Machine Learning

B. N. M. Institute of Technology

Department: Artificial Intelligence and Machine Learning (AI)

Course: Machine Learning (18AI61) effective from the academic year 2018–2019 under VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

Instructor: Pradip Kumar Das

Machine Learning Landscape

What is Machine Learning?

Few Examples

Spam Email Filter

Recommender

Optical Character Recognition

Chatbots

Voice Recognition

Facial Recognition

Forecasting

Autonomous Vehicle

Medical Diagnostics

Fraud Detection

and many more...

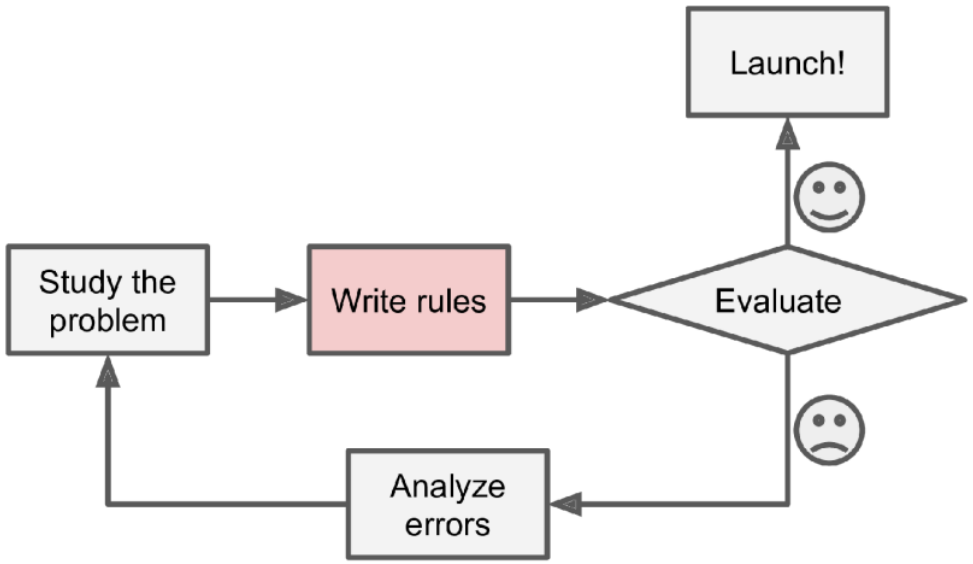
Some Definitions

"Science of programming computer to learn from data

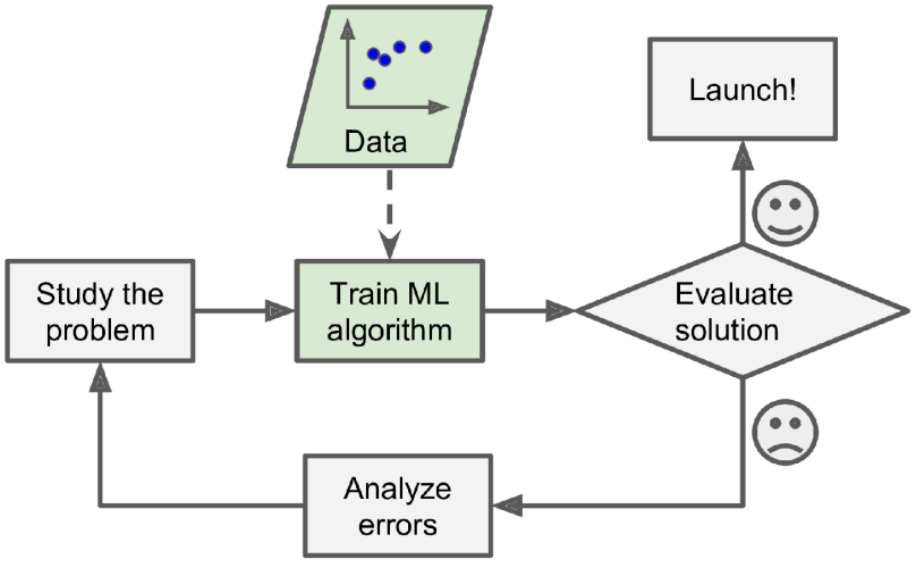
"field of study that gives computers the ability to learn without being explicitly programmed

*"A computer program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E***

Why is Machine Learning Required?

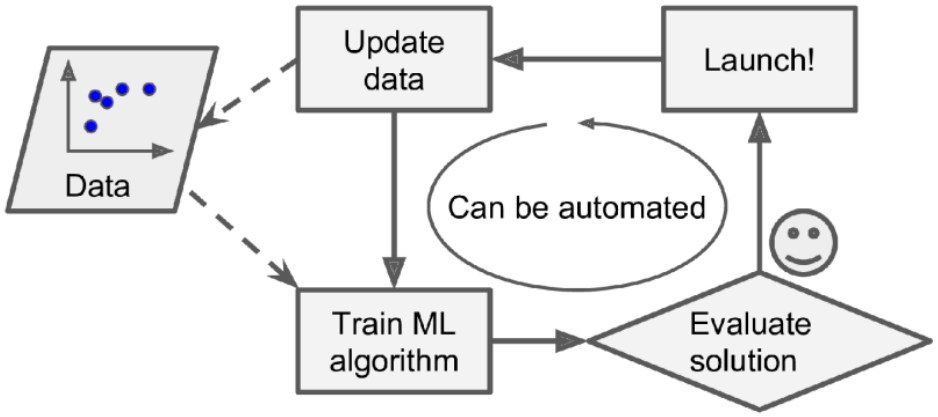


Traditional program writing approach

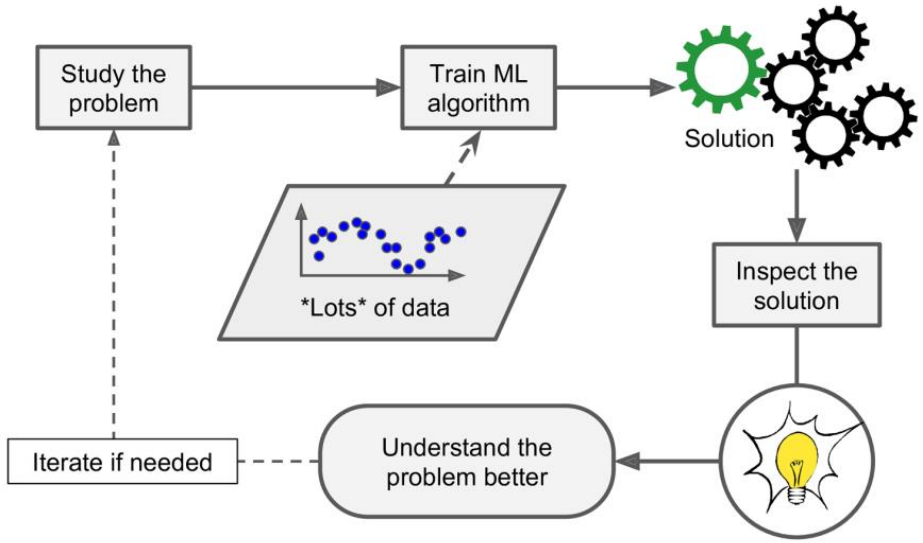


The Machine Learning approach

Why is Machine Learning Required? (Cont.)



Automatically adapting change



Discovering patterns for human learning

Machine Learning is a Good Option for

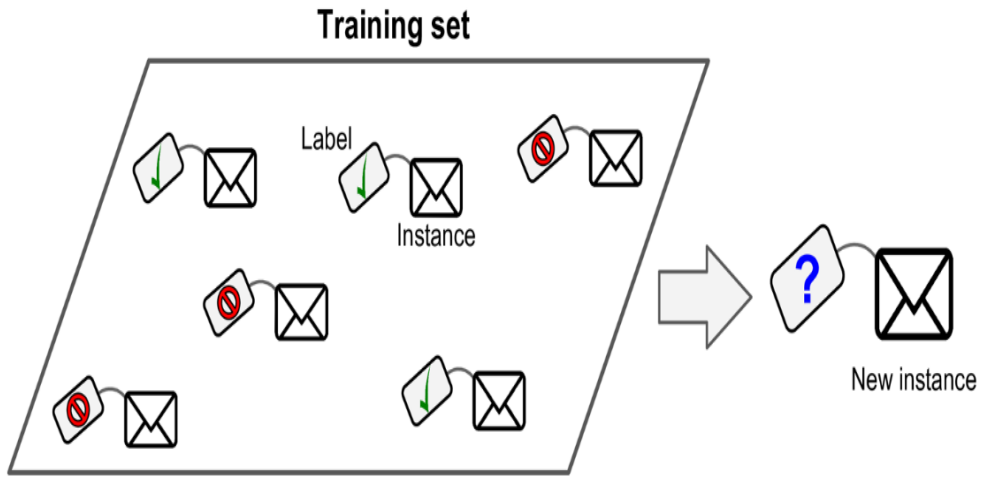
- Problems that require a lot of fine-tuning and long list of rules
- Problems that traditional approaches do not yield good results
- Problems where data distribution changes over time
- Getting insights about complex problems from large volume of data

Types of Machine Learning Systems

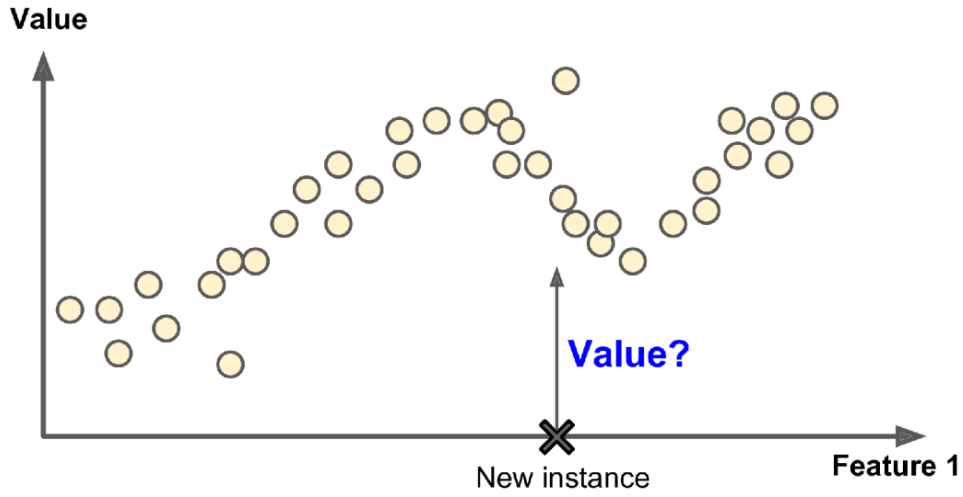
Whether learning requires human supervision

Types of Machine Learning Systems (Cont.)

