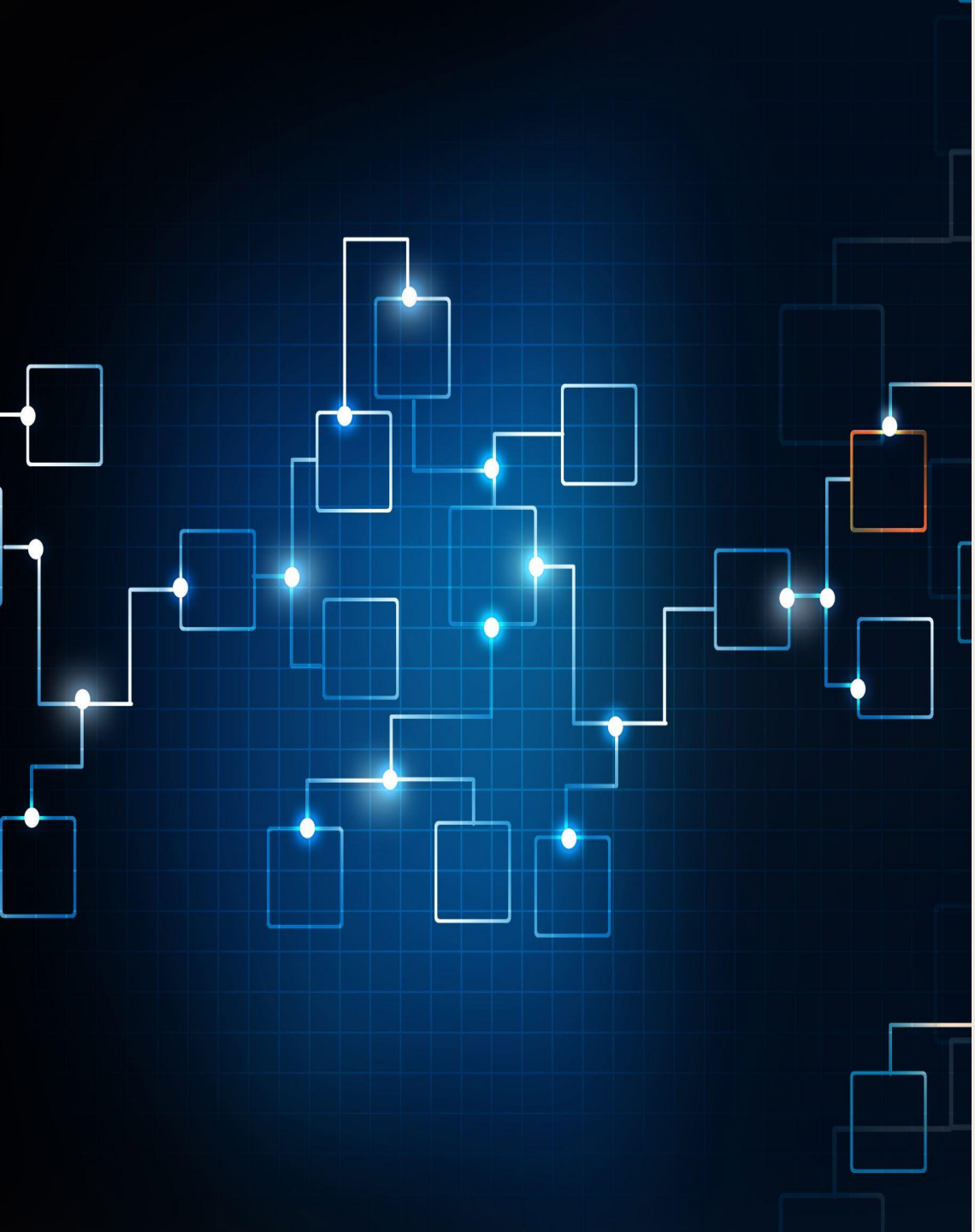


HIT391

MACHINE LEARNING: ADVANCEMENTS AND APPLICATIONS

CHARLES DARWIN UNIVERSITY
AUSTRALIA

Week 6:
Model Evaluation and Finetuning

- **Learning Outcomes**
  - Model Complexity - An example, Training error & testing error
  - Model Selection
  - Cross Validation
    - Holdout cross validation
    - K-fold cross validation
    - Leave-one-out
  - Model Finetuning – Grid Search

Loss Function

- Least squares:

Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for Y based on the ith value of X. Then $e_i = y_i - \hat{y}_i$ represents the ith *residual*
We define the *residual sum of squares* (RSS) as

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

or equivalently as

$$\text{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \ldots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2.$$

The least squares approach chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize the RSS. The minimizing values can be shown to be

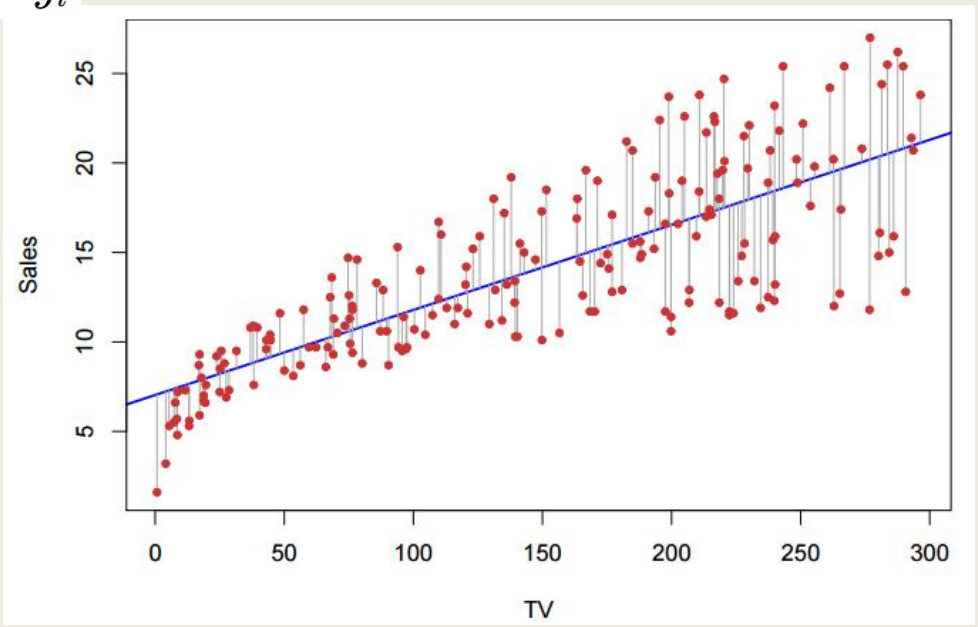
$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

where $\bar{y} \equiv \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{x} \equiv \frac{1}{n}\sum_{i=1}^{n} x_i$ are the sample means.

An Example

- This plot shows the least squares fit of a linear regression model that predicts Sales based on TV advertising budget.
  - The red dots are the actual observed sales for different TV ad budgets.
  - The blue line is the fitted regression line — the line that best summarizes the relationship between TV and Sales using the least squares method.
  - The vertical grey lines show the residuals — the difference between the actual and predicted sales. $\text{Residual} = y_i - \hat{y}_i$

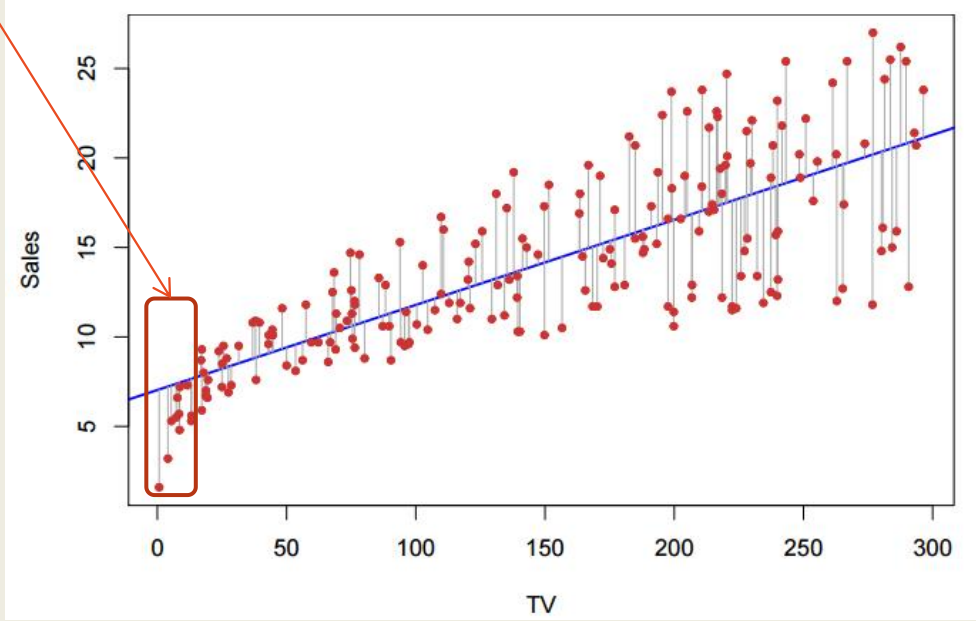


After fitting the model, we compute the **Training Error** – how well our model fits the data it was trained on (e.g., **Mean Squared Error** on **training set**).

