Accuracy Metrics and Model Selection

Lida Kanari

Mathematical Institute
University of Oxford

Introduction to Machine Learning, October 2025

UNIVERSITY OF OXFORD
Mathematical Institute

Oxford Mathematics

Table of Contents

Why Metrics Matter

Supervised learning:

- Goal: Predict $Y \in \mathcal{Y}$ from $X \in \mathcal{X}$.
- Metrics quantify performance of $f : \mathcal{X} \to \mathcal{Y}$.
- Key: Balance between training accuracy and generalization.

Unsupervised learning:

- No ground truth labels Y.
- Metrics measure structure quality (e.g. cluster compactness, neighborhood preservation).
- Evaluation is less straightforward, often requires heuristics.

Accuracy

Accuracy is important to judge the results of a classification task.

- Counts the number of correct predictions of a classification model.
- Typically, it corresponds to the fraction of correct predictions.
- It is an indicative measure of the performance of a model.
- It can help select the best model.

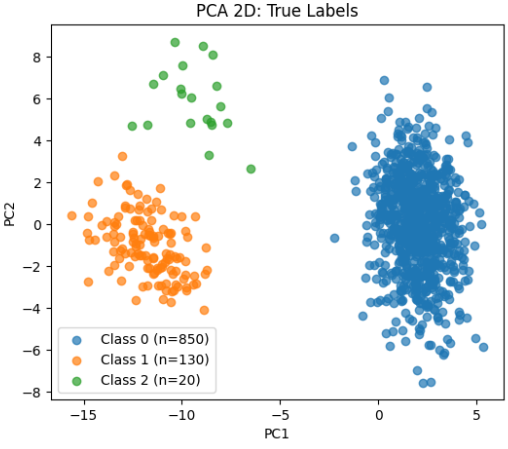
Classification Metrics: Accuracy

Accuracy can be measured as:

$$\text{Acc}(f) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{f(X) = Y\}.$$

by counting the number of correct predictions over the total number of predictions.

Classification Metrics: Accuracy



Classification Metrics: Accuracy

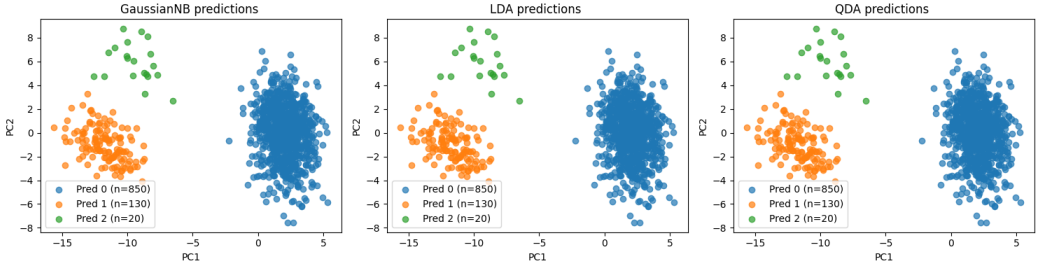
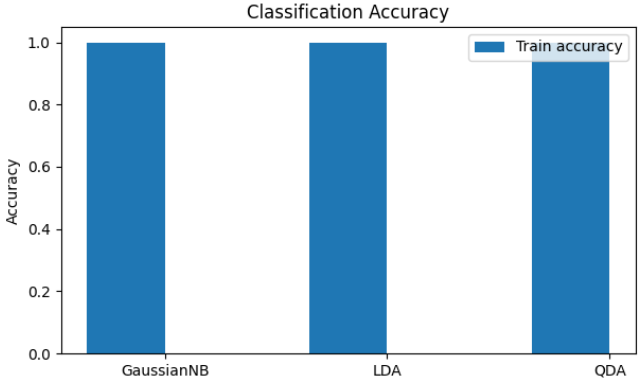


Figure: Classification prediction with different classifiers.

Classification Metrics: Accuracy



Classification Metrics: Accuracy

Question: Can you observe any issues?

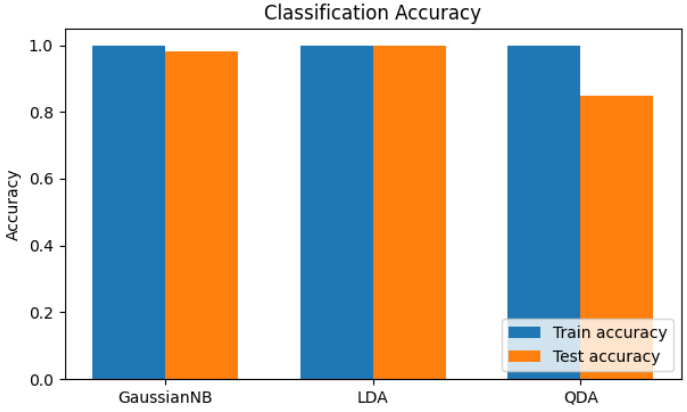
Classification Metrics: Accuracy

Question: Can you observe any issues?

- There is no split between test and training data.
- As a result, all the algorithms perform perfectly well.
- We need to introduce a training set to train the algorithms, and then compute accuracy on the test set.

Classification Metrics: Accuracy

We split the data in train sets: $[68, 10, 2]$ and test sets: $[782, 120, 18]$



Classification Metrics: Accuracy

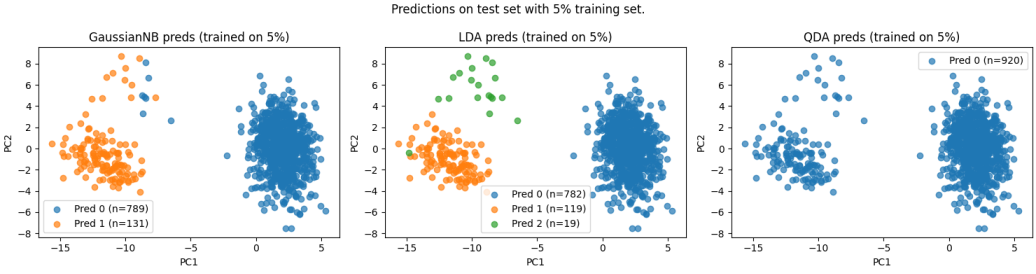


Figure: Classification prediction with different classifiers.

