Diagnóstico de Problemas en los Modelos

TC3006C

Dataset division (splitting)

Problems?

- Data selected usually has noise or randomness
- Randomness can be:
  - Explicit in the algorithm
  - Explicit in the selection of training data
  - Implicit in data
- Randomness produces variance (stochastic)
- **Model Variance**: every time we split a dataset, we usually get different accuracy results

Solution A – Repeat several times

Procedure

- Split dataset several times
- Register accuracy each time
- Record the average accuracy and standard deviation

Problems

- Some data instances may have not been selected
    - Either for training or testing
- Some other instances may have been selected several times
- May skew results

Solution B – Cross validation

- Create **folds**: split data into *k* subsets
- Train *k* times
  - Each time a different fold is used for testing
  - The other *k-1* folds are used for training
- Each instance is used an equal number of times for testing and for training
- Useful when data is not large
  - With large data a "normal" split is enough

Example cross validation k=5

Solution B – Cross validation…

Advantages

- Gives an unbiased estimation of an algorithm's performance
- Helps to understand how the model is generalizing
- Helps in Hyperparameter Tuning

Disadvantages

- Increases training time
- Needs expensive computation
- Cross validation itself uses randomness

Solution C – Multiple Cross Validation

Procedure

- Run Cross Validation several times
- Record mean and standard deviation

Disadvantages

- Computationally expensive

Cross validation is for checking

- Cross validation allows us to check how good is our model for learning on a given dataset
- It helps to compare several models (for instance, kNN vs linear regression)
- Based on the result of cross validation, we can select the best model
- When building the final model, we use all our data for training

Bias, variance and fitting

Prediction Errors

Bias

- Difference between the real value to predict and the value being predicted
- Bias can be seen as error in test data
- Error introduced due to wrong assumptions made on nature of data
    - Assuming is linear when is quadratic
- High bias models:
    - Oversimplifies model
    - Not accurate with real data (misses data points)
    - Misses relations between target and features
    - Usually means underfitting

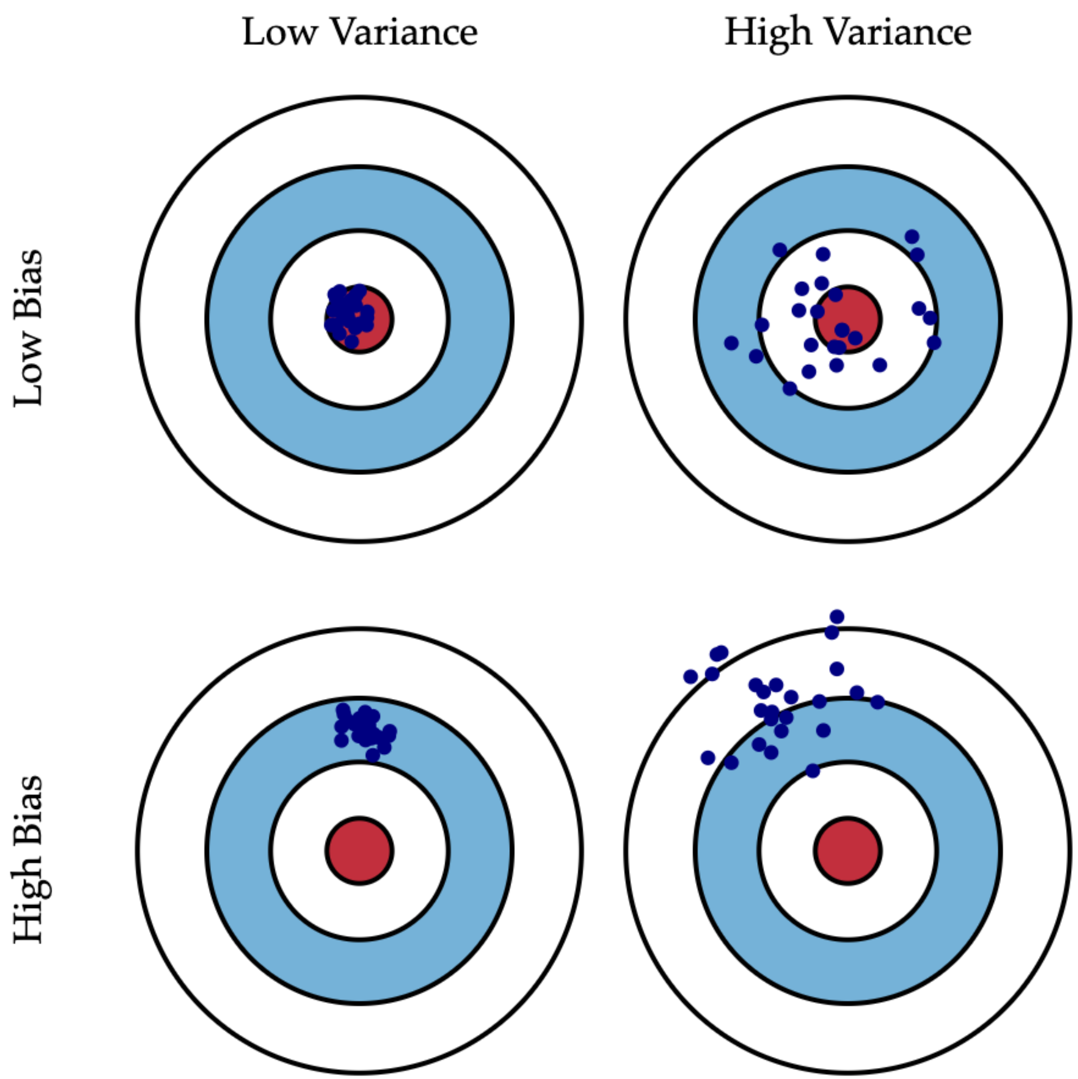
Variance

- Variability of a model prediction for a data point
- Shows how similar bias error is from sample to sample
- Can be seen as error in training data
- A very complex and sensitive to minor variations model usually has high variance
    - Example: high degree polynomial model
- High variance models:
    - Perform very well on training data
    - Performs poorly on unseen data (test data)
    - Focuses on noise (irrelevant relations)
    - Usually means overfitting