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

An Example (Cont.)

- Overall, the linear model does a good job: as TV spending increases, sales also tend to increase.
- However, in the left side of the plot (i.e., for low TV budgets), the linear model appears to underfit — it doesn't capture all the variation, and the points show more deviation from the line.



Training Error

- After fitting the model, we compute the Training Error—how well our model fits the data it was trained on (e.g., Mean Squared Error on training set).
- But a low training error doesn't guarantee the model will perform well on unseen data (Testing set).
- A linear model is often a good starting point, but it might not capture all patterns in the data — especially in areas with high variance or non-linearity.

Training Error vs Testing Error

- Training error (Training dataset)
  - can be easily calculated by applying the **loss function** to the data samples in the **training set**.

	Features			label
	TV	radio	newspaper	sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
6	8.7	48.9	75	7.2
7	57.5	32.8	23.5	11.8

- Testing error (Testing dataset)
  - is the average error that results from using **the trained model** to predict the label on the input, one that was not used in training.

	TV	radio	newspaper	Actual label: sales
8	120.2	19.6	11.6	13.2
9	8.6	2.1	1	4.8
10	199.8	2.6	21.2	10.6
11	66.1	5.8	24.2	8.6
12	214.7	24	4	17.4

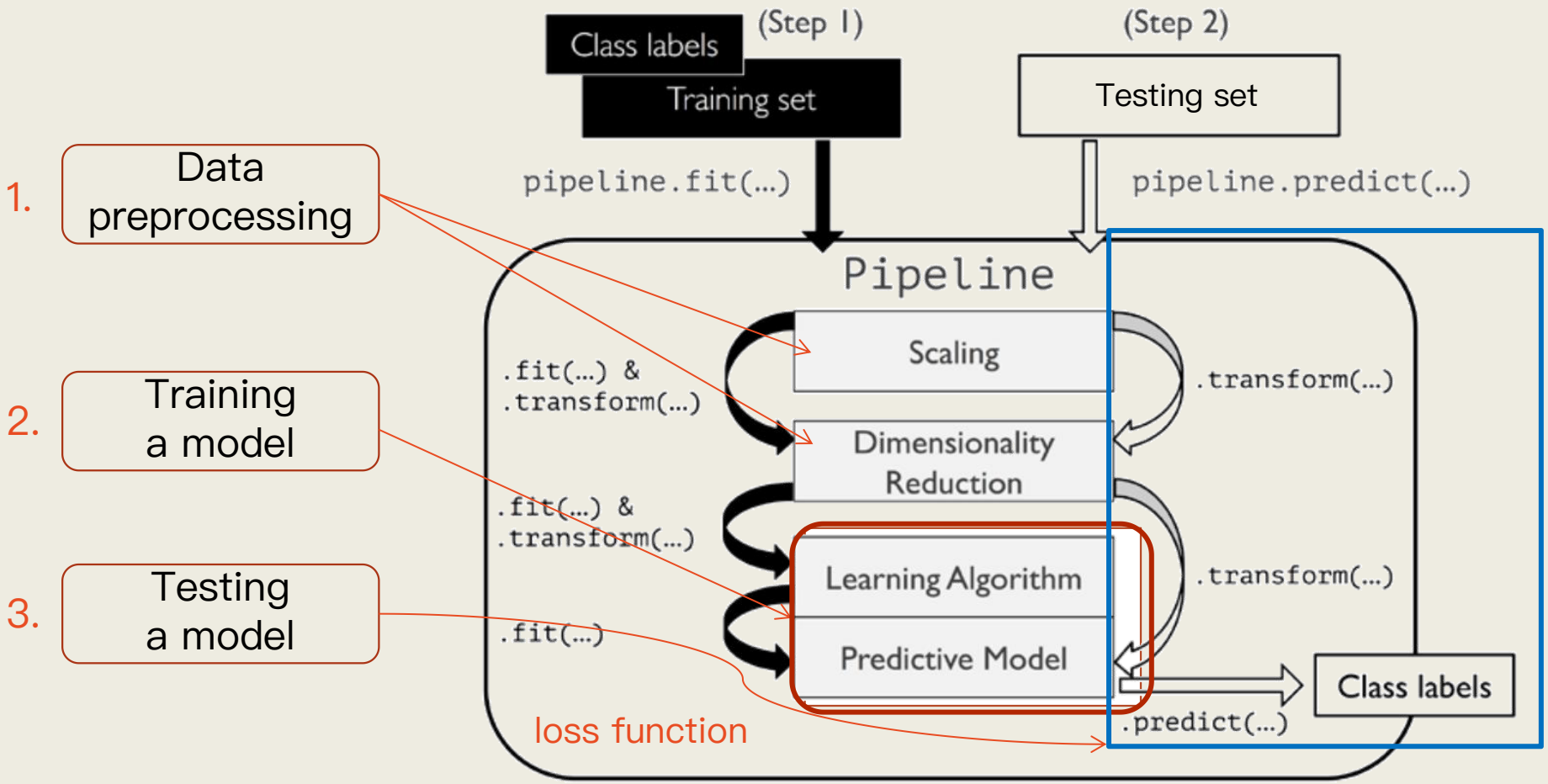
Predict

Predicted sales
