Modul 3

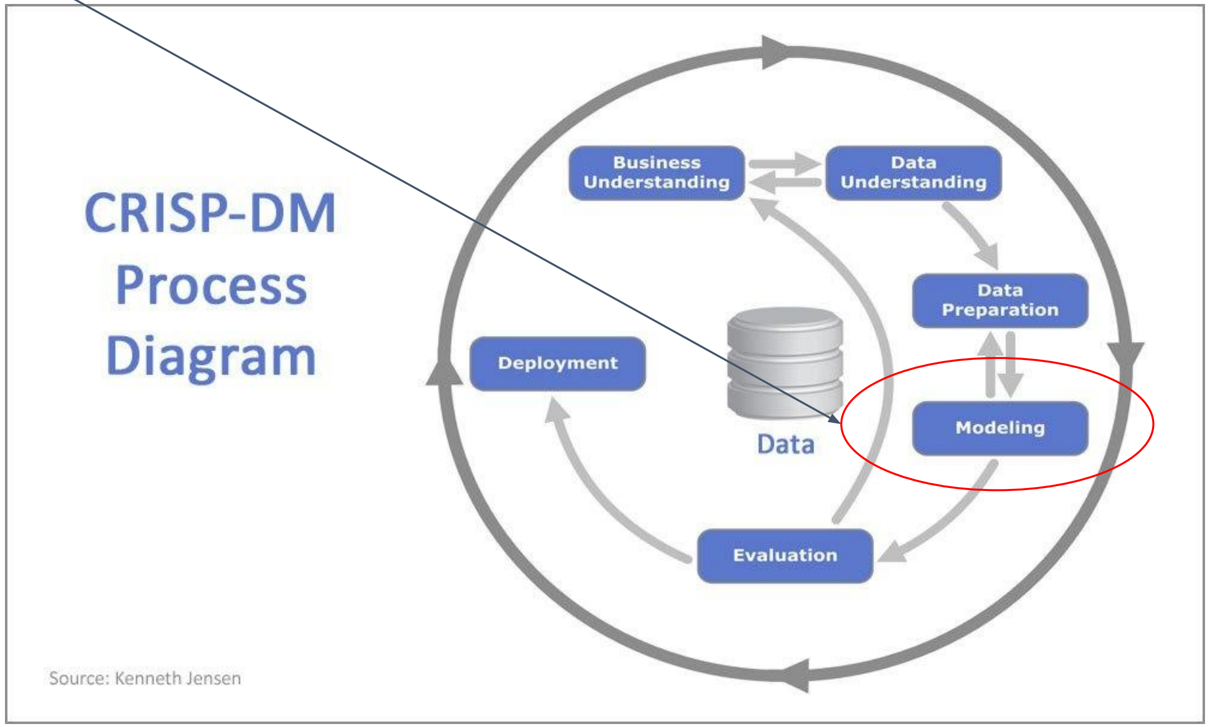
Generalization, Underfitting, Overfitting

