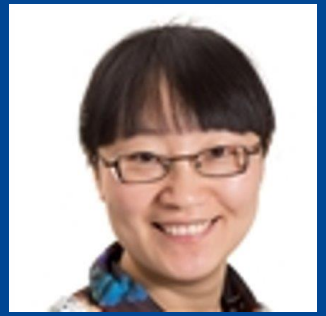
NTNU | Norwegian University of Science and Technology

Evaluation Methods for Machine Learning

Course - TDT4173 - Machine Learning - Fall 2023

Liyuan Xing
Senior Machine Learning Engineer in ANEO
Adjunct Associate Professor at NAIL/IDI




Left or Right

Outline

Introduction

- Why evaluation methods are needed
- How evaluation methods are implemented

Model evaluation techniques

- Holdout
- Cross validation (CV)

Model evaluation metrics

- Classification
- Regression

A typical machine learning process

Representation

- Data structure about variables/features and forecasts

Train model

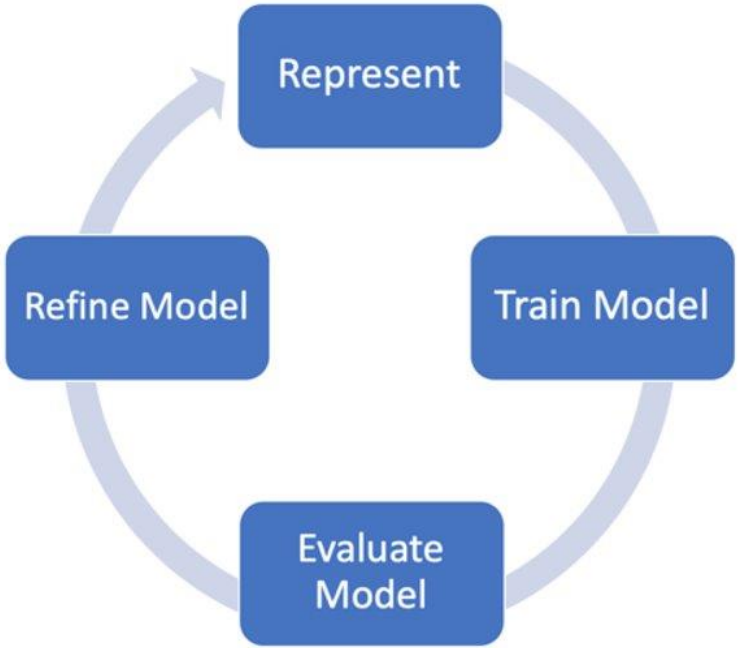
- Using a range of models with different techniques

Evaluate model

- Decides if model is working fine on unseen data
- Decides if model is built heading towards the right path

Refine model

- Further optimizing the selected model and then repeat this iterative cycle



How to evaluate models?

Get an unbiased estimate of a learned model in terms of metrics

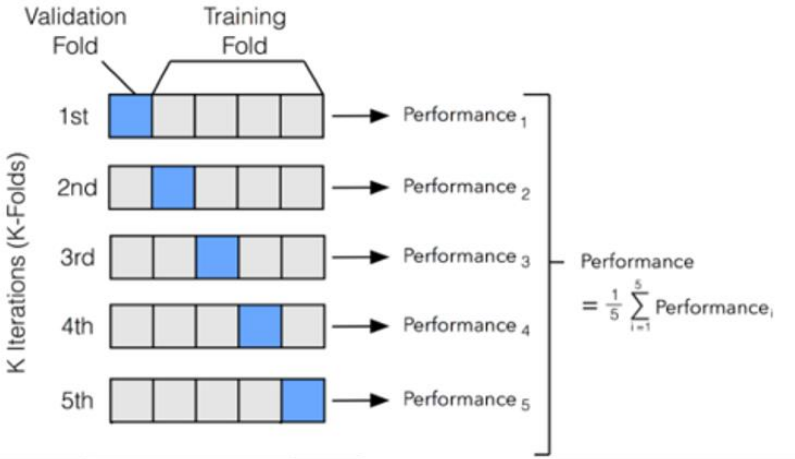
- On future (unseen/out-of-sample) test set, NOT on training set

Model evaluation techniques for splitting data sets and addressing biased issue

- Holdout (random resampling, stratified sampling)
- Cross validation (K-Fold CV, Stratified CV, Time Series CV, Nested CV)

Model evaluation metrics for qualifying performance

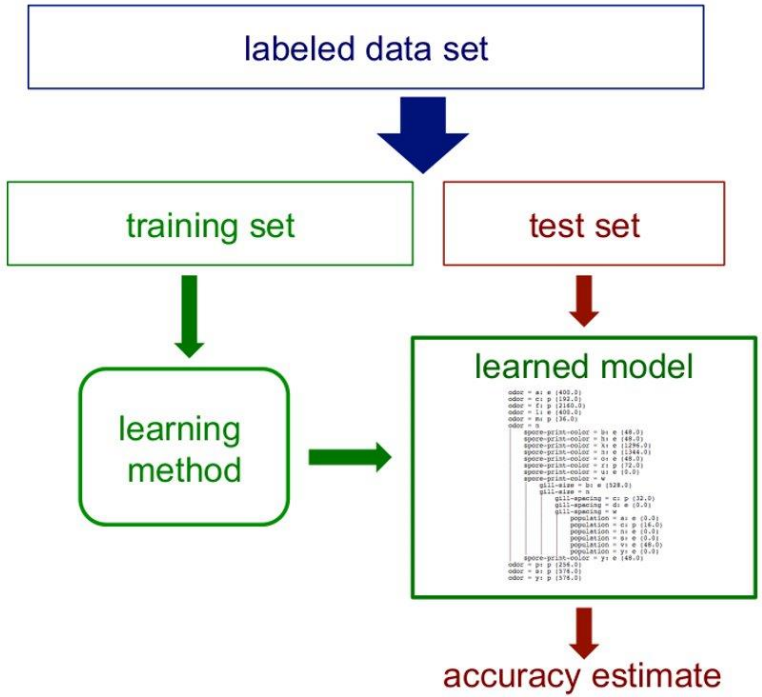
- Supervised models
  - Classification (Precision, Recall, ...)
  - Regression (MAE, RMSE, ...)
- Unsupervised models (Rand index, Mutual Information, ...)



