Python for AI and Machine Learning

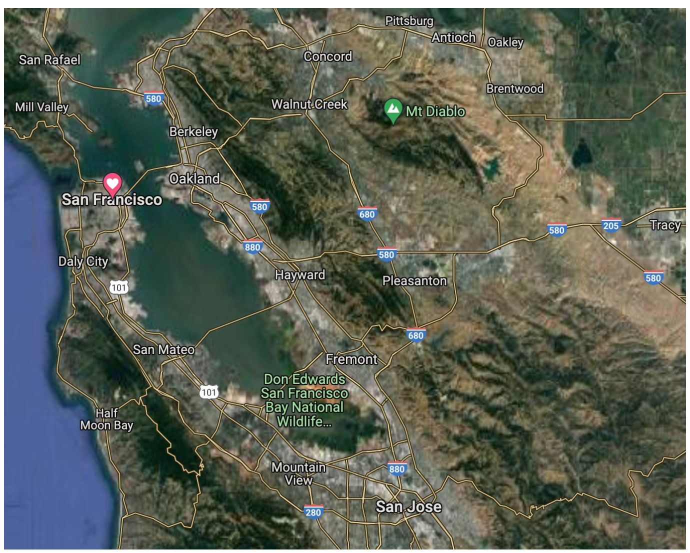
ML/AI Process & Concepts

Part 1

Marilson Campos

UCSC Extension - 2022

Why do we need models?



Computational Models

- Optimization Models
  - Objective function maximize/minimize
  - Solutions limited by a set of constraints.
- Simulation Models
  - Simulate the execution of real cases using randomizations
- Statistical Models
  - Machine Learning models.

Problem-Solving: Traditional Vs AI/ML

- Traditional problem solving
  - Start from Requirements
    - Derive Rules/Sequence of Commands
      - Test program against the set of known examples.

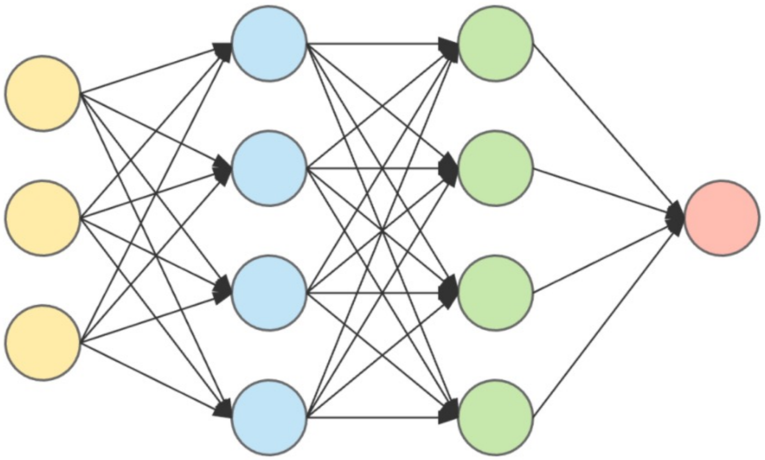
- AI/ML problem solving
  - Start from Data
    - Select ’model type’ and possibly the features to use.
      - Create pipeline to extract knowledge from data.
        - » Validate ‘learnings’ against new data

ML Model Example 1 – Linear Regression

- Assumes the answer is a linear combination of input values.
- Example:
  - Predict rent amount based on:
    - # of bedrooms (beds)
    - # of bathrooms (baths)
    - Neighborhood ‘score’: number from 1 to 5 (zone)
- After training, we could have learned the following formula:
  - Rent = (342 * beds) + (159 * baths) + (250 * zone)

ML Model Example 2 – Neural Network

- Model contains several ‘layers’ of ‘neurons interconnected.



- Example:
  - Recognize the zip code handwritten in a letter.

Terminology - 1

- Model (in ML/Ai) = The mathematical construct we use to learn and make inferences about data.
- Target Variable = The answer('output') of a model.
- Predictor = The inputs used to produce an output.
- Feature = A data element or some derivation of it used as a predictor.

