why not change the world?®

Evaluating Regression & Classification Models

Ahmed Eleish

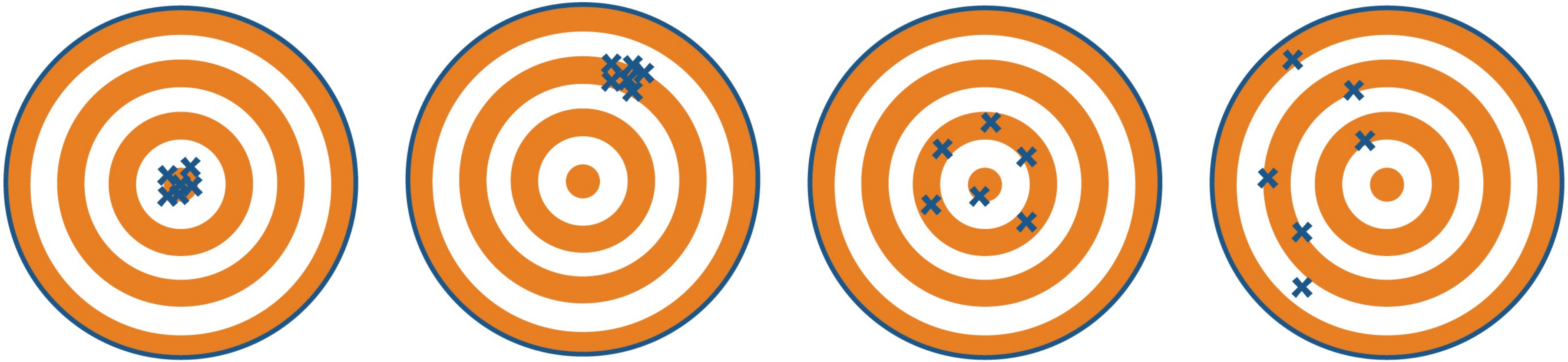
Data Analytics ITWS/MGMT-4600/6600, CSCI-4600, BCBP-4600

February 20th 2026

Tetherless World Constellation
Rensselaer Polytechnic Institute

Contents

Accurate vs. Precise



High Accuracy
High Precision

Low Accuracy
High Precision

High Accuracy
Low Precision

Low Accuracy
Low Precision

http://climatica.org.uk/climate-science-information/uncertainty

Evaluation of Model Training

- ”Training” refers to the process through which a model “learns” patterns from the dataset.
- To robustly evaluate predictive models the training process is repeated multiple times according to commonly used sampling strategies.
- The goal is for model training to be exposed to as much of the variation in structure in the dataset as is reasonably possible… *remember sample vs. population*
- Each training iteration is evaluated separately, with the average performance of the model over the number of training iterations considered an indicator of training success.

Training, Validation and Test sets

- **Training:** subset of dataset used as input to the model's training algorithm
- **Validation:** subset used to evaluate each round of training
- **Test:** subset used to test the final model

e.g.

- The Iris dataset is initially split into a *training* + *validation* set (90% - 135 obs) and a *test* set (10% - 15 obs) ~ this depends on the size of the dataset.
- Over 10 iterations, the *training* + *validation* set is split into *training* (100 obs) and *validation* (35 obs). After training is complete, the average **training error** is calculated.
- The final model is tested on the *test* set (15 obs) and the **test error** is calculated.

Errors

- The error from validation data is called as the **"training error"**
- The error from test data is referred to as the **"test error"**
- The error on the test data is a good indication of how well the classifier will perform on new data **(not used during training)** and this is known as the *generalization*.
- If the model generalizes well, then it will perform well on new data that have *similar structure* to the training data… *sample vs. population*
- The test error is also called the generalization error.

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition

Terminology Confusion!