Classification & Regression in Supervised Learning



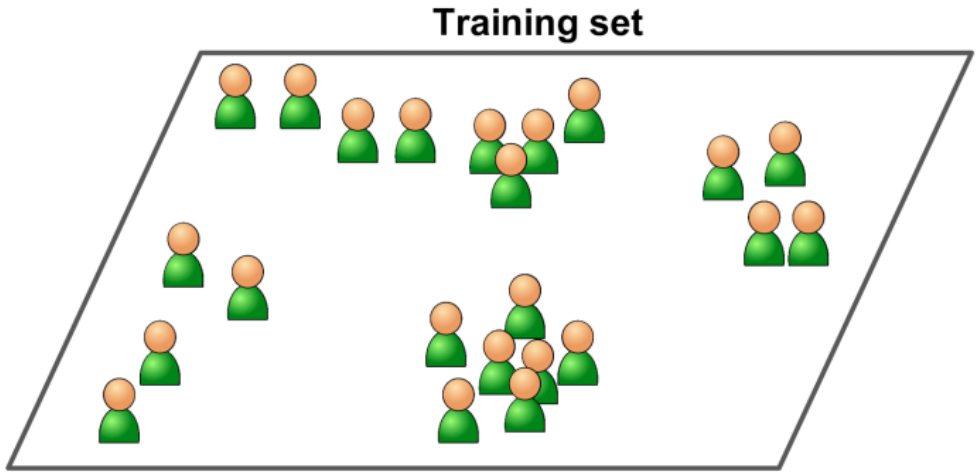
A labeled training set for spam classification



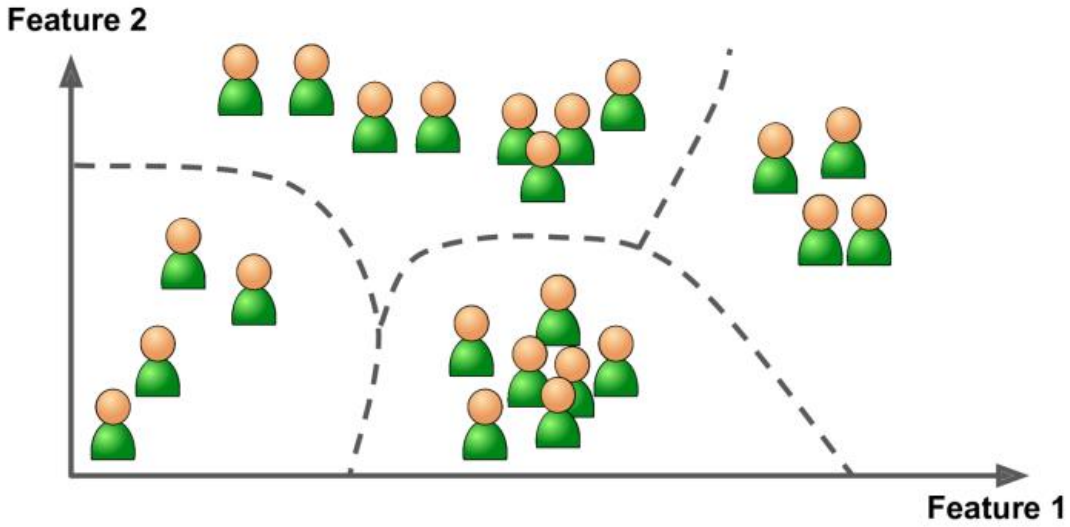
Predicting a continuous value in regression problem

Types of Machine Learning Systems (Cont.)

Clustering in Unsupervised Learning



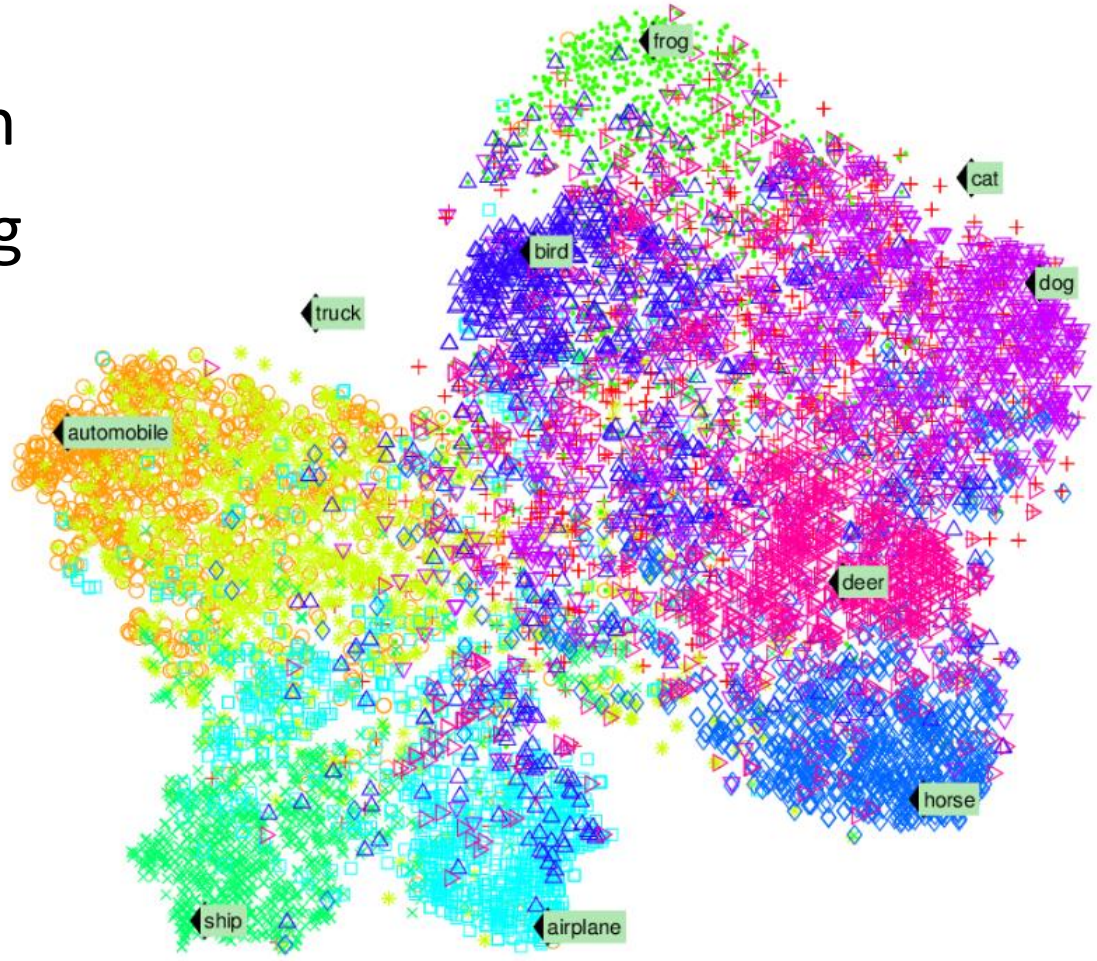
An unlabeled training set for clustering



Detecting groups of similar visitor in clustering

Types of Machine Learning Systems (Cont.)

Cluster Visualization in Unsupervised Learning



Cluster visualization in unsupervised learning

Types of Machine Learning Systems (Cont.)

Dimensionality Reduction in Unsupervised Learning

- Simplifying data without losing too much information
- Merging correlated features into one
- Feature extraction

Types of Machine Learning Systems (Cont.)

Anomaly and Novelty Detection in Unsupervised Learning

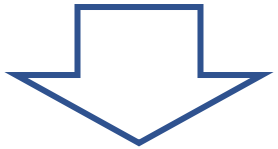
Detection of an anomaly

Types of Machine Learning Systems (Cont.)

Association Rule Learning in Unsupervised Learning

{Potatoes, Onions} => {Burger}

{Barbeque sauce, Potato chips} => {Steak}

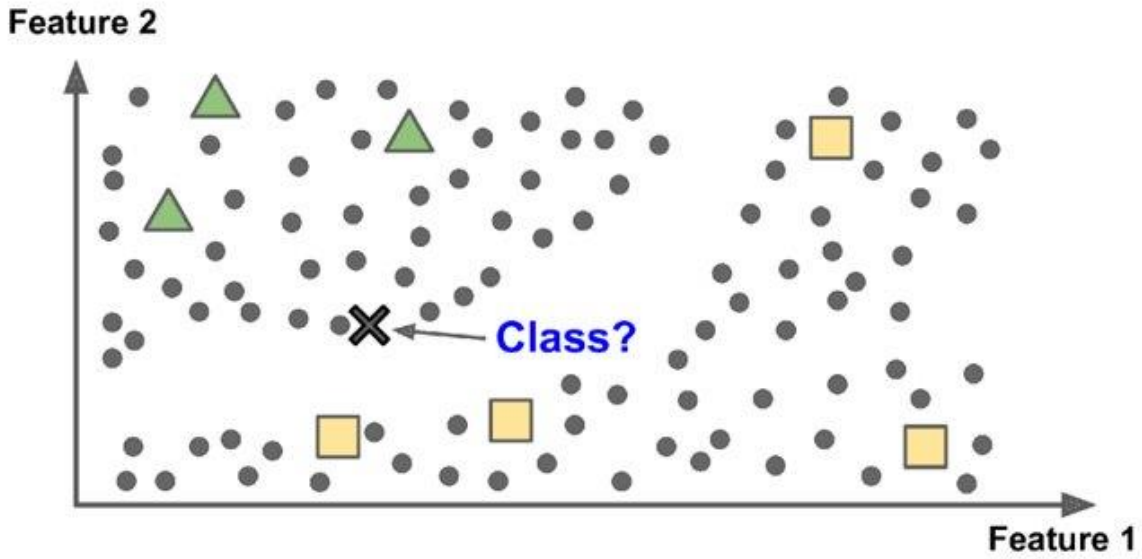


Results in Promotional Pricing and/or Appropriate Product Placement

Sales logs revealing customers' buying pattern

Types of Machine Learning Systems (Cont.)

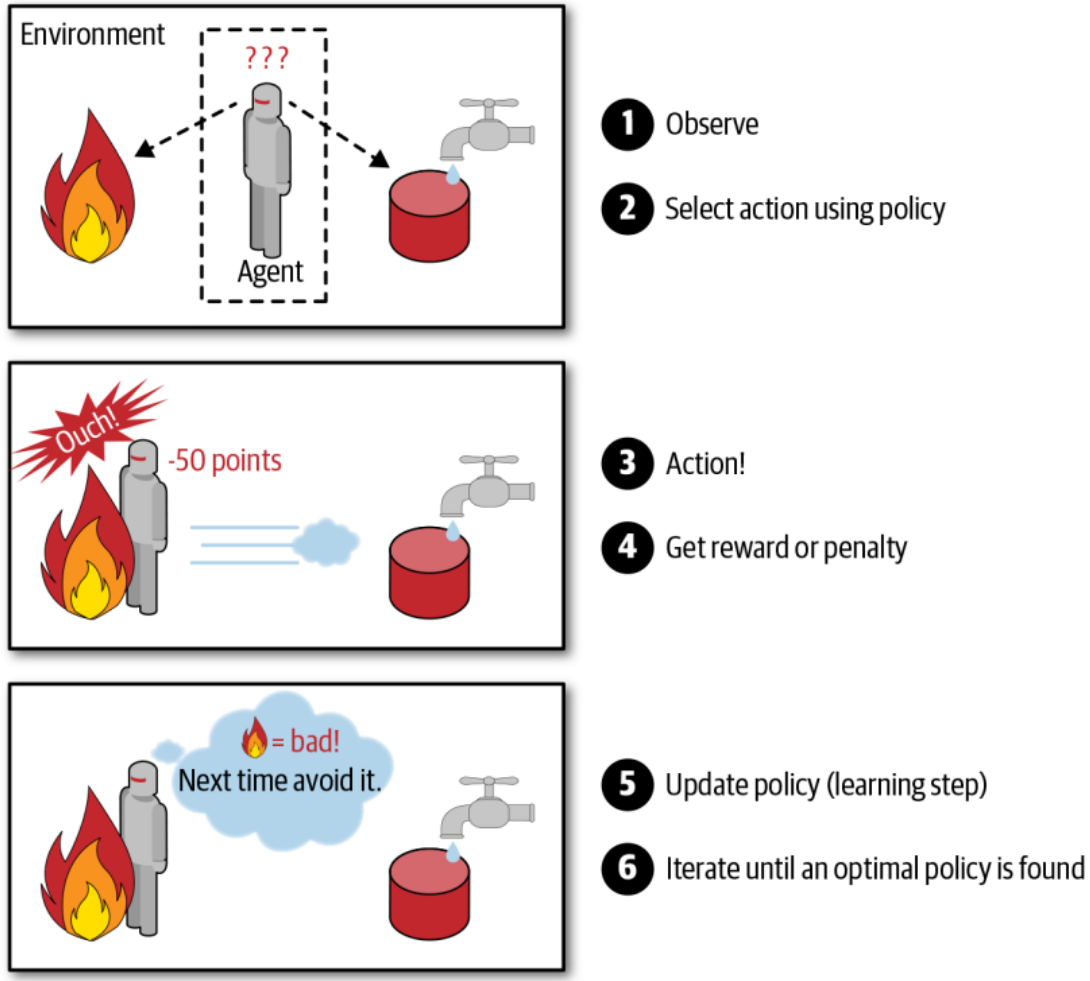
Classification in Semisupervised Learning



