AI & Robotics

How good is my model?

PXL AI & ROBOTICS LAB

Goals

The **junior-colleague**

- can explain in their own words the difficulty of describing the “goodness” of a ML model
- can explain in their own words the metric used for classification
- can explain in their own words the metrics used for regression
- can describe how to go about measuring the “goodness” of an ML model
- can explain overfitting and underfitting in their own words
- can explain the tradeoff between bias and variance
- can explain the importance of dividing your data sets in training, validation and test sets
- can explain the purpose of the training set
- can explain the purpose of the validation set
- can explain the purpose of the test set

What does it mean for a model to be good?

- How do we measure “goodness”?
- How do we analyze it?
- How can we improve it?

How do we measure “goodness”?

Problem situation: a unique ML problem

- No indication what a good model looks like
- The goodness of a model is relative
- The best that we can do is to compare the performance of ML models on your specific data to other models also trained on the same data

How do we measure “goodness”?

A perfect model is practically impossible!

All predictive modeling problems have prediction error, coming from:

- Missing values
- Noise in the data
- Stochastic nature of Machine Learning

Even robots can be wrong. :*-(

Do not trust AI blindly.

Metrics: Classification

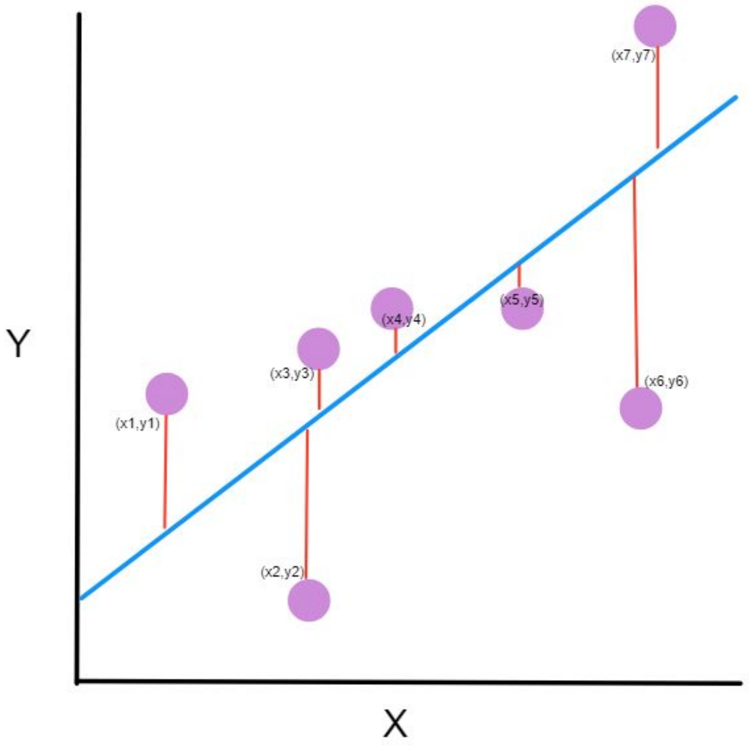
Accuracy

	Actual positive	Actual negative
Predicted positive	True Positive (TP)	False Positive (FP)
Predicted negative	False Negative (FN)	True Negative (TN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

=> 100% accuracy is the best!

Metrics: Regression



Measure the distance between the predicted value and the actual value that the model was trained to predict

(log/root)MSE: the average amount by which the model will be off

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

=> 0.0 error is the best!

