Melbourne Bioinformatics

BIOINFORMATICS + DATA SERVICES + INFRASTRUCTURE, FOR LIFE SCIENCES TODAY

THE UNIVERSITY OF MELBOURNE

COMP90014

Algorithms for Bioinformatics

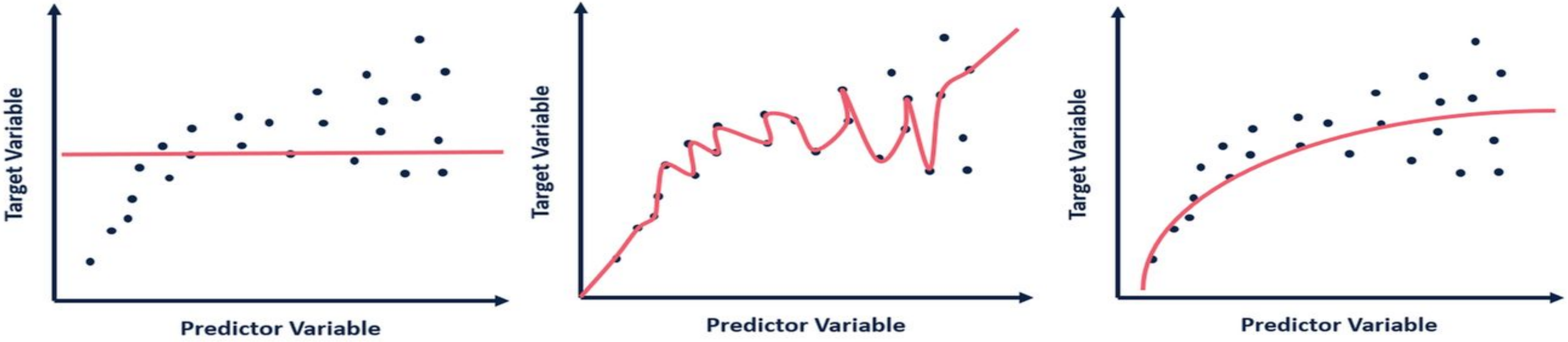
Week 11B: Model Selection | Tuning | Validation

Supervised Learning

- Our goal: Generalization
- Underfitting and Overfitting
- Learning Algorithms
  - K-nearest neighbours (KNN)
  - Naïve Bayes
  - Decision Trees
  - Support Vector Machines (SVMs)
  - Ensemble methods
- Model validation
  - Hold-out & Cross-validation
- Evaluation Metrics
  - Classification
    - Confusion matrix
      - Type I/II errors
    - ROC curves
    - Imbalanced data
  - Regression
    - Correlation coefficient
    - Mean Squared Error
- Tools and Packages

Performance Estimation

- **Overfitting** and **underfitting**

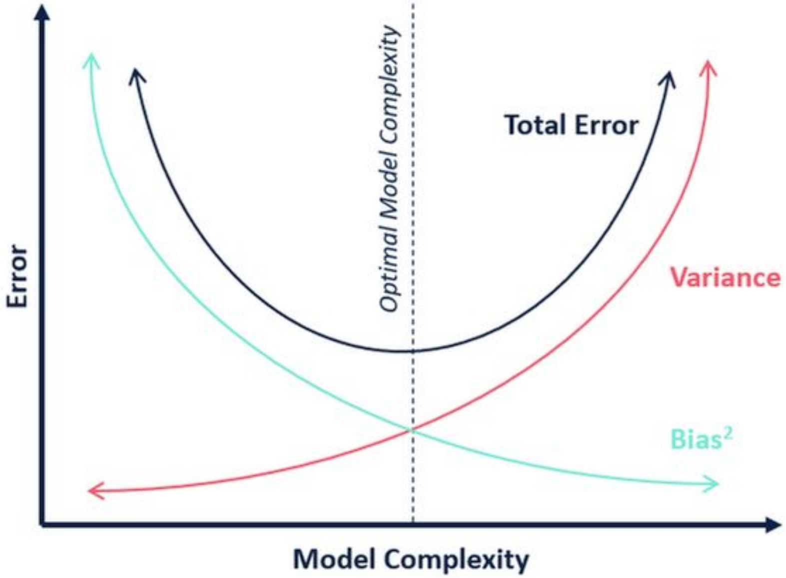


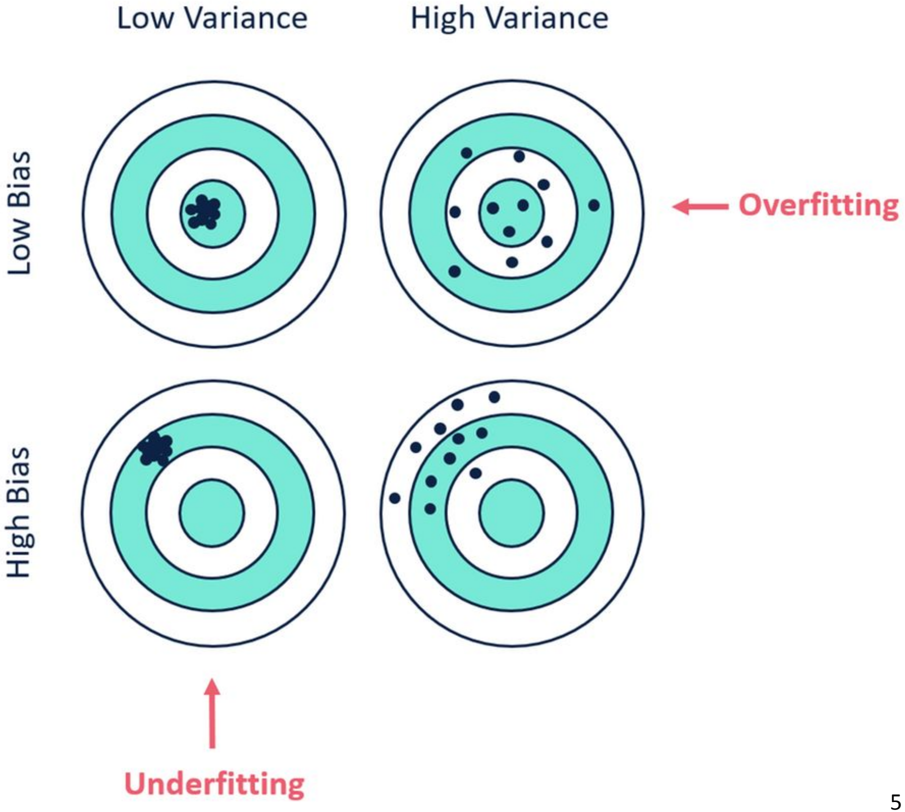
- **Bias**
  - Difference between prediction and real outcome
- **Variance**
  - Variability of predictions

https://community.alteryx.com/t5/Data-Science/Bias-Versus-Variance/ba-p/351862

Performance Estimation

- **Overfitting**
  - Low bias, high variance
- **Underfitting**
  - High variance, low bias

