

The Data Science Process, Part 2

Neba Nfonsang

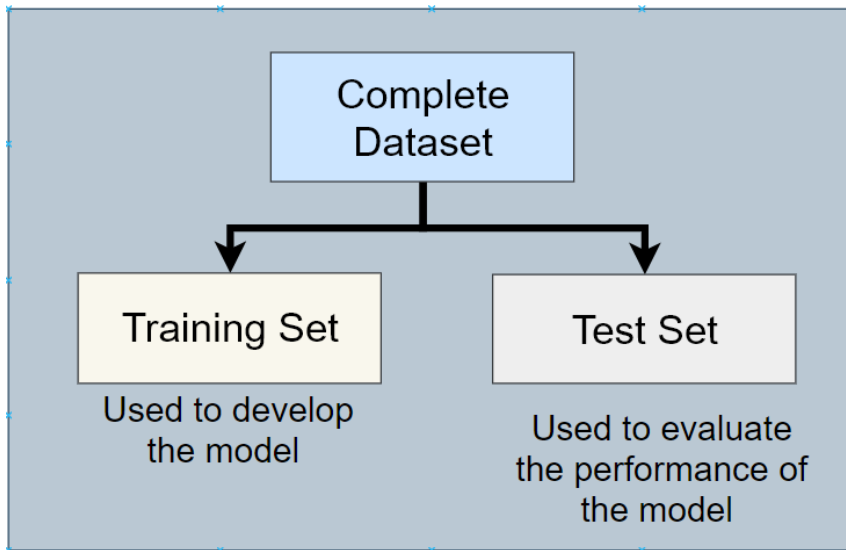
The Data Science Process Outline: Part 2

- Data partitioning
- Model construction
- Model evaluation
- Model evaluation metrics

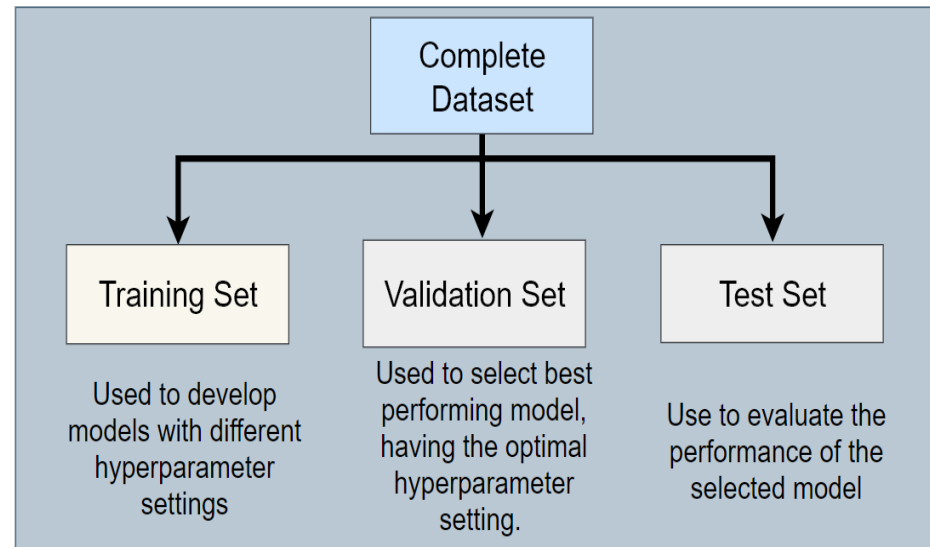


Data Partitioning

Two-way partition



Three-way partition



Data Partitioning (cont.)

- Data partitioning is the splitting of data into training and test datasets. In some situations, the data is split into training, validation, and test datasets
 - The training, validation, and test datasets are used for different purposes
- The training dataset is used for model building
 - The validation dataset is used for model selection or to fine-tune the hyperparameters of the model
 - The test dataset is used to compute model performance on new (or unseen) instances

Data Partitioning (cont.)

- During data splitting, instances from the entire dataset are randomly assigned to training, validation, and test datasets
- Recommended proportions for each dataset is:
 - Training: 60%
 - Validation: 20%
 - Test: 20%
- If the data is split into training and test datasets, then the following proportions are recommended for splitting:
 - Training: 70%
 - Test: 30%

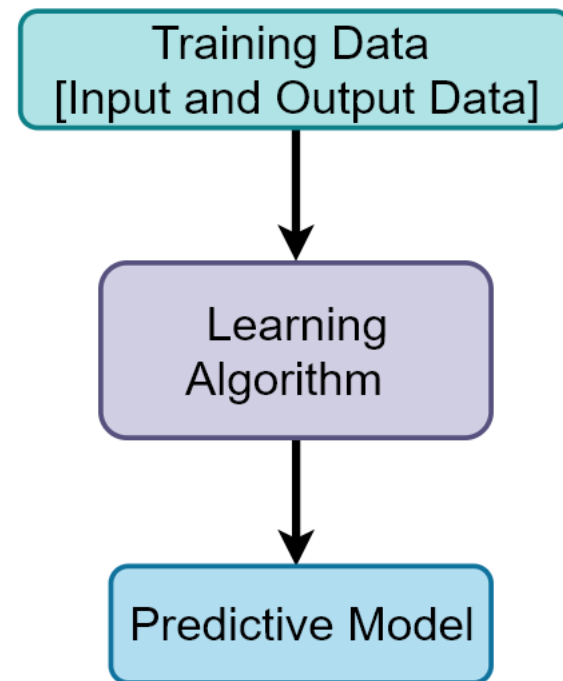
Ways of Data Partitioning

The End

Model Construction

Model Construction

- This involves the application of the appropriate learning algorithm to the training dataset to create the model
- A model is an abstract representation of the structure or relationships in a given dataset



Model Construction (cont.)