Metrics: Regression

Measure the distance between the predicted value and the actual value that the model was trained to predict

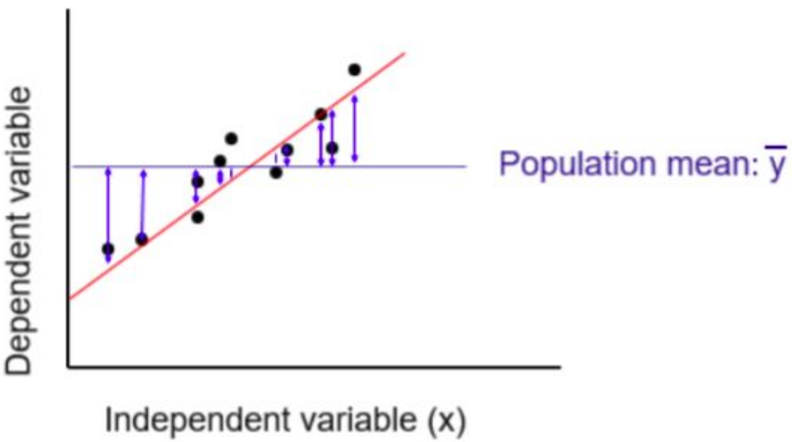
R^2: the proportion of the variance in the dependent variable that is predictable from the independent variable(s)

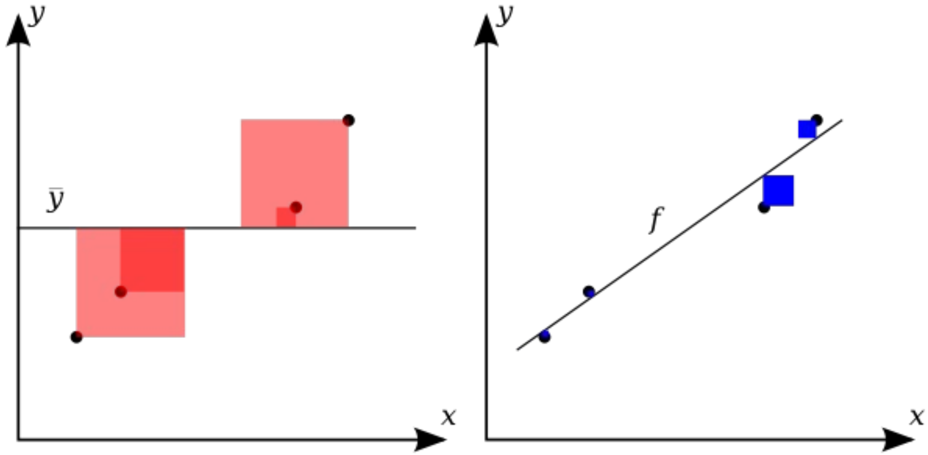
- E.g R^2 of a model is 1.0 => all movements of the dependent variable are completely explained by movements of the input
- E.g. R^2 of a model is 0.50 => half of the observed variation can be explained by the model's inputs

=> 1.0 R^2 score is the best!

$$R^2 = \frac{Explained\ Variation}{Total\ Variation}$$

Metrics: Regression





Explained variation

Alternatively: $$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

But wait, there’s more!

There are even more metrics for different problems (e.g. recommendation systems):

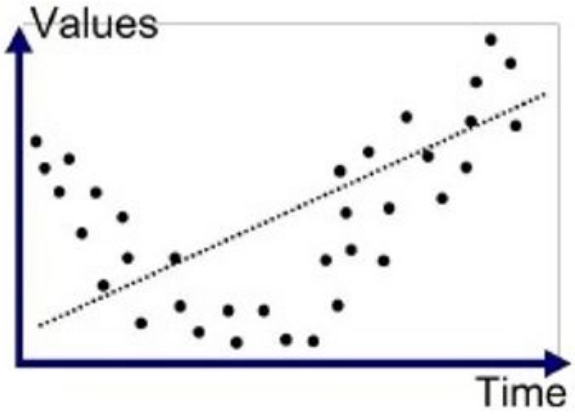
- Precision
- Recall
- ROC analysis
- F1 score
- Etc.