Bias vs Variance

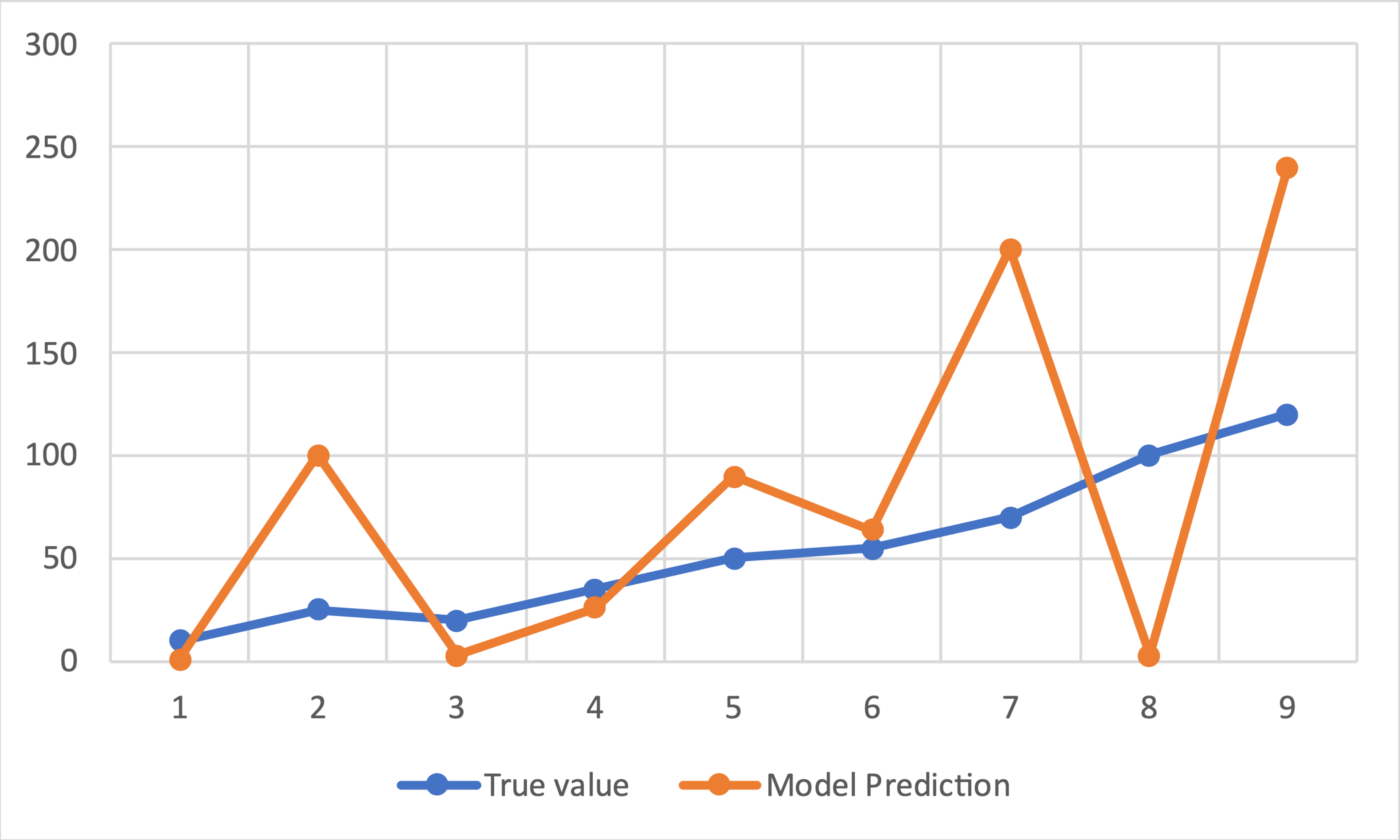
Fortmann-Roe's Analogy

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

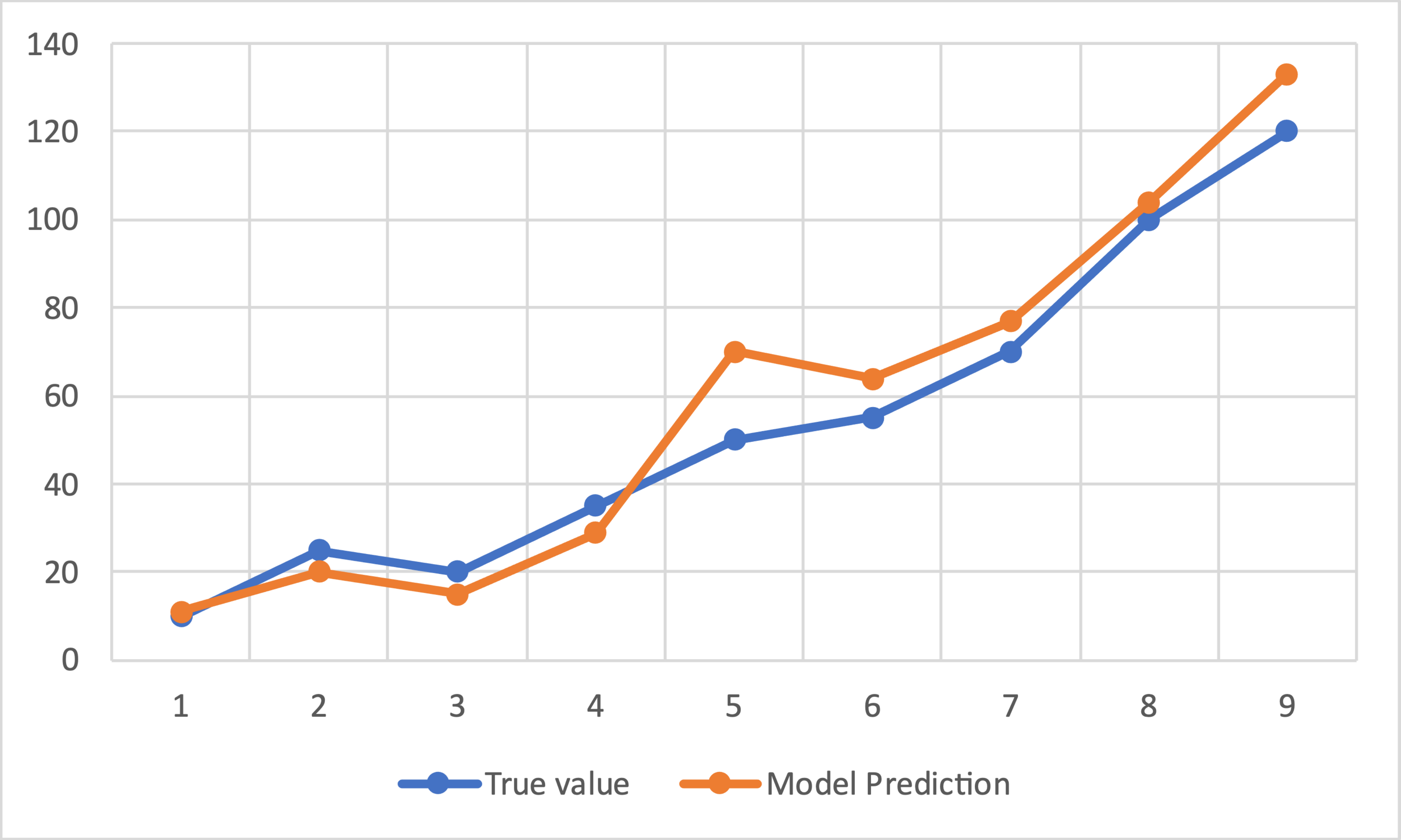


http://scott.fortmann-roe.com/docs/BiasVariance.html

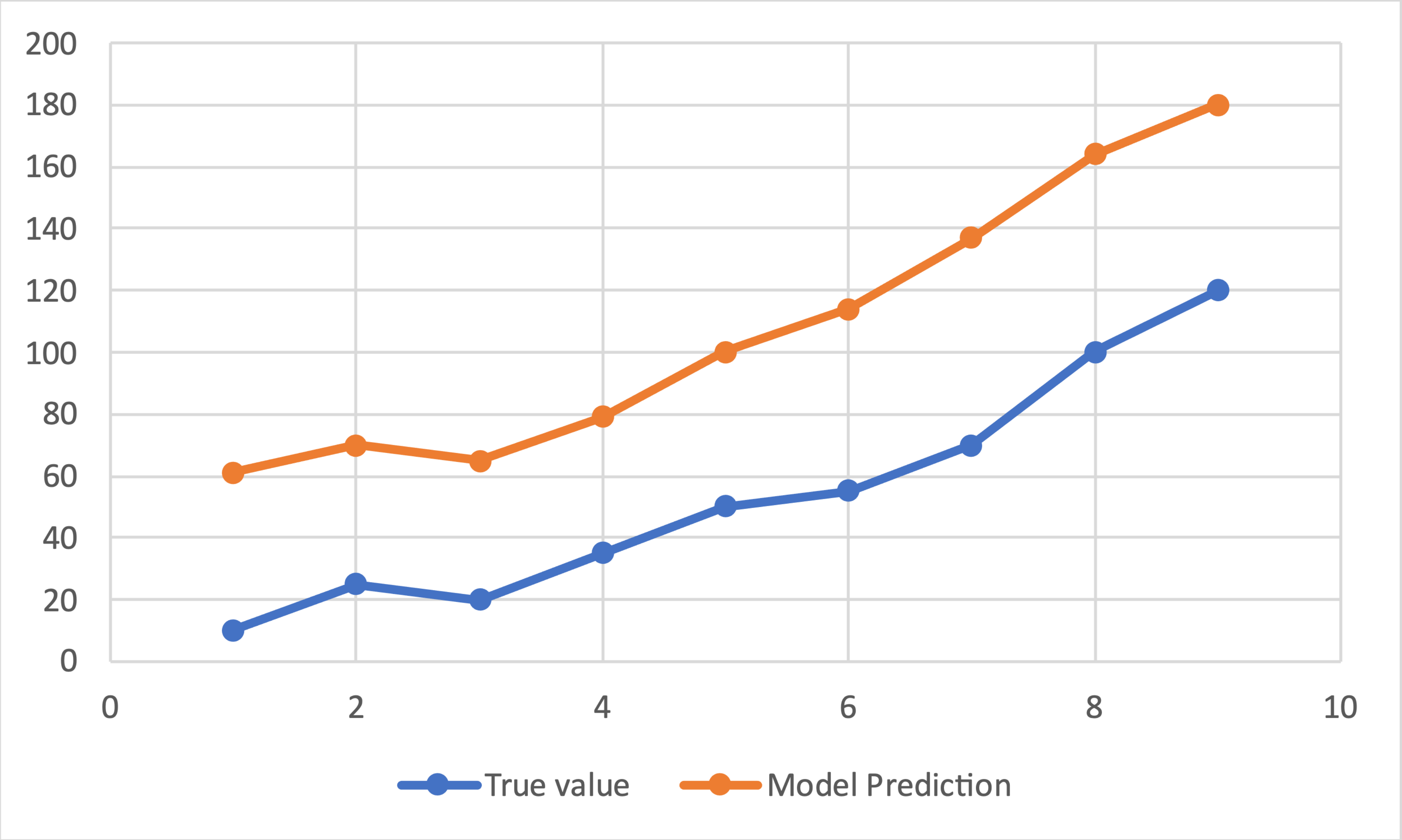