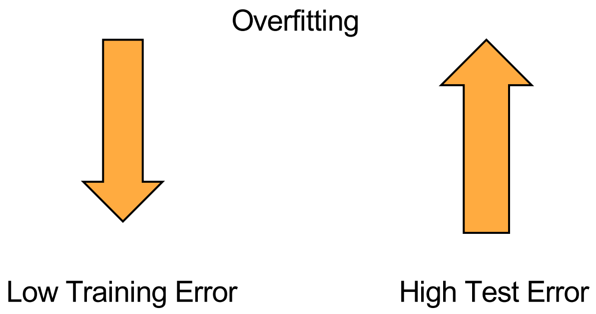
- ‘Test’ and ‘validation’ are used interchangeably in academia and industry!!
- That’s fine… just be clear when documenting your analysis strategy.
- It is also common to split the dataset into only **two** sets, training and validation/testing.
- The decisions made related to model development and evaluation strategies depend on the problem/dataset.

https://en.wikipedia.org/wiki/Training,_validation,_and_test_data_sets

Overfitting

- Another related concept to Generalization is “overfitting”.

- If the model has very **low training error** but it has **high test error**, then it is over fitting.

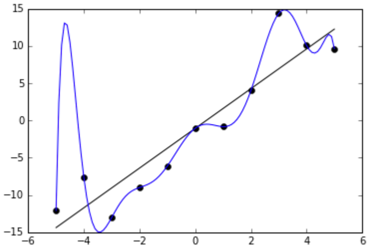


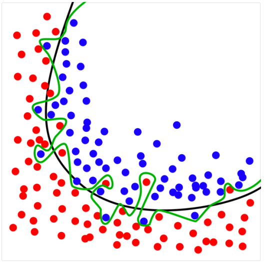
Resource/Reference: Introduction to Statistical Learning with R, 7th Edition

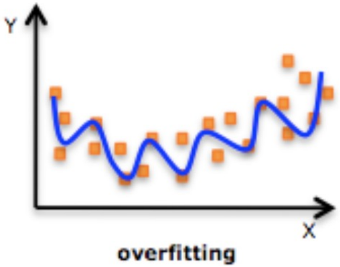
Overfitting

- This is a good indication that the model may have learned to ***model the noise*** in the training data, instead of the learning from the underlying structure of the data.

- Overfitting is an indication of poor generalization.

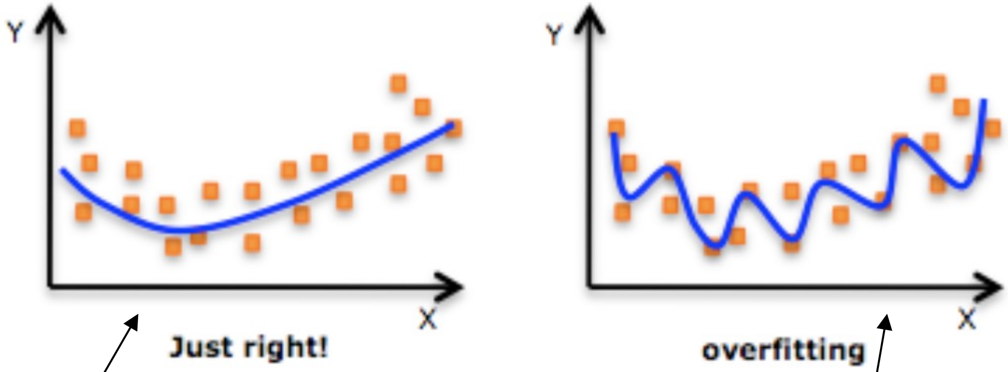






Image/Photo Credit:
https://en.wikipedia.org/wiki/Overfitting#/media/File:Overfitted_Data.png

Image/Photo Credit:
http://pingax.com/regularization-implementation-r/



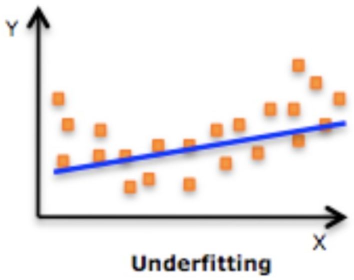
Model is fitting to
the structure of the data

Model is fitting to
the noise of the data

Image/Photo Credit: http://pingax.com/regularization-implementation-r/

Underfitting

- **Underfitting** occurs when a statistical model cannot adequately capture the underlying structure of the data.
- In other words, **underfitting takes place when the model has not properly learned the structure of the data**.