Few labeled examples help classifying new instances

Types of Machine Learning Systems (Cont.)

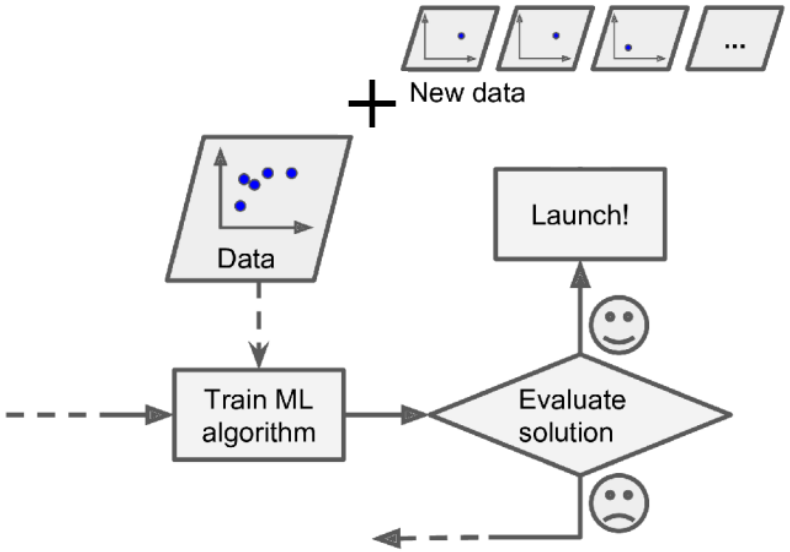
Building Strategy in Reinforcement Learning



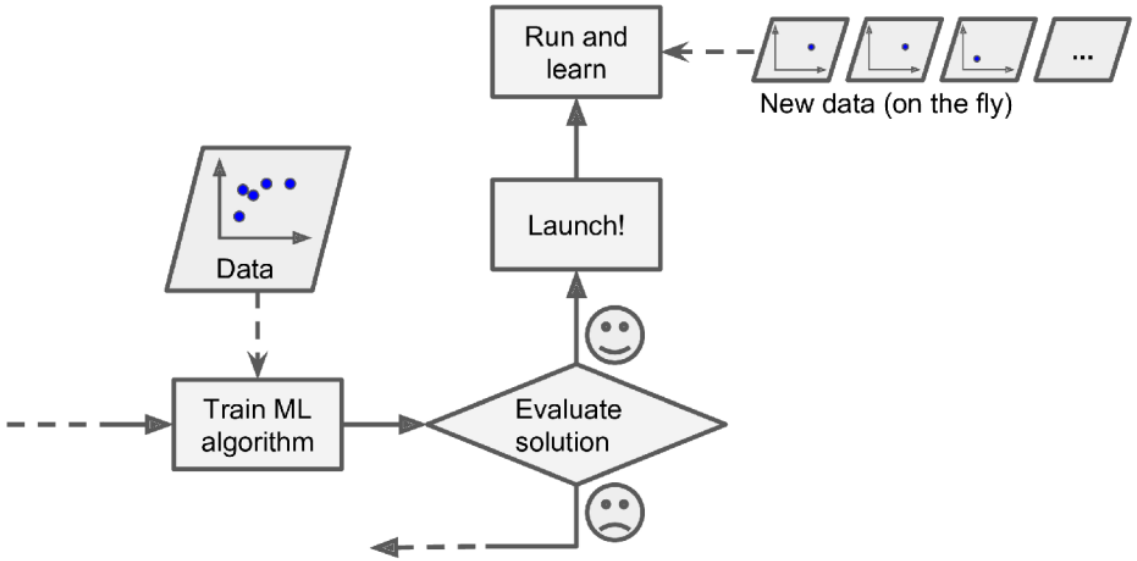
Receiving reward or penalty against its actions and fine-tuning its strategy

Types of Machine Learning Systems (Cont.)

Batch and Online Learning

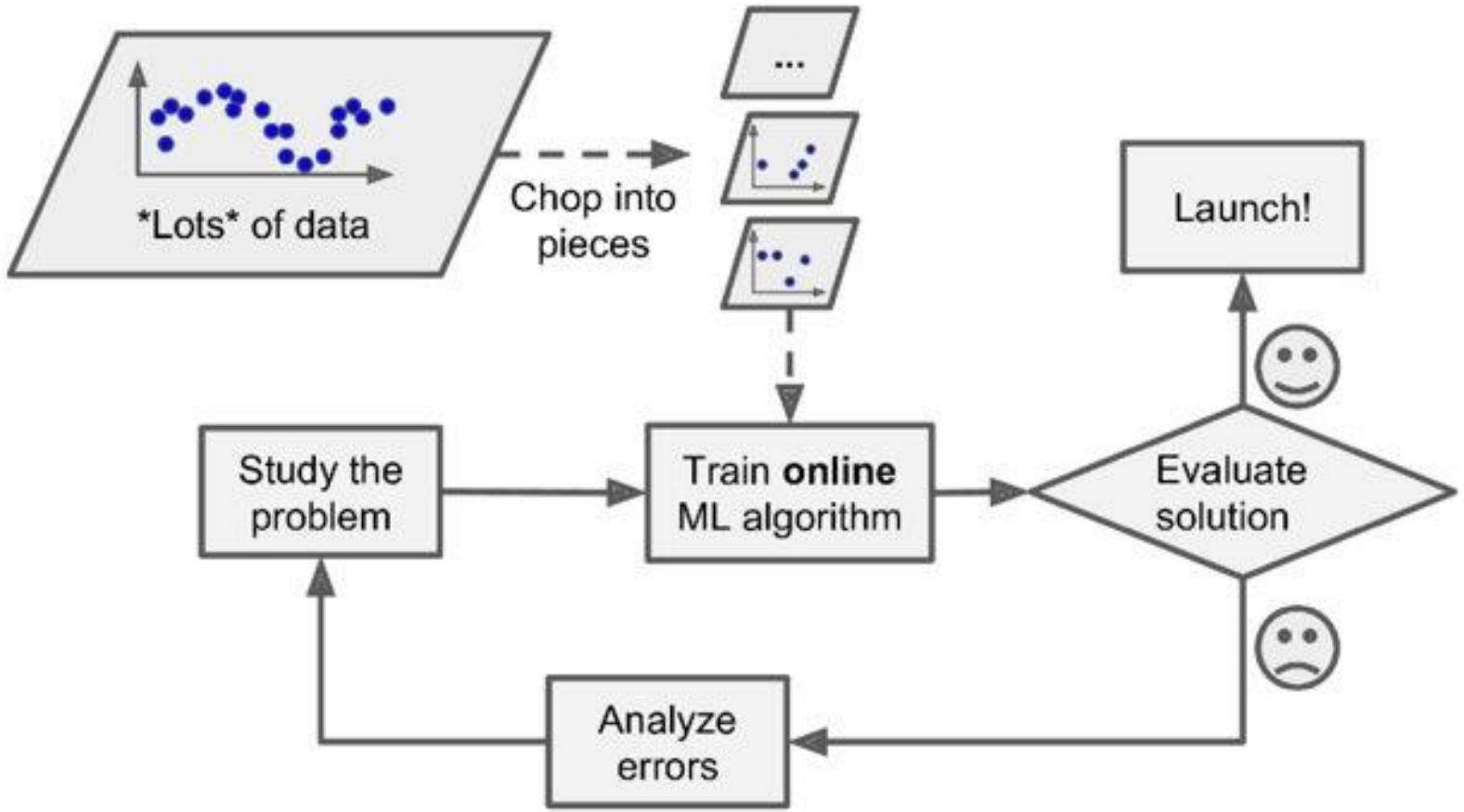


Relearning from scratch on full dataset (old and new data) in Batch Learning



Learning is incremental for new data in Online Learning

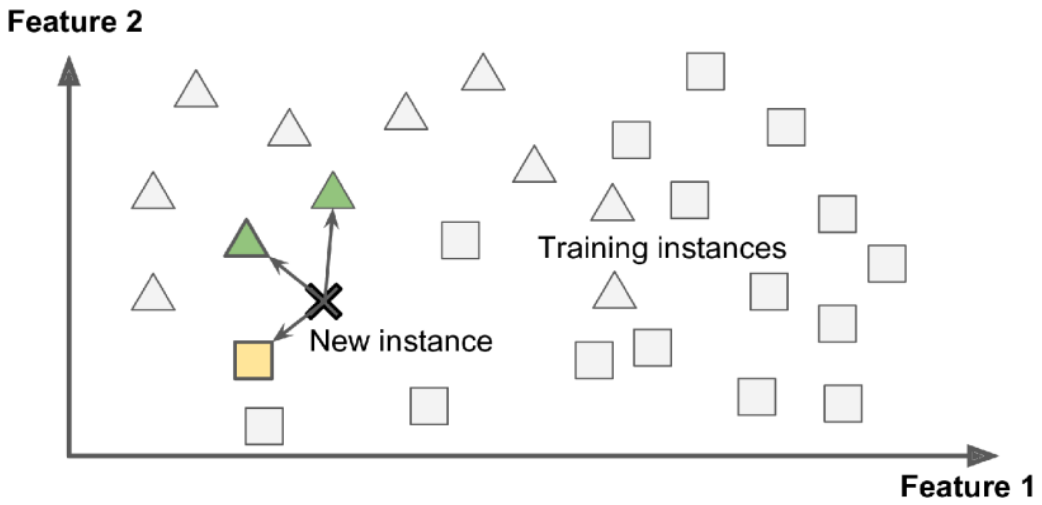
Types of Machine Learning Systems (Cont.)



