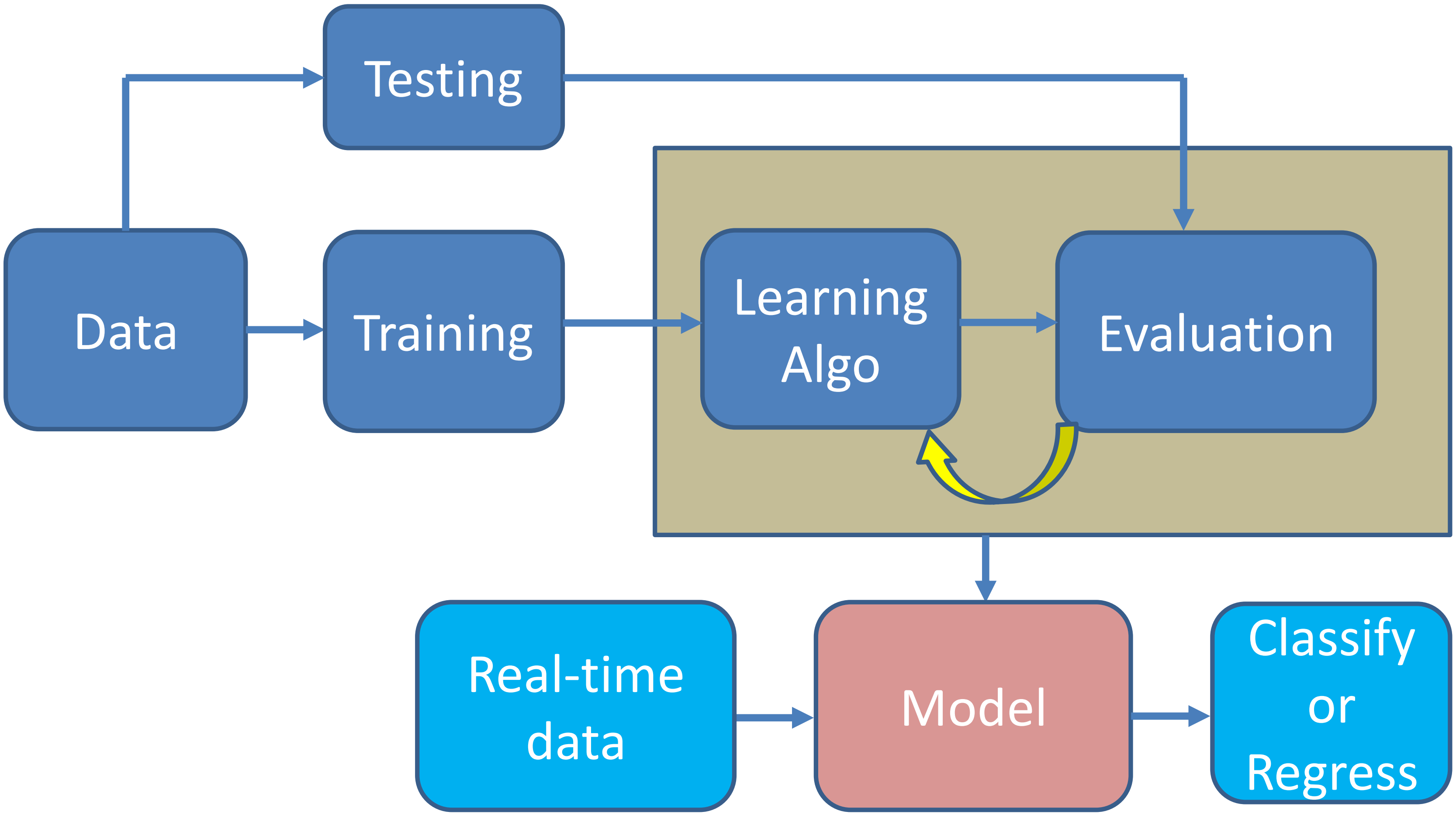
06.09.2024

BITS F464: Machine Learning (1st Sem 2024-25)

MODEL EVALUATION

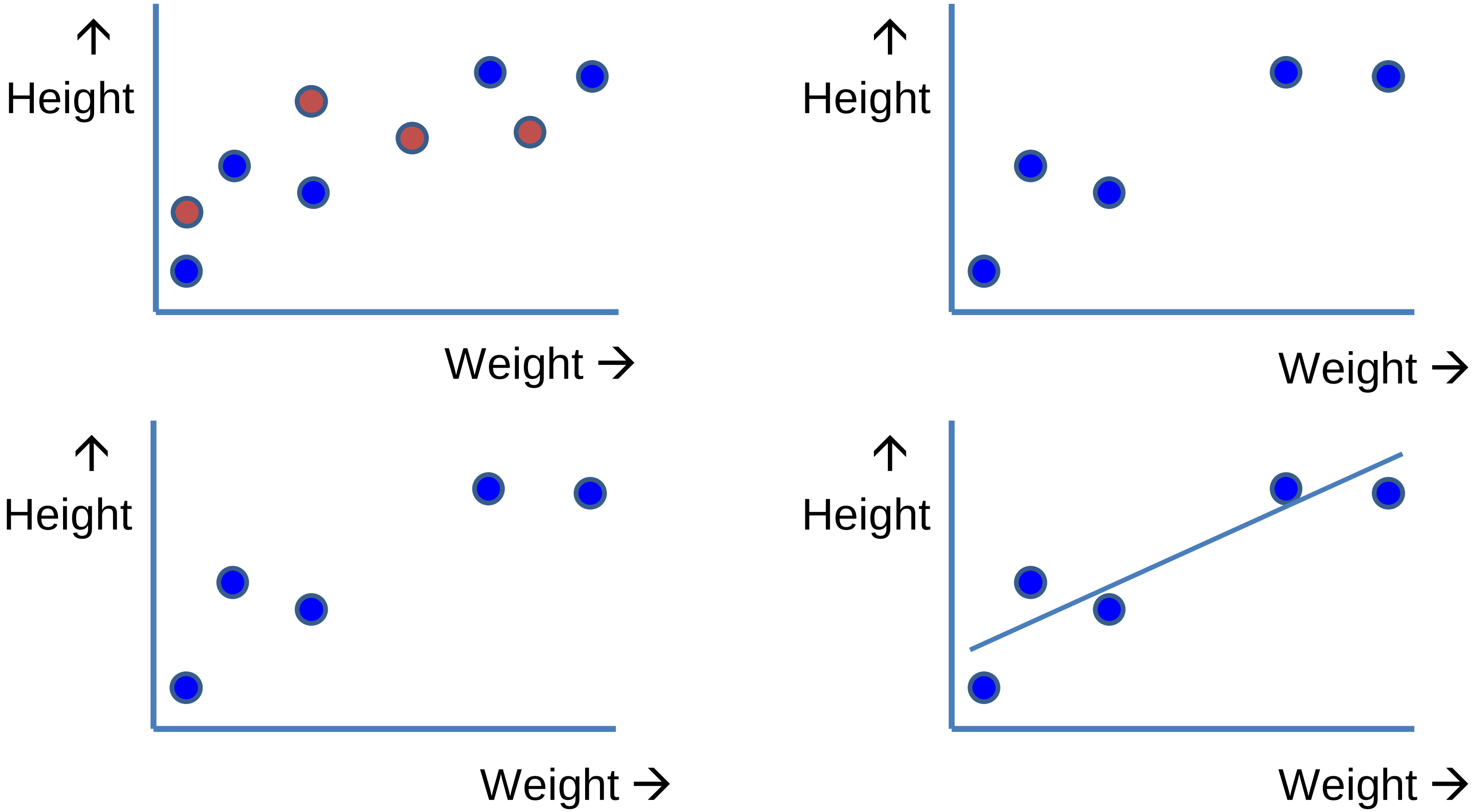
Chittaranjan Hota, Sr. Professor
Dept. of Computer Sc. and Information Systems
hota@hyderabad.bits-pilani.ac.in

