Aprendizado de Máquina e Reconhecimento de Padrões 2021.2

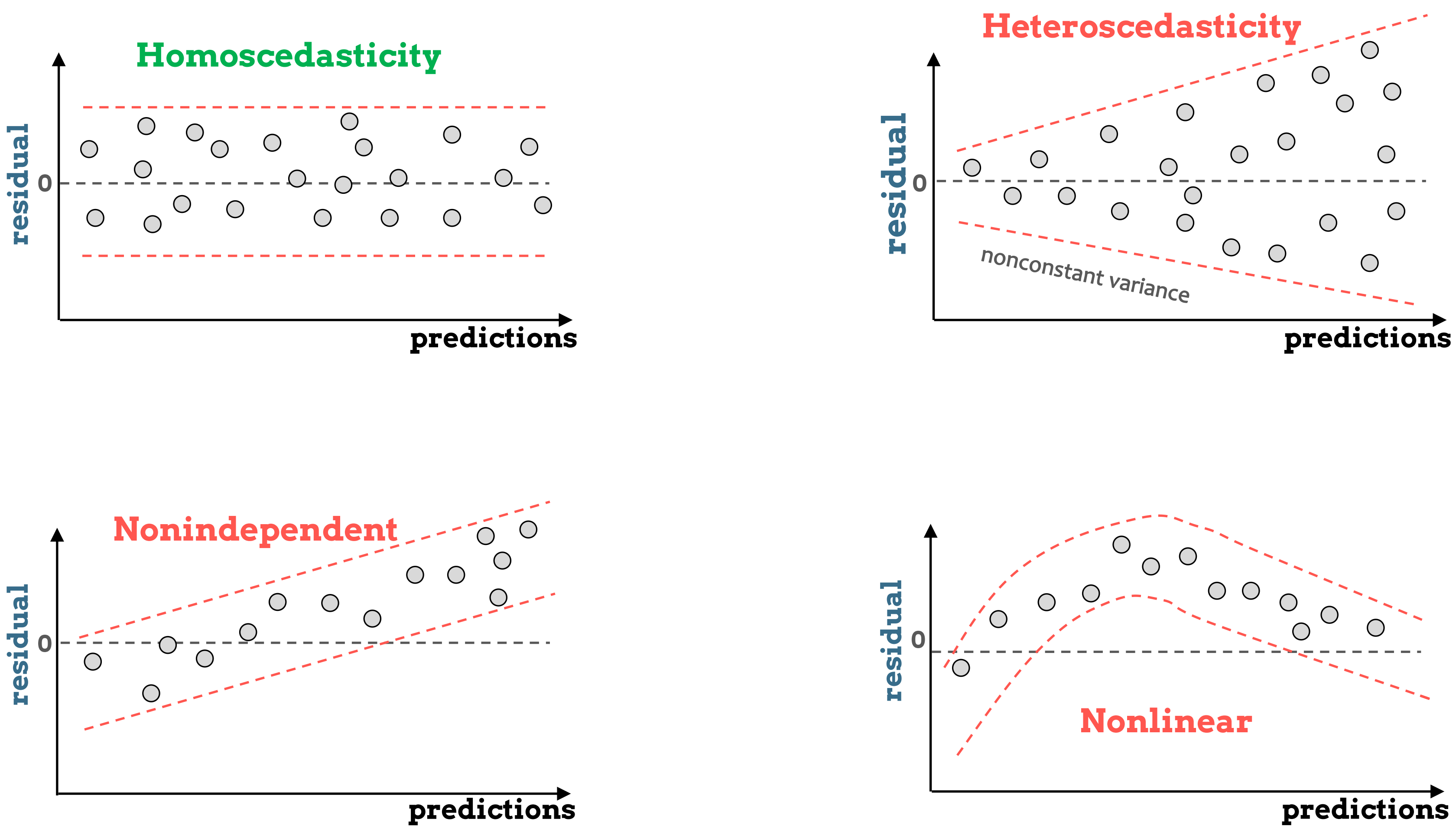
Machine Learning Concepts

Prof. Samuel Martins (Samuka)
samuel.martins@ifsp.edu.br

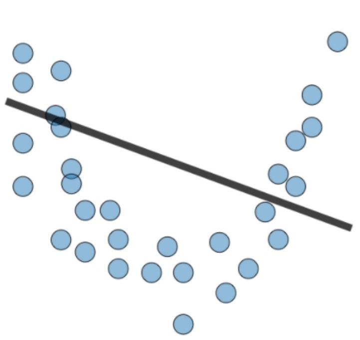
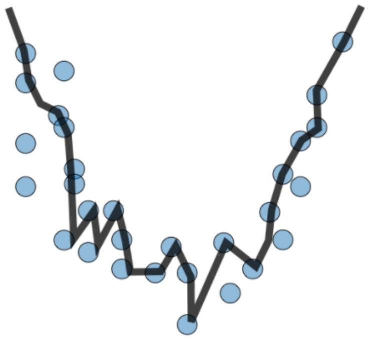
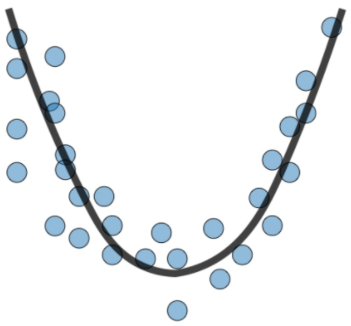
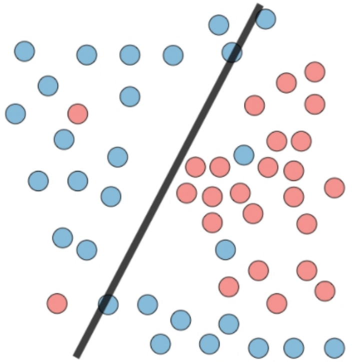
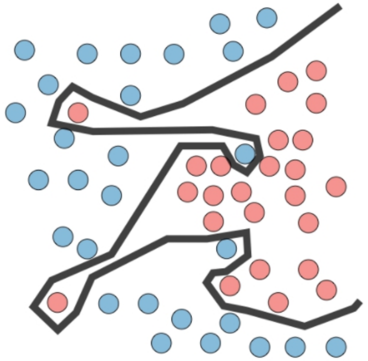
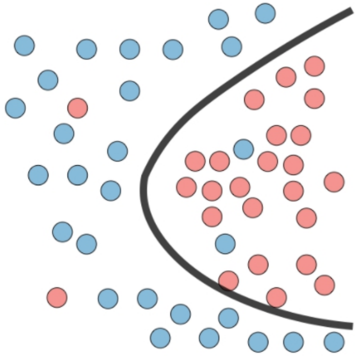
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
São Paulo
Campus Campinas

Checking Homoscedasticity Visually

Checking Homoscedasticity Visually



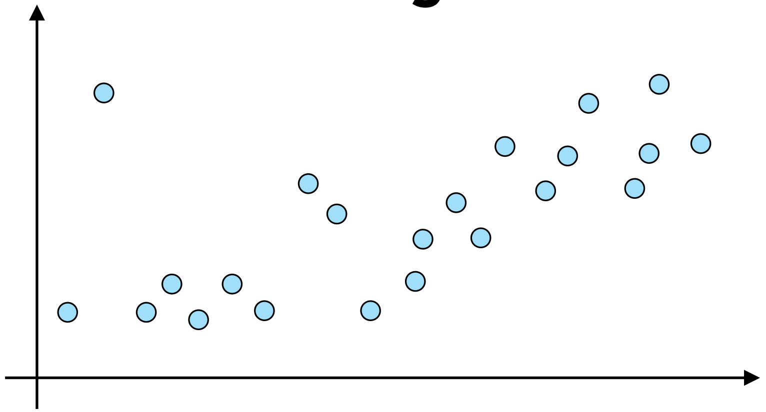
Overfitting vs Underfitting

	Underfitting	Overfitting	Just right
