none of our books are even close to zero volume!

What else might affect the weight?

Well whether or not the book is hardback or paperback would definitely change the weight! We will color our points by the cover type.



Do we notice a trend here? Well, it seems that more often the hard bound books are above the prediction line! How could we fix this?

Add another explanatory variable!

First we notice that cover is a categorical variable that can take just two values.

So we need to decide how to encode this variable. Since there are only two options I we could encode pb=1 and hb=0 in our data frame using a command like

```
DF_model['cover_enc'] = DF_model['cover'].apply
                    (lambda x: 1 if x == 'pb' else 0)
```

Add another explanatory variable!

Instead I am going to show you a new and very usefull command:

```
pd.get_dummies(DF,columns=[],dtype=float)
```

The get dummies command creates new columns in your data frame for each possible category in the columns you provide. In our case we are going to say

```
DF_model =
    pd.get_dummies(DF_model, columns=['cover'], dtype=float)
```

and it will create two new columns in our data frame one for cover_hb and the other for cover_pb

Add another explanatory variable!

	weight	volume	prediction	cover_hb	cover_pb
0	800	885	734.823182	1.0	0.0
1	950	1016	827.654648	1.0	0.0
2	1050	1125	904.896097	1.0	0.0
3	350	239	277.043588	1.0	0.0
4	750	701	604.433948	1.0	0.0
5	600	641	561.915720	1.0	0.0
6	1075	1228	977.885723	1.0	0.0
7	250	412	399.637814	0.0	1.0
8	700	953	783.010508	0.0	1.0
9	650	929	766.003217	0.0	1.0
10	975	1492	1164.965929	0.0	1.0
11	350	419	404.598274	0.0	1.0
12	950	1010	823.402825	0.0	1.0
13	425	595	529.318411	0.0	1.0
14	725	1034	840.410117	0.0	1.0

Interpret the result

How do we interpret this? Well you separate each of the explanatory variables by a plus sign... we add them up. In general now the model looks like

$$y = b_0 + b_1 X1 + b_2 X2 + b_3 X3 + b_4 X4....$$

where $X1, X2, X3, ...$ are each of your explanatory variables. For us

$X1 =$ volume

$X2 =$ cover type $=$ hb

$X3 =$ cover type $=$ pb

Interpret the result

So our equation is

$$weight = 105.939 + 0.718(volume) + 92.023(hardbound) - 92.023(paperback)$$

The variables $hardbound$ and $paperback$ take values of zero or one depending on the book type.

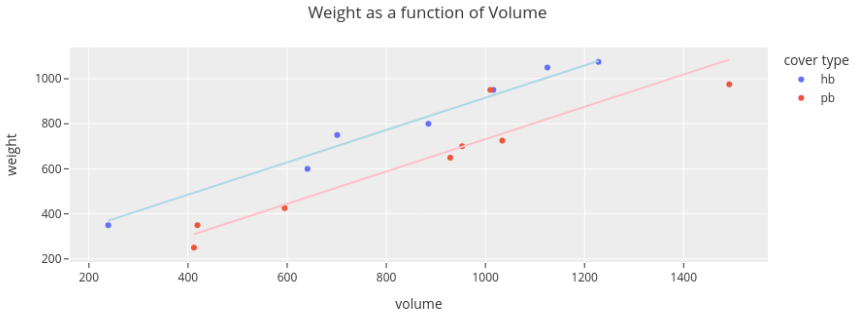
Interpret the result

What does this tell me:

- **All else held constant**, on average paperbacks are 184.046 grams lighter weight than hardback books.
- **All else held constant**, if you increase the volume by one square centimeter then on average it's weight will increase by 0.718 grams.
- Hardbound books with zero volume are expected to weigh 197 grams (extrapolated).
- Paperback books with zero volume are expected to weigh 13 grams (extrapolated).
- We also see that we get a better R^2 when we use two explanatory variables!

Interpret the result

Let's look at a plot of this data. Here I plotted two separate lines one for hard bound and the other for paper back.



