KU LEUVEN

LMSD Mecha(tro)nic System Dynamics

FLANDERS MAKE

How to measure if your condition monitoring models are successful.

MOIRA Industrial training conference

Wrocław 2023/06

Douw Marx

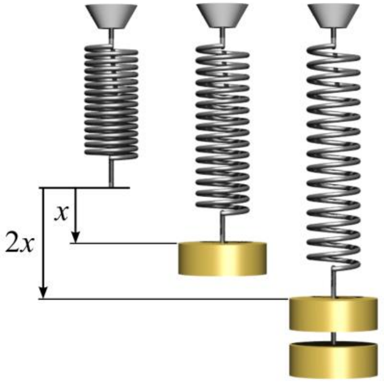
Contents

- Introduction to data driven models
- Splitting data for evaluation
- Metrics for evaluating models
- Open discussion of case studies from the audience

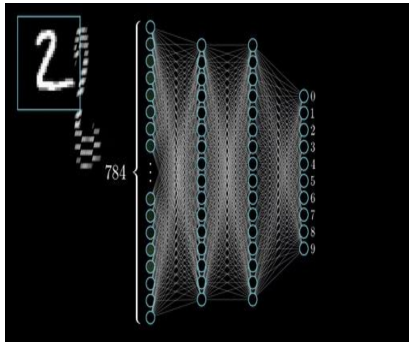
What is a data-driven model?

- Make a simplification of reality that describes your data.
- Choose a mathematical model (function) that could fit to your data.
- Choose the right model parameters that would let it perform best on new, unseen data.

How we measure if the model is performing is critical!



This Photo by Unknown Author is licensed under CC BY-SA

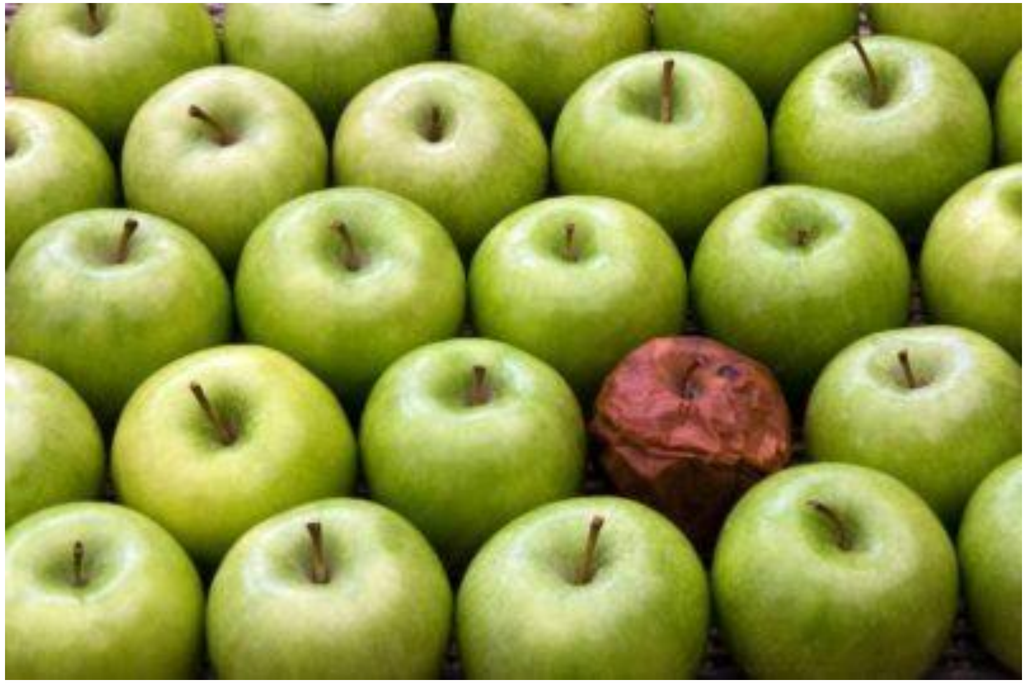


- F = kx
- Learn the parameter “k” from data

- output = f(input)
- Learn hundreds to billions of parameters “w” from data

If it works, it works. Who cares about model evaluation?

- Model evaluation tells you if the model "is working".
- If your measure of if the model "is working" is not effective it can have consequences.



Evaluate the following model with test data on the left:

Model: f(apple) = good
(Model that always says an apple is good)

Accuracy:
accuracy = 25/26 = 96% : High accuracy!

Is this an effective measure if the model is performing well?

Families of models commonly used in engineering

Regression

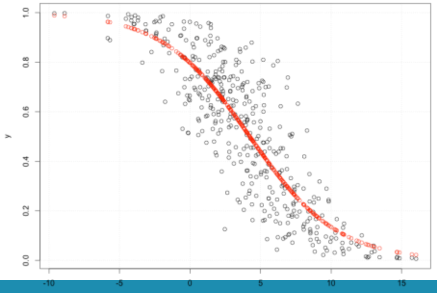
- Predicting the remaining useful life of mechanical components
- Predicting energy consumption based on process parameters

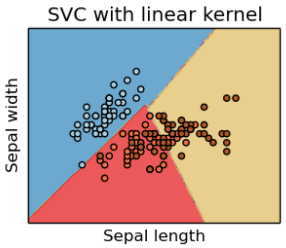
Classification

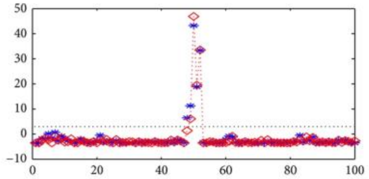
- Classifying parts as defective or normal.
- Categorize different failure modes or operational modes of a machine.