=> will be covered later in this educational programme

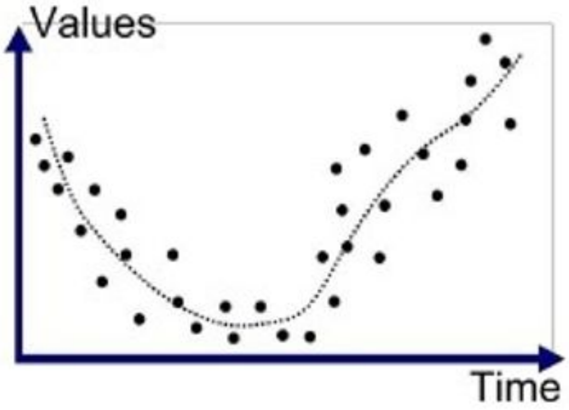
How do we measure “goodness”?

1. Define a baseline
    - Mean outcome value for a regression problem (the average value)
    - Mode outcome value for a classification problem (the value that appears most often)
2. Compare your trained model to the baseline

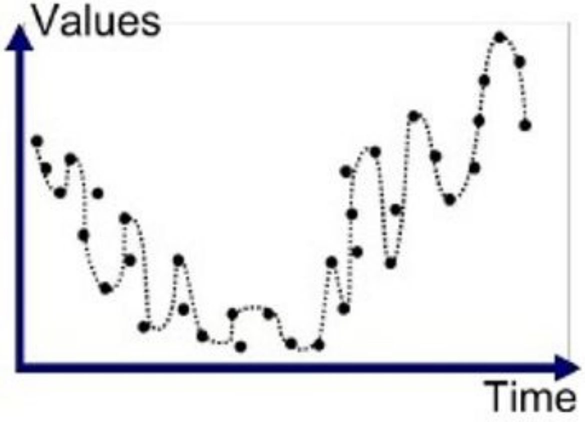