Holdout – A single training/test partition

Testing a model on different data than it was trained on provides an unbiased estimation of learning performance

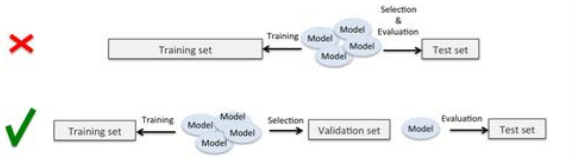
- Training set is a subset of the dataset used to build predictive models
- Test set or unseen data is a subset of the dataset used to assess the likely future performance of a model
- If a model fits to the training set much better than it fits the test set, overfitting is probably the cause

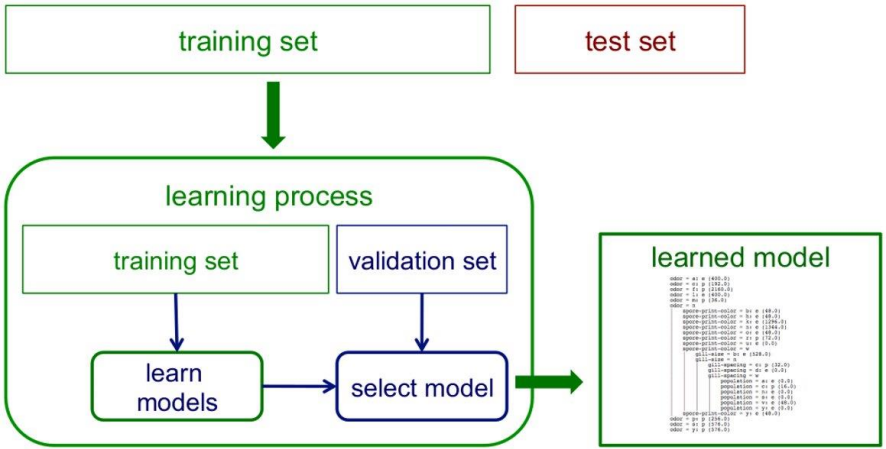


Holdout – With validation set for tuning

Validation set

- is a subset of the dataset used to assess the performance of the model built in the training phase
- is for fine-tuning a model's parameters and selecting the best performing model
  - e.g. to choose the best level of decision-tree pruning
- not all modeling algorithms need a validation set
  - if there is no hyper parameter need to be chosen





Pros and cons of a single partition holdout

Speed, simplicity, flexibility

Often associated with high variability

- differences in the training and test dataset can result in meaningful differences in the estimate of accuracy

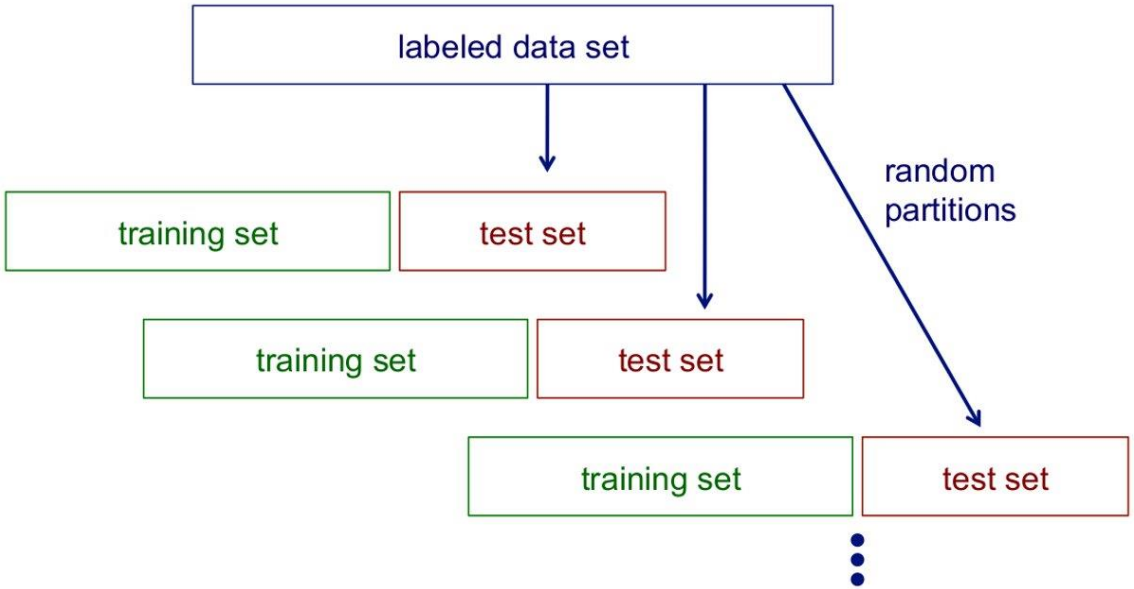
A single training set doesn't tell how sensitive accuracy is to a particular training sample

May not have enough data to make sufficiently large training and test sets

- a larger test set gives us more reliable estimate of accuracy (i.e. a lower variance estimate)
- but… a larger training set will be more representative of how much data we actually have for learning process

Random resampling - multiple partitions

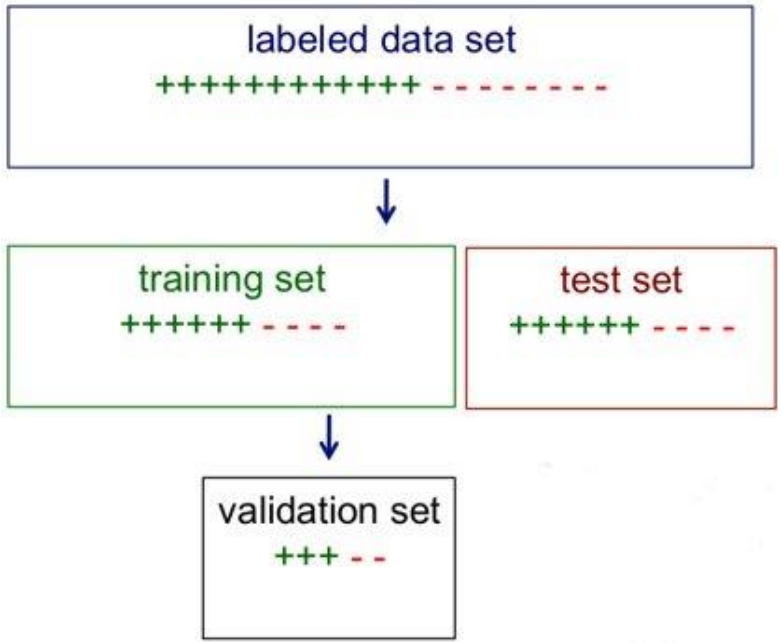
Repeatedly randomly partitioning the available data into multiple training and test sets -- can tell more about the sensitivity



Stratified sampling

When randomly selecting training or validation sets, we may want to ensure that class proportions are maintained in each selected set

This can be done via stratified sampling: first stratify instances by class, then randomly select instances from each class proportionally



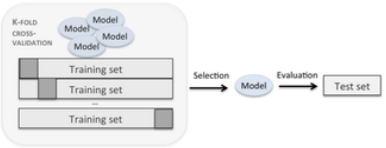
Cross validation – Multiple partitions

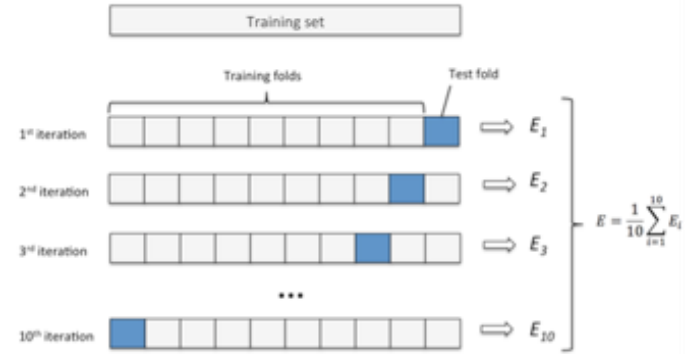
K-fold CV: most common cross validation technique

- Original dataset is partitioned into K equal size subsamples, called folds
- Iteratively leave one fold out for the test set, train on the rest, repeated K times
- The error estimation is averaged over all K trials to get the total effectiveness of our model