Outlier detection

- Detect anomalous sensor readings
- Detect unusual equipment behavior

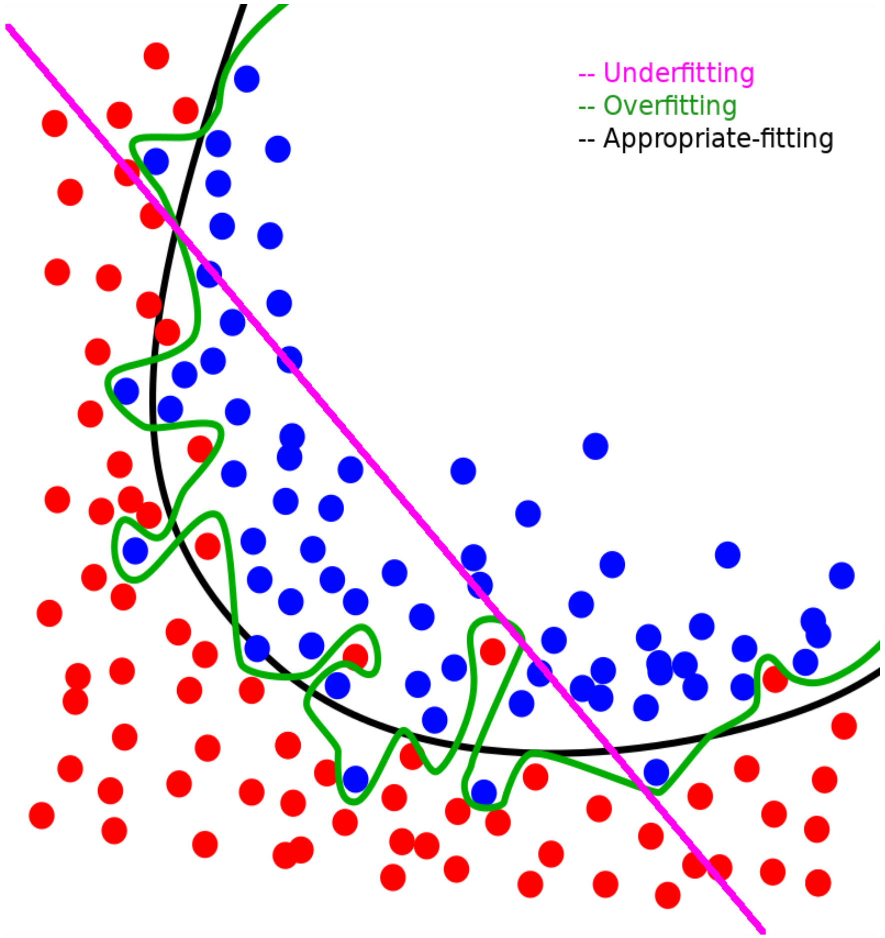






When will the model perform badly on new unseen data?

- The model is too flexible
- The model is not flexible enough
- There is not enough data to accurately capture the model parameters



Chabacano / CC BY-SA (https://creativecommons.org/licenses/by-sa/4.0)

The model performance must be measured on unseen test data to understand if the right model is being used.

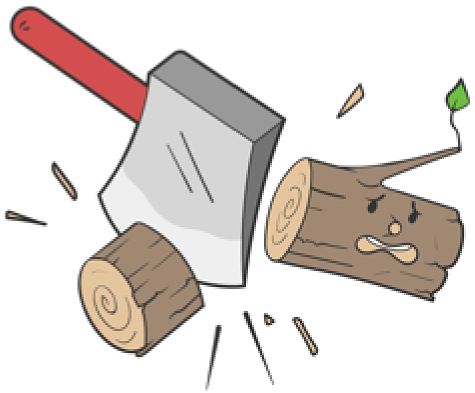
Splitting to prevent overfitting

Split/divide the data you have into different parts/sets that allows you to evaluate the model on unseen data.

- Data to train the model
- Data to evaluate the model (test data that we pretend is unseen)

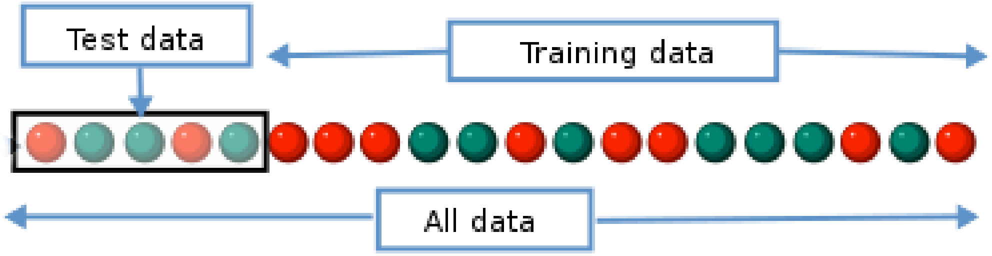
At least two ways to do this

- “Hold out”: Keep for instance 20% of data separate for testing
- “Cross Validation”: Fancy Hold Out with multiple splits



Hold-out splitting

Split the data into test and train data.
For instance, 80% train and 20% test.



Advantages:

- Simple and easy to implement.
- Requires less computational resources compared to cross-validation.
- Provides separate dataset for final evaluation of model.

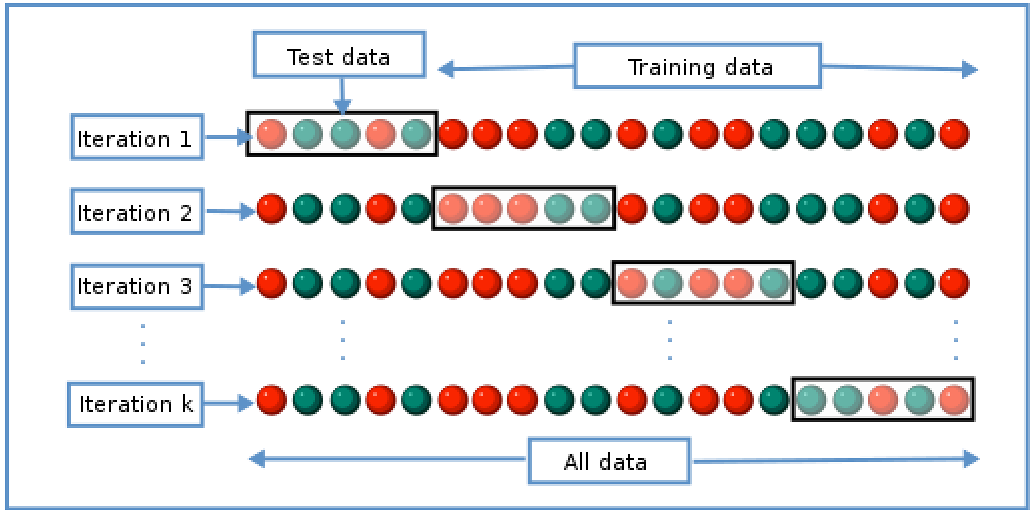
Disadvantages:

- Evaluation may be highly dependent on the particular data split.
- May not represent the overall performance of the model accurately, especially with limited data.
- Can result in overfitting or underfitting if the split is not representative of the underlying data distribution.

Cross Validation Splitting

Divide data into k number of sets (k-folds).

- Leave one set out for testing, train on the rest.
- Average metric on the test results over all folds.



Advantages:

- Robust estimate of mode performance.
- Maximizing data utilization.
- Reduce impact of data variability by averaging results across different folds.

Disadvantages:

- Computationally expensive, especially for large datasets or complex models.
- More complex to implement and interpret compared to holdout test set splitting.

Validation sets vs Test sets

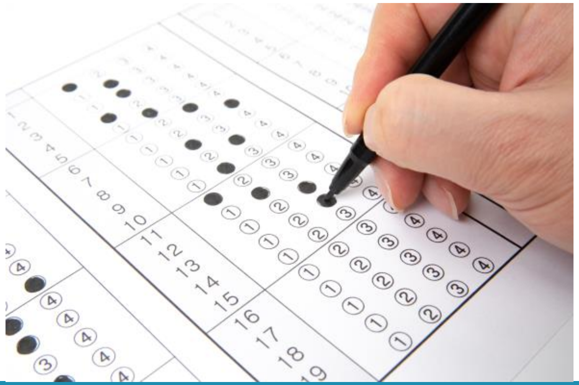
Validation Set:

- Used during training to experiment with the model/ fine tune.
- Adjust the model for better generalization to unseen data.
- Think of the validation set as an additional set split from the training data.

Test Set:

- Separate from training and validation.
- Provides an unbiased evaluation of the final model's performance.
- Serves as the benchmark for how well the model handles new, unseen data.

Validation set is like doing previous exams in preparation for the exam.
Test set is like the final exam.



Commonly used metrics

Regression

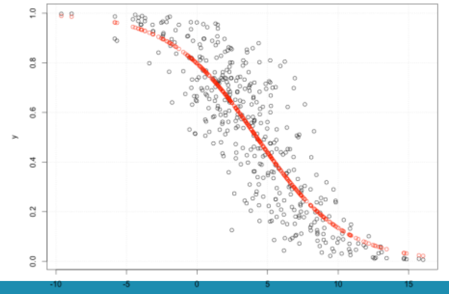
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- R-squared (Coefficient of Determination)
- Mean Absolute Percentage Error (MAPE)

Classification

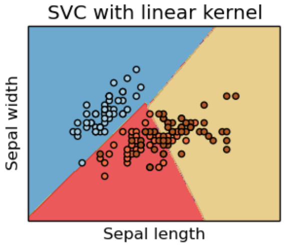
- Accuracy
- Precision
- Recall (Sensitivity)
- F1 Score
- Area Under the ROC Curve (AUC-ROC)
- Confusion Matrix

Outlier detection

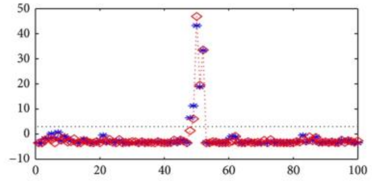
- True Positive Rate (TPR)
- False Positive Rate (FPR)
- Precision-Recall Curve
- Receiver Operating Characteristic (ROC) Curve
- Area Under the ROC Curve (AUC-ROC)
- F1 Score



This Photo by Unknown Author is licensed under CC BY-SA



This Photo by Unknown Author is licensed under CC BY-SA



This Photo by Unknown Author is licensed under CC BY

Which metric to use?

Regression

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- R-squared (Coefficient of Determination)
- Mean Absolute Percentage Error (MAPE)

Classification

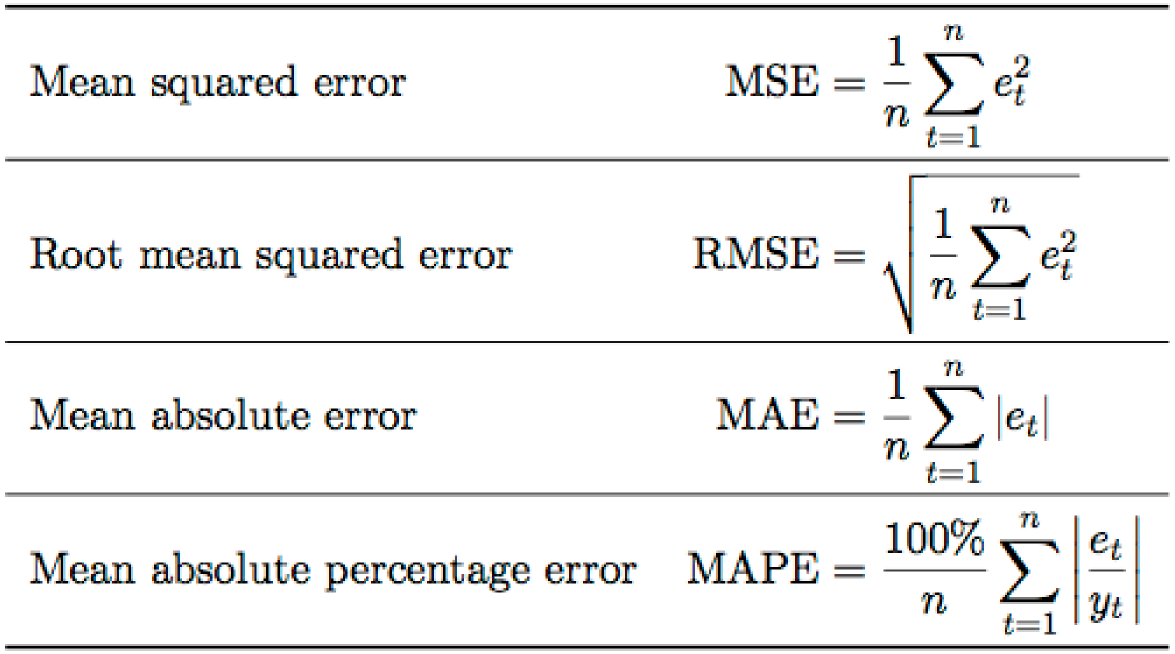
- Accuracy
- Precision
- Recall (Sensitivity)
- F1 Score
- Area Under the ROC Curve (AUC-ROC)
- Confusion Matrix