High variance
High bias



Low variance
Low bias



Low variance
High bias



Overfitting and Underfitting

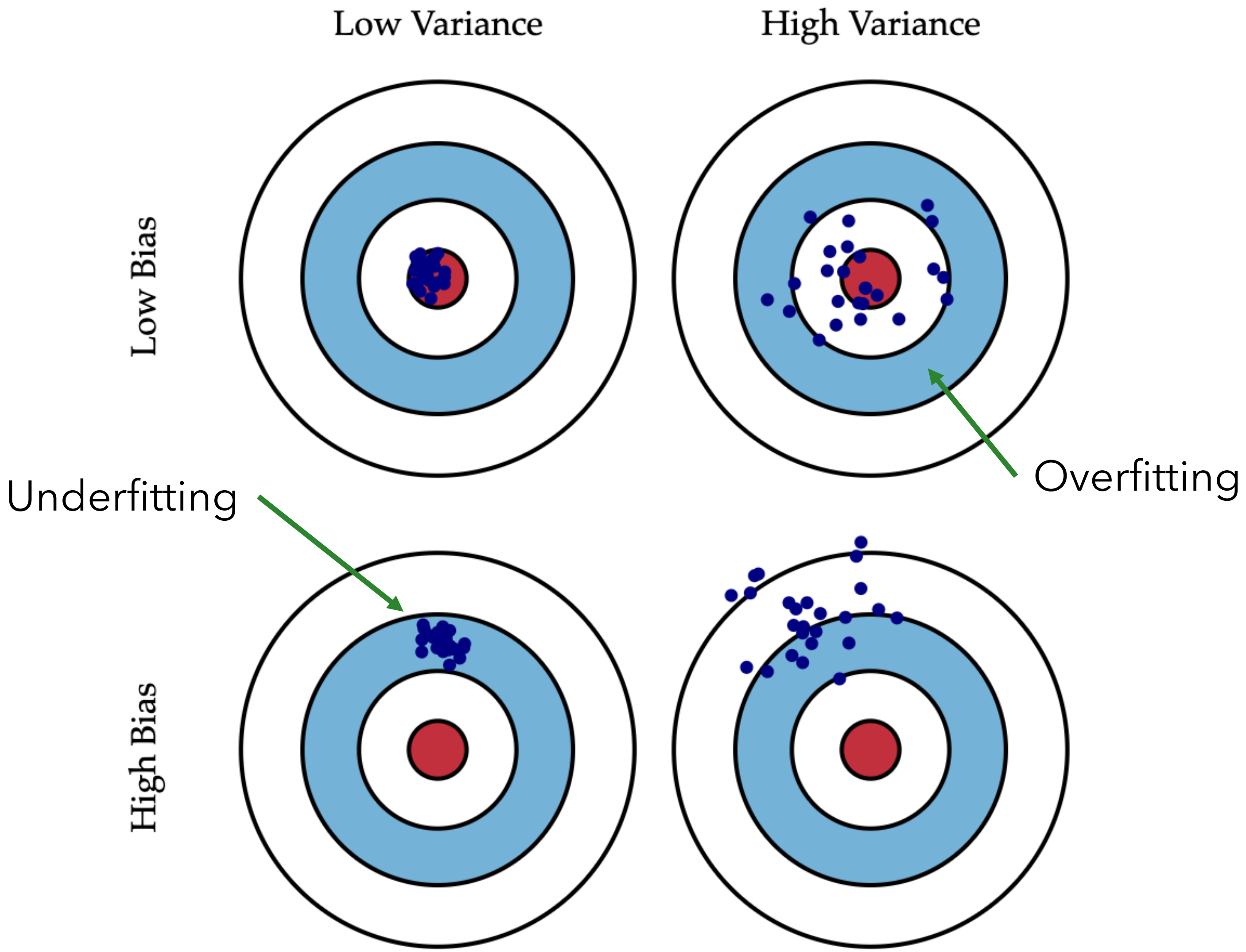
Overfitting

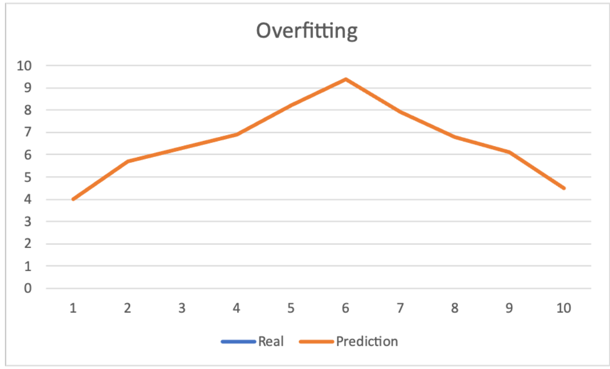
- Model learns every detail of a dataset (rote learning)
