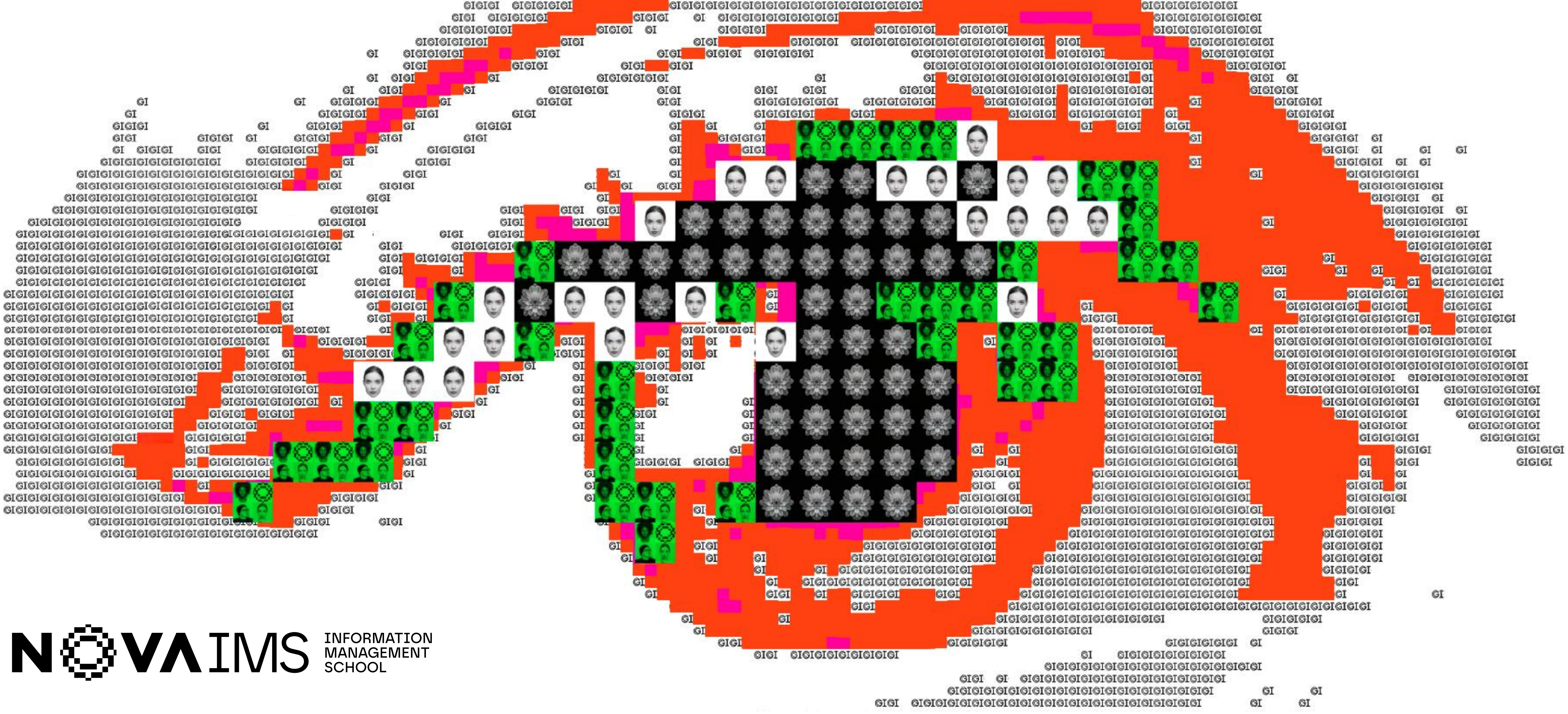
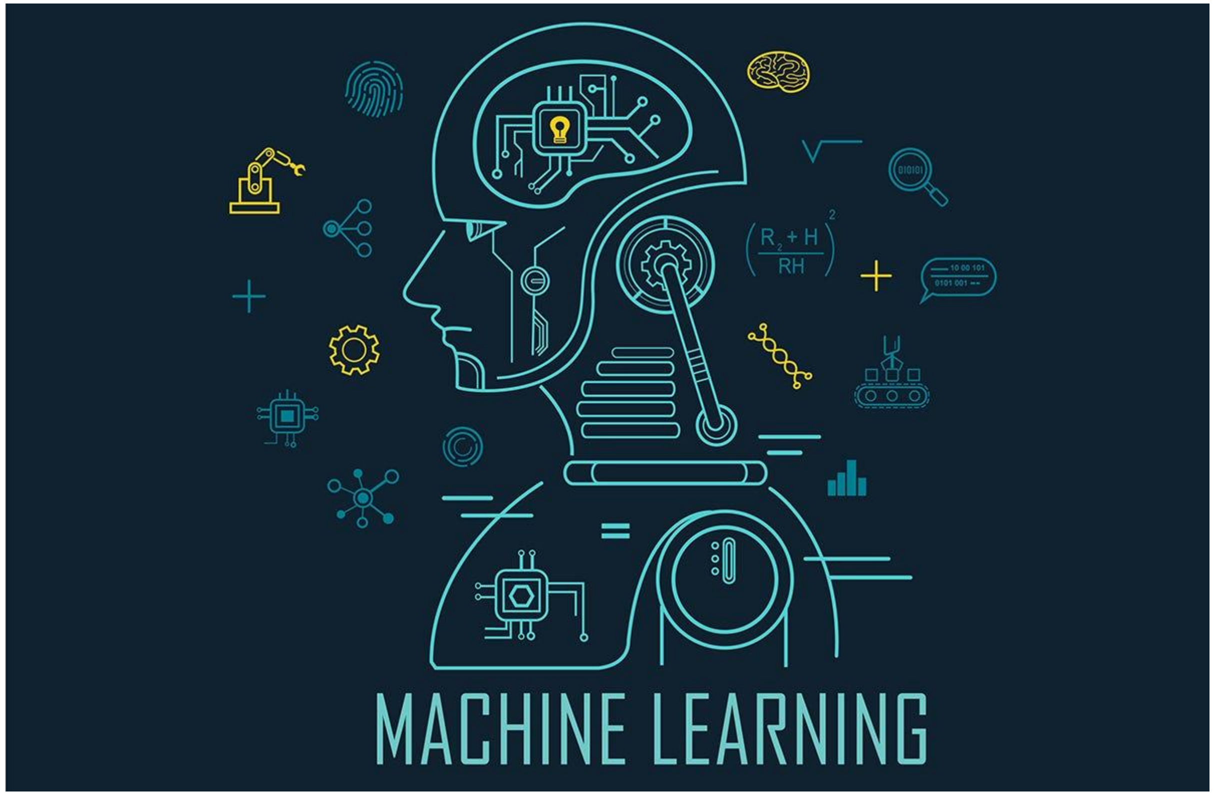


Machine Learning in Finance

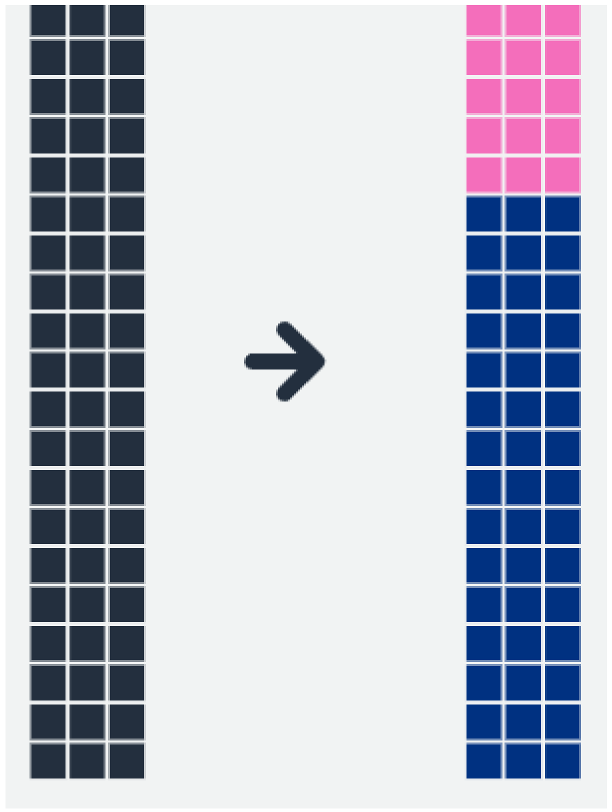
- Prof. Ian J. Scott



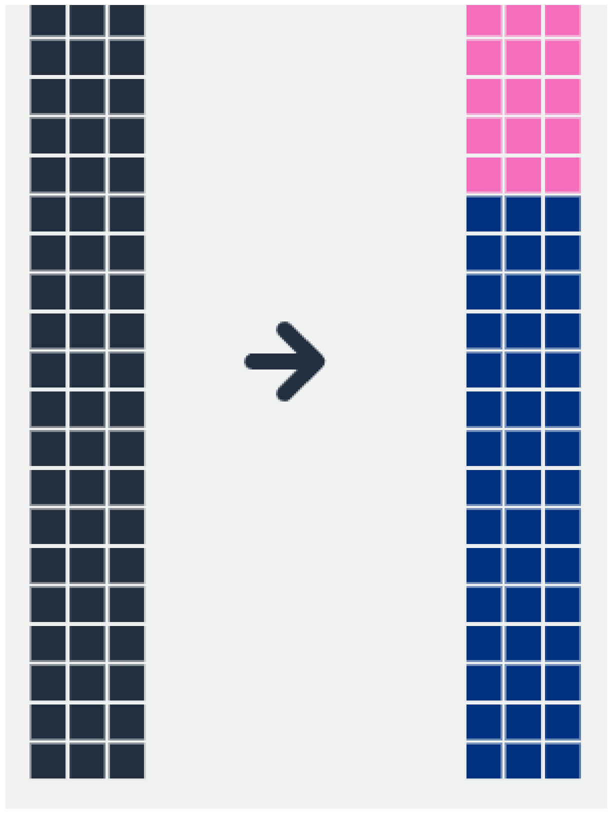
Resampling

The training test split.

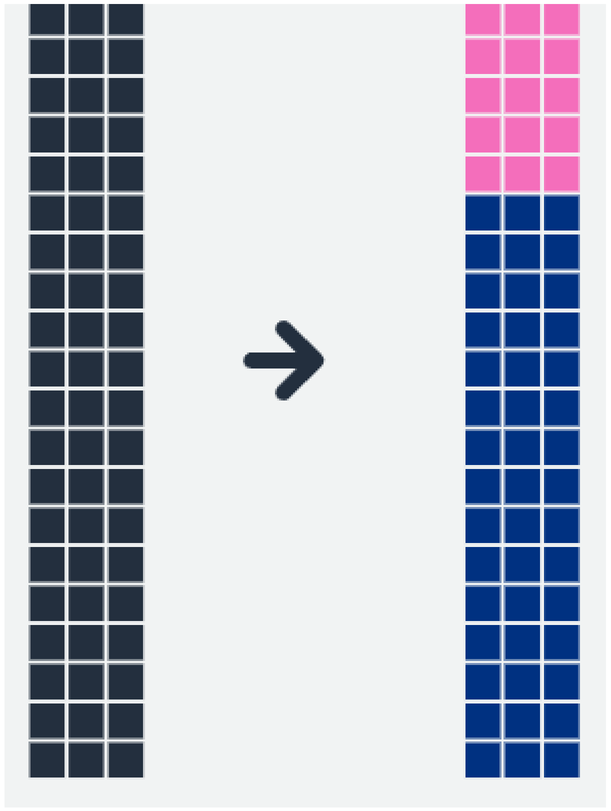
- Simple
- **Important**



- How do we decide which of the models we know works best on our dataset?
- How do we pick the value of K in our K nearest neighbours' algorithm?
- **How do we know the final performance of our model?**



- How do we decide which of the models we know works best on our dataset? **– Model selection**
- How do we pick the value of K in our K nearest neighbours' algorithm? **– Hyperparameter tuning**
- **How do we know the final performance of our model? – Model evaluation**

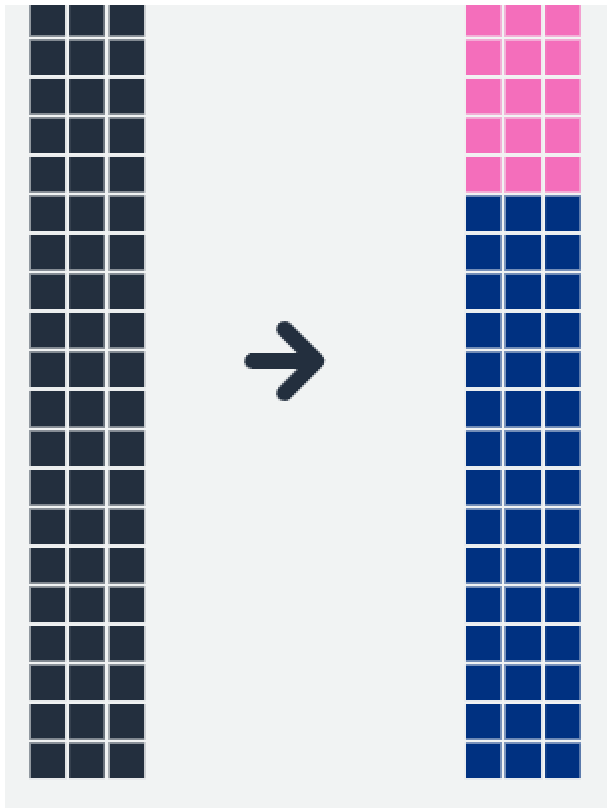


Should I:

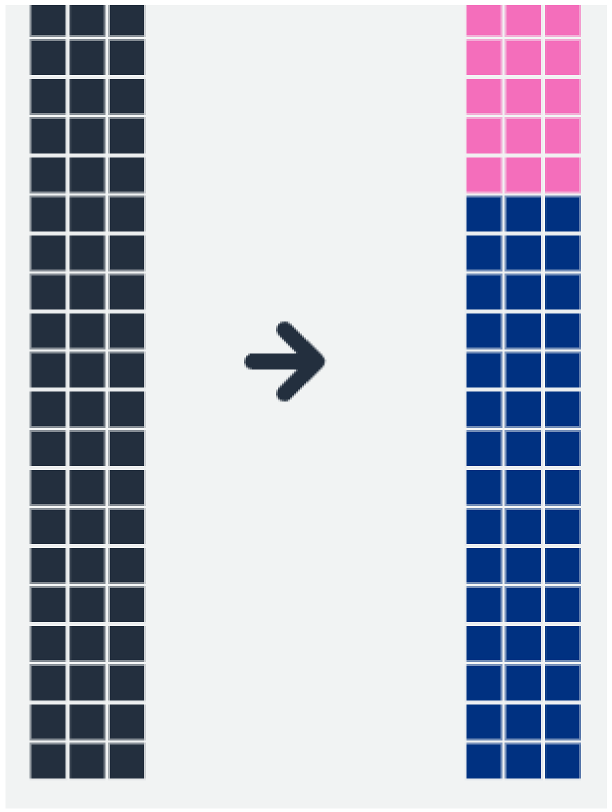
- **1. Perform model selection.** Compare models, find the best model / classifier.
- **2. Perform hyperparameter tuning.** Then with this classifier perform hyperparameter tuning to get the best version of this classifier?

Or should I:

- **1. Perform hyperparameter tuning.** Apply hyper parameter tuning on all my models.
- **2. Perform model selection.** Compare models, find the best model / classifier.



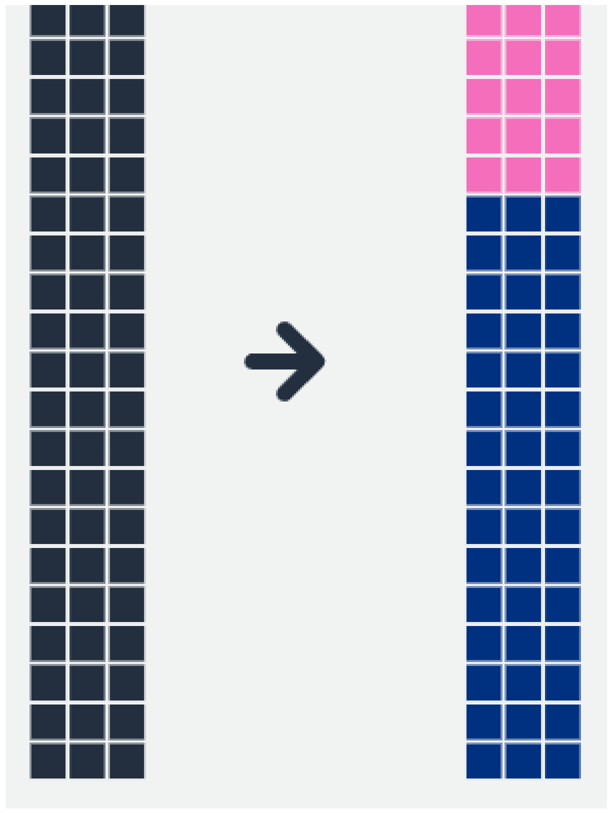
It’s clear what is theoretically better. In practice we usually end up with something of a hybrid.



