Performance Assessment through Validation

So far we know we have seen ways of improving the performance of a predictor by including more features, and more variables.

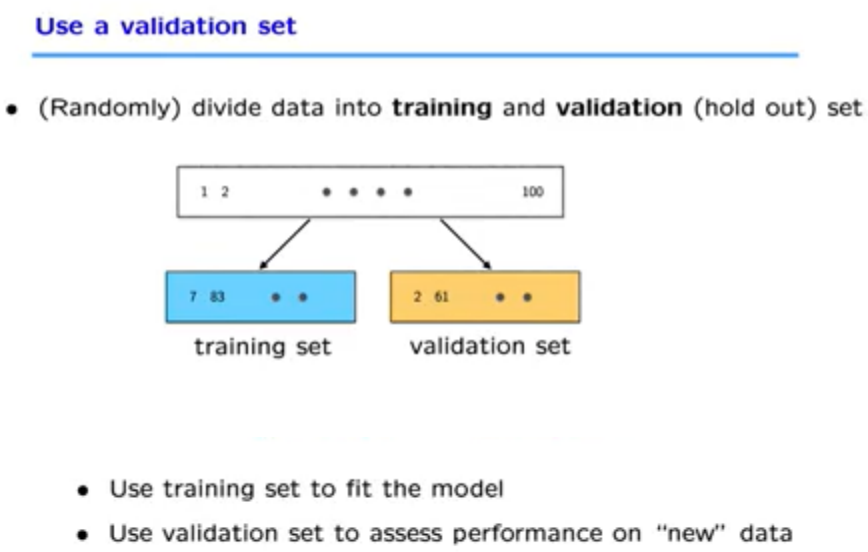
We have seen ways of avoiding overfitting by using regularization. But we end up with lots of possible optimal solutions to different regression problems and somehow we need to choose among them.

So, we need some systematic ways to assess the errors associated with the different predictors and have some systematic way for setting the hyperparameters like those alphas involved in regularization methods. We need some way of choosing which variables to include, and which variables to exclude. We want to try different types of models, more or less complex. We want to choose between different learning algorithms.

How do we do that?
We need some criteria. On the one hand, we're interested in fitting and explaining the existing data, but with this only existing data, we would like to have small errors and have a high R^2. At the same time, we're interested in what is called generalization. We want to perform well on new data and to do that, we want to avoid overfitting.

How do we do this?
There are two aspects to this: one is validation and the other is another method that we call bootstrapping.

What is validation?
We take our data set but instead of running regression using all the data records in our data set, we divide our data set into two parts. One, typically the bigger one, is the training set. These are data records that we will use to run our regressions and we set aside a so-called **holdout set or validation set** on which we're going to check our results.