Outlier detection

- True Positive Rate (TPR)
- False Positive Rate (FPR)
- Precision-Recall Curve
- Receiver Operating Characteristic (ROC) Curve
- Area Under the ROC Curve (AUC-ROC)
- F1 Score

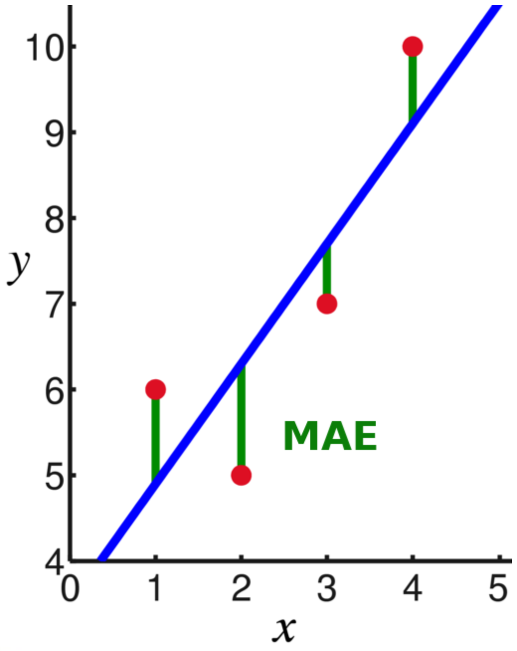
- Each metric “recipe” for measuring the model performance.
- We want to use the right “recipe” so that we can measure the right thing for our problem.
- Example: If you give people over 80 a math test on a computer, you will likely not measure their math ability, but rather their computer literacy. Need to design the right test.

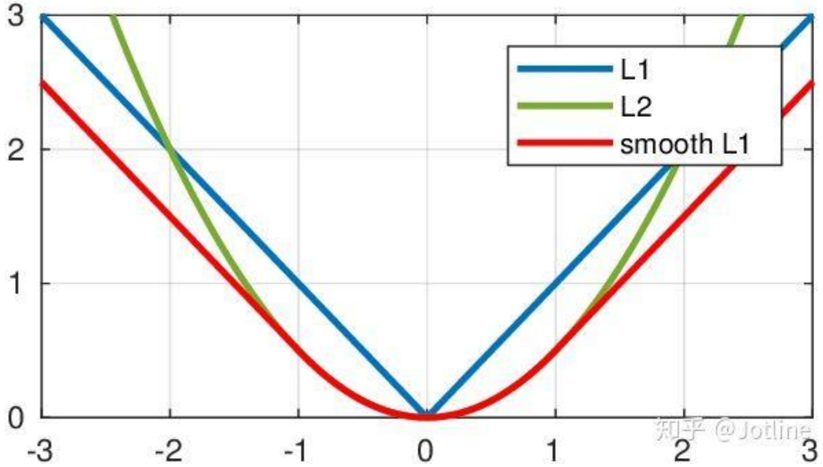
Regression models: MAE vs MSE



Mean squared error	$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2$
Root mean squared error	$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$
Mean absolute error	$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n} \lvert e_t \rvert$
Mean absolute percentage error	$\mathrm{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n} \left\lvert \frac{e_t}{y_t} \right\rvert$

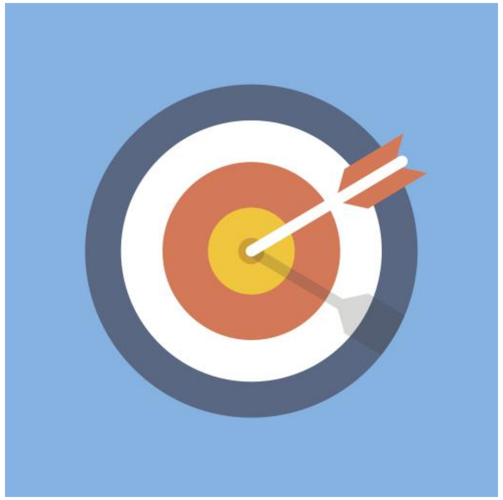
MSE penalizes large errors more strongly than MAE.
MAE is sometimes more interpretable (i.e. meter/ Newton).
Analogy with mean and median.





Classification models: Accuracy is not always the answer

- Imagine you have 950 examples of normal parts in your test data and 50 examples of abnormal parts.
- If you make a model that always predicts that a part is normal, you will still get 95% accuracy and miss all faulty parts.