Classification Metrics: Accuracy

Question: Can you observe any issues?

Classification Metrics: Accuracy

Question: Can you observe any issues?

- Small training set leads to classification problems.
- However, the accuracy metric seems to be quite high.
- This is due to the imbalance in the classes. Most data belong to class 0, so a wrong classifier can perform quite well.
- Alternative accuracy metrics are more appropriate for imbalanced datasets.

Classification Metrics: Confusion Matrix

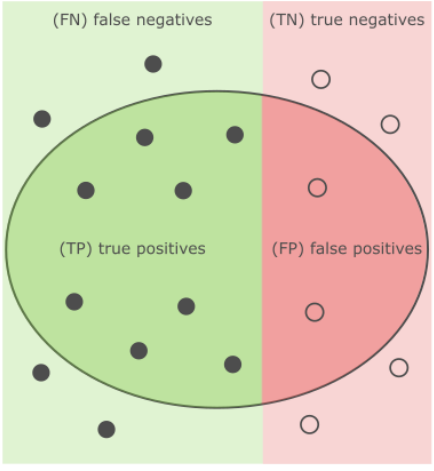
For a binary classification

$$\mathcal{Y} = \{P = 1, N = -1\}$$

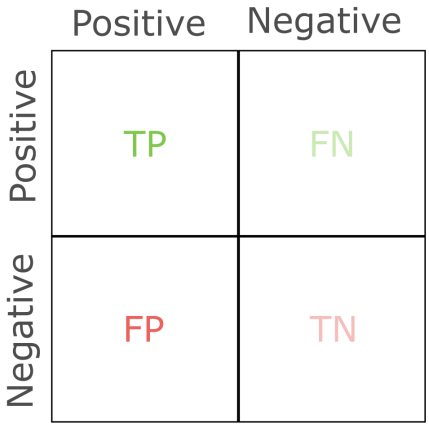
	Pred. P	Pred. N
True P	TP	FN
True N	FP	TN

The dataset is $D = P + N$
The true negative is $N = TN + FP$
The true positive is $P = TP + FN$

Classification Metrics: Confusion Matrix



Classification Metrics: Confusion Matrix



Classification Metrics: Accuracy

Accuracy measures the correct values over all the values:

$$\text{Acc}(f) = \frac{TP + TN}{TP + FP + TN + FN} = \frac{TP + TN}{P + N}$$

Classification Metrics: Precision

Precision measures the positive predictive value of a model, and can be computed as:

$$\text{Prec}(f) = \frac{TP}{TP + FP}$$

Precision is the fraction of true positives over all the positive predictions.

Classification Metrics: Recall

Recall measures the sensitivity of a model and can be calculated by:

$$\text{Rec}(f) = \frac{TP}{TP + FN} = \frac{TP}{P}$$

Sensitivity (true positive rate) is the probability of a positive test result, conditioned on the individual being positive.

Classification Metrics: True negative rate

TNR measures the sensitivity of a model and can be calculated by:

$$\text{TNR}(f) = \frac{TN}{TN + FP} = \frac{TN}{N}$$

Specificity (true negative rate) is the probability of a negative test result, conditioned on the individual being negative.

Classification Metrics: F1-score

F1 can be measured as:

$$\text{F1}(f) = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}} = \frac{2TP}{TP + FP + TP + FN}$$

F1 is 1 when all predictions are correct ($TR = 1, FP = FN = 0$) and 0 when there are no true positives.

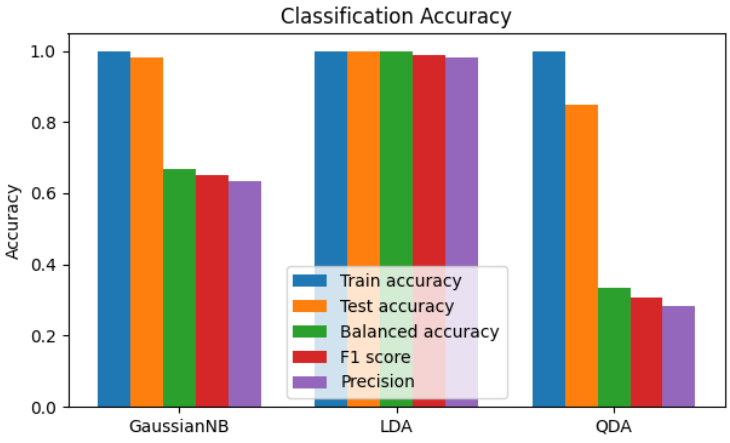
Classification Metrics: Balanced Accuracy

Balanced Accuracy can be measured as:

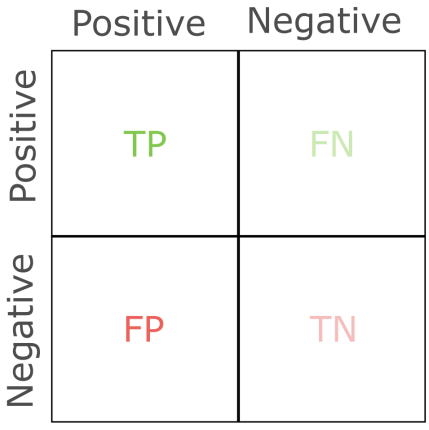
$$\text{BAcc}(f) = (\text{Sensitivity} + \text{Specificity})/2 = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right)$$

For unbalanced datasets, the balanced accuracy gives a better estimate of the correct predictions.