Overfitting vs. underfitting



Underfitted

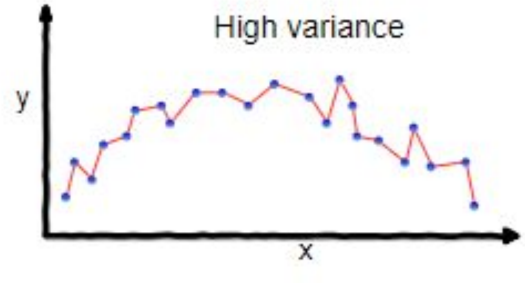


Good Fit/Robust

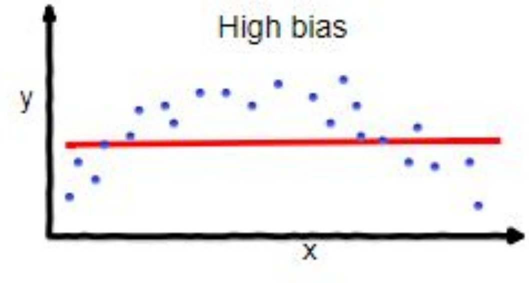


Overfitted

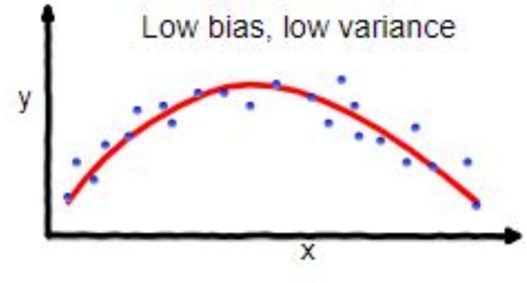
Overfitting vs. underfitting



overfitting

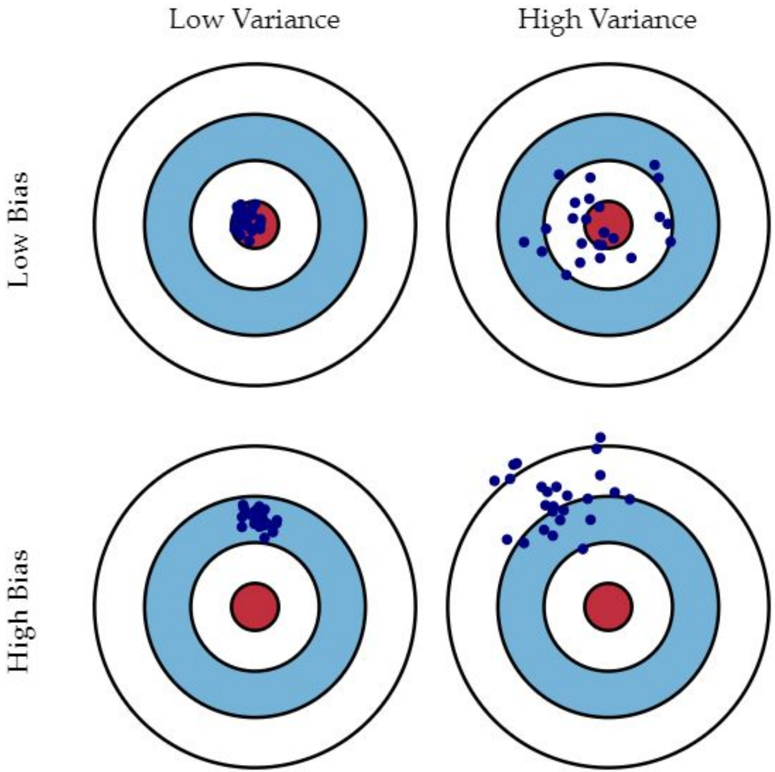


underfitting

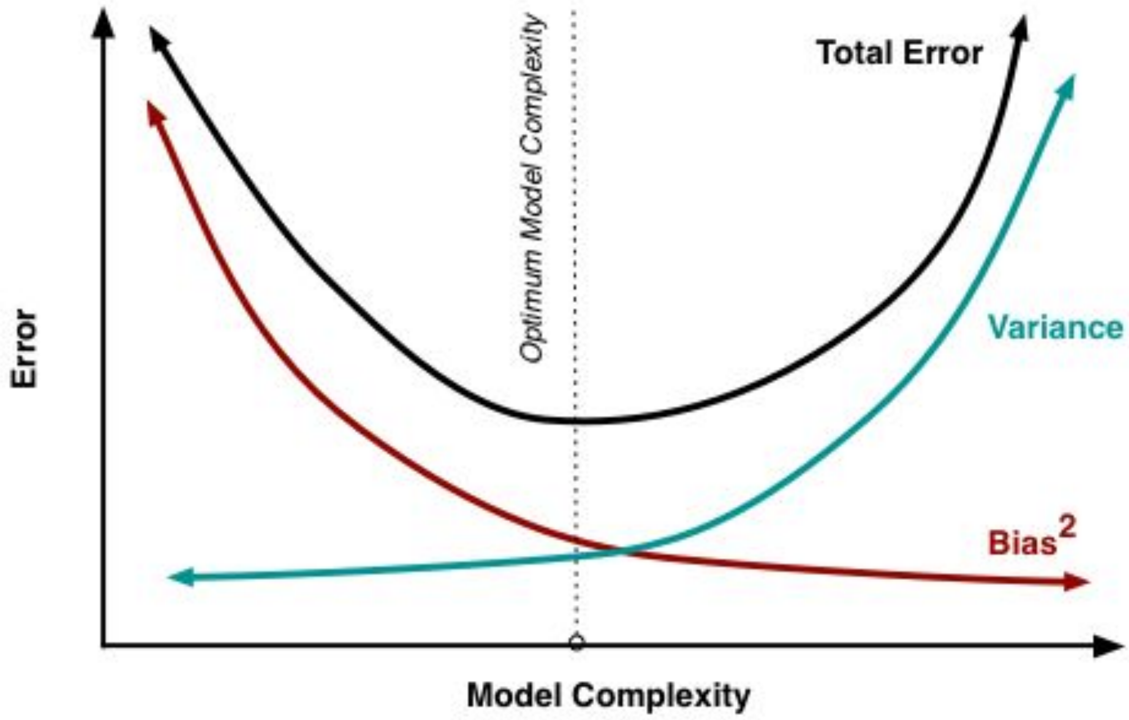


Good balance