12.3
10.1
9.2
15.6
2.5

Predicted label

Procedures of ML

- Classification



Model Complexity

- **Model complexity** refers is a measure of how well a model can capture the underlying patterns in the data.
- In machine learning, model complexity is often associated with the **number of parameters** in a model and its ability to fit both the training data and generalize to new, unseen data.
  - **Simple Models:** Simple models have **few parameters**, making them less flexible therefore they struggle to capture the complexity of the underlying patterns in the data leading to underfitting, where the model performs poorly on the training data as well as on unseen data.
  - **Complex Models:** Complex models have a **larger number of parameters**, allowing them to represent more intricate relationships in the data. While complex models may perform well on the training data, model tends to overfitting.

Model Complexity

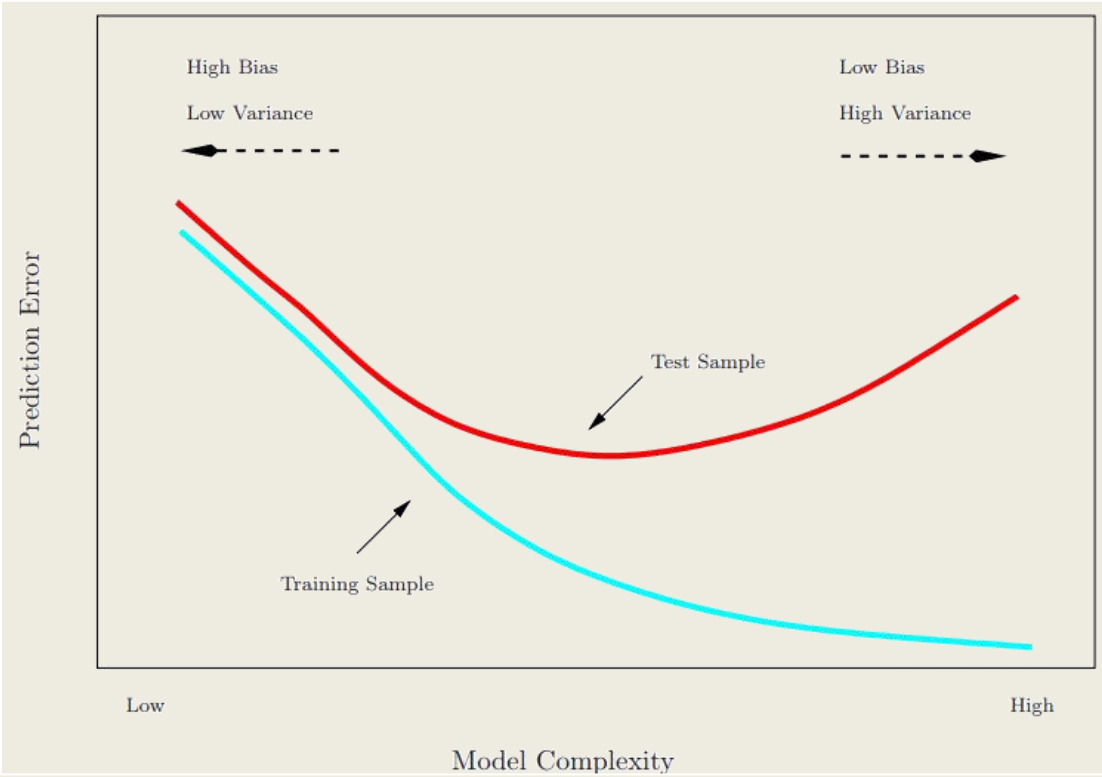
- Modelling complexity can be influenced by several factors:
  - **Number of Features:** The more attributes or features your model scrutinizes, the higher its complexity is likely to be. Too many features can potentially magnify noise and result in overfitting.
  - **Model Algorithm:** The nature of the algorithm used influences the complexity of the model.
  - **Hyperparameters:** Settings such as the learning rate, and regularization parameters can influence the complexity of a machine learning model.