Image/Photo Credit: http://pingax.com/regularization-implementation-r/

Cross-Validation

Robustly Validating Models

- Cross-validation is a method to robustly evaluate/validate models by iteratively splitting the dataset into subsets for training and testing, training the model, and computing and evaluation metric

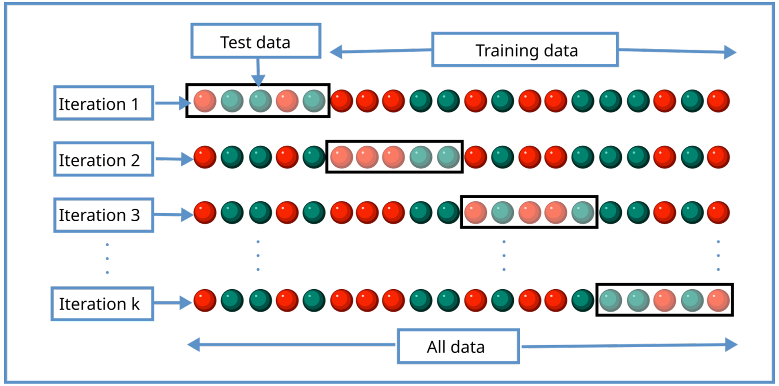
- There are several cross-validation strategies
  - K-fold cross-validation
  - Monte Carlo cross-validation
  - Leave-One-Out cross-validation

https://en.wikipedia.org/wiki/Cross-validation_(statistics)

K-fold Cross Validation

- In k-fold cross validation, the data are segmented into *k* **disjoint partitions**.

- During each iteration, one partition is used as the test set and the remaining k-1 (combined) are used for training; The process is repeated *k* times, so that each partition is used exactly one time for the validation.



By Gufosowa - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=82298768

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition - Chapter 5

Monte Carlo Cross Validation (Repeated random sub-sampling)

- In Monte Carlo cross validation, the dataset is split into training/test sets over *n* iterations with the samples in each set selected at random.
- The ratio between partition sizes may be constant or vary over the iterations.
- Commonly used in research, considered robust because of the averaging effect over multiple iterations.
- Downside: since selection is random, some observations may not end up in test sets and some may be oversampled

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition - Chapter 5

Leave One Out Cross Validation (LOOCV)

- Given a dataset with *n* observations, for *n* iterations drop *one* observation and use all the others for training; test using the 1 observation left out

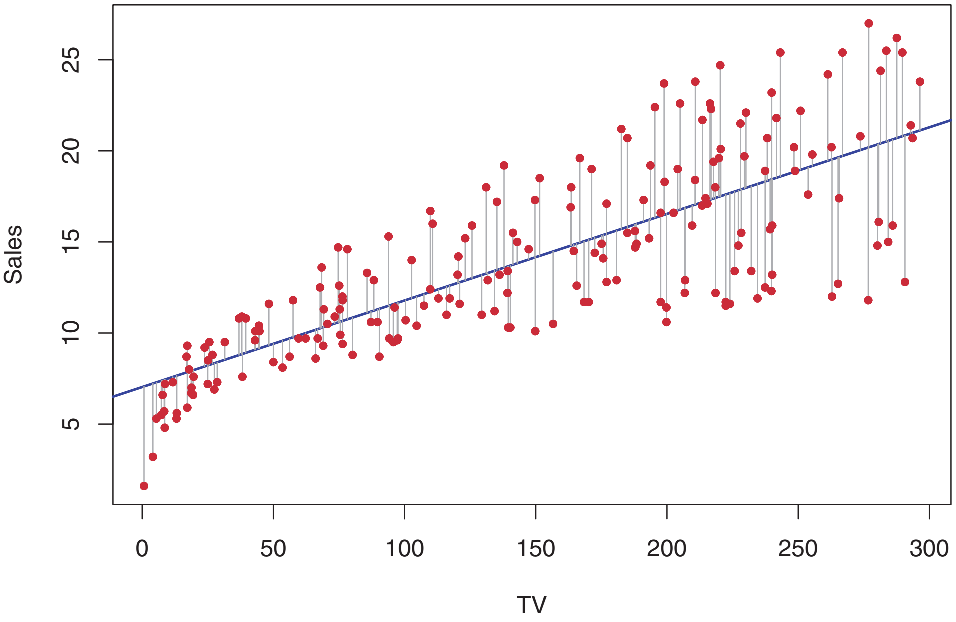
- Depending on the size of the dataset, may be computationally expensive.