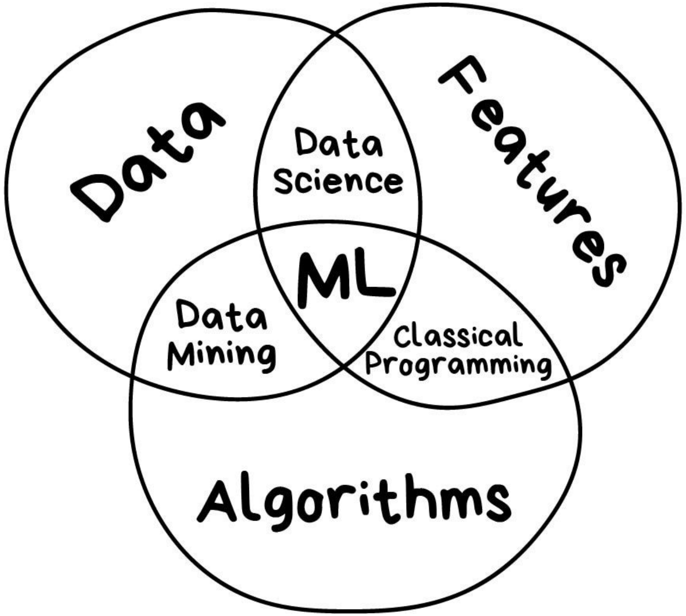




https://community.alteryx.com/t5/Data-Science/Bias-Versus-Variance/ba-p/351862

Performance estimation

- How do we know that it generalizes well to unseen data rather than simply memorize the training data?
    - And how do we select a **good predictive model**?
    - Perhaps a different **algorithm** would be more appropriate?
    - Do we need more (or higher quality) **data**?
    - Do we need to investigate different **features**?



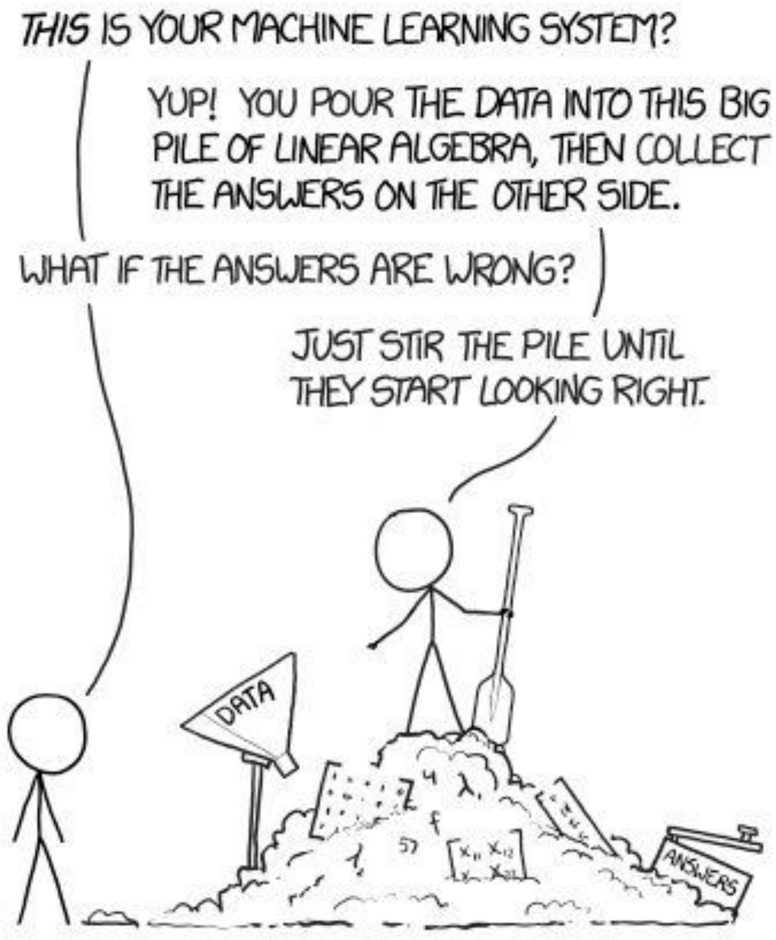
https://i.vas3k.ru/7r8.jpg

Performance Estimation

- Avoid **overfitting** and **underfitting**
  - A robust validation strategy
  - Choosing the right performance metrics

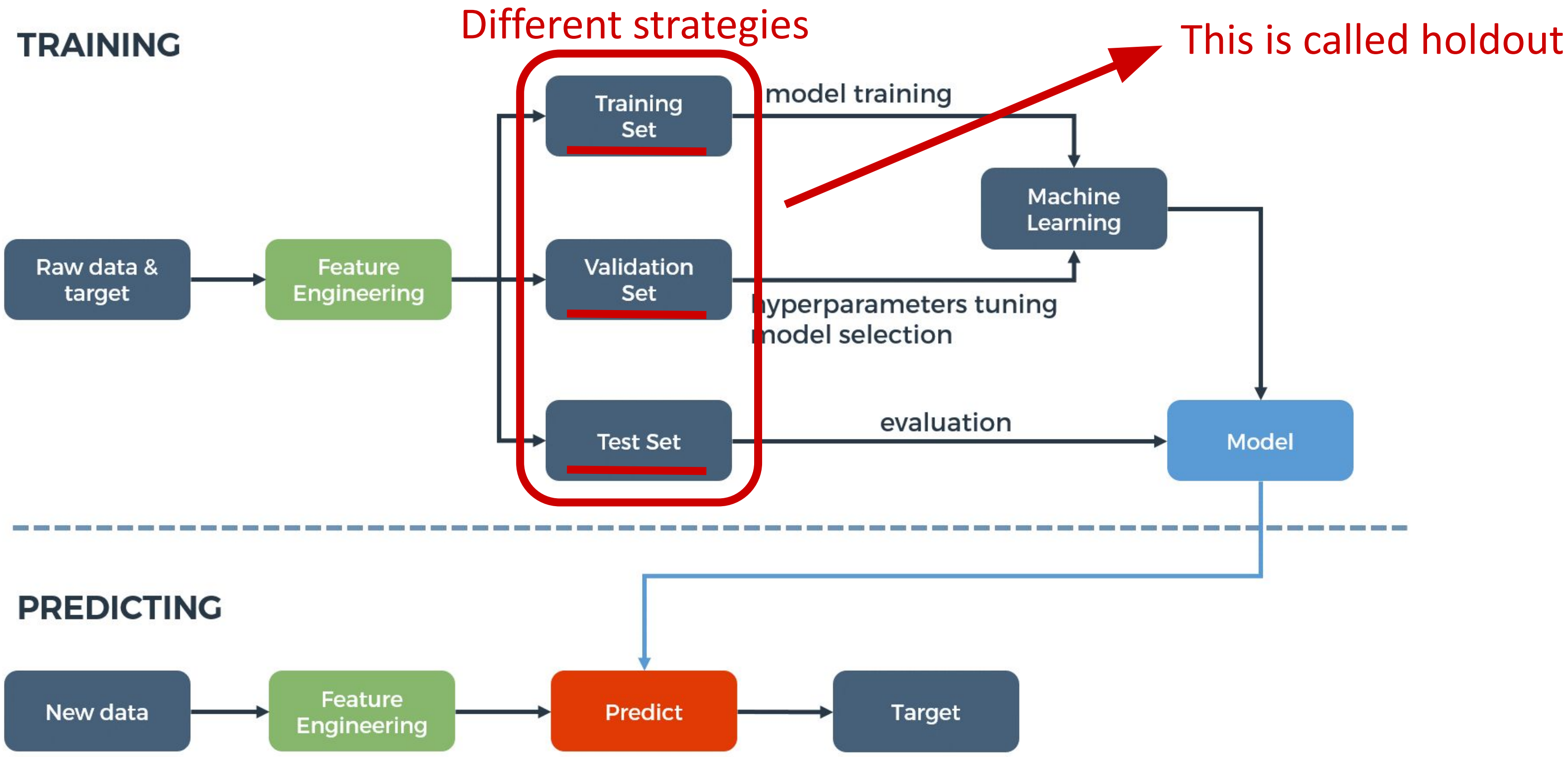
- We want to estimate the **generalization performance**
  - Predictive performance on unseen data

