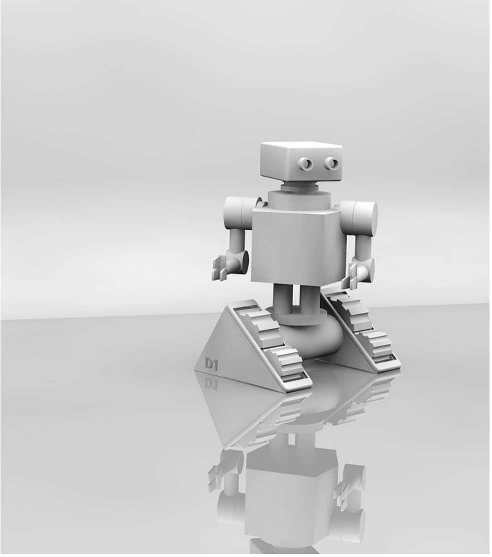


CPSC 481
Artificial Intelligence

Dr. Mira Kim
Mira.kim@fullerton.edu

What we will cover this week

- Introduction to Machine Learning
- Types of ML algorithms

What is Machine Learning?

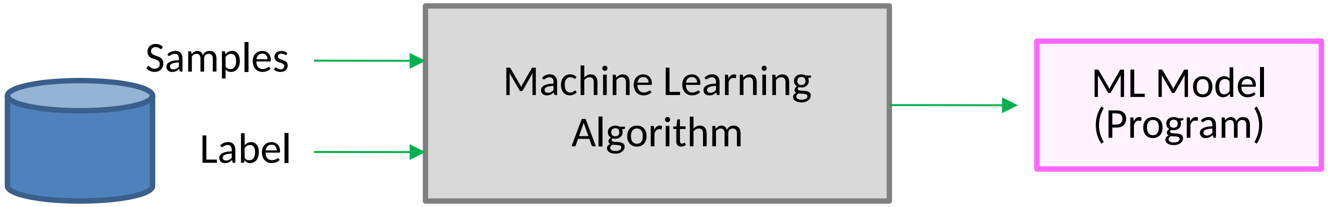
Machine Learning

- General Definition
  - A subfield of AI that gives computers the ability to learn the knowledge without being explicitly programmed
- Engineering Definition
  - Machine Learning is an algorithm that improves their performance P at some task T with experience E.
    - <P, T, E>
    - Learning from experience ‘E’ for a given task ‘T’
  - Example) Spam Filter
    - T : To flag spam for new incoming emails
    - E : Training Data
    - P : Ratio of correctly classified emails (i.e., accuracy)