How do you Evaluate a ML Model?

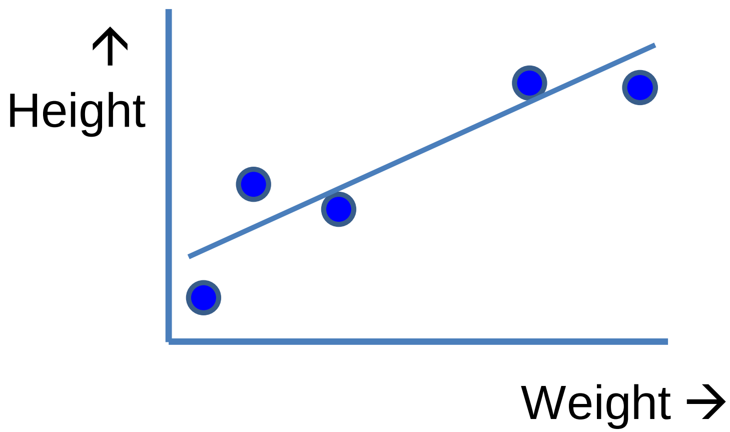


What is Bias in Learning?

- Bias (error) is the amount that a model's prediction differs from the target value. Unable to capture the complexities.

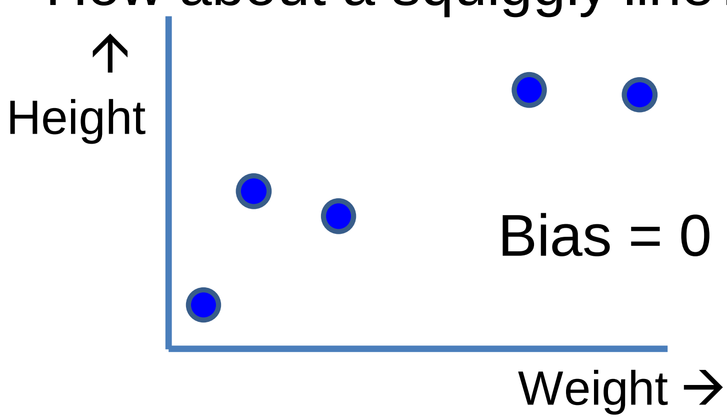


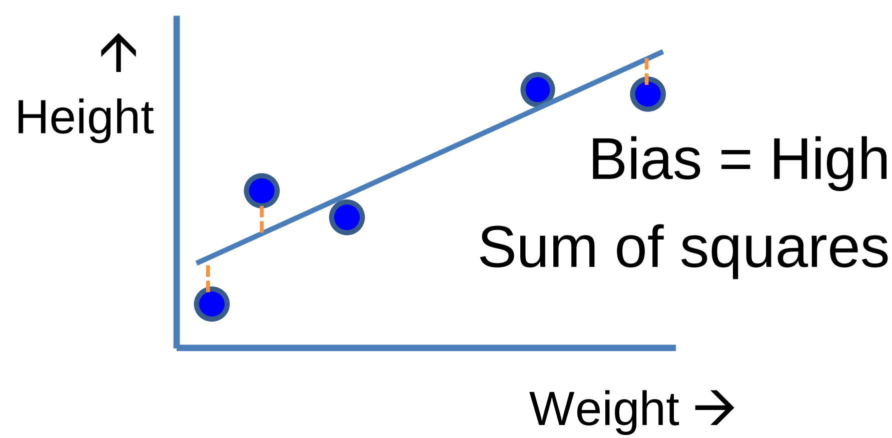
Continued…



- Can you make the line represent the true relationship?
- Inability of the line to take the shape of the curve: **Bias**
- What is the accuracy on the training set?

How about a squiggly line?

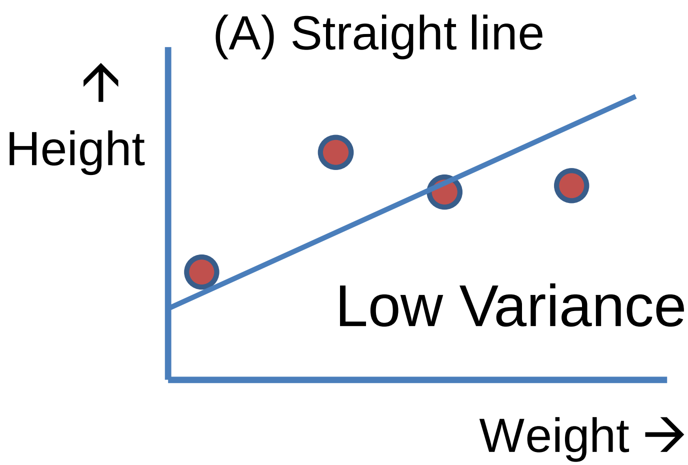


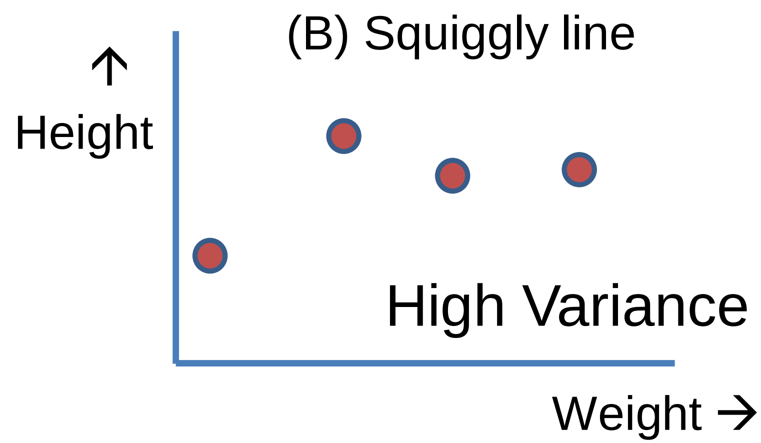


Squiggly line wins in Training set…

Accuracy on the Test set!