Classification Metrics



Classification Metrics: Confusion Matrix



Classification Metrics: Confusion matrix

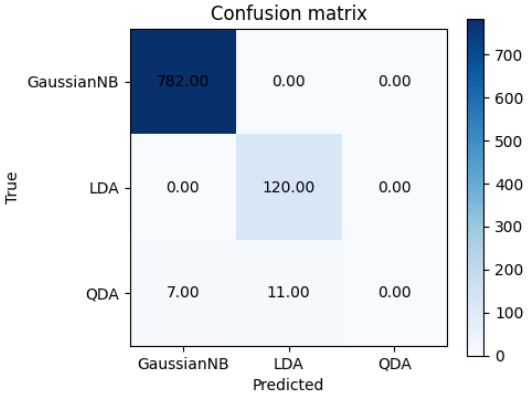


Figure: Confusion matrix for GNB.

Classification Metrics: Confusion matrix

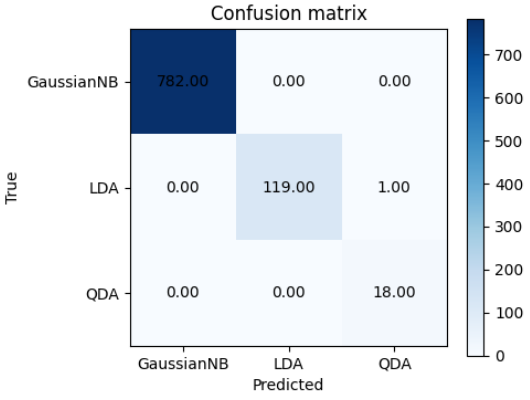


Figure: Confusion matrix for LDA.

Classification Metrics: Confusion matrix

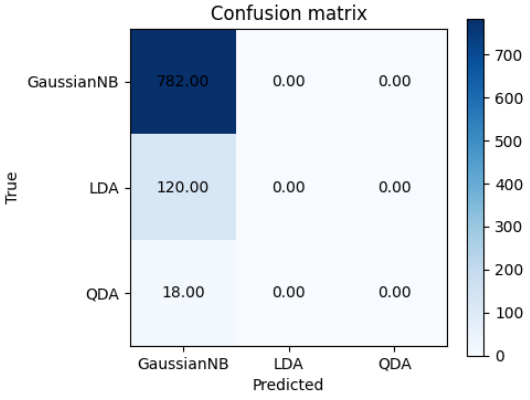
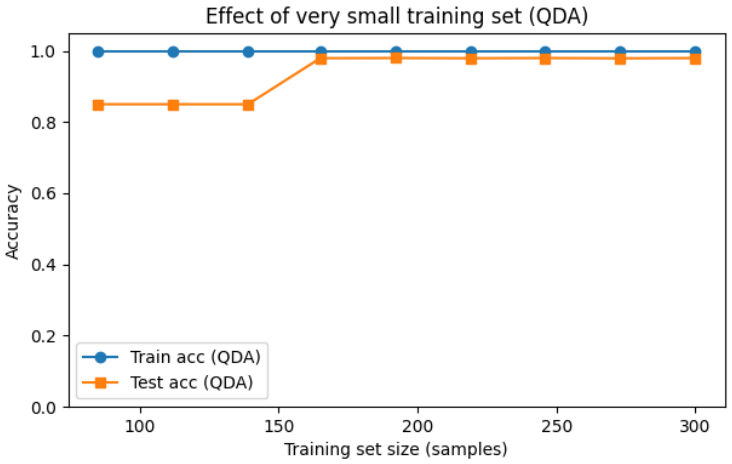


Figure: Confusion matrix for QDA.

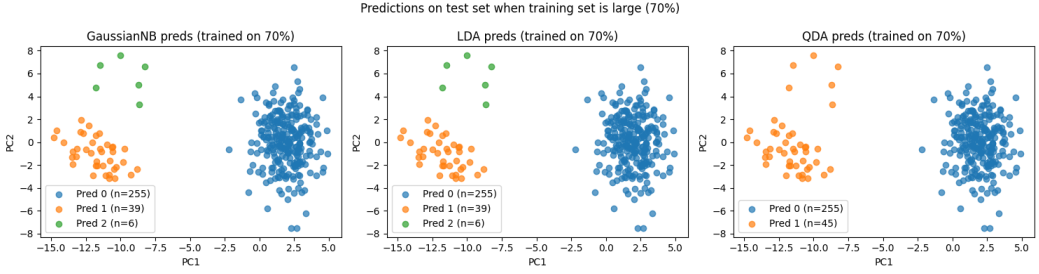
Classification Metrics: Comparison

model	accuracy	balanced_accuracy	precision_macro	recall_macro	f1_macro
GaussianNB	0.980435	0.666667	0.635720	0.666667	0.650573
LDA	0.998913	0.997222	0.982456	0.997222	0.989596
QDA	0.850000	0.333333	0.283333	0.333333	0.306306
LogisticRegression	1.000000	1.000000	1.000000	1.000000	1.000000
RandomForest	0.991304	0.851852	0.981202	0.851852	0.895102
kNN	0.980435	0.666667	0.623188	0.666667	0.643411

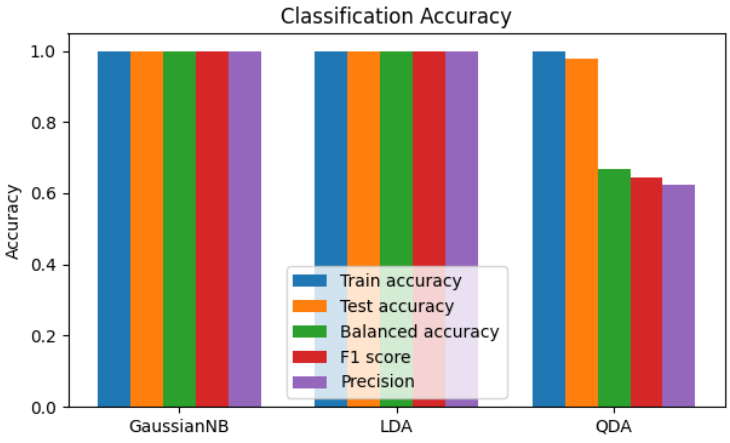
Classification with Balanced Train - Test



