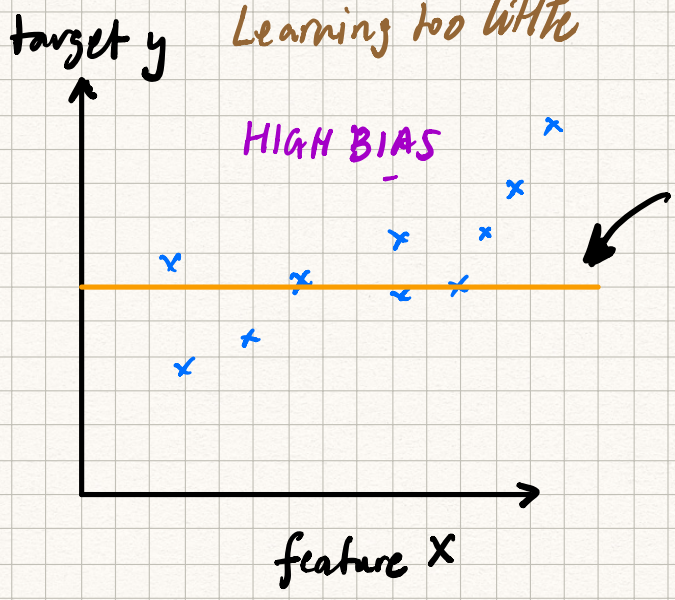
How to Choose and Improve Machine Learning Models

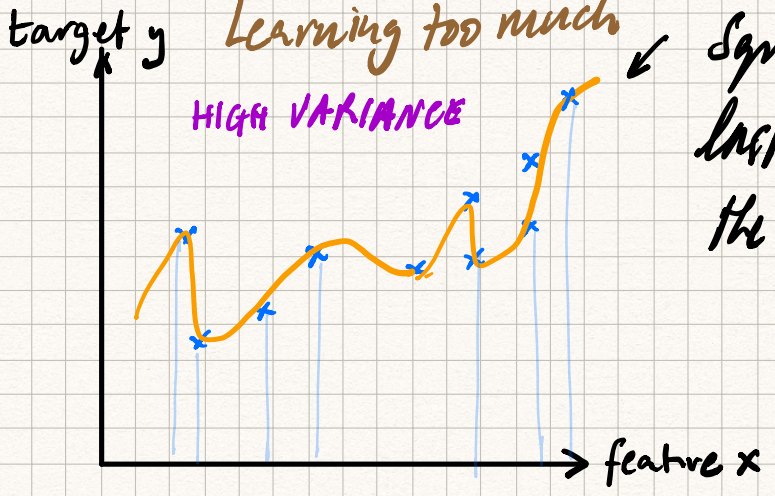
- [x] What models should I try?
- [ ] How do I choose between models?
- [ ] How do I know I have the best model?

What is it to learn from data?

2 (extreme) ways to learn from data



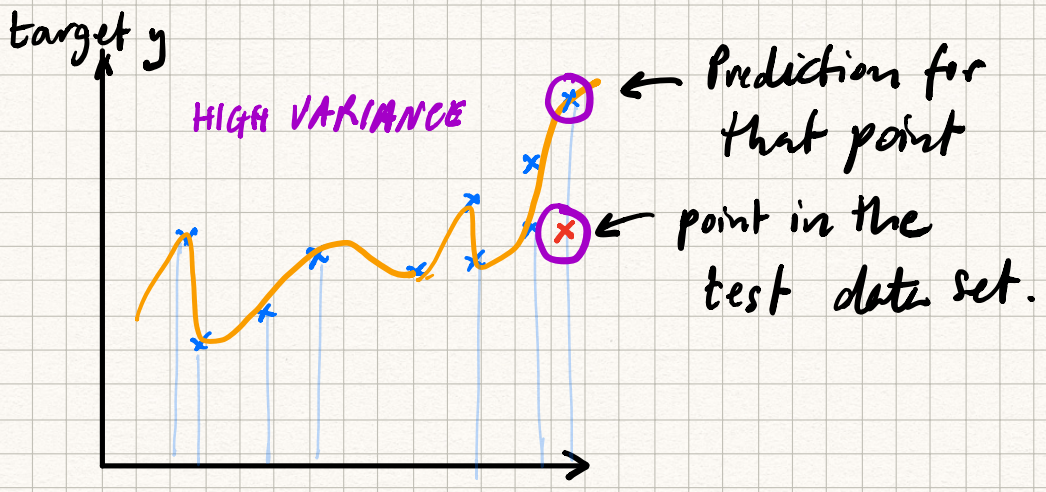
always predict the same value no matter the feature value



Squeezing every last drop out of the data!