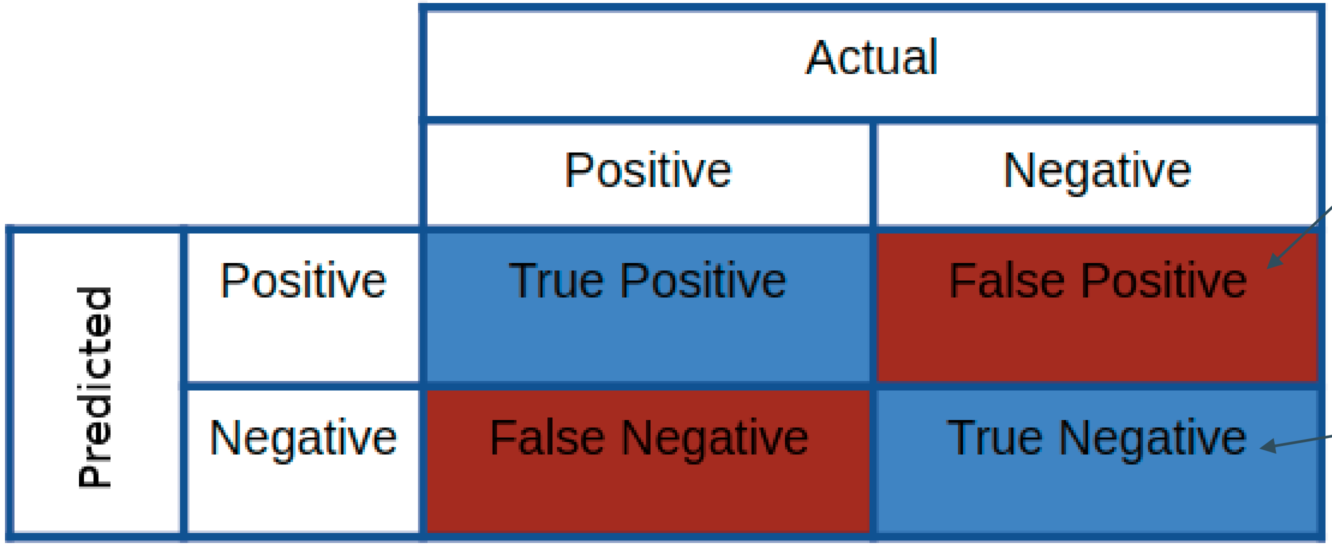


If the dataset is balanced (Similar number of examples from each class) then it is OK to use accuracy as metric.

Classification models: Alternatives to accuracy

The confusion matrix: For which classes are the model typically confused?

This is relevant because the cost of a False positive is often different than a false negative



		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

“Falsely classified as the positive class”

“Truly(correctly) classified as the negative class”

“High vibration levels detected. The machine is faulty! Send help!”
False Positive

“Everything is fine…”
False Negative

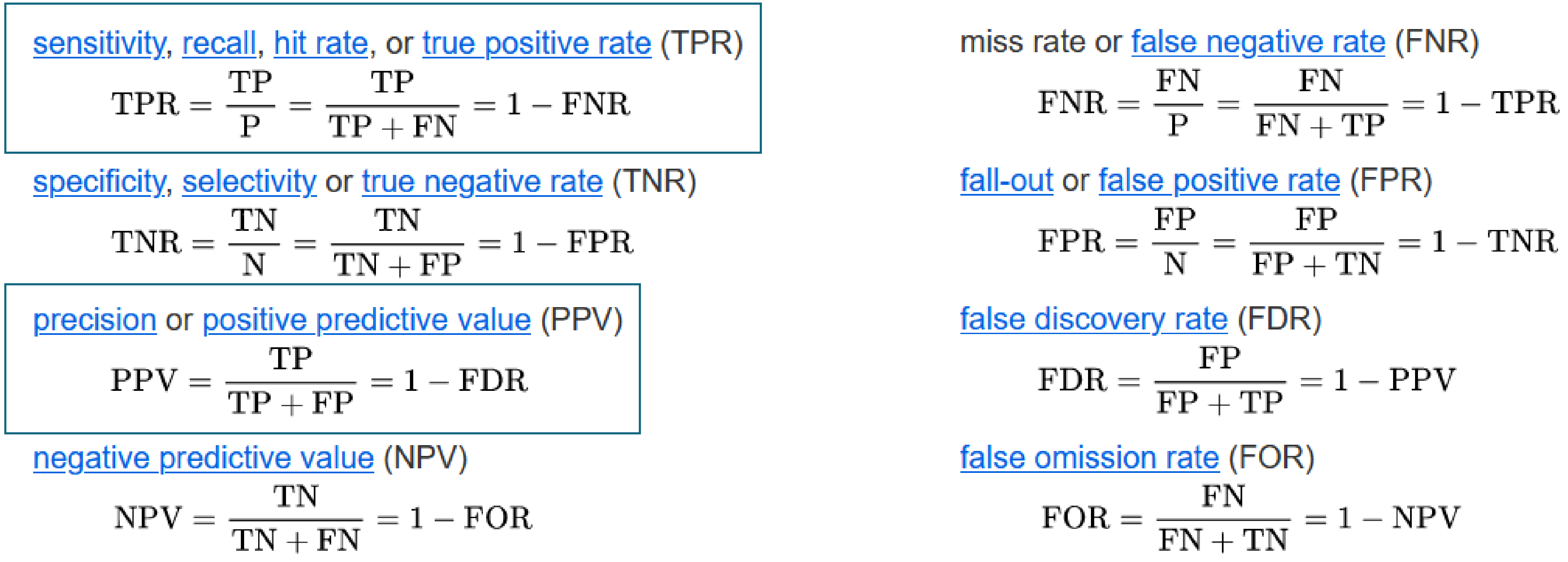
Classification models: Alternatives to accuracy

Model: f(apple) = good (Model that always says an apple is good)

		Actual	
		Positive	Negative
Predicted	Positive	True Positive 0	False Positive 0
	Negative	False Negative 1	True Negative 25

Confusion matrix cocktails

- There are many metrics derived from the confusion matrix for classification and outlier detection.
- How to make sense of these metrics?