K is a user-specified number

- 5-fold or 10-fold cross validation is preferred
- Smaller values of K are often used when learning takes a lot of time





Pros of cross validation

- every observation from the original dataset has the chance of appearing in training and test set

Significantly reduces bias and variance, and add effectiveness of model

- The higher value of K leads to less biased model

Able to detect overfitting which fails to generalize a pattern

Makes efficient use of the available data for testing

Various extensions of CV for addressing different issues

- Stratified CV: uses stratified sampling when partitioning the data
- Nested CV: use cross validation within a training set to select a model
- Time series CV: adapts normal CV for time series data

Nested CV: Almost unbiased estimate of true error

Outer loop for error estimation

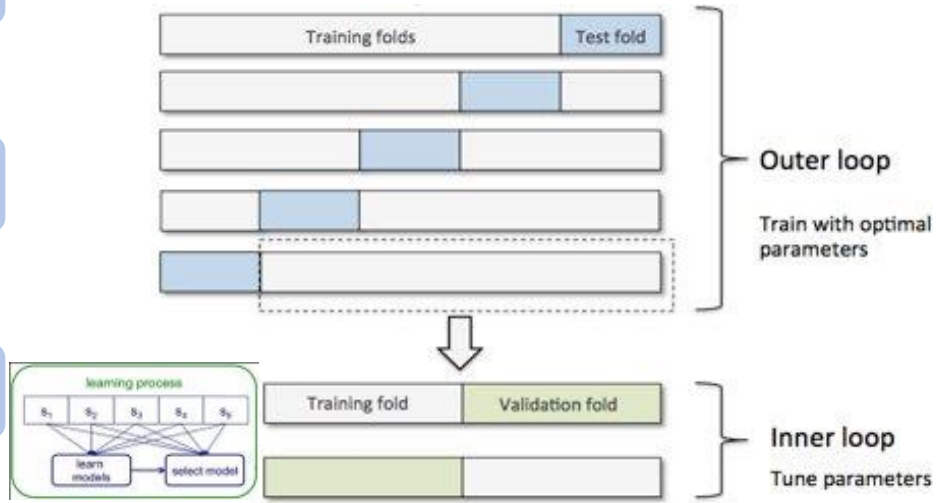
- Outer K-fold CV loop to split the data into training and test folds

Inner loop for parameter tunning

- Inner loop is used to select the model via K-fold cross validation on the training fold

Connection between inner and outer loop

- For a given training set in outer loop, the optimal parameters tuned from its inner loop are used to train the model using the given entire training set



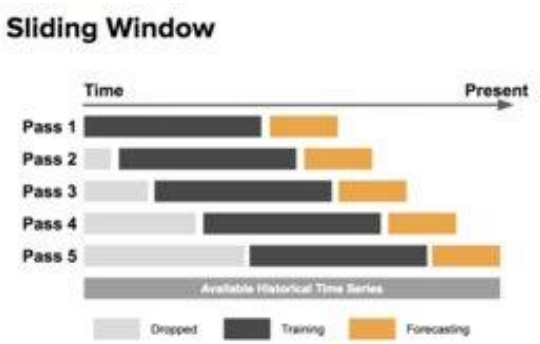
(Nested) CV for time series

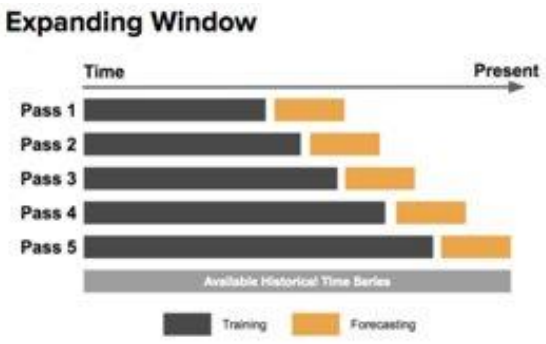
Noraml CV (like K-fold) should not be used for two reasons

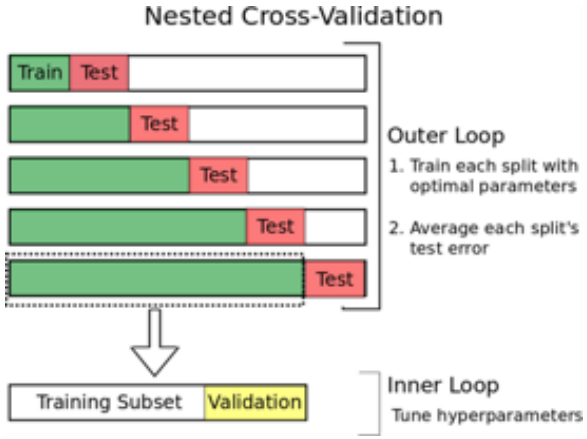
- Time series has temporal dependencies
- Arbitrary choice of test set by normal CV results in data leakage

Adaption

- Train on a set of data that is older than the test data
- Use expanding window when there is limited time series data







Statistical significance tests

Is the difference in mean scores statistically significant?

- between two machine learning models or hyper-parameter sets

Statistical significance tests

☹
- Paired Student's T-test for K-fold cross validation
  - violates the independence assumption, since a given observation is used in the training dataset (K-1) times

☺
- Modified paired Student's T-test with (K/2)×2 cross validation
  - K/2 repeats of 2-fold cross validation, ensuring that each observation appears only in the train or test dataset for a single estimate of model

☺
- McNemar's test for single-run classification evaluation results
  - determine whether the difference in observed proportions in the algorithm's contingency table are significantly different from the expected proportions

Paried Student's T-Test

t test: how significant the differences between group means are

- t score: ratio between the difference between two groups and the difference within the groups
  - larger t scores = more difference between groups
  - ± indicates the direction, can be ingnored
- p value: the probability that the results from your sample data occurred by chance.
  - a p-value of 0.05 means there is only a 5% probability that the results from an experiment happened by chance
  - low p-values indicate your data did not occur by chance
  - find the p-value in the t-table, using the degrees of freedom https://www.statisticshowto.com/tables/t-distribution-table/
- compare your t-table value (2.228) to your calculated t-value (-2.74)
  - the p-value is less than the alpha level: $p<0.05$
  - the calculated t-value is greater than the table value at an alpha level of 0.05
  - So reject the null hypothesis that there is no difference between means

Subject #	Score 1	Score 2	X-Y	(X-Y)^2
1	3	20	-17	289
2	3	13	-10	100
3	3	13	-10	100
4	12	20	-8	64
5	15	29	-14	196
6	16	32	-16	256
7	17	23	-6	36
8	19	20	-1	1
9	23	25	-2	4
10	24	15	9	81
11	32	30	2	4
		SUM:	**-73**	**1131**

$$t = \frac{(\sum D)/N}{\sqrt{\frac{\sum D^2 - \left(\frac{(\sum D)^2}{N}\right)}{(N-1)(N)}}} = \frac{-73/11}{\sqrt{\frac{1131 - \left(\frac{(-73)^2}{11}\right)}{(11-1)(11)}}} = -2.74$$

Two Tails T Distribution Table

df	a = 0.2	0.10	0.05
∞	t_a = 1.282	1.645	1.960
1	3.078	6.314	12.706
2	1.886	2.920	4.303
3	1.638	2.353	3.182
4	1.533	2.132	2.776
5	1.476	2.015	2.571
6	1.440	1.943	2.447
7	1.415	1.895	2.365
8	1.397	1.860	2.306
9	1.383	1.833	2.262
10	1.372	1.812	2.228
11	1.363	1.796	2.201