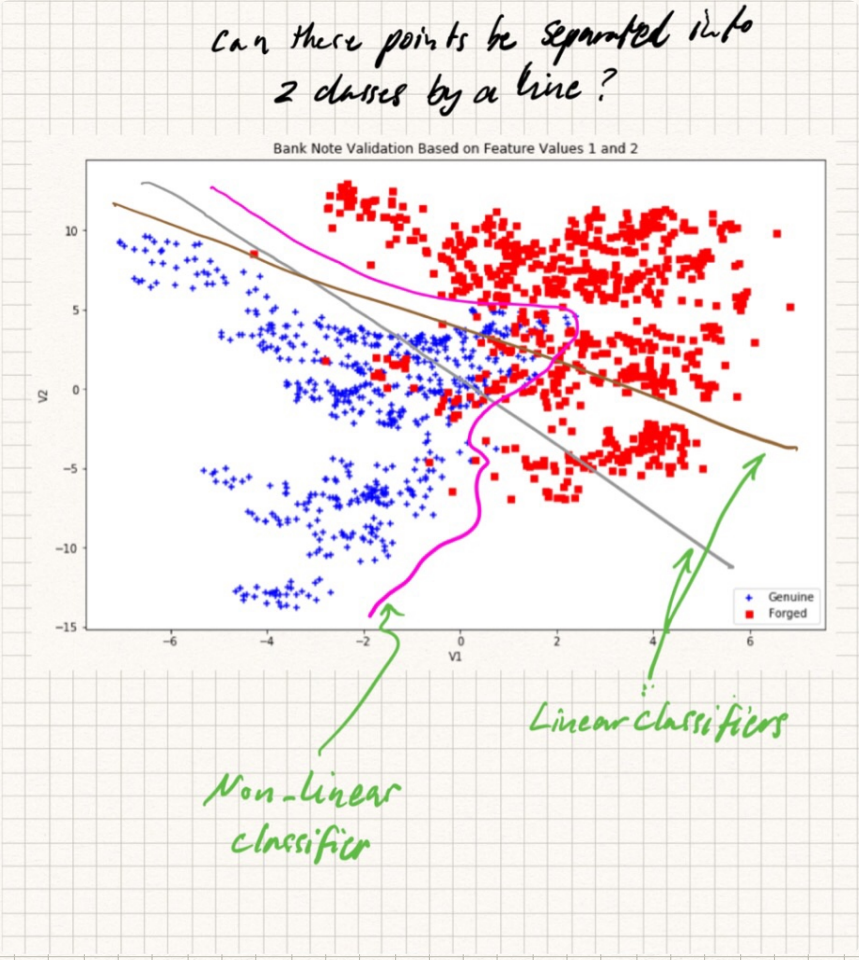
Note: these are <u>not</u> scatterplots.

Why is learning too much a bad thing?



When models learn too much they make (wild) mistakes. This makes them error prone.

Learning too little or too much in classification problems



Models that learn too little

Models that learn too much

1) What models should I use?

What kind of problem is it?

Predict a numerical value

- regression, linear/non-linear
- SVM, linear/non-linear
- Tree
- Neural Network

Predict a class/category

- logistic regression, linear/non-linear
- SVM, linear/non-linear
- Tree
- Neural Network

2) Which of these models should I choose?

Which model performs the best on the TEST data?

How to measure the performance of a model?