Training Error vs Testing Error

- The training error rate often **is quite different** from the testing error rate, and in particular the former can dramatically underestimate the latter.

- Model complexity:

  – **The number of features** we use in our regression function



[1] The plot comes from *Chapter 2 of Hastie, Tibshirani, and Freedman's 'Elements of Statistical Learning'*

Model Selection

- Model selection involves choosing the best predictive model from a set of candidate models based on their performance.
- The goal is to select a model that generalizes well to unseen data(testing set), avoiding underfitting and overfitting.

Cross–validation(CV)

- Cross-validation is often employed as a practical method for model selection.
- It provides an unbiased and reliable estimate of model performance by evaluating models on multiple validation sets.
- The model with the best average performance across folds (usually measured by metrics such as accuracy, RMSE, AUC, etc.) is typically selected as the optimal model.

Cross–validation(CV)

- Model selection
  - a given classification problem, for which we want to select the optimal values of **finetuning parameters** (also called **hyperparameters**).

    (1) Holdout cross validation

    (2) K-fold cross validation

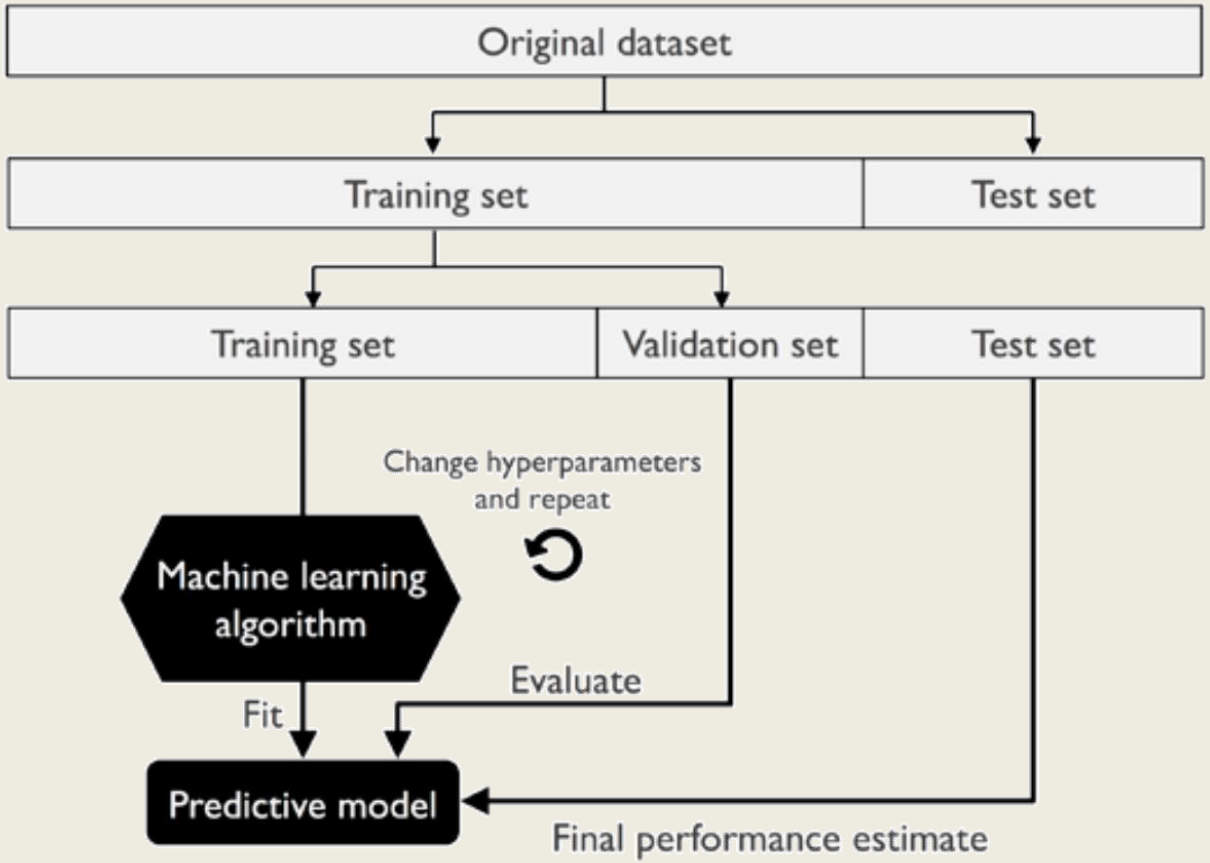
    (3) Leave-one-out

- Dataset preparation:
  - separate a dataset into **three parts**:

    1) a training dataset - for training a model
    2) a validation dataset - for finetuning a model (model selection)
    3) and a test dataset - for testing a model

1. Holdout Cross-validation

- Use **a validation dataset** to **repeatedly evaluate** the performance of the model after training using different hyperparameter values



Disadvantages – Holdout CV

- The performance estimate may be very sensitive to **how we partition the training dataset into the training and validation subsets**;
  - the estimate will vary for different examples of the data.

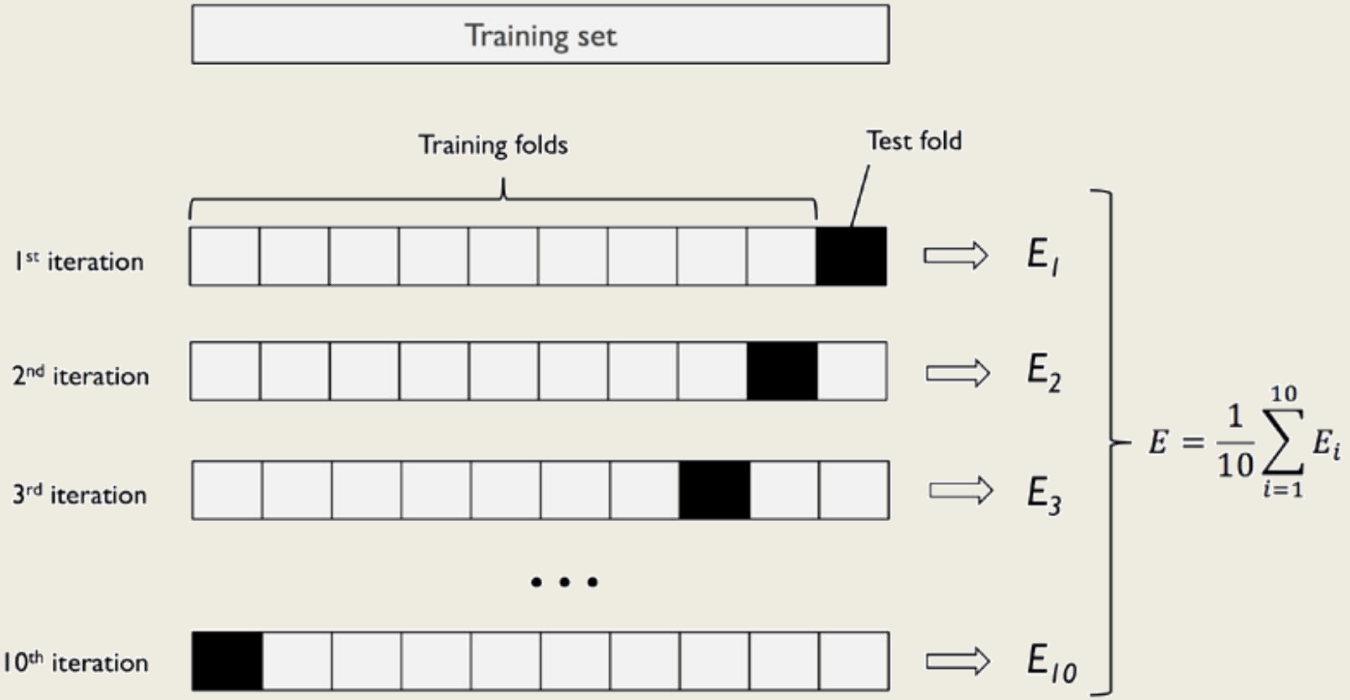
  —> **k-fold cross-validation:** a more robust technique for performance estimation, , where we repeat the holdout method **k** times on **k** subsets of the training data.

