Model Selection & tuning

So far, we have covered some basics of ML as well as some basic algorithms. Next, we need to learn how to evalute a model, how to choose a model and how and what to tune. This is what we will cover here.

* Evaluation of a model
  - In vs Out - sample performance
  - Metrics

* Model analysis
  - Right amount of complexity
  - Validation curve & learning curve

* Tuning the hyperparameters of a model

Metrics

Regression: MSE: $\frac{1}{n_s} \sum_i \left(Y_i - f(x_i) \right)^2$ → loss → Metric

MAE: $\frac{1}{n_s} \sum_i |Y_i - f(x_i)|$

....

Classification:

Accracy: $\frac{1}{n_s} \sum_i Y_i == f(x_i)$: # instances that were correctly predicted.

Recall & Precision

Consider a situation where one of the classes is different in significance, e.g. entangled states vs non-entangled states.

Fake news vs else

⋮

* For instance, we may be interested in predicting fake news than getting both classes right. → Acc may not be the right metric.

* One class has a larger population. 0.1 vs 0.9 → Then → Acc is not a good metric.

Confusion Matrix:

For a binary classification problem we can draw this table for a predictive model.

	Predicted 0	Predicted 1
True 0	A	C
True 1	D	B

→ Now, assume that we are more interested in class 1 (e.g. it is less frequent).

Consider the following situations:

* It is critical that if we guess class 1, it is the right/precise guess. → $\frac{B}{B+C}$ should be high.

→ Precision

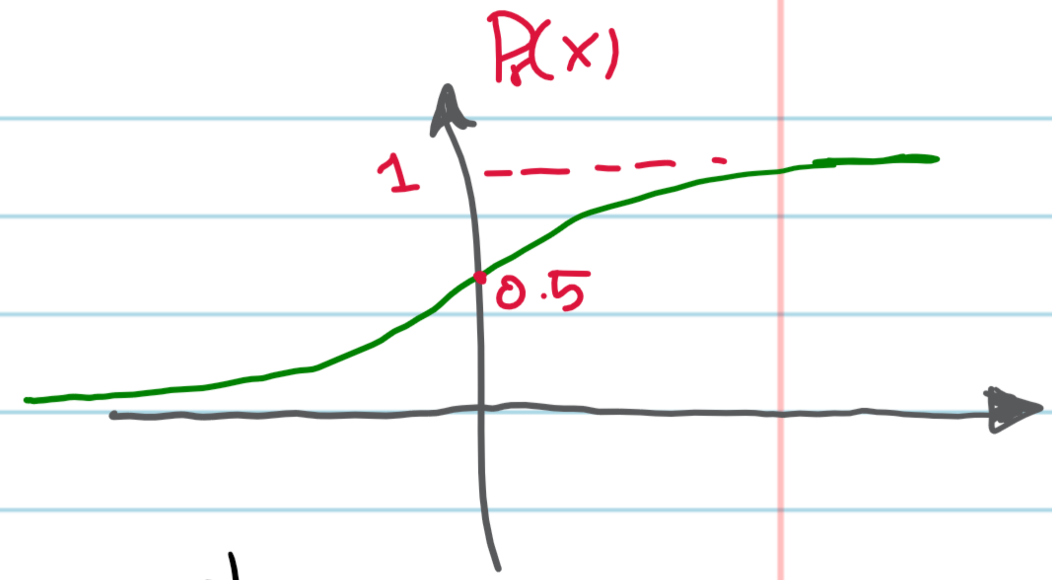
* It is important that we detect all the instances in class 1. → $\frac{B}{B+D}$ should be high.

→ Recall / Detection rate.

* We want to have both to some extent. →

F1_Score

How to optimize for a certain metric?