With our data we have two tasks we want to perform:

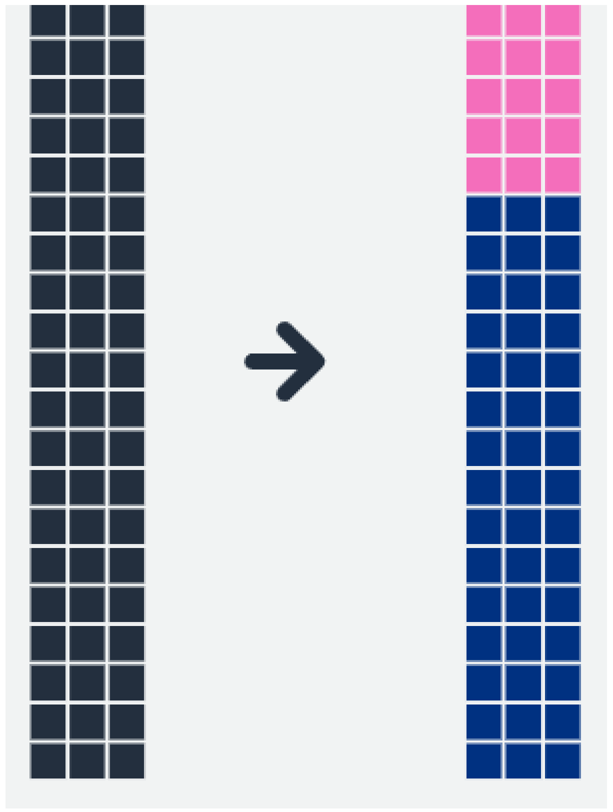
- Compare models and pick the best one
- Estimate the ability of our model

These two tasks are incompatible with the same split.



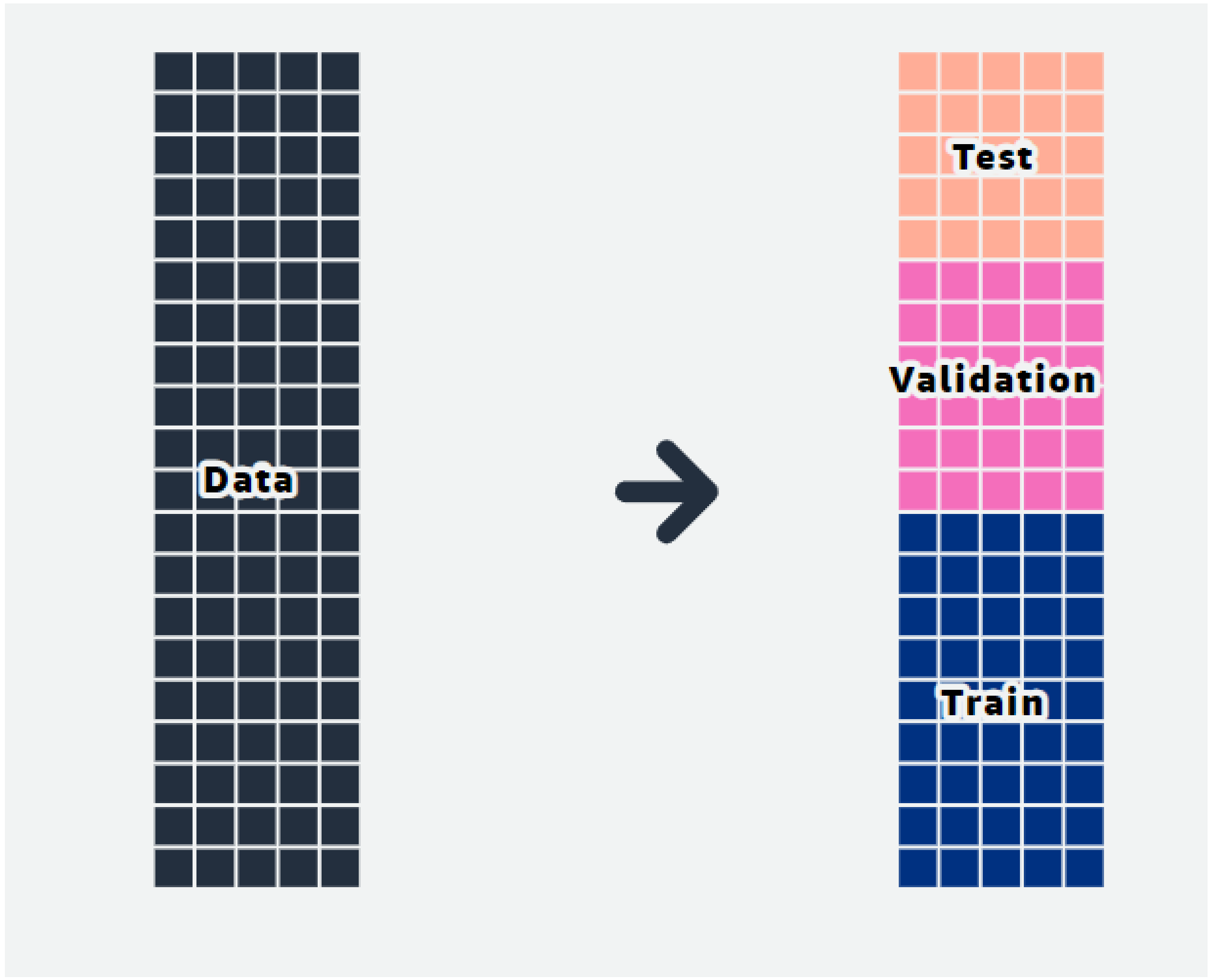
Previously we were using the test data for both of these tasks, but this is a little unfair, because we pick the model that performs best on the test dataset so of course it will have good performance on this dataset.

This undermines our ability to understand the performance of our model on new data!



Now we want to split our data into three:

- Training
- Validation – Compare models or parameters
- Test – Completely unseen benchmark of model



To clarify the process is the following:

Start as normal

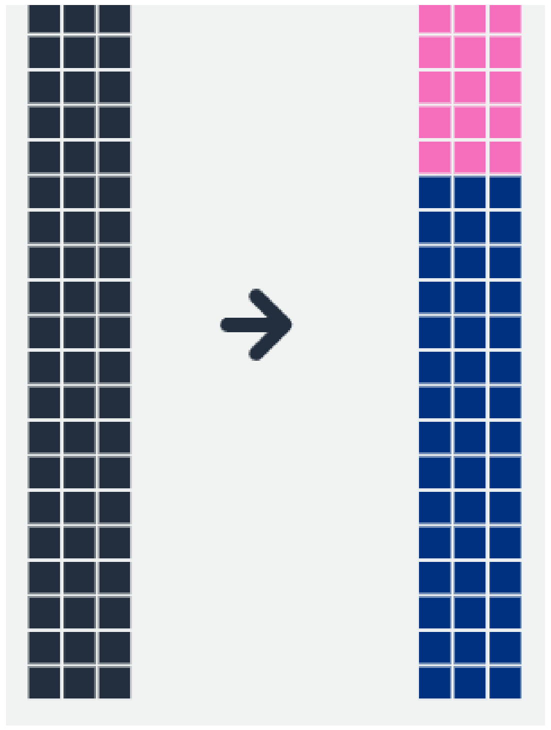
1. Split Data into test and training as usual

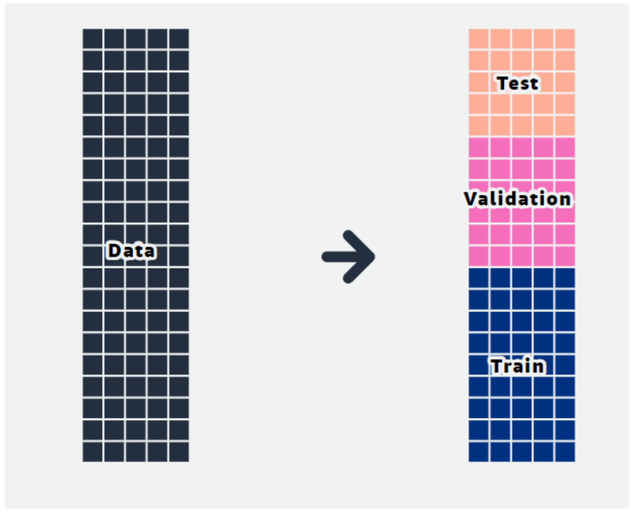
Within the training data

2. Split the training data into "validation" and "train".
3. Fit models on "train".
4. Compare on validation. Can also check different model parameters **hyper parameter tuning.**
5. Pick the best model
6. Fit the best model on all the training (train + validation) data.

With the test data:

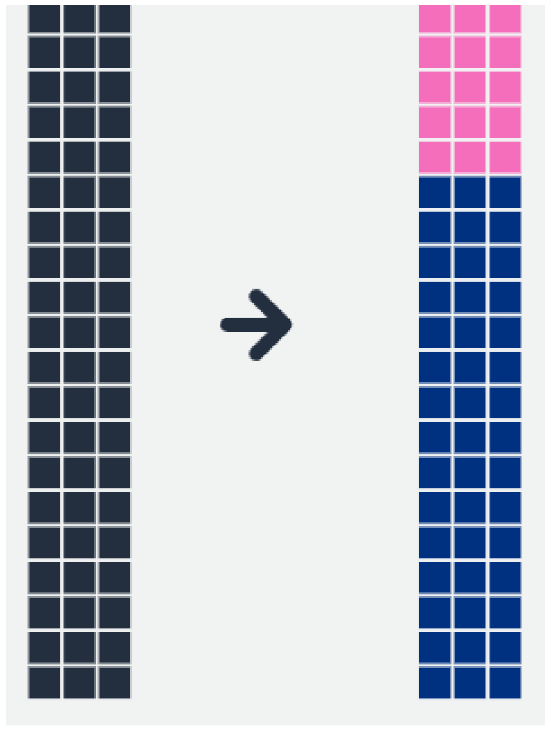
7. Assess the model on the test set.

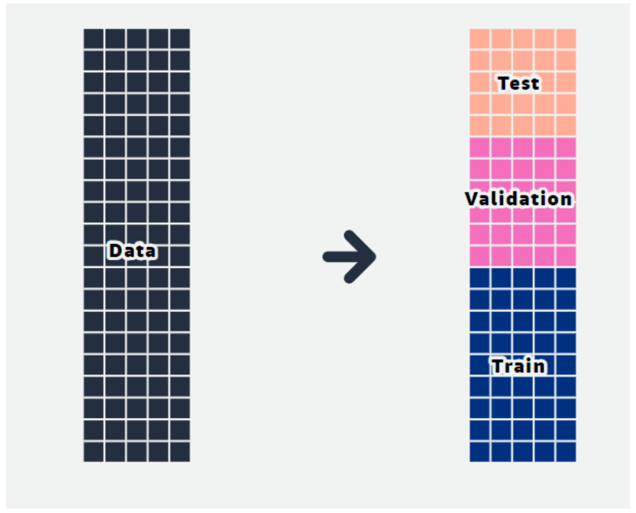




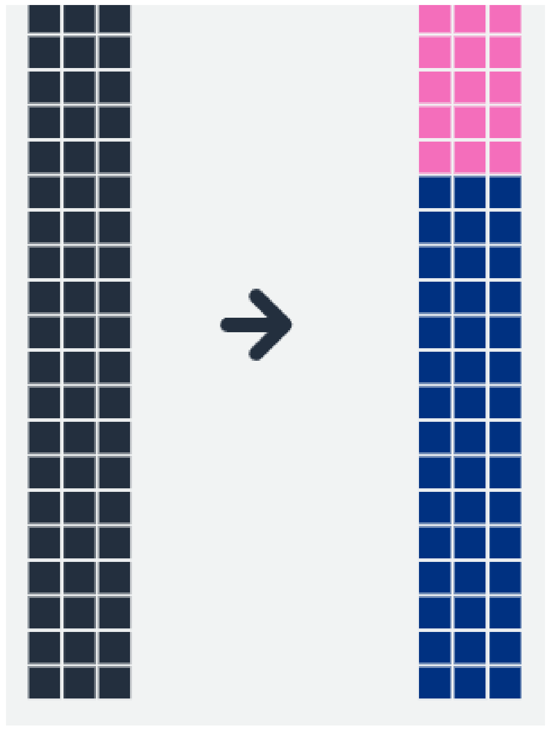
Hyperparameter tuning is a critical step in the machine learning pipeline.

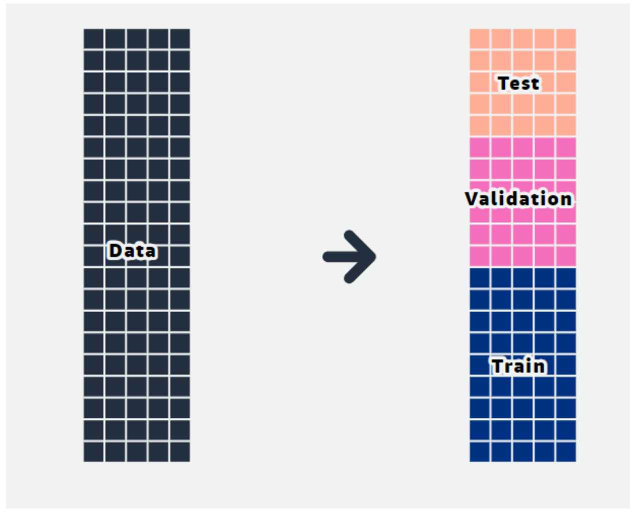
- **Hyperparameters** are external configurations of the model that cannot be learned from the data. Like “k” in K Nearest Neighbors.
- Hyperparameters are essentially the settings or configurations of the model that are set prior to training. They are not learned from the data but are rather **set by the practitioner.**
- The right choice of hyperparameters can significantly improve the performance of a model.





More advanced ML models often have multiple hyperparameters to parameters to set.





Support Vector Machines (SVM)

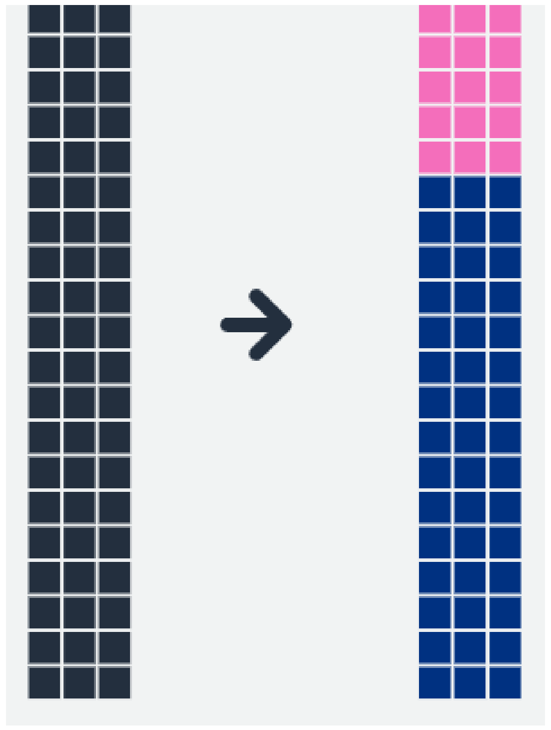
- Kernel Type: Determines the type of hyperplane used to separate the data. Common kernels are linear, polynomial, radial basis function (RBF), and sigmoid.
- C (Regularization Parameter): Controls the trade-off between achieving a low error on the training data and minimizing the norm of the weights.
- Gamma: Used in the RBF kernel; it defines how far the influence of a single training example reaches.