Terminology - 2

- Cost function = A function that defines the ‘price’ of each error the model makes. The goal is to minimize the average price we pay for errors.

- Model Parameter = An internal value of a model that is unknown and will be ‘learned’ during model training.

- Model Hyperparameter = A parameter that the modeler provide to the model. Example: Number of trees to use in tree-based models.

Goal of Model Training

- Use data ‘examples’ to build an artifact (model) that captures the ‘general’ trends of that data.

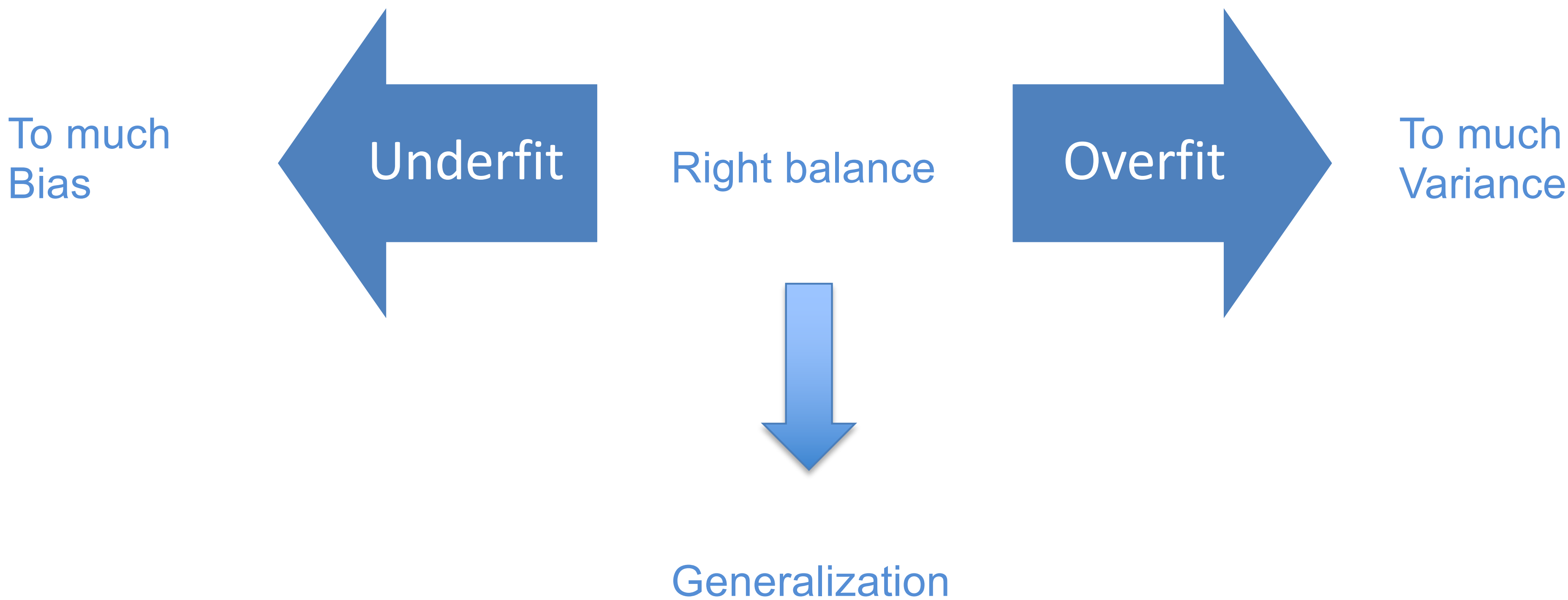
- Generalization is the process of extracting this underlying knowledge that goes beyond the ‘current data set’

Bias / Variance Trade off

- Bias: A model with high bias is more ‘stubborn’ and pays less attention to the dataset. It tends to oversimplify (underfit = adapt less)

- Variance: A model with high variance pays too much attention to the data. A single record can shift the model wildly. It tends to over complicate (overfit = adapt more)

Find the sweet spot in the Bias/Variance blend.