Classification with Balanced Train - Test



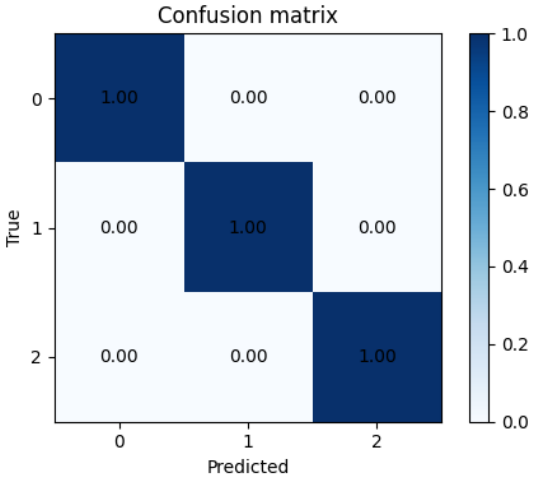
Classification with Balanced Train - Test



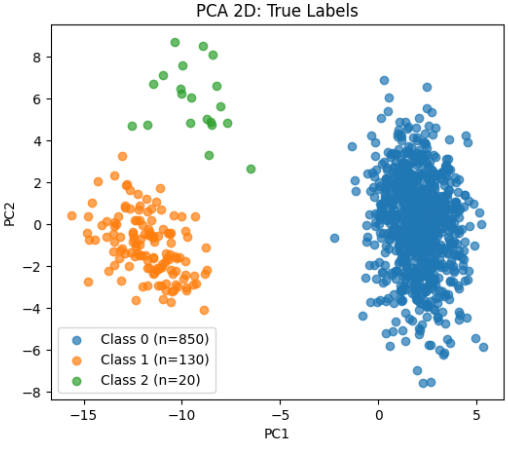
Classification with Balanced Train - Test

model	accuracy	balanced_accuracy	precision_macro	recall_macro	f1_macro
GaussianNB	1.00	1.000000	1.000000	1.000000	1.000000
LDA	1.00	1.000000	1.000000	1.000000	1.000000
QDA	0.98	0.666667	0.622222	0.666667	0.642857
LogisticRegression	1.00	1.000000	1.000000	1.000000	1.000000
RandomForest	1.00	1.000000	1.000000	1.000000	1.000000
kNN	1.00	1.000000	1.000000	1.000000	1.000000

Classification with Balanced Train - Test



Feature dependence



Feature dependence

- ML algorithms work better for independent features (see example of Naive Bayes Classifier)
- However, in practice, features are never independent.

Feature dependence

- ML algorithms work better for independent features (see example of Naive Bayes Classifier)
- In practice, features are never independent.

1. How can we check feature dependence?
2. Can we resolve dependencies?

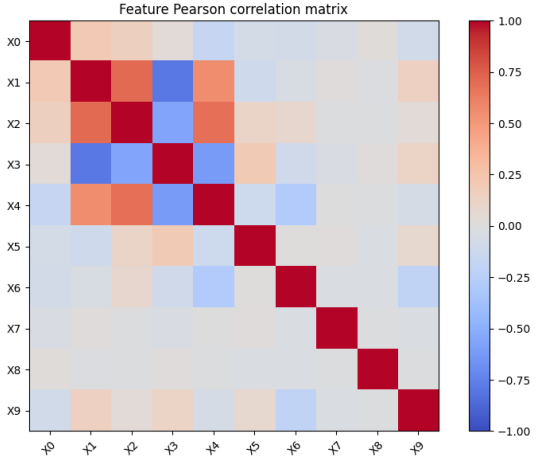
Feature dependence through correlation

Pearson correlation measures the linear correlation between two variables.

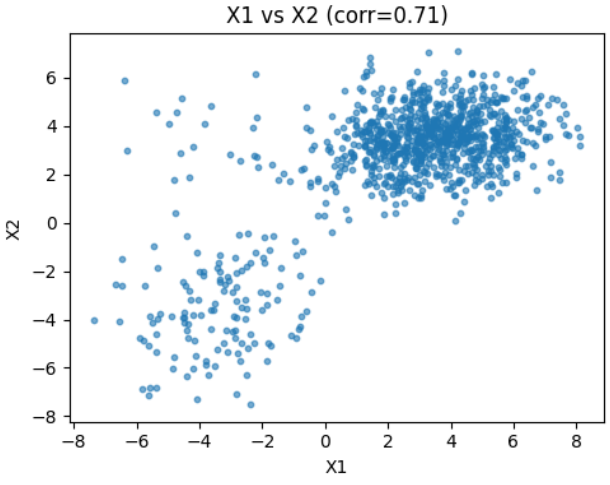
$$PC(\mathcal{X}_1, \mathcal{X}_2) = \frac{Cov(\mathcal{X}_1, \mathcal{X}_2)}{\sigma_1 \sigma_2}$$

Pearson correlation can be computed between each pair of features in a classification problem, to judge interdependence.

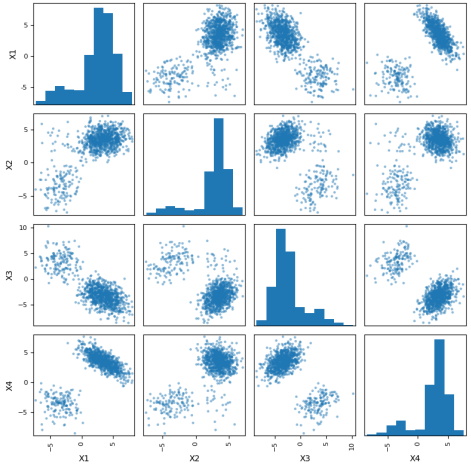
Feature dependence: pairwise correlation



Feature dependence: correlated features



Strongly correlated features



What is feature importance?

- **Feature importance** is a quantitative measure of how much each feature contributes to a model's predictive performance.
- **Permutation approach:** randomly shuffle a single feature's values and measure the performance drop; large drop $\Rightarrow$ high importance.