- To construct a specific model, there could be several possible algorithms that could be used
- For example, a decision tree model can be constructed with the ID3 algorithm or CART algorithm
- The appropriate algorithm should be selected for building a specific model
- One key thing is to understand how the algorithm you use works, know its inductive bias, assumptions, advantages, and limitations

Model Construction (cont.)

- The model to be constructed is determined by the type of task or problem to be solved
- Supervised learning task:
 - Classification
 - Regression
- Unsupervised learning task:
 - Clustering
- For each type of task, such as classification, an approach such as decision tree, rule induction, neural network, Bayesian model, K-NN, etc., is selected
- For a chosen approach, a suitable algorithm is selected

Model Construction

The End

Model Evaluation

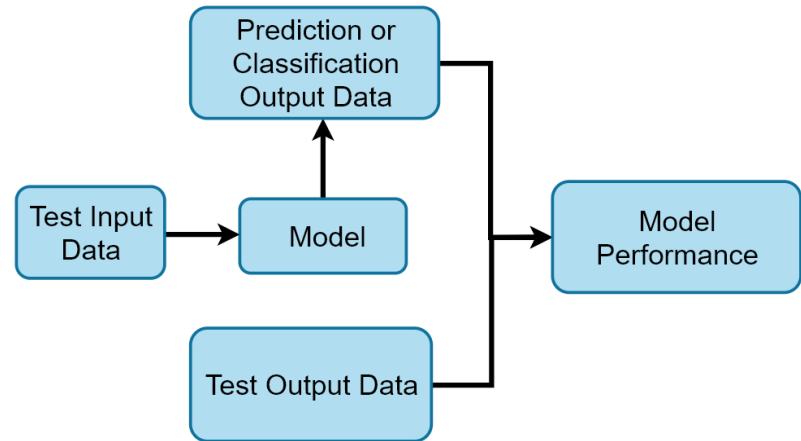
Part I

Model Evaluation

- Model evaluation, also known as model assessment, is a process of estimating the performance of a model
- The performance of a model on future or unseen data is called generalization (or test) error
- Model evaluation involves:
 - Estimating the performance of a model on unseen data
 - Tuning the hyperparameters of a model to select the proper level of flexibility (model selection)
 - Comparing algorithms and selecting the best one based on the model performance and algorithm efficiency

Model Evaluation (cont.)

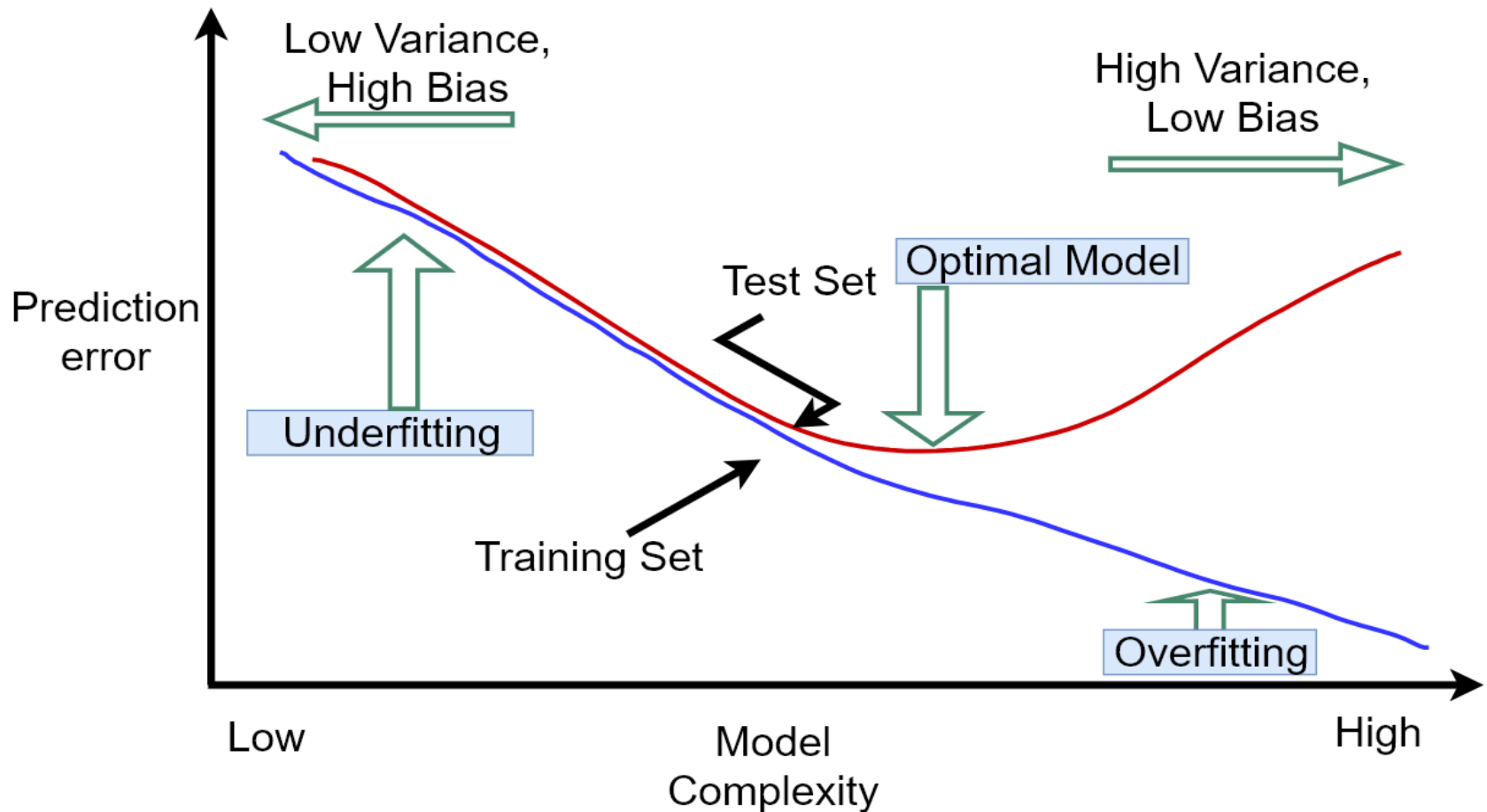
- We will focus on model evaluation of supervised learning
- Model performance is estimated using prediction error or accuracy, which involves comparing predicted output values to actual output values.