- Variance: It's the variability of the model's predictions for different instances of training data. Learns noise from the training data.





Whose sum of squares is better (Low)?

(A)

What is the Variance level of A and B?

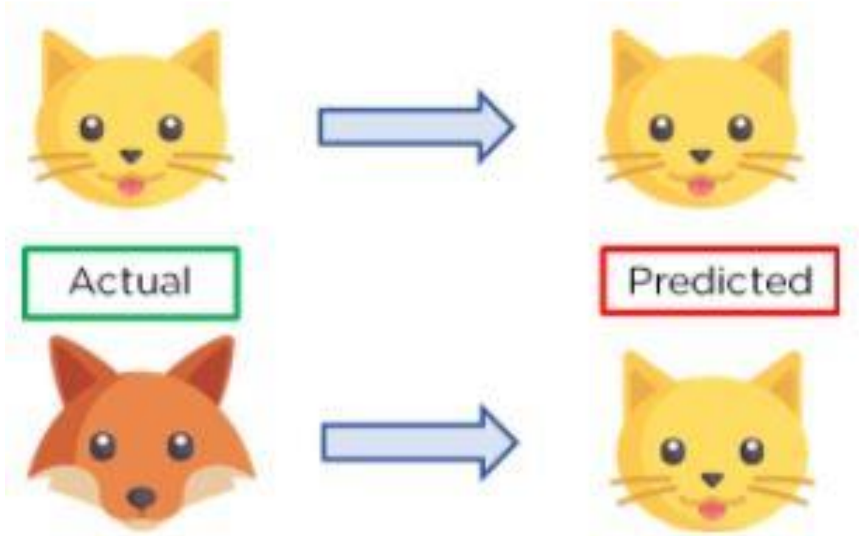
Largely different sums of squares in different data sets. (**Variance**)

Straight line wins in Testing set…

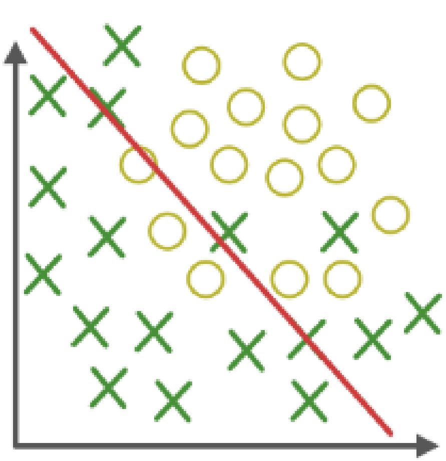
Overfitting due to High Variance

- Low Bias and High Variability: It might do well sometimes, and other times it might perform very poorly. This is Overfitting.

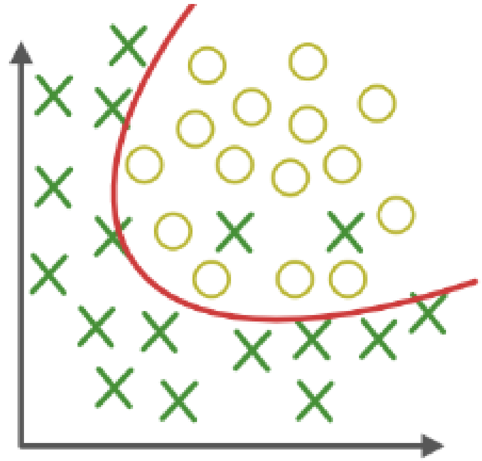
- High Bias and Low Variability: It might do good all the times (consistently) but not great predictions.



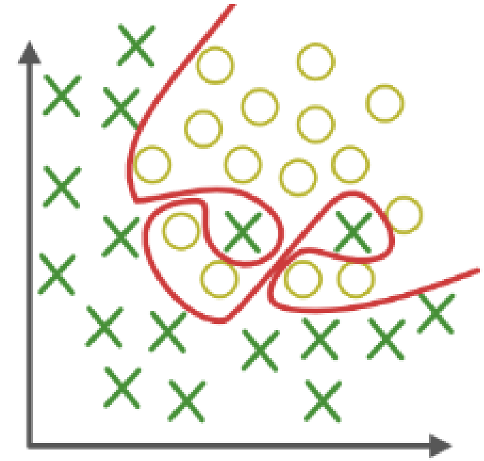
What is desirable in Learning?



(High Bias) (Under-fitting)



(Low bias and variance)



(High variance) (Over-fitting)

1. Model is too simple
2. Inadequate features
3. Size of Training set is not enough
4. Features are not scaled

← **Reasons** →

1. High variance and low bias
2. Model is too complex
3. Size of Training set is small

Increase no. of epochs, model complexity, and features, remove noise etc.

← **Solution** →

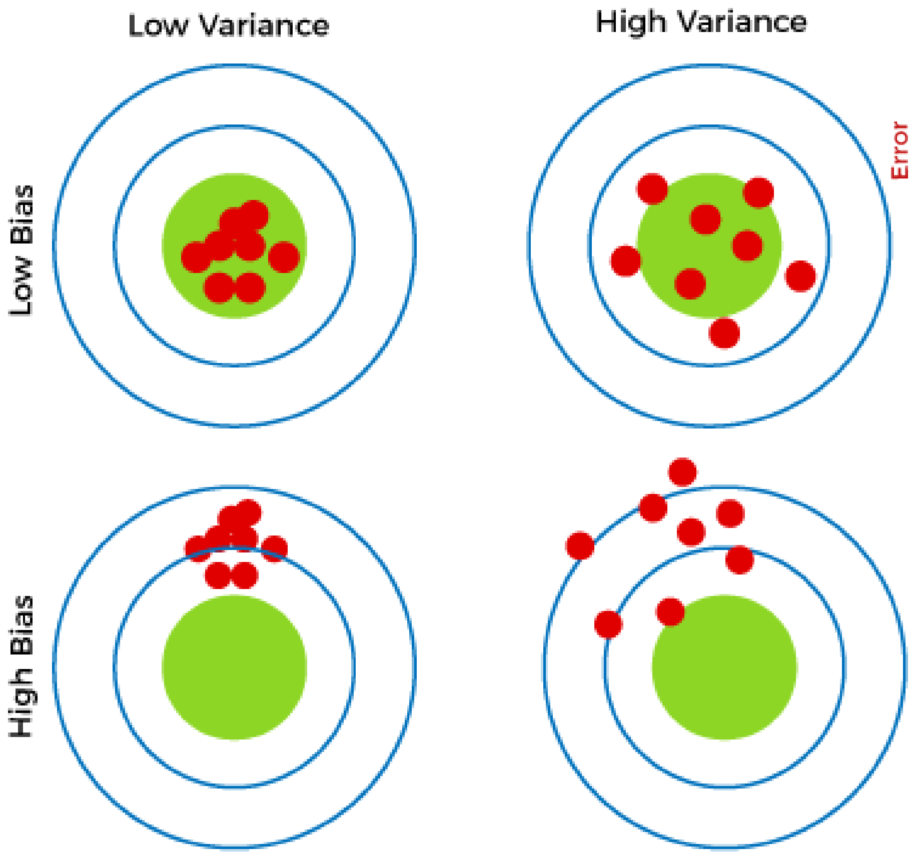
By using Regularization, K-fold Cross validation, Ensemble.