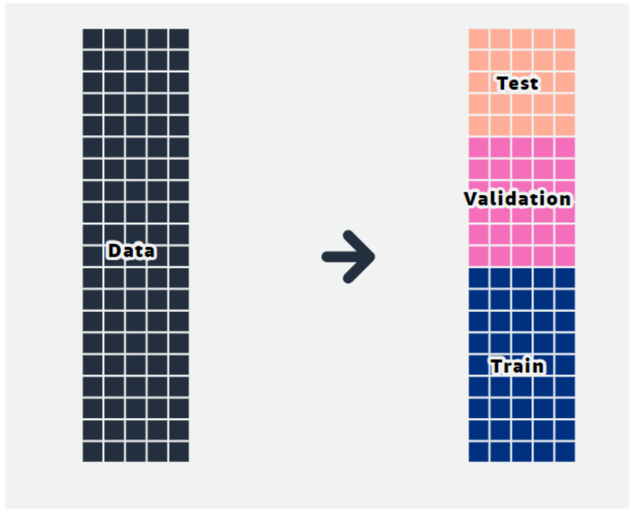
Decision Trees and Random Forests

- Max Depth: The maximum depth of the tree. Deeper trees can model more complex patterns but might lead to overfitting.
- Min Samples Split: The minimum number of samples required to split an internal node.
- Max Features: The number of features to consider when looking for the best split.
- Min Samples Leaf: The minimum number of samples required to be at a leaf node.

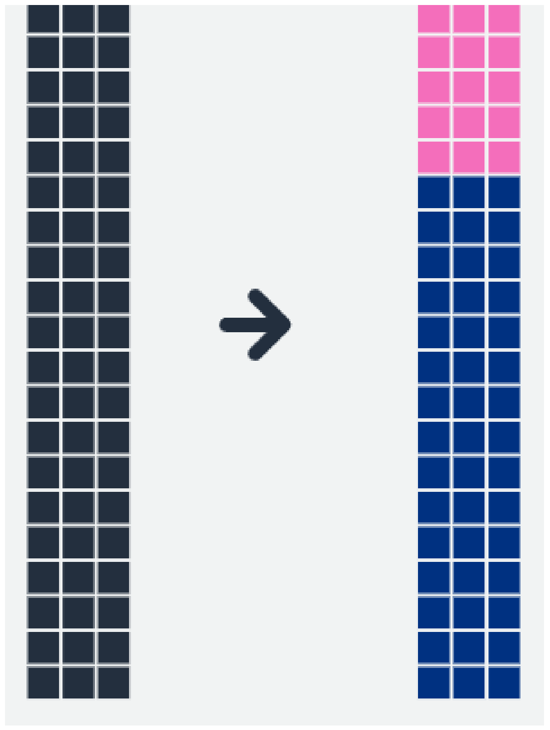
Gradient Boosting Machines (GBM)

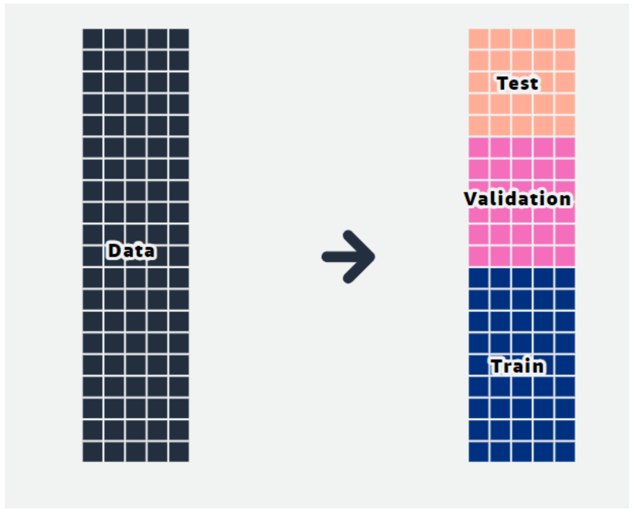
- Learning Rate: Similar to neural networks, it controls the contribution of each tree to the final outcome.
- N_estimators: The number of sequential trees to be modeled.
- Subsample: The fraction of samples to be used for fitting the individual base learners.



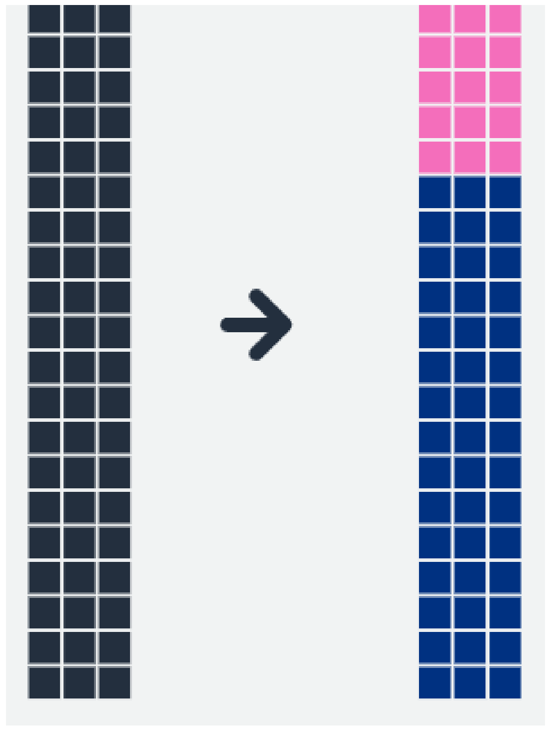


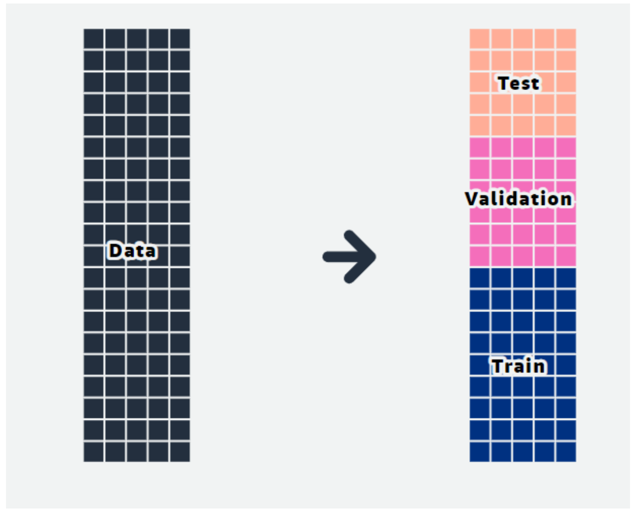
We need a search strategy for searching possible parameter combinations.





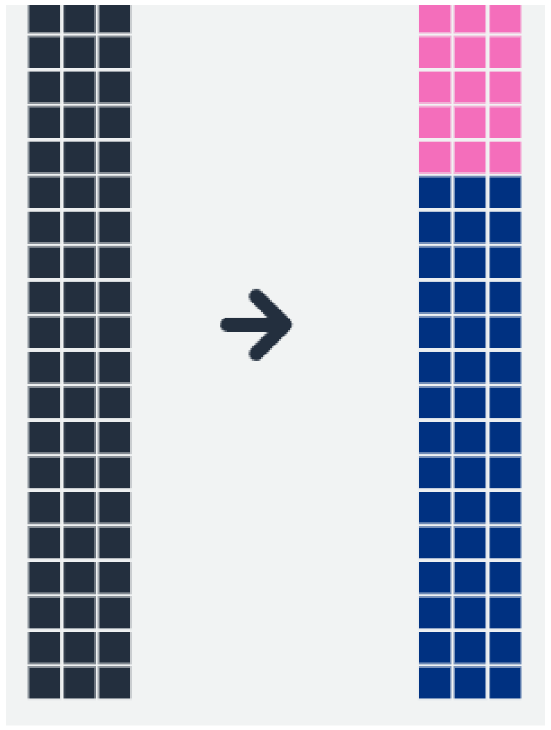
- **Grid Search -** Grid search is a brute-force approach that exhaustively tries all possible combinations of hyperparameters specified in a grid.
    - **Halving grid search -** Experimental
- **Random Search -** Random search randomly selects combinations of hyperparameters to evaluate.

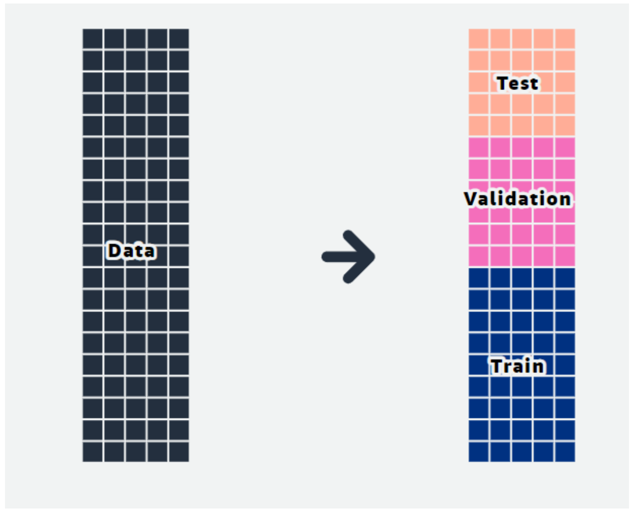




Advanced

- **Bayesian Optimization -** Bayesian optimization uses past evaluation results to choose the hyperparameters that are most likely to yield the best performance.
- **Optuna –** library for hyperparameter tuning.
- **Gradient-Based Optimization -** Some algorithms allow for gradient-based optimization of hyperparameters.





However, more problems –

- the validation estimate of the test error rate can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set.

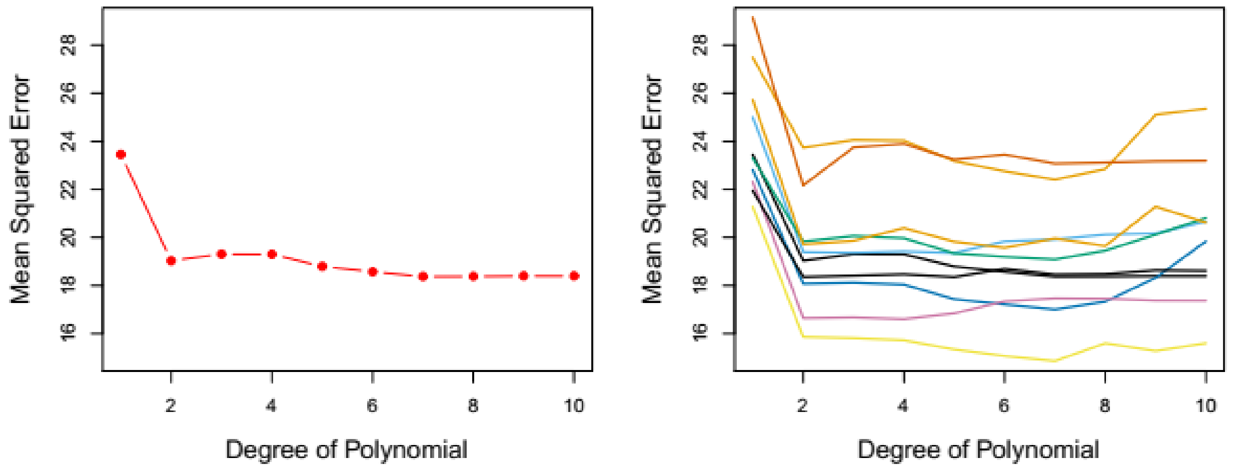
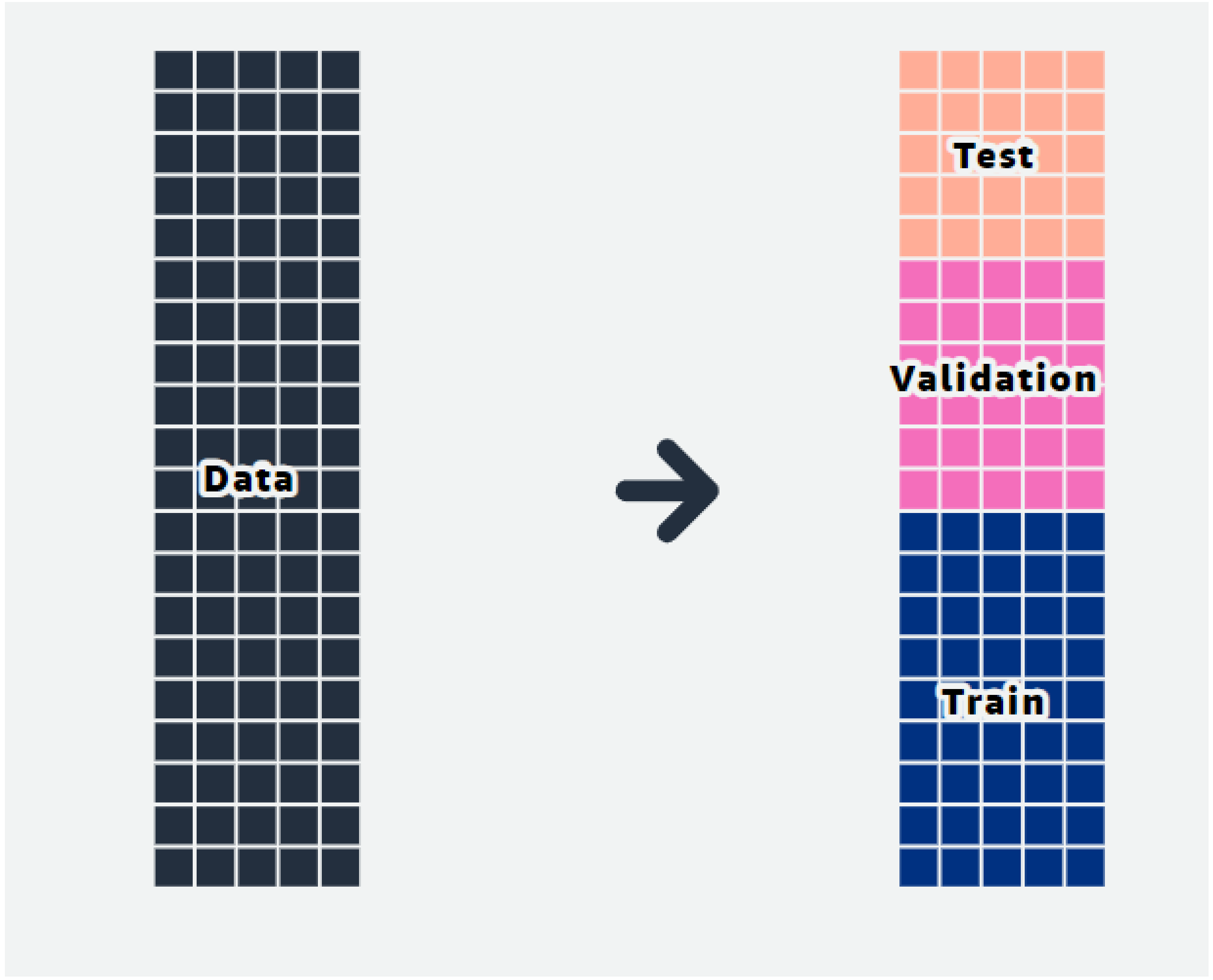
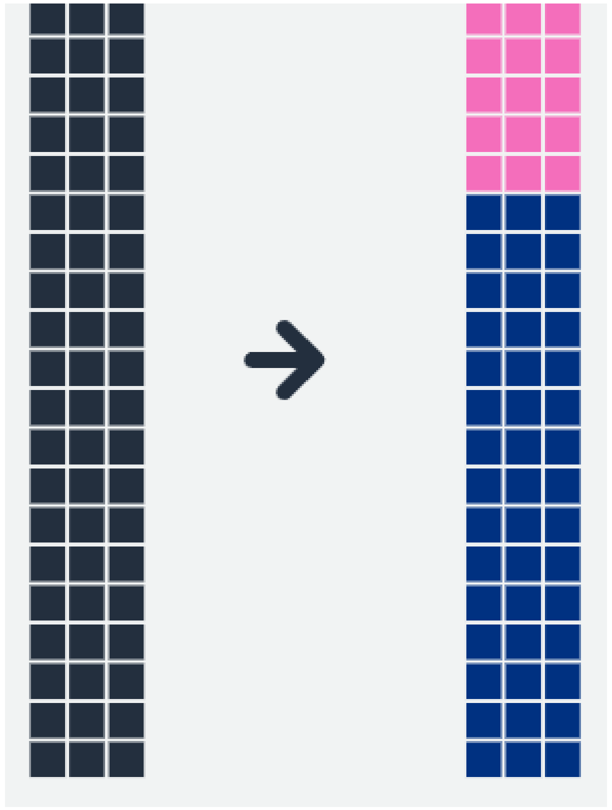


FIGURE 5.2. *The validation set approach was used on the* Auto *data set in order to estimate the test error that results from predicting* mpg *using polynomial functions of* horsepower. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

- In the validation approach, only a subset of the observations—those that are included in the training set rather than in the validation set—are used to fit the model. Since statistical methods tend to perform worse when trained on fewer observations, this suggests that the validation set error rate may tend to overestimate the test error rate for the model fit on the entire data set.

