Basics- Training, Validation, Test, Cross validation and Out of the Bag score

In machine learning we have the generalization of the data samples but in any other model or data analysis we only have insights pertaining to the observations we have currently.

GOAL - If we generalize good then our model is successful. For example, if we have a data set and we predict cats from dogs then if we go on and predict for any two pictures whether they are a cat or a dog then our **MODEL GENERALIZES WELL.**

How to Generalize well?

Mostly you take a Random Sample from your data and then build your model on the left-over sample and see if your model predicts good for the random sample that you took.

Original set must have at least **22 observations of the target class to have any kind of meaningful model built upon it.**

Reason is based on the fact that the T-distribution becomes Normal for greater than 22 samples.
The reason why THAT actually helps the model, I need to search it.

- **Training Dataset.**

All the data other than that random sample you took in the test set.

- **Test Dataset**

All the data that is in the random sample and set aside for the testing is test data set.

Problem->What if the model doesn't generalize?
That means you can change your model, finetune your hyper parameters, etc. So, let us say it does generalize at some time but it can be the fact that it has only generalized for the test set or you know **OVERFIT** for the test set.

What to do?

- ## Get a validation set.

So, a validation set is like the first layer or checking if our model generalizes and if it does, on the validation set, then we check it on the test set to see it hasn't overfit for the validation.
How to get it? Pick ANOTHER random sample.

- ## Picking the validation set size

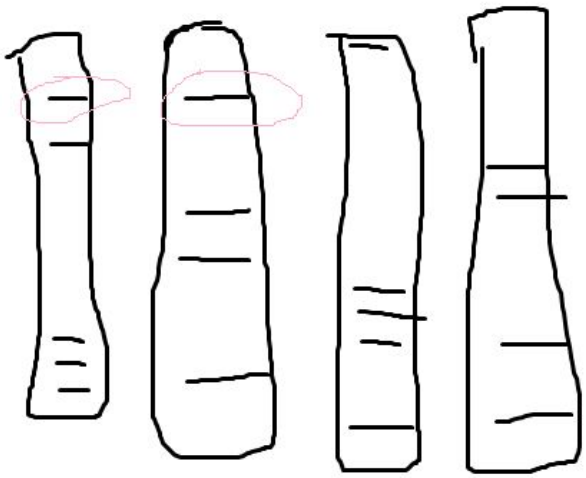
Suppose we are classifying CATS vs DOGS.
If suppose one of the validation set has a 99.1% accuracy and another one has 99.2% accuracy in the validation set. Now see how much your error has changed. Let it be changed from 0.9% to 0.8%.
So, you have to see that the validation set you built would have a 10% less error than the previous validation set.
If you were building a fraud detection model for a company, then a change in error rate from 0.6% to 0.8% would mean a change of 25% and thus 25% cost of fraud.
So, picking up a validation set depends upon HOW ACCURATE YOU WANT YOUR MODEL TO BE.
The minimum number of TARGET variables that you would want in a data set would be at least 22 as at that number your distribution starts behaving NORMALLY and the analysis becomes easier on the data considered.

The basic idea for choosing it would be to choose a set of like 1000 observations, train your model and then look if the accuracy varies or not.
 If it does not then your validation set is good. If it does then we can try increasing the size of validation or using cross validation.

- **The OOB score.**

What does it do? It just says that whenever we choose the rows for a Random Forest tree some rows are by themselves left out of the tree so why not calculate the SCORE based on those rows which were left out.
Then we average that score out for the forest. How? We average out the score from the trees for which that row was not part of the training.

A reason why OOB score is a bit worse than the validation score.



So in this we can see that pink row is left out in the both first two trees , so the oob score is calculated by averaging the prediction for that row across the first two trees and then averaging it out.

This image tells us that every row is going to be using a subset of trees; WHICH SUBSET? That subset which does not have that row as the training row.

SO – in all we see that PREDICTION VALUES ARE ONLY CALCULATED ACROSS THOSE TREES WHICH DON'T INCLUDE IT.

Hence, the prediction is not as accurate as all trees do not see a row for OOB.

Kaggle uses a technique. It splits the test set into a random sample of 30% evaluates your model on it and then it evaluates on the other 70% for the private leaderboard.
At the end of the competition we see our scores get changed as we are evaluated on the private leaderboard.

Why cannot we take a random sample as a validation set?