For numerical prediction:
- MSE - Mean Square Error
- RMSE - Root Mean Square Error

For class/category prediction:
- F1
- Accuracy (if you know the target values are balanced)

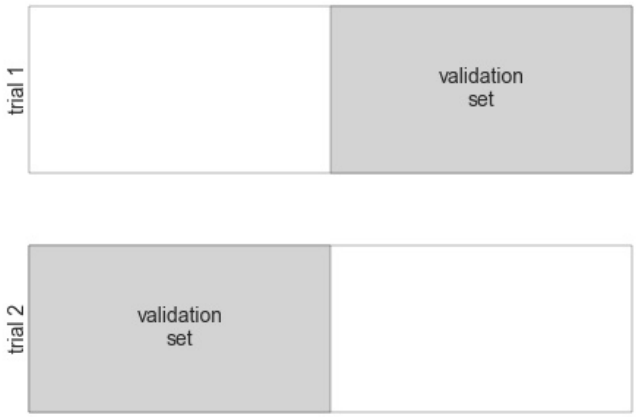
For practical reasons:

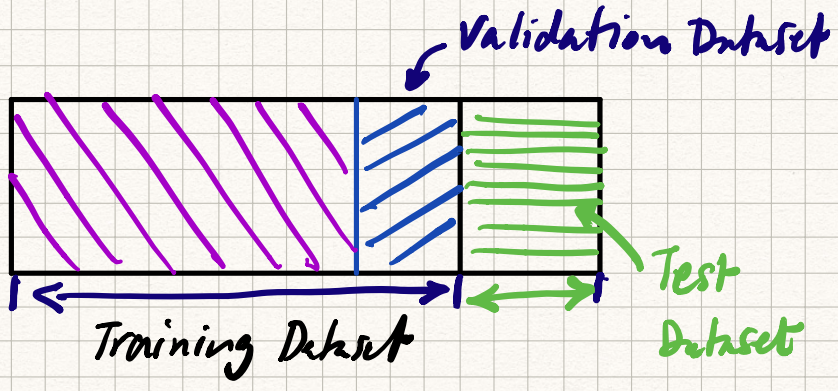
How performance is calculated.

K-Fold cross validation on the TRAINING dataset

2 - Fold Cross Validation

Jake VanderPlas - Data Science Handbook

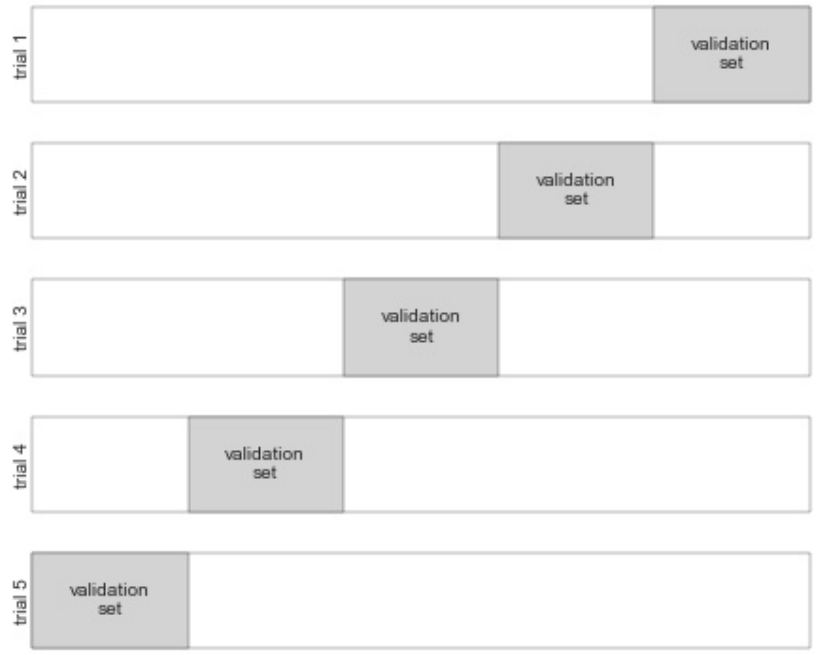




Full Data Set

5-Fold Cross Validation

Jake VanderPlas - Data Science Handbook



3) OK, I know which model is best. But I just used default values for each model.

- What if these defaults change?
- Will the relative performance of the models change?