"Accuracy on positive class"

sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = 1 - \mathrm{FNR}$$

specificity, selectivity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} = 1 - \mathrm{FPR}$$

"How often was the model correct when it was betting on the positive class"

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} = 1 - \mathrm{FDR}$$

negative predictive value (NPV)

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}} = 1 - \mathrm{FOR}$$

miss rate or false negative rate (FNR)

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{P}} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TP}} = 1 - \mathrm{TPR}$$

fall-out or false positive rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} = 1 - \mathrm{TNR}$$

false discovery rate (FDR)

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TP}} = 1 - \mathrm{PPV}$$

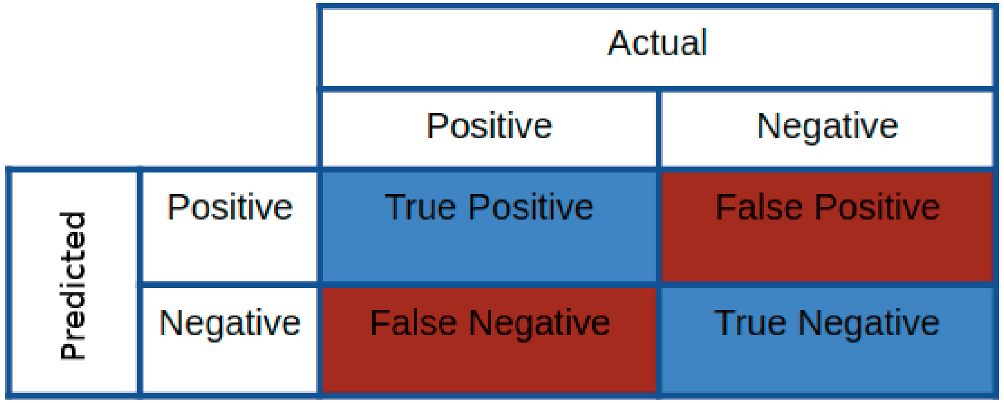
false omission rate (FOR)

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TN}} = 1 - \mathrm{NPV}$$

Precision and recall

$$recall = \frac{true\ positives}{true\ positives\ + false\ negatives} \qquad precision = \frac{true\ positives}{true\ positives + false\ positives}$$

- If your model tends to quicky predict the positive class when something seems different it has a high sensitivity/recall.
- You might have high sensitivity, but of all those positive predictions the model makes, only a small number might be correct, leading to a low precision.
- Useful when your test sets are not balanced.



		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

F1 score

- We want to find a compromise between precision (If model says the data point is positive it is actually positive) and recall/sensitivity (You have a high accuracy on the positive class).
- Measure between 0 and 1

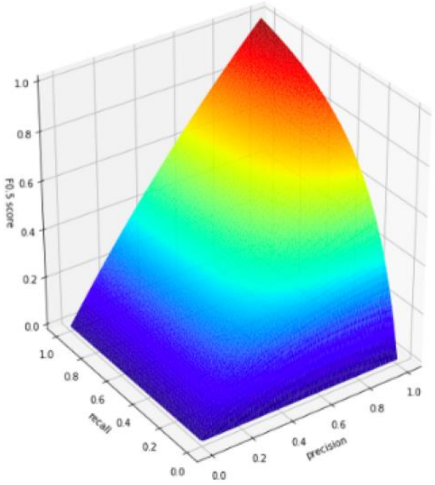
$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

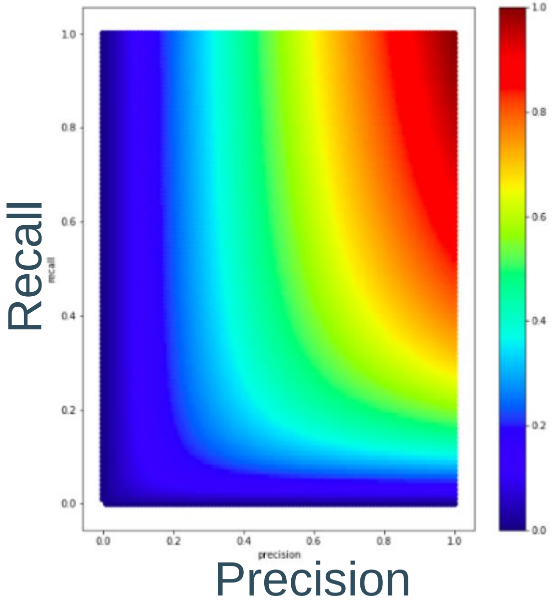
Example

- True positives (TP): 75 correctly identified as faulty machinery.
- False positives (FP): 10 samples incorrectly identified as faulty machinery.
- False negatives (FN): 5 faulty machinery samples incorrectly classified as non-faulty.
- True negatives (TN): 10 samples were correctly identified as non-faulty machinery.

- Precision = TP / (TP + FP) = 75/(75+10) = 0.882
- Recall = TP / (TP + FN) = 75/(75+5) = 0.938