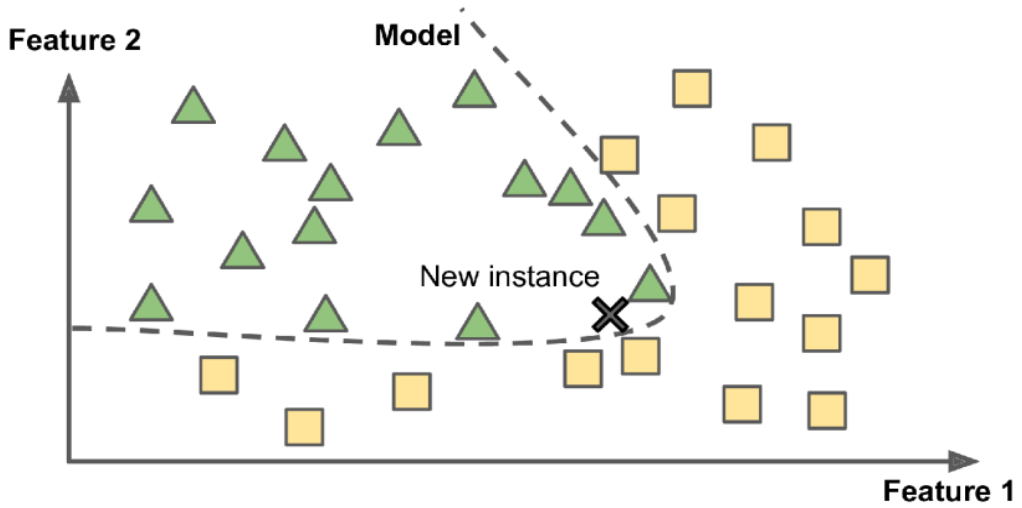
Handling huge dataset in Online Learning

Types of Machine Learning Systems (Cont.)

Instance-based or Model-based Learning systems on how they generalized



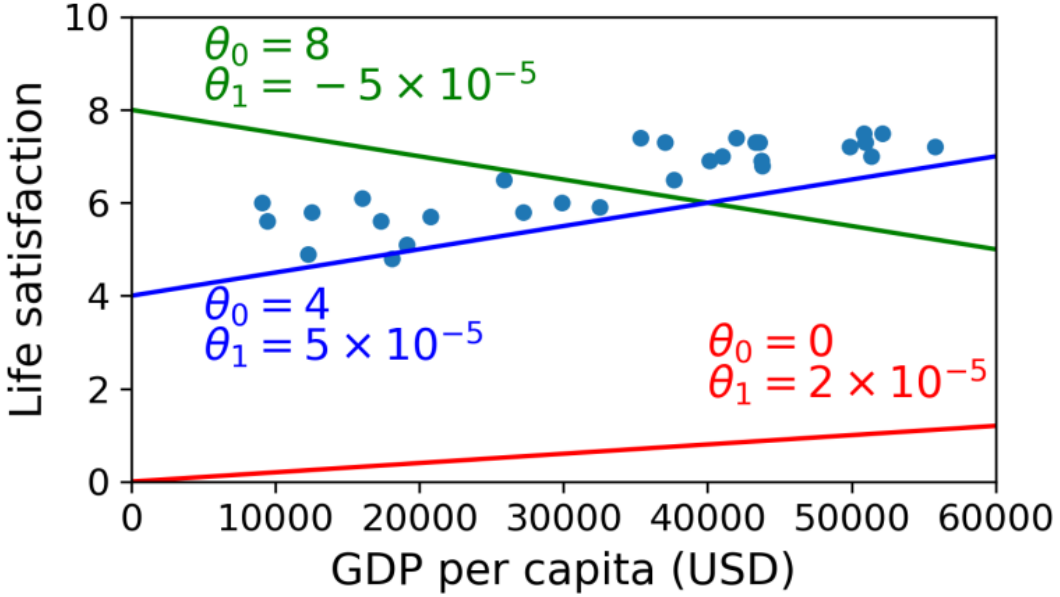
Instance-based Learning is by heart and it then generalizes to new cases by using similarity measures



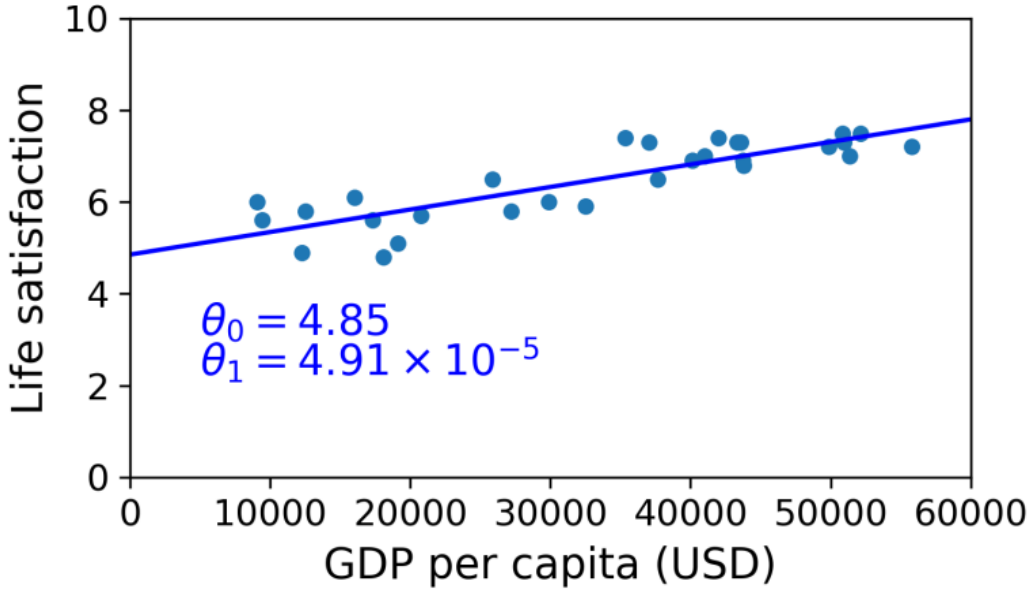
Model is created in Model-based Learning and then it make predictions

Types of Machine Learning Systems (Cont.)

Fitting Model to Data in Model-based Learning



Possible linear models

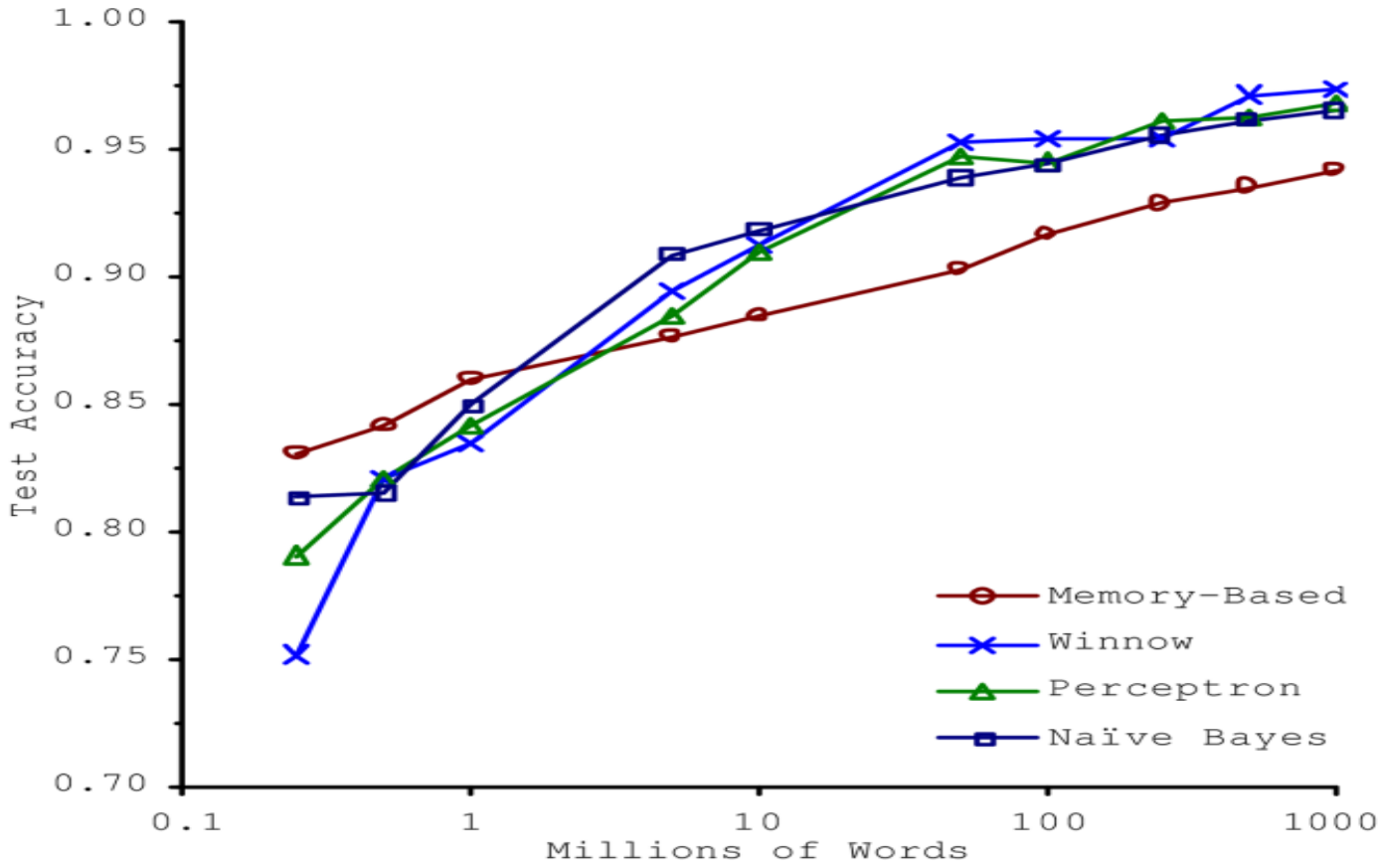


Fitted linear model

Main Challenges in Machine Learning

Insufficient Quantity of Training Data

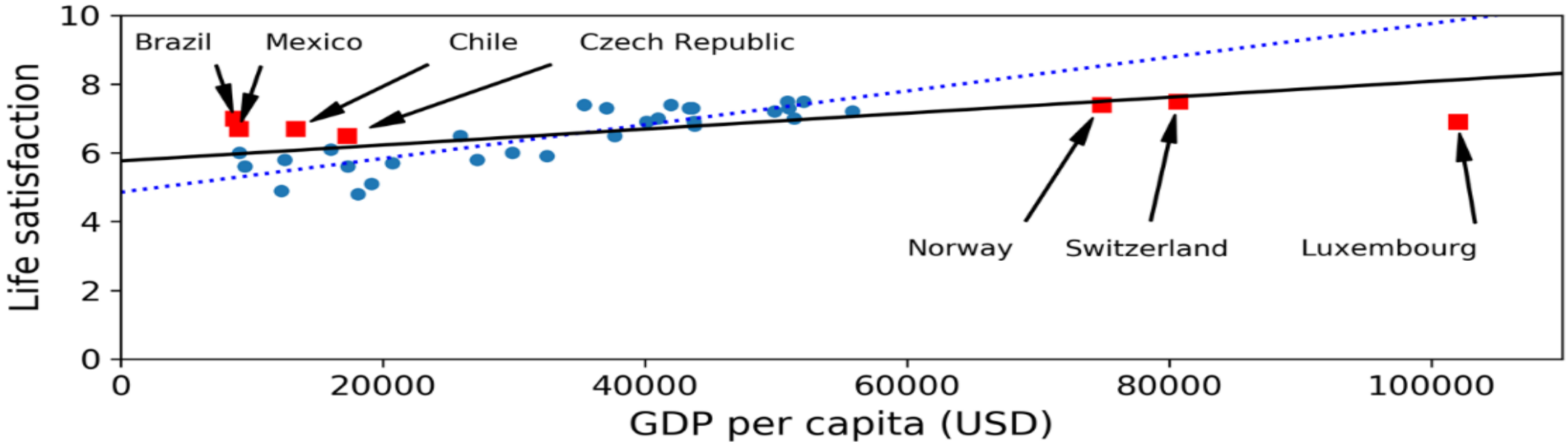
Trade-off between spending time and money on algorithm development and spending these on corpus development



Main Challenges in Machine Learning (Cont.)