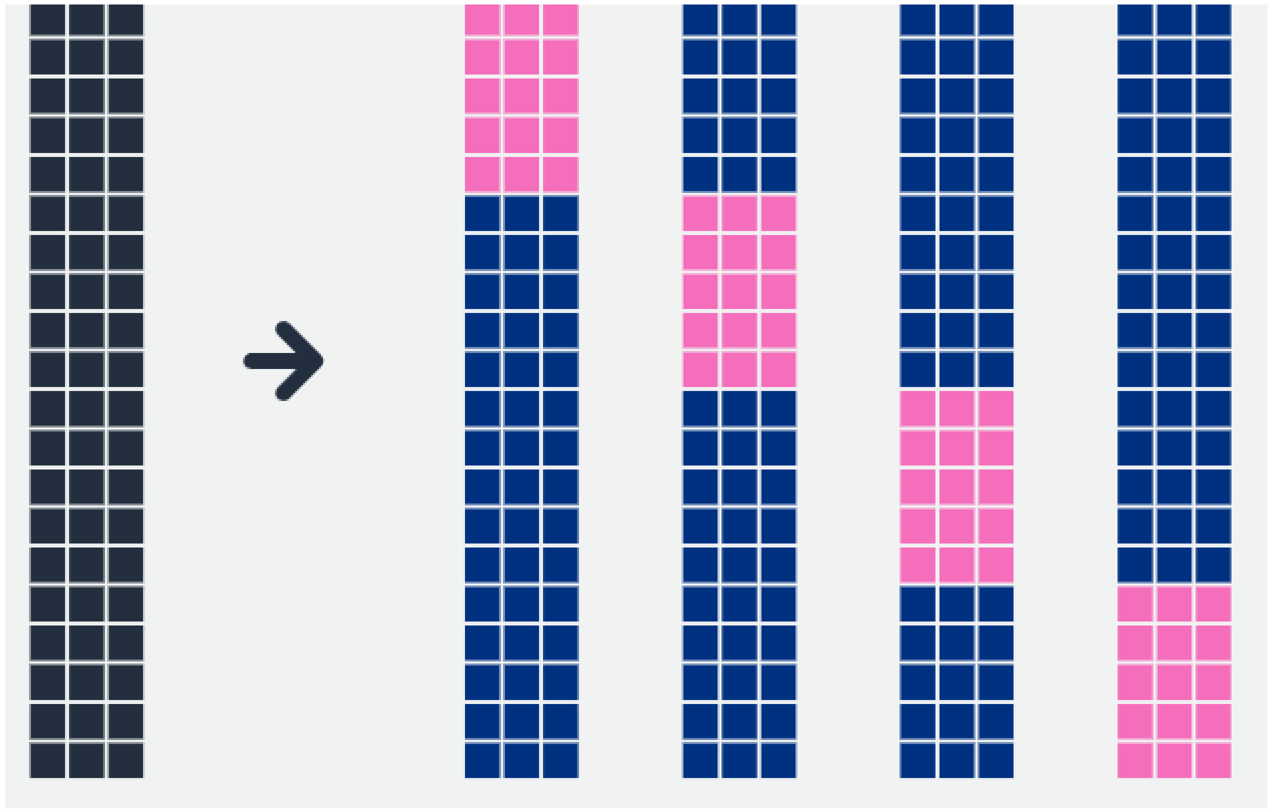


Resampling methods are an indispensable tool in modern statistics. They involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model.



Resampling methods are an indispensable tool in modern statistics. They involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model.

In this cotext we call the resampling and assessment process **cross-validation (CV).**

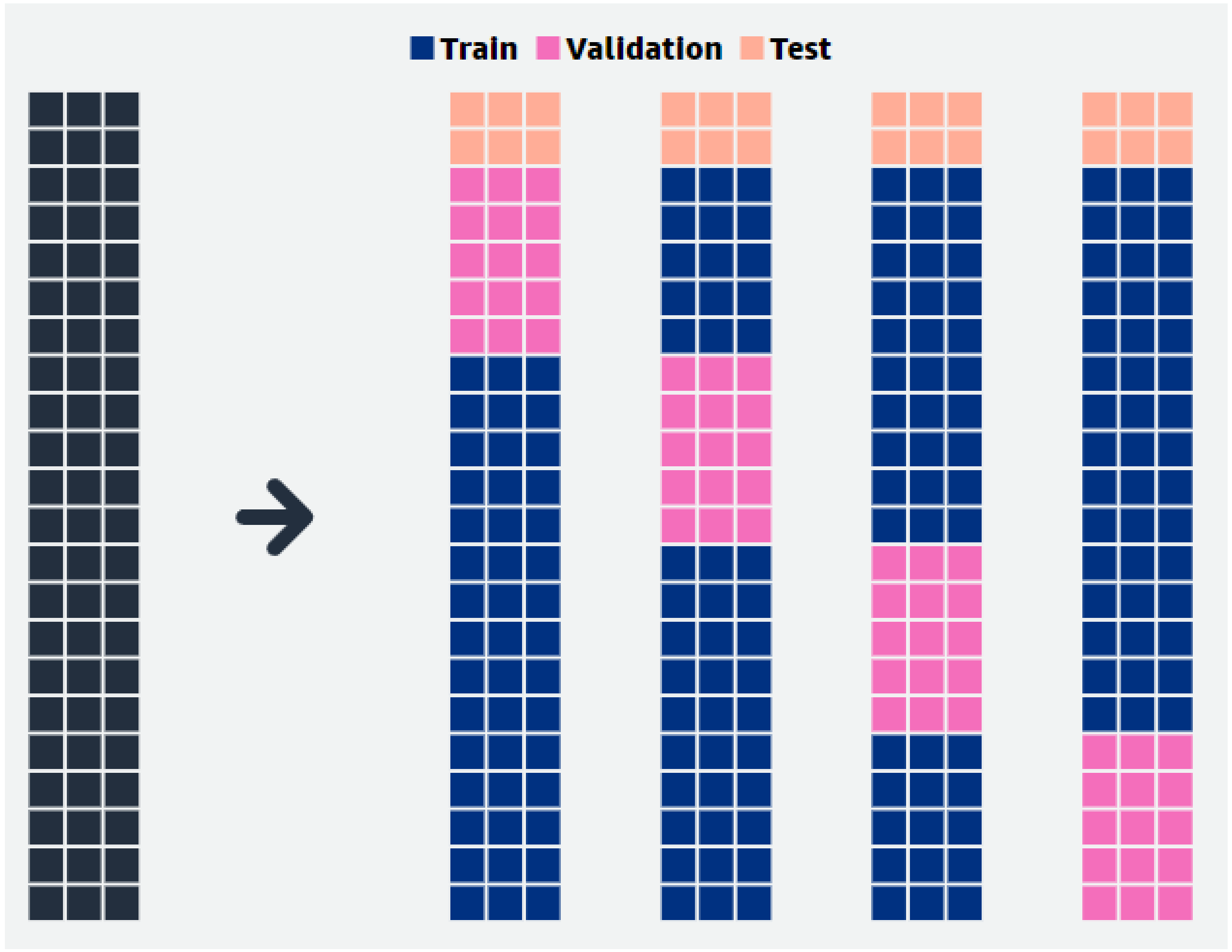


Using CV:

We separate -

- model assessment - The process of evaluating a model's performance
- model selection - the process of selecting the proper level of flexibility for a model.

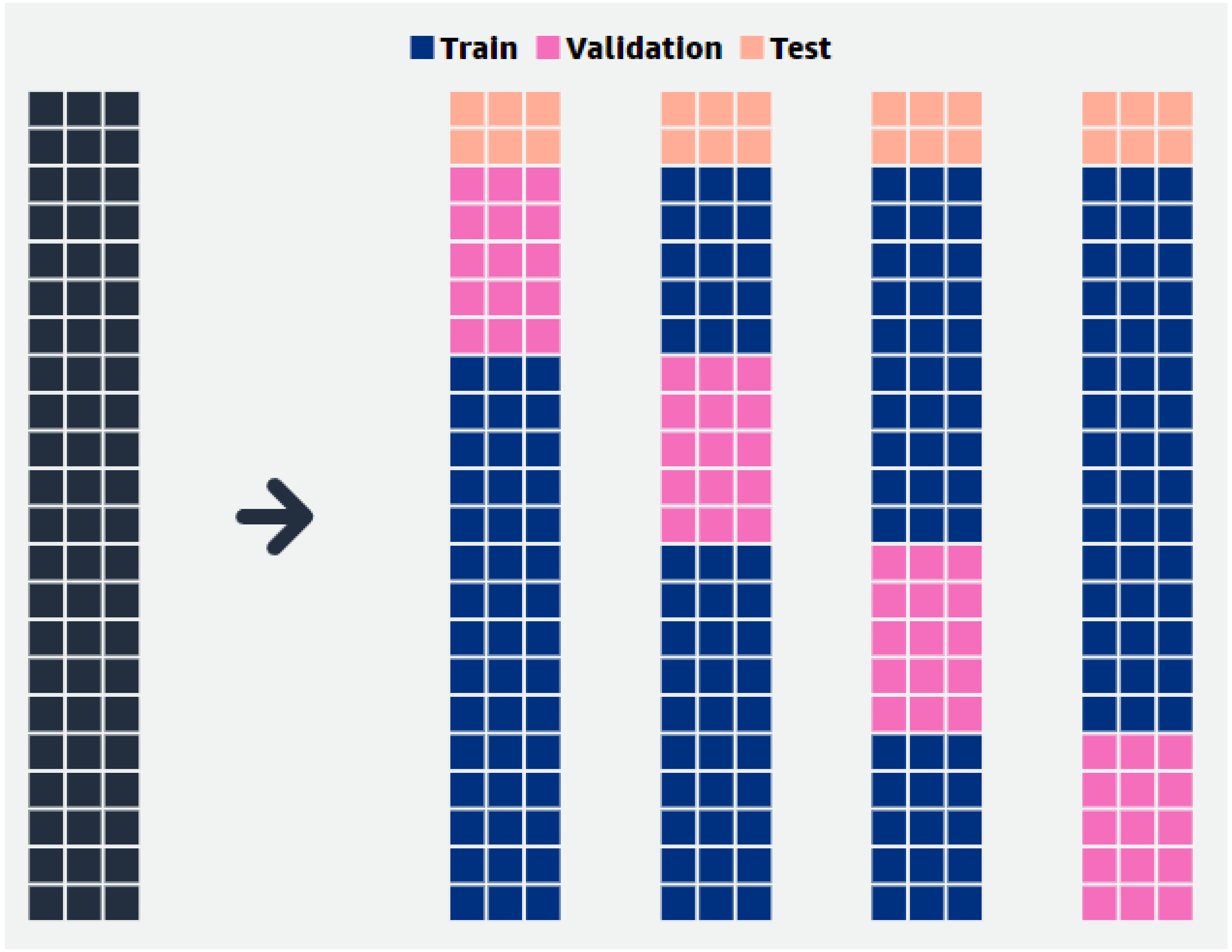
We improve the variability of model selection.



K-fold cross validation help us address these issues.

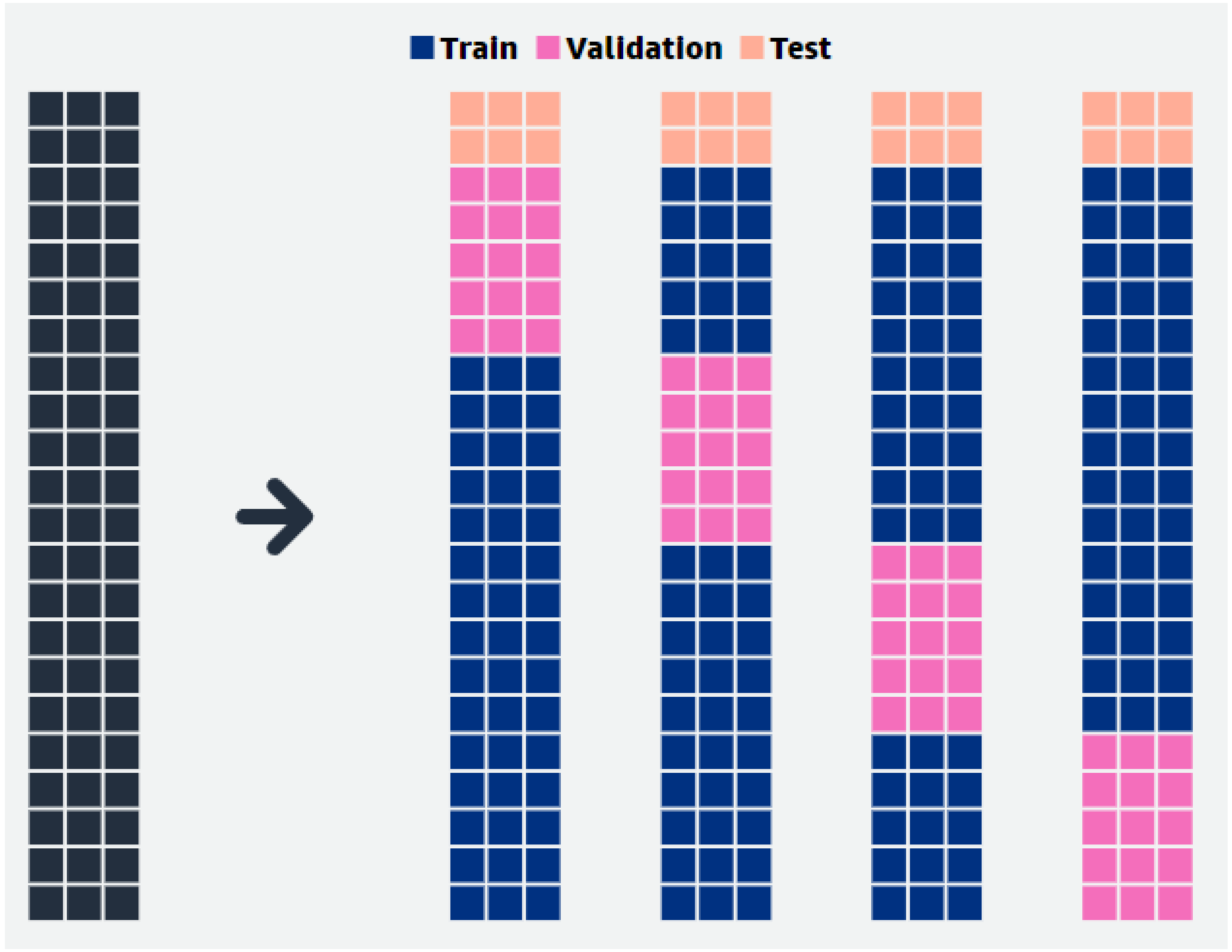
This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size.

The mean squared error, MSE1, is then computed on the observations in the held-out fold. This procedure is repeated k times; each time, a different group of observations is treated as a validation set. This process results in k estimates of the test error, MSE1, MSE2, . . . , MSEk. The k-fold CV estimate is computed by averaging these values.