Symptoms	• High training error • Training error close to test error • High bias	• Very low training error • Training error much lower than test error • High variance	• Training error slightly lower than test error
Regression illustration			
Classification illustration			
Possible remedies	• Complexify model • Add more features • Train longer	• Perform regularization • Get more data	

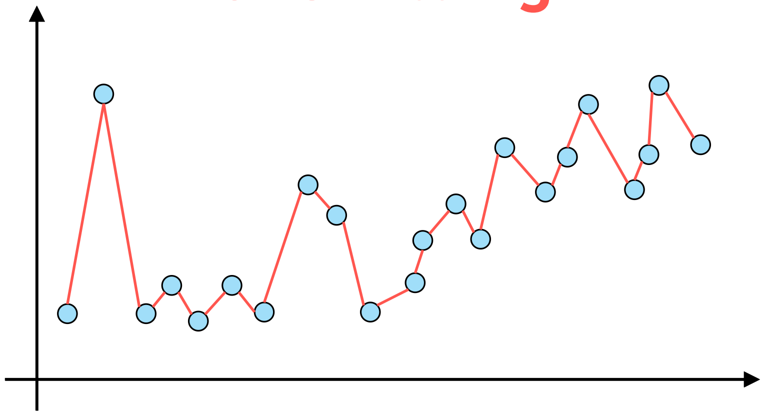
https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks

Overfitting vs Underfitting

Training Data

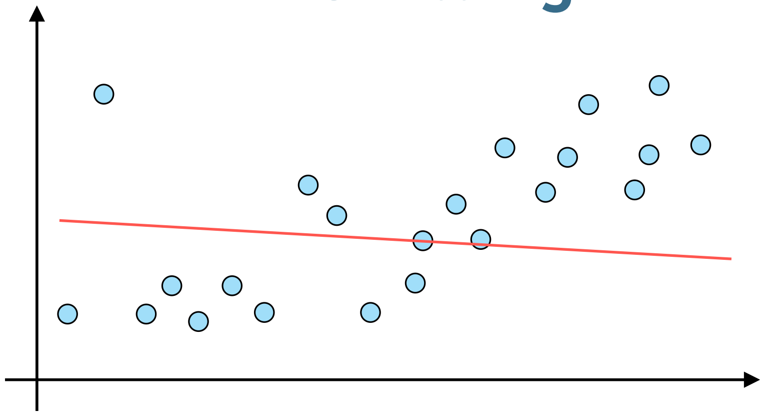


Overfitting



The **overall cost/error is really small**, but **the generalization of the model is unreliable:** the model learned "too much" from the training data.