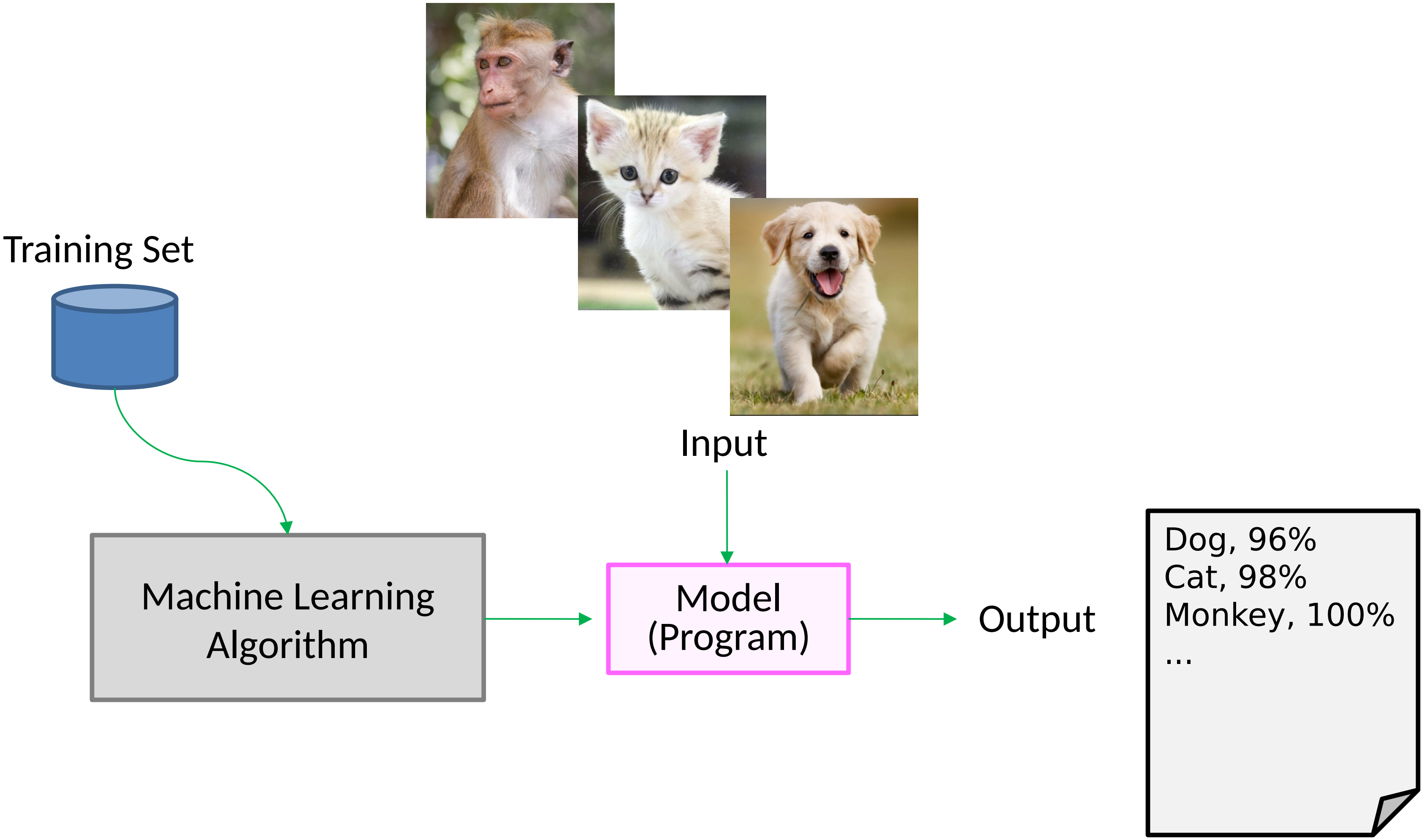
Comparison

- Tradition Programming
  - To write all the rules by hand after studying a problem

- Machine Learning
  - To let the ML algorithm that learns patterns or rules from the training data