- We need to be able to **compare predictive models** to choose the **best/most appropriate one**
  - Assessing different algorithms,
  - Parameters and
  - Feature combinations



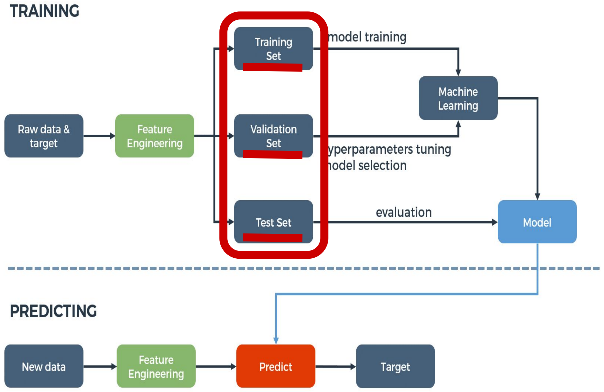
https://pbs.twimg.com/media/DpuxyjqWkAAZk2c.jpg

Model Selection and Validation



https://hackernoon.com/hn-images/1*oU3LAye3LxFcHg0UePmbSA.png

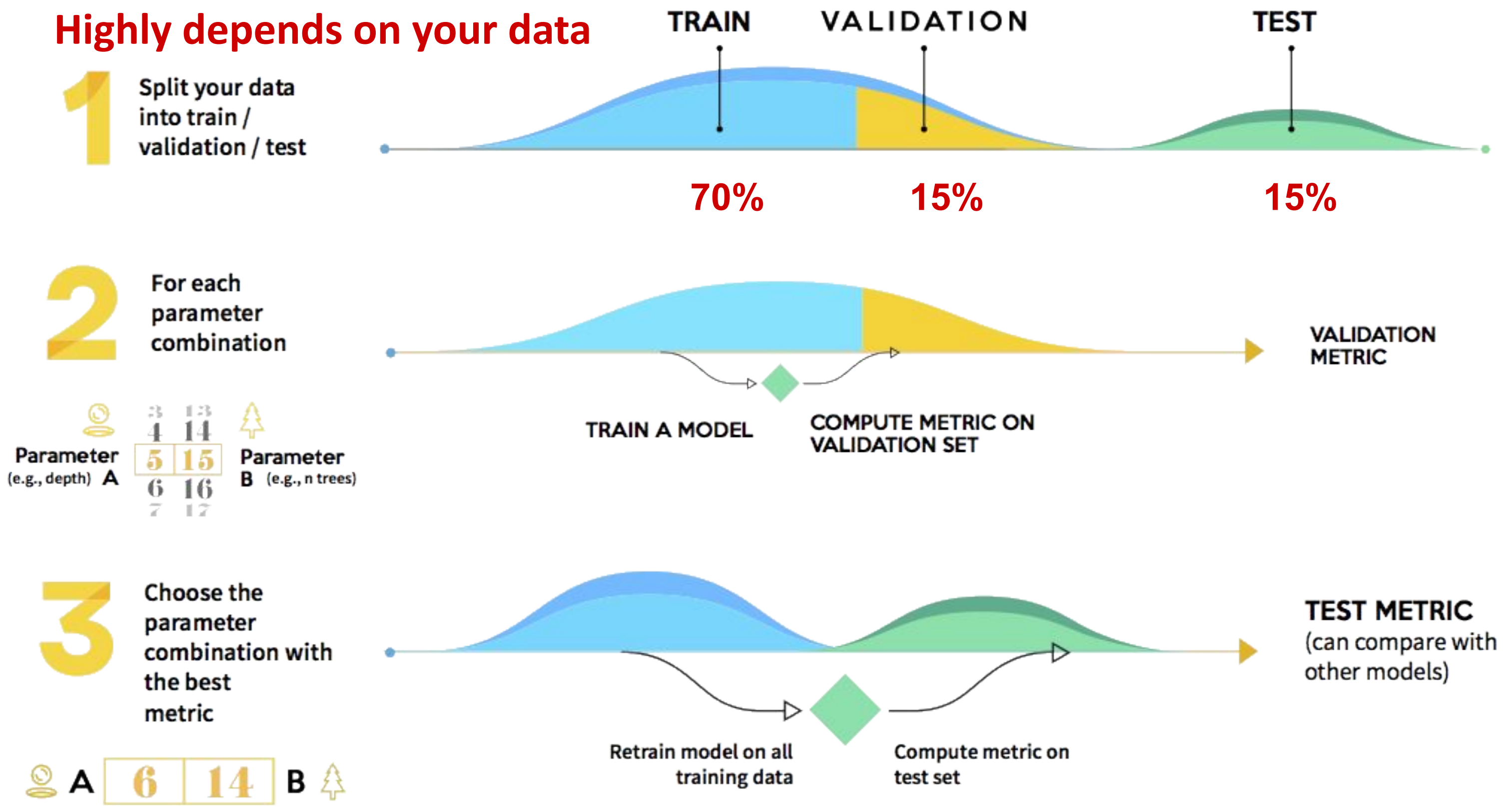
Model Validation: Holdout



- **Holdout Validation**
  - You can't use the same data to train and test the model
    - That would be cheating!
  - Test a model on different data than it was trained on
    - Provides an unbiased estimate of learning performance
  - **Holdout:** dataset is randomly divided into <u>three subsets</u>
    - Training set to build predictive models
    - Validation set to assess performance in training and to fine-tune parameters
    - Independent test set (blind test) to assess the likely performance on unseen data
  - What if performance on training is much better than on the test set? **Overfitting**