Feature Importance based on permutations

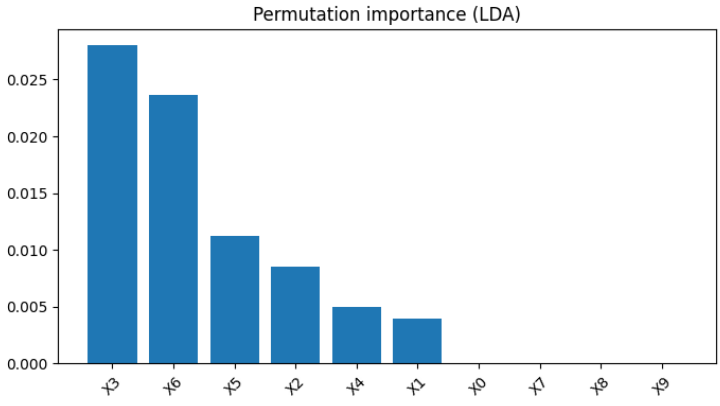
1. For a trained classification model and on the dataset $(\mathcal{X}, \mathcal{Y})$ choose an accuracy metric m for example, accuracy, F1).
2. Compute the *reference score* on the original dataset $s = m(\mathcal{X}, \mathcal{Y})$.
3. For each feature $\mathcal{X}_j$ generate a modified dataset $\tilde{X}_j$ by randomly permuting the entries of feature j. Repeat K times to get an average.
4. Compute the score on the modified data :

$$s_j = \frac{1}{K} \sum_{k=1}^{K} m(\tilde{X}_{k,j}).$$

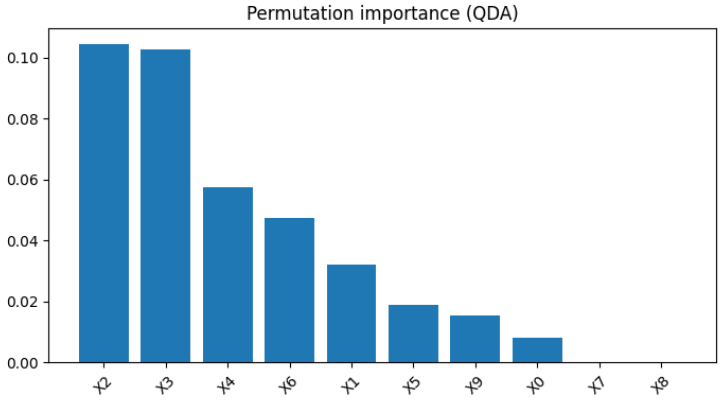
5. The feature importance for feature j is the average performance decrease:

$$i_j = s - s_j.$$

Feature importance: LDA



Feature importance :QDA



Feature importance

What did you notice between the feature importance of LDA and QDA?

Feature importance

- Feature importance depends on the algorithm.
- There is no universal feature selection process.
- Correlation can be used to identify feature dependencies.
- However, removing correlation does not guarantee independence.
- Another way to deal with inter-dependencies is dimensionality reduction.

Overfitting and Underfitting

Underfitting:

- Model too simple $\rightarrow$ poor training and test performance.

Overfitting:

- Model too complex $\rightarrow$ low training error, high test error.

Goal: Find a model that generalizes well.

Overfitting

Intuition: A model that is too complex can "memorize" the training data, including noise, instead of learning patterns that generalize well to new data.
Mathematical view:

$$\hat{R}(f) \text{ small}, \qquad R(f) - \hat{R}(f) \text{ large}.$$

Detection:

- Training error $\downarrow$ but validation/test error $\uparrow$.
- Large gap between $\hat{R}(f)$ (training) and $R(f)$ (test).
- Learning curves show divergence between training and validation errors.

Underfitting

Intuition: A model that is too simple fails to capture the underlying structure in the data.

Mathematical view:

$$\hat{R}(f) \text{ high}, \qquad R(f) \text{ high}.$$

Examples:

- Linear regression on nonlinear data.
- Very shallow decision tree.

Problem: Model has high bias and cannot achieve a low error even with more data.

Overfitting and Underfitting

- Overfitting and underfitting can be better understood by comparing the performance of a classifier on the training and test data.
- It is not sufficient to compute accuracy measurements to judge the performance of a classifier.

Regression

- Regression problem as an example.
- The inputs are n pairs (x_i, y_i) generated from an unknown function f
- The objective is to approximate the function f
- Practically this can be achieved by minimizing a loss function ℓ.

Regression Metrics

Mean Squared Error (MSE):

$$\mathrm{MSE}(f) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2.$$

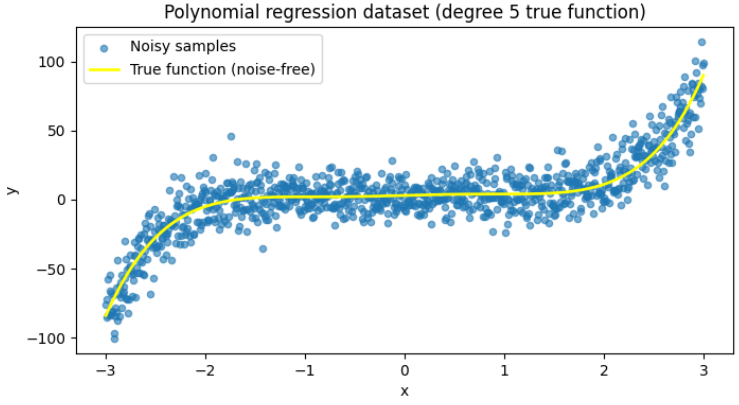
Mean Absolute Error (MAE):

$$\mathrm{MAE}(f) = \frac{1}{n}\sum_{i=1}^{n}|y_i - f(x_i)|.$$

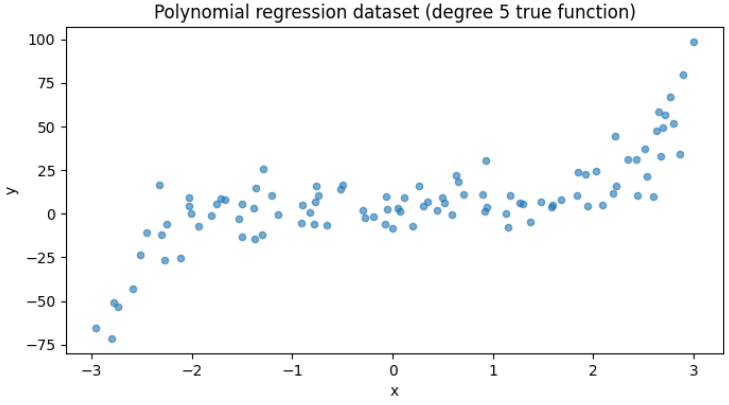
Coefficient of determination (R^2):