Linear Regression: ↑Bias, Variance↓ Decision Trees: ↓Bias, Variance↑

Bias-Variance Trade-offs

Good

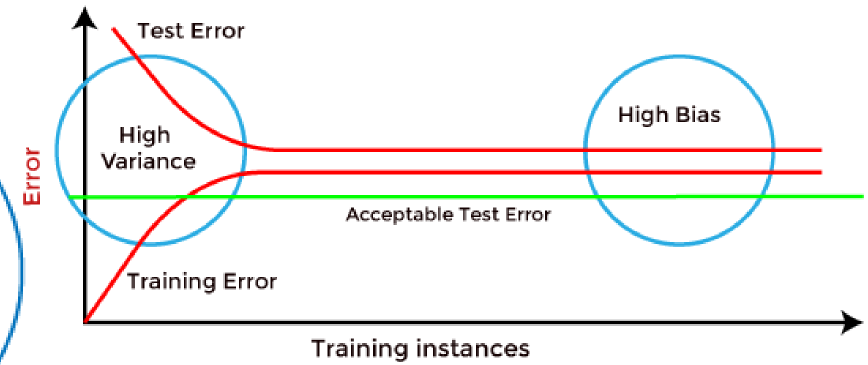
Overfitting



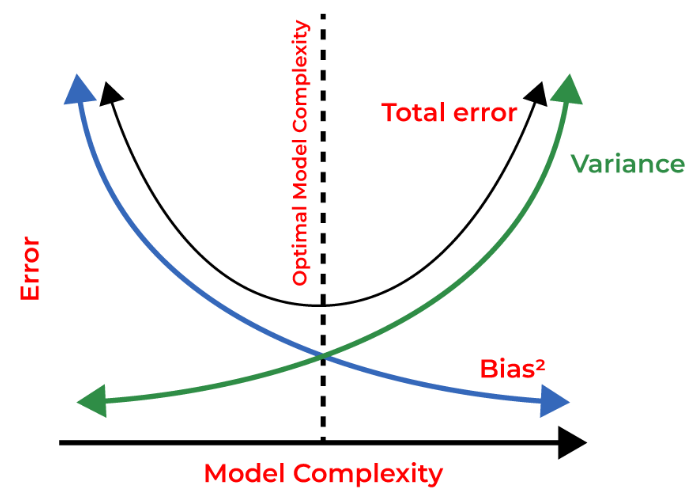
Underfitting

Inconsistent and Inaccurate

Source: https://www.javatpoint.com/

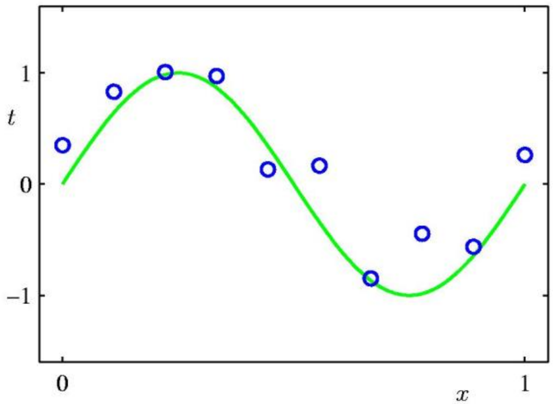


(How to Identify?)

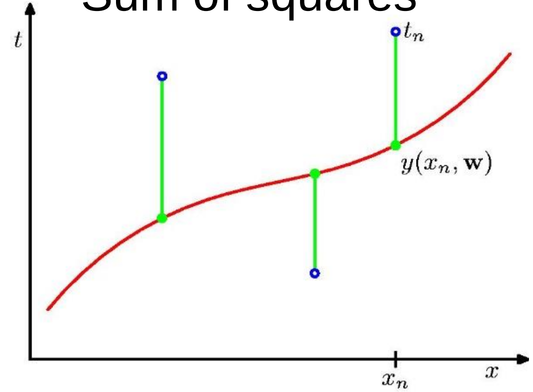


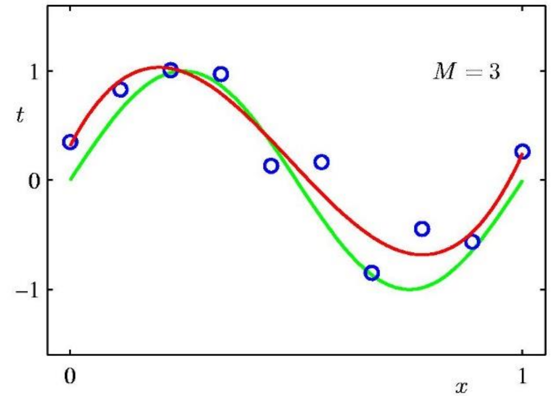
Which one is good and which is bad? Model complexity Vs Error