Underfitting



The model ***has not learned enough*** from the training data, resulting in **low generalization** and **unreliable predictions**.

Feature Scaling

Normalization

```
X_new = (X - X_min)/(X_max - X_min)
```

Standardization

```
X_new = (X - mean)/Std
```

Robust Scaler

X_new = (X – median)/IQR

Normalization

```
X_new = (X - X_min)/(X_max - X_min)
```

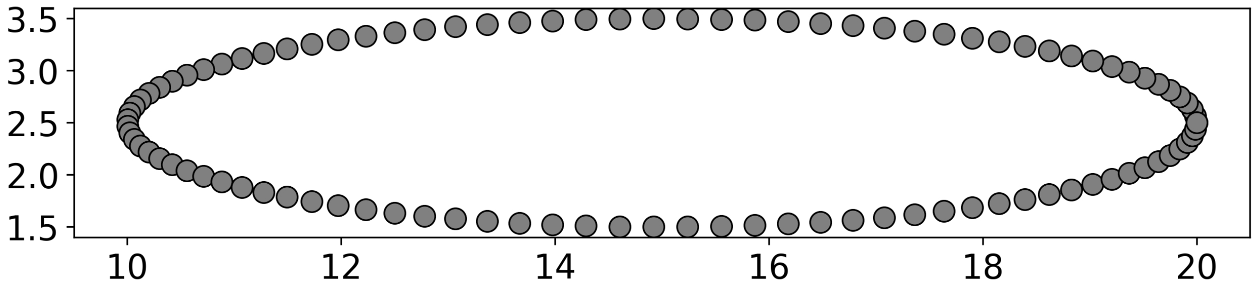
Standardization

```
X_new = (X - mean)/Std
```

