Data-Driven Problem Solving in Mechanical Engineering

Selected Topics

Masoud Masoumi

ME 364 - Spring 2022

Department of Mechanical Engineering
The Cooper Union for the Advancement of Science and Art

April, 2022

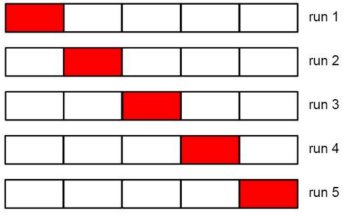
Cross-Validation

Cross Validation

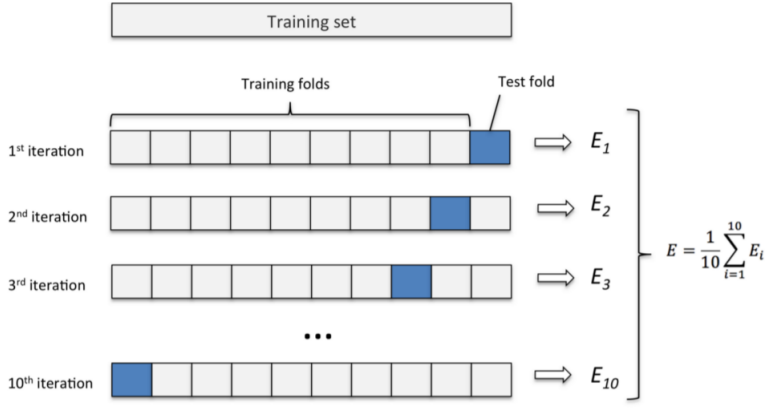
Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure works as follows:

- Shuffle the data set randomly.
- Split the data set into **k** groups
- For each unique group:
    (a) Take the group as a hold out or test data set
    (b) Take the remaining groups as a training data set
    (c) Fit a model on the training set and evaluate it on the test set
    (d) Retain the evaluation score and discard the model
- Summarize the model performance using the evaluation scores



Cross Validation

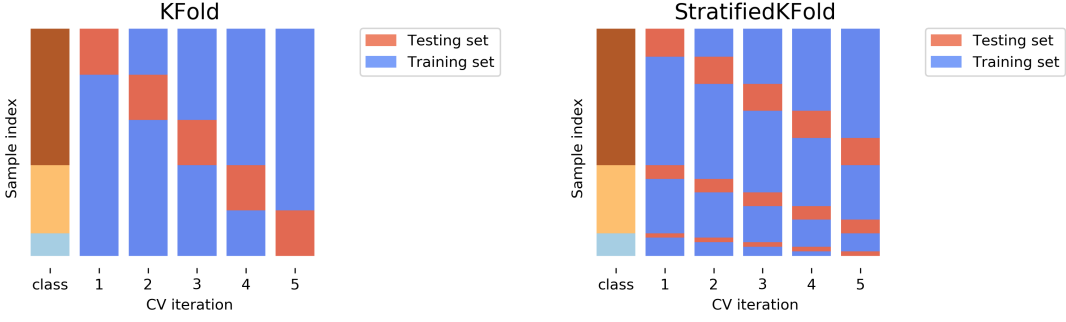


Remember: The purpose of cross-validation is to find the best estimate possible of the model ability to learn and predict. when you do K-fold cross validation, you are testing how well your model is able to get trained by some data and then predict data it hasn't seen.

Cross Validation for Unbalanced Data

When using KFold approach, the data is split into k-folds with a uniform probability distribution.

This might work fine for data with a balanced class distribution, but when the distribution is severely skewed, it is likely that one or more folds will have few or no examples from the minority class.



Ref: https://amueller.github.io/aml/04-model-evaluation/1-data-splitting-strategies.html

Hyperparameter Tuning

Hyperparameter Tuning

- **Model parameters:** These are the parameters that are estimated by the model from the given data. For example the weights (coefficients) in a linear regression model.

- **Model hyperparameters**: These are the parameters that cannot be estimated by the model from the given data. These parameters are used to estimate the model parameters. For example, 'K' in K-nearest neighbors, 'C' and 'Gamma' or 'kernel' type in an SVM model, numerical solver options, regularization methods, etc

Hyperparameter tuning is the process of determining the right combination of hyperparameters that allows the model to maximize model performance.