McNemar T-Test

Reports on the different correct or incorrect predictions between the two models, not the accuracy or error rates

- whether the two models disagree in the same way (or not)

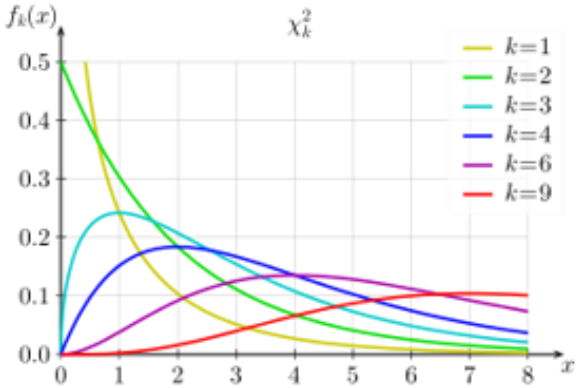
Calculating the Test from contingency table: a tabulation or count of two categorical variables

- calculate the statistic `statistic = (Yes/No - No/Yes)^2 / (Yes/No + No/Yes)`
- get the p value from the test statistic which has a Chi-Squared distribution with 1 degree of freedom
- compare the p value to the given significance level alpha
  - p value > alpha: fail to reject H0, classifiers have a similar proportion of errors on the test set
  - p value <= alpha: reject H0, classifiers have a different proportion of errors on the test set

Instance,	Classifier1 Correct,	Classifier2 Correct
1	Yes	No
2	No	No
3	No	Yes
4	No	No
5	Yes	Yes
6	Yes	Yes
7	Yes	Yes
8	No	No
9	Yes	No
10	Yes	Yes

	Classifier2 Correct,	Classifier2 Incorrect
Classifier1 Correct	Yes/Yes	Yes/No
Classifier1 Incorrect	No/Yes	No/No

	Classifier2 Correct,	Classifier2 Incorrect
Classifier1 Correct	4	2
Classifier1 Incorrect	1	3



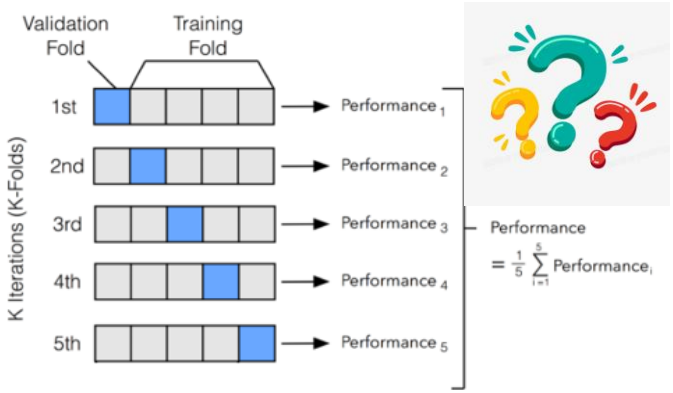
What's an evaluation metric?

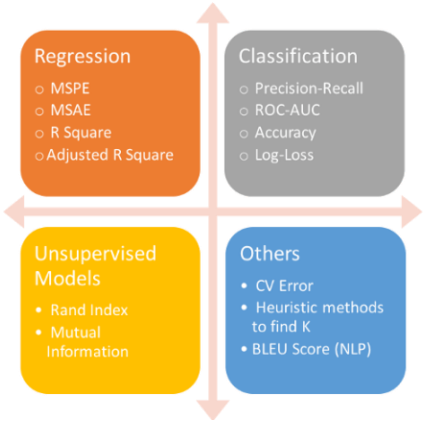
A way to qualify of a machine learning model

- Evaluation metric ≠ Loss function
  - Validation/Test sets vs Training set
  - Measure performance of a learned model vs Optimize to build a model

How to choose evaluation metrics?

- Data itself
  - imbalanced vs balanced
- A given machine learning task
  - supervised classification or regression vs unsupervised model
- Business problem
  - can tolerance more false negative or false positive





Supervised metric types

Classification metrics

- Binary classification
  - Accuracy, Precision-Recall, F1 score, Specificity-Sensitivity, PR/AUC, ROC/AUC, Log loss
- Multi-task classification
  - Precision: Micro, macro, weighted
  - Multi-class log loss

Regression metrics

- Point estimation
  - MAE, RMSE, MAPE, sMAPE, MASE, R^2
- Probability estimation
  - Reliability: statistical tests
  - Sharpness: pinball loss, CRPS

Binary classification: Confusion matrix

True Positive (TP)

- Actual true and model predicted it true

True Negative (TN)

- Actual false and model predicted it false

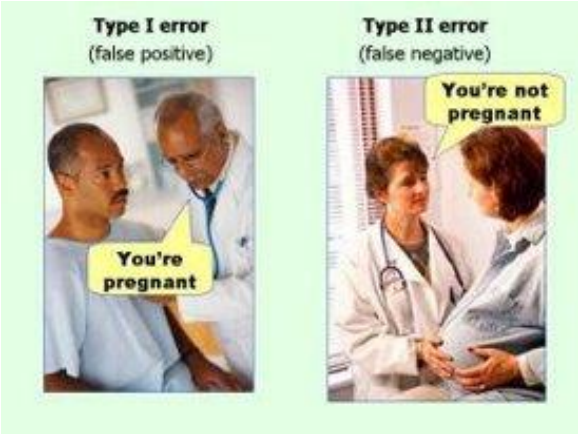
False Positive (FP) or Type I error

- Actual false and model predicted it true

False Negative (FN) or Type II error

- Actual true and model predicted it false

		PREDICTED	
		0 (Negative)	1 (Positive)
ACTUAL	0 (Negative)	TN	FP
	1 (Positive)	FN	TP



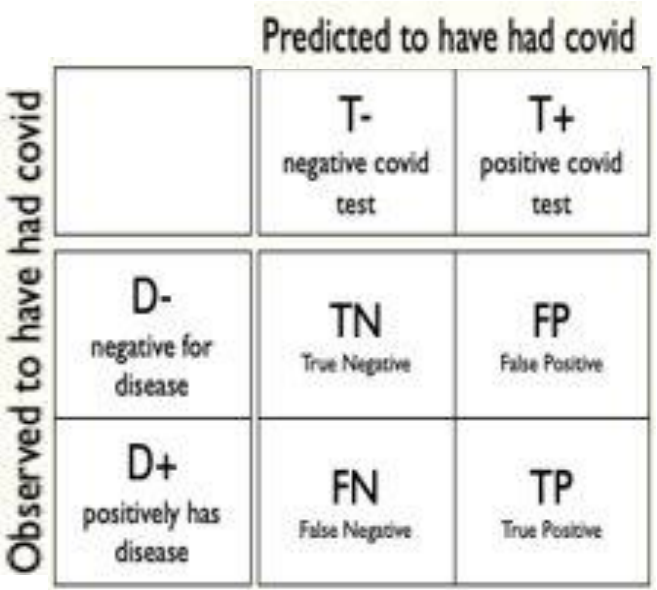
Binary classification metrics

Number of items correctly identified as positive out of total true positives	Recall Sensitivity True positive rate (TPR)	$\frac{TP}{FN+TP} = \frac{TP}{P}$
Number of items wrongly identified as positive out of total true negatives	False positive rate (FPR) False alarm rate	$\frac{FP}{TN+FP} = \frac{FP}{N}$
Number of items correctly identified as negative out of total true negatives	Specificity True negative rate (TNR)	$\frac{TN}{TN+FP} = \frac{TN}{N} = 1 - FPR$
Number of items correctly identified as positive out of total items identified as positive	Precision	$\frac{TP}{TP+FP}$
Number of items wrongly identified as negative out of total true positives	False negative rate (FNR)	$\frac{FN}{FN+TP} = \frac{FN}{P}$
Number of items correctly identified out of total	Accuracy	$\frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN}$