- Usually a model has low bias and high variance

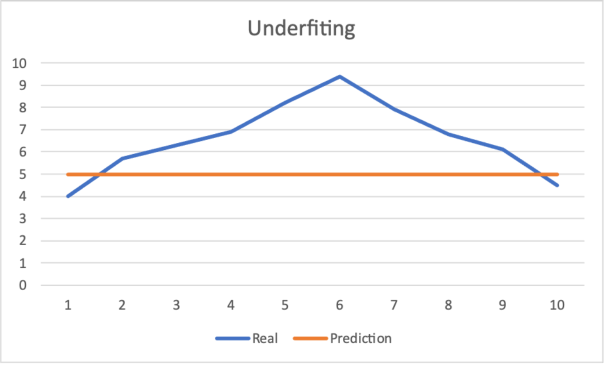
Underfitting

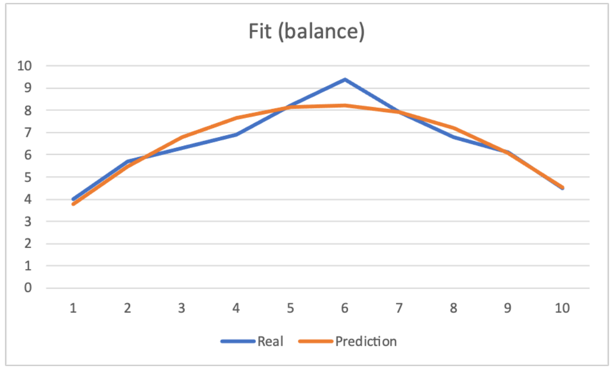
- Model does not generalize
- Model unable to capture underlying pattern in data
- Usually, high bias and low variance