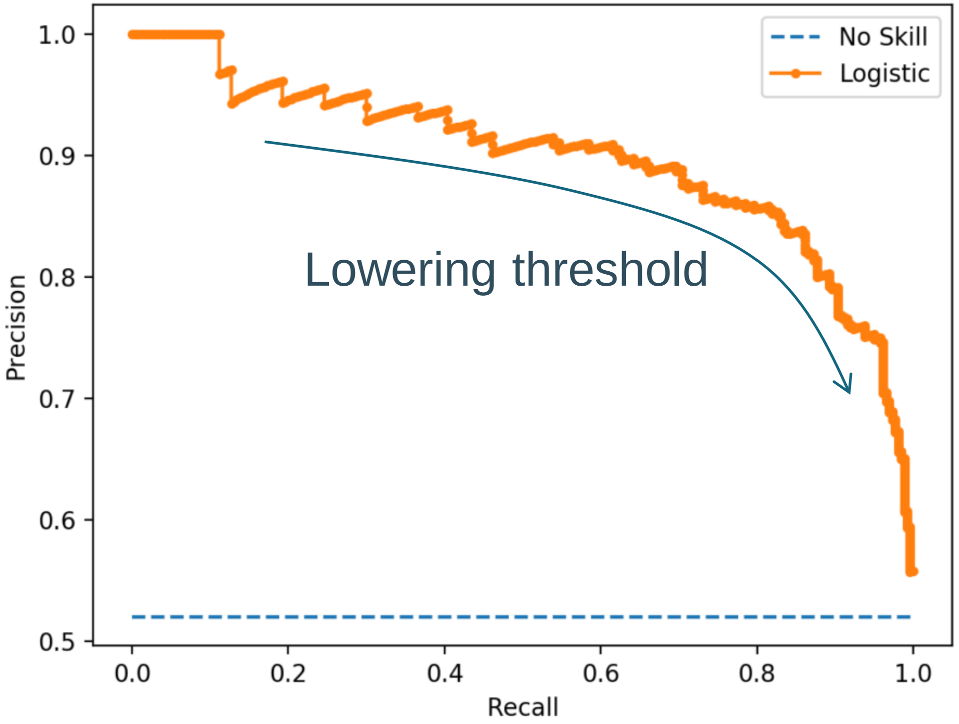
- F1 score = 2 * ((Precision * Recall) / (Precision + Recall)) = 2 * ((0.882 * 0.938) / (0.882 + 0.938)) = 0.909





Precision-Recall Curves

- Show the trade-off between precision and recall/sensitivity as the threshold of detection is varied.



Lowering threshold

Threshold example:

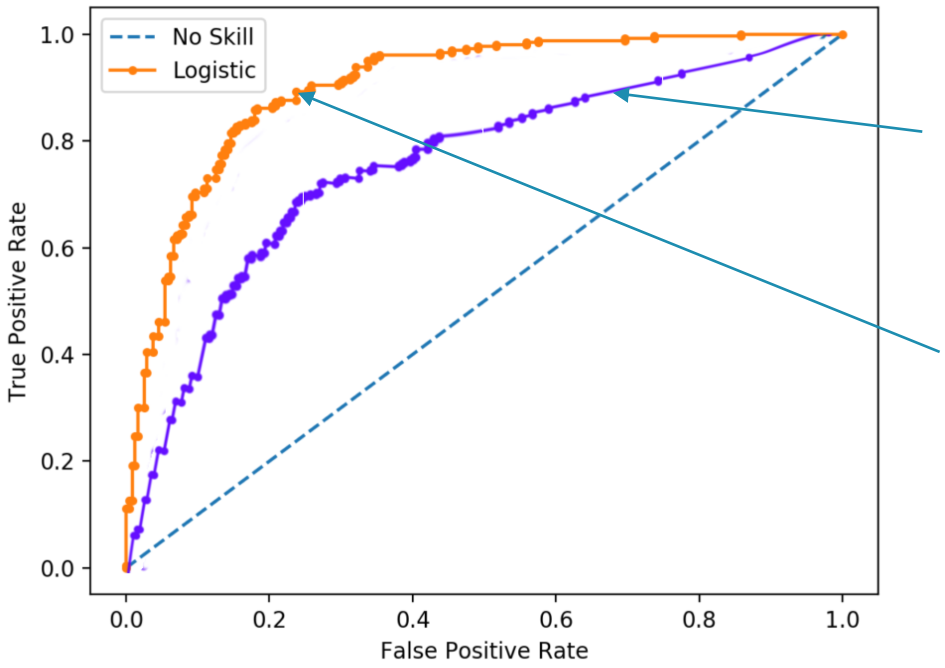
Faulty if RMS vibration above 8G vs 9G vs 10G.

Jason Brownlee, ROC Curves and Precision-Recall Curves for Imbalanced Classification, Machine Learning Mastery, Available from https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/, accessed May 3rd, 2020.

Receiver operating curves (ROC)

- Trade-off with varying threshold between probability of detection and probability of false alarm.