Robust Scaler

X_new = (X – median)/IQR

Data



Normalization

```
X_new = (X - X_min)/(X_max - X_min)
```

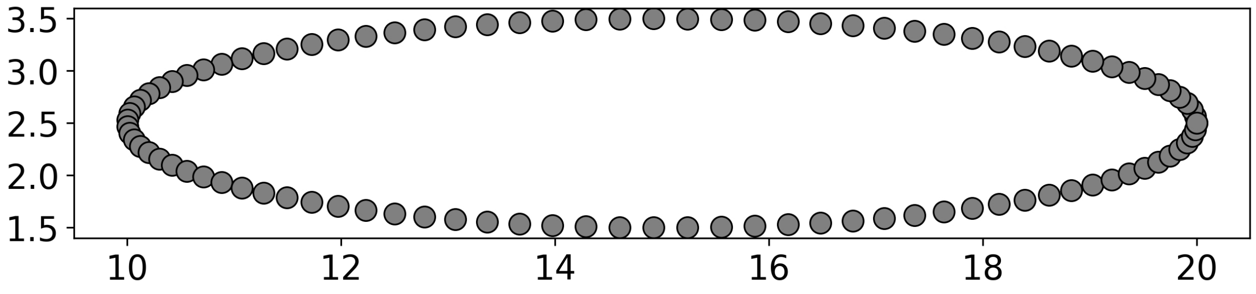
Standardization

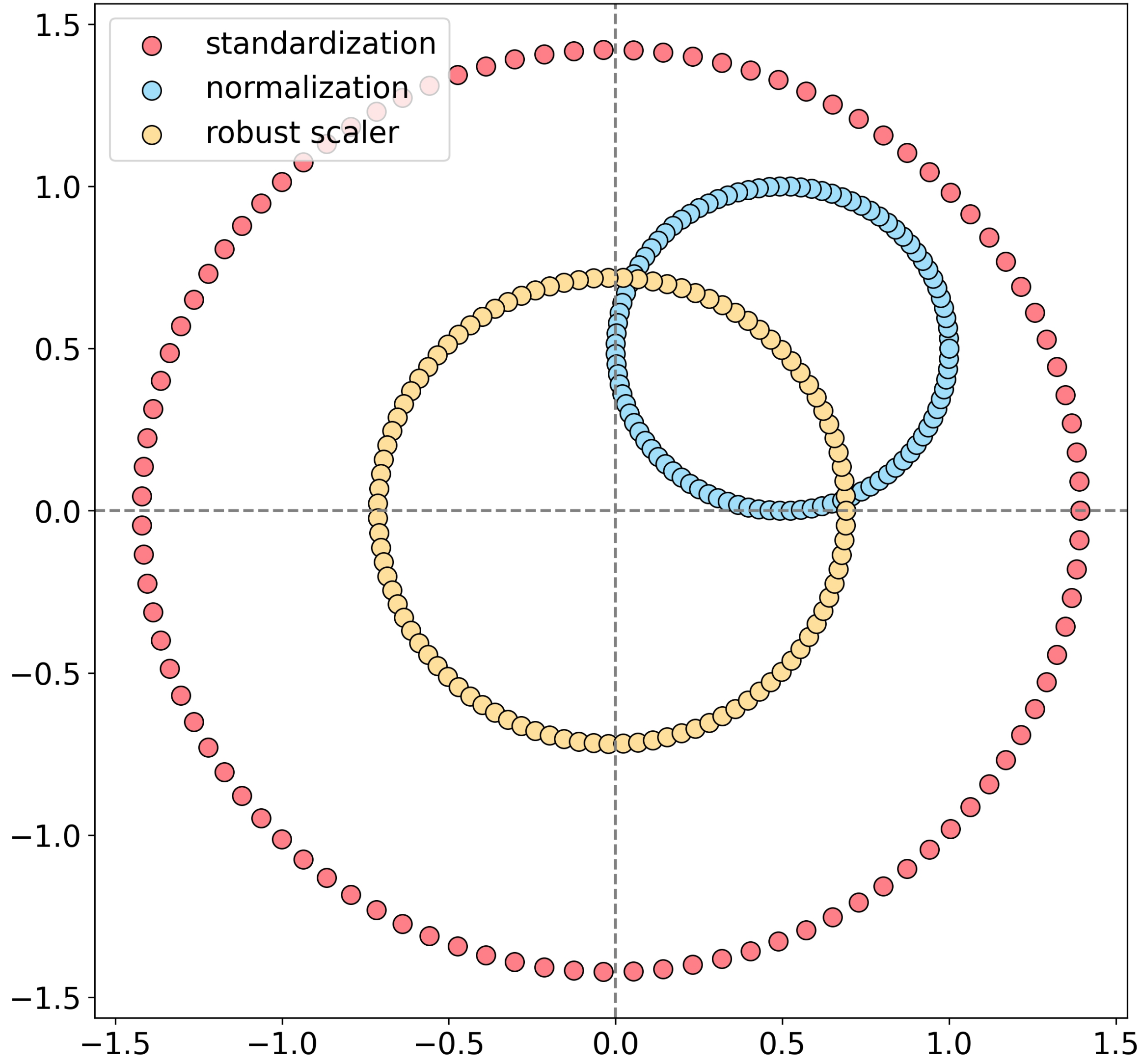
```
X_new = (X - mean)/Std
```

Robust Scaler

X_new = (X – median)/IQR

Data





Normalization

```
X_new = (X - X_min)/(X_max - X_min)
```

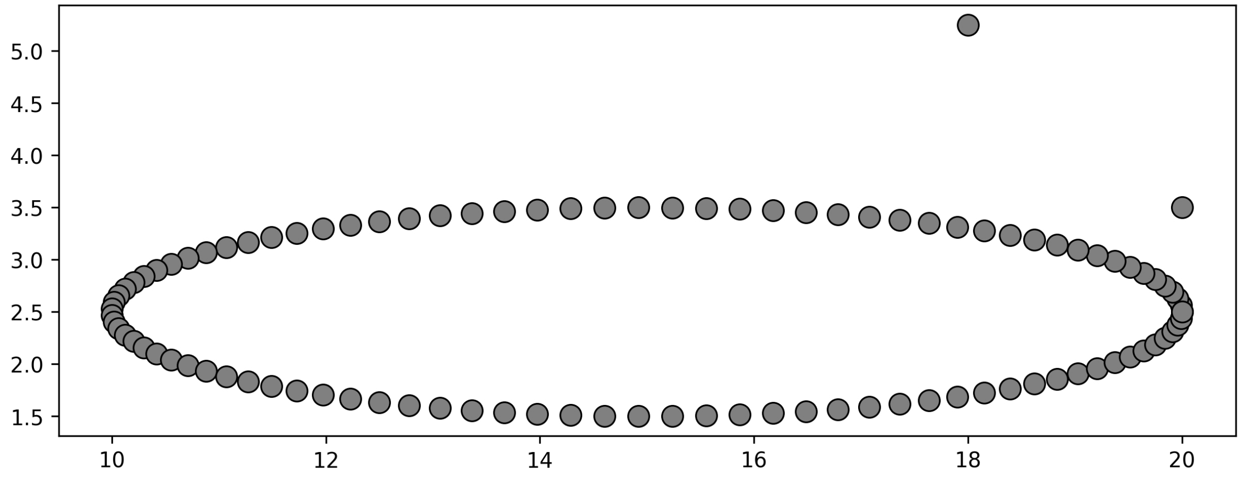
Standardization

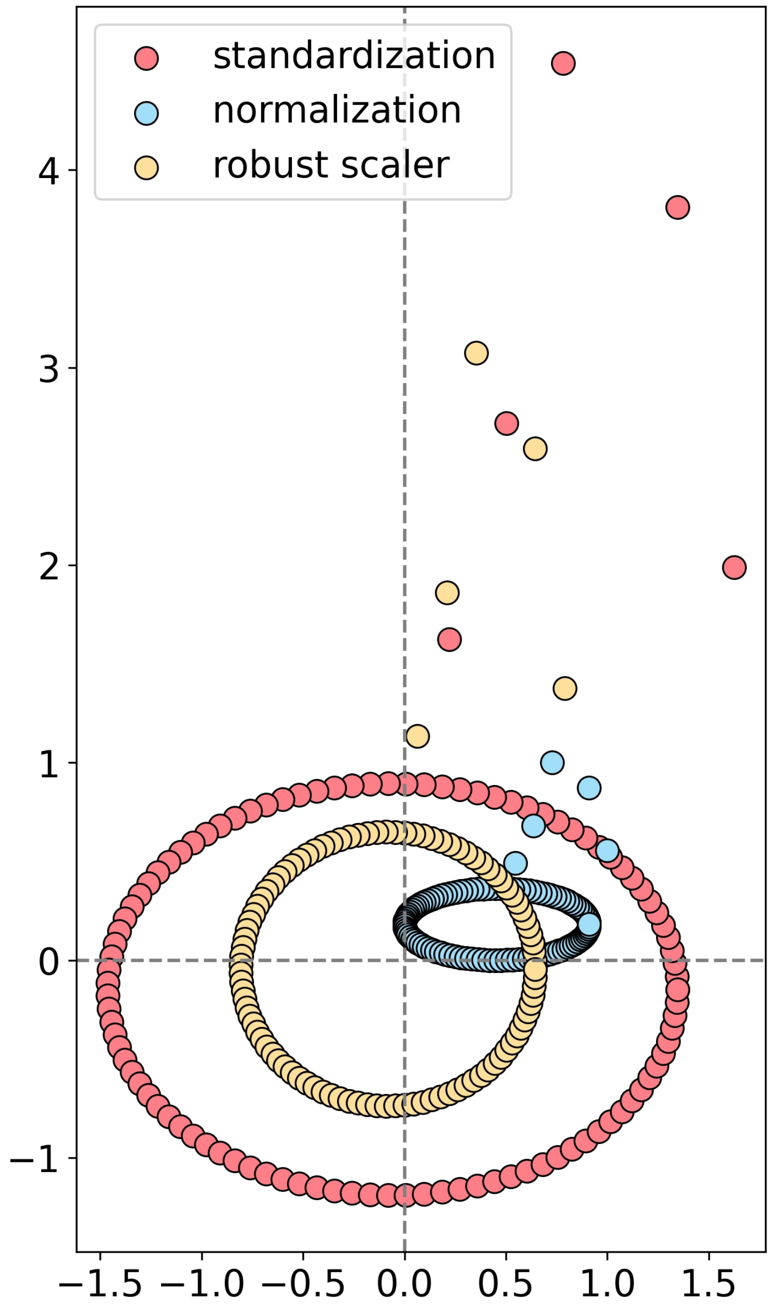
```
X_new = (X - mean)/Std
```

Robust Scaler

X_new = (X – median)/IQR

Data with Outliers





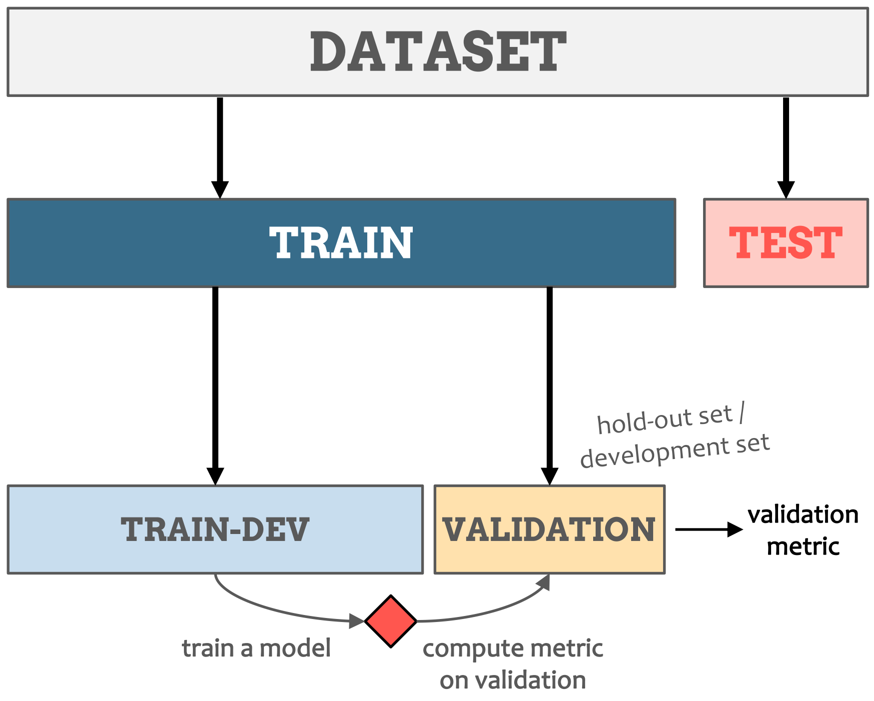
Normalization (Min-Max Scaling)	Standardization (Z-Score Normalization)	Robust Scaler
`X_new = (X - X_min)/(X_max - X_min)`	`X_new = (X - mean)/Std`	X_new = (X – median)/IQR
Minimum and maximum value of features are used for scaling	Mean and standard deviation is used for scaling.	Alternative to Standardization to be yet much less affected by **outliers.**
It is used when features are of different scales.	It is used when we want to ensure zero mean and unit standard deviation.	
Scales values between [0, 1] or [-1, 1].	It is not bounded to a certain range.	
It is really affected by outliers.	It is much less affected by outliers.	
Scikit-Learn provides a transformer called `MinMaxScaler` for Normalization.	Scikit-Learn provides a transformer called `StandardScaler` for standardization.	
This transformation squishes the n-dimensional data into an n-dimensional unit hypercube.	It translates the data to the mean vector of original data to the origin and squishes or expands.	
It is useful when we don't know about the distribution	It is useful when the feature distribution is Normal or Gaussian.	
It is a often called as Scaling Normalization	It is a often called as Z-Score Normalization.	
`sklearn.preprocessing`.MinMaxScaler	`sklearn.preprocessing`.StandardScaler	`sklearn.preprocessing`.RobustScaler