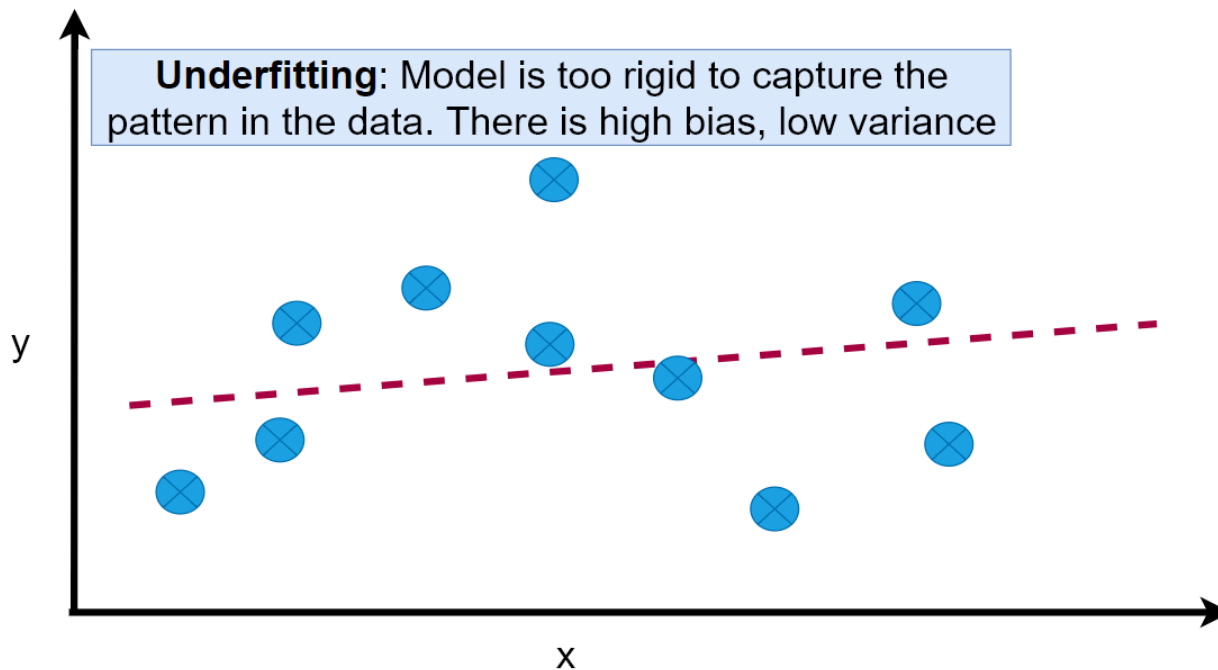
Model Evaluation (cont.)

- Model performance could be estimated for the training, validation, and test datasets
- We are interested in knowing how well the model performs on new or unseen data; so, model performance should be based on the test dataset.
- It is possible that a model could perform well on the training dataset and performs poorly on a test dataset; this situation is called **overfitting**.
- If the model performs poorly on both the training and test datasets, this is called **underfitting**

Model Evaluation (cont.)

