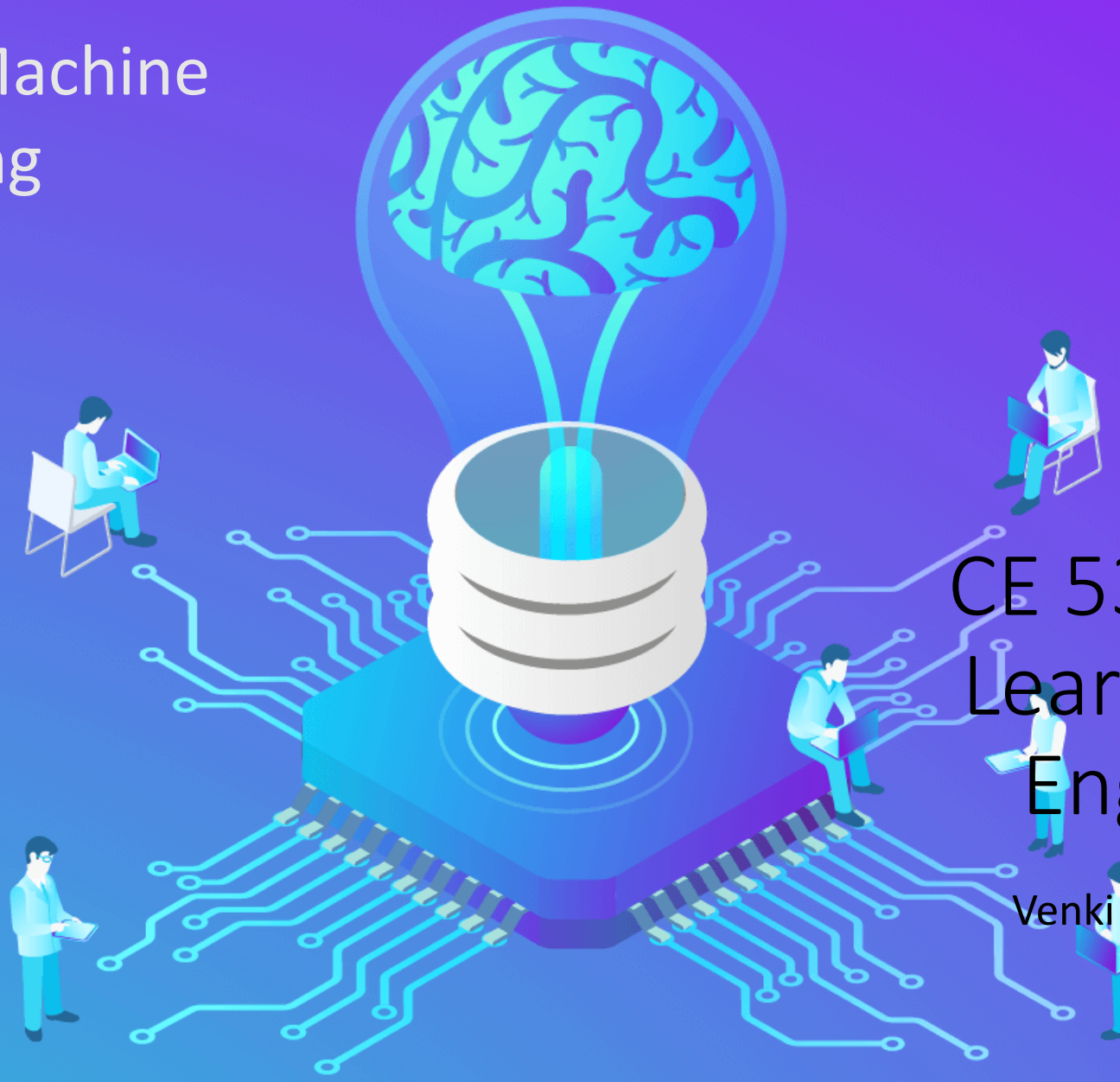
Python for Machine Learning



CE 5331 Machine Learning for Civil Engineers

Venki Uddameri, Ph.D. , P.E.