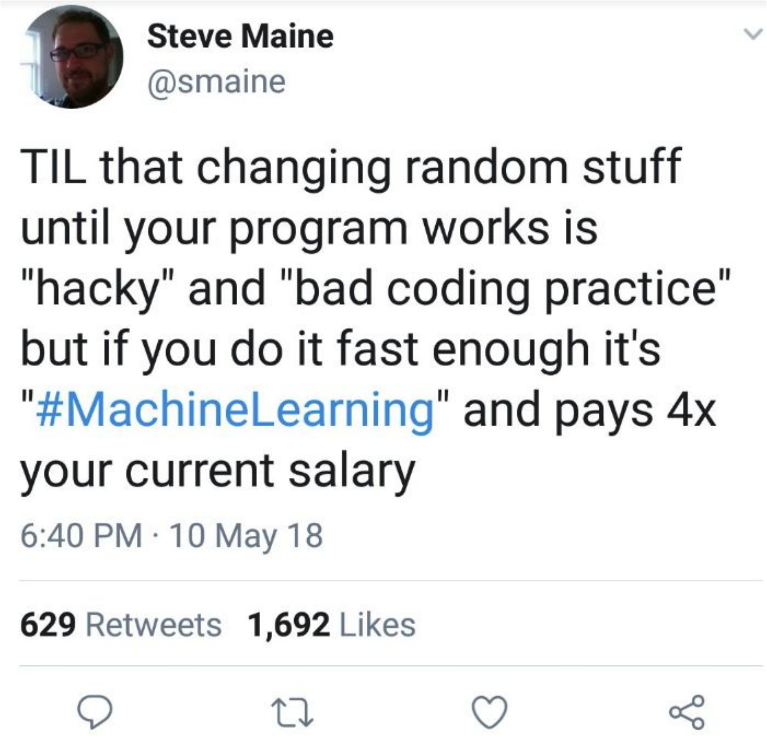
Overfitting vs. underfitting



Overfitting vs. underfitting



Overfitting vs. underfitting

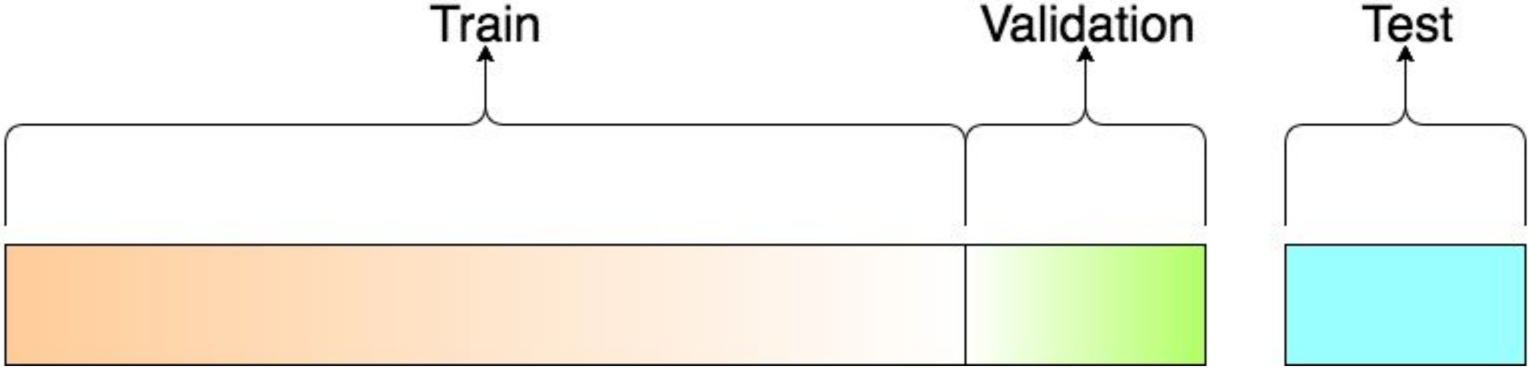

Steve Maine
@smaine

TIL that changing random stuff until your program works is "hacky" and "bad coding practice" but if you do it fast enough it's "#MachineLearning" and pays 4x your current salary