Overfitting and Underfitting

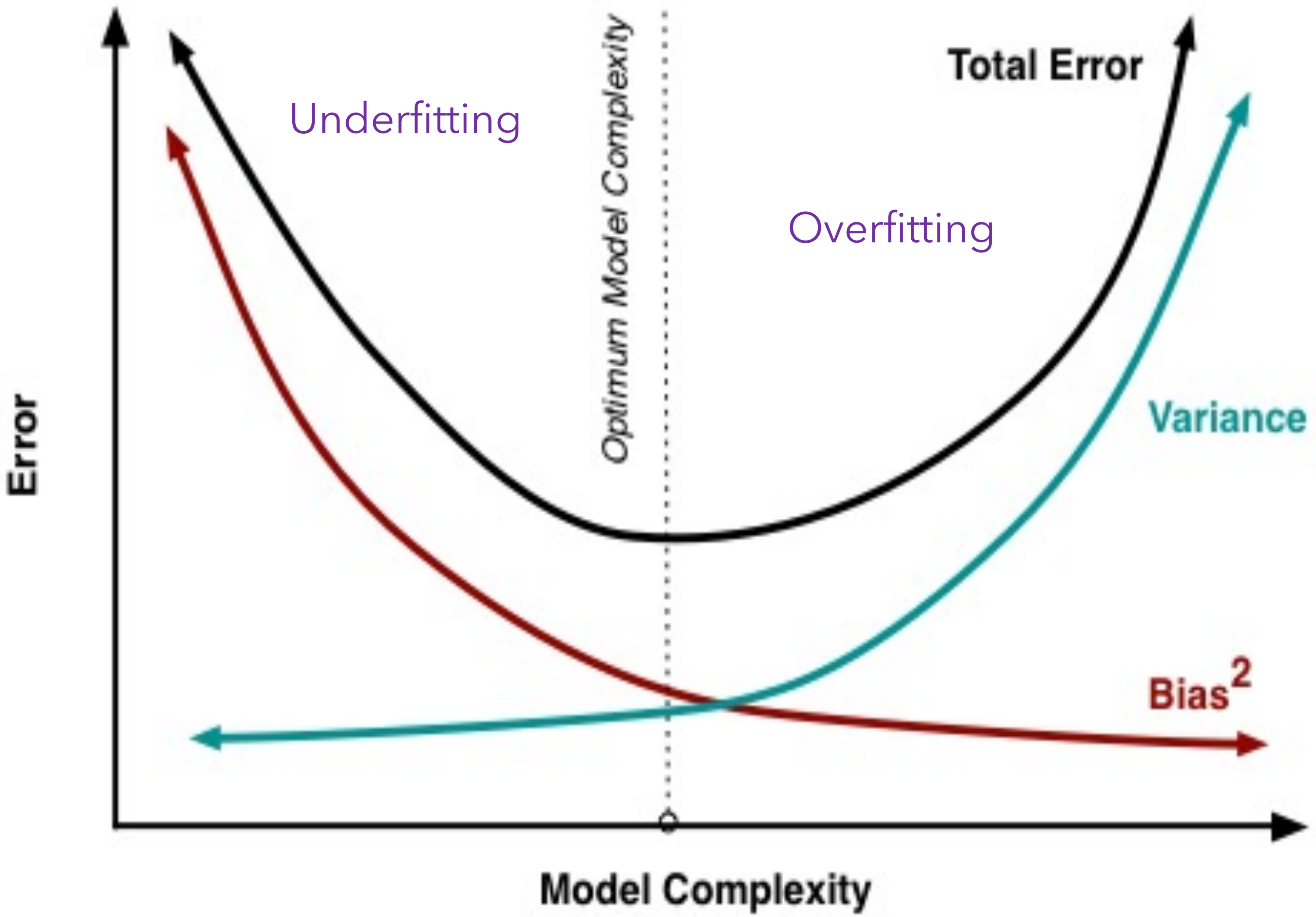


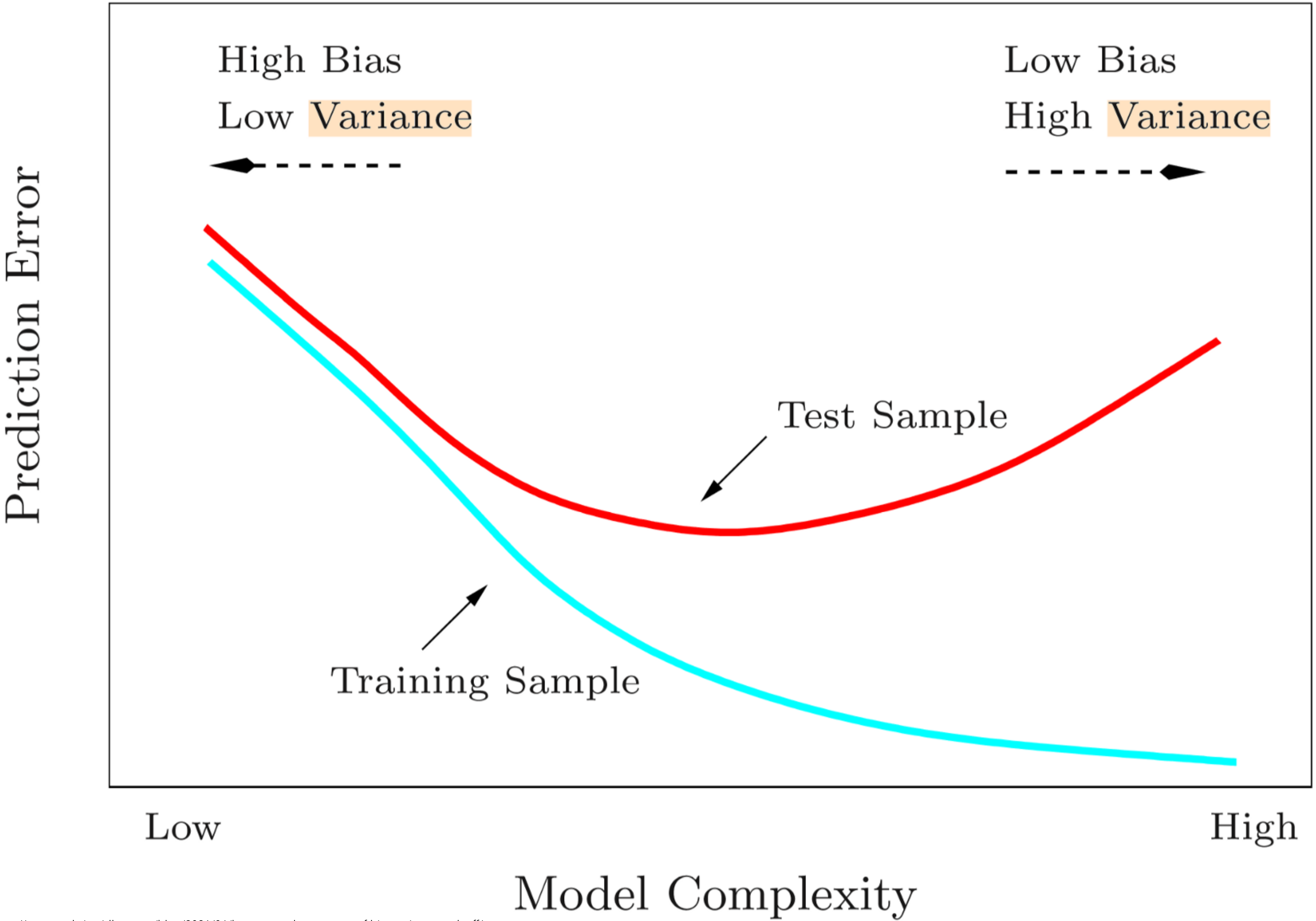

Overfitting
0 1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
Real
Prediction


Underfiting
0 1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
Real
Prediction


Fit (balance)
0 1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
Real
Prediction

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$





https://www.analyticsvidhya.com/blog/2021/06/how-to-get-the-most-out-of-bias-variance-tradeoff/

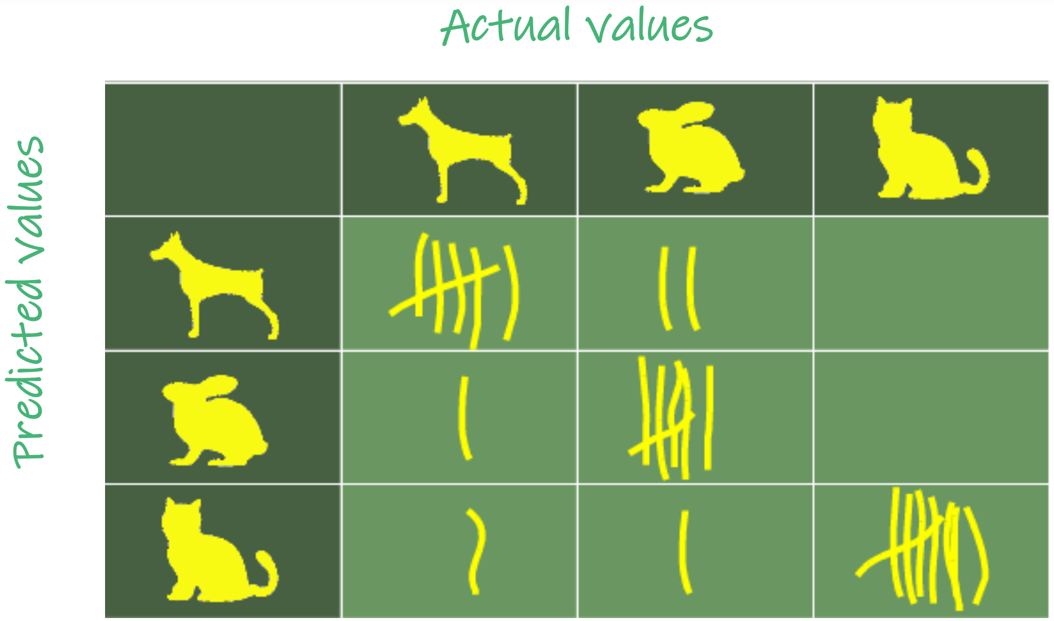