$$R^2 = 1 - \frac{\sum_i (y_i - f(x_i))^2}{\sum_i (y_i - \bar{y})^2}.$$

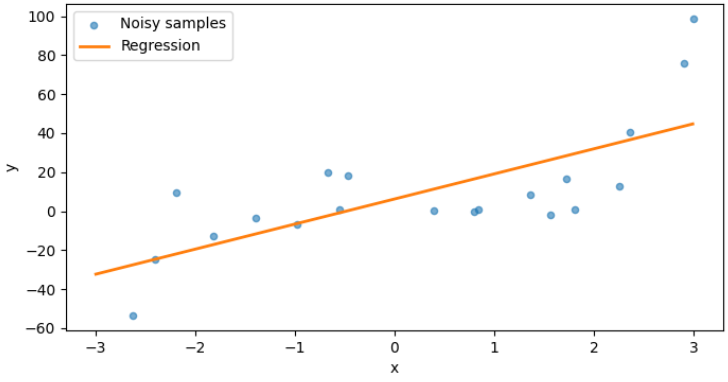
Regression


Polynomial regression dataset (degree 5 true function)
Noisy samples
True function (noise-free)
100
50
0
−50
−100
y
−3
−2
−1
0
1
2
3
x

Regression


Polynomial regression dataset (degree 5 true function)
100
75
50
25
0
−25
−50
−75
y
−3
−2
−1
0
1
2
3
x

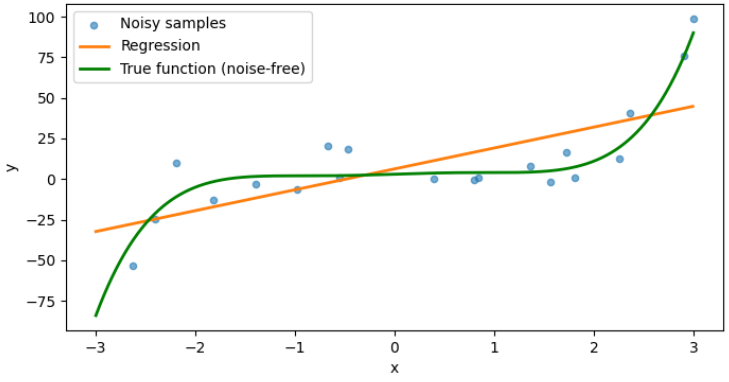
Regression



Regression

What is the problem here?

Regression - Underfitting

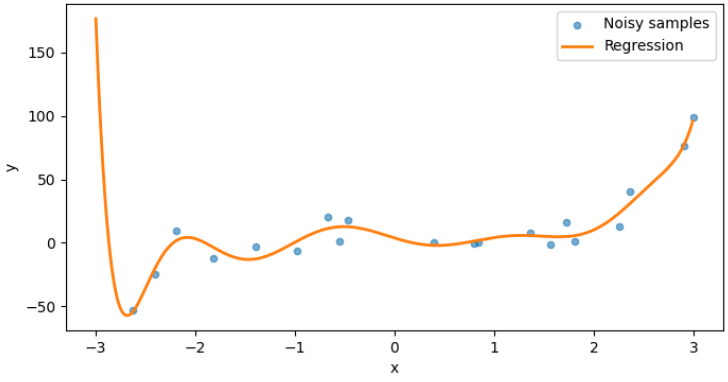


Regression - Underfitting

Problem

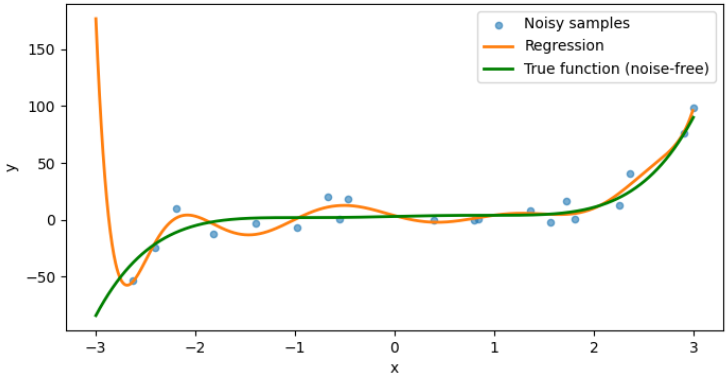
- The proposed model is too simple.
- Cannot capture the complexity of the data
- It is not possible to approximate f with this model
- Increase of the input dataset cannot improve the result

Regression



What is the problem here?

Regression - Overfitting with polynomial degree 20



Problem

- The proposed model is too complex.
- There are not sufficient data.
- It is not possible to approximate f with this model

Solutions

- Increase the input dataset
- Modify the model

Model Selection Based on Metrics

Principle: Choose the model with the best performance.

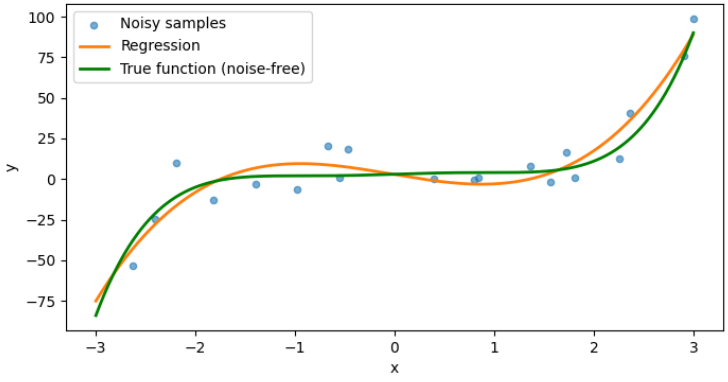
Cautions:

- Metric choice depends on application (accuracy vs recall vs F1).
- It is better to use multiple metrics for robustness.