Some points -

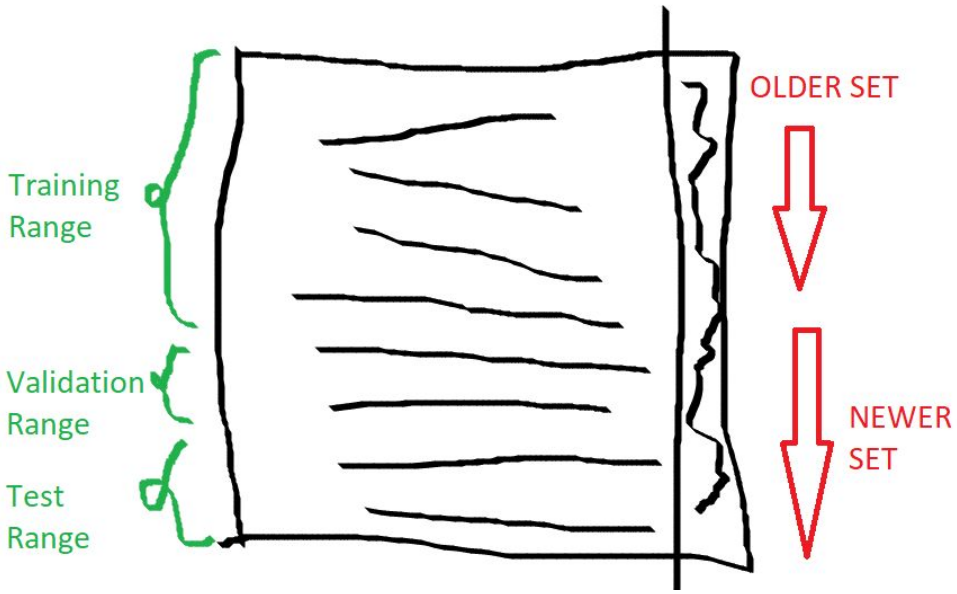
- **We may accidentally remove the temporal or time dependent variable like our model depends on the day before and day before and like that … so if we take a random sample, we destroy that.**
- **Another thing is that suppose we have a random ordering and we have some QUALITY related to a data. For example, we are predicting the lung cancer patients from there details and suppose we have a validation set that doesn't even have the lung cancer patients then your score would not really be a really effective evaluation of model.**

- **VERY IMPORTANT –Why could it be totally wrong?**

Whenever you build a model you have a systematic error that is that you are going to deploy it at a later date than you are building it. Most of the time it matters. What if the world changes while you put it into production?

So, random sampling would not be very good at predicting. Each time we have a temporal piece, assuming we have a sorted by time, then a recent training set would have similar features than the test set that we are predicting.

Now >



NOTE IF OOB IS GOOD THEN YOU ARE NOT OVERFITTING.

HOW TO PICK?

Choose a more recent data set rather than bootstrapping (63.2%).