Data Science Program

Purwadhika
Startup and Coding School

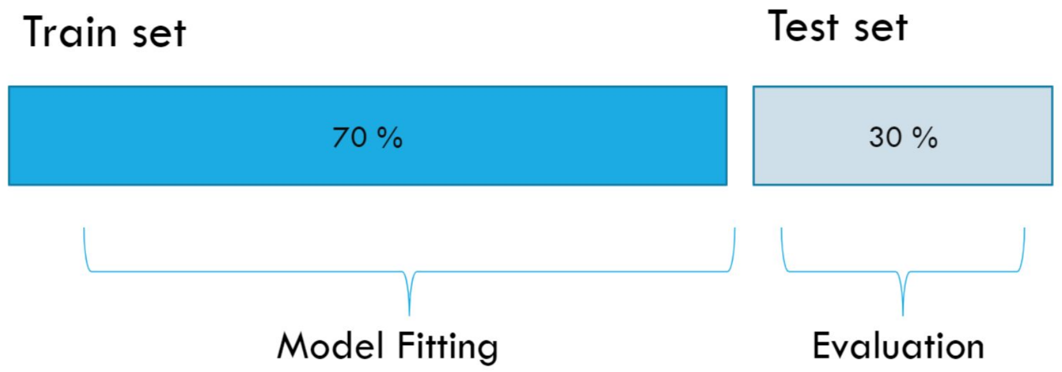
Outline

- What is Generalization ?
- What are Overfitting and Underfitting ?
- Model Complexity vs Performance
- Illustration of Generalization, Overfitting and Underfitting:
  - KNN
  - Decision Tree
- Generalization in Linear Model



What is Generalization ?

- In supervised learning, we build model on a dataset (seen data) and then be able to make accurate predictions on new data (unseen data)
- Thus, we need to divide data into two set, training set and test set
- **Training set** is used to **fit** the **model** and from **test set** we can **infer** the ML algorithm **performance**
- When building any model, test data can’t get involved at all
- If a model is able to make accurate predictions on the unseen data, we say it is able to generalize from the training set to the test set



Why is Generalization ?

- We are interested in the accuracy of the prediction that we obtain when we apply our method to new unseen data
- In practice, We might try several different method
- No one method dominates all others over all possible data set
- We want to build an ML that is able to generalize as accurately as possible

Illustration : Predicting Customer Who will buy a boat

Age	Number of cars owned	Owns house	Number of children	Marital status	Owns a dog	Bought a boat
66	1	yes	2	widowed	no	yes
52	2	yes	3	married	no	yes
22	0	no	0	married	yes	no
25	1	no	1	single	no	no
44	0	no	2	divorced	yes	no
39	1	yes	2	married	yes	no
26	1	no	2	single	no	no
40	3	yes	1	married	yes	no
53	2	yes	2	divorced	no	yes
64	2	yes	3	divorced	no	no
58	2	yes	2	married	yes	yes
33	1	no	1	single	no	no

Goal :
- send promotional email to people who are likely to actually make a purchase

Let's build some rule:
- if the customer is older than 45, has less than 3 children and is not in divorce, then they want to buy a boat. (this result 100% accurate)
- and still so many rules you can find by looking at this dataset
- Which rule could generalize new unseen data ?

Purwadhika
Startup and Coding School

What are Underfitting and Overfitting ?

Age	Number of cars owned	Owns house	Number of children	Marital status	Owns a dog	Bought a boat
66	1	yes	2	widowed	no	yes
52	2	yes	3	married	no	yes
22	0	no	0	married	yes	no
25	1	no	1	single	no	no
44	0	no	2	divorced	yes	no
39	1	yes	2	married	yes	no
26	1	no	2	single	no	no
40	3	yes	1	married	yes	no
53	2	yes	2	divorced	no	yes
64	2	yes	3	divorced	no	no
58	2	yes	2	married	yes	yes
33	1	no	1	single	no	no

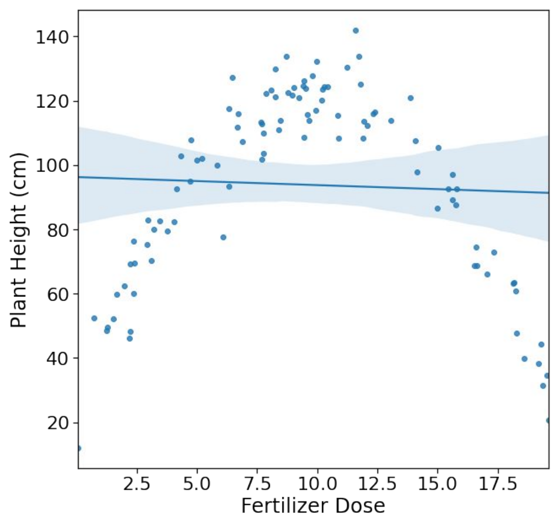
Underfitting :
- too simple
- “Everybody who owns a house buys a boat”