Avoiding Overfitting: Size of dataset



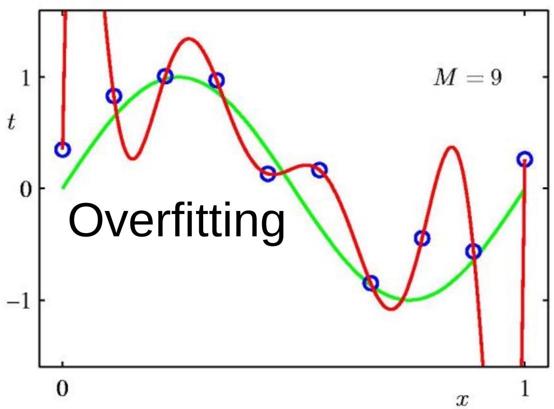
Sum of squares

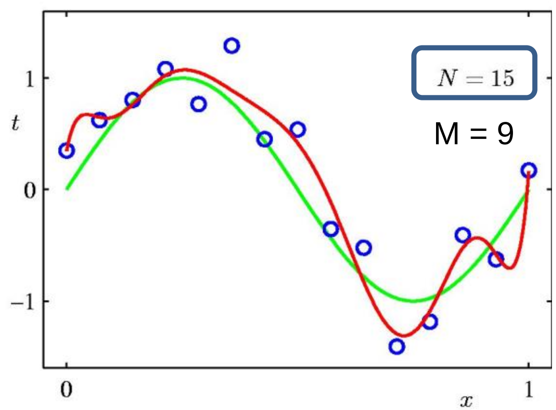


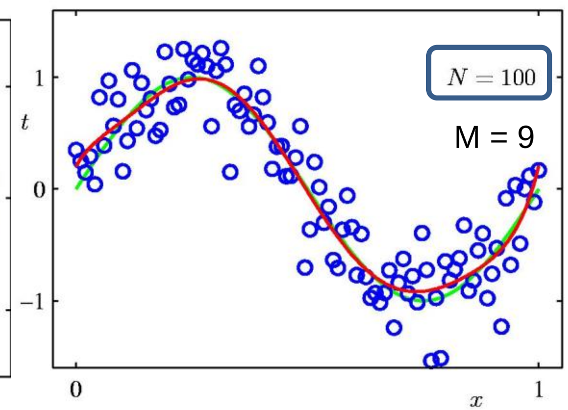


$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$







(How big the size should be? **Heuristics:** The number of data points should be no less than 5 or 10 times the number of adaptive parameters in the model) For, ex: Decision Trees? Max depth, Max features, etc…

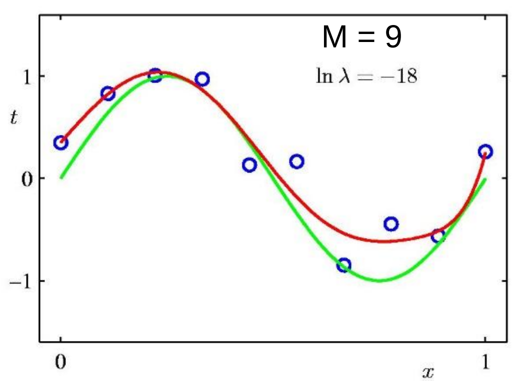
Avoiding Overfitting: Regularization

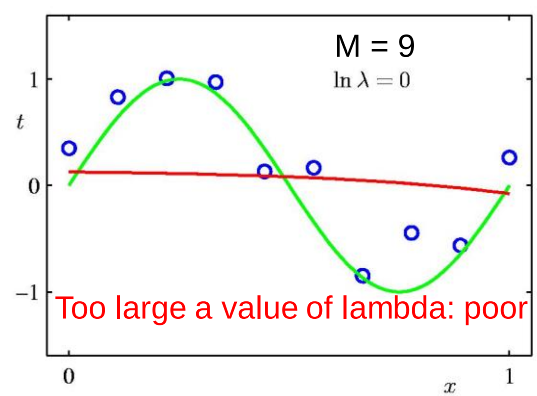
	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

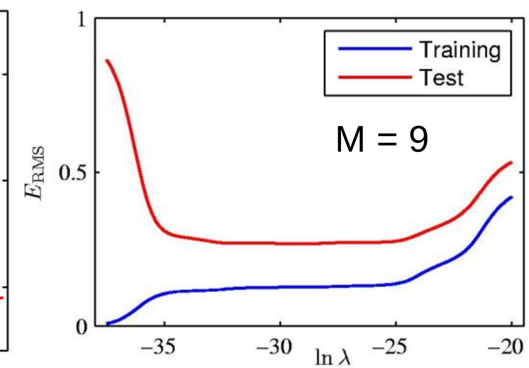
$$(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Is it better?

λ : Controls the degree of Overfitting



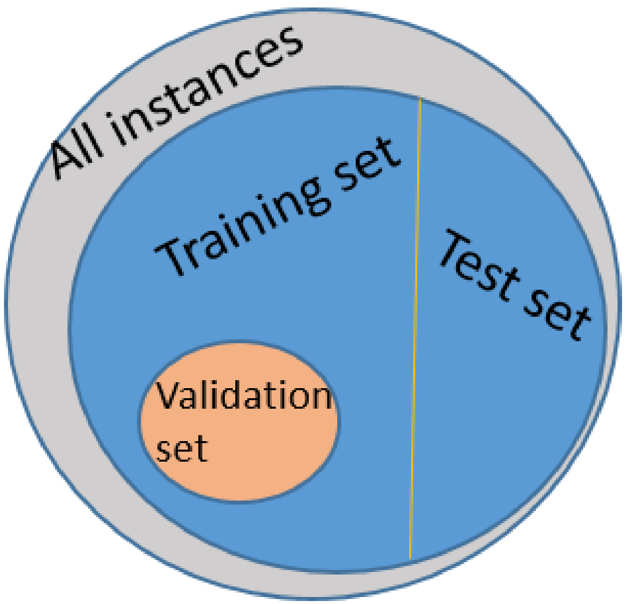


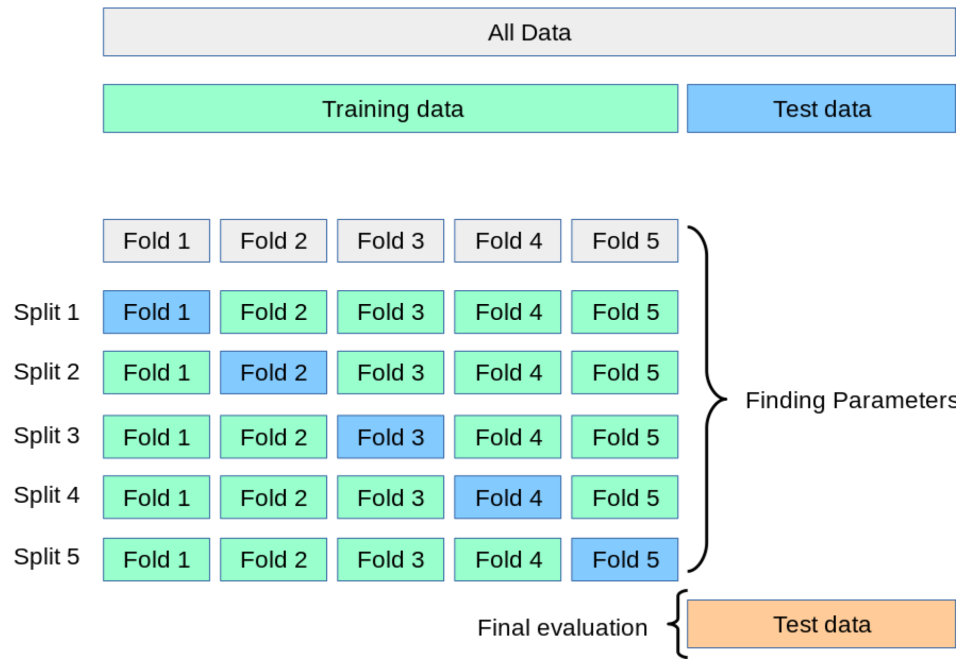


Limitations of single train/test split

- How do you learn a particular algorithm, say decision tree in this course!
  - Model, Training set, Testing set, Validation set, Cross-validation, Accuracy, Type of learning?
- Earlier model of our evaluation (Test1, Test2, Compre, …) Vs the current model. A larger test set tells of what about the performance (learning outcome)? Will some of you not perform consistently? (variance?)
- Larger training datasets may improve accuracy by reducing the complexity of the model, hence lessening the risks of Overfitting.
- A single training set does not tell us how sensitive accuracy is to a particular training sample. The reasons: Noise, Outliers, and Irrelevant information.