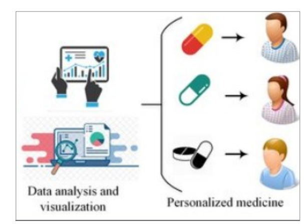


Prediction with Model



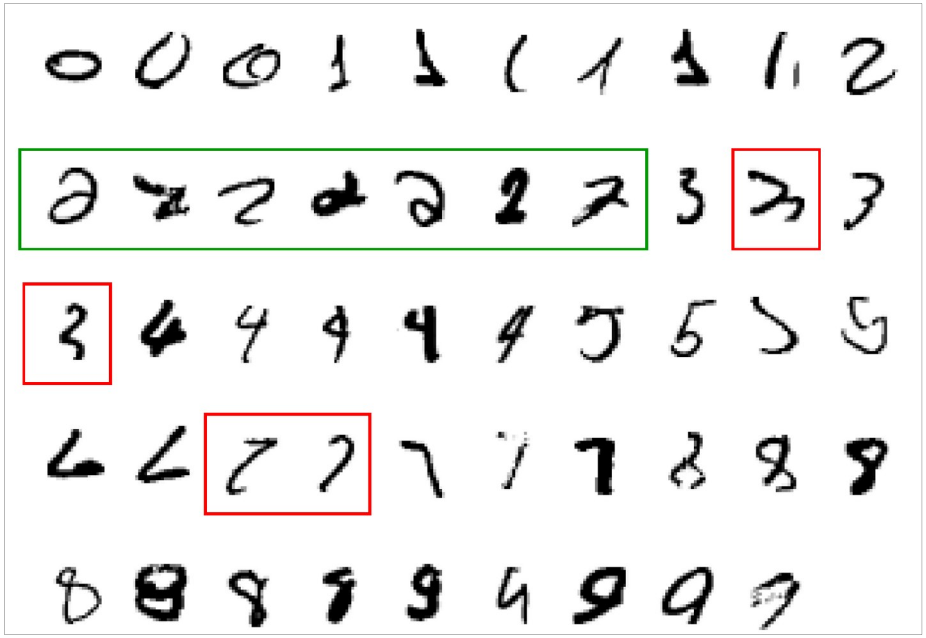
Benefits of Machine Learning

- Handling Complex Problems
  - Applied to problems that are too complex for the traditional approach
  - Ex) Genomics
- Handling New Problems
  - Applies to problems where solutions/algorithms are not known
  - Ex) Navigating on Mars
- Adapting to Changes
  - Capable of automatically adapting to changes in patterns and rules
  - Ex) recommendation systems, speech recognition
- Exploring New Patterns/Rules
  - Helping humans learn or discover new patterns
  - Ex) Personalized Medicine



Benefits of Machine Learning

- Example Requiring Machine Learning
  - What make a number '2'?



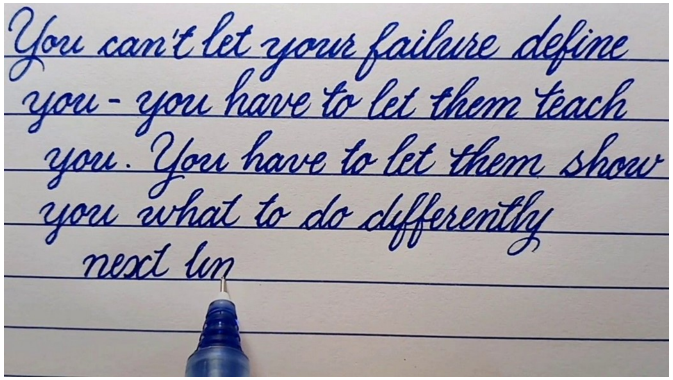
Defining the Learning Task

- Checker Player
  - T: Playing checkers
  - P: Percentage of games won
    against an arbitrary opponent
  - E: Playing practice games against itself

Improve on task T, with respect to performance metric P, based on experience E

Defining the Learning Task

- Handwriting Recognizer
  - T: Recognizing hand-written words
  - P: Percentage of words correctly classified
  - E: Database of human-labeled images of handwritten words



Improve on task T, with respect to performance metric P, based on experience E

Defining the Learning Task

- Autonomous Driving
  - T: Driving on local and highways using various sensors
  - P: Safety record, accuracy of object and event detection
  - E: Sensor data, simulation data, real-world driving experience, etc

Improve on task T, with respect to performance metric P, based on experience E

Defining the Learning Task

- Email spam classification
  - T: Categorize email messages as spam or non-spam
  - P: Percentage of email messages correctly classified
  - E: Database of emails, some with human-given labels

Improve on task T, with respect to performance metric P, based on experience E

Problems Requiring Machine Learning

- Recognizing Patterns
  - Facial Identities
  - Facial Expressions
  - Handwritten Words
  - Speech recognition
  - Diseases in Medical Images
- Generating patterns
  - Generating Images
  - Generating Motion Sequences
- Recognizing Anomaly
  - Unusual Credit Card Transactions
  - Unusual Patterns of Sensor Readings in Nuclear Power Plant
- Prediction
  - Future Stock Prices
  - Currency Exchange Rates

Challenges of Machine Learning

Challenges of Machine Learning

- Acquiring Sufficient Training Set
  - A larger volume of training set tends to yield a higher performing ML model.
- Acquiring High-Quality Training Set
  - Avoid non-representative, bias, low-quality data/image, errors, outliers, and noises

Challenges of Machine Learning

- Selection of Right ML Algorithm
  - Each ML algorithm has its own applicability.
- Defining Well-defined Features
  - Feature Selection
  - Feature Extraction

Challenges of Machine Learning

- Right Fitting
  - Avoid Overfitting/Underfitting
  - Overfitting
    - The model may perform well on the training set, but may not generalize.
    - Complex models tend to be overfitted to the train data.
  - Underfitting
    - The model is too simple to learn the underlying structure of the data.
    - The model can’t satisfy requirements.