Accuracy

☹ Accuracy may not be useful measure in cases where

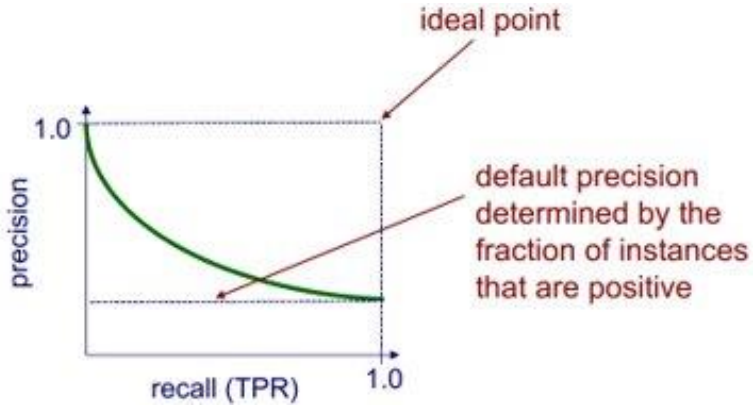
- there is a large class skew
  - Is 98% accuracy good if 97% of the instances are negative? --> No, a dummy model has 97% accuracy
- there are differential misclassification costs – say, getting a positive wrong costs more than getting a negative wrong
  - Consider a medical domain in which a false positive results in an extraneous test/cost but a false negative results in a failure to treat a disease
- we are most interested in a subset of high-confidence predictions
  - Consider a medical domain how about if we do not care about the costs, but do not want to miss a patient who is having disease



Precision or Recall?

What do you care about?

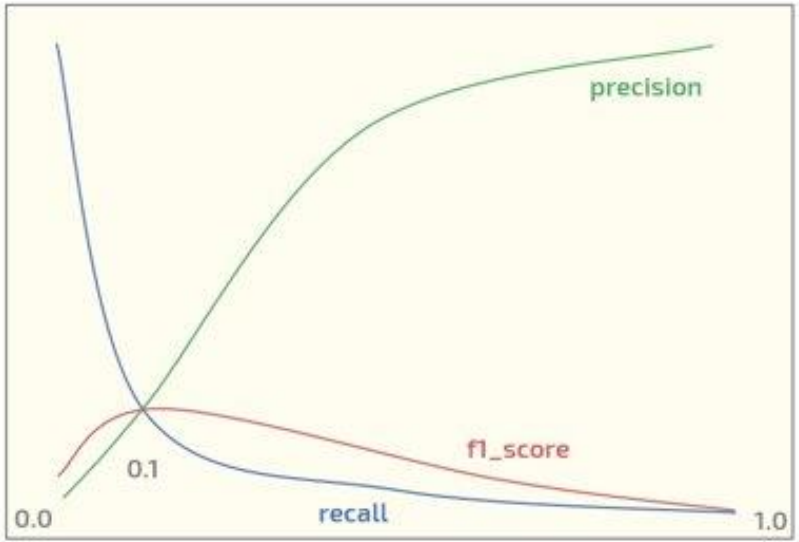
- Low FP and high Precision
  - Don't want to let a person does not have disease to be detected as having disease, which results in extraneous test/cost
- Low FN and high Recall
  - Don’t want to miss on a patient who is having disease but goes undetected
- Ideal point: high precision and high recall
  - One increases the other one goes down



F1 score

$$\text{F1 score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}}$$

- Harmonic mean of Recall and Precision, taking the contribution of both
- Is sensitive to which class is positive, or which class is negative
- Due to the product in the numerator if one goes low, the final F1 score goes down significantly
- ☹ One drawback is that both precision and recall are given equal importance



Precision/Recall (PR) curves

A curve between precision and recall for various threshold values

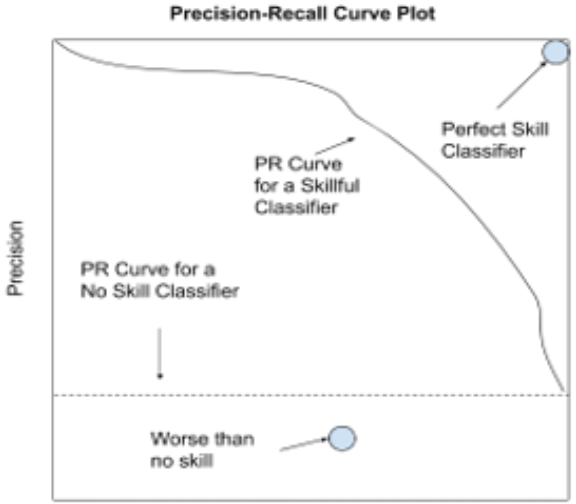
- Different thresholds are set to cutoff the probability of an instance being positive to first have binary classification, then derive values for precision and recall

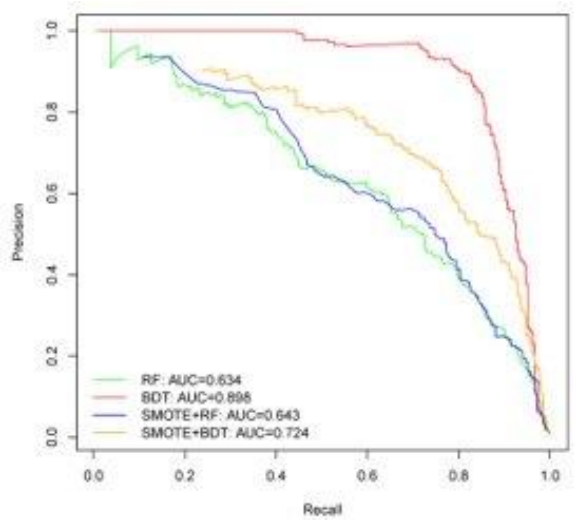
Skillful or skillless classifier?

- Perfect skill classifier is at top right
- Default precision of no skill classifier is determined by the fraction of instance that are positive

PR AUC: area under curve

- The higher its numerical value the better





ROC curves

A curve between TPR and FPR for various threshold values

Skillful or skillless classifier?