There are two common ways for hyperparameter tuning using scikit-learn library:

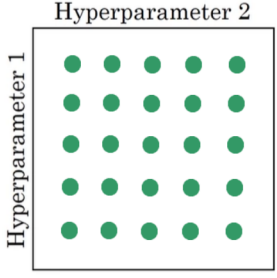
(1) **Grid Search**

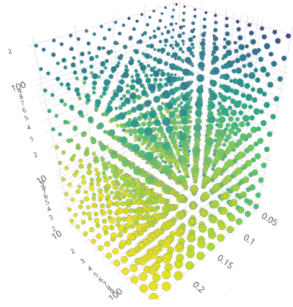
(2) **Random Search**

Hyperparameter Tuning - Grid Search

Here is how it works:

- We create a grid of possible values for hyperparameters.
- We fit the model on each and every combination of hyperparameter possible and record the model performance.
- Finally, the search returns the best model with the best hyperparameters.





Ref: https://www.andreaperlato.com/aipost/hyperparameters-tuning-in-ai

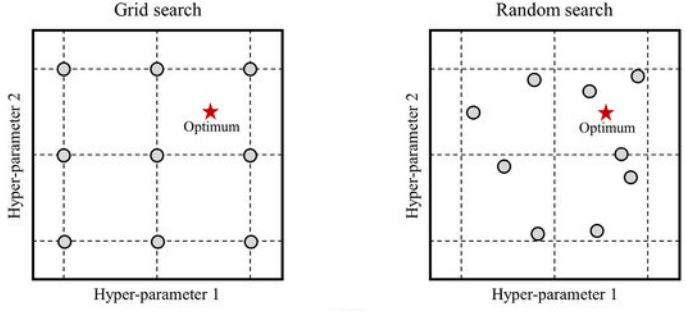
Hyperparameter Tuning - Random Search

Here is how it works:

- We create a grid of possible values for hyperparameters.

- We fit the model on random combination of hyperparameters from this grid and record the performance.

- Finally, the search returns the combination of hyperparameters which provided the best performance.



Deep learning-based phase prediction of high-entropy alloys: optimization, generation, and explanation by Lee, Soo Young, et al.

Feature Selection

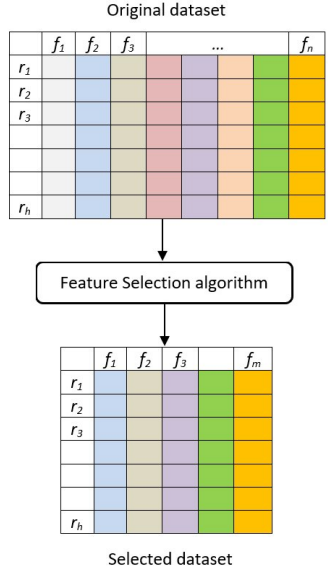
Feature Selection

The features in your data will directly influence the predictive models you use and the results you can achieve.

You can say that: the better the features that you prepare and choose, the better the results you will achieve.

Which features should you use to create a predictive model? This is a difficult question that may require deep knowledge of the problem domain.

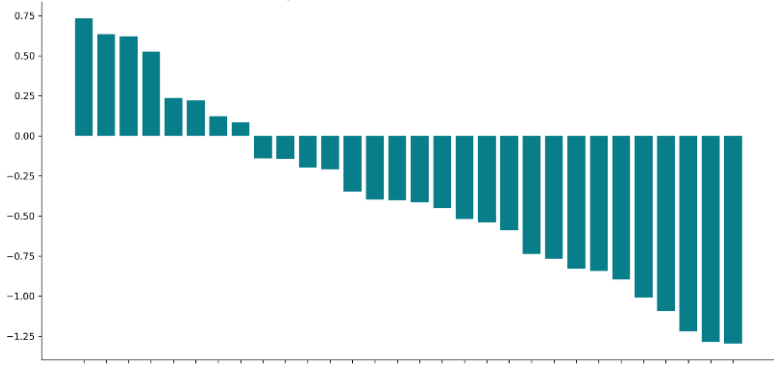
It is possible to automatically select those features in your data that are most useful or most relevant for the problem you are working on. This is a process called **feature selection**.



Feature Selection - Feature Importance

Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable.

We fit a model developed using a algorithm for regression or classification and then we look at the importance or effect that each feature has in predicting the target variable.



Ref: https://betterdatascience.com/feature-importance-python/

Feature Selection - RFE

Recursive Feature Elimination (RFE) works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains.

There are two important configuration options when using RFE:

- Number of features to select
- The algorithm used to help choose features

In its most straightforward implementation, here is how it works:

- It fits the given machine learning algorithm
- It ranks the features by importance and discards the least important features
- It then re-fits the model.
- This process is repeated until a specified number of features remains.