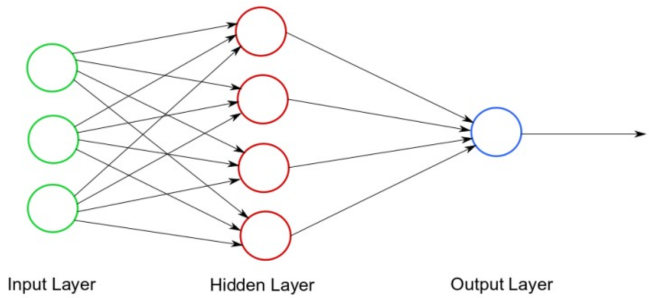
Machine Learning Libraries

SciKit-Learn

- Machine Learning library for Python
- Supported Algorithms
  - Classification, Regression and Clustering algorithms
    - Support Vector Machine (SVM)
    - Random Forest
    - Gradient Boosting
    - K-Means
    - DBSCAN
- Built on NumPy, SciPy and matplotlib

TensorFlow

- Developed by Google Brain
- Notion of Tensor and Flow between Tensors
- Related Terms
  - (Artificial) Neural Network
  - Deep Learning
    - Based on Neural Network for the purpose of Learning
    - Deep Neural Networks
    - Deep Belief Networks
    - Convolutional Neural Network
    - Recurrent Neural Network



- TensorFlow Architecture
  - Can be configured in various ways, supporting various deep learning algorithms.

Keras

- Library running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML.
- Enables fast experimentation with deep neural networks.
  - Keras simplifies the coding for deep learning applications.
- Focuses on being user-friendly, modular, and extensible.
- In 2017, TensorFlow team decided to support Keras in TensorFlow's core library.
- Supports commonly used Neural Network building blocks;
  - Layers, Objectives, Activation Function, and Optimizer

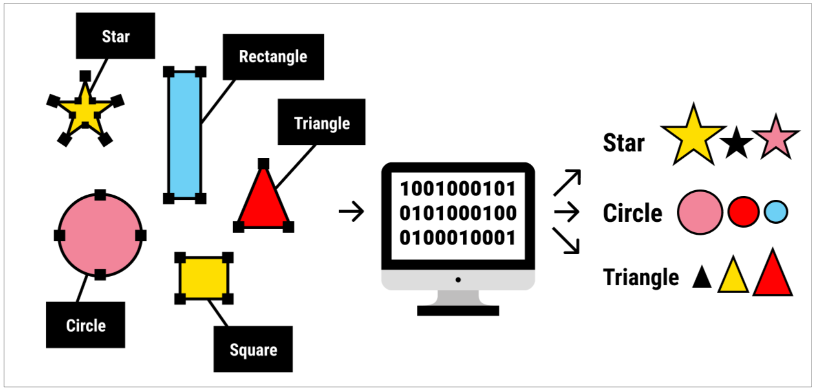
Types of Machine Learning Algorithms

Various Classifications

- Classify by Human Supervision
  - Supervised Learning
  - Unsupervised Learning
  - Semi-supervised Learning
  - Reinforcement Learning
- Classify by Batch or Incremental
  - Batch Learning
  - Incremental/Online Learning
- Classify by Instance-based or Model-based
  - Instanced-based Learning
    - Comparing new data point to known data points
  - Model-based Learning
    - Detecting patterns in the training data and building a predictive model

Supervised Learning

- Learning from Labeled Training Set
  - Learning a function that maps an input to an output based on example input-output pairs. Here, the 'output' is called 'label'.



Graphic Shape Classifier

Labels

	0	1	2	3	4
id	7129300520	6414100192	5631500400	2487200875	1954400510
date	10/13/2014	12/9/2014	2/25/2015	12/9/2014	2/18/2015
price	221900	538000	180000	604000	510000
bedrooms	3	3	2	4	3
bathrooms	1	2.25	1	3	2
sqft_living	1180	2570	770	1960	1680
sqft_lot	5650	7242	10000	5000	8080
floors	1	2	1	1	1
waterfront	0	0	0	0	0
view	0	0	0	0	0
condition	3	3	3	5	3
grade	7	7	6	7	8
sqft_above	1180	2170	770	1050	1680
sqft_basement	0	400	0	910	0
yr_built	1955	1951	1933	1965	1987
yr_renovated	0	1991	0	0	0
zipcode	98178	98125	98028	98136	98074
lat	47.5112	47.721	47.7379	47.5208	47.6168
long	-122.257	-122.319	-122.233	-122.393	-122.045
sqft_living15	1340	1690	2720	1360	1800
sqft_lot15	5650	7639	8062	5000	7503