Balancing Variance and Bias

- Select best algorithm for data
- Reduce features (dimensions)
- Reduce error
- Use regularization techniques
- Tune hyperparameters
- Use cross validation

Model Evaluation

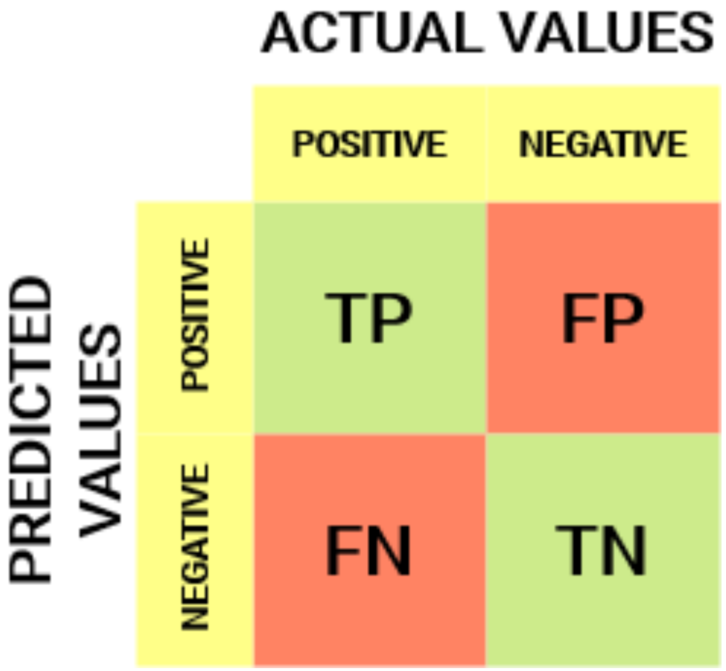
Confusion Matrix

- *N x N* matrix
  - N = number of classes
- Evaluates performance of a classification model
- Compares actual target values with predicted values



https://www.python-course.eu/images/confusion.matrix_image.png

Binary confusion matrix



https://cdn.analyticsvidhya.com/wp-content/uploads/2020/04/Basic-Confusion-matrix.png