Solution to Overfitting: k-fold Cross Validation





Source: https://scikit-learn.org/

- Unfortunately, datasets are never large enough to do this. So we should do our best with small datasets. This is done by repeated use of the same data split differently; this is called *cross-validation*.
- The catch is that this makes the error percentages dependent as these different sets share data.

k-fold Cross Validation: An Example

- Cross-validation helps to reduce variance by providing a more accurate estimate of the model's performance on new data.

```
Total instances: 25
Value of k      : 5
No. Iteration              Training set observations                  Testing set obser
 1      [ 5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]   [0 1 2 3 4]
 2      [ 0  1  2  3  4 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]   [5 6 7 8 9]
 3      [ 0  1  2  3  4  5  6  7  8  9 15 16 17 18 19 20 21 22 23 24]   [10 11 12 13 14]
 4      [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 20 21 22 23 24]   [15 16 17 18 19]
 5      [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]   [20 21 22 23 24]
```

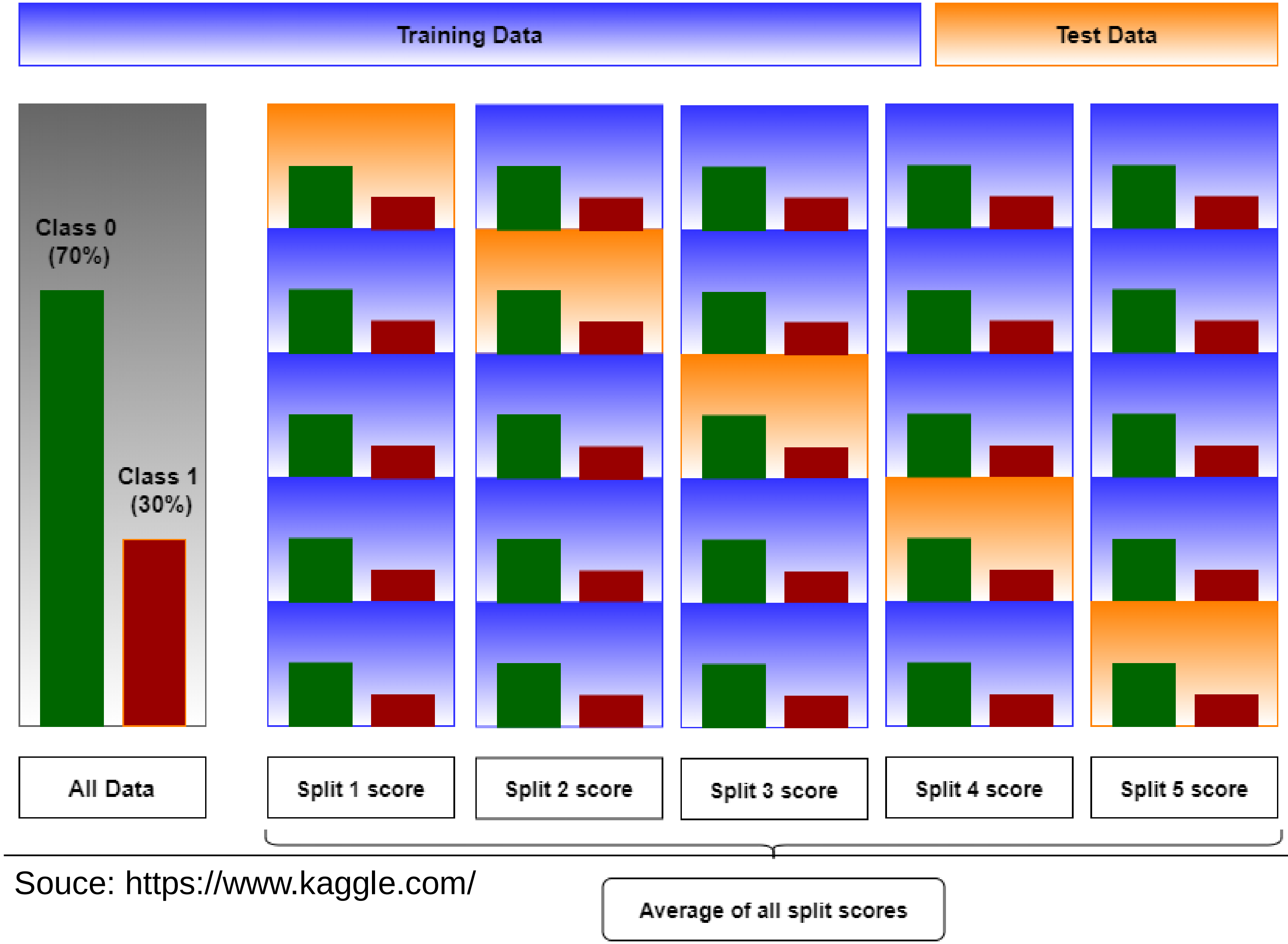
Advantages: Limits Overfitting, Model selection, Hyper-parameter tuning (λ)

Disadvantages: Computationally expensive, not suitable for time-series data as it assumes data points to be independent and identically distributed (IID), Bias-variance trade-off (High value of k: Low Bias & High variance, Lower values of k: High Bias and Low variance).

Source: https://www.geeksforgeeks.org/

Stratified k-fold Cross Validation

- When just random shuffling and splitting is not sufficient.