Nonrepresentative Training Data

- Sampling Noise
- Sampling Bias



A better model fitted over more representative training data

Main Challenges in Machine Learning (Cont.)

Poor-Quality Data

- Errors
- Missing values
- Outliers

Main Challenges in Machine Learning (Cont.)

Irrelevant Features

Applying Feature Engineering involving the following steps

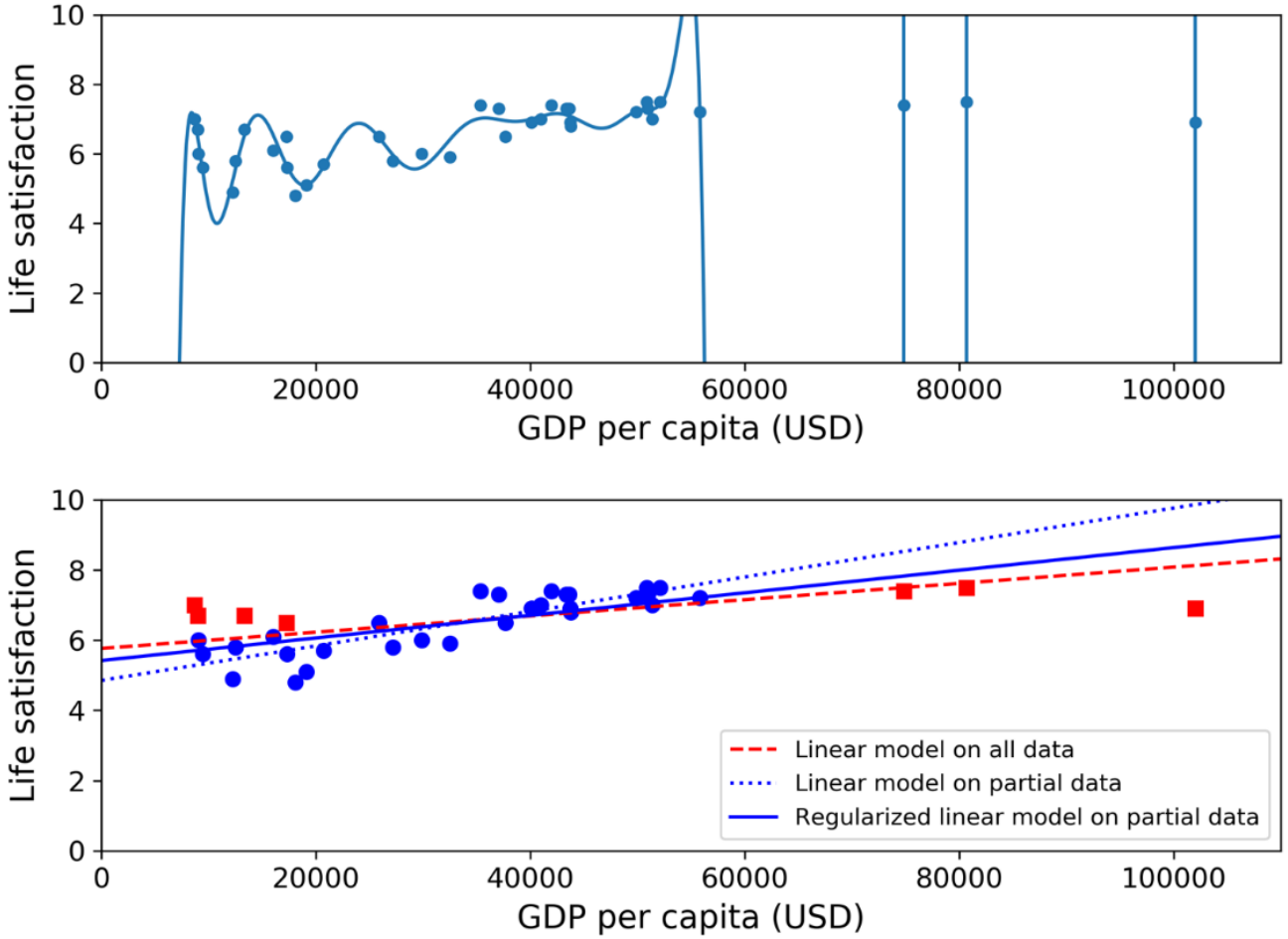
- Feature selection
- Feature extraction
- Creating new features

Main Challenges in Machine Learning (Cont.)

Overfitting the Training Data

Resolving by

- Simplifying model
- Applying constraints (regularization)
- Gathering more training data
- Fixing data quality issues



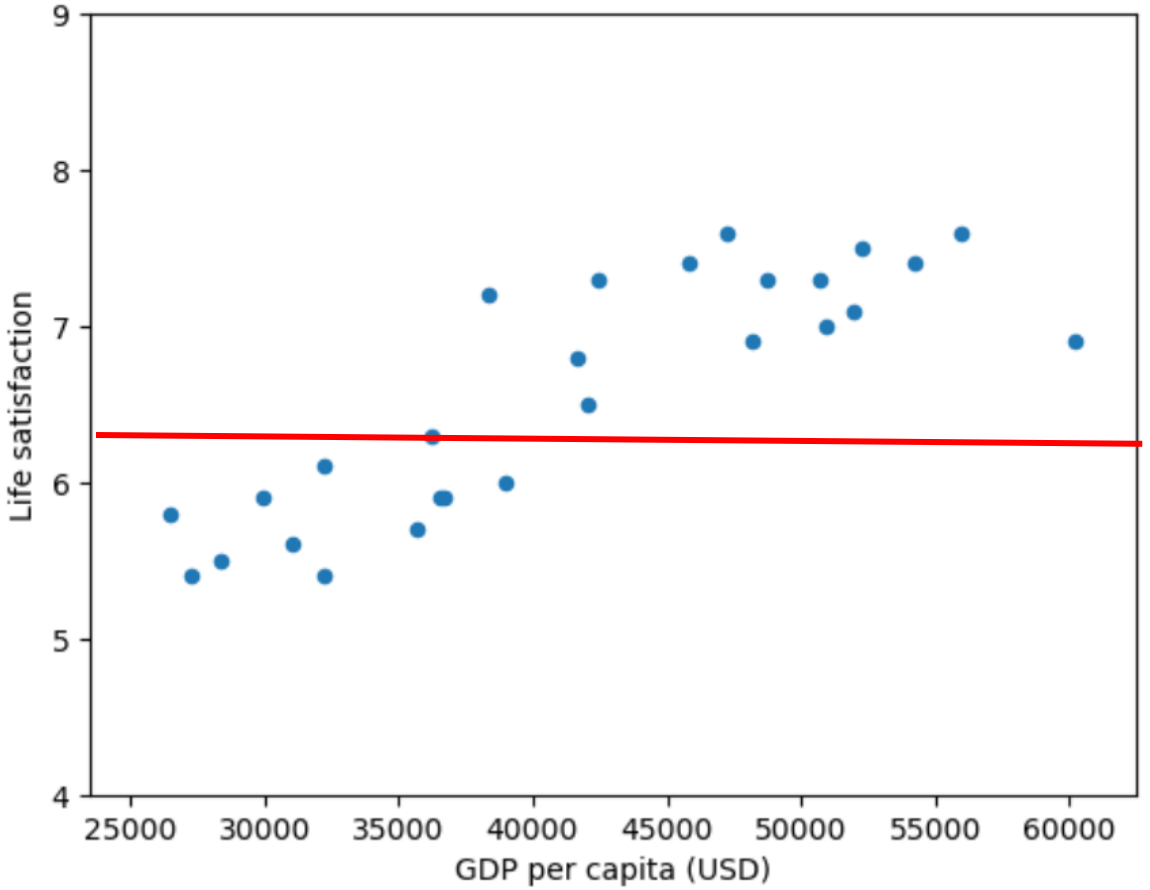
Overfitting and applying regularization to avoid it

Main Challenges in Machine Learning (Cont.)

Underfitting the Training Data

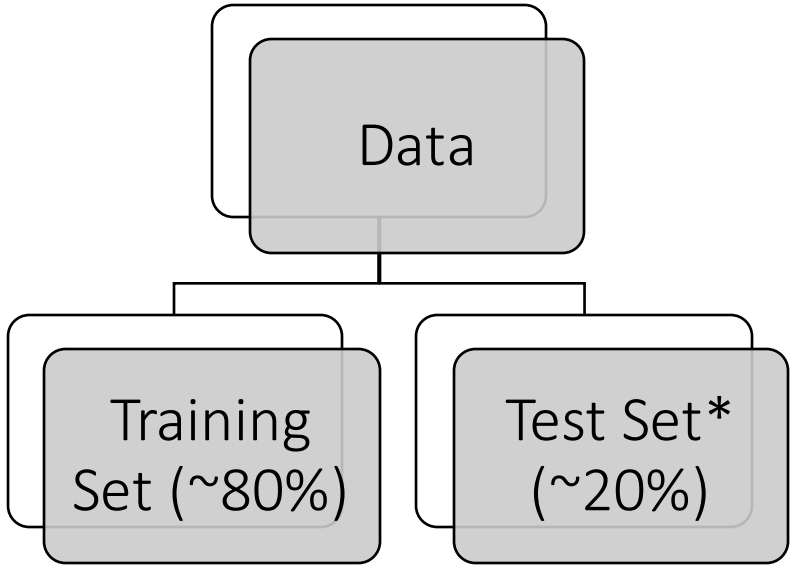
Resolving by

- Selecting more powerful model
- Feeding better features
- Reducing constraints or regularization