https://hackernoon.com/hn-images/1*oU3LAye3LxFcHg0UePmbSA.png

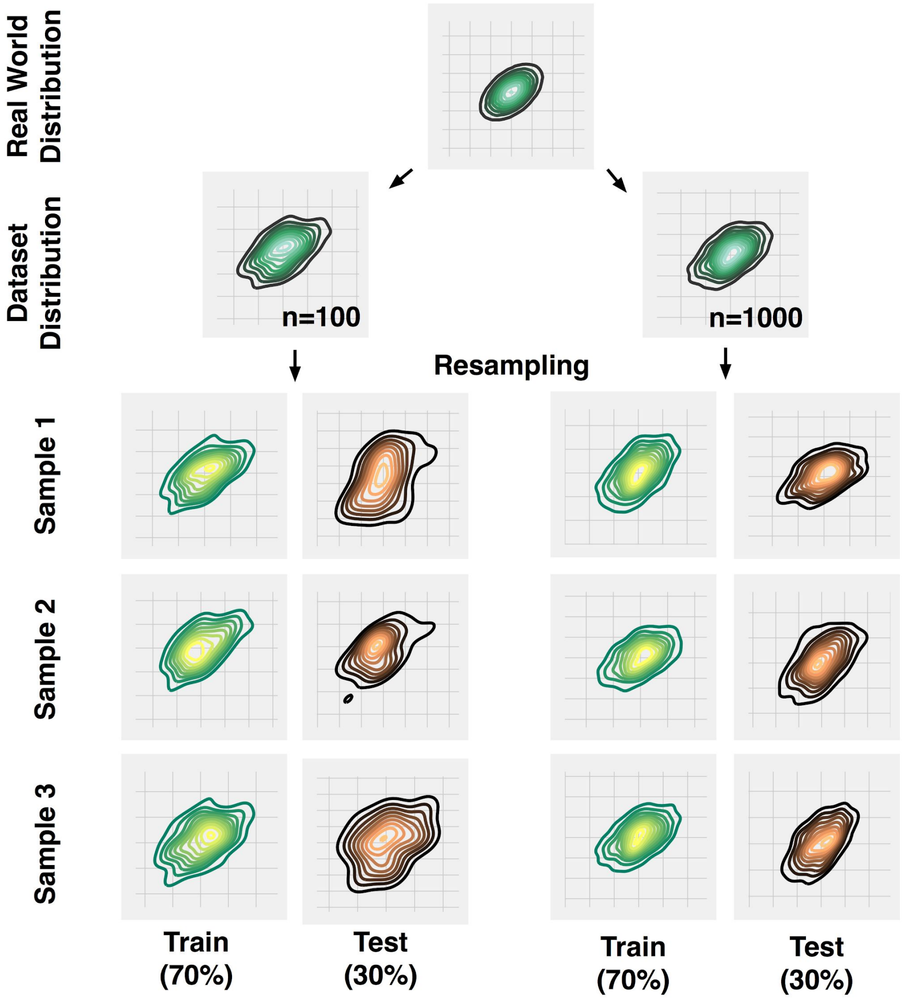
HOLDOUT STRATEGY



https://www.kdnuggets.com/wp-content/uploads/dataiku-kfold-strategy.jpg

Model Evaluation

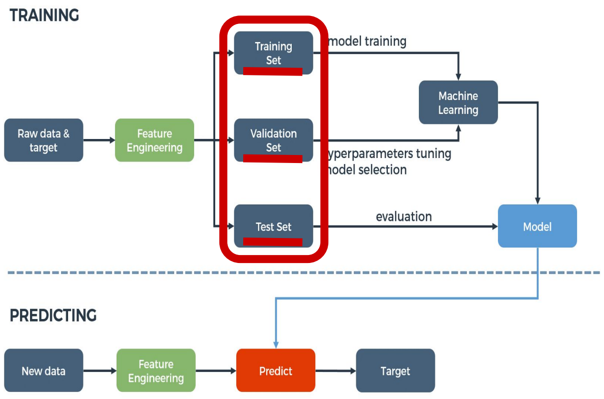
- **Holdout Strategy Limitation**
  - Prone to selection bias
- **Solution**
  - Repeated Holdout Validation
    - 100x, 1000x
  - Bootstrapping
    - Resampling with replacement



https://sebastianraschka.com/images/blog/2016/model-evaluation-selection-part2/resampling_gauss.png

Model Validation: K-fold Cross Validation

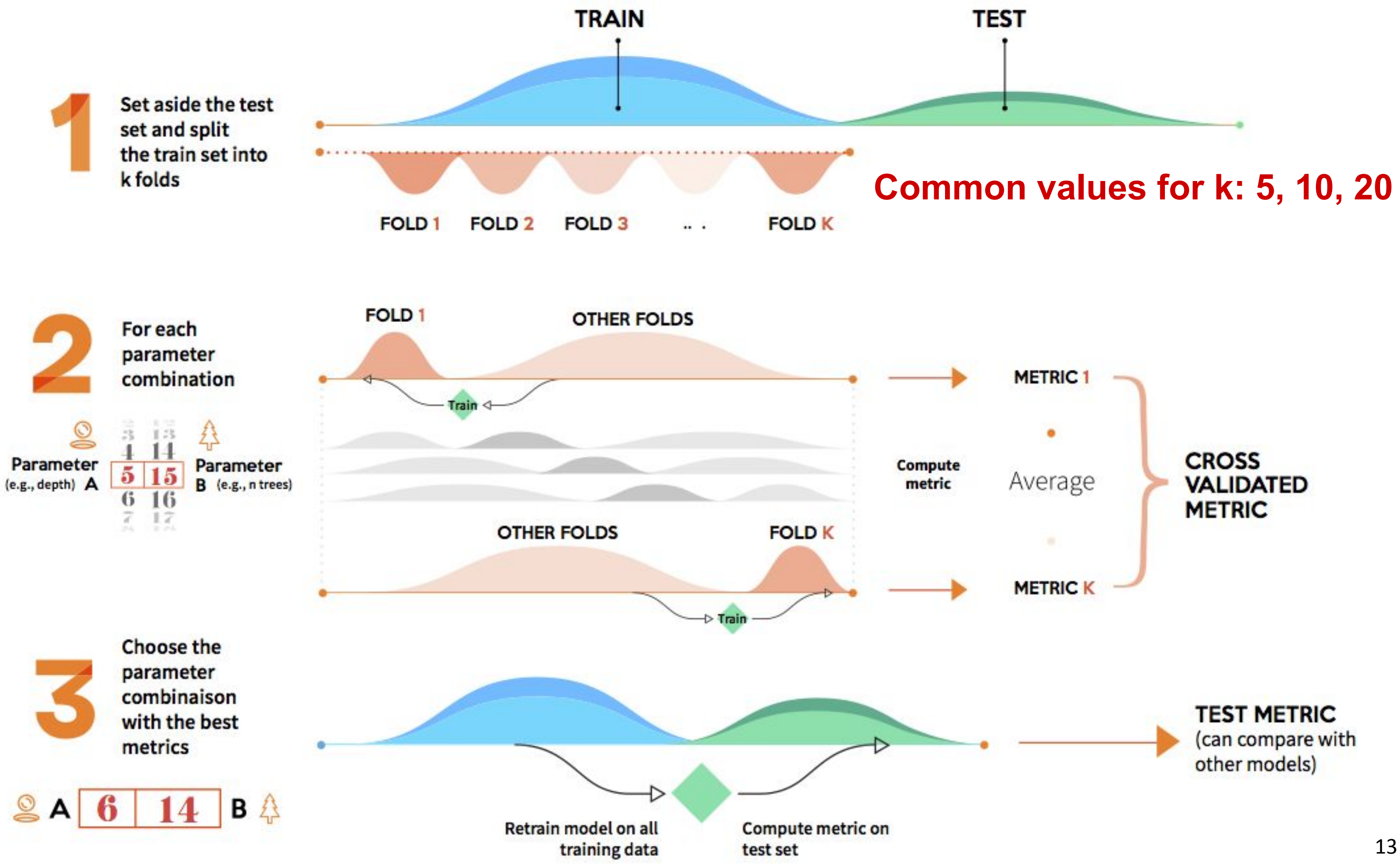
- **K-fold Cross Validation**
  - Partition the dataset into k equal size subsamples (folds)
  - We will iteratively use one of the subsets as validation and the other k-1 subsets are put together to form a training set
  - Performance estimation is averaged over all k iterations
  - Every data point gets to be in a validation set exactly once and gets to be in a training set k-1 times
  - Reduces bias
    - Most of the data for training
  - Reduces variance
    - All data is eventually used for validation