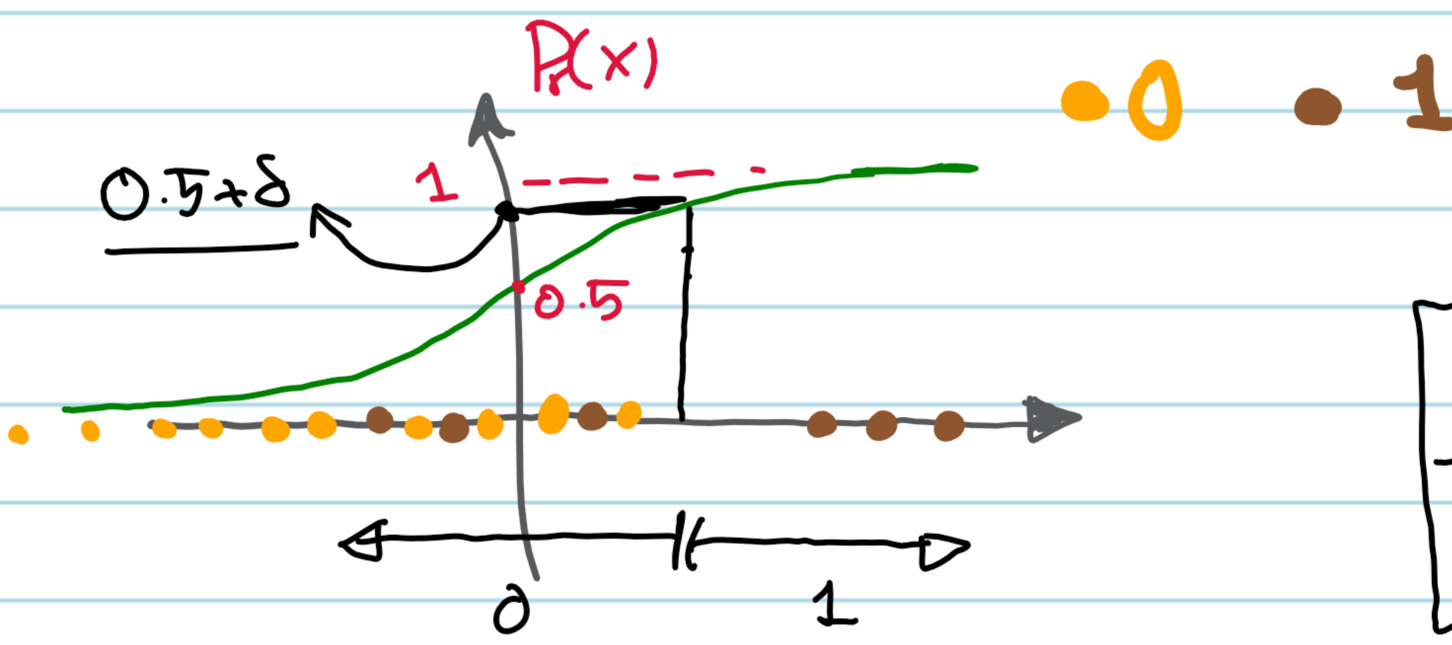


Consider a logistic reg. Classifier

Typically we optimize for acc and $f(x) \geq 1/2 \to$ class.