Testing and Validating

Ensuring Model Generalizes Well



1

Model is built by fitting it on the training data

2

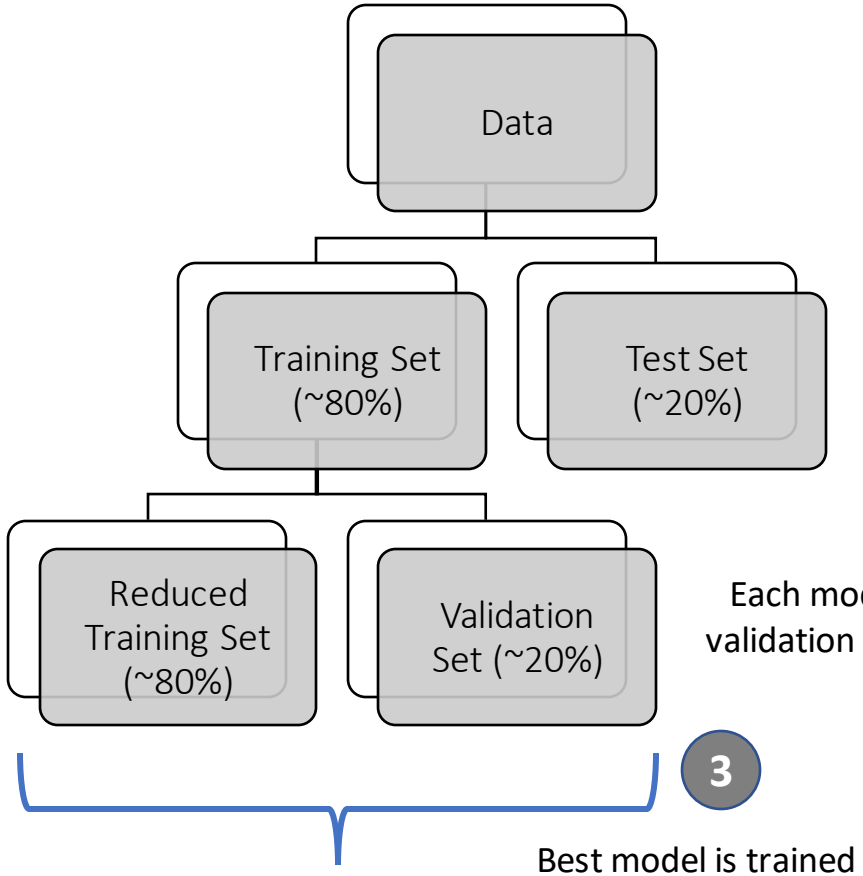
Test data helps estimating how the model generalizes

a.k.a. Hold-out set

A better model fitted over more representative training data

Testing and Validating (Cont.)

- Hyperparameter Tuning
- Model Selection
- Validation
- Cross-validation



1

Multiple models with different hyperparameters are created on reduced training data set

2

Each models gets tested on validation or hold-out data set

3

Best model is trained on whole training data set

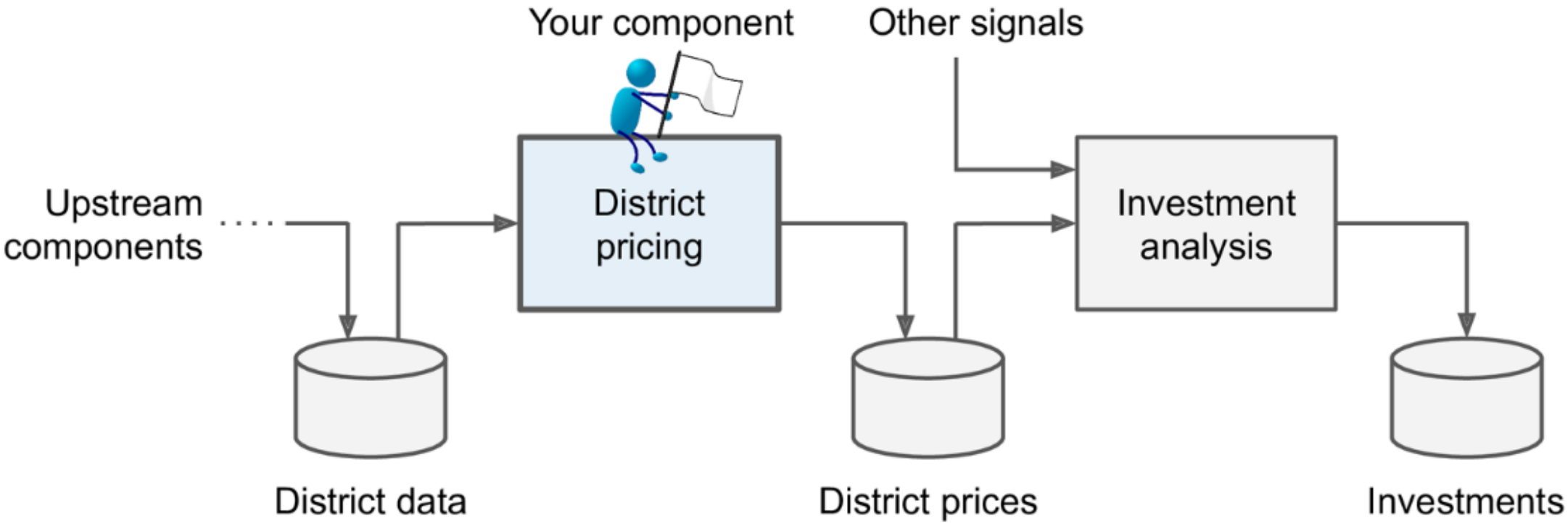
4

Final model gets tested on test set to estimate generalization error

No Free Lunch (NFL) Theorem

- No model that is *a priori* guarantees to be a better one
- Evaluating all models is only way to know which one works best – *Not practical*
- Hence, making *reasonable assumption about data* and *evaluating few reasonable models* is better option

The Pipeline



High-level machine learning pipeline for real estate investment

Defining Learning Problems & Designing a Learning System

Defining Learning Problem

Refer the following Machine Learning definition once again.

*"A computer program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E***

To have a well-defined learning problem, the followings are to be identified

- The Task (***T***)
- The Performance Measure (***P***), and
- Training Experience (***E***)

Defining Learning Problem (Cont.)

Examples:

Learning Problem	Task (T)	Performance Measure (P)	Training Experience (E)
A checkers learning problem	Playing checkers	Percentage of games won against opponents	Playing practice games against itself
A handwriting recognition learning problem	Recognizing and classifying handwritten words within image	Percentage of words correctly classified	Database of handwritten words with given classification
A self-driving vehicle learning problem	Driving on one or multi-lane road	Average distance travelled before error occurs	Sequence of image and steering commands

Designing a Learning System

(Understanding through Checker game examples)

A. Choosing Training Experience
B. Choosing Target Function
C. Choosing Target Function Representation
D. Choosing a Function Approximation Algorithm
E. The Final Design

Designing a Learning System
(considering Checker game examples here)

Choosing Training Experience

a) Attributes of Training Experience Learning

1. Learning from feedback
   - *Direct* feedback
   - *Indirect* feedback
     - *Credit Assignment*
   - Learning from *Direct* feedback is earlier than learning from *Indirect* feedback
2. Controlling sequence of training examples
   - Teacher to provide board states with correct move for each
   - Learner to propose confusing board states asking teacher for the correct move
   - Learner experiments with novel board state (or minor variation of line of play) with indirect feedback while playing against itself
3. Distribution of training examples

Designing a Learning System (Cont.)

Choosing Training Experience (Cont.)

b) Defining learning problem

- Task *T*: Playing Checker
- Performance measure *P*: Percent of games won
- Training experience *E*: Games played against itself

Designing a Learning System (Cont.)

Choosing Target Function

- Determining *type of knowledge* to learn such that it can used in performance measurement (generating *legal* moves from any board state – the *Optimization* problem)
- Options for *target* function to learn

Target Function	Description	Comments
ChooseMove: B --> M	B is set of legal moves and M is chosen output move	Difficulty in learning from indirect feedback
V : B --> $\mathbb{R}$	Mapping any legal board state from the set of B to some real value	Easier to learn to assign a numerical score to each any board state

Designing a Learning System (Cont.)

Choosing Target Function (Cont.)

V : B --> $\mathbb{R}$

Assigning higher score to better board state

Condition	*V(b)* Output
***b* is final board state that is won**	100
***b* is final board state that is lost**	-100
***b* is final board state that is drawn**	0
***b* is not a final board state**	*V(b')* where *b'* is the best board state that can be achieved starting from *b* and playing optimally till the end of game

V(b)=V(b') is a *non-operational* definition as it is not efficiently computable. Instead, an *operational* version of *V^ (read V-hat)* is to be discovered and *approximated*, and this process is also called *Function Approximation*.

Designing a Learning System (Cont.)

Choosing Target Function Representation

The Options:

- A large table with a distinct entry specifying the value for each distinct board state
- A collection of rules that match against features of the board state
- A quadratic polynomial function of predefined board features
- An artificial neural network

An Example $V\hat{}$ Representation:

$$V\hat{}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

w_0 through w_6: weights or coefficients
x_1: No. of black pieces
x_2: No. red pieces
x_3: No. of black kings
x_4: No. of red kings
x_5: No. of black pieces threatened by red
x_6: No. of red pieces threatened by black

Designing a Learning System (Cont.)

Choosing a Function Approximation Algorithm

- Training examples each describing a specific board state *b* and training value for *b* in the form of <*b*, $V_{train}(b)$> are required to learn approximation function V^

Example:

<<x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0>, +100> where black has won the game

- Estimating training values
  - It's easy to assign a value to a board state that corresponds to the end of game
  - It's difficult to assign values to all intermediate board states before game ends
  - Rule for estimating training values: $V_{train}(b)$ <-- *V^(Successor(b))*

Designing a Learning System (Cont.)

Choosing a Function Approximation Algorithm (Cont.)

- Adjusting the weights
  - Learning algorithm to choose values for weights w_i to best fit training examples $\{\langle b, V_{train}(b)\rangle\}$
  - Minimizing squared error E using learning algorithm such as Least Mean Square (LMS)

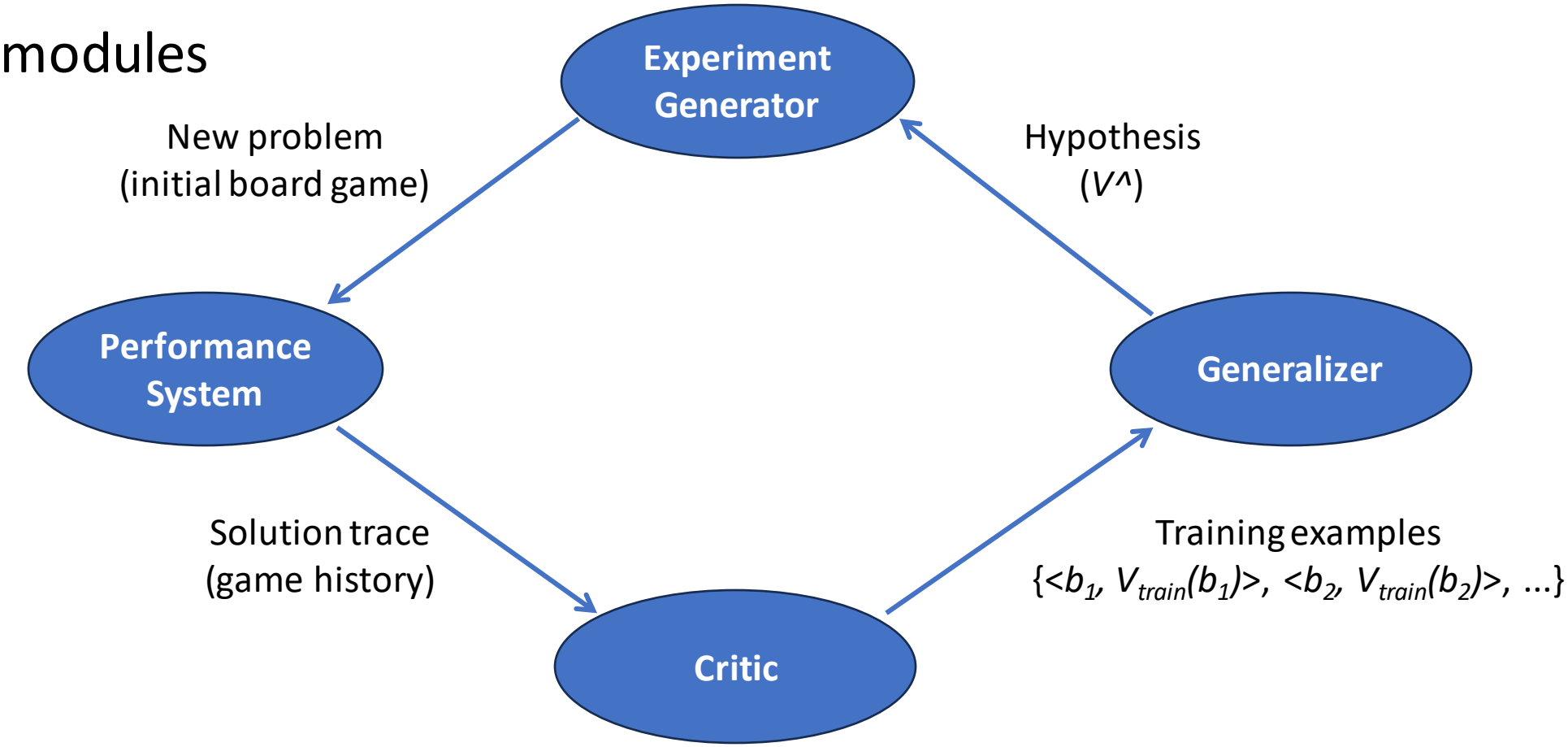
$$E \equiv \sum_{\langle b, V_{train}(b)\rangle \in \text{training examples}} (V_{train}(b) - \hat{V}(b))^2$$