Overfitting:
- too complex
- “the customer older than 45, has less than 3 children and is not n divorce, then they want to buy a boat”

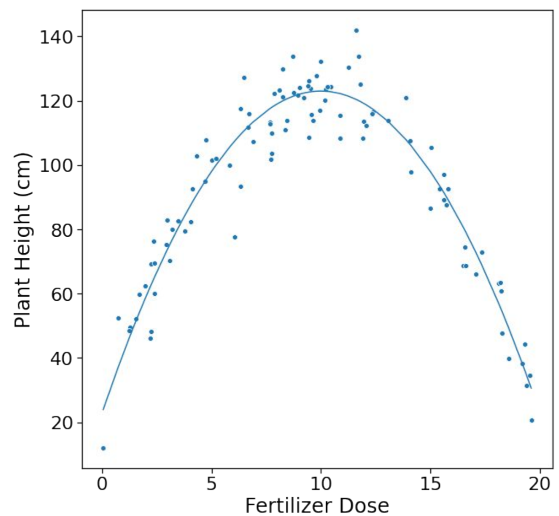
Purwadhika
Startup and Coding School

Capturing Underlying Data Trends

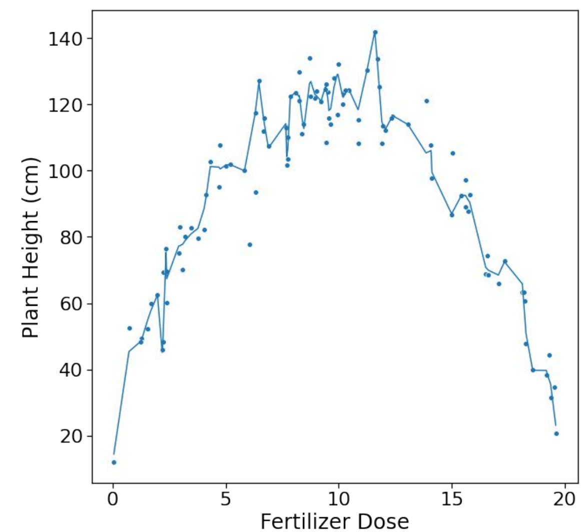
Which one is your preference?