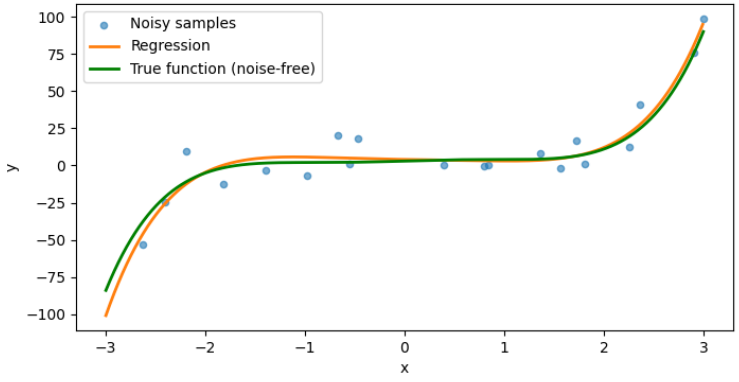
Regression - Model selection

model	mse_train	mse_test	mae_train	mae_test	r2_train	r2_test
Linear	487.3688	374.7133	17.7379	15.2274	0.5170	0.5084
Poly(deg=3)+Linear	114.1698	158.1634	9.0454	10.0361	0.8869	0.7925
Poly(deg=5)+Linear	94.4406	143.6254	8.2439	9.3838	0.9064	0.8116
Poly(deg=10)+Linear	49.9045	792.4995	5.9728	14.1337	0.9505	-0.0396
Poly(deg=20)+Linear	49.9045	792.4995	5.9728	14.1337	0.9505	-0.0396
Ridge(deg5,alpha=1.0)	94.4497	143.3460	8.2494	9.3745	0.9064	0.8120
Lasso(deg5,alpha=0.1)	94.5023	142.2112	8.2667	9.3516	0.9063	0.8134
RandomForest	40.6753	210.7079	5.0096	11.1607	0.9597	0.7236
kNN(k=5)	345.4941	322.3642	13.1055	12.6565	0.6576	0.5771

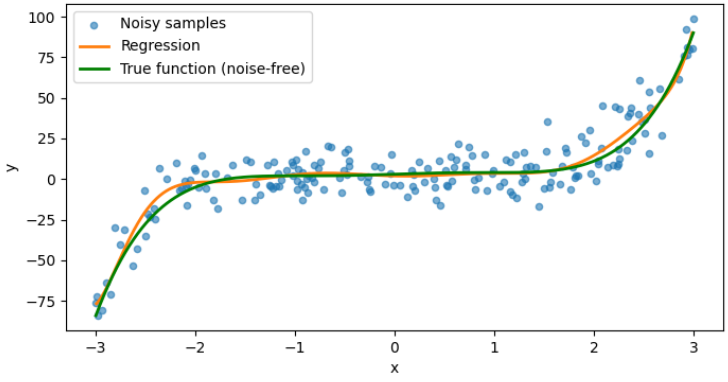
Regression - Model Selection: Polynomial degree 3



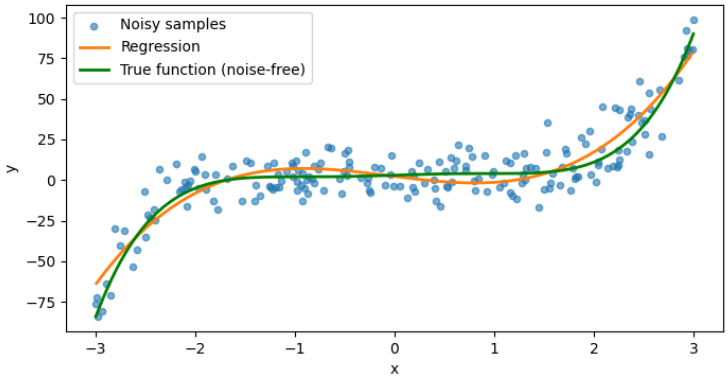
Regression - Model Selection: Polynomial degree 5



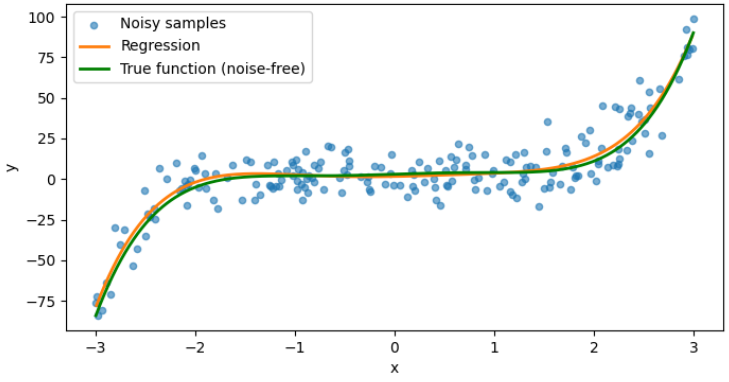
Regression - Input Data - Polynomial degree 10



Regression - Input Data - Polynomial degree 3



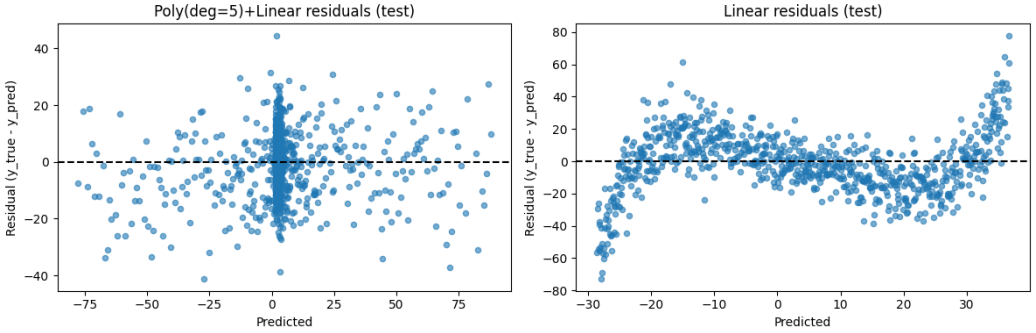
Regression - Input Data - Polynomial degree 5