Regression

- Statistical Data-Driven modeling tool
  - Establishes relationships between inputs and output
  - Can be linear or nonlinear
- The inputs are either continuous or discrete
- The outputs can either be
  - Continuous
  - Discrete – Binary – Logistic regression
  - Multiple levels – Multinomial regression (ordered or unordered)
  - Count – Poisson regression

Regression

- Input parameters are assumed to be fully known
  - No errors in them
- Output parameter is a random variable
- The difference between the observed and predicted is the residual or error term
  - Error term is white-noise
  - Has no information
  - Is normally distributed with a zero mean and a constant variance

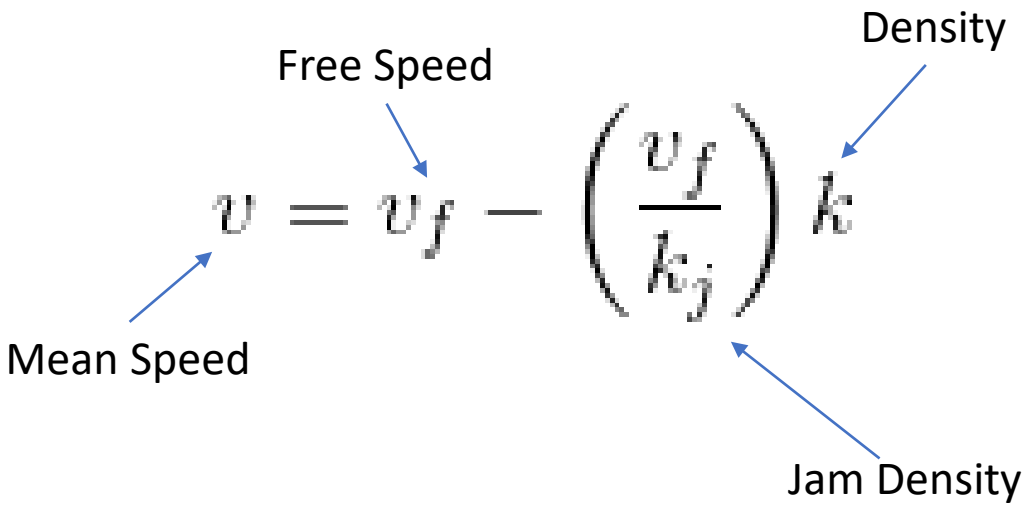
Linear Regression

- Linear Regression coefficients are often estimated using ordinary least squares (OLS) minimization
  - Maximum likelihood can also be used.
- OLS provides the best linear unbiased estimates (BLUE)
- Inference of these parameters however requires certain assumptions be met
  - The residuals are not autocorrelated
  - Residuals are not heteroskedastic
    - Have constant variance
  - The residuals are normally distributed
- For multivariate models it is also important that the independent variables do not exhibit multicolinearity

Regression Example

Optimization - Regression

- The Greenshields model provides a relationship between mean traffic speed and density in an uninterrupted section as follows:



Empirical Data is used to calibrate the Greenshields model

Notice the linear relationship between speed and density

Linear Regression is used when there are more data than unknowns

The unknowns are obtained in a best-fit sense

The sum of squared residuals (SSR) is minimized to obtain the best fit parameters

Fit the Greenshields Model using the rural traffic dataset provided to you (ruraldensityspeed.csv)

Modeling Approach

- Read the data in
  - Pandas library
- Write a function to calculate the SSE
- Specify initial guesses for A and B
- Minimize the SSE function to find optimal values of A and B

$$v = v_f - \left(\frac{v_f}{k_j}\right) k$$

Original Model

$$v_{pred} = A + Bk + e$$

Regression Form
(A and B are unknown coefficients)

$$e_i = v_{obs,i} - v_{pred,i}$$

Error term

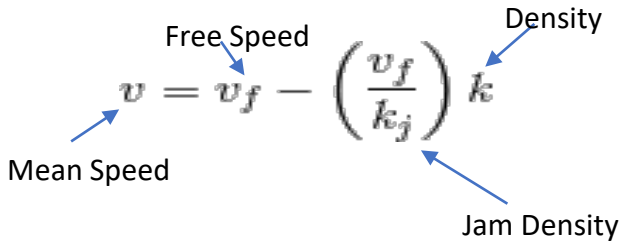
Objective Function minimizes the sum of squared error term

$$SSE = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (v_{obs,i} - v_{pred,i})^2 = \sum_{i=1}^{N} (v_{obs,i} - [A + Bx_i])^2$$

This is also referred to as the loss function in Machine Learning Literature

Linear Regression using Unconstrained Optimization

Slope = -0.53
Intercept = 62.56



Therefore – Free speed = 62.65 mph and Jam Density is 118.47 vehicles/mile/lane

You can also use linregress function in scipy stats module to perform linear regression