Model Validation

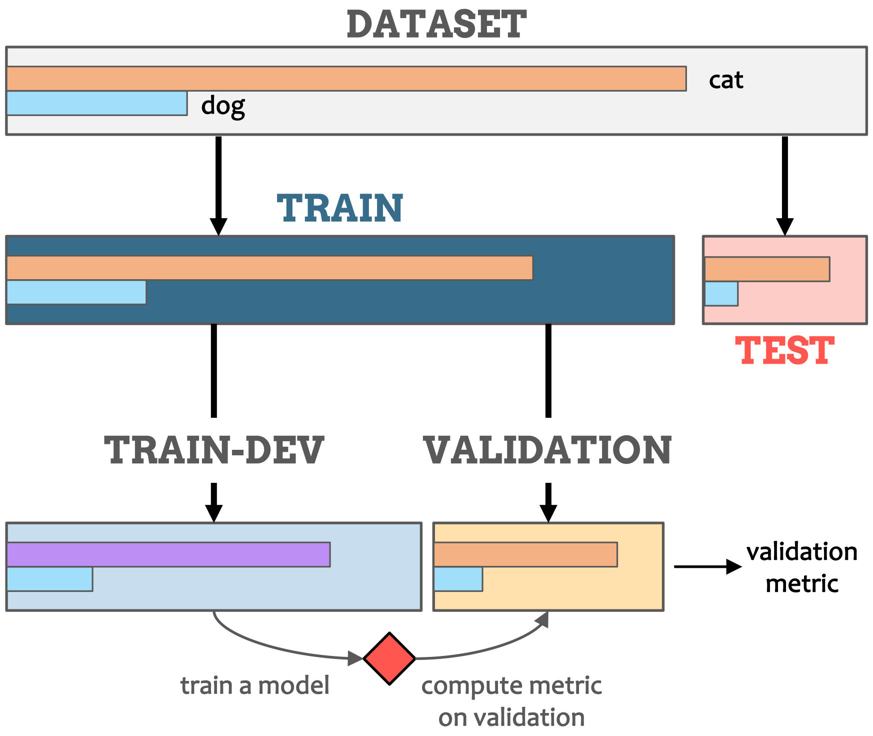
Model Validation

- Technique used to estimate model performance on **unseen data**.
- It is used to check if **our model** is overfitted, particularly in those cases where the **amount of data** may be **limited**.
- Two of the most popular strategies are the hold-out validation and the cross-validation (especially, **k-fold cross-validation)**.