But to improve the precision (for class 1) we can set

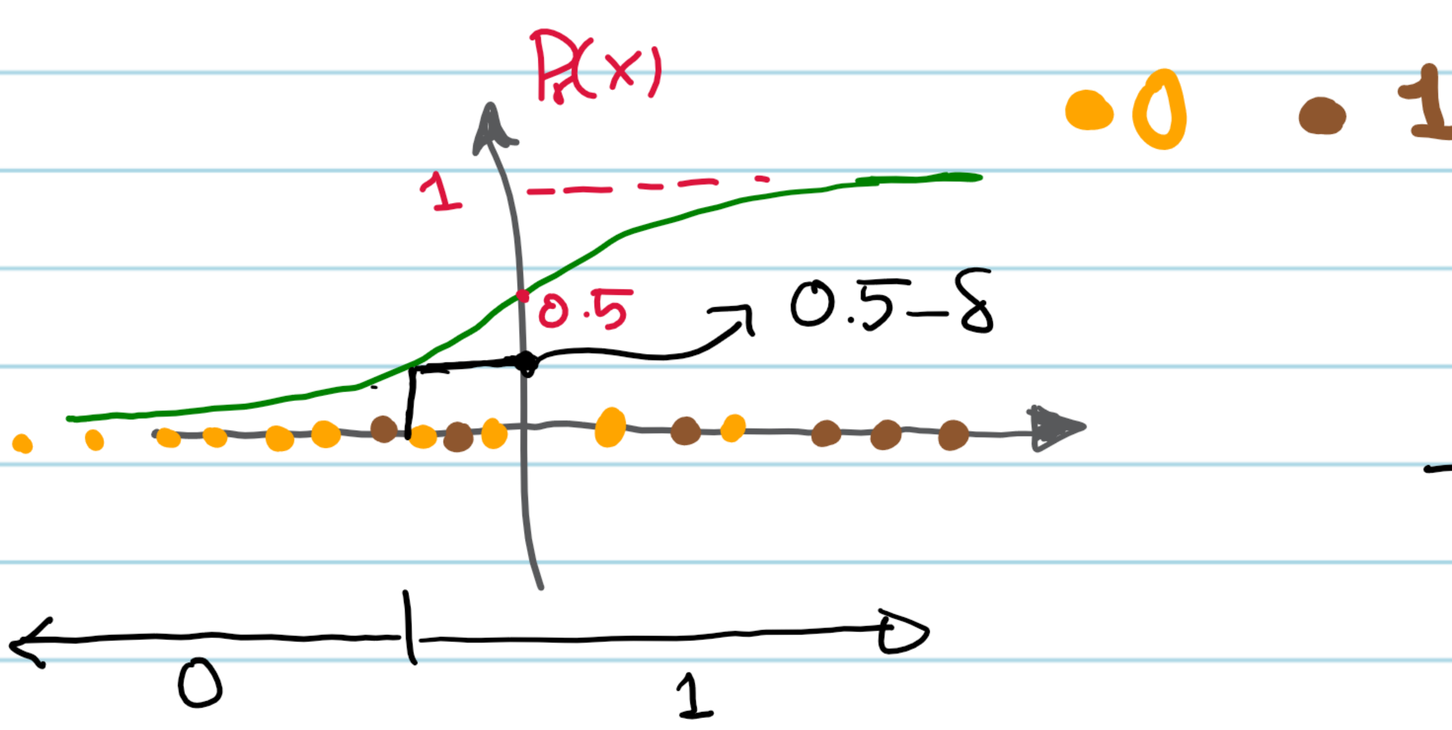
$$f(x) \geq 0.5 + \delta$$



10	0
3	3

Or if we want to detect all the instances of class 1,

we can set it to $f(x) \geq 0.5 - \delta$



6	4
1	5

Recall $\frac{5}{6}$

$$F_\beta = \frac{P.R}{\beta^2 P + R}(1+\beta^2)$$

$$F_1 = \frac{P\,R}{P+R}$$

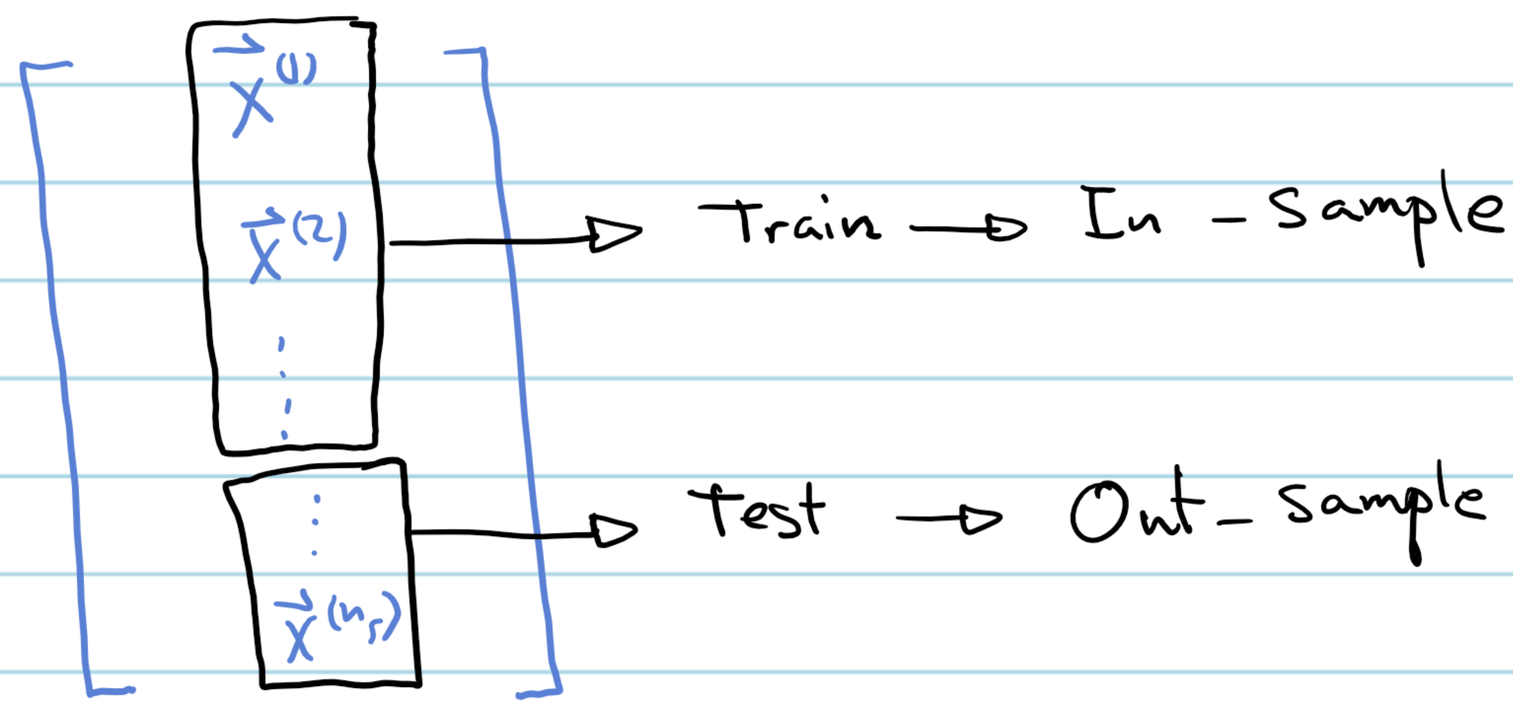
Also look at

Precision_Recall curve

ROC curve