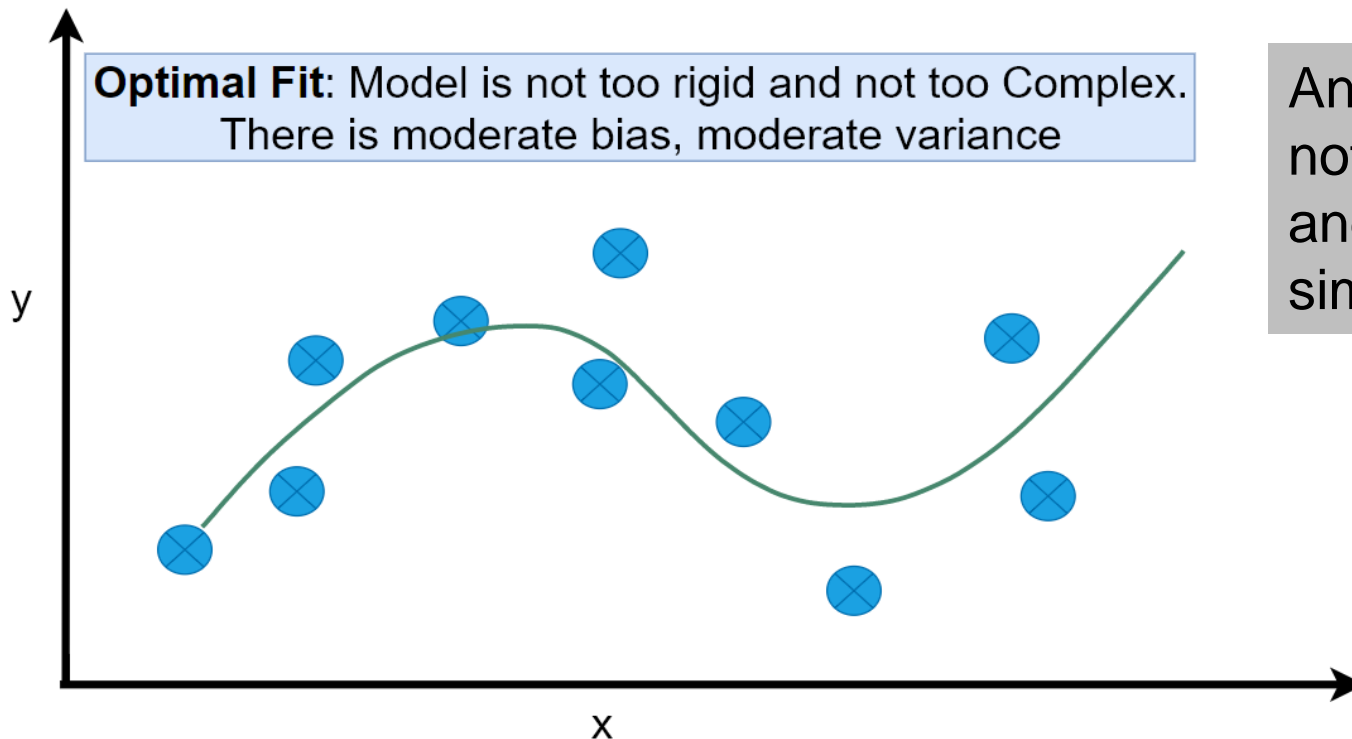


Model Evaluation (cont.)



Models that are too simple or rigid underfit the data.

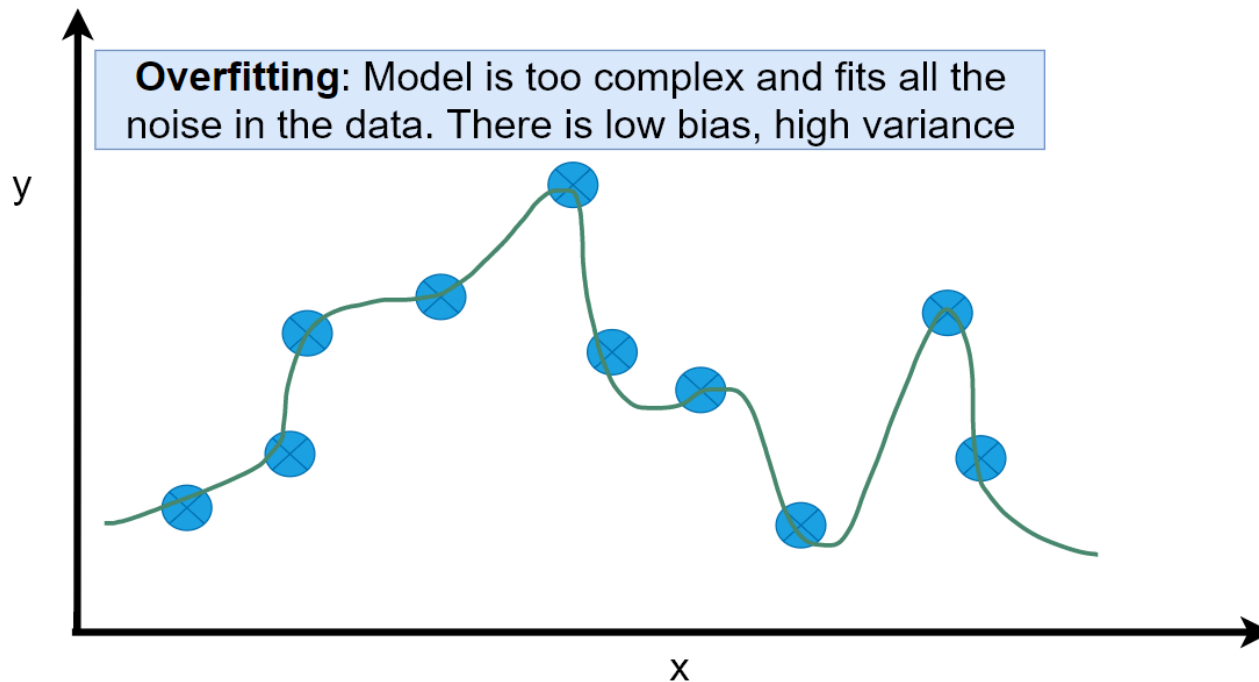
Model Evaluation (cont.)



Optimal Fit: Model is not too rigid and not too Complex.
There is moderate bias, moderate variance

An optimal model,
not too complex
and not too
simple, is desired.

Model Evaluation (cont.)



Models that are too complex memorize the training examples and overfit the data.

Model Evaluation (cont.)

- The complexity or flexibility of a model increases with increase in number of hyperparameters.
- Hyperparameters are parameters that need to be specified a priori (before the model is fitted).

Model	Hyperparameter
Polynomial regression	Degree of polynomial
Naïve Bayes	Number of attributes
Decision Tree	Number of nodes in the tree
K-Nearest Neighbor	Number of nearest neighbors

Model Evaluation (cont.)

Model selection

- Model selection involves selecting the best performing model (with the proper level of flexibility) from a given hypothesis space.
- The validation dataset is used for model selection.
- This process of searching for the best performing model, with the optimal hyperparameter value is called hyperparameter tuning.