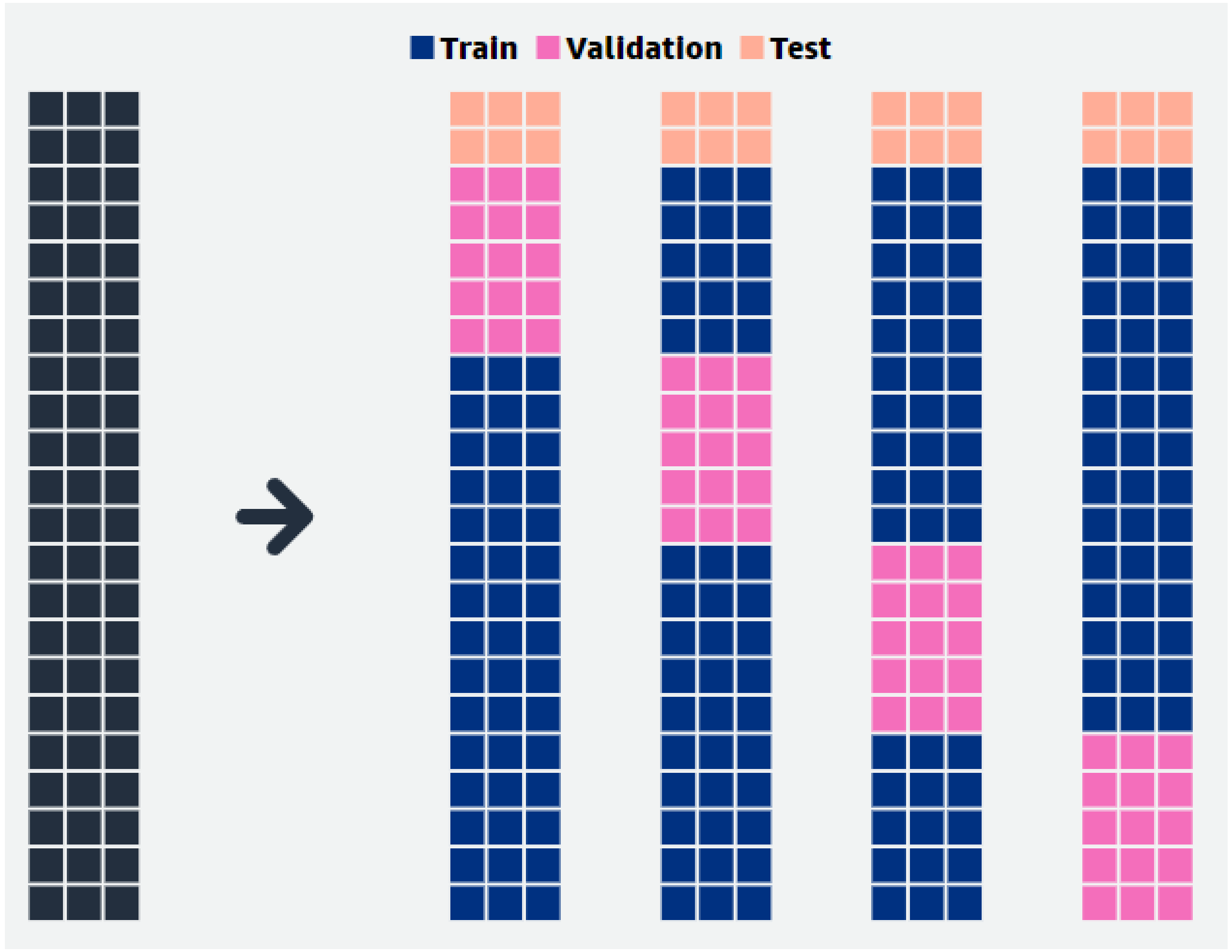


$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i.$$

The main benefit is that, because we train our model on multiple subsets of our data and take the average of the evaluation scores on those subsets, **our evaluation estimates from K-Fold Cross-Validation will have lower variance** than will the evaluation estimates from the validation set approach.

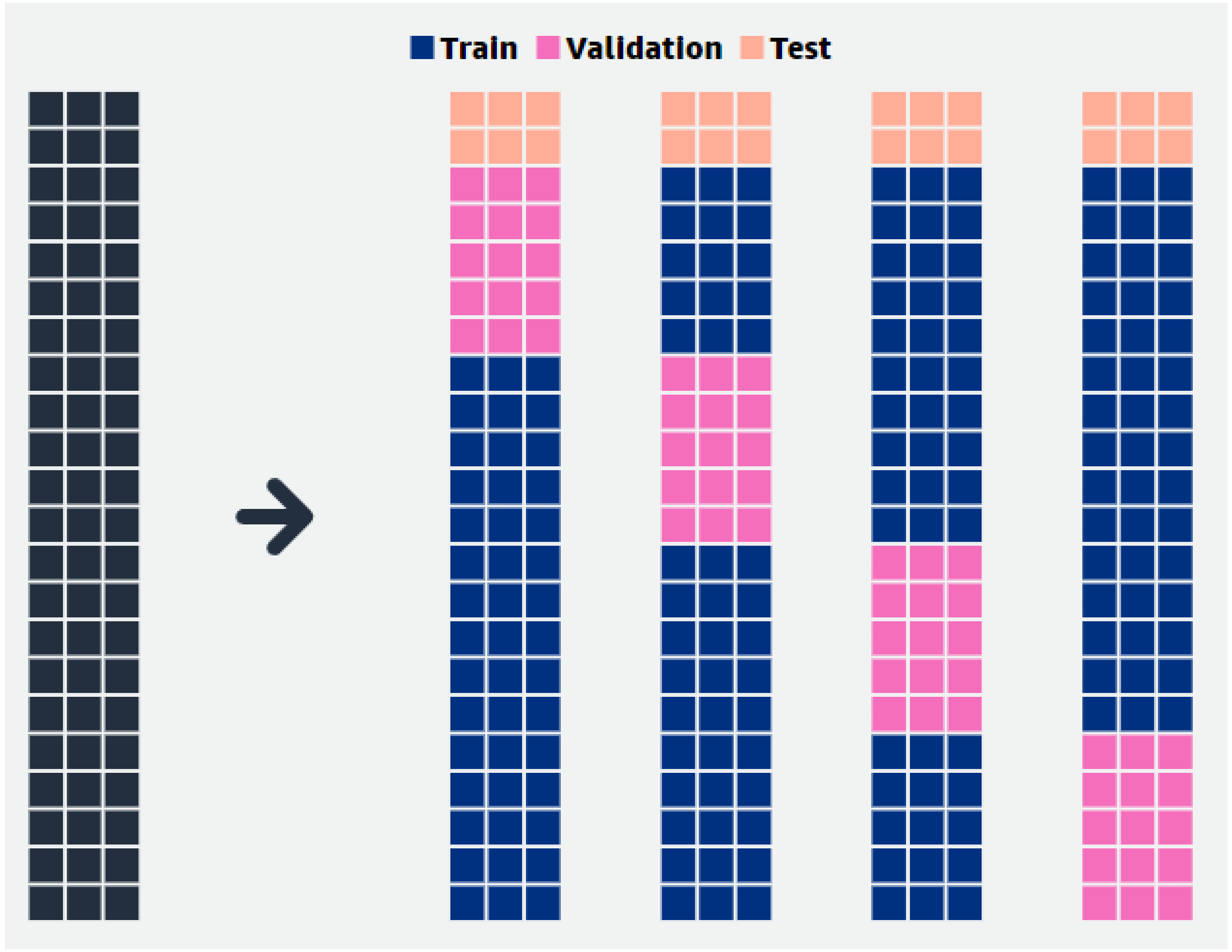


With K-Fold Cross-Validation, the whole ensemble uses all of the data, so **every data point will get included in the training of a model**, and the evaluation of that model will then be factored into the final evaluation estimate.

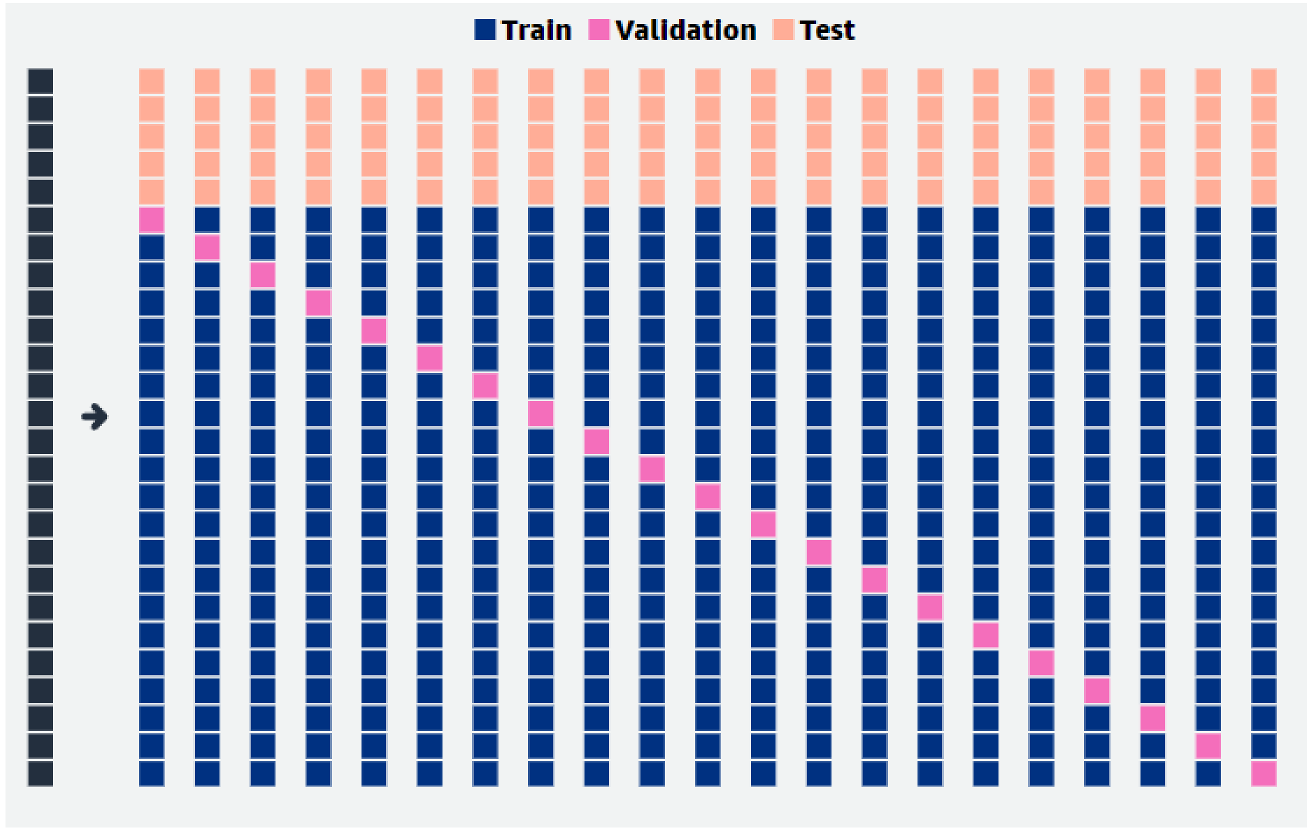


Finally, it's worth stating the obvious fact that test error estimates are more accurate when more data is used in the training set.

Even for modest values of k in K-Fold Cross-Validation (e.g. k=5), the training set comprises 80 percent of our data, so the approach typically doesn't overestimate the test error as much as the validation set approach could for small training set sizes.

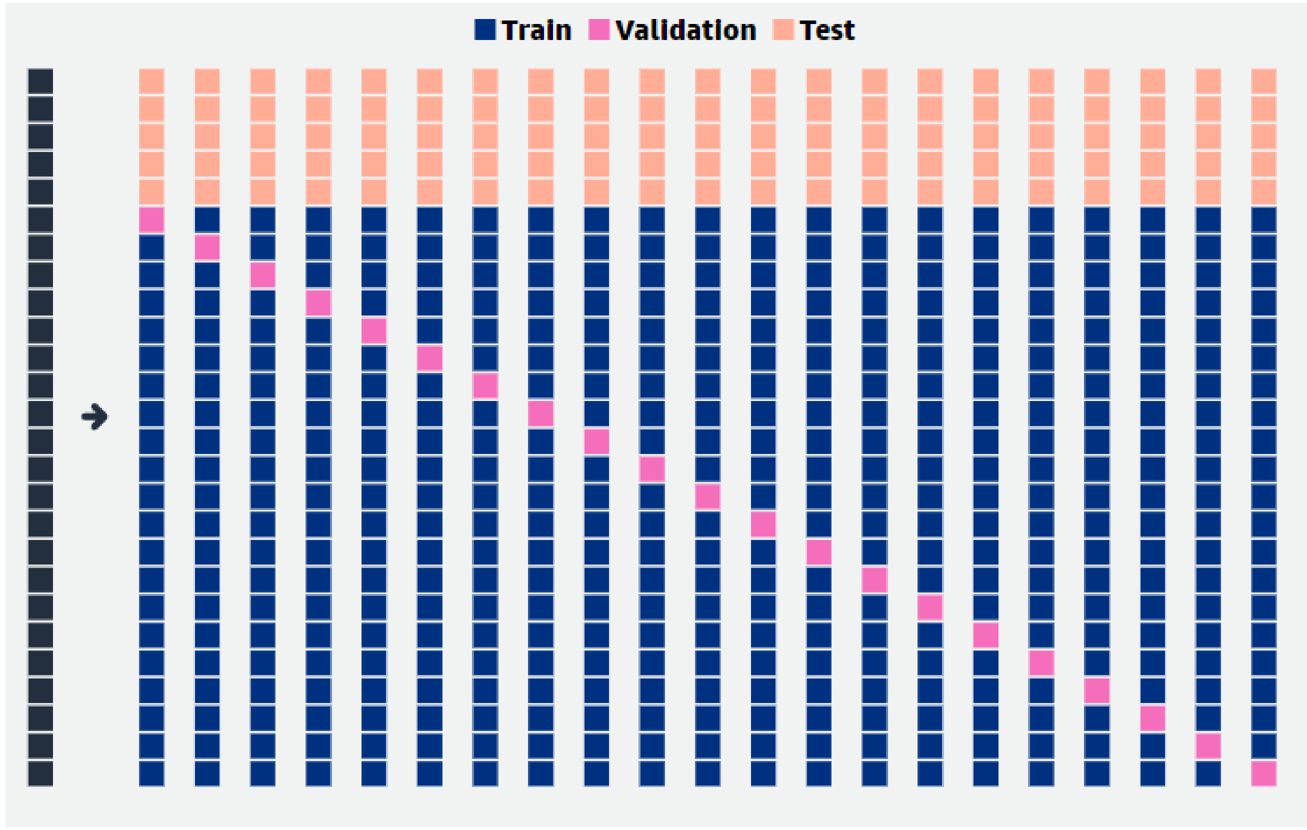


A special case of K-Fold Cross-Validation, **Leave-One-Out Cross-Validation (LOOCV)**, occurs when we set *k* equal to *n*, the number of observations in our dataset.



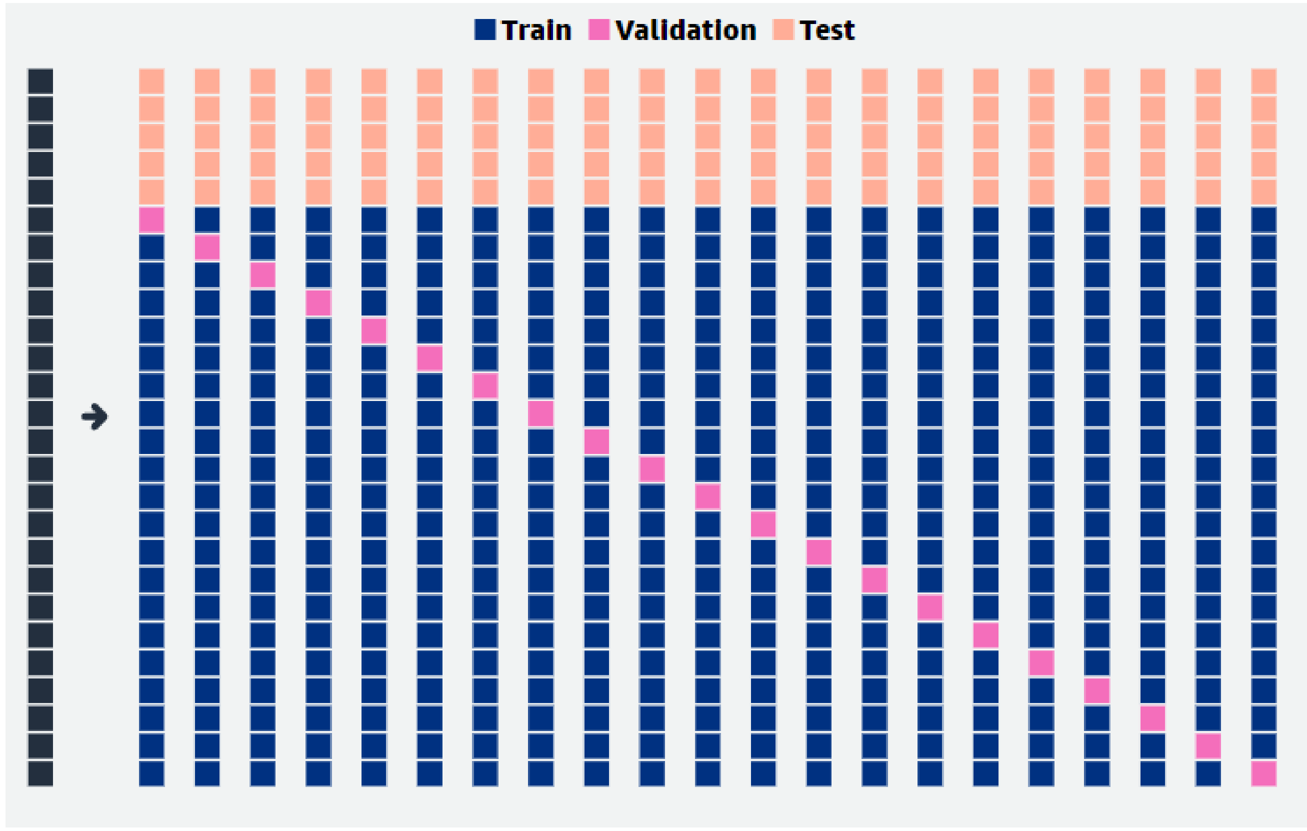
The **oft-cited intuition is that the variance of the CV estimator should be higher for larger *k***, so very high for LOOCV, since each model is trained on nearly identical data, but assessed on every individual point. Meaning that every model should be essentially the same, but assessed across all outliers or extreme points.