AUC "

In & Out-sample performance



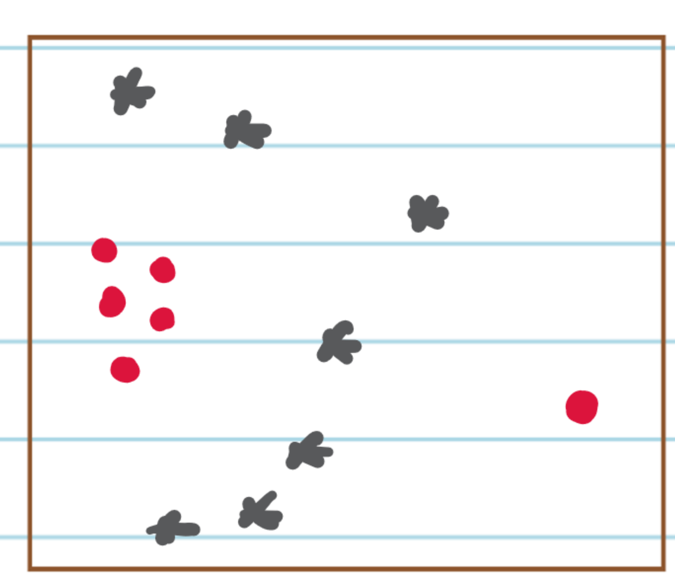
Ideally, we want models to have high performance on both train & test data.

A critical assumption here is that both test & train set come from the same distribution.

→ So we need to shuffle the data before splitting the data into train & test.

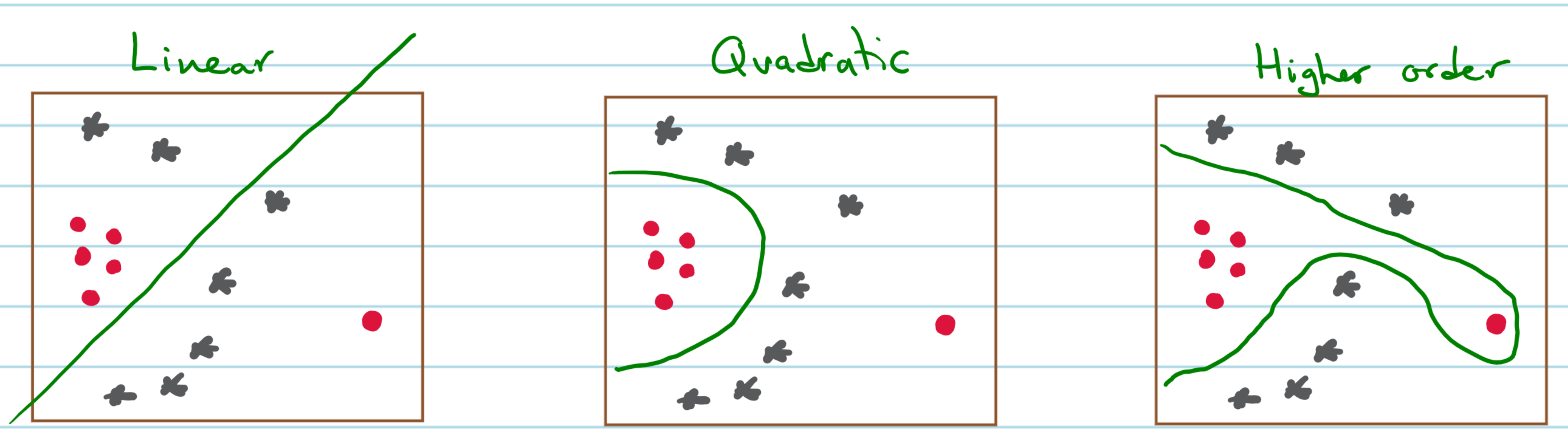
Overfitting & Underfitting

Consider the data in this picture.