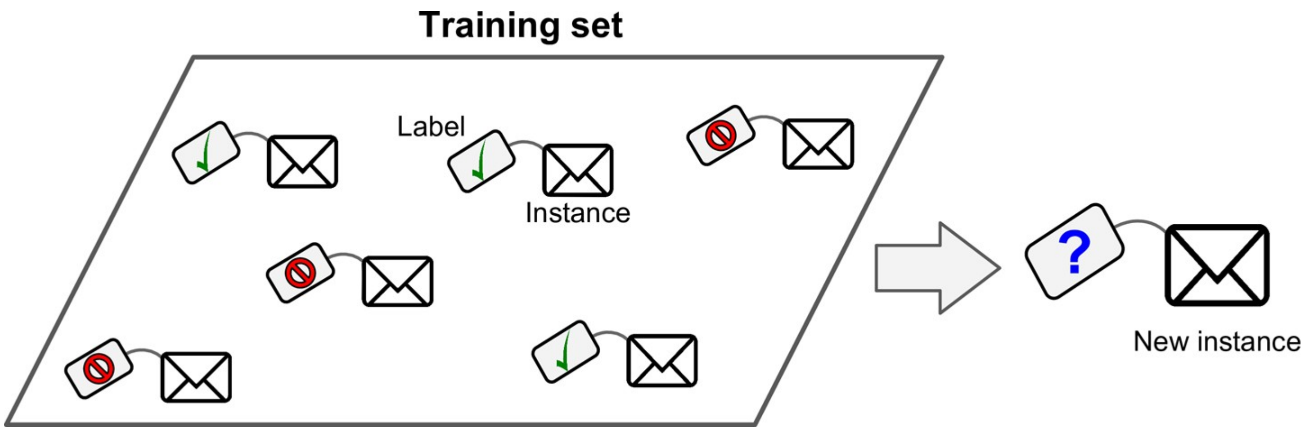
House Price Predictor

Supervised Learning

- Types
  - Regression
  - Classification
- Algorithms
  - k-Nearest Neighbors
  - Linear Regression
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree and Random Forests
  - Neural Networks

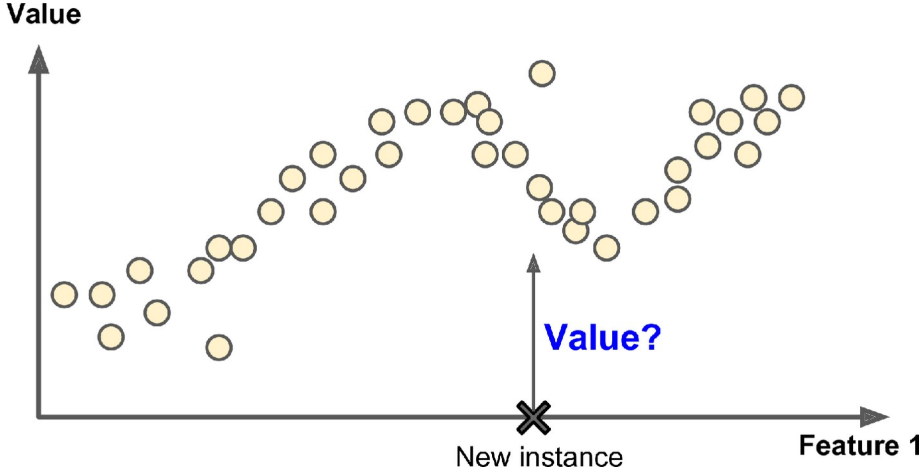
Supervised Learning

- Classification Model with Supervised Learning
  - To filter spam emails
  - Training Set includes labels, ‘normal’ and ‘spam’.



Supervised Learning

- Regression Model with Supervised Learning
  - To predict a value using the feature of new instance



  - Predicting the Price of House to sell
    - Linear Regression Model

Applications of Supervised Learning

1. **Email Filtering**: Classifying emails as spam or not spam.
2. **Image Classification**: Identifying and categorizing objects within images (e.g., cats vs. dogs).
3. **Medical Diagnosis**: Predicting diseases based on patient data.
4. **Credit Scoring**: Predicting the probability of a customer defaulting on a loan.
5. **Sentiment Analysis**: Determining the sentiment of text data (e.g., positive, negative, neutral).
6. **Speech Recognition**: Converting spoken language into text.
7. **Handwriting Recognition**: Translating handwritten text into machine-encoded text.
8. **Stock Price Prediction**: Forecasting future stock prices based on historical data.
9. **Churn Prediction**: Predicting which customers might leave a subscription service.
10. **Weather Forecasting**: Predicting weather conditions based on meteorological data.