  - LMS weight update rule
    - For each training example $\langle b, Vtrain(b)\rangle$
      - Use the current weight to calculate $V\hat{}(b)$
      - For each weight w_i, update it as $w_i \leftarrow w_i + \eta(V_{train}(b) - V\hat{}(b))x_i$

Designing a Learning System (Cont.)

The Final Design

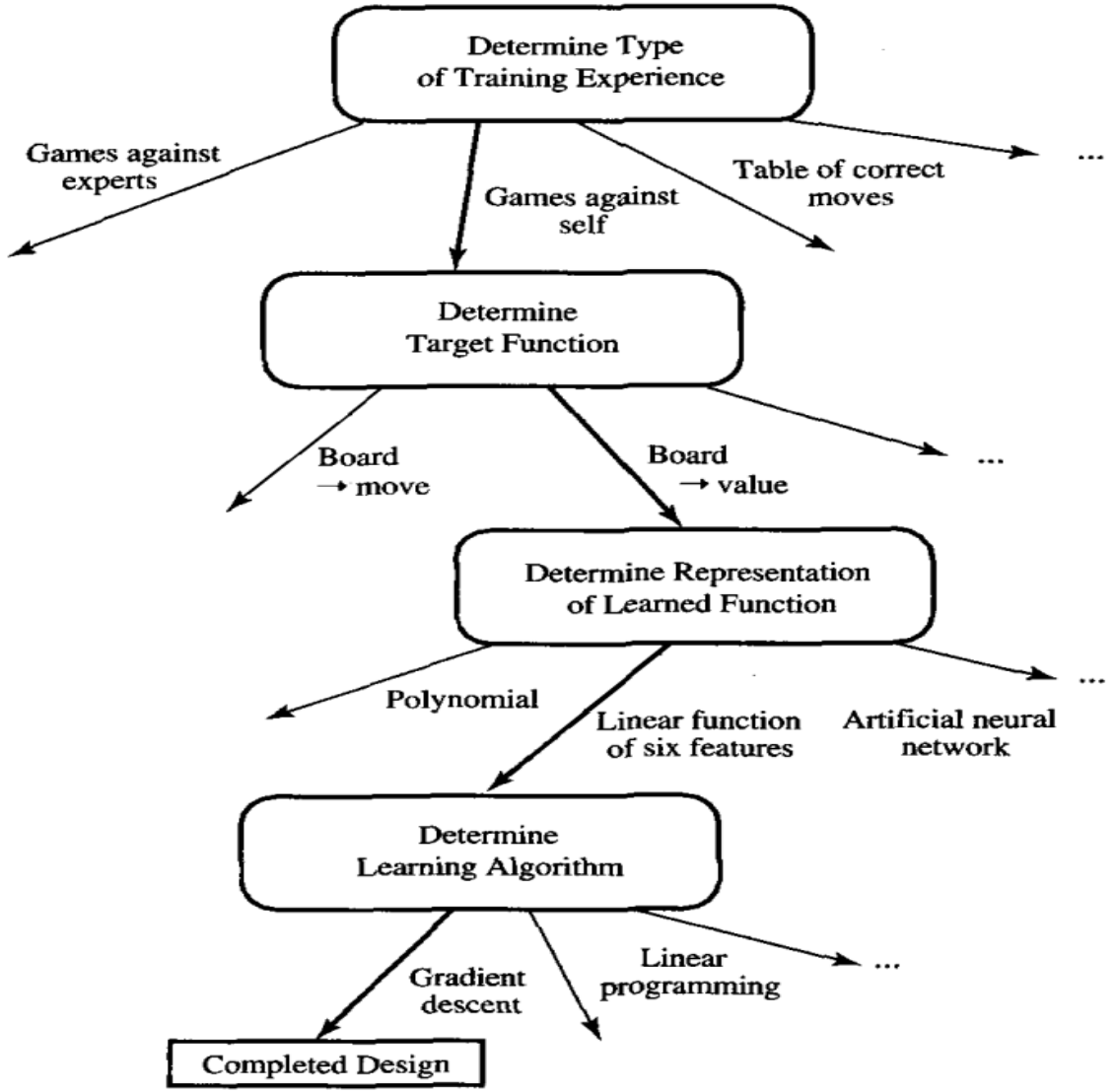
Four core modules



Designing a Learning System (Cont.)

The Final Design (Cont.)

The (sequence of) design choices



Perspectives on Machine Learning

- Involves searching a large space of possible hypotheses to determine one that best fits the observed data
  - for checker, hypothesis space consist of evaluation functions that can be represented by some choices of values for the weights w_0 through w_6
- Algorithms that searches through hypothesis space
  - Linear Functions
  - Decision Trees
  - Artificial Neural Network, etc.
- Relationship between the size of the hypothesis space to be searched and number of training examples available, and the confidence that a hypothesis consistent with the training data will correctly generalize to unseen examples

Issues in Machine Learning

- Algorithms
  - What algorithms exist for learning general target function from specific examples?
  - In what settings will a particular algorithm converge to the desired function, given sufficient training data?
  - Which algorithms perform best for which type of problems and representations?
- Training data
  - How much training data is sufficient?
- Learning function
  - What's the best way to reduce learning task to one or more function approximation problems?
  - How can the learner automatically alter its representation to improve its ability to represent and learn the target function?

End-to-End Machine Learning Project

Major Steps Involved

1. Looking at the big picture.
2. Getting the data.
3. Discovering and visualizing the data to gain insights.
4. Preparing the data for Machine Learning algorithms.
5. Selecting a model and train it.
6. Fine-tuning your model.
7. Presenting your solution.
8. Launching, monitoring, and maintaining your system.

The Big Picture

- **The Context:** Automatic prediction of district's median house prices using census data
- **Framing the Problem:**
  - Understanding current business problem
    - Time consuming and costly manual operations
    - Performance is not always good
  - Proposed solution
    - A model to predict house price
      - Considering if it should be a supervised, unsupervised or reinforcement learning
      - Considering if it should be classification, (univariate or multivariate) regression or any other type of task
      - Whether it should be batch or online learning
    - Downstream apps to consume the predictions for further business processes and to decide on investments

The Big Picture (Cont.)

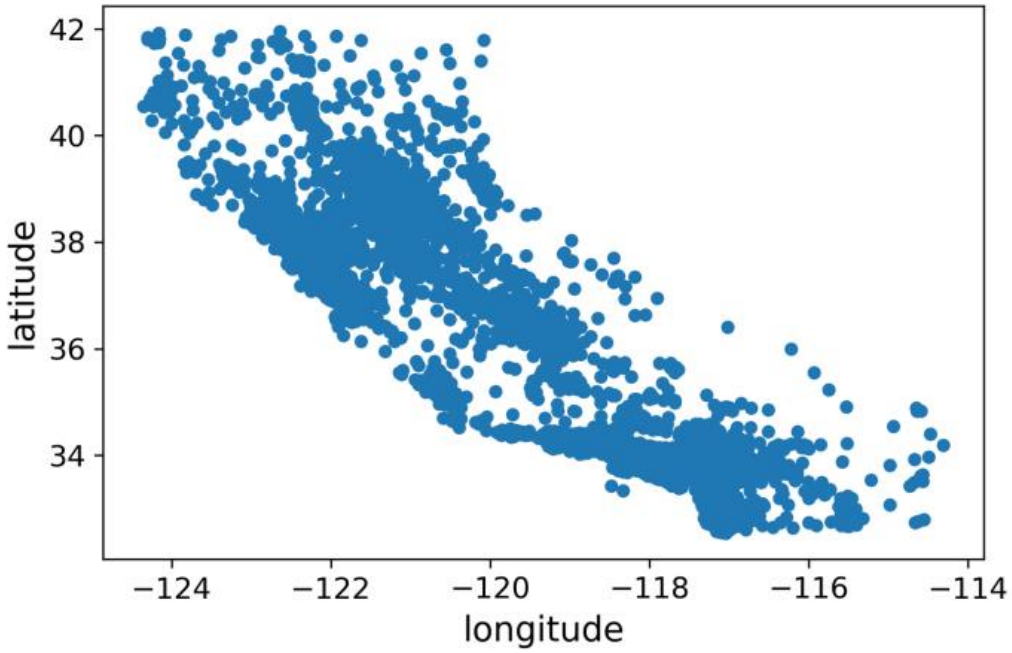
- **Deciding on performance metrics:**
  - Means Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Error (MAE)
- Choosing candidate algorithms
- Realizing development effort