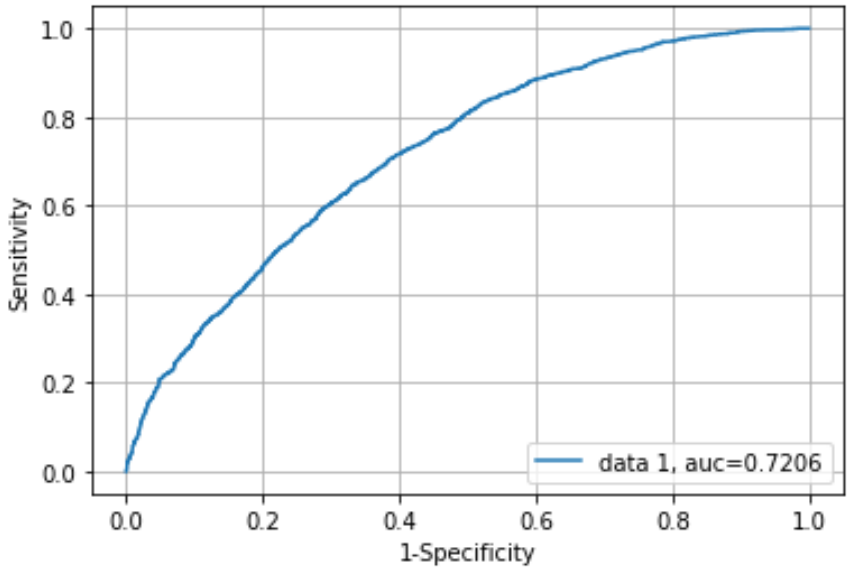
```
# Use scipy stats model to perform linear regression
# Now you can extract statistics as well
from scipy import stats
slope, intercept, r_value, p_value, std_err =
stats.linregress(k,vobs)
round(slope,2), round(intercept,2)
```

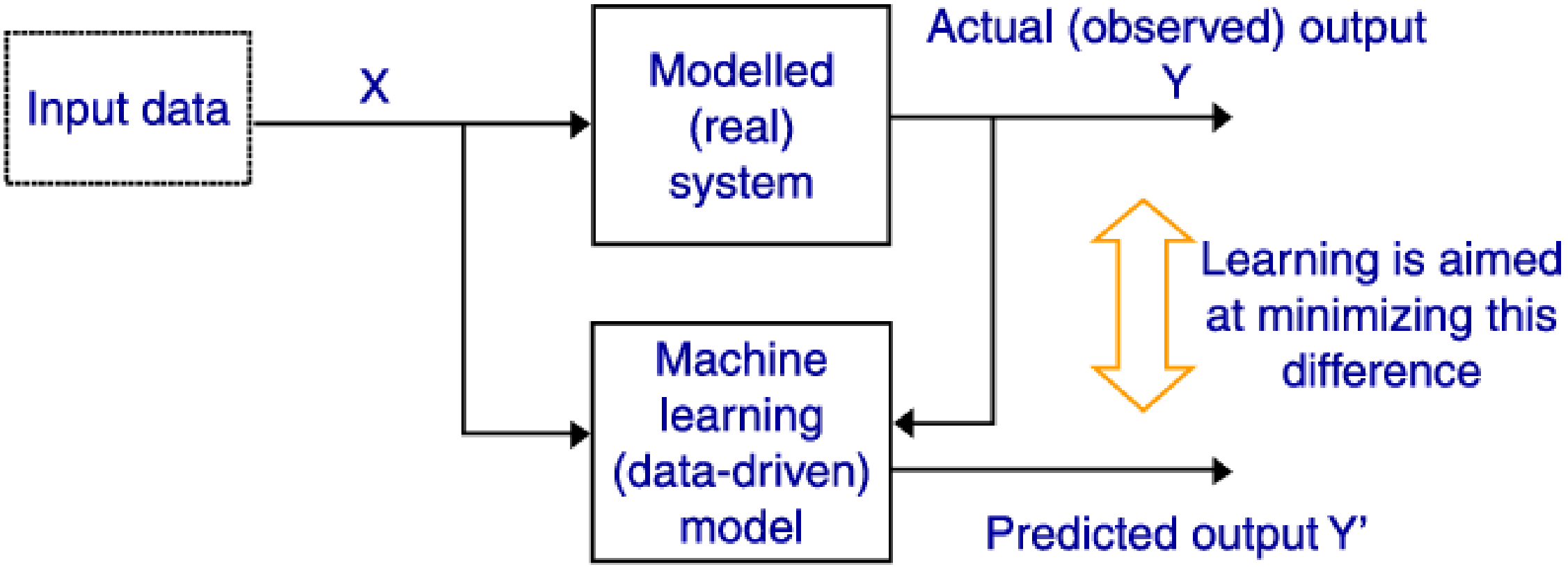
We will study linear regression in greater depth in the course

```
# Linear Regression using Unconstrained Optimization
# Venki Uddameri, TTU
# Step 1: Load Libraries
import os
import numpy as np
import pandas as pd
from scipy.optimize import minimize

# Set working directory
os.chdir('D:\\Dropbox\\000CE5333Machine Learning\\Module9\\Codes')

# Read data from csv file and extract variables
a = pd.read_csv('ruraldensityspeed.csv')
vobs = a['Speed']
k = a['Density']

# Define function for computing SSE
def funsse(A,k,vobs):
    pred = A[0] + A[1]*k
    err = (vobs-pred)**2
    sse = np.sum(err)
    return(sse)
# Call minimize functions
init = (1,1) # Starting values for slope and intercept
res = minimize(funsse,init,method='Nelder-Mead',args=(k,vobs,))
res.x # Write slope and intercept to the console
```