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition - Chapter 5

Evaluating Regression Models

Evaluating Linear Models

- Sales vs. TV ad spending
- Sales in 1000s of units
- TV ad spending in 1000s of $



Residual Sum of Squares (RSS)

For given data $(x_1,y_1), \ldots, (x_n,y_n) \in \mathbb{R}\times\mathbb{R}$,

- Residual Sum of Squares (RSS), the *i*th residual $e_i = y_i - \hat{y}_i$

$$\mathrm{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$$

Evaluating Linear Models

1. Assessing the coefficient estimates

	Coefficient	Std. error	t-statistic	p-value
Intercept	7.0325	0.4578	15.36	< 0.0001
TV	0.0475	0.0027	17.67	< 0.0001

1.1 Values of coefficients >> their Std. errors

$$SE(\hat{\beta}_1) = \frac{RSE}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

Quantity	Value
Residual standard error	3.26
R^2	0.612
F-statistic	312.1

1.2. High t-statistic

$$t = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)}$$

$$\mathrm{RSE} = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

1.3. Low p-value

Hypothesis (more TV ads → more sales)

X ~~H0 : There is no relationship between X and Y~~

√ Ha : There is some relationship between X and Y

Evaluating Linear Models

Quantity	Value
Residual standard error	3.26
R^2	0.612
F-statistic	312.1

2. Assessing the model's fit

Residual Standard Error

- Mean sales $\approx$ 14,000 units
- RSE = 3.26 = 3,260 units

$$\mathrm{RSE} = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

R^2

- Measures the how much of the variability in *Y* can be explained using *X* (as a predictor)
- has a value between 0,1

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad TSS = \sum_{i=1}^{n}(y_i - \bar{y}_i)^2$$

Measures of Model Error

Mean Absolute Error

- Mean(||Predicted value - Real value||)

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

Mean Squared Error

- Mean((Predicted value - Real value)^2)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root Mean Squared Error

- SquareRoot(Mean((Predicted value - Real value)^2))

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y_i})^2}{n}}$$

R code:
https://rpi.box.com/s/ct1l2cbz06j4qev5euzqw6l1v46uxyt1

Evaluating Classification Models

Classification Accuracy

- ***Accuracy = (Number of correct predictions) / (Total number of data points)***

$$= \frac{TP+TN}{N}$$

- Simplistic evaluation of model

- Classification error = 1 – ***Accuracy***

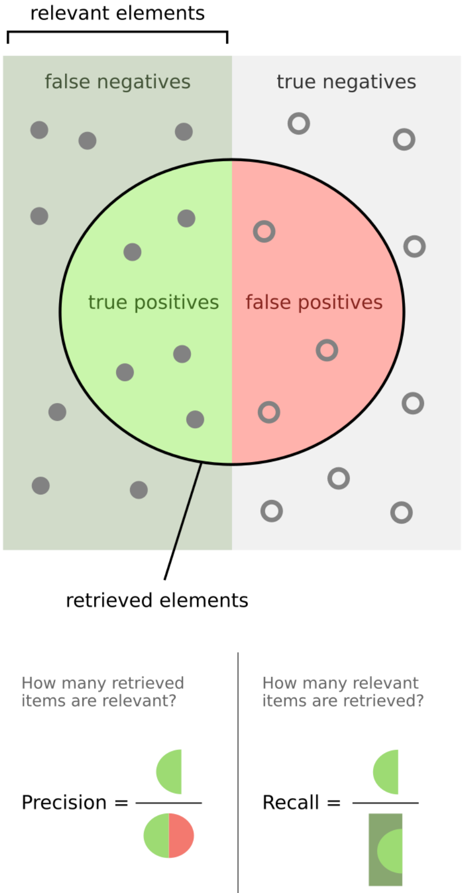
$$= \frac{FP+FN}{N}$$

		Predicted Value	
		Positive	**Negative**
Real Value	**Positive**	TP	FP
	Negative	FN	TN