2. K–fold Cross–validation

- Randomly split the training dataset into k folds without replacement.
  - k – 1 folds - for training a model
  - the remaining 1 fold - for finetuning a model (model selection)
- This procedure is repeated k times so that we obtain k models and performance estimates.

Example of k=10

- 9 folds - for training
- the remaining 1 fold - for the model evaluation
- the estimated performances, E_i (for example, classification accuracy or error), for each fold are then used to calculate the estimated average performance, E, of the model:



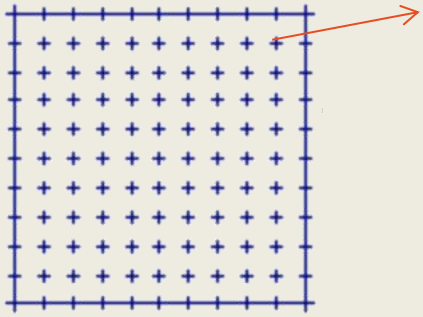
3. Leave–one–out Cross–validation(LOOCV)

- Leave-one-out cross-validation (LOOCV)
  - **A special case** of k-fold cross-validation methods
  - we set the number of folds equal to the number of training examples (k = n)
  - n-1 training examples - for training the model
  - only 1 training example - for testing during each iteration
- A recommended approach for working with very small datasets.

Model Finetuning – Grid Search

- Grid search
  - For hyperparameters of a model - e.g., k in KNN
  - search k from a range of {1,2, ..., 10}
- More than one hyperparameters
  - Perform grid search for all parameters , one selects a finite set of "reasonable" values for each

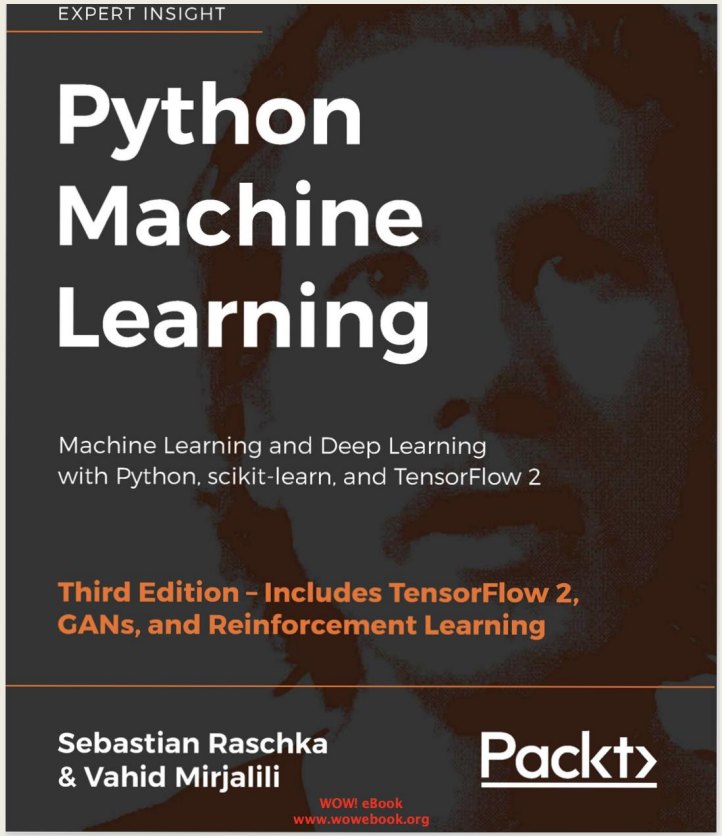
$$C \in \{10, 100, 1000\}$$
$$\gamma \in \{0.1, 0.2, 0.5, 1.0\}$$



Summary

- Model Complexity - An example, Training error & testing error
- Model Selection
- Cross Validation
  - Holdout cross validation
  - K-fold cross validation
  - Leave-one-out
- Model Finetuning – Grid Search

References



- https://www.statlearning.com/resources-python
- https://www.statlearning.com/
- Sebastian Raschka, etl., ‘Python machine learning’, Third Edition