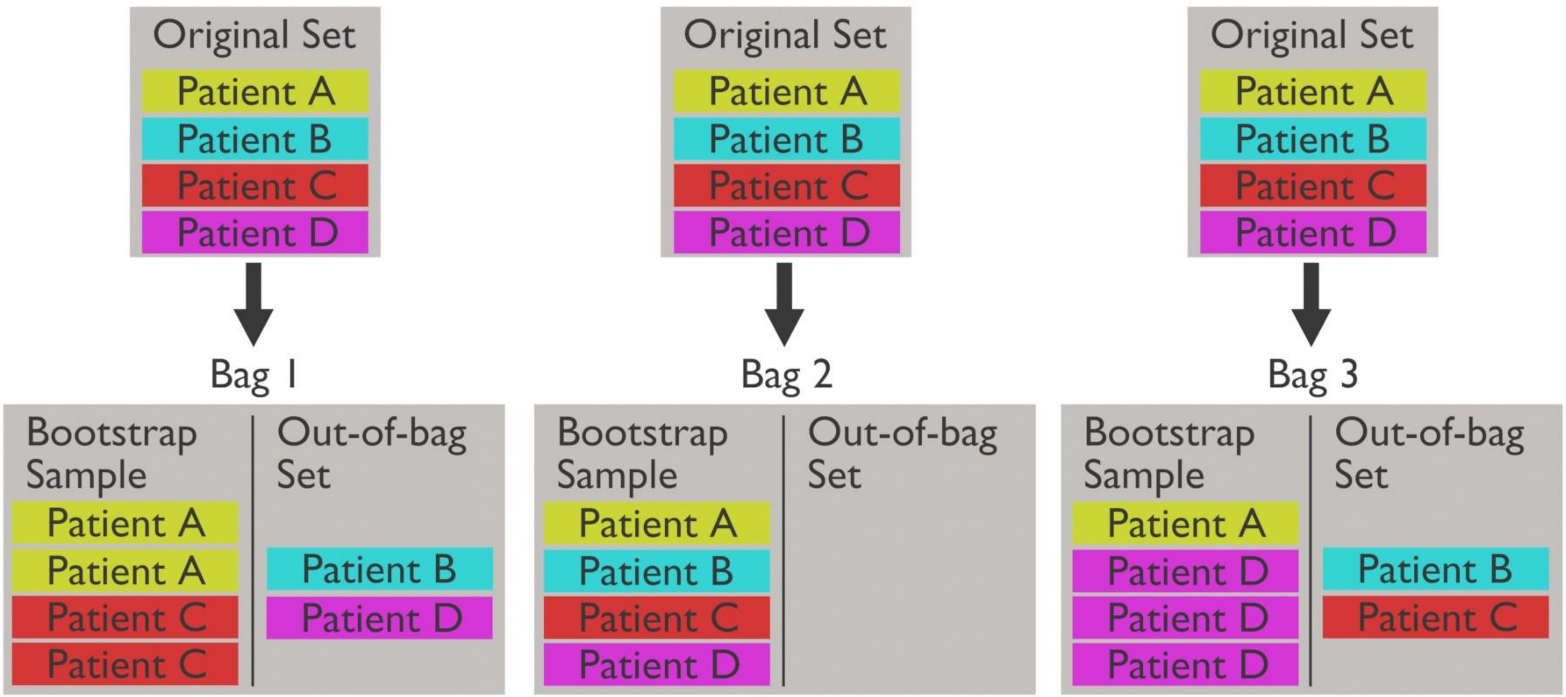
classification ✓

Souce: https://www.kaggle.com/

Out-Of-Bag (OOB) Evaluation Metric

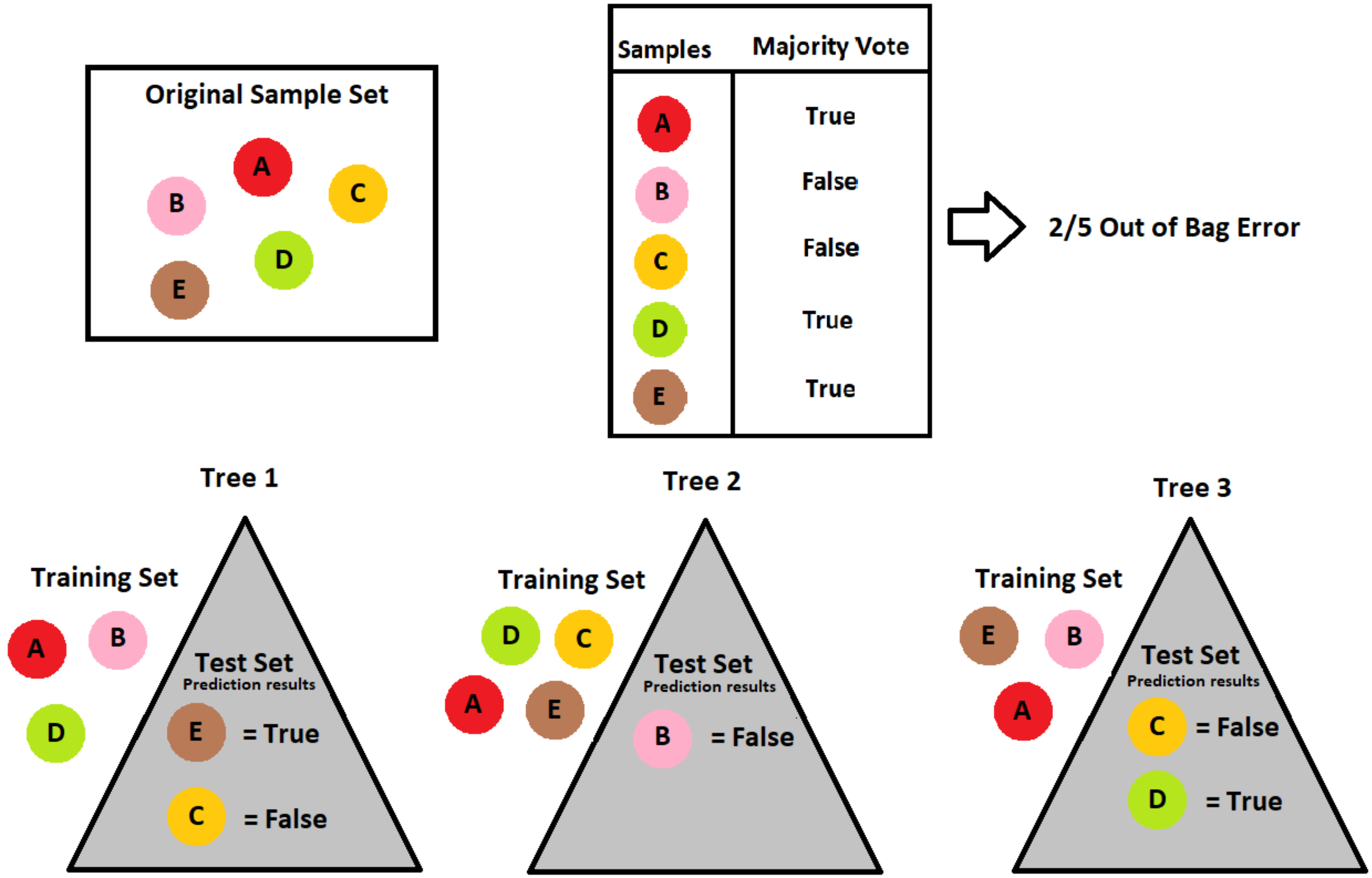
Sampling with Replacement

```
[INFO 24-01-31 12:30:46.6883 UTC kernel.cc:887] Train model
[INFO 24-01-31 12:30:46.6885 UTC random_forest.cc:416] Training random forest on 399 example(s) and 5 feature(s).
[INFO 24-01-31 12:30:46.6904 UTC random_forest.cc:802] Training of tree  1/100 (tree index:0) done accuracy:0.73125 logloss:9.68673
[INFO 24-01-31 12:30:46.7014 UTC random_forest.cc:802] Training of tree  11/100 (tree index:11) done accuracy:0.793451 logloss:2.45525
[INFO 24-01-31 12:30:46.7094 UTC random_forest.cc:802] Training of tree  21/100 (tree index:20) done accuracy:0.817043 logloss:1.0483
```