Per Class Evaluation

$$\text{Precision} = \frac{\text{Relevant retrieved instances}}{\text{All } \textbf{retrieved} \text{ instances}}$$

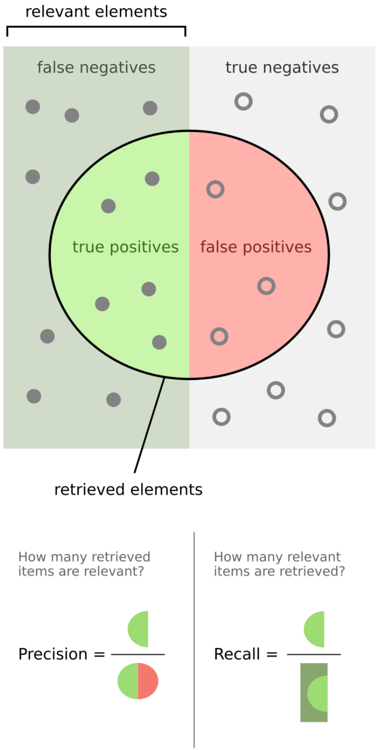
$$\text{Recall} = \frac{\text{Relevant retrieved instances}}{\text{All } \textbf{relevant} \text{ instances}}$$



https://en.wikipedia.org/wiki/Precision_and_recall

Evaluation Metrics – Per Class

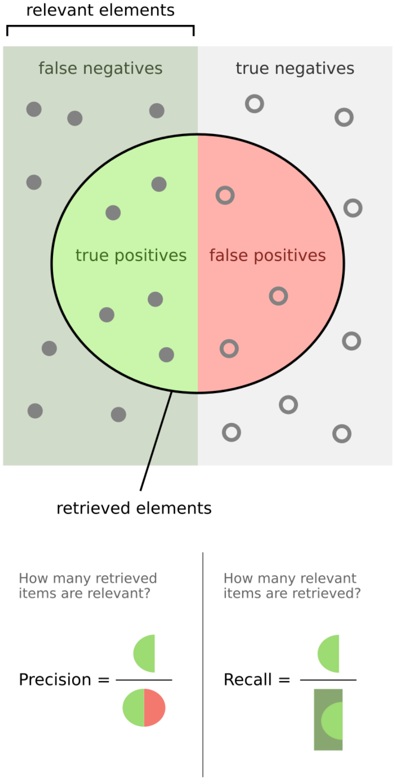
- ***Precision = (True Positive) / (True Positive + False Positive)***
  - ***Fraction of positive predictions that belong to the positive class***
- ***Recall = (True Positive) / (True Positive + False Negative)***
  - ***Fraction of positive class correctly identified***
- ***F1 = 2 [(Recall * Precision) / (Recall + Precision)]***
  - ***F1 = (True Positive) / [True Positive + 1/2*(False Positive + False Negative)]***
  - ***Harmonic mean (weighted average) of precision and recall***



Credit (unmodified): Walber (own work) - CC BY-SA 4.0 - https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg

Additional Evaluation Metrics – Per Class

- ***Specificity = (True Negative) / (True Negative + False Positive)***
  - *Fraction of correct predictions belonging to negative class*
- ***Fall-out = (False Positive) / (True Negative + False Positive)***
  - *Fraction of negative class correctly classified*
- ***Miss Rate = (False negative) / (True positive + False negative)***
  - *Fraction of positive class misclassified*



Example

- Accuracy = (31+37+34) / 105 = ~97%
- Versicolor:
  - **TP**: 37
  - **FP**: 1
  - **FN**: 2
  - **TN**: 65
  - **Precision** = TP/TP + FP = 37/(37+1) = **0.973**
  - **Recall** = TP/TP + FN = 37/(37+2) = **0.948**
  - **F1** = TP / [TP + 0.5*(FP+FN) = 37/(37+0.5*(1+2)) = **0.961**

Predicted

Actual

	setosa	versicolor	virginica
setosa	31	0	0
versicolor	0	37	2
virginica	0	1	34

R code:

https://rpi.box.com/s/ct1l2cbz06j4qev5euzqw6l1v46uxyt1

Thanks!