Getting the Data

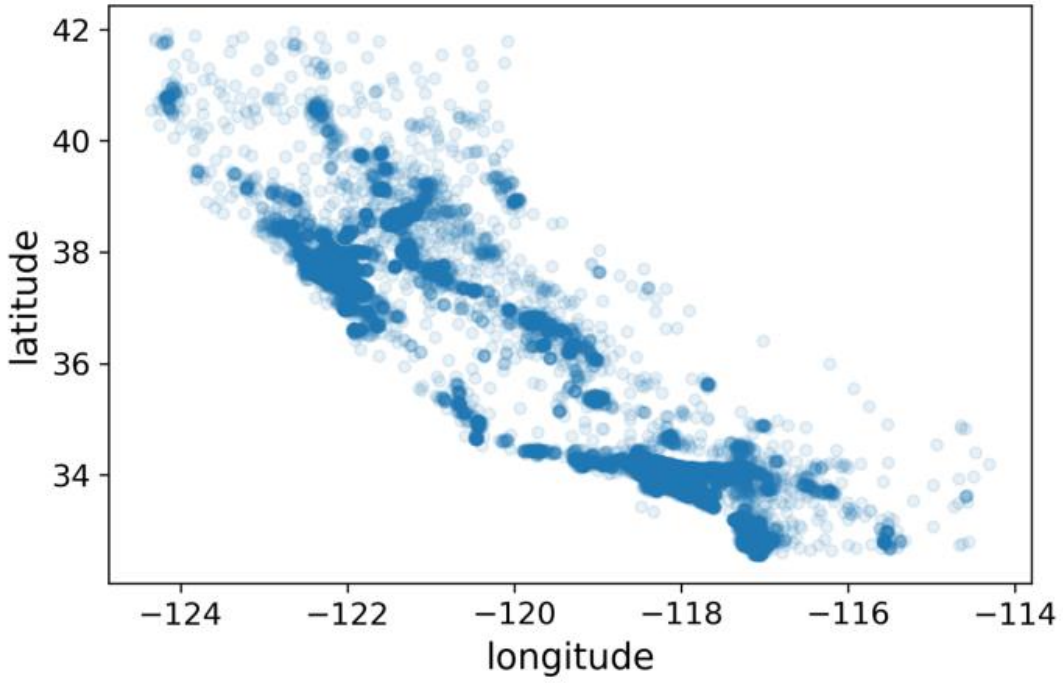
- Prerequisites (setting up development environment)
- Downloading the relevant data
- Taking a quick look at the data
  - Basic information about dataset
  - Elementary statistics
  - Visualizing
  - Distribution
- Creating test set
  - Randomizing the index of the observations
  - Splitting data into train and test data set (~80:20)
  - Considering stratification, if required

Exploring & Visualizing the Data

Visualizing geographic data



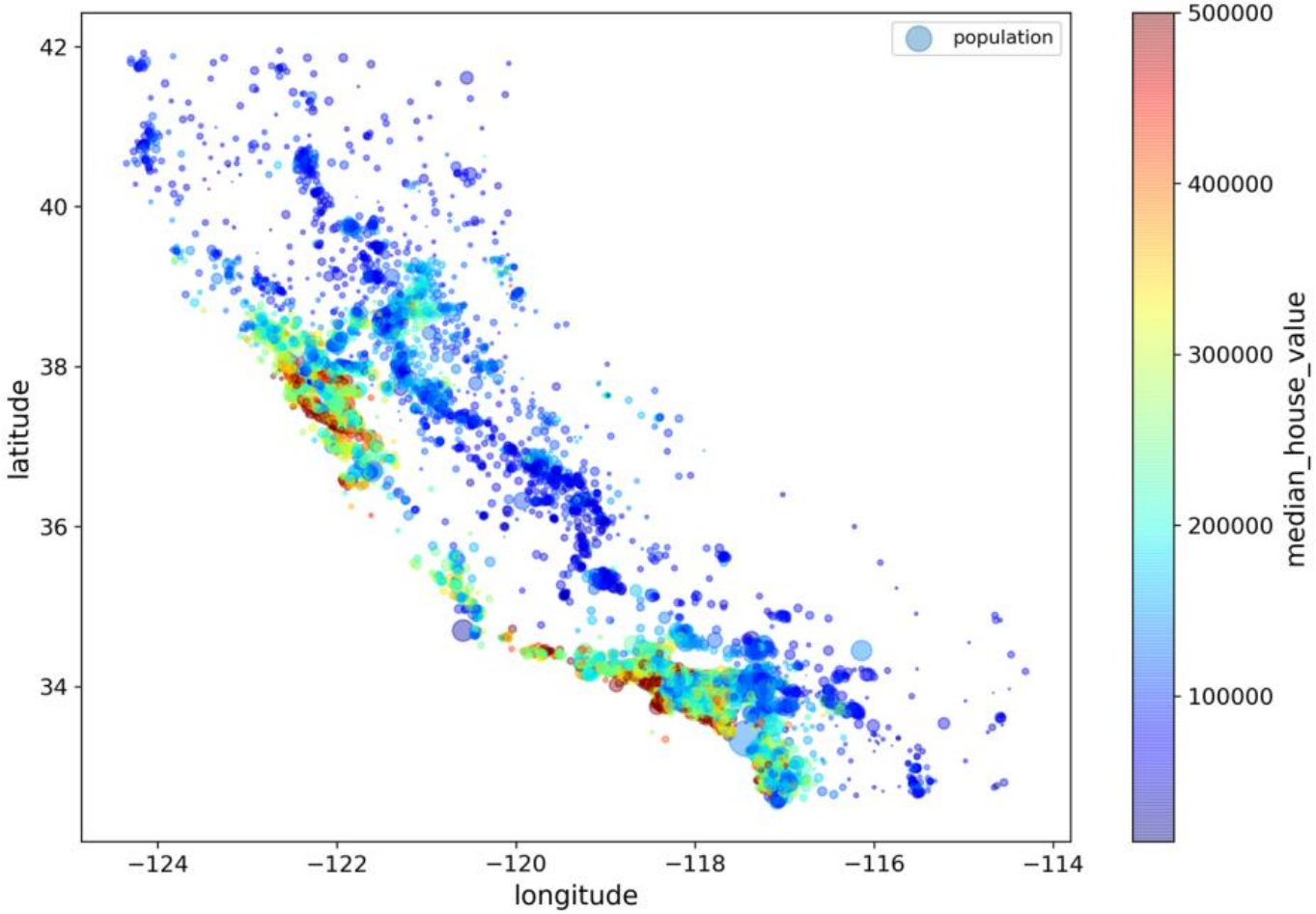
A geographical scatterplot



A better visualization of geographical data highlighting high-density areas

Exploring & Visualizing the Data (Cont.)

Better visualization

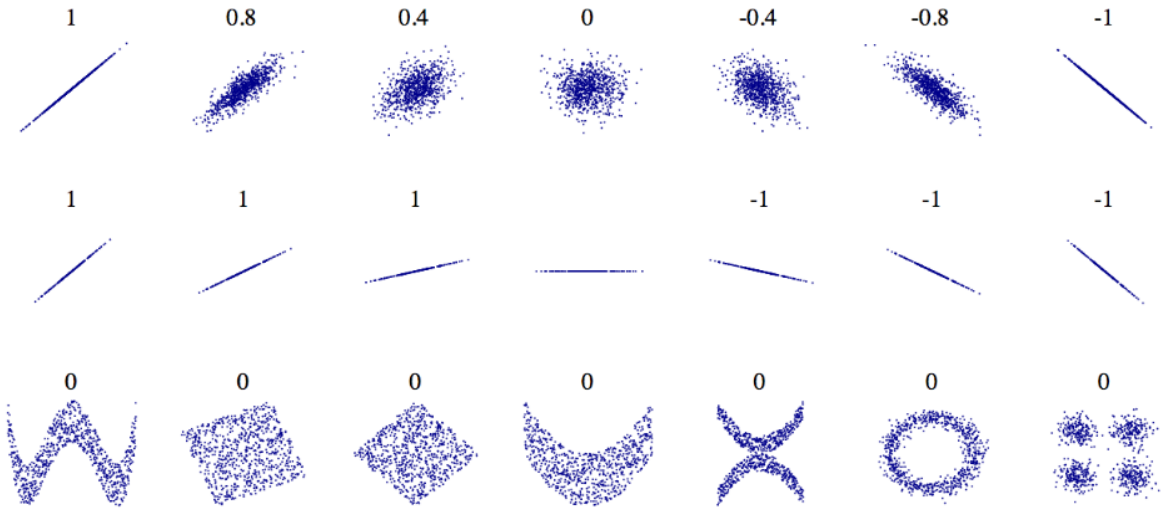


California housing prices: red is expensive, blue is cheap, larger circles indicate areas with a larger population

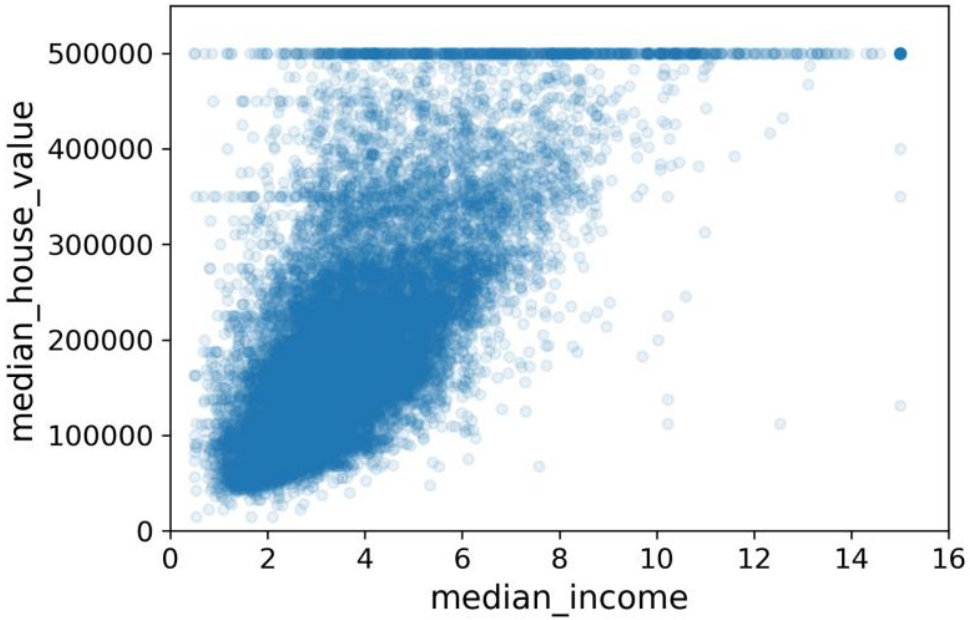
Exploring & Visualizing the Data (Cont.)

Finding correlations

- Correlation coefficient and its range
- Linear correlation: Positive & Negative correlation
- Nonlinear correlation



Standard correlation coefficient of various datasets



Strong relation between median income and medium house value

Preparing Data

- Data cleaning
  - Handling missing values
    - Removing relevant data, or
    - Removing attribute from entire dataset, or
    - Setting the values with some value (zero, mean, median, etc.)
  - Handling text and categorical attributes
    - Ordinal encoding
    - One-hot encoding
  - Transforming data
    - Scaling features
      - Min-max
      - Standardization
    - Pipelining for transformations

Selecting & Training a Model

- Training model
  - Simple models
  - Complex models
- Evaluating model
  - Evaluation on training set
  - Evaluation over cross-validation
- Considering better model (if applicable)

Fine-tuning Model

- Grid searching
- Randomized searching
- Ensembling
- Analyzing model performances
- Evaluating final model against test set

Launching, Monitoring & Maintaining Models

- Deployment
  - Deploying model as web service
  - Deploying model over Cloud

Model deployed as a web service and being consumed by an application

- Monitoring
  - Monitoring model performance on regular basis
  - Setting up alerts to be triggered when performance falls below threshold
- Maintenance
  - Scripting for automatic model building and hyperparameters tuning
  - Keeping model backup
  - Versioning data sets

References

Textbooks:

1. Tom M. Mitchell. Machine Learning, McGraw-Hill Education, 2013
2. Aurélien Géron. Hands-On Machine Learning with Scikit-Learn, Keras &TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly, 2019