Model Evaluation: Part I

The End

Model Evaluation

Part II

Model Evaluation: Approaches

- Some approaches to model evaluation involve model selection and estimating model performance.
- Some other approaches to model evaluation involve estimating model performance only (without model selection).

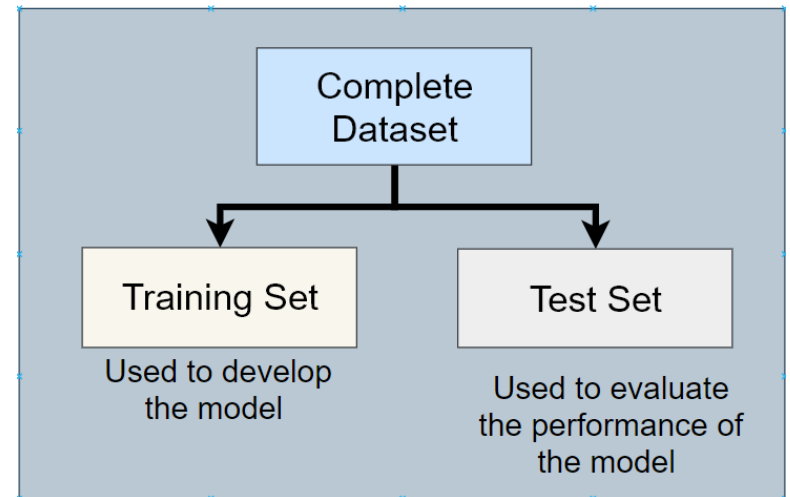
Approaches to model evaluation:

- Two-way holdout validation
- Three-way holdout validation
- Cross validation
 - Leave-one-out validation
 - K-fold cross-validation

Model Evaluation

Two-way holdout validation

- This involves randomly splitting the available dataset into training set and test set.
- The training set is used to train the model and the test set is used to estimate the model performance.

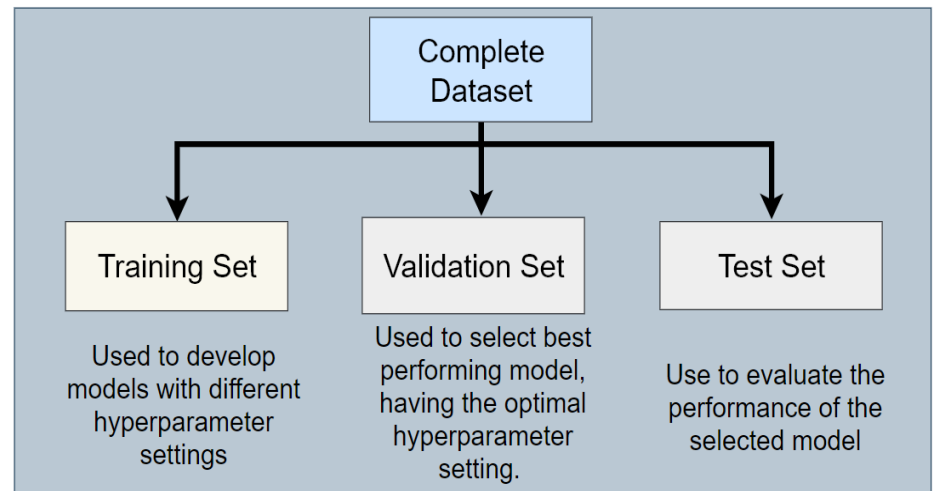


Hyperparameters are not tuned. A fixed hyperparameter is specified based on intuition or the default hyperparameter in a software package is used.

Model Evaluation (cont.)

Three-way holdout validation

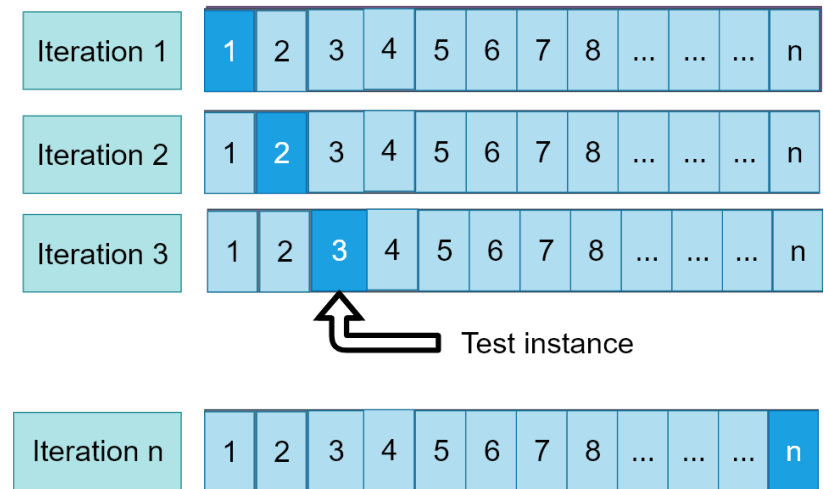
- The dataset is randomly split into training, validation, and test datasets used for model fitting, model selection, and for the final evaluation of the selected model, respectively.



Model Evaluation (cont.)

Leave-one-out cross-validation

- Leave-one-out cross validation (LOOCV) uses a single instance as the test set and the rest of the instances are used as training set.
- Training set is used to develop model and test set is used to evaluate model performance.



Model Evaluation (cont.)

Leave-one-out cross-validation

- N-iterations are executed, and during each iteration, a different instance is selected as test set and the rest as training set and model is trained and performance estimated.
- The iterations terminate when all examples have been used as test sets.
- N-test errors are computed and averaged to get the final model performance.