High Bias = Learning too little

= Low Complexity

High Variance = Learning too much

= High Complexity

Complexity of a Model

The Complexity of a model depends on:

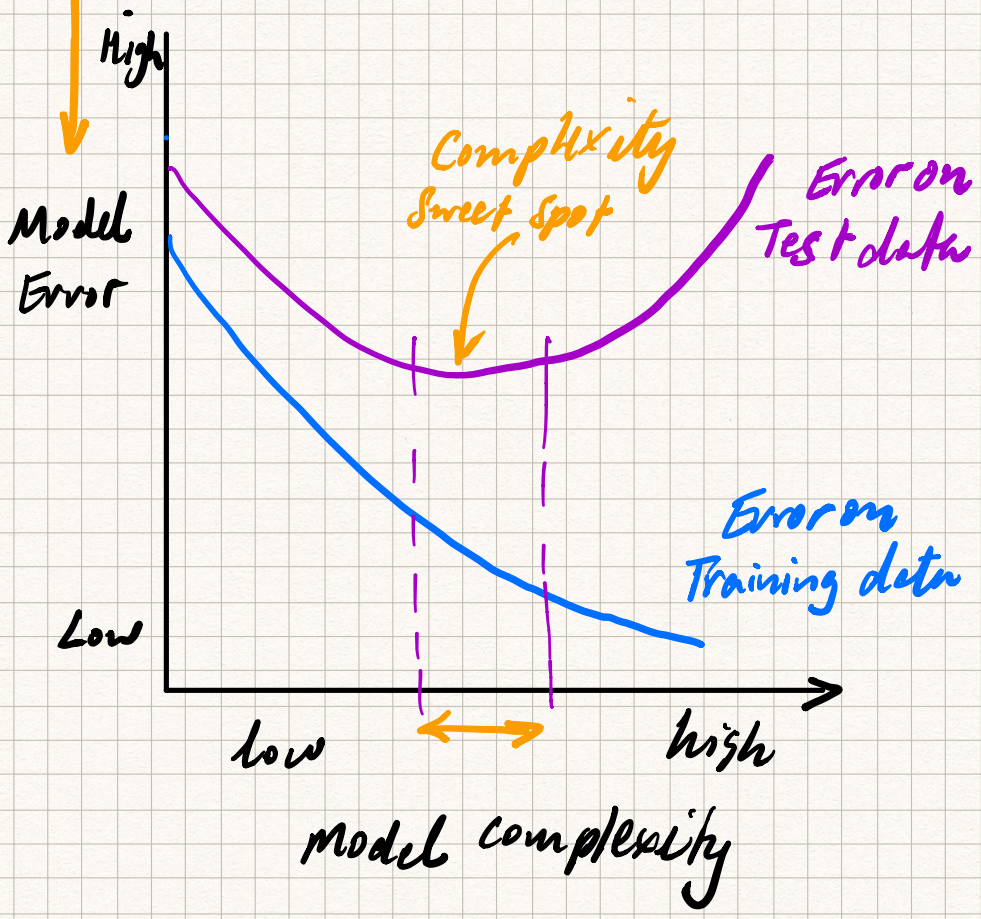
- The features
- The model's hyperparameters
- the amount of training data

Let's look at these in Orange

- Regularization
- Linear / Non-Linear (SVM)
- Layers, # of units (Neural Network)

Validation Curves

Careful: when F1 is high error is low
when F1 is low error is high



Let's see how this is done in Orange.

We've seen how to:

- apply a handful of models
- measure the performance of each model
- increase or decrease the complexity of the model to find its complexity sweet spot.

SUMMARY Use the model tuned to the right complexity that performs best on the test data.

So far:

1) Built a handful of models with "off the shelf" or default settings for hyperparameters and feature complexity.

2) Ranked the performance of these models by comparing their performance measured using k-fold cross validation on the training data set.

3) Sanity checked that these performance values are similarly ranked in the test data set.

4) Fine tuned the complexity of each model using validation curves. (Found the complexity sweet spot.)

End Result

A model that performs best among the ones we started with. Furthermore, this model is tuned to its optimal complexity.

Tuning models is a lot of work!

The 2 most time-consuming activities in data science are:

- Handling data (acquiring data, loading it, cleaning it, maintaining the data pipeline)
- Tuning models to improve prediction performance.

It would be nice to know which "direction" to tune a model.

