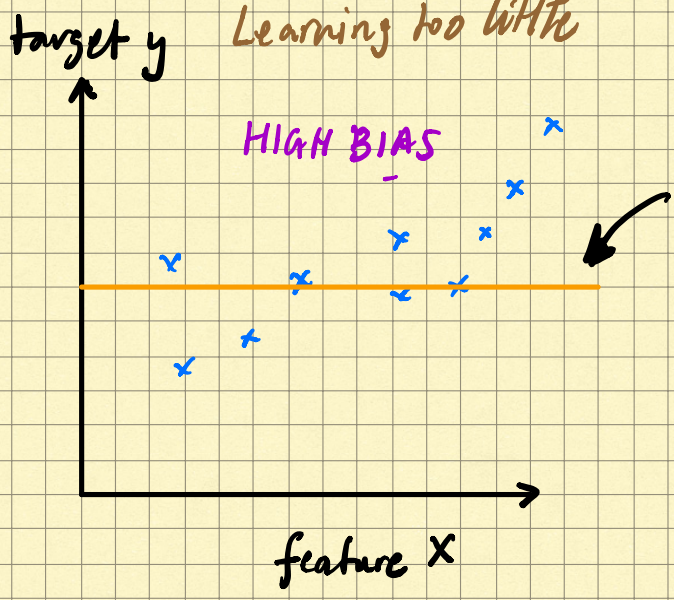
How to Choose and Improve Machine Learning Models

- [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

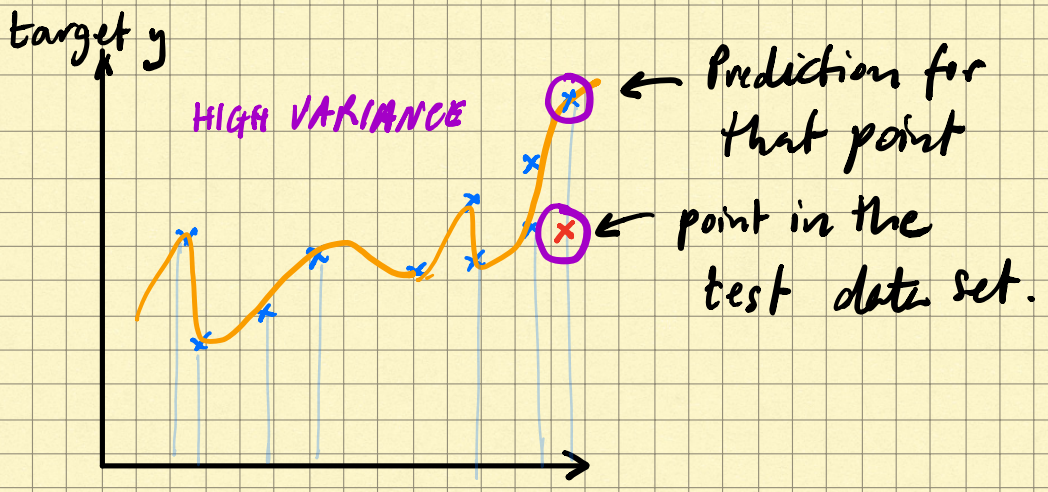


always predict the same value no matter the feature value

Squeezing every last drop out of the data!