Model Evaluation (cont.)

K-fold cross-validation

- The complete dataset is randomly split into k-folds of approximately equal size.
- Iteratively, every fold serves as a test set while the rest of the k-1 folds serve as the training set.



Model Evaluation (cont.)

K-fold cross-validation

- The training set is used to train the model and the test set is used to estimate model performance.
- The process is repeated k-times where a different fold is used as test set and the rest as training set in each iteration.
- K-test errors are computed and averaged to get the final model performance
- Compared to two-way and three-way holdout validations, cross validation is used when there are not enough examples to have a separate holdout (test) set.

Model Evaluation (cont.)

- Cross validation can be used for model selection as well.
- In this case, the data is first split into training set and test set.
- The training set is then used for the cross validation.
- Cross validation is done using models with different hyperparameter values.
- The model with the optimal hyperparameter is applied to the test set to estimate performance.

Model Evaluation: Part II

The End

Model Evaluation Metrics

Part I

Model Evaluation Metrics

- For regression models, accuracy or performance measures such as MSE is used. However, we will focus on the accuracy measure of classification models.
- There are different accuracy measures for classification models.
- The **confusion matrix** is commonly used to compute the performance of a classification model
- The **confusion matrix** displays the number of cases that are correctly predicted and incorrectly predicted by the classifier.

The Confusion Matrix

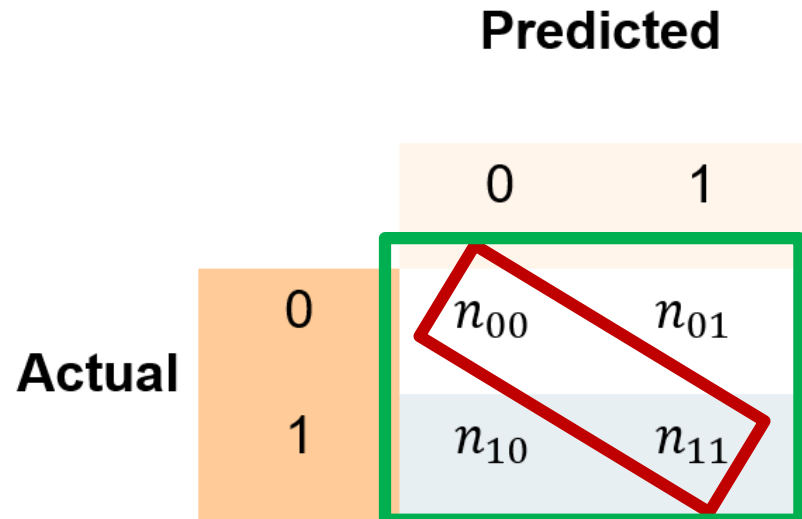
- n_{00} = True Negatives (TN)
- n_{11} = True Positives (TP)
- n_{01} = False Positives (FP)
- n_{10} = False Negatives (FN)
- If the classifier is trying to predict those who default or do not default on a loan, then:
- 0 = default, 1=no default

		Predicted	
		0	1
Actual	0	n_{00}	n_{01}
	1	n_{10}	n_{11}

Accuracy

- Accuracy: This is the proportion of cases that were correctly predicted or classified.
- $$\text{Accuracy} = \frac{n_{00} + n_{11}}{n_{00} + n_{11} + n_{01} + n_{10}}$$
- Accuracy = 1 – Overall Error Rate

		Predicted	
		0	1
Actual	0	n_{00}	n_{01}
	1	n_{10}	n_{11}

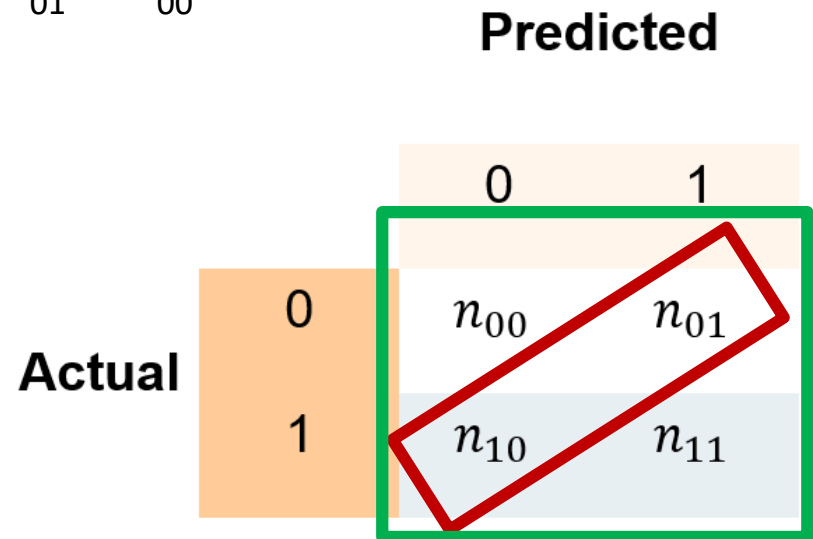


Overall Error Rates

- **Overall error rate** is the number of cases incorrectly predicted divided by the total number of cases in the dataset.

$$\text{Overall error rate:} = \frac{n_{10} + n_{01}}{n_{11} + n_{10} + n_{01} + n_{00}}$$

What proportion of observations (in both classes) were incorrectly predicted?



Class 1 Error Rate (False Negative Rate)

- **Class 1 error rate** is the number of actual class 1 cases incorrectly predicted divided by total number of actual cases in class 1.
- Class 1 error rate =
$$\frac{n_{10}}{n_{11} + n_{10}}$$

What proportion of observations in class 1 were incorrectly predicted?