However, this intuition isn't always correct and can depend heavily on the type of model and dataset used!



For instance, for linear regression LOOCV actually has the smallest asymptotic bias and variance amongst all choices of *k* in K-Fold Cross-Validation.

One thing is certain: higher values of *k* require training more models, so often *k*=5 or *k*=10 is the largest you might be able to do in practice.



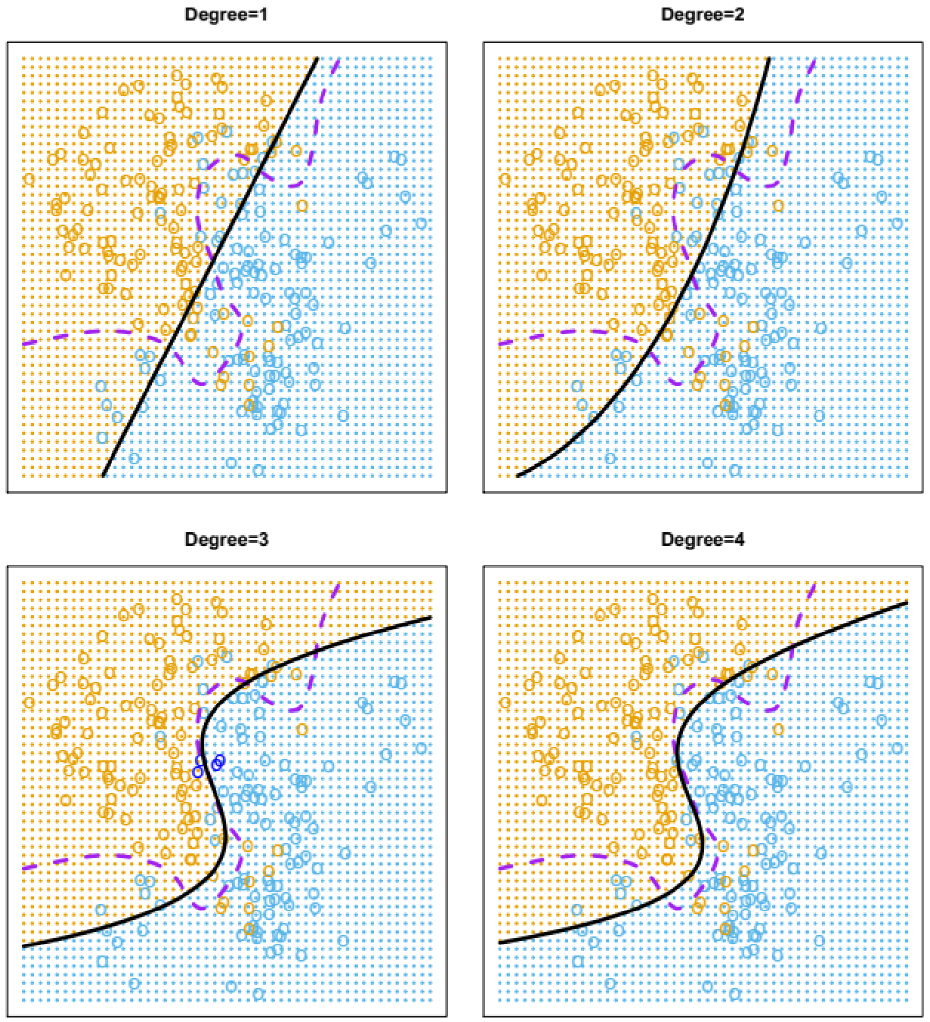


FIGURE 5.7. *Logistic regression fits on the two-dimensional classification data displayed in Figure 2.13. The Bayes decision boundary is represented using a purple dashed line. Estimated decision boundaries from linear, quadratic, cubic and quartic (degrees 1–4) logistic regressions are displayed in black. The test error rates for the four logistic regression fits are respectively* 0.201, 0.197, 0.160, *and* 0.162, *while the Bayes error rate is* 0.133.

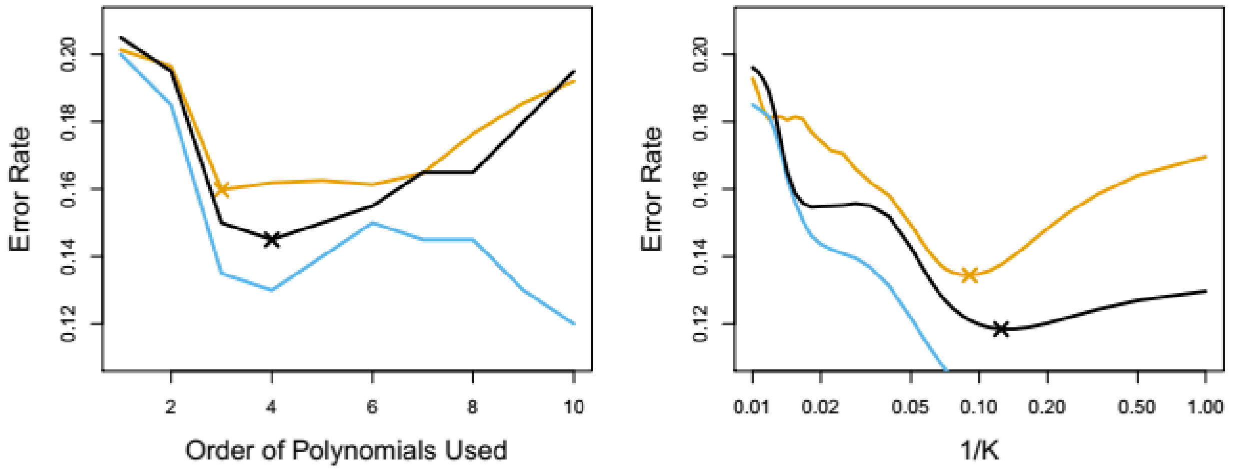
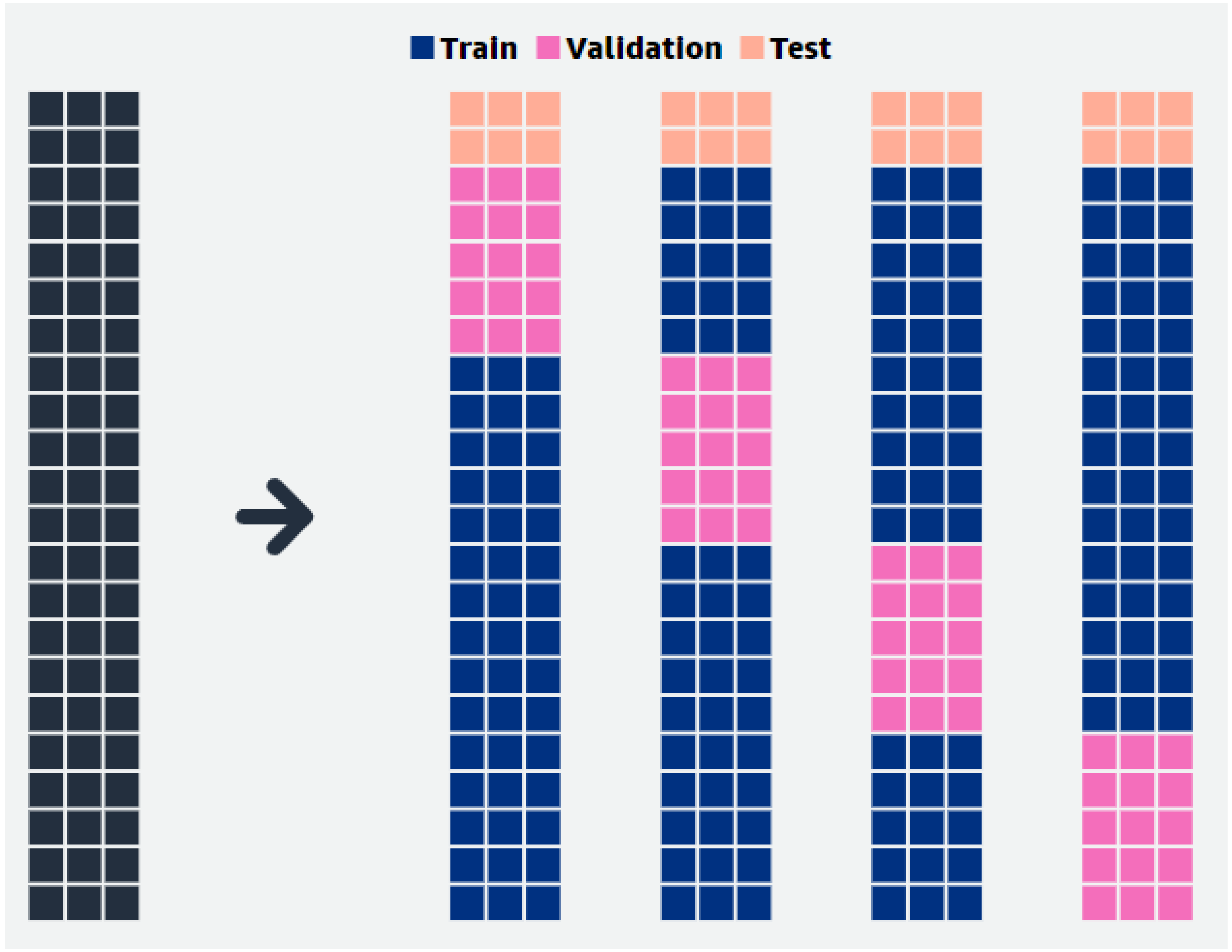
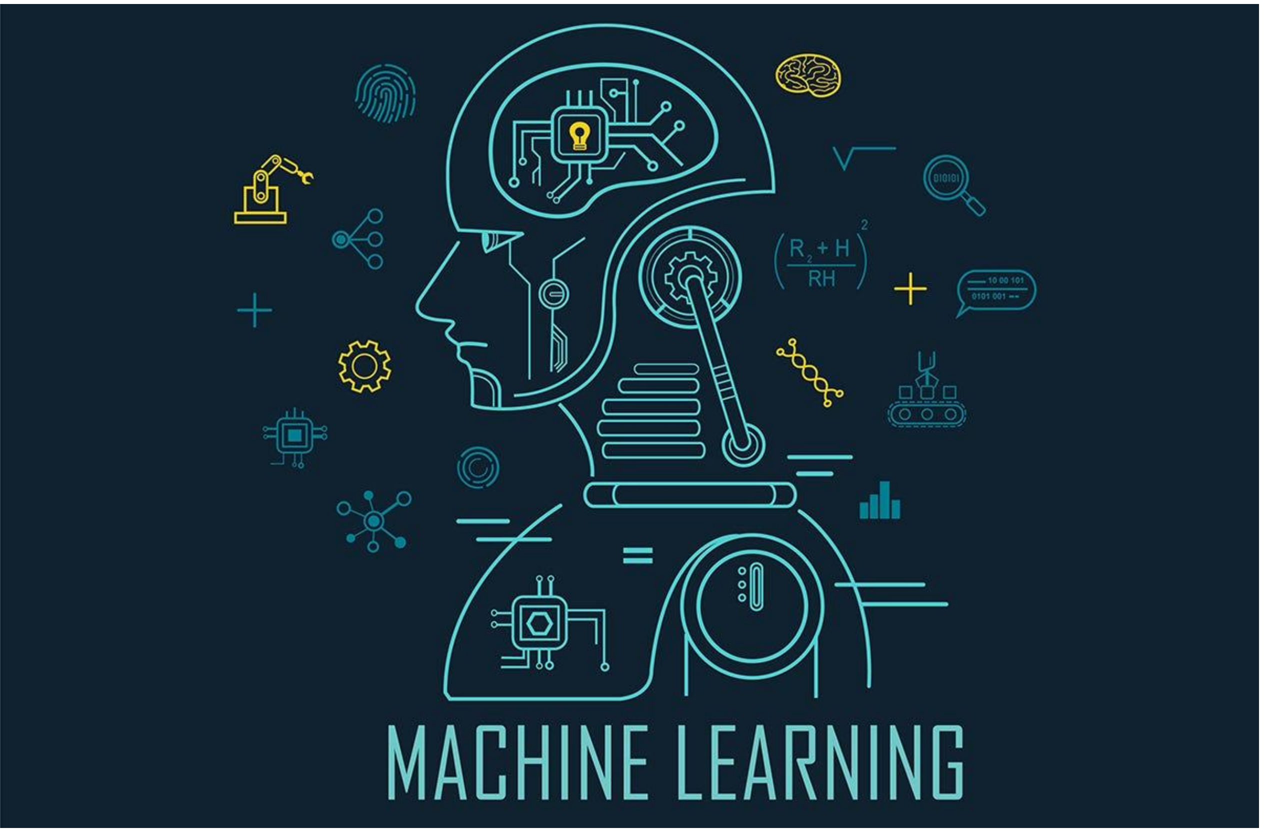


FIGURE 5.8. *Test error (brown), training error (blue), and 10-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7.* Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The KNN classifier with different values of K, the number of neighbors used in the KNN classifier.*

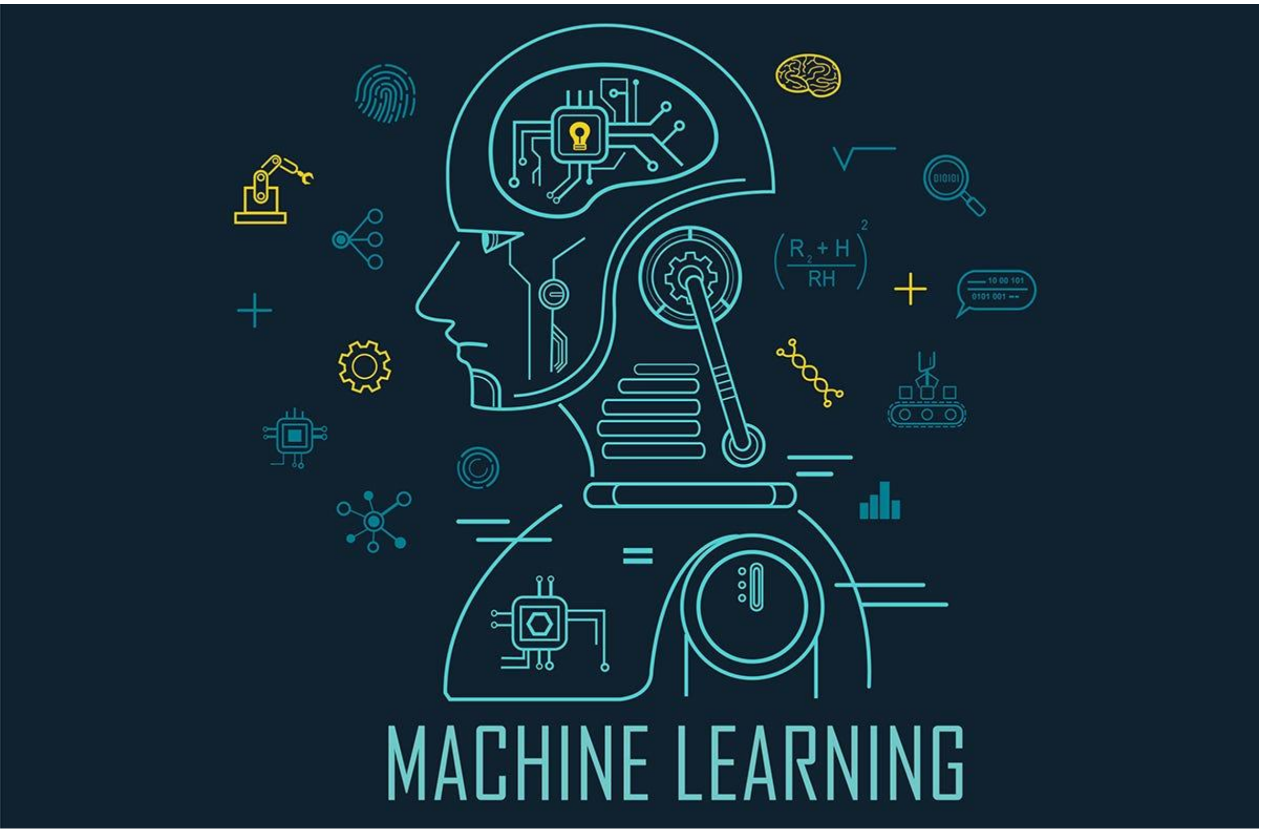
https://mlu-explain.github.io/cross-validation/





- Cross validation particularly important in finance!