- We still need a independent test set

https://www.kdnuggets.com/wp-content/uploads/dataiku-holdout-strategy.jpg

K-FOLD STRATEGY



Common values for k: 5, 10, 20

https://www.kdnuggets.com/wp-content/uploads/dataiku-holdout-strategy.jpg

Cross Validation

- **Leave-one-out Cross Validation (LOOCV)**
  - K-fold Cross Validation when K equals the number of data points we have for training/validation
  - Per round, each data point is considered individually in the validation/evaluation set
  - Maximizes the amount of information available for training
    - Good for small data sets
  - Computationally intensive

https://sebastianraschka.com/images/blog/2016/model-evaluation-selection-part3/loocv.png

Predictive Performance Metrics

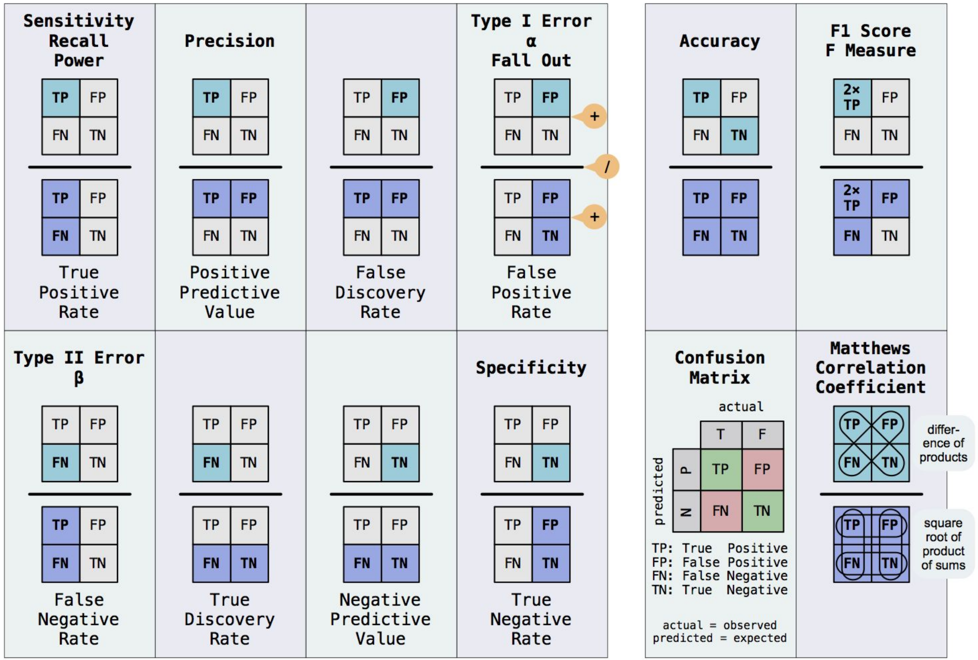
- To select the **best performing model** we need o be able to **compare them**

- **Predictive performance metrics**
  - On the validation set
  - On an independent test set
    - Make sure they are consistent

- There are multiple metrics for **classification** and **regression**

- **Classification**
  - We can derive several metrics from a confusion matrix

Statistical Classification Metrics



https://www.bluemontlabs.com/images/statistical-classification-metrics.png

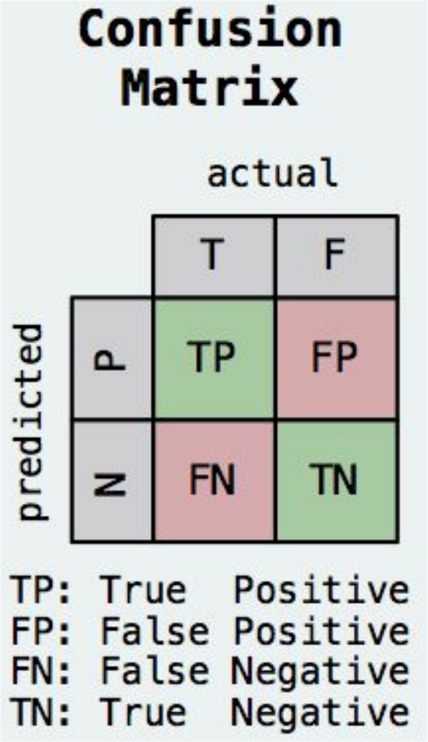
Confusion Matrix

- Table layout that allows visualisation of predictive performance
  - Rows represent the instances in a predicted class
  - Columns represent the instances in an actual class

- For binary classification (two classes, **positive** and **negative**)

- **True Positives (TP)**
  - Correctly predicted as belonging to the positive class
  - A cancer test correctly identifying a patient who has cancer

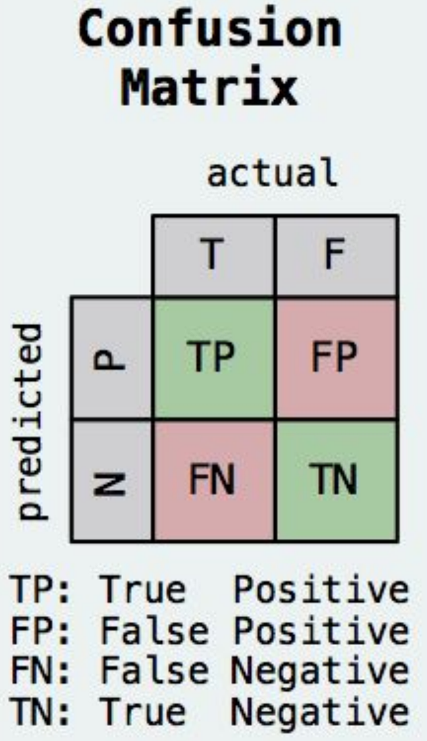
- **True Negatives (TN)**
  - Correctly predicted as belonging to the negative class
  - A cancer test correctly identifying a patient who doesn’t have cancer