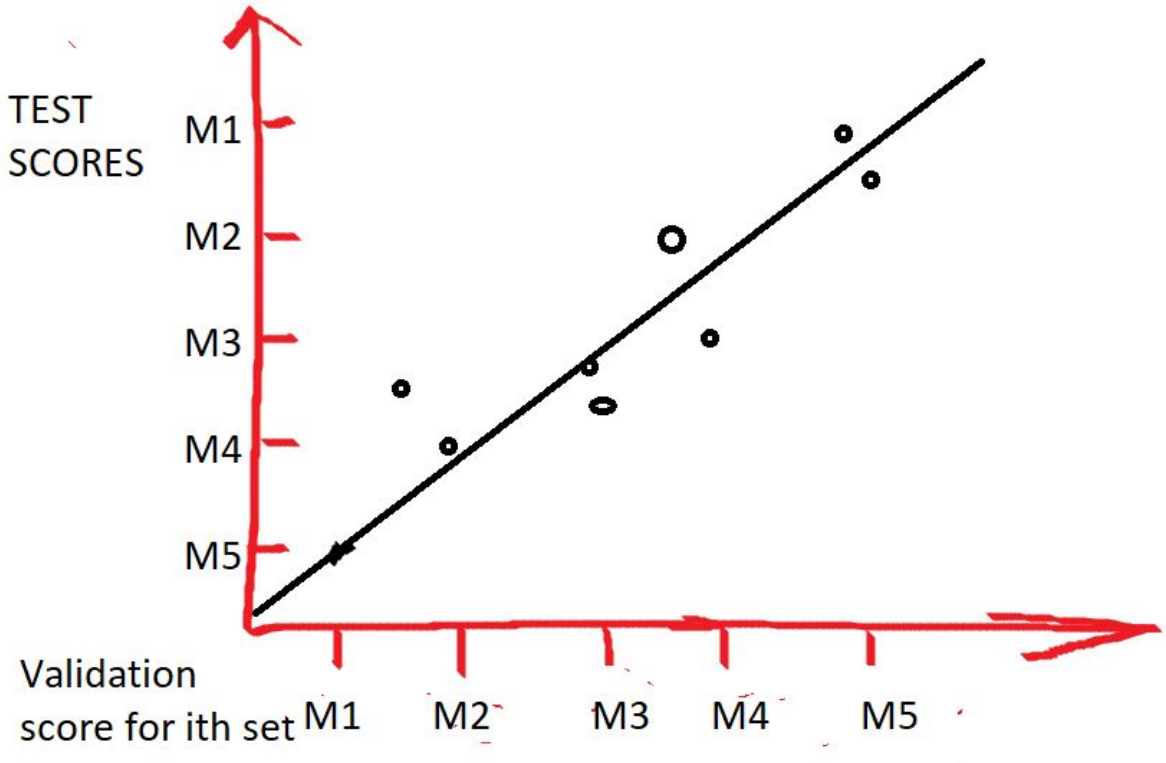
>ONE KEY POINT

When you get a good score on the validation set then YOU SHOULD'T use it directly on the test set as the training would not include the validation range.

YOU NEED TO TRAIN THE MODEL AGAIN AFTER YOU HAVE TRAINED IT FOR THE VALIDATION SET SO THAT YOU HAVE ALL THE DATA TO PREDICT YOUR TEST.

HOW TO KNOW THAT YOUR VALIDATION IS REPRESENTATIVE OF YOUR TEST SET.

- Build multiple models.
- Then evaluate score for the validation set
- Then evaluate the score for the test set.
- Plot them-> like this



So, take different validation sets and build like five models. Now, plot each of the scores of the models of a particular validation set against the test set scores and those scores which form a straight line indicate that you have a good validation set.

Note that you have to retrain the model when you evaluate for the test set.

Suppose you have a timepiece and you want to predict the dependent variable for the next month's sales.

Some possible validation sets can be-

1. The last six months' data.
2. The last 8 weeks data.
3. The last month data
4. All the data
5. All the months previous to your month across all years

And then you could train five different models which can be then tested on all the five sets individually and the scores plotted against the test set scores.

CROSS VALIDATION SET.

BASIC - It just says instead of picking a single validation set pick multiple
Points -

- **Randomly shuffle the data.(important)**
- **Divide into five or so parts.**
- **Take one part as validation other four as training for all the five columns.**
- **This gives you 5 validation sets.**
- **Now we calculate the score and rmse of each of the sets and then average about all the possibilities.**

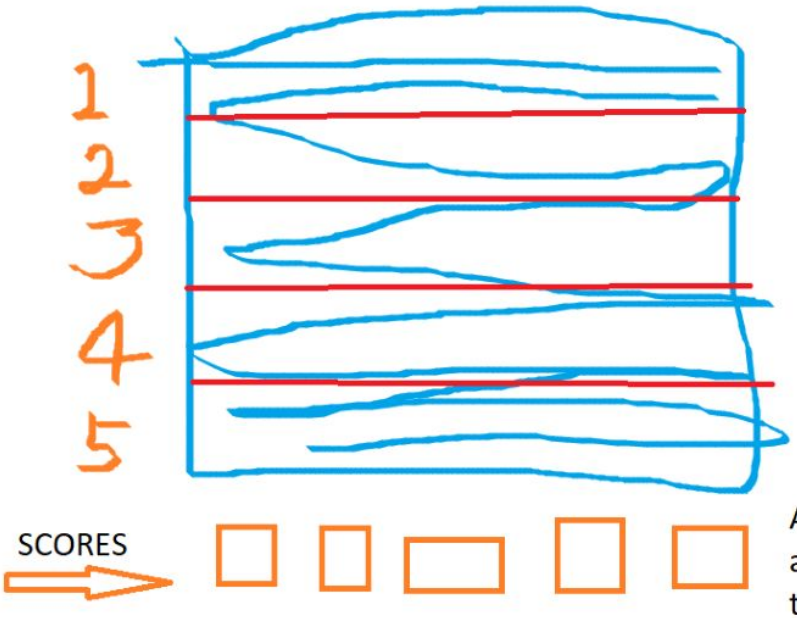
ADVANTAGES

- **You see all of the data as you have different validation sets hence you can see all of the data while you train.**

DISADVANTAGES

- **Obviously, you would require a lot of time to train your models as now you would have 5 sets to train whereas before you had only 1.**
- **Since at the start of our cross-validation prep, we randomly shuffled the data so that means that if you wanted a temporal piece in as your validation set like a future time set then there would be no way to find that property in each of your validation sets.**

FIGURE



- Divide randomly shuffled data set into 5 sets.

- Now suppose take 1 as validation and 2-5 as training set
- Then 2 as validation and 1 and 3-5 as training and so on.

After you get the scores average them out across the five models.