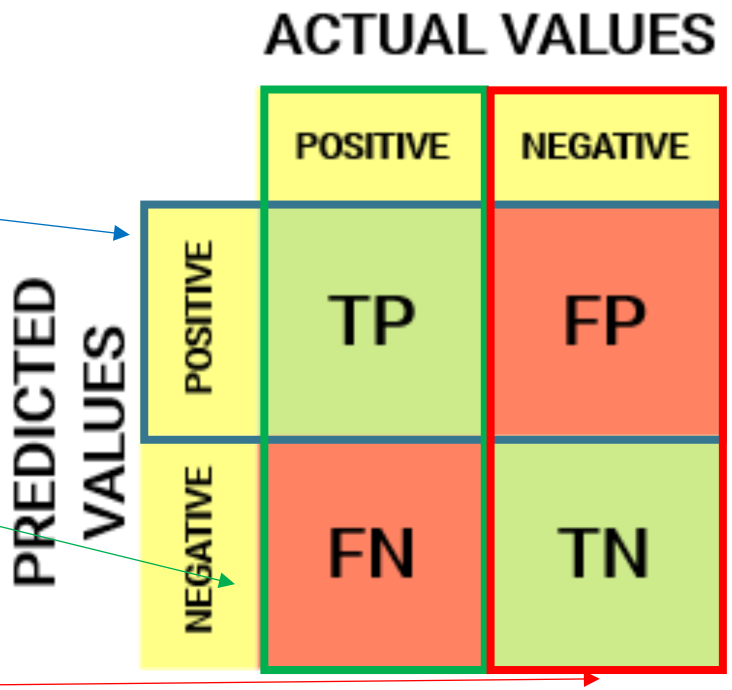
- True Positive (TP)
  - Predicted value matches actual
  - Both were positive
- True Negative (TN)
  - Predicted value matches actual
  - Both are negative
- False Positive (FP)
  - Type I error
  - Predicted value falsely predicted
  - Actual value Negative
- False Negative (FN)
  - Type II error
  - Predicted value falsely predicted
  - Actual value Positive

Confusion matrix metrics

- Accuracy
  - Fraction of predictions model correctly classified
  - $Accuracy = \frac{Correct\ predictions}{Total\ number\ predictions}$
  - For binary classification
    - $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
  - Very simple, but does not take into consideration class imbalances and data unevenly distributed

Confusion matrix metrics…

- Precision
  - Proportion **predicted positives** identified correctly
  - $Precision = \frac{TP}{TP+FP}$
- Recall (Sensitivity)
  - Proportion **actual positives** identified correctly
  - $Recall = \frac{TP}{TP+FN}$
- Specificity
  - Proportion **actual negatives** identified correctly
  - $Specificity = \frac{TN}{TN+FP}$



- Precision used when FP is a higher concern than FN
  - From the positives, how many are really positive?
- Recall used when there is a high cost associated with FN
  - How many positive were correctly classified?
  - A higher recall ensures more actual positive values are being identified

Confusion matrix metrics…

- F1 Score
  - Helps understand balance between Precision and Recall
  - $F1 = \frac{2}{\frac{1}{recall} x \frac{1}{precision}} = 2\ x\ \frac{precision \times recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$
  - Values range from 0 to 1
    - A value close to 1 means it is a better model
  - Used when
    - there is a need to balance this two metrics
    - Not easy to decide if Type I or Type II errors is preferred

ROC & AUC

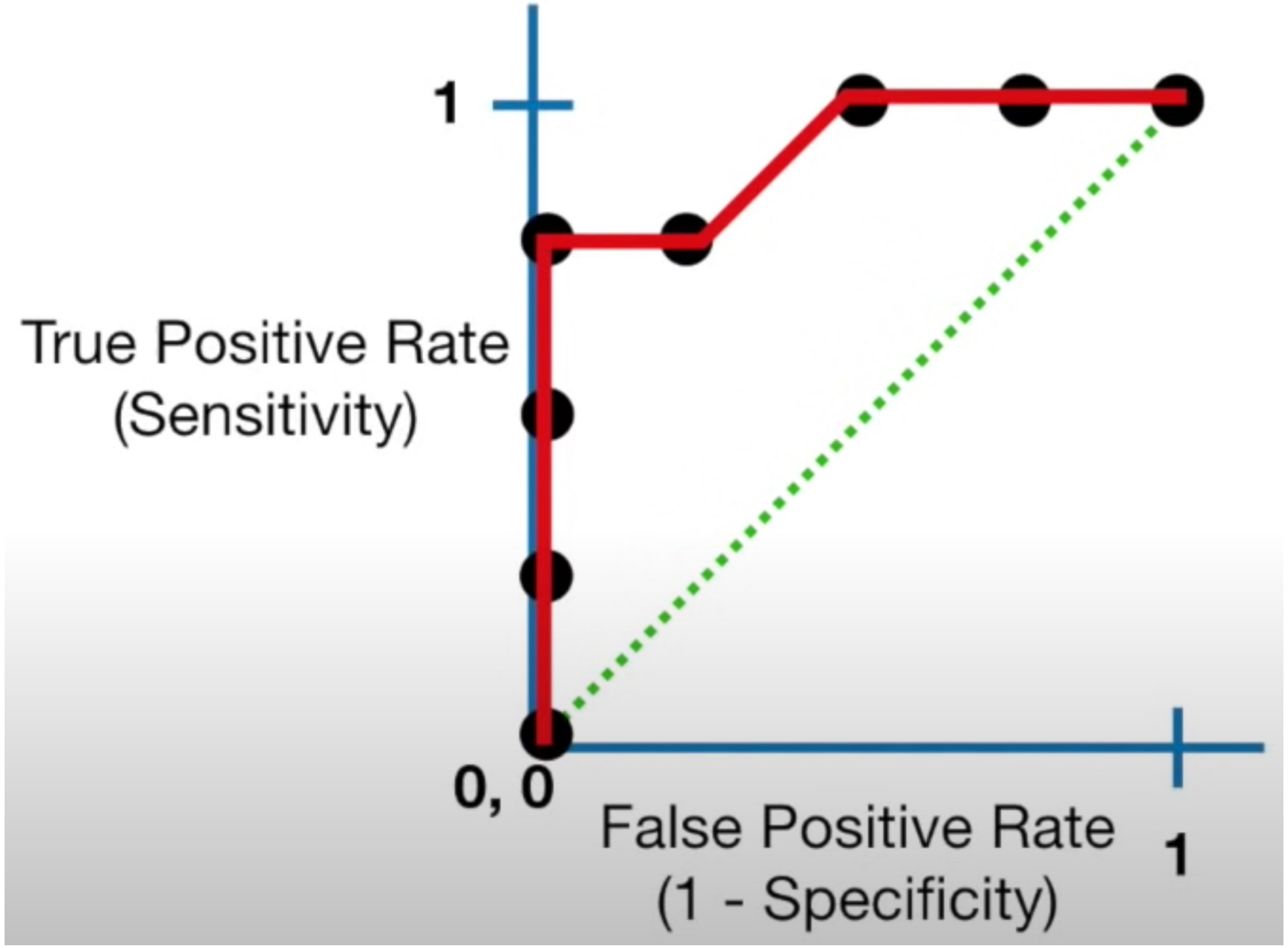
- When a classifier is not reporting the values we desire, we can move the threshold for classification