Underfitting Model:
y = a + bx



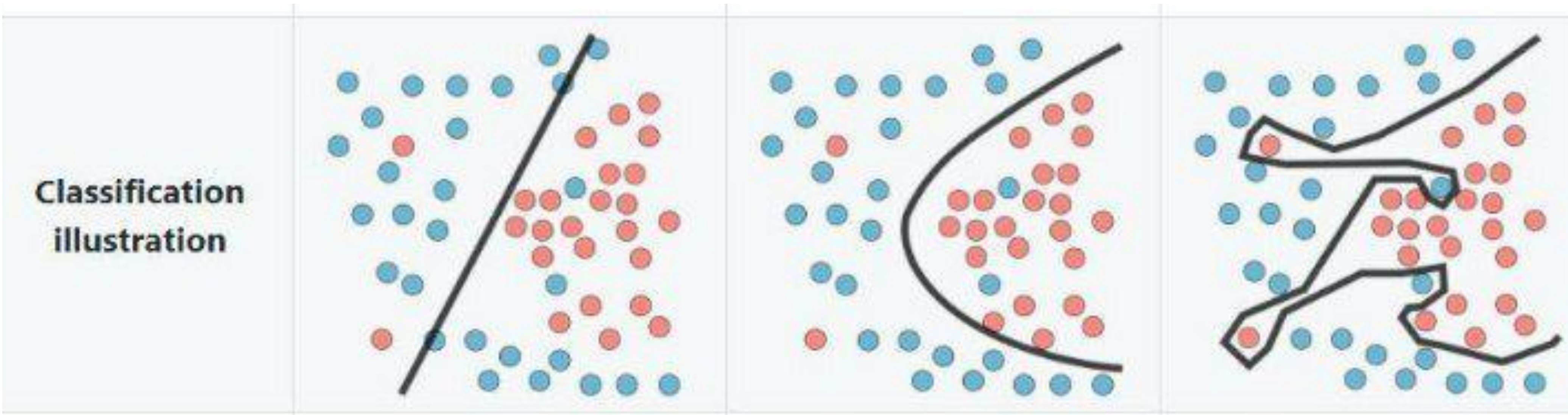
Just Right:
y = a + bx + cx^2



Overfitting Model:
lowess regression

Purwadhika
Startup and Coding School

Capturing Underlying Data Trends

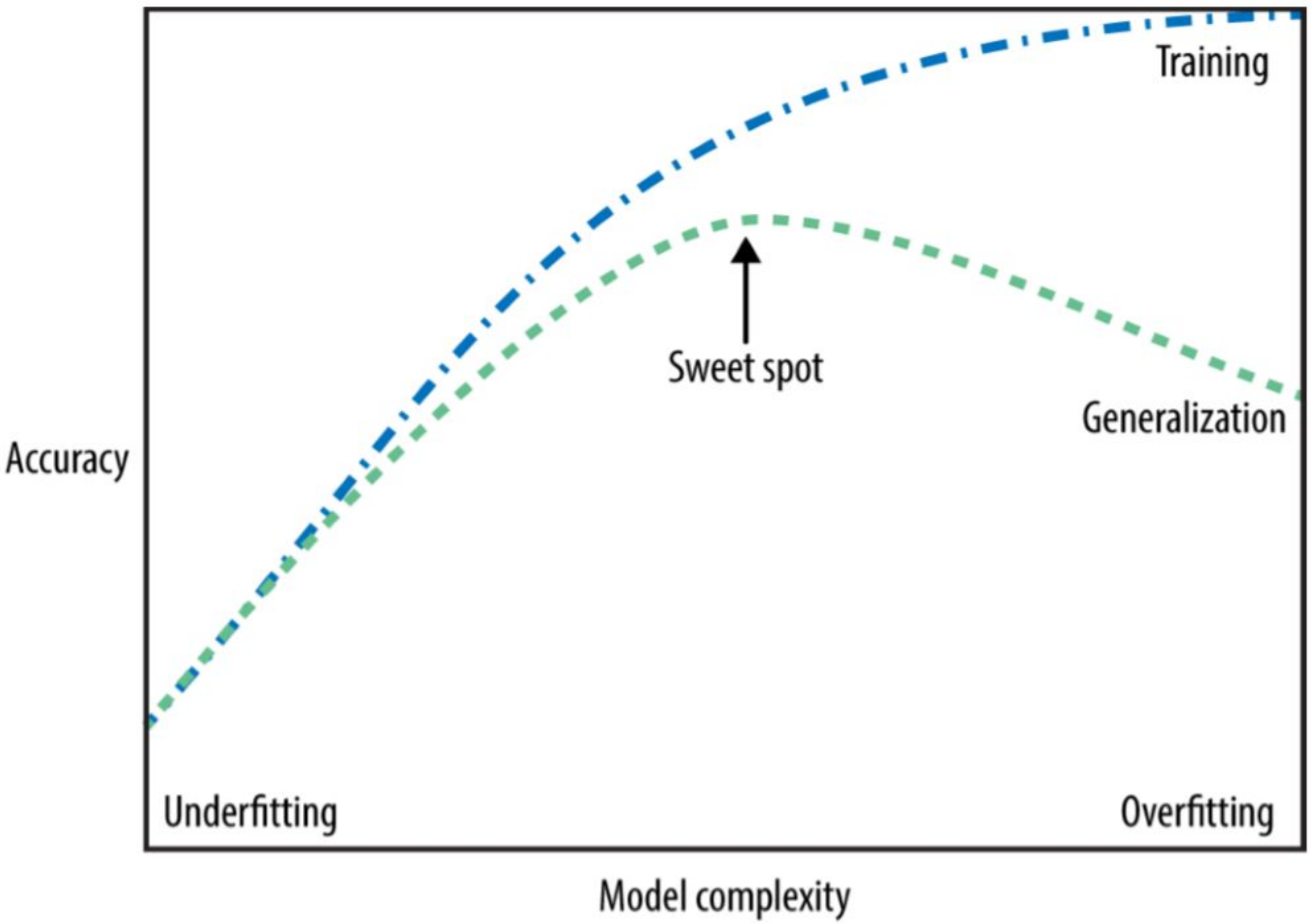


Underfitting Model | Just Right | Overfitting Model

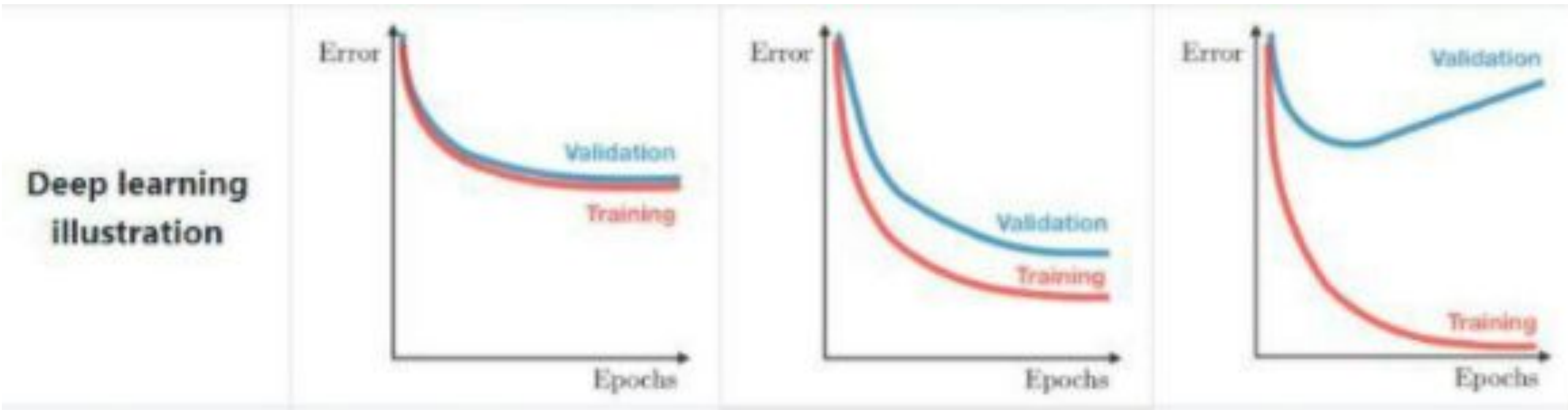
Purwadhika
Startup and Coding School

Model Complexity vs Model Performance

- High Training Set Error and High Validation Set Error ->
  - High Bias (Underfitting)
- Low Training Set Error and High Validation Set Error ->
  - High Variance (Overfitting)
- There is a sweet spot in between that will yield the best generalization performance. This is the model we want to find.
- Data has strong relation to model complexity : If more data means more variation, More variation in data allow you to train more complex model without overfitting



Overfitting and Underfitting in Deep Learning



Overfitting and Underfitting Recap

OVERFITTING

- Too complex
- Fits the training data too well
- Good performance in training set but bad at test set
- High variance, low bias

Solutions:

✔ Reduce number of features

✔ Increase training samples

UNDERFITTING

- Too Simple
- Doesn't capture underlying data trend
- Bad performance both at training set and test set
- High bias, low variance

Solutions:

✔ Train more complex model

✔ Obtain more features

A good model has slightly lower training error than test error

Purwadhika
Startup and Coding School

Underfitting and Overfitting in KNN

Analyze data bankloan.csv

- Apply KNN Classifier
  - target : default
  - features : employ, debtinc, creddebt, othdebt
- Using different k (1,3,5,...100) : Apply scaling and Validate the model using accuracy in 20% testing data and 80% training data
- compare accuracies obtained from training data and testing data

Purwadhika
Startup and Coding School

Model Complexity in Decision Tree

Analyze data bankloan.csv

- Apply Decision Tree Classifier
  - target : default
  - features : employ, debtinc, creddebt, othdebt
- Using different maximum depth of the tree (1,2,3,...25) : Validate the model using accuracy in 20% testing data and 80% training data
- compare accuracies obtained from training data and testing data
- you may try another hyperparameter such as minimum samples split, minimum samples leaf, etc.

Generalization in Linear Model

- Too many feature used in linear models makes model more complex and may leads to overfitting
- We can either use :
  - reduce the effect/magnitude of certain features (Ridge)
  - make zero effect/magnitude for certain features (Lasso)
- Ridge or Lasso can be used as a solution to multicollinearity
- Which one to use ?
  - as the simplest way, you can directly check the performance on test data

Purwadhika
Startup and Coding School

Ridge (L2 Regularization)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k + \varepsilon$$

Reduce the magnitude close to zero

Formula :

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

Hyperparameter Properties

1. Alpha : typically used {....1000,100,10,1,0.1,0.01,0.0001,...}
2. Increasing alpha
   a. forces coefficients to move more toward zero but never zero
   b. reduce model complexity

Purwadhika
Startup and Coding School

Lasso (L1 Regularization)

$$\boxed{Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \varepsilon}$$

Reduce some of the magnitude to zero

Formula :

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \mathrm{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

Hyperparameter Properties

1. Alpha : typically used {....1000,100,10,1,0.1,0.01,0.0001,...}
2. increasing alpha
   a. less feature used because of zero magnitude
   b. reduce model complexity
   c. getting easier to interpret

Purwadhika
Startup and Coding School

Model Complexity in Ridge

Analyze data boston dataset from sklearn

- Apply Decision Tree Classifier
  - target : target (house price)
  - features : CRIM, ZN, INDUS, CHAS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B, LSTAT, MEDV
- Using different alpha (100000, 10000, 1000, 100, 10, 1, 0.1, 0.001) : Validate the model using mse in 20% testing data and 80% training data
- compare mse obtained from training data and testing data