Linear Regression - statsmodels

- The library statsmodels provides another convenient way to perform OLS in python
  - Produces R like output
- Provides several important measures to evaluate the model





Inputs and Outputs are the Key Elements of Data Driven (Machine Learning) Models

Some Guidelines – Inputs and Output

- The output is a function of the objectives of your study
  - Want to predict flood characteristics?
    - Peak flow, flood duration, total flood volume?
- A candidate set of inputs can be ascertained based on the following
  - What does the theory tell us?
    - Science Inspired
  - What parameters do we have the data for?
    - May not always be the best approach
  - Are there good surrogates for some parameters the theory says is important but we don't have the data for?
- Are the data we have independent?
  - Avoid inputs with significant multicollinearity problems
- Do we have too many variables?
  - Use a data reduction technique such as PCA

Sensing, statistics and scientific considerations must be jointly evaluated to select model inputs

Guidelines – Inputs and Outputs

- Two basic approaches
  - Start with a full model and remove variables as necessary
    - Backward selection
  - Start with an empty model and add parameters as necessary
    - Forward selection
- Need some objective criteria in either cases
  - Does it improve the fit ? (how much reduction in RSS)
  - Is the complexity warranted (how much penalty for nonparsimony)
  - Akaike Information Criterion is one such metric
    - Model with lowest AIC or AICc
  - Statistical methods including p-values are another

$AIC = 2k - 2\ln(L)$ or $AIC = 2k + n\ln(RSS)$ for small samples, $AICc = AIC + (2k^2+2k)/(n-k-1)$

Which technique to use

- There are many ML algorithms so how to justify which one to choose
- Never choose a single algorithm (say ANN) always try to use more than one
- Justify the selection of your choice
  - Previous studies in your area perhaps to a different problem
  - Previous studies outside your area but with similar situations
    - Highly nonlinear relationships, datasets exhibit similar characteristics that do not satisfy statistical assumptions
- Always include current state of the practice (e.g., linear regression, logistic regression) as part of your comparison
  - If a ML model does as good or worse then a simple regression than there is no need to do ML
- Compare and contrast what a particular technique gives you
  - Superior forecasts, is blackbox ok?
  - Do we get insights on inner workings and mechnisms underlying the data?

Have a clear understanding of what is that you want out of the machine learning exercise

Structure of the Model

- Structure of the model is tied to number of inputs but also includes other choices made during the modeling process
  - These choices are often referred to as ‘Hyper-parameters’
  - Internal workings of the model
- Try to understand what these ‘hyper-parameters’ parameters do
- Perform sensitivity analysis
  - Try multiple methods to make sure which one gives better results
  - For examples try multiple optimization methods for parameter estimation if choices are available.
- No clear guidance is available on what the structure of the model should be
  - Complex models → more parameters → difficult to estimate
  - Simple models → less parameters → easy to estimate → Are they adequate?

