$\ln(\hat{y}) = b_0 + b_1 x_1 + \ldots + b_p x_p$

$\Downarrow$

$\hat{y} = e^{b_0} e^{b_1 x_1} \cdots e^{b_p x_p}$

$= m_0 m_1^{x_1} \cdots m_p^{x_p}$

(multiplicative model)

if x_1 increases by 1, then y has a proportion increase of b_1. So if $b_1 = 0.25$ and x_1 goes up by 1, then $\hat{y}$ increases by 25%.

if x_1 increases by 1 then $\hat{y}$ is multiplied by m_1.

if x_1 and $\log(x_1)$ are both in the model, then the model is less interpretable.

We talked about polynomials and logs. Augmenting the function space $\mathcal{H}$ with these allow for a model with curves, a model that looks like:

$$g(x_1, \ldots, x_p) = g_1(x_1) + g_2(x_2) + \cdots + g_p(x_p) \Rightarrow \frac{\partial}{\partial x_j}\left[g_i(x_i)\right] = 0$$

this is called a generalized additive model (GAM). What are we missing in this candidate space? The possibility of features interacting with one another. Consider the following transformation:

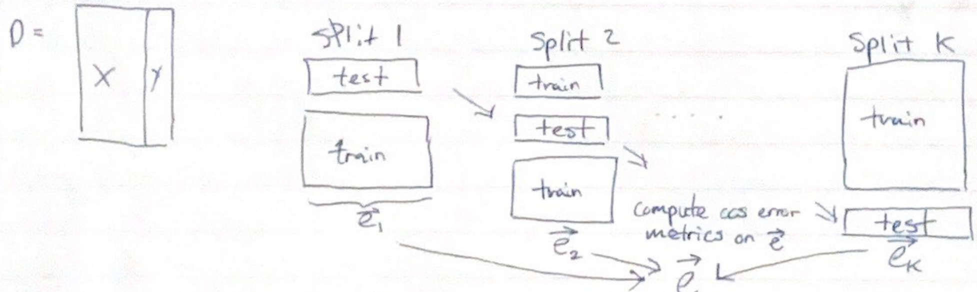
$$X_{raw} = \begin{bmatrix} 1 & x_{11} & x_{21} \\ 1 & x_{12} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} \end{bmatrix} \xrightarrow{\text{transform}} X = \begin{bmatrix} 1 & x_{11} & x_{21} & x_{11} \cdot x_{21} \\ 1 & x_{12} & x_{22} & x_{12} \cdot x_{22} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & x_{1n} \cdot x_{2n} \end{bmatrix}$$

The transformation is called a "first-order interaction". Consider an OLS model on this new design matrix:

$$g(x_1, x_2) = \hat{y} = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1 x_2$$
$$= b_0 + \underbrace{(b_1 + b_3 x_2)}\, x_1 + b_2 x_2 = b_0 + b_1 x_1 + (b_2 + b_3 x_1)\, x_2$$

Let's go back to our discussion about validation. Our validation procedure split the original dataset randomly by taking $1/K\ n$ observations into a test set and the rest into a training set. Thus, the oos error metrics can vary based on the specific random split. The oos error metrics are random variables. And if their variance is high, then our estimates aren't so useful.

How can we reduce the variance in our oos error metrics? Answer: make many splits and repeat the train-test validation procedure. There are many ways to "make many splits". One popular way is called "cross-validation" (CV) or "K-fold CV" which goes like so:



Each observation in the original dataset gets represented once inside of a test set. Each split "crosses over" the test set to the next set of $1/K * n$ indices. This reduces variance because we are averaging many realizations of the rv (the oos error metric).

Another bonus is that we can also compute oos metrics in each of the K splits. For example oos $s_{e_1}, s_{e_2} \ldots s_{e_K}$. So you can gauge the variability of the oos s_e via:

$$s_{s_e} := \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}\left(s_{e_k}-\bar{s}_e\right)^2}$$

this gives you some degree of guarantee of your oos estimate.

$$\Downarrow$$

$$CI_{\sigma_e,\ 95\%} = \left[\bar{s}_e \pm 2\frac{s_{s_e}}{\sqrt{K}}\right]$$

Caution: for this to be valid, the s_e's have to be independent. Are they? No.

mid 1 No... Since they use a lot of the same data. But... we use it anyway.

mid 2 We talked about $K=5$ and $K=10$ being good defaults. What is the tradeoff of K being lower vs higher?