(Source: Wiki)

Out-Of-Bag Error: An Example



(Source: Wiki)

Over many iterations, the Cross validation & OOB should produce a very similar error estimate.

Classification accuracy for Imbalanced datasets

Img. Source: https://flickr.com/photos/esqui-ando-con-tonho/41295716874/

Model Evaluation Metrics: Confusion Matrix

- A table used in classification problems to assess where errors in the model were made.
- An Example: (12 Individuals diagnosed with/ without diabetes)

Individual Number	1	2	3	4	5	6	7	8	9	10	11	12
Actual Classification	1	1	1	1	1	1	1	1	0	0	0	0

Individual Number	1	2	3	4	5	6	7	8	9	10	11	12
Actual Classification	1	1	1	1	1	1	1	1	0	0	0	0
Predicted Classification	0	0	1	1	1	1	1	1	1	0	0	0

(Source: Wiki)

Can you find out how many True Positives are there here?

Individual Number	1	2	3	4	5	6	7	8	9	10	11	12
Actual Classification	1	1	1	1	1	1	1	1	0	0	0	0
Predicted Classification	0	0	1	1	1	1	1	1	1	0	0	0
Result	FN	FN	TP	TP	TP	TP	TP	TP	FP	TN	TN	TN

Continued...

Can you fill these values?

Actual values

Predicted values

	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Individual Number	1	2	3	4	5	6	7	8	9	10	11	12
Actual Classification	1	1	1	1	1	1	1	1	0	0	0	0
Predicted Classification	0	0	1	1	1	1	1	1	1	0	0	0
Result	FN	FN	TP	TP	TP	TP	TP	TP	FP	TN	TN	TN

Accuracy = (TP + TN) / (TP + FN + FP + TN)

= (6+3)/12 = 0.75

$$\text{Precision} = \frac{TP}{TP + FP}$$

= 6 /(6+1) = 0.85

$$\text{Recall} = \frac{TP}{TP + FN}$$

= 6 /(6+2) = 0.75

Error rate = (FP + FN) / (TP + FN + FP + TN)

= (1+2) /12 = 0.25

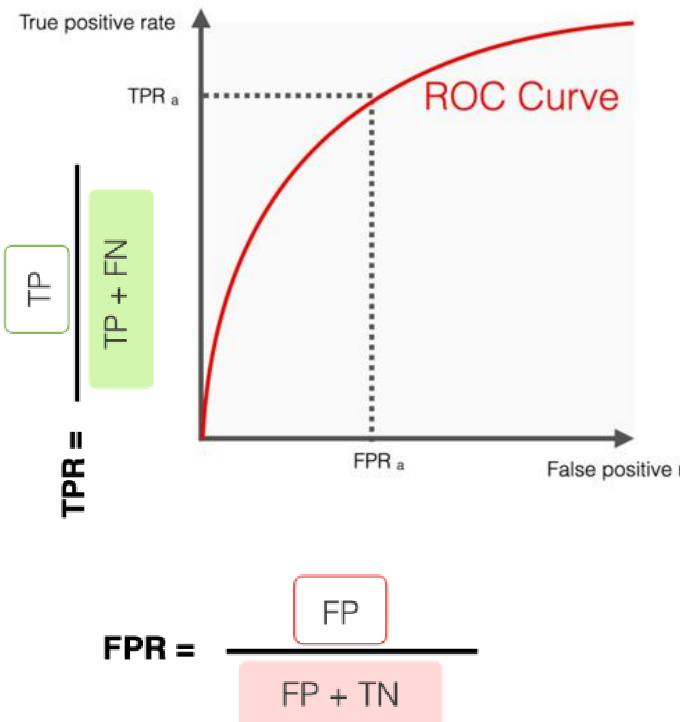
$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

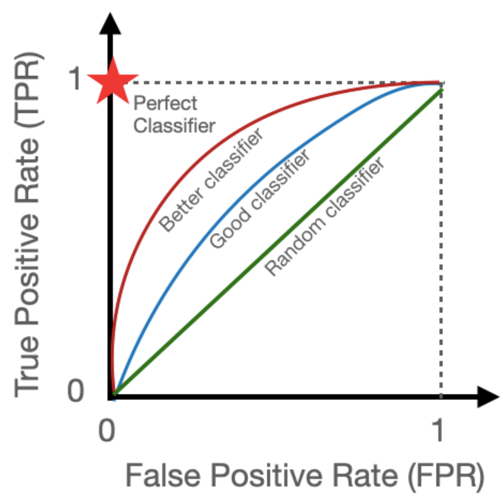
(Class wise performance)

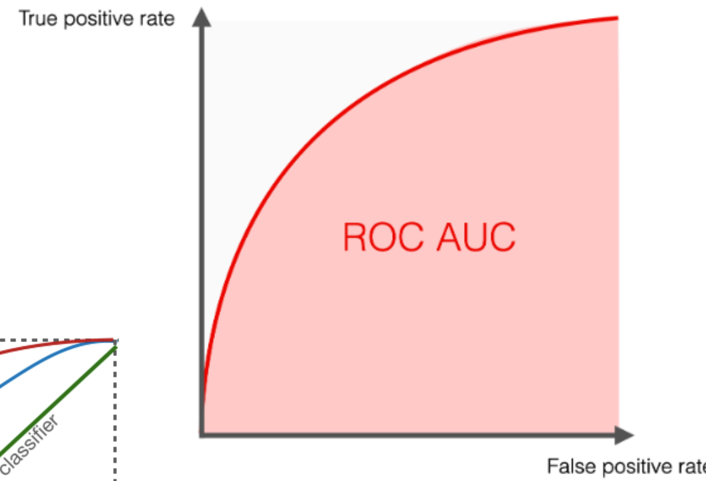
= (2X.85X.75) /(.85+.75)
= 0.25

Receiver Operating Characteristic Curve

- Graphically represent the performance of a binary classifier.







Thank You!