https://www.bluemontlabs.com/images/statistical-classification-metrics.png

Confusion Matrix

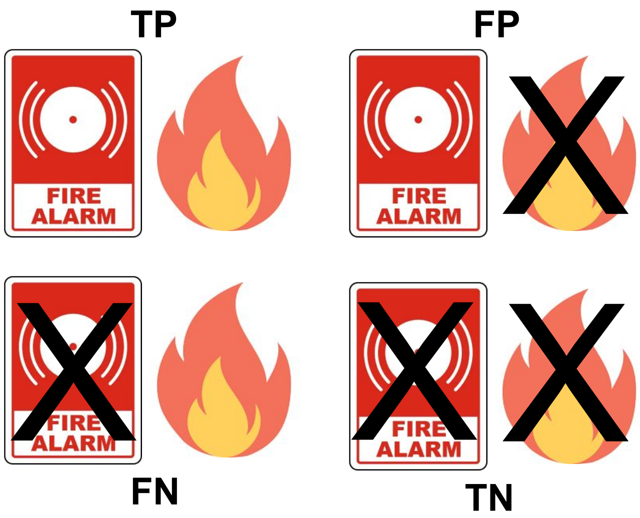
- **False Positives (FP) - Type I Error**
  - Incorrectly predicted as belonging to the positive class
  - A cancer test saying a patient has cancer, while they actually don't
- **False Negatives (FN) - Type II Error**
  - Incorrectly predicted as belonging to the negative class
  - A cancer test saying a patient doesn't have cancer, while they actually do
- **We want to:**
  - Minimize False* & Maximize True* cases
  - Which Type Error is worse?
    - Depends on the <u>problem</u>



https://www.bluemontlabs.com/images/statistical-classification-metrics.png

Type I and II errors

- **Which Type Error is worse?**
  - Fire alarm

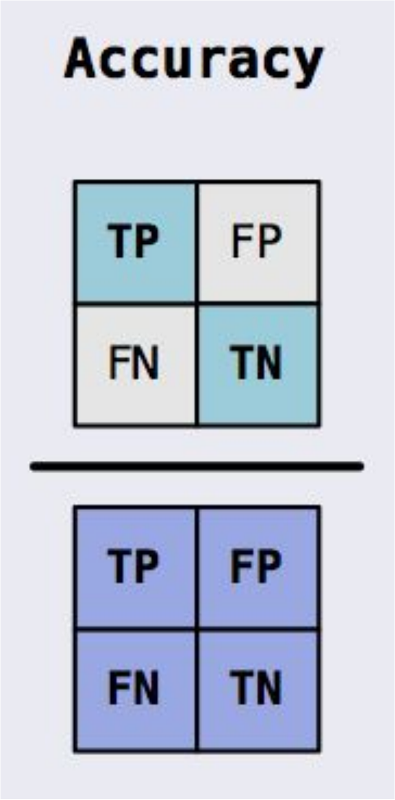


- **Type I:**
  - Fire alarm rings when there is no fire
- **Type II:**
  - Fire alarm fails to ring when there is fire

- **Which Type Error is worse** in this case?

Accuracy

- **Accuracy** is the number of **correct predictions** made by the model **over all predictions** made
  - `Accuracy =(TP+TN)/(TP+TN+FP+FN)`
- **When can I use Accuracy?**
  - Accuracy is a good measure when the target classes in the data are nearly balanced
  - Similar number of data points belonging to each class
- **When NOT to use Accuracy?**
  - Imbalanced data sets
  - *e.g.,* 95% of the data belong to class A, 5% to class B
    - A predictor that only guesses class A has 95% accuracy



https://www.bluemontlabs.com/images/statistical-classification-metrics.png

Accuracy

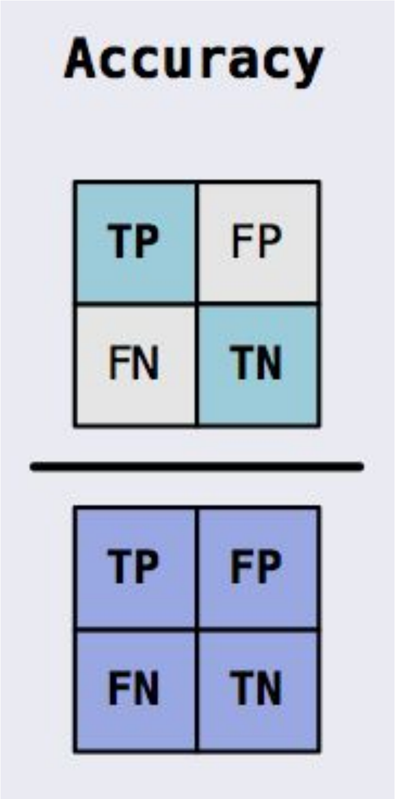
Spam filter (25 spam messages, 125 not spam)

- **73.3%**

	Spam	**Not spam**
Pred. spam	10 (TP)	25 (FP)
Pred. not-spam	15 (FN)	100 (TN)

- **83.3%**

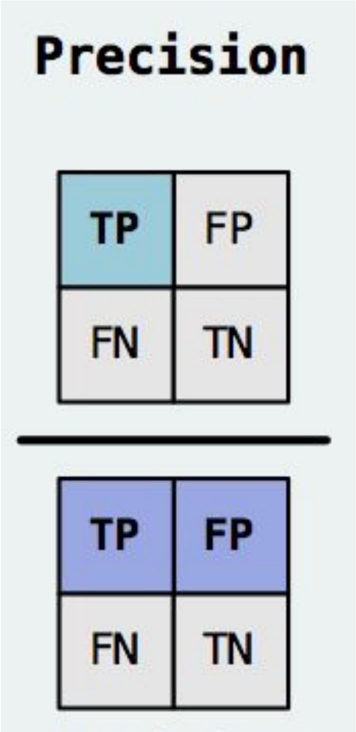
	Spam	**Not spam**
Pred. spam	0 (TP)	25 (FP)
Pred. not-spam	0 (FN)	125 (TN)



https://www.bluemontlabs.com/images/statistical-classification-metrics.png

Precision

- **Precision** is the proportion of predicted positives that truly are positives
  - Precision =(TP)/(TP+FP)
- Takes into account Type I Error
- **Precision** is a valid choice of evaluation metric when we **want to be very sure** of our **positive class prediction**.
  - *e.g.,* if we want to to predict if we should decrease the credit limit on a particular account
    - We want minimum FP otherwise it may result in customer dissatisfaction
    - Maximise precision