		Predicted	
		0	1
Actual	0	n_{00}	n_{01}
	1	n_{10}	n_{11}

Error Rate (False Positive Rate)

- **Class 0 error rate:** Number of actual class 0 cases incorrectly predicted divided by total number of cases in class 0.
- Class 1 error rate = $\frac{n_{01}}{n_{01} + n_{00}}$

What proportion of observations in class 0 were incorrectly predicted?

		Predicted	
		0	1
Actual	0	n_{00}	n_{01}
	1	n_{10}	n_{11}

The Cost of Error Rates

- Note that the error rates are based on the cutoff values used by the algorithm implemented in solving the classification problem.
- We could plot error rates vs cutoff values for both class 1 and class 2 error rates
- Error rates tell us how much the model misclassifies.
- The cost of false positive and false negative errors are not the same.
- In classifying patients for testing and treatment, false negative is more costly since the patient would not be treated immediately.

Tradeoff in Error Rates

- The default error rate for most algorithms is 0.5. Generally, cases above 0.5 are classified into the class 1. If the cut-off value decreases, more cases would be classified into class 1 and class 1 error rate would decrease while class 0 error rate increases.
- The ROC (Receiver Operating Characteristics) curve is used to display the tradeoff between the classifier's ability to correctly predict class 1 and class 0 error rates.

Model Evaluation Metrics: Part I

The End

Model Evaluation Metrics

Part II

Sensitivity

- **Sensitivity or recall** measures the ability to predict class 1 observations.

$$\text{Sensitivity} = 1 - \text{Class 1 error rate} = \frac{n_{11}}{n_{11} + n_{10}}$$

What proportion of observations in class 1 were correctly predicted?

		Predicted	
		0	1
Actual	0	n_{00}	n_{01}
	1	n_{10}	n_{11}

Specificity

- Specificity measures the ability to predict class 0 observations.

$$\text{Specificity} = 1 - \text{Class 0 error rate} = \frac{n_{00}}{n_{11} + n_{10}}$$

What proportion of observations in class 0 were correctly predicted?

		Predicted	
		0	1
Actual	0	n_{00}	n_{01}
	1	n_{10}	n_{11}

Precision

- Precision is the number of cases correctly predicted to be in class 1 divided by total number of cases predicted to be in class 1.

$$\text{Precision} = \frac{n_{11}}{n_{11} + n_{01}}$$

What proportion of observations predicted to be in class 1 were correctly predicted?

		Predicted	
		0	1
Actual	0	n_{00}	n_{01}
	1	n_{10}	n_{11}

F-Score

- F - score = $2 * \text{precision} * \text{recall} / (\text{precision} + \text{recall})$

$$\text{F1 Score} = \frac{2n_{11}}{2n_{11} + n_{01} + n_{10}}$$

Admission Data

	admit	gre	gpa	rank
0	0	380	3.61	3
1	1	660	3.67	3
2	1	800	4.00	1
3	1	640	3.19	4
4	0	520	2.93	4

- View the entire data through the [UCLA Statistical Consulting website](#).
- GRE scores, GPA and rank of university is used to predict if students would be admitted or not.
- Admit is the outcome with values: 0 = not admitted, 1 = admitted

Data Preparation

- Dummy variables are created for categorical independent variables.
- Missing data is checked and addressed
- Ensure that the outcome data is coded as 0s and 1s.
- Data is later split into train and test datasets and standardized.

	admit	gre	gpa	rank	rank_1	rank_2	rank_3	rank_4
0	0	380	3.61	3	0	0	1	0
1	1	660	3.67	3	0	0	1	0
2	1	800	4.00	1	1	0	0	0
3	1	640	3.19	4	0	0	0	1
4	0	520	2.93	4	0	0	0	1

Probabilities of Being Predicted Into Class 1

- Logistic Regression was used to estimate the probability of each case belonging to class 1. The test set is used for classification.

	admit	prob	admit0	admit1	admit2	admit3	admit4	admit5	admit6	admit7	admit8	admit9	admit10
332	0	0.394765	1	1	1	1	0	0	0	0	0	0	0
111	0	0.177804	1	1	0	0	0	0	0	0	0	0	0
352	1	0.198543	1	1	0	0	0	0	0	0	0	0	0
205	1	0.369504	1	1	1	1	0	0	0	0	0	0	0
56	0	0.201721	1	1	1	0	0	0	0	0	0	0	0
379	0	0.244645	1	1	1	0	0	0	0	0	0	0	0
81	0	0.330106	1	1	1	1	0	0	0	0	0	0	0
214	1	0.409658	1	1	1	1	1	0	0	0	0	0	0
142	0	0.313159	1	1	1	1	0	0	0	0	0	0	0
110	0	0.206851	1	1	1	0	0	0	0	0	0	0	0

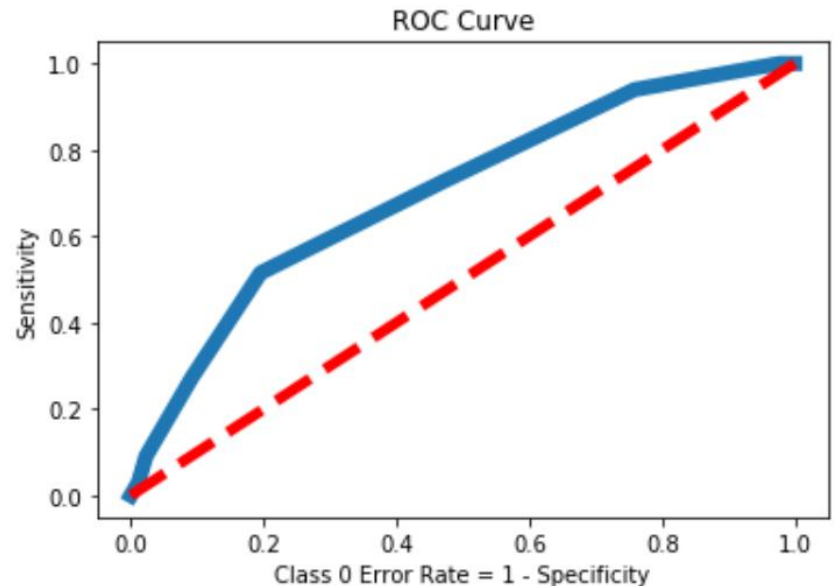
Classification Based on Different Cutoffs