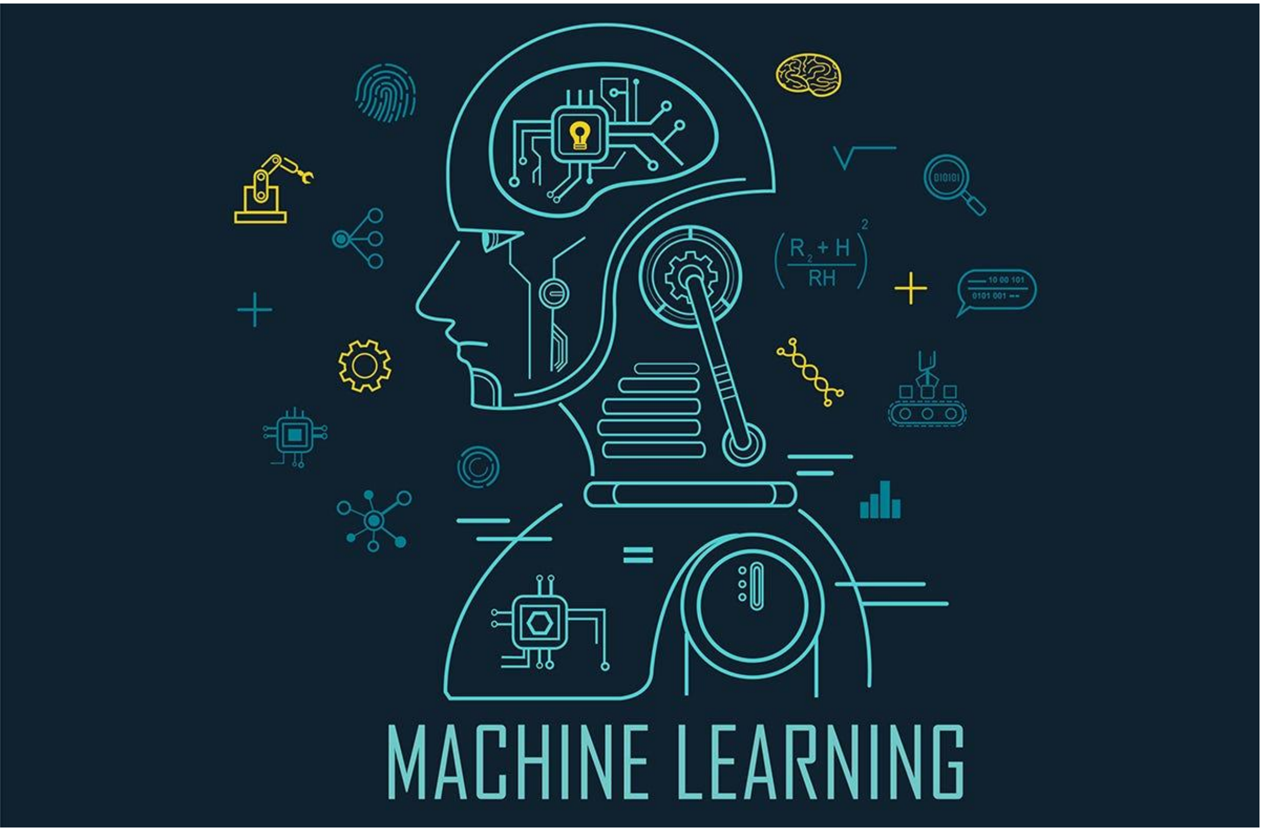
De Prado, Marcos Lopez. *Advances in financial machine learning*. John Wiley & Sons, 2018.



"Many papers in finance that present k-fold CV evidence that an ML algorithm performs well. Unfortunately, it is almost often the case that those results are wrong."

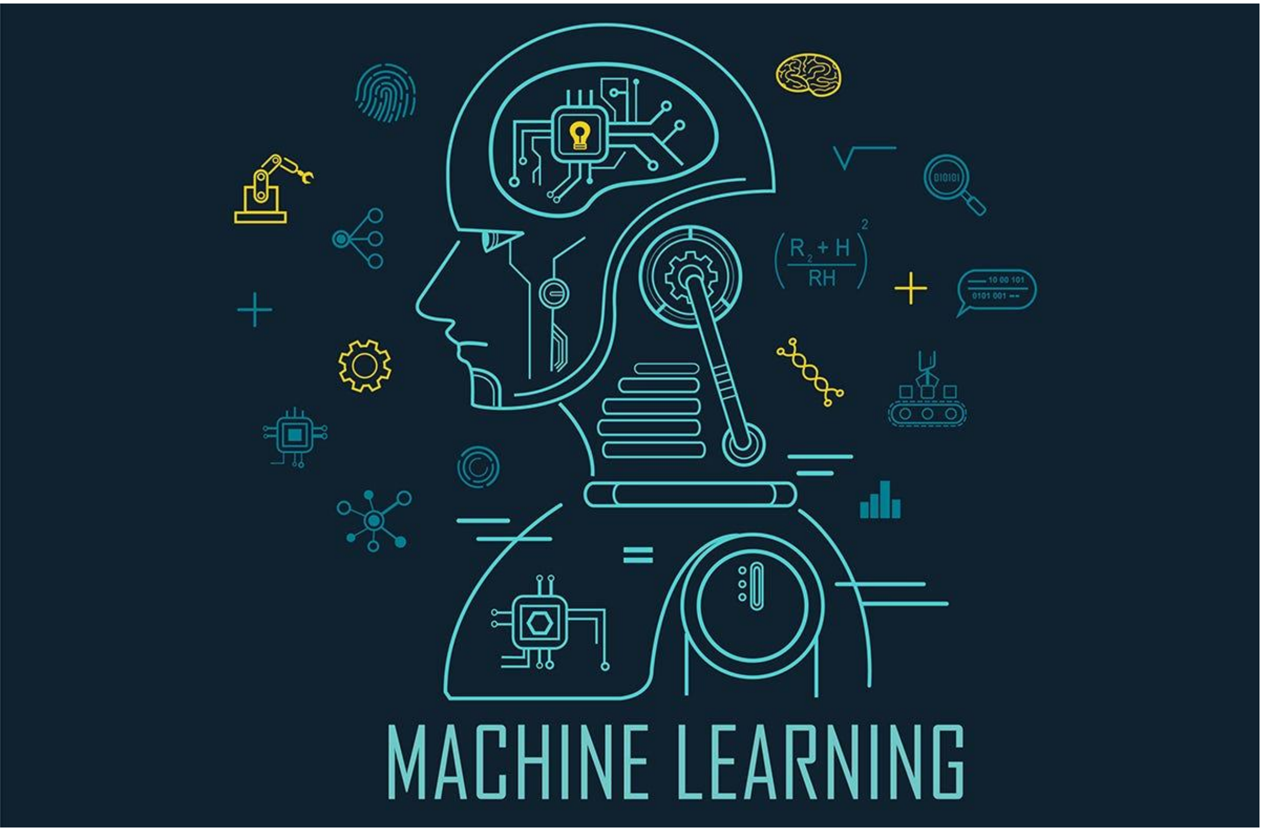
One reason k-fold CV fails in finance is because observations cannot be assumed to be drawn from an IID process.

De Prado, Marcos Lopez. *Advances in financial machine learning*. John Wiley & Sons, 2018.



Leakage takes place when the training set contains information that also appears in the testing set.

De Prado, Marcos Lopez. *Advances in financial machine learning*. John Wiley & Sons, 2018.



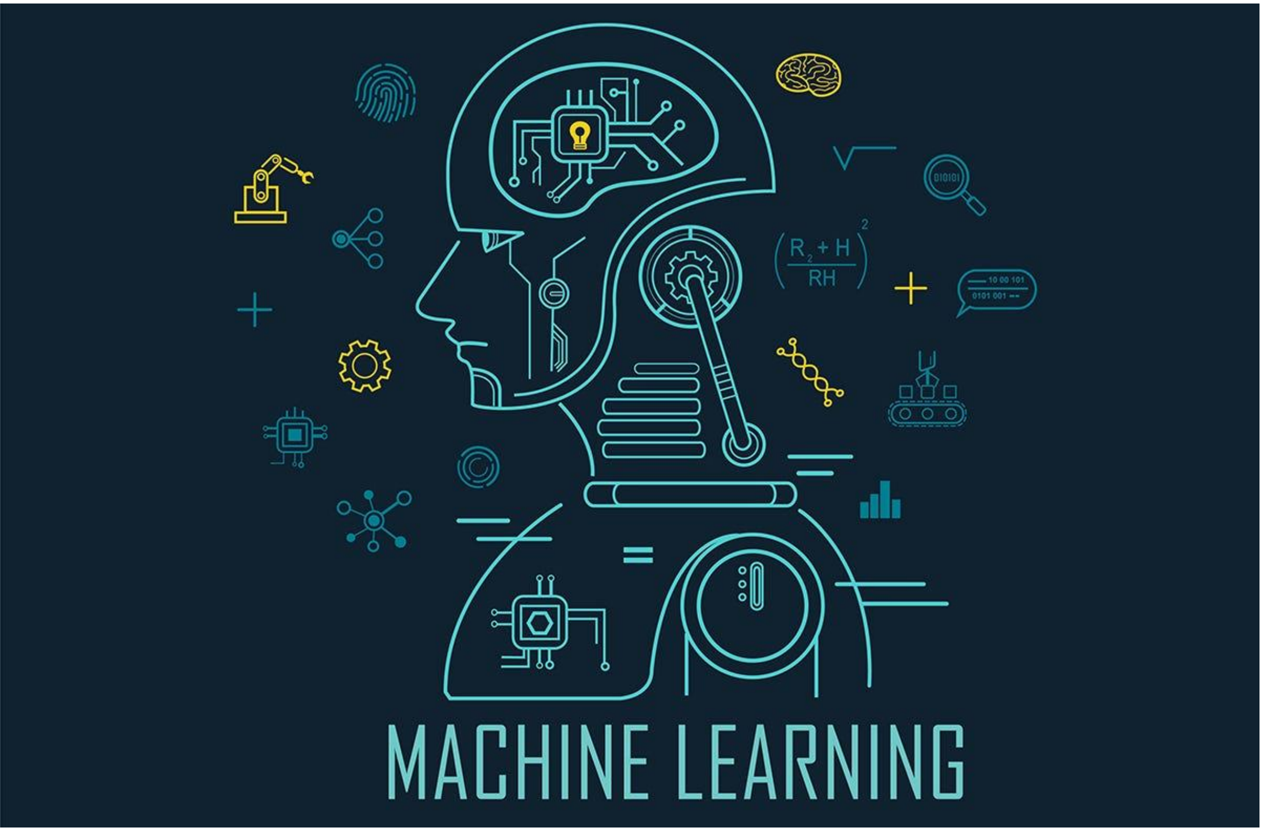
This is a problem in finance often.

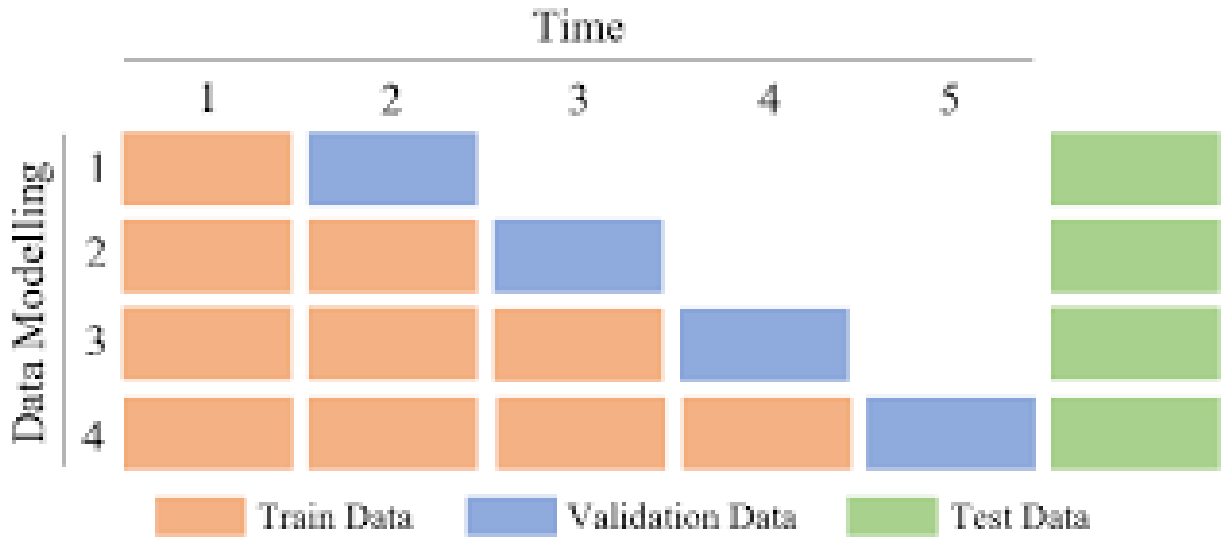
Consider a serially correlated feature X that is associated with labels Y that are formed on overlapping data:

- Because of serial correlation
- Because labels are derived from overlapping data points.

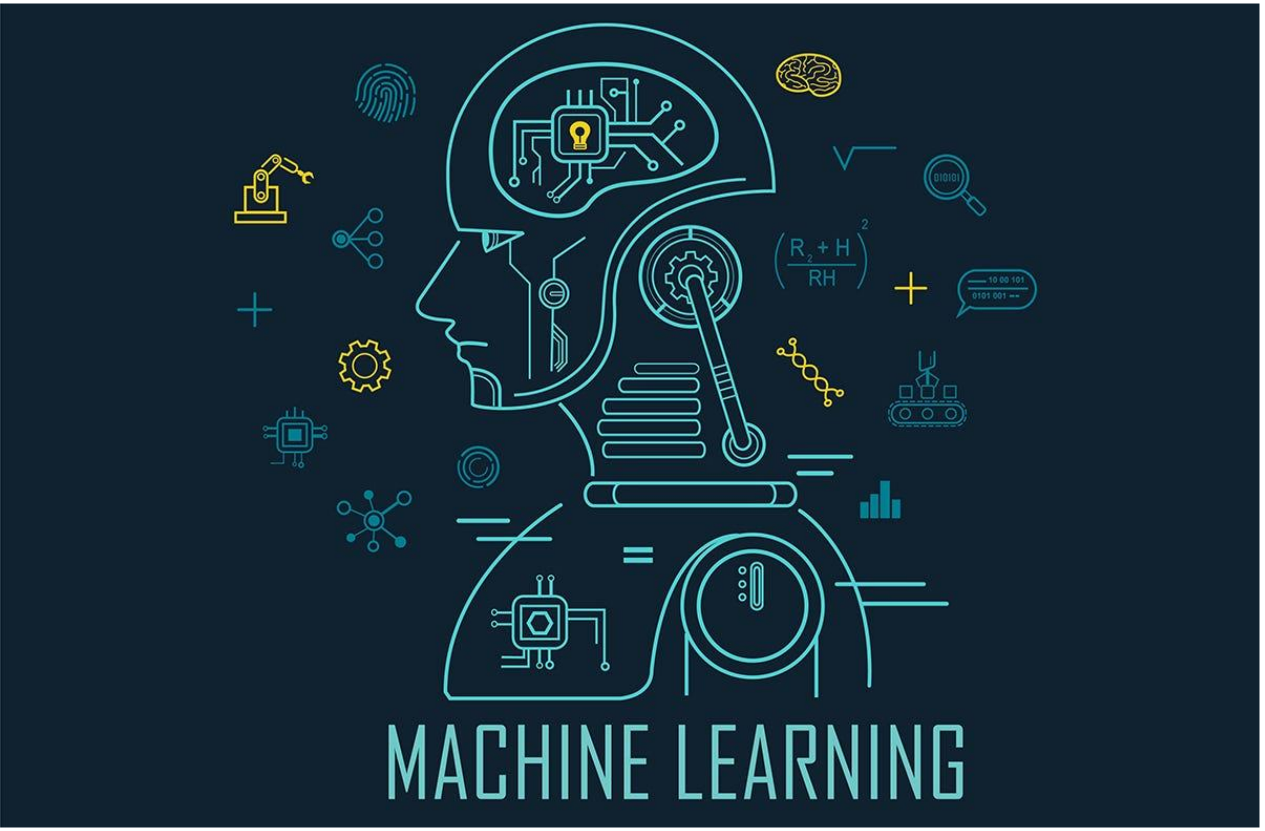
By placing t, and t+1 in different datasets information is leaked.