Holdout Validation



Stratified Holdout Validation

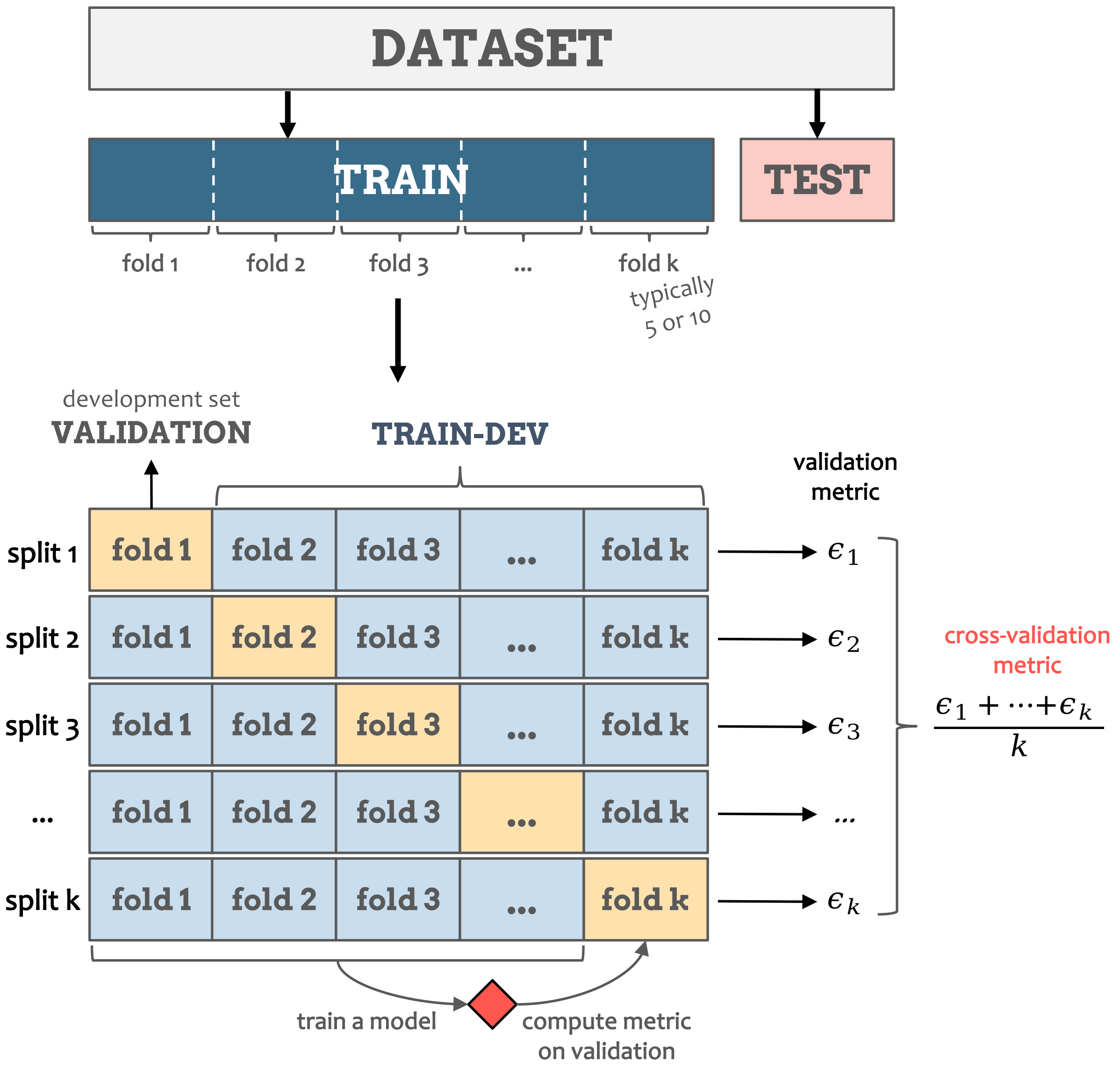


Rule of Thumb

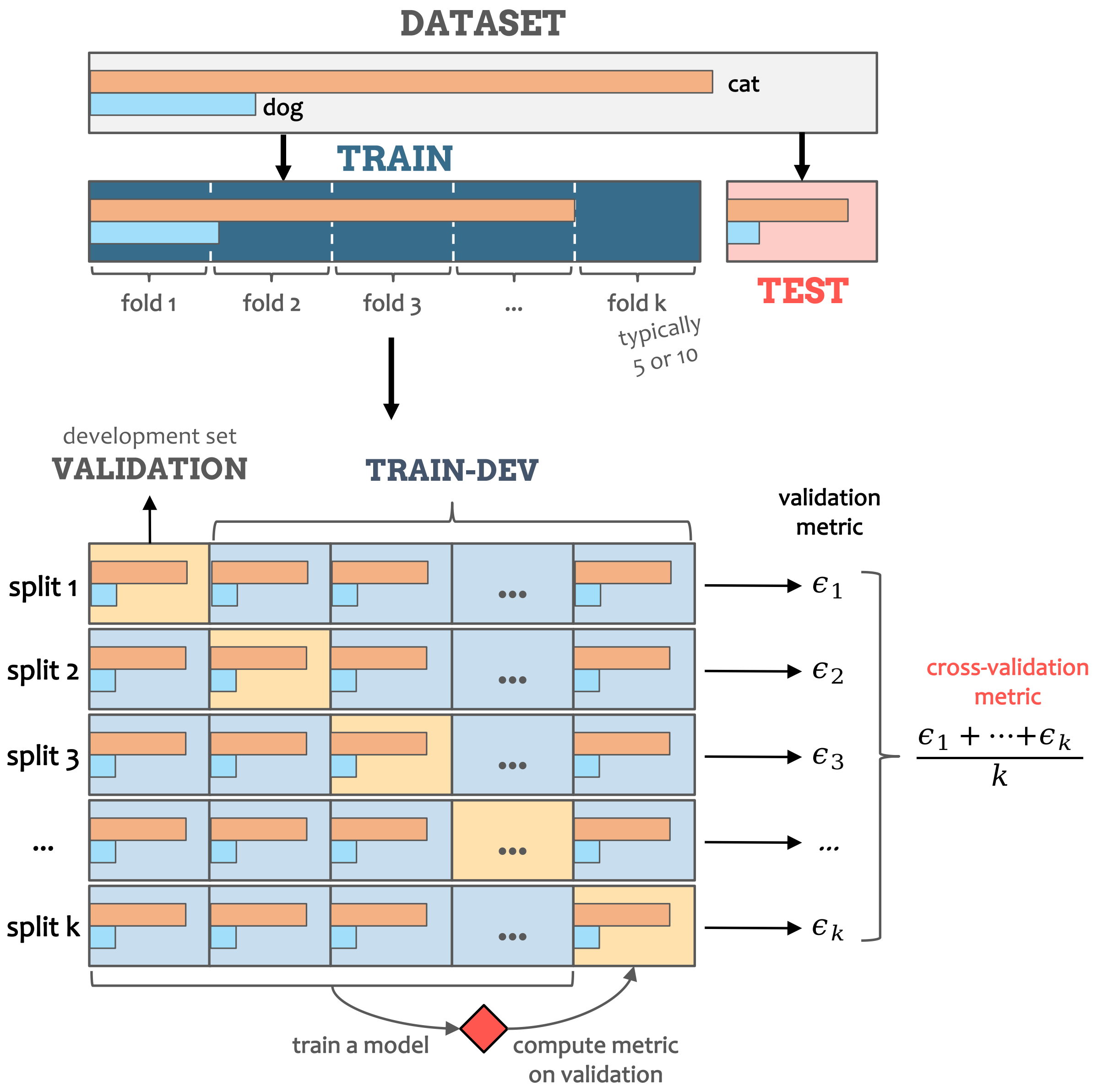
Train set: 80% of the dataset = 60% (train-dev set) + 20% (validation set)