Types of effects

- A **main effect** refers to the independent impact of one variable on the dependent variable. This means that each of our variables are in linear combination:

$$weight = 197.963 + 0.718(volume) - 184.047(covertype)$$

so the weight changes at the **same rate** regardless of whether or not we are modeling paperback or hardback books. In both cases as the volume increase the weight increases by 0.718. Does this make sense?

Types of effects

- An **interaction effect** occurs when the effect of one variable changes depending on the level of another variable.

Maybe we should have different slopes for theses lines. Do hardback book get heavier per square centimeter?

Should we try to account for this?

Should we add interaction effects?

In pursuit of Occam's razor

- Occam's Razor states that among competing hypotheses that predict equally well, the one with the fewest assumptions should be selected.
- Model selection follows this principle.
- We only want to add another variable to the model if the addition of that variable brings something valuable in terms of predictive power to the model.
- In other words, we prefer the simplest best model.

Try modeling an interaction

We can do this by engineering some new features (variables) in our data. Let's try adding a variable to our model that is the cover type times the volume. The interaction between cover_hb and volume.

```
DF_model['cov_vol'] = DF_model['cover_hb']*DF_model['volume']
```

Try modeling an interaction

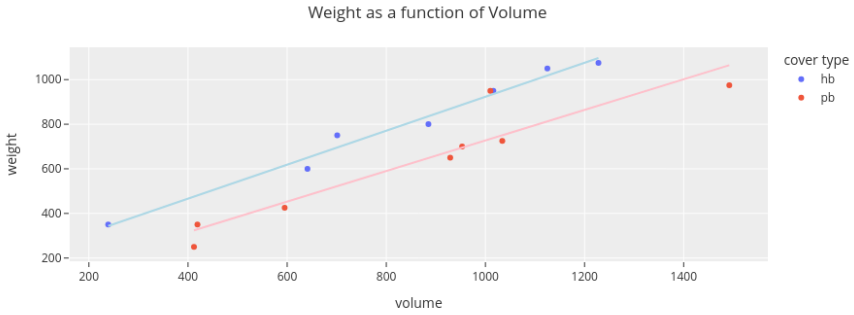
Now we will train a new linear model using three explanatory variables:

$$weight = b_0 + b_1(volume) + b_2(hardbound) + b_3(paperback) + b_4(hardbound * volume)$$

Try modeling an interaction

```
X = DF_model[['volume','cover_hb', 'cover_pb','cov_vol']].valu
Coefs:
[  0.68585918  60.10703285 -60.10703285   0.07573366]
Intercept:
101.4795085635086
Score:
0.9297136949284579
```

Try modeling an interaction



How much did this help?

- The graphs look very similar. In the second graph the two lines have different slopes with the hardbound slope being slightly larger.
- The two models have similar scores:
  - Main Effect Model Score = 0.9274775756821679
  - Interaction Effect Model Score = 0.9297136949284579
  - The interaction model does a slightly better job of explaining the variability in the response variable but it adds complexity.

How much did this help?

What does Occam's Razor say?

R^2 does a good job at looking at one model, but not a great job of comparing models because more variables will tend to make R^2 better. We need to account for the number of variables and the number of observations.

Adjusted R^2

$$adjR^2 = 1 - \left((1 - R^2) \frac{(n-1)}{(n-k-1)} \right)$$

where

- R2: The R2 of the model
- n: The number of observations
- k: The number of predictor variables

This assigned a penalty for putting in lots of extra variables

Adjusted R^2

In our case $n = 15$ or len(DF), $k = 3$ for the first model and $k = 4$ for the second model.

```
# Model 1
Rsq =  0.9274775756821679
k = 3
n = len(DF)
AdjRsq = 1-((1-Rsq)*(n-1)/(n-k-1))

Adjusted R2 for Model 1
0.9076987326863956

Adjusted R2 for Model 2
0.901599172899841
```

Adjusted R^2 results

We see that model 2 has a lower adjusted R^2 value so we are probably better sticking with the original model. It is simpler (uses fewer variables) and gets a good R^2 value!