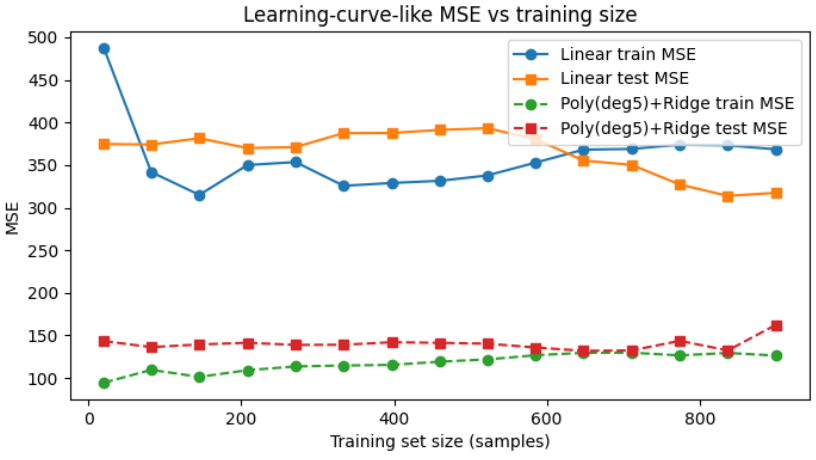
Regression - Model selection

model	mse_train	mse_test	mae_train	mae_test	r2_train	r2_test
Linear	350.4465	369.1024	14.4748	14.5177	0.4924	0.5305
Poly(deg=3)+Linear	131.7244	155.4680	9.4793	9.8002	0.8092	0.8023
Poly(deg=5)+Linear	109.2966	141.1239	8.4416	9.2281	0.8417	0.8205
Poly(deg=10)+Linear	105.8086	143.2886	8.4460	9.3723	0.8467	0.8177
Poly(deg=20)+Linear	105.8086	143.2886	8.4460	9.3723	0.8467	0.8177
Ridge(deg5,alpha=1.0)	109.2971	141.0927	8.4416	9.2270	0.8417	0.8205
Lasso(deg5,alpha=0.1)	109.4377	140.7534	8.4576	9.2181	0.8415	0.8210
RandomForest	23.4670	204.1351	3.9201	11.2747	0.9660	0.7403
kNN(k=5)	93.7886	163.3131	7.8771	10.0347	0.8641	0.7923

Regression - Model selection



Regression - Model selection



General Considerations: Bias–Variance

Expected error decomposition:

$$\mathbb{E}[(Y - f(X))^2] = \text{Bias}^2 + \text{Variance} + \text{Irreducible error}.$$

Bias–variance tradeoff:

- High bias $\rightarrow$ underfitting.
- High variance $\rightarrow$ overfitting.
- Reducing bias often increases variance and vice versa.

Bias measures the systematic error of an estimator.
For an estimator $\hat{f}(x)$ of the true function $f(x)$:

$$\text{Bias}[\hat{f}(x)] = \mathbb{E}[\hat{f}(x)] - f(x)$$

where $\mathbb{E}[\hat{f}(x)]$ is the expected prediction over training sets.
Interpretation:

- Captures how far the average prediction is from the true function.
- High bias $\Rightarrow$ the model is too simple, cannot capture structure.

Variance measures the sensitivity of an estimator to data fluctuations.
For the estimator $\hat{f}(x)$:

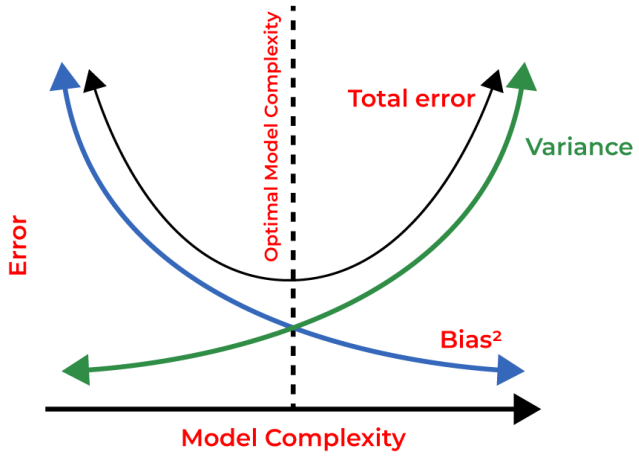
$$\text{Var}[\hat{f}(x)] = \mathbb{E}\Big[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2\Big]$$

where $\hat{f}(x)$ is the predicted value by the model and $\mathbb{E}[\hat{f}(x)]$ the expected prediction over training sets.

Interpretation:

- Quantifies how much $\hat{f}(x)$ changes with different training sets.
- High variance $\Rightarrow$ model overfits, capturing noise in data.

Bias - Variance tradeoff



Model Capacity and Generalization Gap

Model capacity: Size/expressiveness of the hypothesis class.

- Low capacity $\rightarrow$ underfitting (high bias).
- High capacity $\rightarrow$ overfitting (high variance).

Generalization gap:

$$\text{Gap}(f) = R(f) - \hat{R}(f).$$

Training, Validation, Test Sets

Splitting data:

- **Training set:** learn parameters.
- **Validation set:** tune hyperparameters.
- **Test set:** final evaluation of generalization.

Warning: Do not use the test set during model selection.

Cross-Validation

Mathematical Institute

k-fold cross-validation:

- Partition data into k subsets.
- Train on $k-1$, validate on the remaining one.
- Repeat and average errors.

Advantages:

- Efficient use of limited data.
- Reduces variance in performance estimates.

Thank you! Questions?