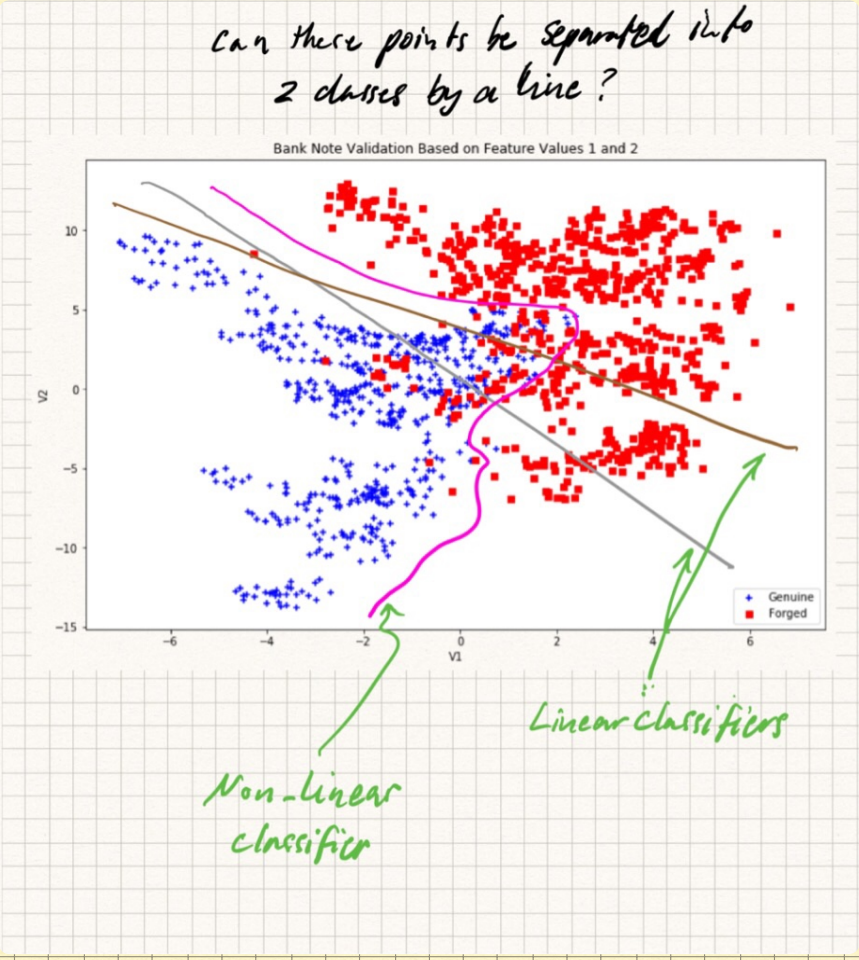
Note: these are not scatterplots.

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

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

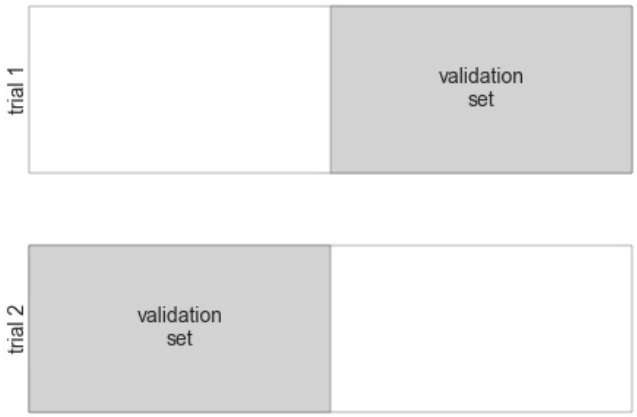
For class/category prediction:
- F1
- Accuracy (if you know the target values are balanced)

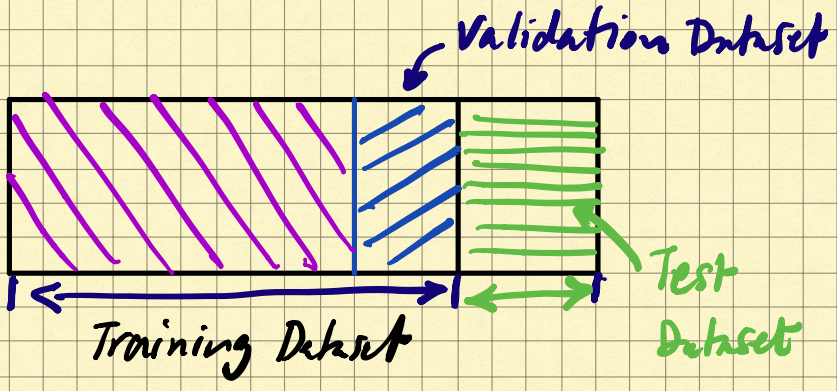
How performance is calculated.