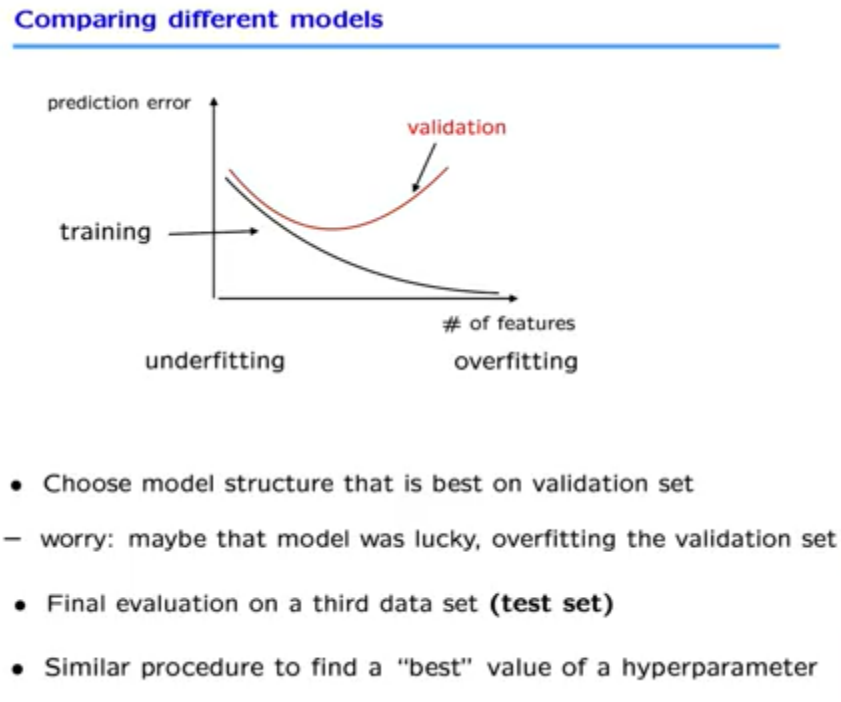


How do we do this division into two subsets?
The usual way is to do it at random. We take our records and that random set some into the training set and some into the validation set. We run our regression, that aspect in the machine learning languages is called training. We use training on the training set to fit a model using some algorithm and then to assess our performance, we rely on the validation set and see what kind of prediction errors we get on the validation set. So, this is like trying our predictors on "new" data that had not been seen when we were training our regression model. So, we can do this to compare different predictors and different algorithms for constructing predictors.

We repeat this process for each candidate predictor and each candidate algorithm and the typical situation that's going to happen is something like this: let us say that we want to compare different models that have different numbers of features and we look at the resulting prediction error. We can look at the prediction error on the training set. On the training set, when you include more and more features, of course, you are going to do better and better. The prediction error is going to decrease and if you have enough features, you can pretty much fit all the data that you have in your data set and get a very small prediction error.

On the other hand, when you test the performance on the validation set, typically in the beginning you are going to get improvements as you get more and more features.

But after some point, you will see that your predictors are going to perform quite poorly on the validation set.
What's happening here is that we have two regimes.



One regime is the underfitting. You're using fewer features than would be natural for the model that you're considering. And so, as you're considering more and more features, you do a better job in capturing the underlying phenomenon and your performance improves. But at some point, when you have too many features, you end up overfitting the data and getting very noisy and unreliable estimates, the theta hats and this is then verified by seeing that on the validation set, you perform poorly.

If you have a picture of this kind, then what you should do would be to choose the particular number of features or the particular number of models that do best on the validation set. So, in this above figure, it would be somewhere around the middle. This is the sweet spot where you get the best performance on new data.
Now here, there is another worry of the overfitting kind. It is a subtle one and it is the following. If you repeat this process using a thousand different algorithms, just by accident one of those algorithms might happen to be good for the validation set that

you have. But this is like overfitting the validation set. Maybe that model would be lucky so, it is usually recommended to do a final evaluation using a third data set, the test set as it is called.

So, when we start with our original data set, we split it into three pieces. One piece is used to do the training, that is, to run a regression, and we run lots and lots of different regressions with different numbers of features on the training set. Then, we use the validation set to choose which one of those regressions seems to be the best one. But before you go to your boss and report what kind of prediction error you expect to have, you should run a final evaluation and the test set that has not been seen before and report the prediction error on that third test set.

So, we discussed this procedure in the context of choosing the number of features, but of course the same procedure can be used to set, for example, the parameter alpha if you are using regularization or in any other algorithm that has some parameters or hyperparameters that you can set and tune. You want to try different choices of those parameters. Train with different choices of parameters, and train on the training set. But then, use the validation set to figure out which choice of the parameter is the best one.

The validation process that we have discussed is attractive in many ways. However, it does have its drawbacks. One drawback is that if we don't have enough data in our data set, we are wasting that precious data. We are setting aside some of them and putting them in a validation set and we are not using them for training. Another drawback is that if we are picking a validation set just once at random, that random choice might be determining which method is going to win and do best on the validation set. But if we were to choose a different validation set, maybe another method would win. So, this randomness in choosing the validation set causes some randomness in our final choice of which particular regression model is going to win. There are certain ways of removing these drawbacks which will be discussed in the next lectures.