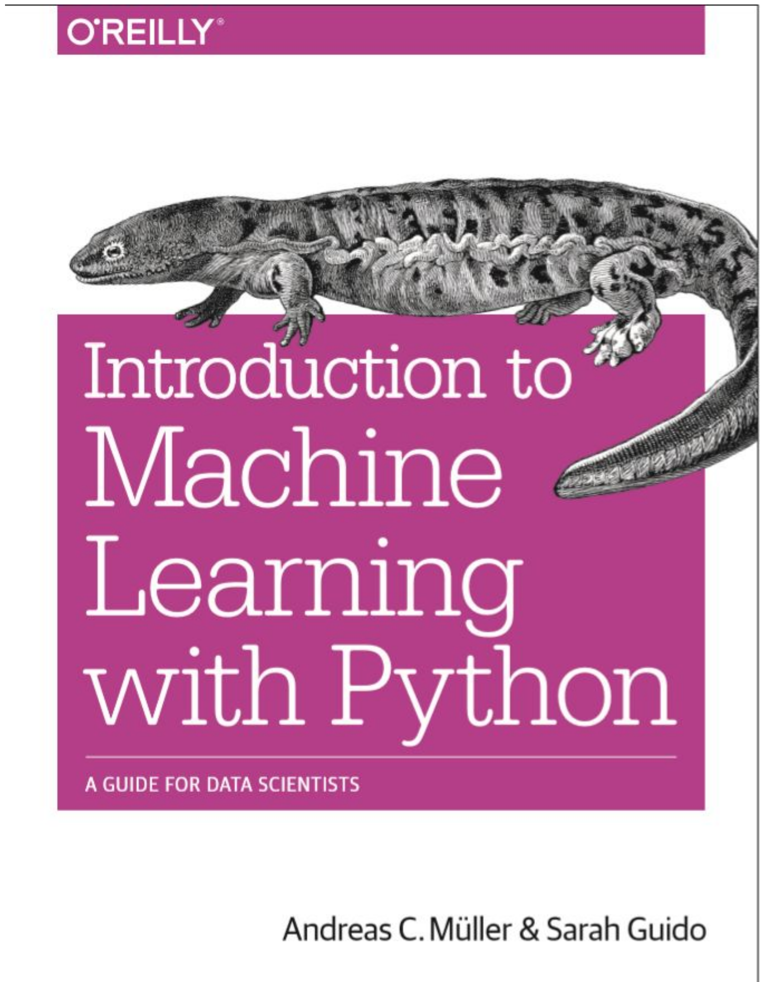
Model Complexity in Lasso

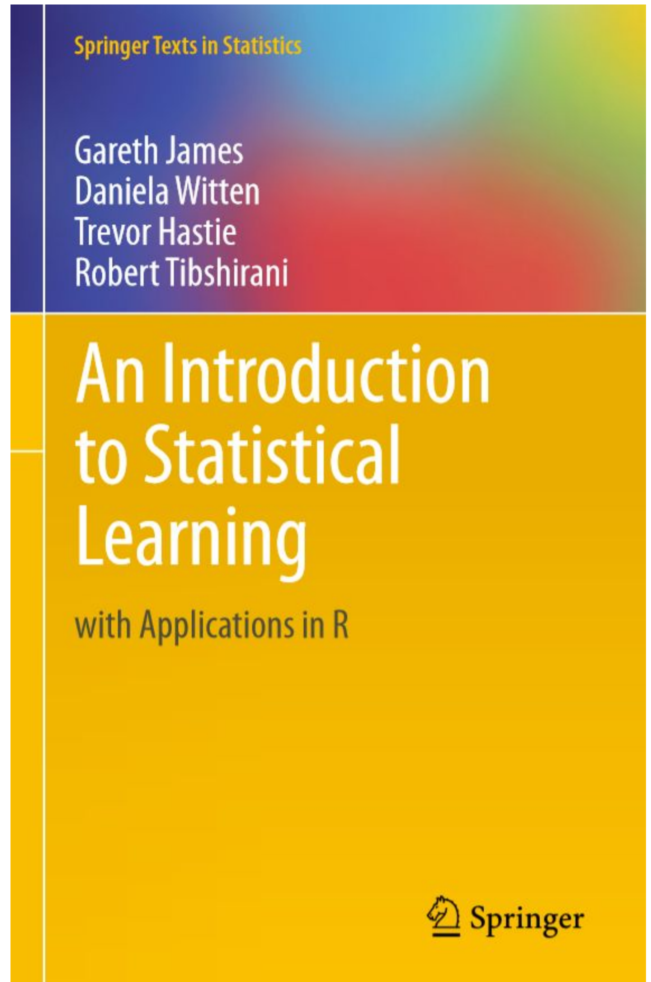
Analyze data boston dataset from sklearn

- Apply Decision Tree Classifier
  - target : target (house price)
  - features : CRIM, ZN, INDUS, CHAS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B, LSTAT, MEDV
- Using different alpha (100000, 10000, 1000, 100, 10, 1, 0.1, 0.001) : Validate the model using mse in 20% testing data and 80% training data
- compare mse obtained from training data and testing data

Purwadhika
Startup and Coding School

References





Purwadhika
Startup and Coding School

References

https://www.the-modeling-agency.com/crisp-dm.pdf

https://scikit-learn.org/stable/

Purwadhika
Startup and Coding School