K-fold cross validation on the TRAINING dataset

2 - Fold Cross Validation

Jake VanderPlas - Data Science Handbook

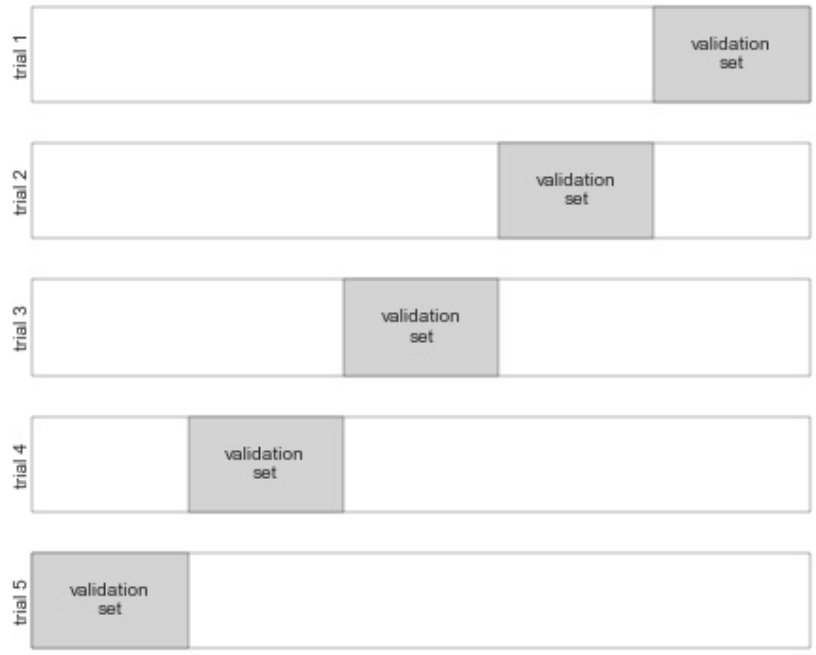




Full DataSet

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

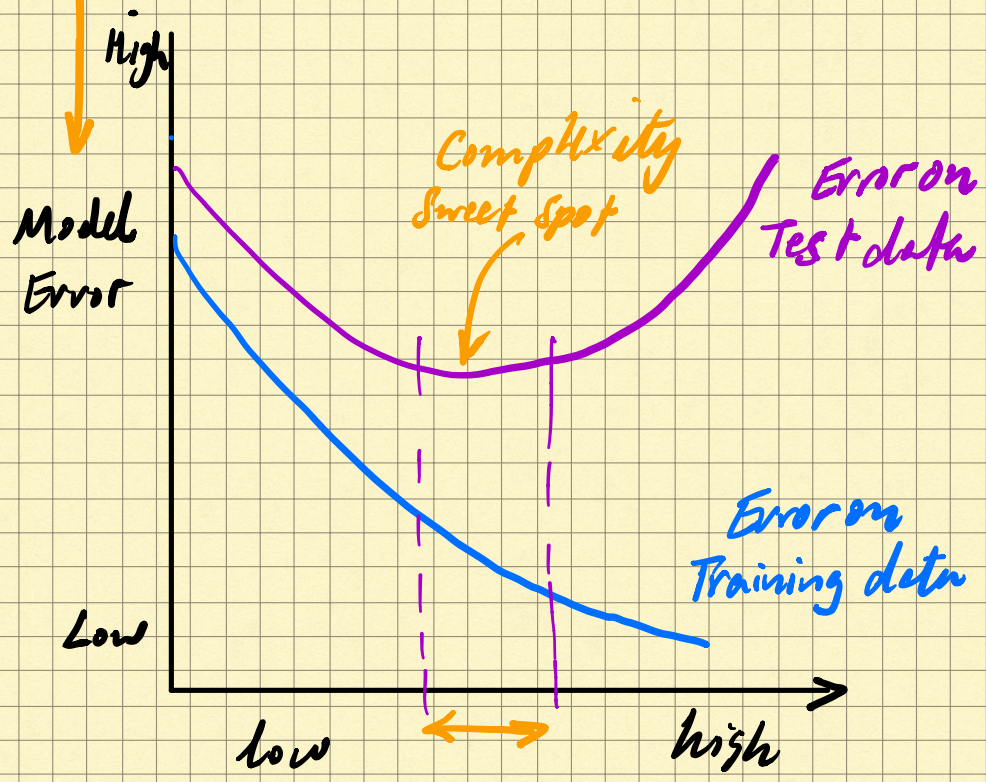
- The features
- The model's hyperparameters

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

One last thing

How complex a model is depends on the amount of training data you have.

A complex model will become less complex (lower variance) when you add more training data.

Even the best model I have isn't performing well.

Should I get more data?

One way to decide is by simulating what the addition of data will do to the performance of a model.

This is called a learning curve.

Learning Curves

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