- Cutoff values of 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 were use to predict whether each case belongs to class 1 or not.
- If probability is greater than cutoff, the case is classified to class 1.

	admit	prob	admit0	admit1	admit2	admit3	admit4	admit5	admit6	admit7	admit8	admit9	admit10
332	0	0.394765	1	1	1	1	0	0	0	0	0	0	0
111	0	0.177804	1	1	0	0	0	0	0	0	0	0	0
352	1	0.198543	1	1	0	0	0	0	0	0	0	0	0
205	1	0.369504	1	1	1	1	0	0	0	0	0	0	0
56	0	0.201721	1	1	1	0	0	0	0	0	0	0	0
379	0	0.244645	1	1	1	0	0	0	0	0	0	0	0
81	0	0.330106	1	1	1	1	0	0	0	0	0	0	0
214	1	0.409658	1	1	1	1	1	0	0	0	0	0	0
142	0	0.313159	1	1	1	1	0	0	0	0	0	0	0
110	0	0.206851	1	1	1	0	0	0	0	0	0	0	0

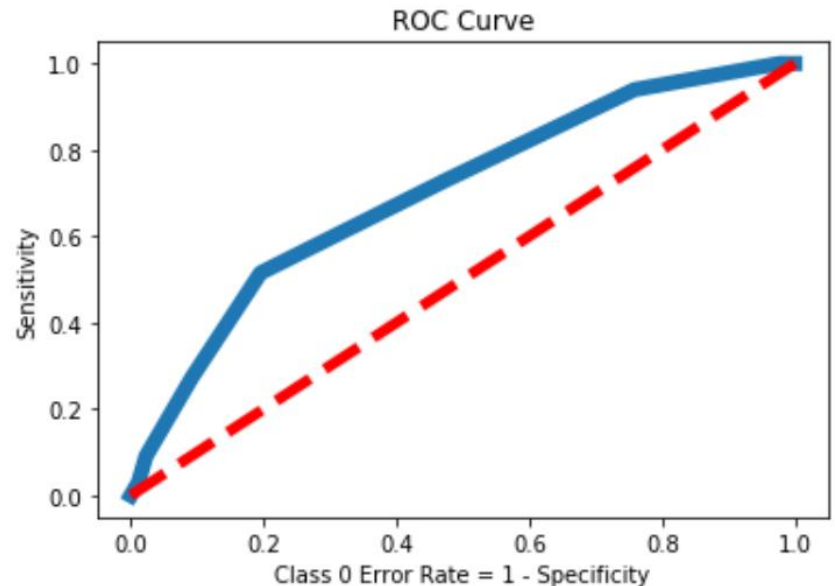
ROC Curve

- ROC curve is a graphical display of sensitivity (ability to correctly predict class 1 cases) and class 0 error rate.
- The area under the ROC curves is used to gauge the quality of the classifier.
- The area under the red line indicates random classification.



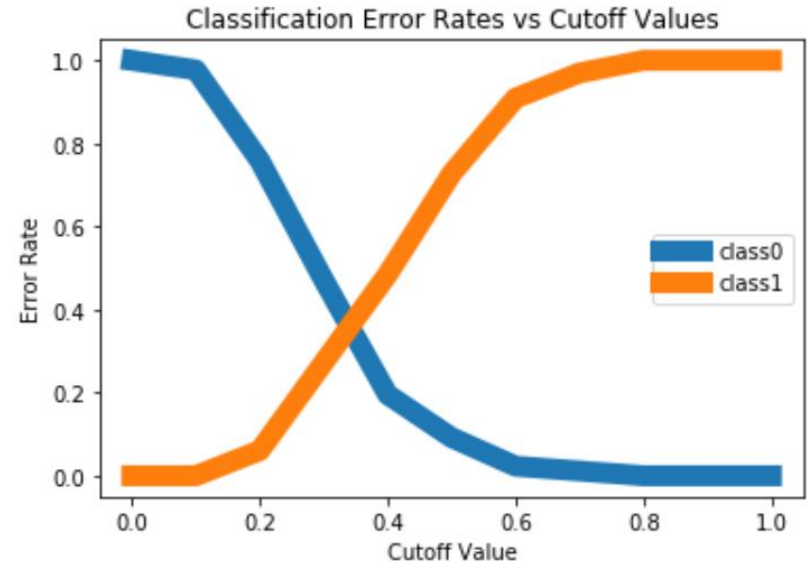
ROC Curve (cont.)

- The area under the red line is 0.5. The ROC curve is above the red line meaning the classifier is providing some value above random classification (or above guessing).
- The area under the curve measures the performance of the classifiers, the larger the better.



Classification Error Rates vs. Cutoffs

- The graph shows that when cutoff value increases, the class1 error rate increases and class 0 error rate decreases.



Model Evaluation Metrics: Part I

The End