Test set: 20% of the dataset

k-Fold Cross-Validation

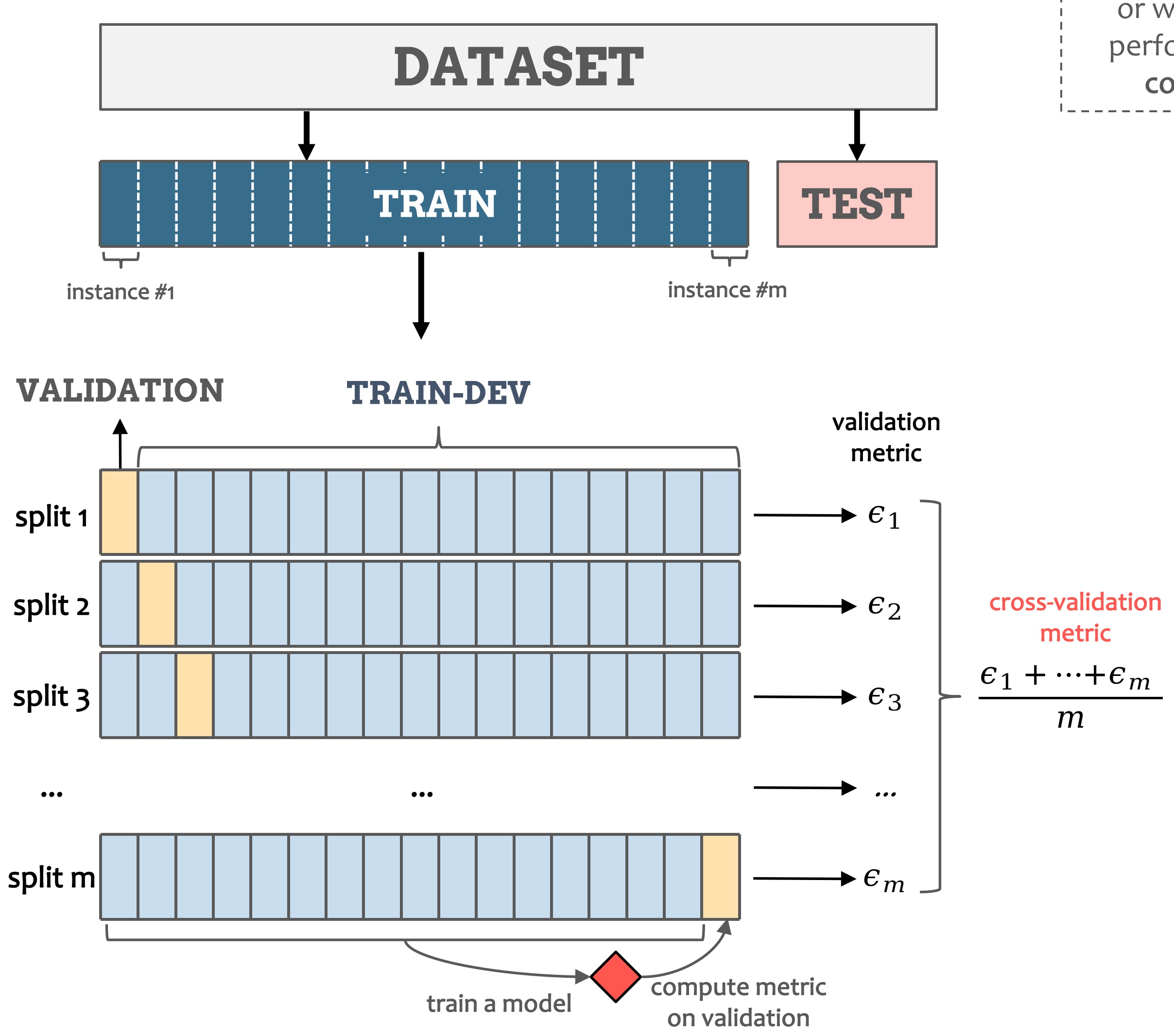


Stratified k-Fold Cross-Validation



Leave one out Cross-Validation

Appropriate when you have a small dataset or when an **accurate estimate** of model performance is **more important** than the **computational cost** of the method.



Holdout vs Cross-Validation

- Cross-validation is usually the preferred strategy for model validation because your model has the opportunity to train on **multiple train-dev sets**.
- This gives you a better indication of **how well** your model will perform on **unseen data**.

- Leave one out cross validation is appropriate when you have a small dataset or when an **accurate estimate** of model performance is **more important** than the **computational cost** of the method.

- Hold-out is dependent on just one **train-dev set** which make its score dependent on how the data is split into train-dev and validation sets.
- Hold-out is good to use when you have a **very large dataset**, you are on a time crunch, or you are starting to build **an initial model** in your project.

Aprendizado de Máquina e Reconhecimento de Padrões 2021.2

Machine Learning Concepts

Prof. Samuel Martins (Samuka)
samuel.martins@ifsp.edu.br

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
São Paulo
Campus Campinas