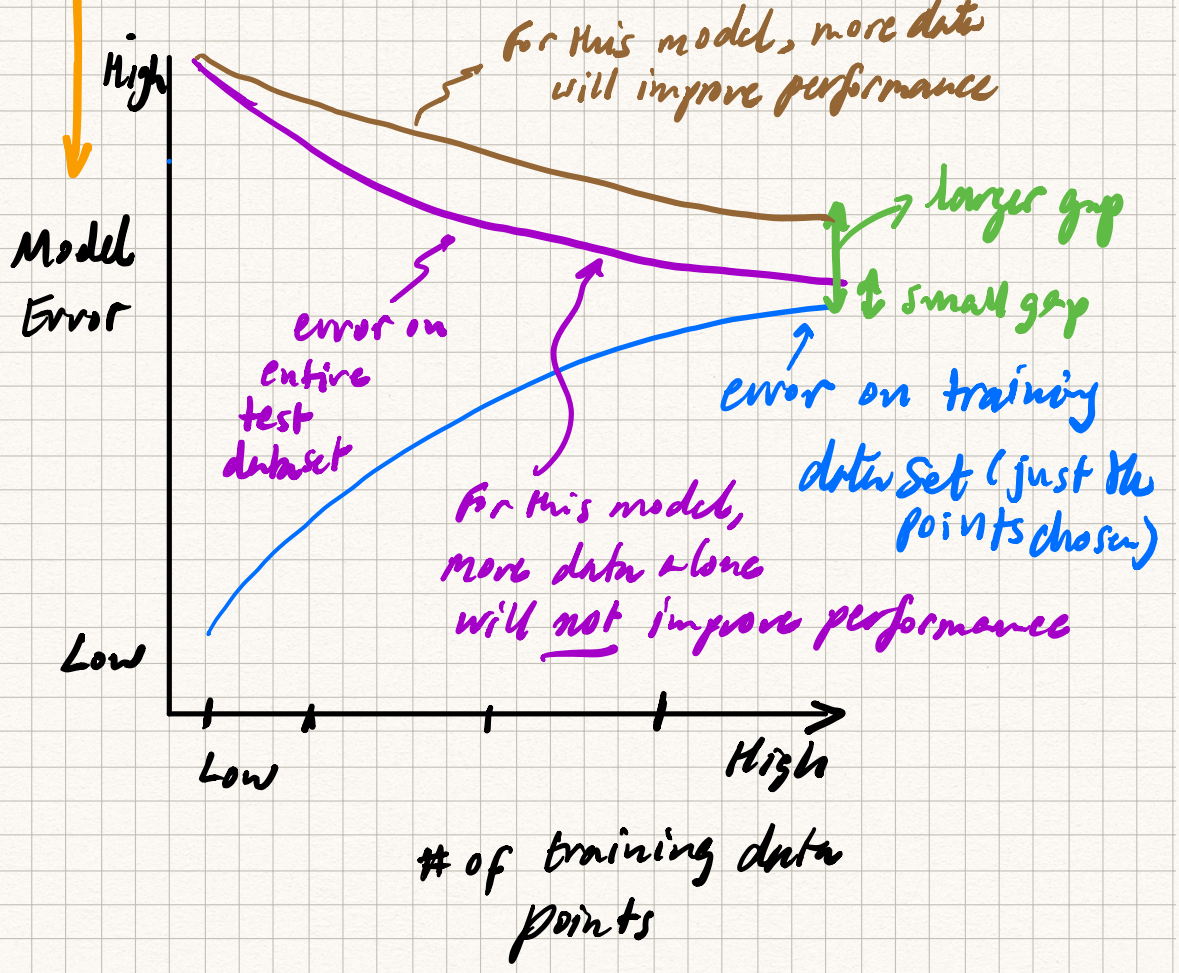
Learning Curves

A way to know which way to tune the complexity of a model.

In other words, a way to find out if a model has high bias or high variance (or neither)

Learning Curves

Careful: When F1 is high error is low
When F1 is low error is high



— High Bias Model (small gap)

— High Variance Model (large gap)

SUMMARY

How to build the right model?

1) Pick a bunch of models off the shelf (default settings) that fit the problem.

2) Use learning curves for each model to find out whether to increase or decrease the complexity of each model.

Summary (contd.)

3) Use the information from the learning curves analysis to tune the complexity of the models. Check the optimal complexity using Validation Curves.

4) Measure the prediction performance of the models using k-fold Cross Validation on the training data set. [See Training, Validation, and Test Datasets.]

Summary (Contd.)

5) Sanity check the performance measures on the test data set.

6) Choose the model that has the highest prediction performance on the test data set.