There are two classes (red & gray).

Consider the following classifications:



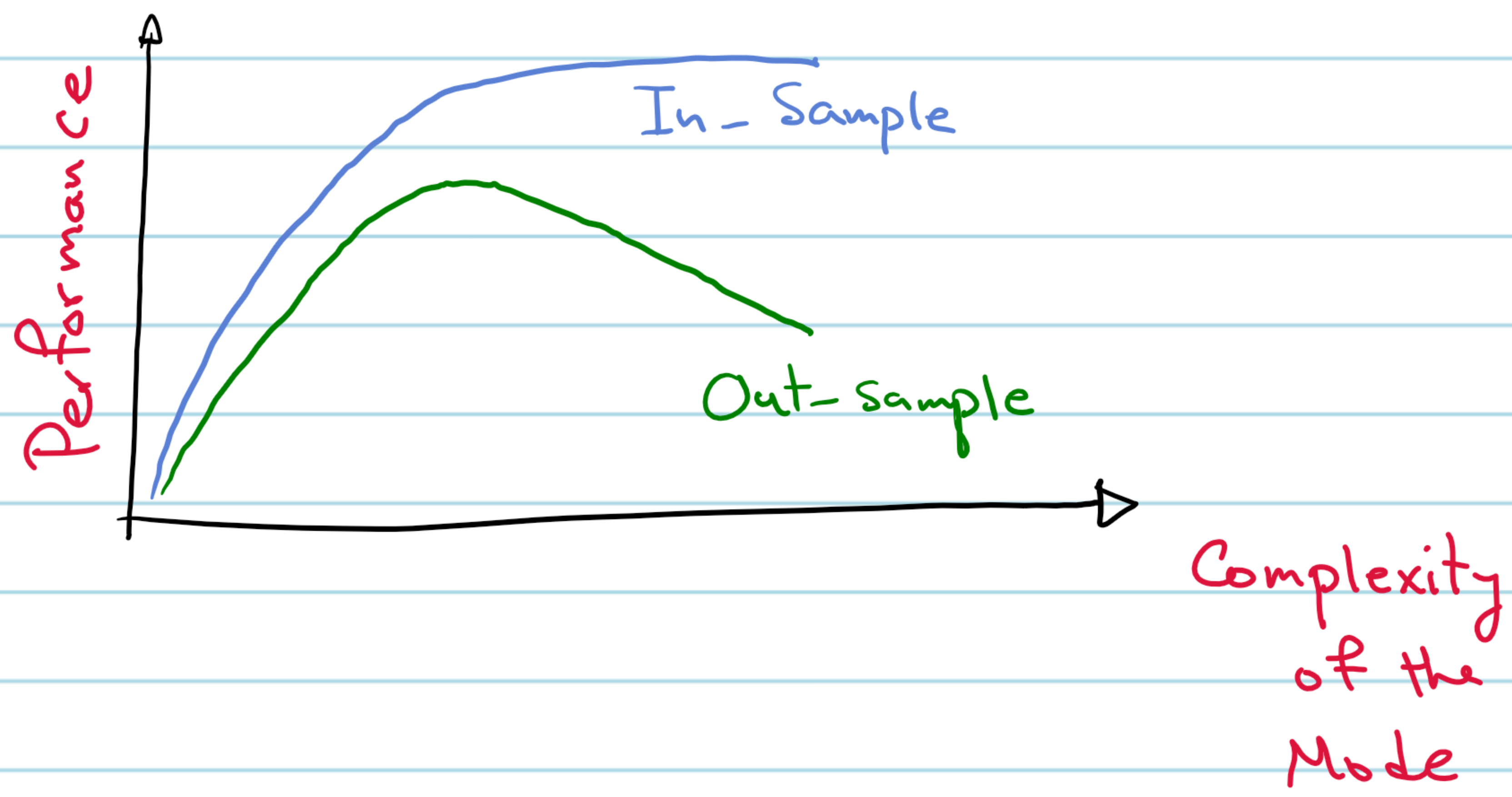
Under_fit — High Bias

Over_fit — High Variance.

How do you guess these models do in each category?

	Linear	Quadratic	Higher orders
In-Sample accuracy	Low	High	Very high
Out-Sample accuracy	Low (close to in-sample)	High (close to in-sample)	Low

Typically



This give a notion of where to stop.