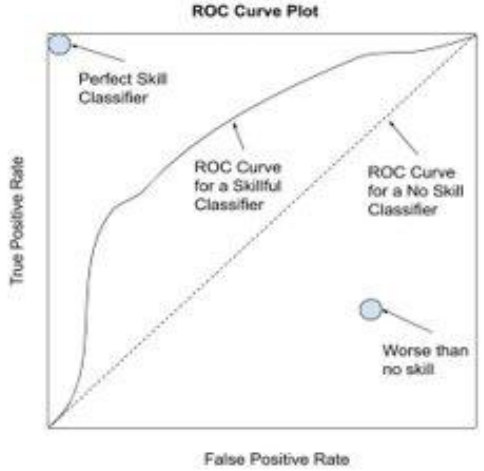
- Perfect skill classifier is at top left
- No skill classifier is a random guessing at diagonal line

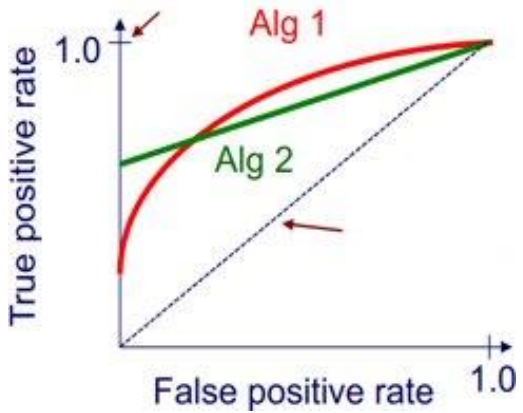
Different models can work better in different parts of ROC

- This depends on cost of false positive vs. false negative

ROC AUC: area under curve

- The higher its numerical value the better





PR vs ROC curve

PR curves

- well suited for imbalanced classes with lots of negative instances - due to the absence of TN in the PR equation
- show the fraction of predictions that are false positives

ROC curves

- are useful when both the classes are important to us - due to the consideration of TN in the ROC equation
- insensitive to changes in class distribution (ROC curve does not change if the proportion of positive and negative instances in the test set are varied)
- can identify optimal classification thresholds for tasks with differential misclassification costs

Log loss

$$LogLoss = -\frac{1}{n}\sum_{i=0}^{n}[y_i log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Another way of taking into account uncertainty of model predictions - probability

Larger penalty for confident false predictions

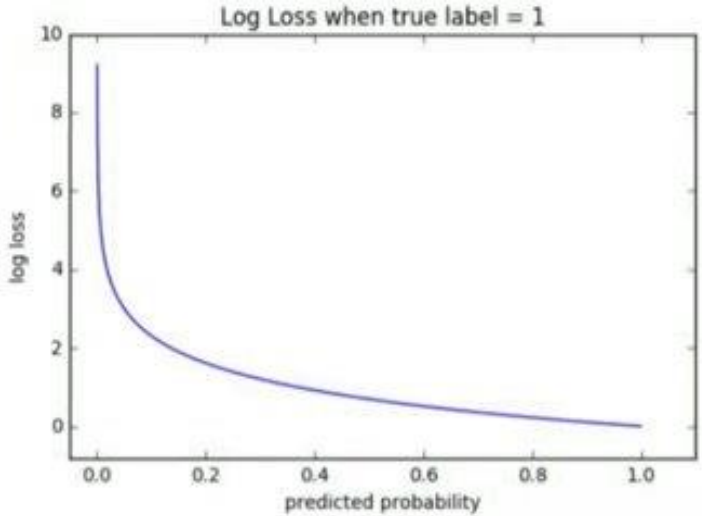
The smaller the better, with a perfect model having a log loss of 0

Where;

- N is the number of test images
- $\hat{y}_i$ is the predicted probability of the image being a dog
- y_i is 1 if the image is a dog, 0 if it's a cat
- log() is the natural (base e) logarithm

A smaller loss is better

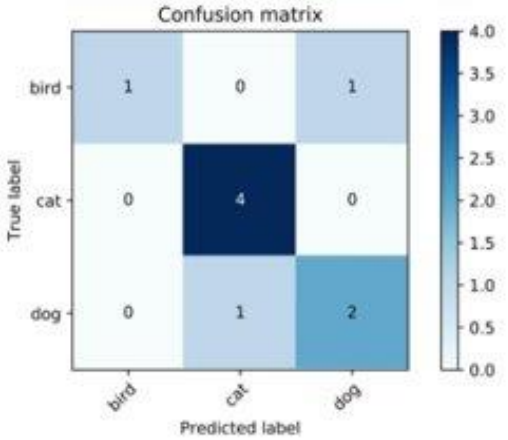
True label	Predicted prob. of class 1	Log loss
1	0.9	0.105360515657
1	0.55	0.597837000755
1	0.10	2.302585092994
0	0.95	2.995732273553



Multi-class metrics

Precision, Recall, F1 score

- Micro-averaged: all samples equally contribute to the average
- Macro-averaged: all classes equally contribute to the average
  - is preferable if there is a class imbalance problem
- Weighted-average: each class's contribution to the average is weighted by it's size



	TP	FP	Pr.	N samples
bird	1	0	1	2
cat	4	1	0.8	4
dog	2	1	0.6666	3
TOTAL	7	2		

$$\text{micro pr} = \frac{7}{7+2} = 0.7777$$

$$\text{macro pr} = \frac{1}{3}(1 + 0.8 + 0.6666) = 0.8222$$

$$\text{weighted pr} = \frac{1*2 + 0.8*4 + 0.6666*3}{2+4+3} = 0.8$$

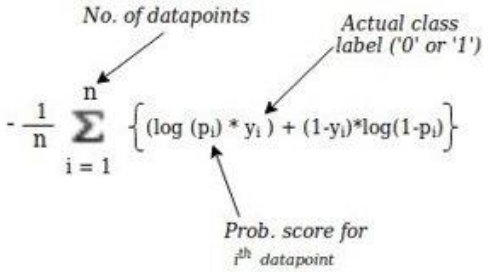
Multi-class metrics (2)

Multi-class log loss

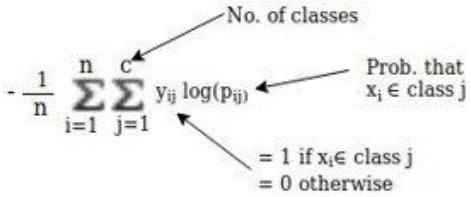
- It is no different than log loss for the binary problem
- The intuition is exactly the same

Pros and cons of log loss

- Useful to compare model with similar performance values such as accuracy, precision, recall, F1 score etc.
- Less interpretable as its value starts from 0 and have no upper bound
  - Log loss 0.25 is good or not
  - Set a base model as benchmark to visualize performance of a model on the basis of its log loss score

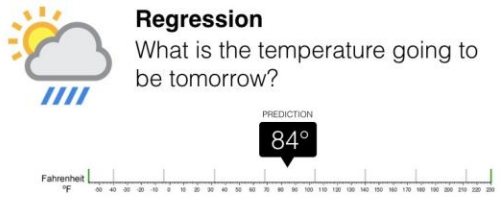


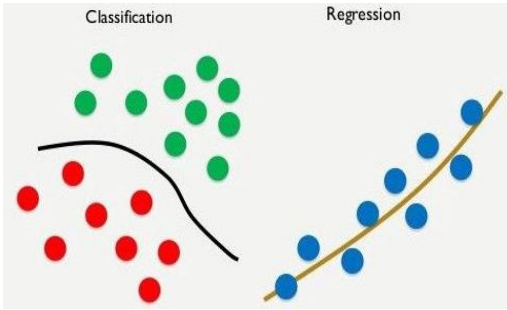
(a) log-loss formulae for binary classification