Models should be made simple but not any simpler (Albert Einstein)

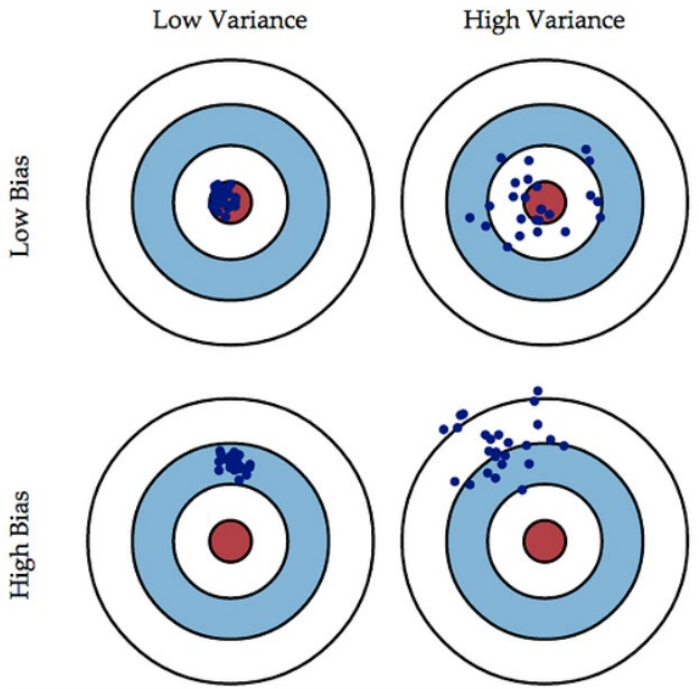
How should we train the models

- All machine learning models require training
- Training is providing data to the model and have it establish required relationships
  - Model training implies estimating unknown model parameters
- Which data should be used for training and which should be used for testing?
  - Training data should cover the range of values over which the model will be used.
    - If not you are asking the model to extrapolate

Data for model training

- Split the data into 2 random sets one for training and one for testing
  - Any where from 60:40 – 80:20
  - Have at least 20% in the testing dataset
  - Widely used. However, the splits can be different (due to randomness) and could yield different fits
- Split the data into 3 random sets – Training + Testing + Evaluation
  - Train the Data to train the model
    - 40% – 50%
  - Testing data is used to provide an unbiased evaluation of the trained model and further adjust the parameters
    - Avoid over-fitting and select a final (optimal) model
    - 10% - 20%
  - Evaluation (hold-out) is used to independently check the final model
    - 30% - 40%

K-Fold Cross-Validation



- Randomly Split the data into K-equal parts
  - Hold back Kth part for testing
  - Use K-1 parts for training
  - Repeat over all parts
- Useful when the data are limited
  - The amount of data controls what K should be
  - If K = 1 then it is called jack-knifing
- The data is used K -1 times for training and 1 time for testing.
- Typically K value is taken to be between 5 – 10
  - Experimentation may be necessary to find optimal K
    - Bias is used as a measure to identify an optimal K
  - Larger K → smaller difference in size between training and testing
    - Lesser bias
- K-fold validation allows you to compute variance associated with your prediction errors

Evaluating Models

- The metrics used to evaluate the model must match the objectives of building the model
- Common to test
  - Root mean square error – Penalty for very large errors
    - Nash-Sutclife Efficiency metric is a variant of this criteria
  - Mean Absolute deviation – Penalty is more even across both small and large errors
  - Correlation – Is the model capturing linear or monotonic trend
  - Bias – Does the model over-estimate or under-estimate in all
    - A good model with random errors should neither over or underestimate
- You can evaluate model for various conditions
  - Peak flows, flow volume
- You can compute these metrics for both training and testing datasests

Model Over-Fitting

- Model over-fitting occurs when the error associated with training is far less than that for testing
- Over-fitting implies the model has learned the training data but is unable to generalize to testing data
- Over-fitting is a general problem that arises when we use training and testing data
  - The issue of over-fitting is less when using
    - Training + Testing + Evaluation
    - K-fold cross-validation

Some Final Thoughts

- Developing Machine learning models is both an art and science
- Don’t simply use a ML technique because it is new or someone else has used it
- Make sure you have the objectives of your study clear before you embark on modeling
  - What type of precision and accuracy do you want from these models? Why?
- Make sure to try multiple ML methods using standard methods (e.g., regression) as a baseline to compare
- Perform systematic exploration of model “hyper-parameters’ to find optimal
- Ensure you are using right combination of data splits for training and testing
  - 2 or 3 Splits or CV?
- Evaluate the model performance over a set of metrics
  - Check for over-fitting
- Clearly state the limitations of the model
  - All models are wrong, some are useful (Niels Bohr)