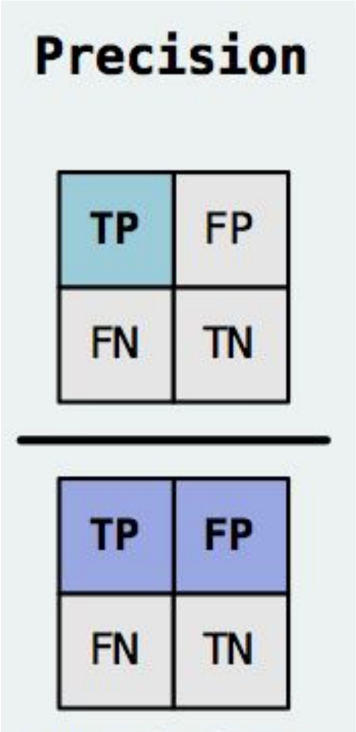
https://www.bluemontlabs.com/images/statistical-classification-metrics.png

Precision

Spam filter (10 spam messages, 90 not spam)

	Spam	Not spam
Pred. spam	1 (TP)	0 (FP)
Pred. not-spam	9 (FN)	90 (TN)

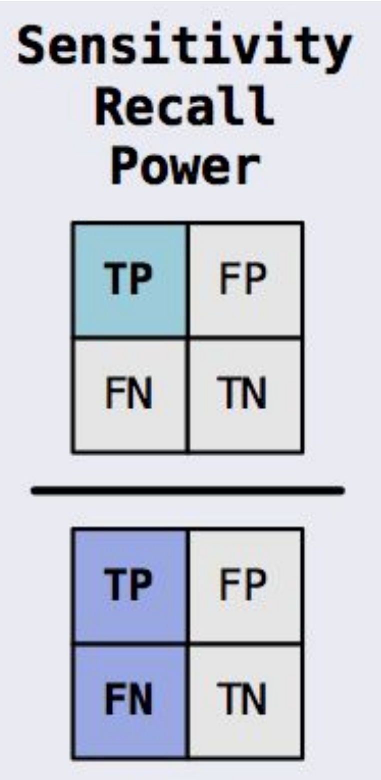
- What's the **precision** for the spam class?
  - **100%**



https://www.bluemontlabs.com/images/statistical-classification-metrics.png

Recall or Sensitivity

- **Recall** is the proportion of actual positives that are correctly classified
  - `Recall =(TP)/(TP+FN)`
- Takes into account Type II Error
- **Recall** is used when we **want to recover as many positives as we can**
  - *e.g.*, If we want to predict if a patient has a disease or not, we want to capture the disease even if we are not very sure
    - We want minimum FN otherwise we might discharge a patient that needs treatment - maximise recall
- **Caveat**
  - **If we predict everything as positive, recall will be 100%**

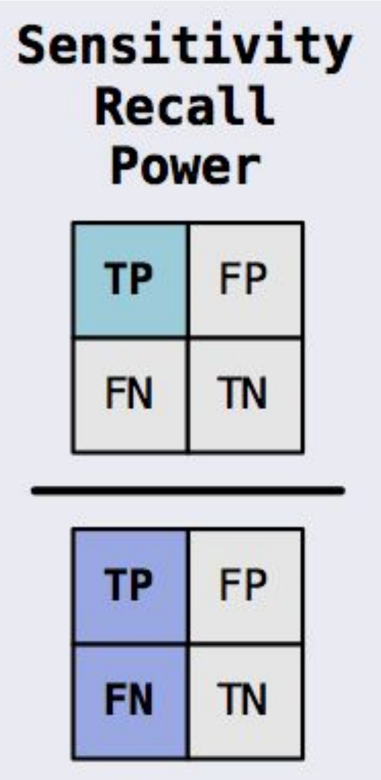


https://www.bluemontlabs.com/images/statistical-classification-metrics.png

Recall or Sensitivity

Spam filter (10 spam messages, 90 not spam)

	Spam	Not spam
Pred. spam	10 (TP)	90 (FP)
Pred. not-spam	0 (FN)	0 (TN)



- What's the **recall** for the spam class?
  - **100%**

https://www.bluemontlabs.com/images/statistical-classification-metrics.png

F1-score or F1-measure

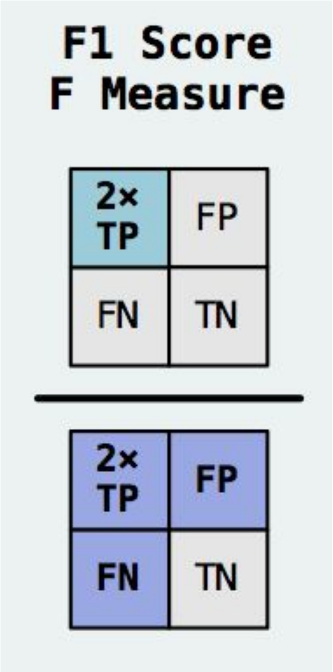
- **How to find a compromise between precision and recall?**
  - Does simply taking their arithmetic mean work?

  - *e.g.*, a predictive model with 20% recall and 100% precision
    - `mean(Precision+Recall) = 60%`

- F1-score is the harmonic mean of precision and recall

  - `F1 = 2*(Precision*Recall)/(Precision+Recall)`
  - For the example: `F1 = 33%`

- F1-score will penalise large discrepancies between precision and recall

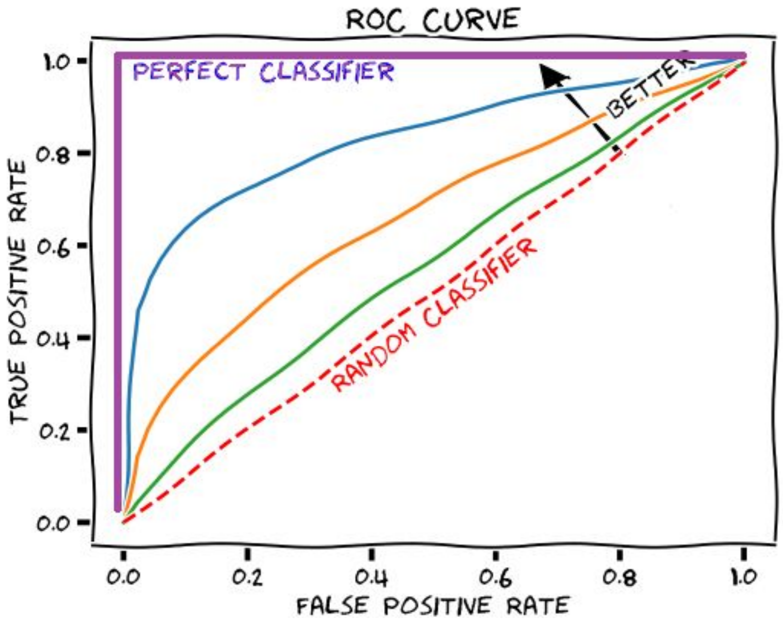


https://miro.medium.com/max/532/1*5XuZ_86Rfce3qyLt7XMlhw.png

Receiver Operating Characteristic Curves

- **ROC curves**
  - Is a graph showing the performance of a classification model at different thresholds
    - *i.e.*, class probabilities from our classifier
  - Y-axis: true positive rate (recall)
  - X-axis: false positive rate (1-specificity)
    - Specificity = TN/(TN+FP)