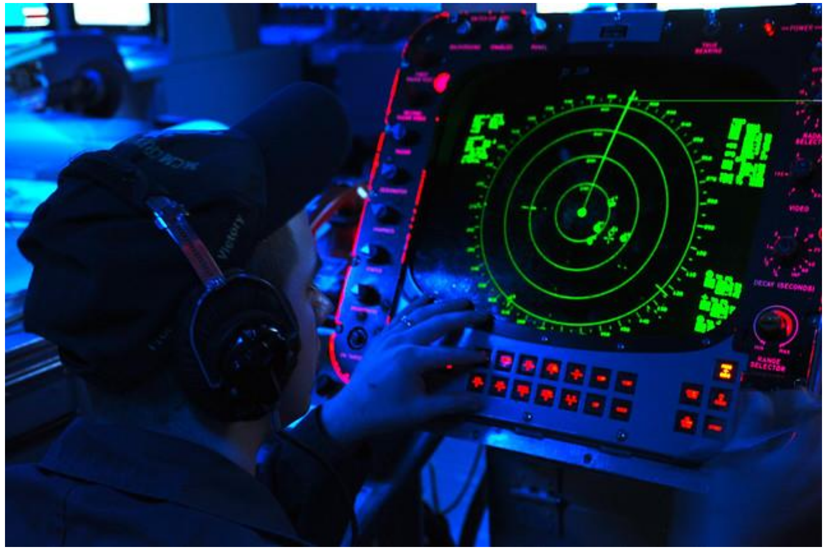
Probability of detection



Probability of false alarm

This guy

The over-achiever in the class

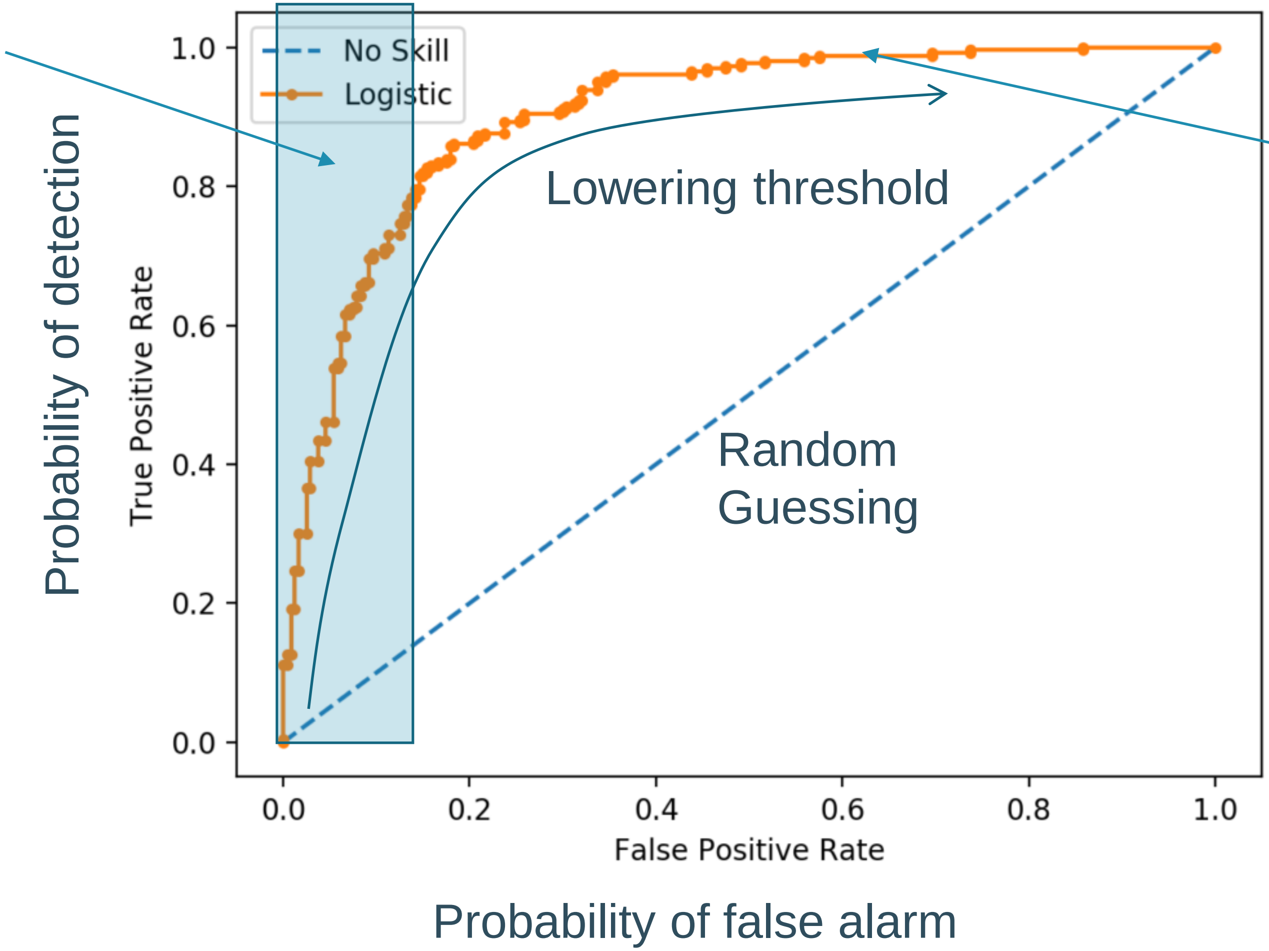


This Photo by Unknown Author is licensed under CC BY

Developed during WW2 to evaluate radar operators.
Good for comparing models.

Receiver operating curves (ROC)

We mostly care about this region for fault detection



We can compute the area under this curve as summary metric for how well a model is doing over all thresholds.

Test set should be balanced when using ROC!

Discussion of case studies from the audience: Questions to ask when evaluating a model

- Is my hold out test data representative of going out to collect new data?
- Did the test data in any way had a say in the model parameters?
- Have you modified the model so many times, or tried so many approaches, on this same data set that you (the human) are overfitting it?

Acknowledgement

Slides in part based on:

- Menéndez González V. Evaluating Machine Learning Models [version 1; not peer reviewed]. F1000Research 2020, 9:329 (slides) (https://doi.org/10.7490/f1000research.1117855.1)
- https://pages.cs.wisc.edu/~dpage/cs760/evaluating.pdf

Images:

- https://purnasaigudikandula.medium.com/a-beginner-intro-to-neural-networks-543267bda3c8
- https://duet-cdn.vox-cdn.com/thumbor/0x0:554x302/640x427/filters:focal(277x151:278x152):no_upscale():format(webp)/cdn.vox-cdn.com/uploads/chorus_asset/file/9494627/Bezos_Smash.gif
- https://tenor.com/view/explosion-windturbine-strongwinds-destruction-nature-gif-3569843
- https://www.youtube.com/watch?v=9RrgOIZZLIM
- https://www.mikulskibartosz.name/assets/images/2019-02-04-f1-score-explained/f05_score.png

The authors gratefully acknowledge the European Commission for its support of the Marie Sklodowska Curie program through the ETN MOIRA project (GA 955681).

bert.pluymers@kuleuven.be

www.mech.kuleuven.be/lmsd

VACANCIES www.mech.kuleuven.be/lmsd-joboffers

lmsd_kuleuven

KU Leuven Mecha(tro)nic System Dynamics (LMSD)

lmsd.kuleuven

lmsd-kuleuven

lmsd_kuleuven

KU Leuven Mecha(tro)nic System Dynamics (LMSD)