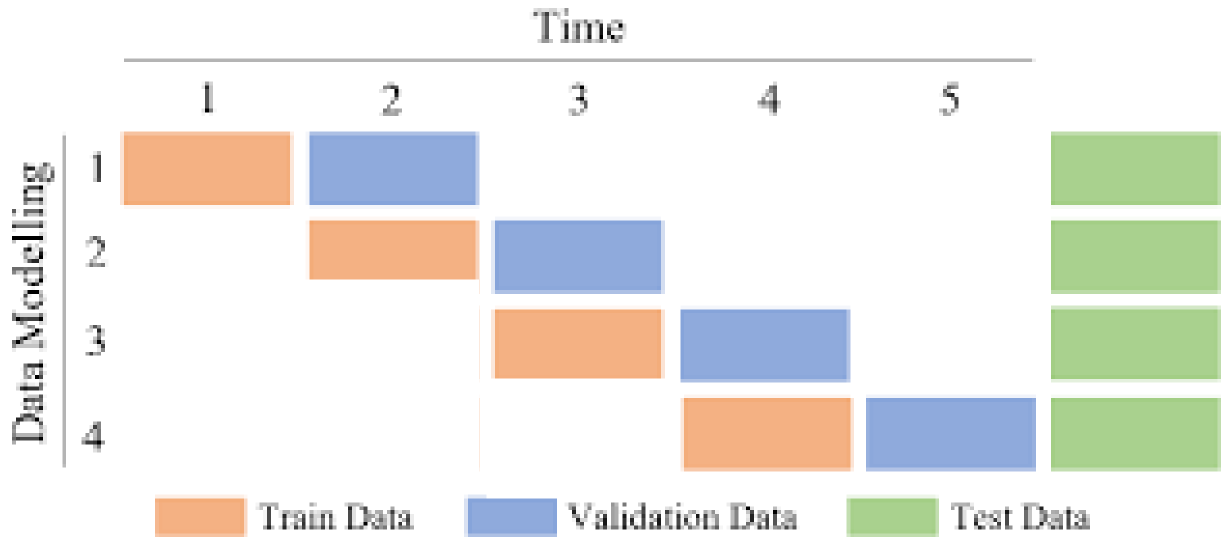
De Prado, Marcos Lopez. *Advances in financial machine learning*. John Wiley & Sons, 2018.



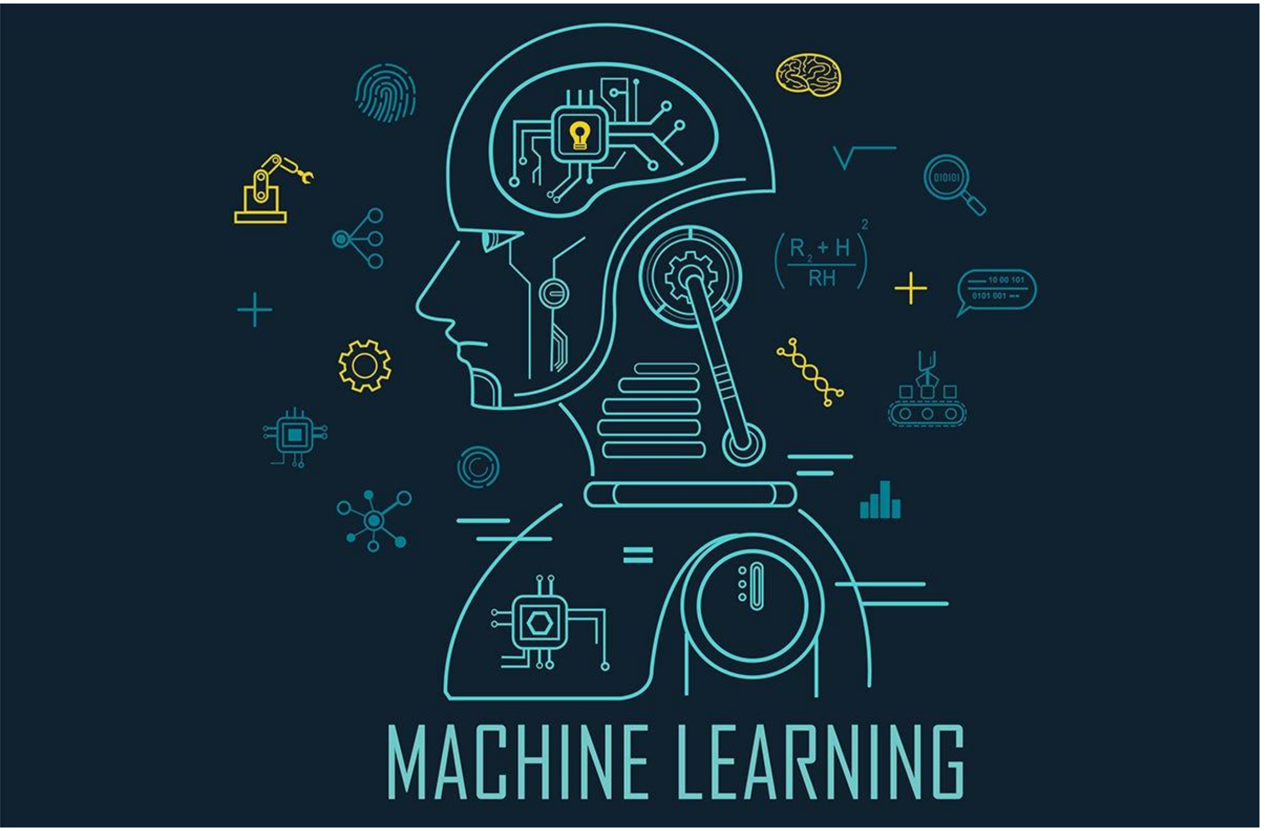


De Prado, Marcos Lopez. *Advances in financial machine learning*. John Wiley & Sons, 2018.



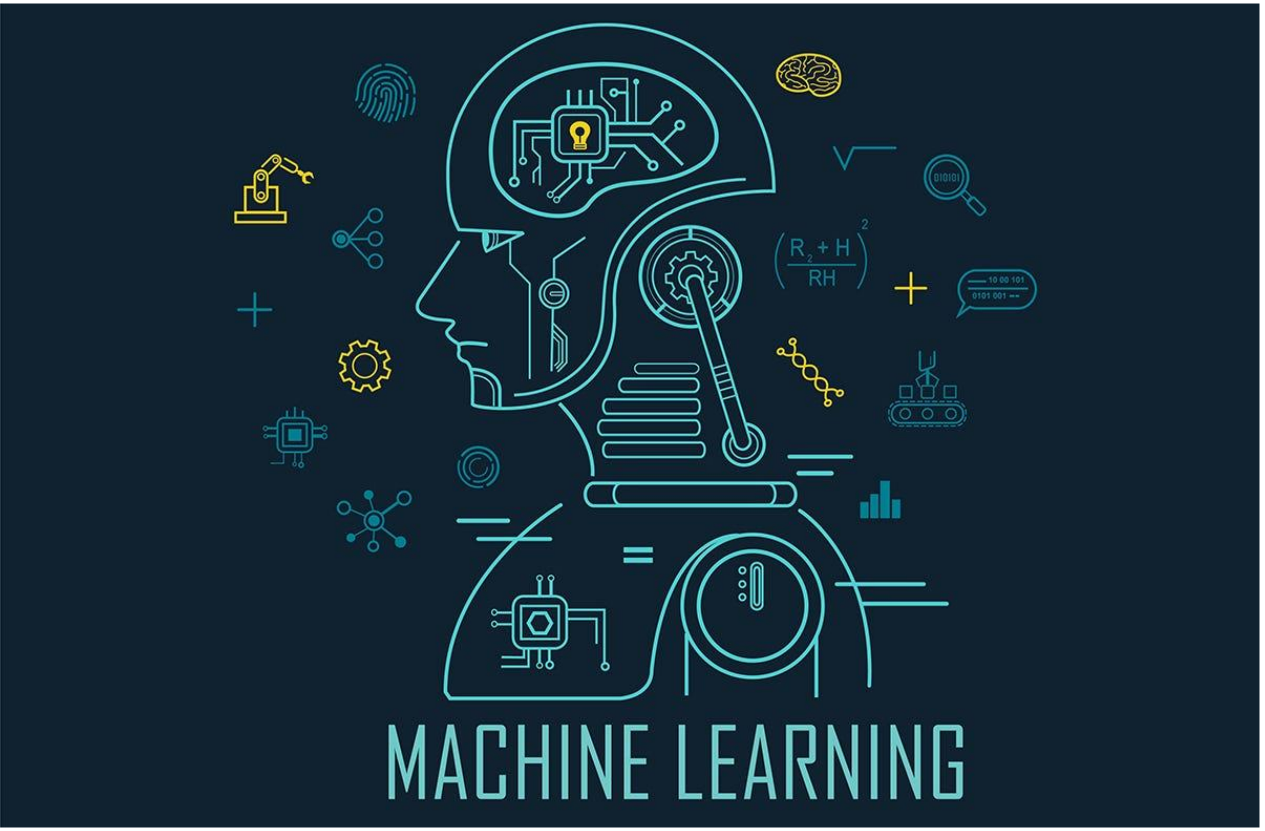


De Prado, Marcos Lopez. *Advances in financial machine learning*. John Wiley & Sons, 2018.



One way to reduce leakage is to purge from the training set all observations whose labels overlapped in time with those labels included in the testing set. I call this process "purging." In addition, since financial features often incorporate series that exhibit serial correlation (like ARMA processes), we should eliminate from the training set observations that immediately follow an observation in the testing set. I call this process "embargo."

De Prado, Marcos Lopez. *Advances in financial machine learning*. John Wiley & Sons, 2018.



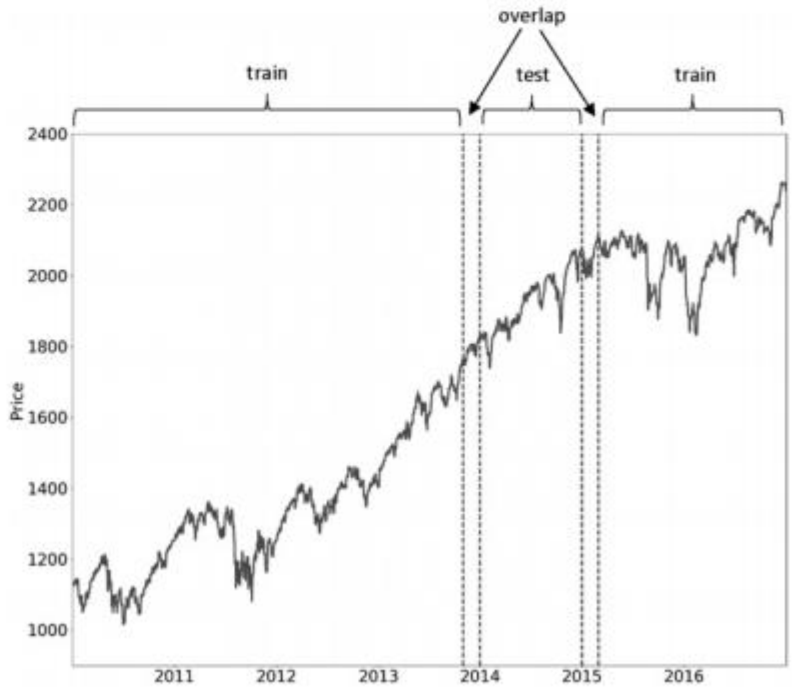
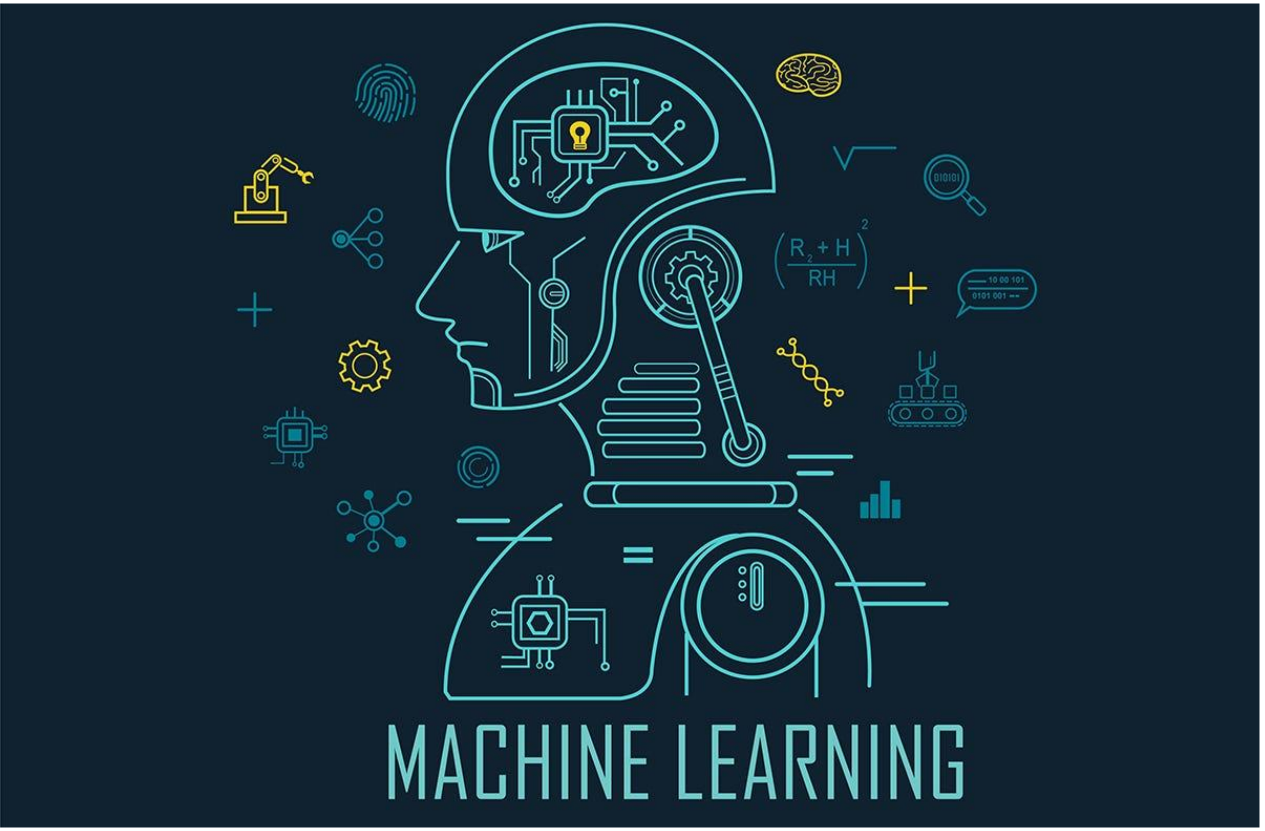


FIGURE 7.2 Purging overlap in the training set

De Prado, Marcos Lopez. *Advances in financial machine learning*. John Wiley & Sons, 2018.



- Labs