- **AUC** (Area Under the ROC Curve)
  - Varies from 0 to 1
  - A random binary classifier: AUC of 0.5



Check it out! (StatQuest Channel)
https://www.youtube.com/watch?v=4jRBRDbJemM

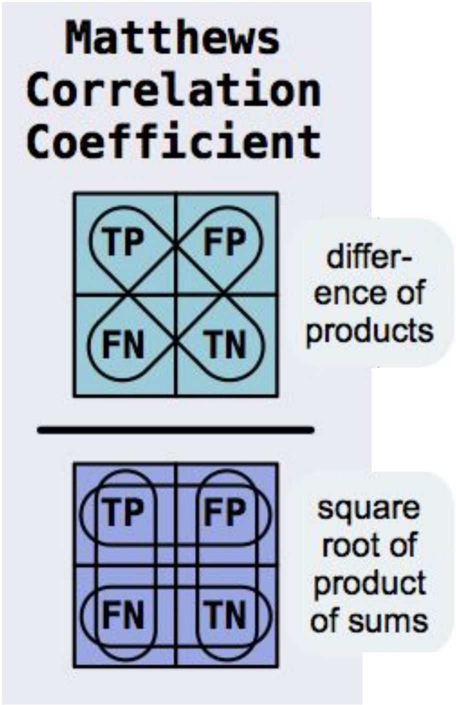
https://qph.fs.quoracdn.net/main-qimg-f76391a3bea78ef9bc7f34f09f553646

Matthews Correlation Coefficient (MCC)

- **Matthews Correlation Coefficient (MCC)** takes into account true and false positives and negatives
  - It is considered a **balanced metric**

- Very good metric for **imbalanced** data sets
  - Even classes are of very different in sizes
  - In contrast with accuracy

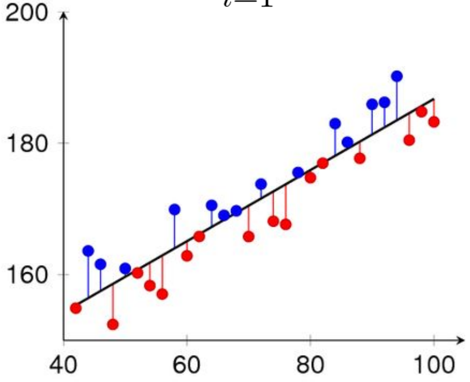
- **Ranges between −1 and 1**
  - 1 shows a perfect prediction
  - 0 equals to the random prediction
  - −1 indicates total disagreement between predicted and actual labels



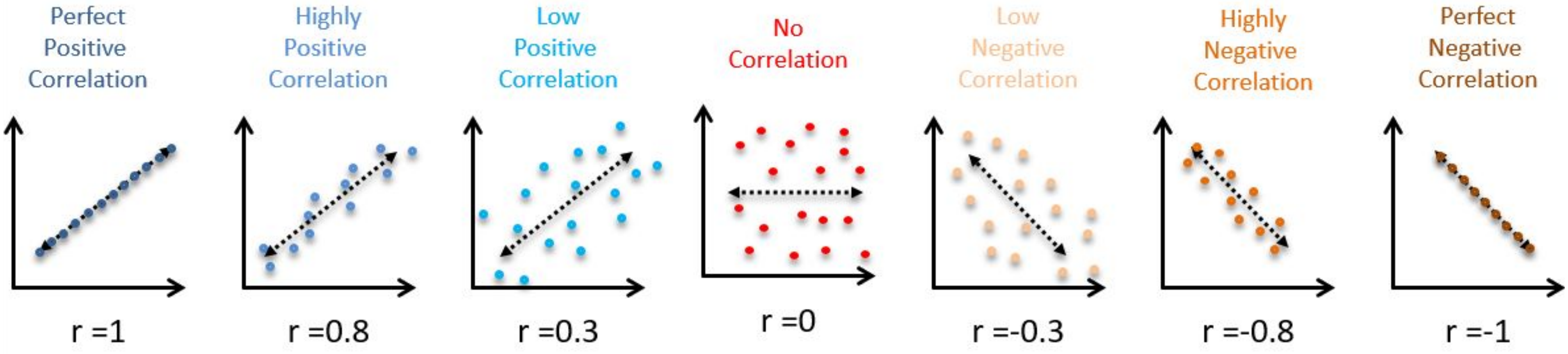
$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FN} \times \mathrm{FP}}{\sqrt{(\mathrm{TP}+\mathrm{FN})(\mathrm{TP}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})}}$$

Evaluating regression models

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



- **Mean Square Error (MSE)**
  - The average of squared differences between the predicted predicted and the actual values
- **Pearson Correlation Coefficient**
  - A measure of the linear correlation between two variables (predicted vs. actual)



http://www.cqeacademy.com/wp-content/uploads/2018/06/Scatter-Plots-and-Correlation-Examples.png

Which models is the best one?

Metric	MODEL1	MODEL2
Recall	0.6667	**0.8333**
Specificity	**0.8333**	0.6667
Precision	**0.8000**	0.7143
Accuracy	0.7500	0.7500
F1 Score	0.7273	**0.7692**
MCC	0.5071	0.5071

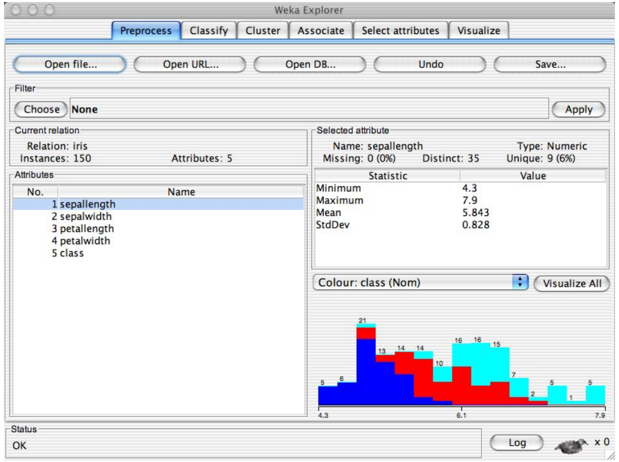
- **Model 1**
  - To minimise Type I Error (better precision)
- **Model 2**
  - To minimise Type 2 Error (better recall)
- **Benign *vs.* malignant cancer hypothetical cases**

MODEL 1	Actual disease	Actual healthy
Predicted disease	200	50
Predicted healthy	100	250

MODEL 2	Actual disease	Actual healthy
Predicted disease	250	100
Predicted healthy	50	200

Tools and Packages

- **Scikit Learn**
  - Machine Learning in Python
  - Data analysis
  - Built on NumPy, SciPy, and matplotlib
  - Open source
- **TensorFlow**
  - Python
  - Developed by Google
  - Deep Learning
- **Weka**
  - GUI
  - Java



Melbourne Bioinformatics
BIOINFORMATICS + DATA SERVICES + INFRASTRUCTURE, FOR LIFE SCIENCES TODAY

THE UNIVERSITY OF MELBOURNE

Thank you!

Today: Model selection, tuning, validation

Next time: Recap I