Unsupervised Learning

- Learning from Unlabeled Training Set
  - Drawing <u>inferences</u> from unlabeled datasets
  - Most common one is Cluster Analysis
    - Exploratory data analysis to find hidden patterns or grouping in data

Age	Gender	Income	Profession	Tenure	City
35	M	60,000	IT	12	KRK
23	F	90,000	Sales	3	WAW
18	M	12,000	Student	1	KRK
42	F	128,000	Doctor	13	KRK
34	M	63,000	Manager	8	WAW
56	M	82,000	Teacher	30	WAW

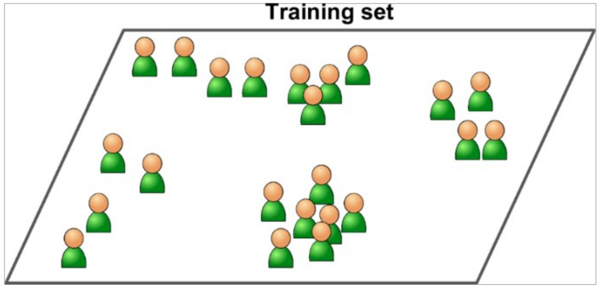
Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio
1	41	2	6	19	0.124	1.073	NBA001	6.3
2	47	1	26	100	4.582	8.218	NBA021	12.8
3	33	2	10	57	6.111	5.802	NBA013	20.9
4	29	2	4	19	0.681	0.516	NBA009	6.3
5	47	1	31	253	9.308	8.908	NBA008	7.2
6	40	1	23	81	0.998	7.831	NBA016	10.9
7	38	2	4	56	0.442	0.454	NBA013	1.6
8	42	3	0	64	0.279	3.945	NBA009	6.6
9	26	1	5	18	0.575	2.215	NBA006	15.5
10	47	3	23	115	0.653	3.947	NBA011	4
11	44	3	8	88	0.285	5.083	NBA010	6.1
12	34	2	9	40	0.374	0.266	NBA003	1.6

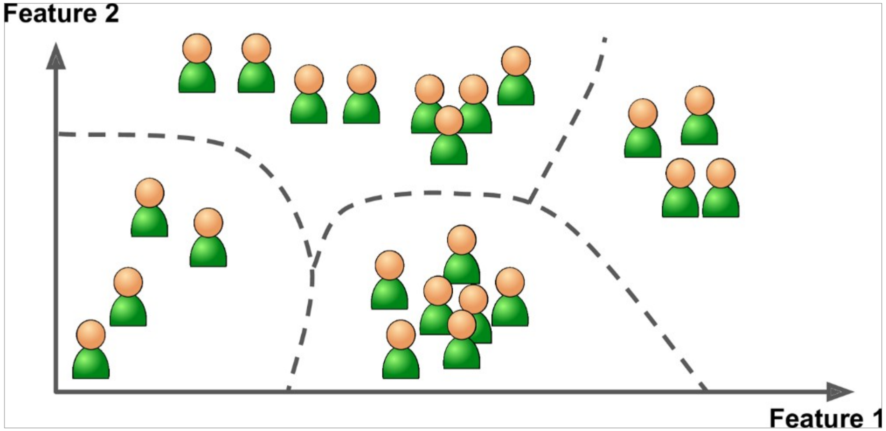
Unsupervised Learning

- Algorithms
  - Clustering
    - k-Means
    - Hierarchical Cluster Analysis (HCA)
    - Expectation Maximization
  - Visualization and Dimensionality Detection
    - Principal Component Analysis (PCA)
    - kernel PCA
    - Locally-Linear Embedding (LLE)
    - t-distributed Stochastic Neighbor Embedding (t-SNE)
  - Association Rule Learning
    - Apriori
    - Eclat