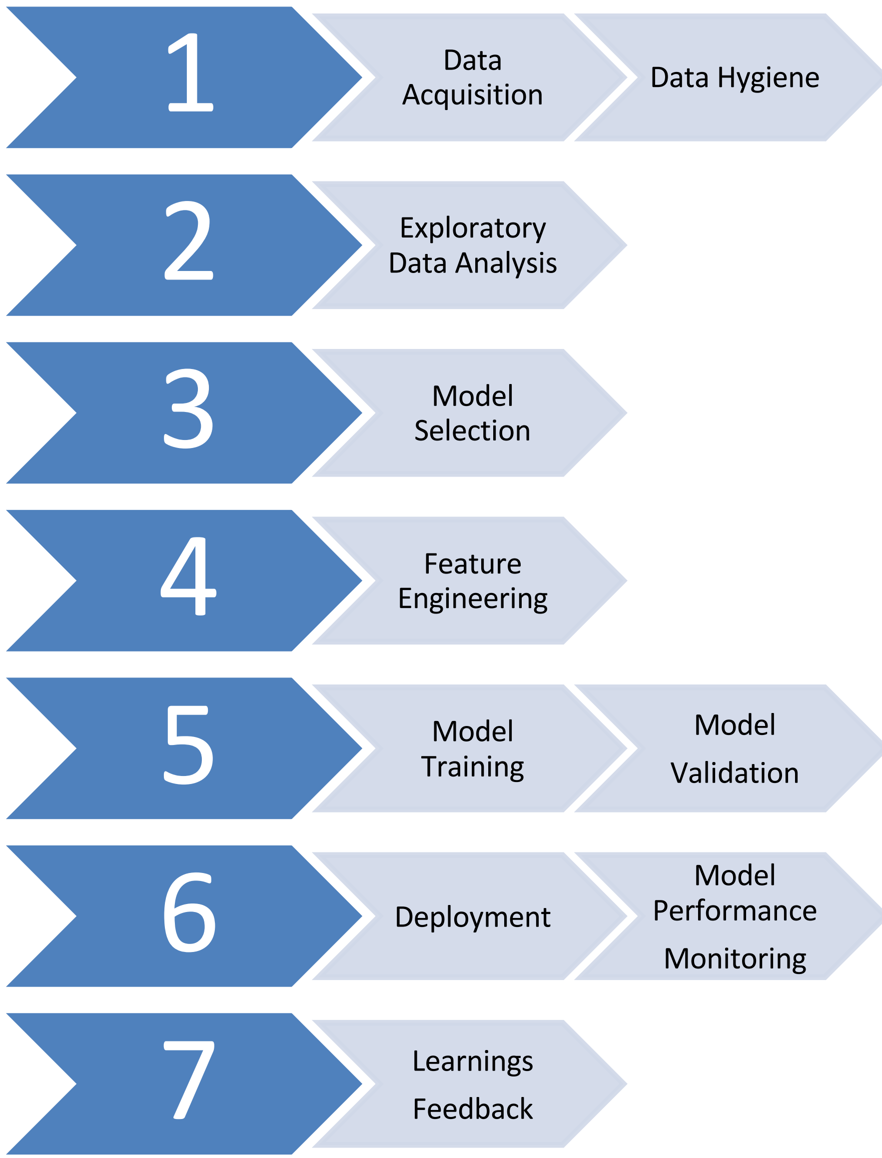


Bias / Variance Trade off

- How to control Bias/Variance
  - Option 1: Pick models that are flexible depending on the problem.
    - Ex: linear regression vs. tree-based models

- - Option 2: Regularization
  - Regularization allows you to put a price on model complexity. So simpler models are preferred unless the errors are a lot larger.

ML Model Lifecycle



Ensemble Models

- Ensemble ML models are models that build a set of ML models internally and use some form of voting to determine the result.

- 1. Bagging is ensemble algorithm that selects random subsets of a dataset and builds models for each of them and finally combine the results.

- 2 Random Forest is an extension of Bagging that select subsets of rows and subsets of columns and build models that are later combined.

- 3. Boosting is another ensemble algorithm that build models in sequence and the input of the next sequence is the 'error' (delta) from the previous one.