(a) log-loss formulae for multi-class classification

Regression metrics – Point estimation



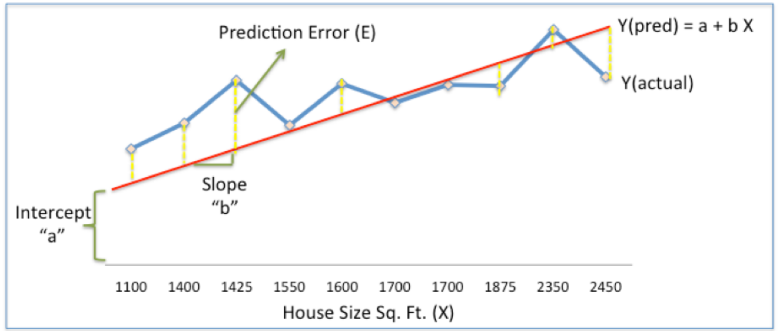


Qualify performance of a regressor, not a classifier

- Continuous value, instead of discrete classification label
- Do not distinguish error types, such as false negative and false positive in classification

A prediction error is the difference between an observed value and its prediction

- Error does not mean a mistake, it means the unpredictable part of an observation
- Residual error (Training set) ≠ Prediction error (Test set)
- Categories
  - Scale-dependent error
  - Percentage error
  - Scaled error



Scale-dependent metrics: MAE, RMSE

Both

- The difference between an observed value and its predictions $e_{T+h} = y_{T+h} - \hat{y}_{T+h|T}$
- On the same unit as y values
- Cannot be used to make comparisons between series that involve different units
- The smaller the better

Mean absolute error: MAE $\text{MAE} = \text{mean}(|e_t|)$

- Popular as it is easy to both understand and compute
- Robust to outliers, since error is not squared
- Loss function leads to predictions of the median

Root mean squared error: RMSE $\text{RMSE} = \sqrt{\text{mean}(e_t^2)}$

- Widely used, despite being more difficult to interpret
- Give a relative high weight to large errors, since error is squared before average
- Loss function leads to predictions of the mean

Metrics for percentage error: MAPE, sMAPE

Both

- Scale-dependent error divided by an observation value $p_t = 100e_t/y_t$
- Unit free, and frequently used to compare regression performances between data sets
- The smaller the better
- Infinite or undefined
- Assume the unit of measurement has a meaningful zero

Mean absolute percentage error: MAPE $\text{MAPE} = \text{mean}(|p_t|)$

- Put a heavier penalty on negative errors than on positive errors

Symmetric MAPE: sMAPE $\text{sMAPE} = \text{mean}\left(200|y_t - \hat{y}_t|/(y_t + \hat{y}_t)\right)$

- Can be negative, not really a metric of 'absolute percentage errors' at all

Metrics for scale errors: MASE, R^2

Both

- Unit free, since scaled on either training set or test set
- Relative to another simple regression method

Mean absolute scaled error: MASE $\mathrm{MASE} = \mathrm{mean}(|q_j|)$

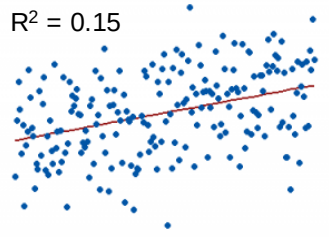
- Scaling the errors based on the training MAE from a naïve forecast
- MASE<1: a better forecast than the average naïve forecast
- MASE>1: a worse forecast than the average naïve forecast

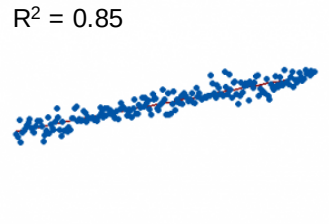
$$q_j = \frac{e_j}{\frac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}|}$$

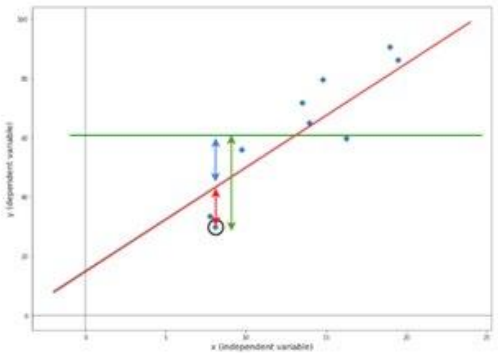
R^2 (coefficient of determination) $R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$

- Scaling the errors based on the total variance of the true values in test set
  - Can be explained as sum of square error from an average model (green line)
- Indicates how well a regression model (red line) fits the true values (blue points)
- 1 = perfect fit to the true values and explains all of the variations
- 0 = Dummy regressor predicting average and does not explain any of the variation
- Does not indicate if a regression model provides an adequate fit to your data
- A good model can have a low R^2 value, and a biased model can have a high R^2 value
- Trust R^2 only if your model is unbiased, which can be checked by plotting the prediction error
  - Unbiased means that the predicted values are not systematically too high or too low anywhere in the observation space, and randomly scatted around zero
- Adjust R^2 and predicted R^2
- Address particular problems with R^2

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







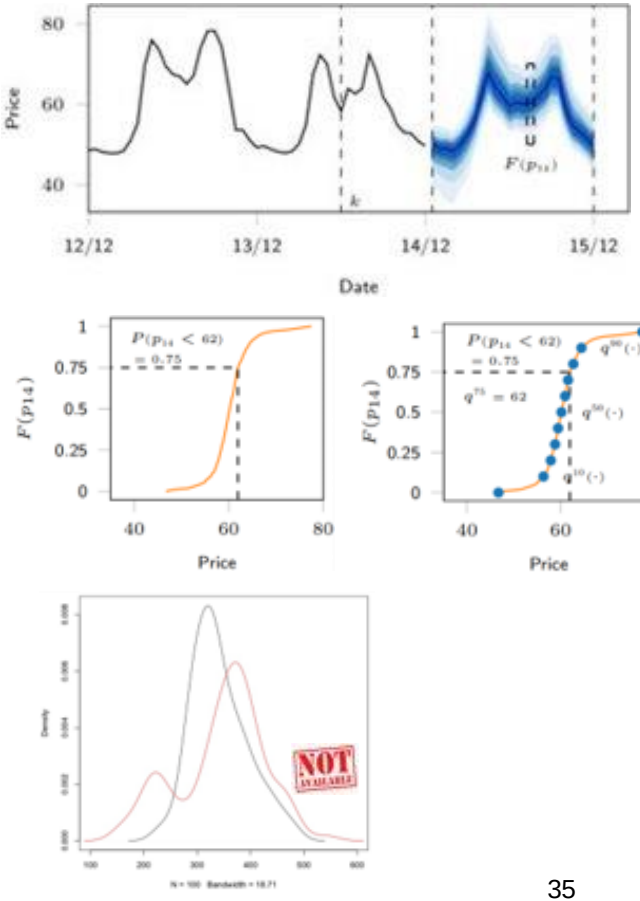
Regression metrics - Probability estimation

Qualify the performance by comparing two distributions

- True distribution: missing, can not be observed
  - Only the past observations
- Predictive distribution: from regression
  - Often not cumulative distribution but prediction interval, quantiles

How to evaluate probabilistic forecasts?