6:40 PM · 10 May 18

629 Retweets 1,692 Likes

Train, Validation, Test



Train, Validation, Test

Training set

- The dataset used to train the model
- The model learns from this data

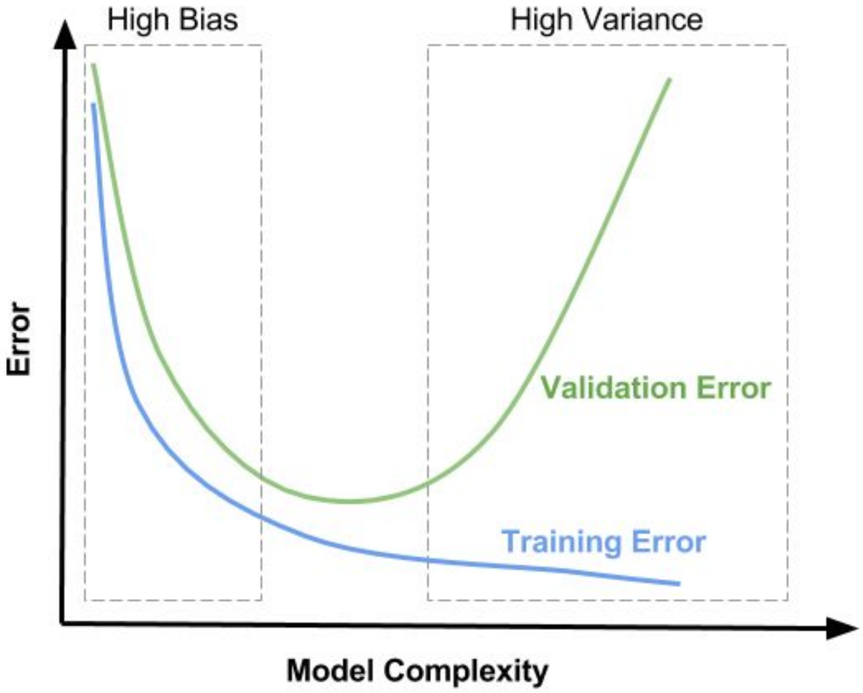
Validation set

- The dataset used to evaluate the model during training and optimization
- Used while fine tuning the model hyperparameters
- The model doesn’t learn from this data

Test set

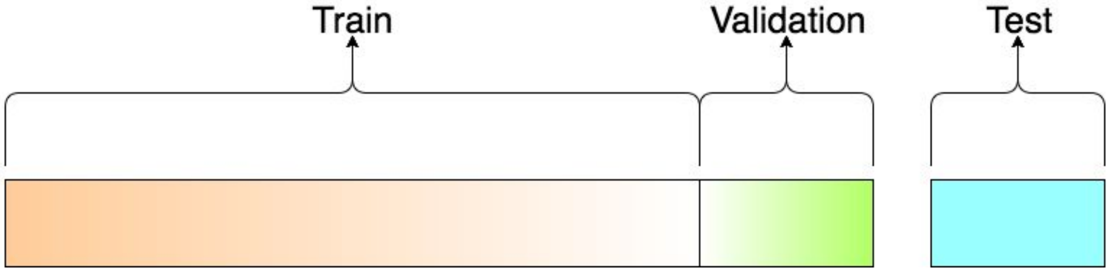
- The dataset used to evaluate the model after training
- The model doesn’t learn from this data

Train, Validation, Test



- Training error =/= Validation error
- Use the validation set to tune the model

How to split?



- Full set of samples: usually [80/20]::[train/test]
- Training set: usually [80/20]::[train/validation]
- It’s important to choose good test and validation sets
  - Make sure there's a good spread in the test samples
  - Make sure your validation set conforms to the test set

=> more on this later