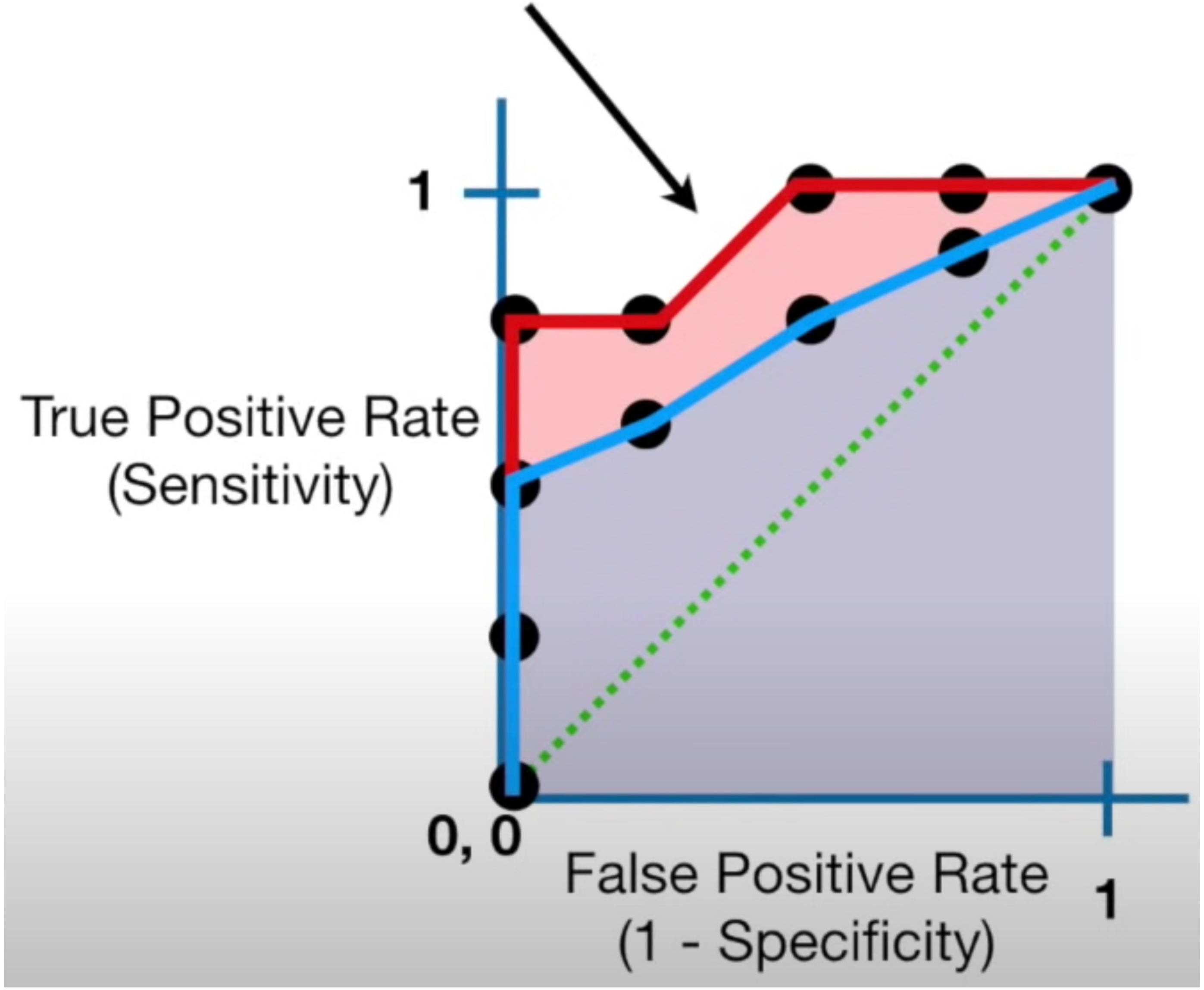
- Moving threshold can increase/decrease recall, precision and specificity values

- ROC and AUC can help us determine the best threshold
  - Receiver Operator Characteristic (ROC)
  - Area Under the Curve (AUC)

ROC

- Summarizes all confusion matrices produced with different thresholds
- Diagonal line is where TP rate is equal to FP rate
- Points above the diagonal represent a good classifier
- The best classifier would be (1,0)





AUC

- Helps to compare different ROC graphs
- The greater the value of the AUC, the better the model is for classifing that data

References

- Alpaydin, Ethem (2004). *Introduction to Machine Learning*. The MIT Press.
- Mitchell, Tom (1997). *Machine Learning*. WCB McGraw-Hill.
- Edwards, Gavin (2018). *Machine Learning, an introduction*. Towards Data Science (https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0)
- Josh Starmer (2019). *ROC and AUC clearly explained!* StatQuest with Josh Starmer, YouTube Channel