- Reliability: statistical consistency between distributional forecasts and observations
  - Joint property of the predictions and the observations
  - Formal statistical tests: Kupiec test, Christoffersen test, Berkowitz test
- Sharpness: how tightly the predictive distribution covers the actual one, i.e., to the concentration of the predictive distributions
  - Property of the predictions only
  - A-single-number metrics: pinball loss, winkler score, CRPS(continuous rank probability score)



Pinball loss

Closely related to the concept of proper scoring rules

- assigning a numerical score, based on the predictive distribution, and on the actual observation
- quoting the true distribution as the forecast distribution is an optimal strategy in expectation, i.e., it minimizes the score

pinball loss == quantile loss $(\tau - 1)\sum_{y_i<q}(y_i - q) + \tau\sum_{y_i\geq q}(y_i - q)$

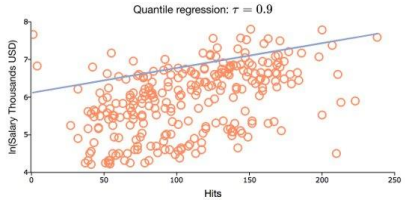
- Evaluation metric for measuring performance of a quantile regression in test set
- Loss function to be minimized in quantile regression to train model in training set

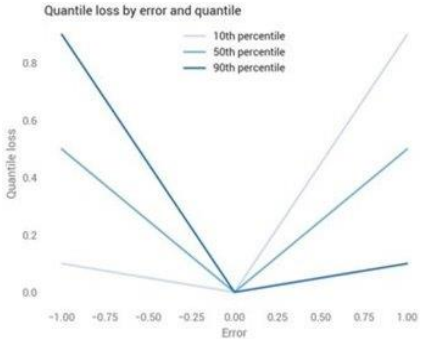
A measure of fit for one quantile

- Averaged across different quantiles and whole test set to provide an aggregate score
- The closer the desired quantile gets to 100%, the more the pinball loss penalizes predictions which are less than the true value, meaning assign more of a weight to negative error than to positive error
- A lower score indicates a better prediction

$$L(q_a, y) = \begin{cases} \left(1 - \frac{a}{100}\right)(q_a - y), & \text{if } y < q_a \\ \frac{a}{100}(y - q_a), & \text{if } y \geq q_a, \end{cases}$$

where *y* is the observation used for forecast evaluation, q_a is a quantile forecast with *a/100* as the target quantile





Regression metrics in competitions

Point estimation

- GEFcom 2012
  - Root mean squared error (RMSE) for wind power forecasts
  - Weighted RMSE (WRMSE) for hierarchical load data
    - Different weights for system, zonal levels, as well as for forecasted and backcasted weeks
- M4 2018 (100000 time series with different units)
  - Symmetric mean absolute percentage error (sMAPE)
  - Mean absolute scaled error (MASE)
- M5 – Accuracy 2020
  - Weighted Root Mean Squared Scaled Error (WRMSSE) for hierarchical retail good data

Probabilistic estimation

- GEFcom 2014
  - Pinball loss for energy forecasts
    - Averaged across 99 quantiles and 24 hours of the target day
- M5 – Uncertainty 2020
  - Weighted Scaled Pinball Loss (WSPL) for hierarchical retail good data
    - Weights of each series are different
    - Weights for all hierarchical levels are equal
    - Same calculation for weights in WRMSSE
- M6 – Forecasts performance
  - Ranked Probability Score (RPS)

Summary

There is no 'one fits all' evaluation techniques or metrics

Get to know our data

- How much data (limited amount or near unlimted)? How many outliers? Imbalanced data or not? What types of data (time series, hierarchical data, across data sets)?

Keep in mind business objective of your machine learning tasks

- Can tolerance more false negative or false positive?

More specific tasks for different types of data and business objectives

- Segmentation of images, machine translation, case-based reasoning, etc
- Require customized evaluation techniques and metrics

Reading list

Book: FORECASTING: Principles and Practice

- Chapter 5.8 - Evaluating forecast accuracy, Accessed October 4, 2023. https://otexts.org/fpp3/

Video

- 1. "Machine Learning Fundamentals: Cross Validation." Accessed September 9, 2020. https://www.youtube.com/watch?v=fSytzGwwBVw
- 2. "Machine Learning Model Evaluation Metrics." Accessed September 9, 2020. https://www.youtube.com/watch?v=wpQiEHYkBys or post https://www.mariakhalusova.com/posts/

Competitions related materials (only need pay attention to the evaluation metrics)

- 1. Section 4: Jakub, Nowotarski, and Rafał Weron, "Recent advances in electricity price forecasting: A review of probabilistic forecasting", Renewable and Sustainable Energy Reviews 81 (2018) 1548–1568
- 2. Section 2.1 and 3.1: Hong, Tao, Pierre Pinson, and Shu Fan, "Global Energy Forecasting Competition 2012," International Journal of Forecasting 30, no. 2 (2014): 357–363.
- 3. Section 3.2: Hong, Tao, Pierre Pinson, Shu Fan, Hamidreza Zareipour, Alberto Troccoli, and Rob J. Hyndman, "Probabilistic energy forecasting - Global Energy Forecasting Competition 2014 and beyond", International Journal of Forecasting 32, no. 3 (2016): 896–913.
- 4. Section 3.2: Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos, "The M4 Competition: 100,000 time series and 61 forecasting methods", International Journal of Forecasting 36, no. 1 (2020): 54–74.
- 5. Section Evaluation: "The M5 Competition – Competitors' guild." Accessed October 4, 2023. https://mofc.unic.ac.cy/m5-competition/
- 6. Section Measuring the performance of the forecasts: "The M6 Financial Duathlon Competition", https://mofc.unic.ac.cy/wp-content/uploads/2022/02/M6-Guidelines-1.pdf

Reading list - links

- Wadhawan, Neha , "Machine Learning Model Evaluation Methods: which one to use? Accessed October 4, 2023. https://medium.com/@wadhawan.neha06/machine-learning-model-evaluation-methods-which-one-to-use-f659cd20d759
- Mutuvi, Steve, "Introduction to Machine Learning Model Evaluation." Accessed October 4, 2023. https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f
- Raschka, Sebastian, "Machine Learning FAQ How do I evaluate a model?" Accessed October 4, 2023. https://sebastianraschka.com/faq/docs/evaluate-a-model.html
- Page, David, "Machine Learning Methodology." Accessed October 4, 2023. http://pages.cs.wisc.edu/~dpage/cs760/
- Hansen, Casper, "Nested Cross-Validation Python Code." Accessed October 4, 2023. https://mlfromscratch.com/nested-cross-validation-python-code/#/
- Cochrane, Courtney, "Time Series Nested Cross-Validation" Accessed October 4, 2023. https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9
- Brownlee, Jason, "Statistical Significance Tests for Comparing Machine Learning Algorithms" Accessed October 4, 2023. https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/
- Swalin, Alvira, "Choosing the Right Metric for Evaluating Machine Learning Models — Part 2" Accessed October 4, 2023. https://medium.com/usf-msds/choosing-the-right-metric-for-evaluating-machine-learning-models-part-2-86d5649a5428
- D., Erika, "Looking at R-Squared." Accessed October 4, 2023. https://medium.com/@erika.dauria/looking-at-r-squared-721252709098