Unsupervised Learning

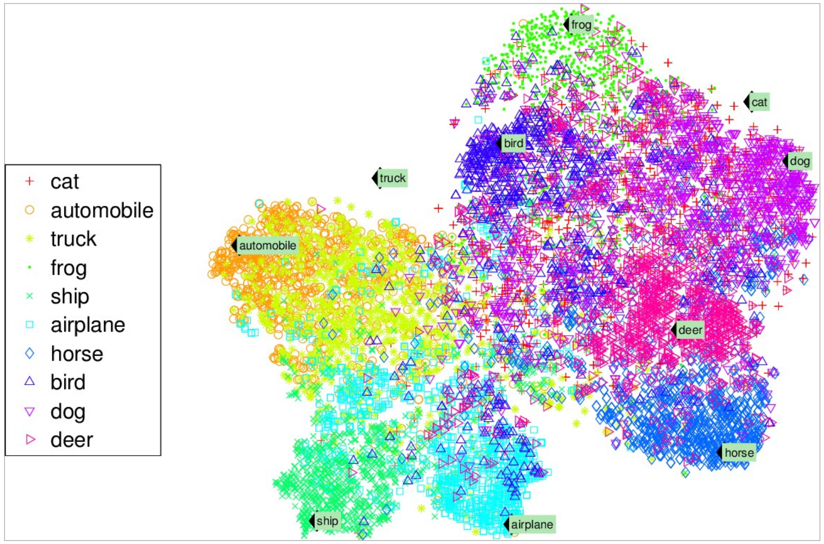
- Clustering with Unsupervised Learning





Unsupervised Learning

- Visualization with Unsupervised Learning
  - t-SNE visualization highlighting semantic clusters
  - Animals are rather well separated from vehicles, how horses are close to deer but far from birds, and so on.

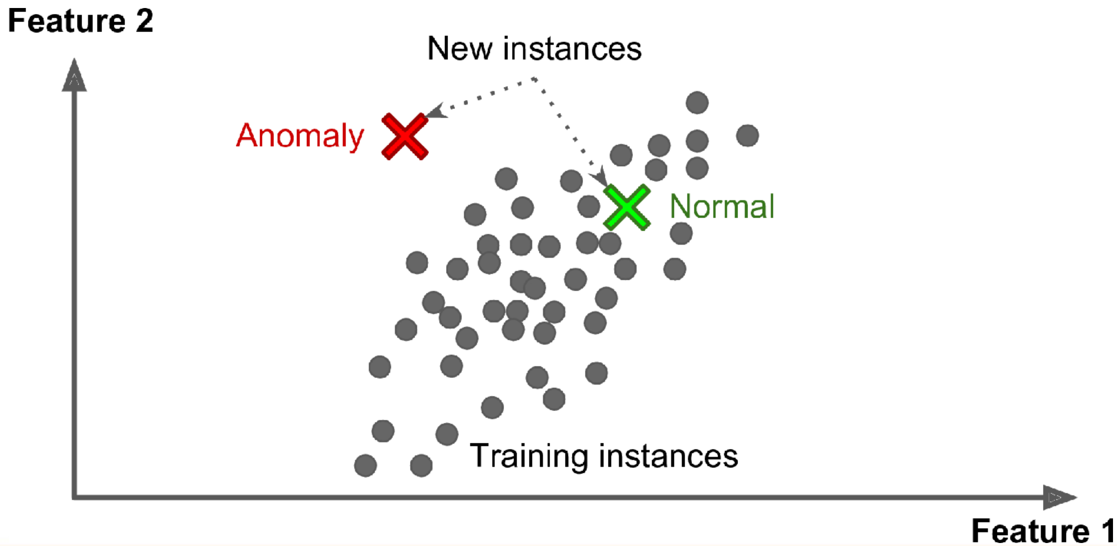


Unsupervised Learning

- Dimensionality Reduction
  - To simplify the data without losing too much information such as merging several correlated features into one.
- Example
  - A car’s mileage is highly correlated with its age, so the dimensionality reduction algorithm will merge them into ‘1 feature’ that represents the car’s wear and tear.
  - Feature Extraction
- Benefits
  - Increasing performance and decreasing storage consumption

Unsupervised Learning

- Anomaly Detection
  - Train the system with normal instances
- Example
  - Detecting unusual credit card transactions, catching manufacturing defects



Unsupervised Learning

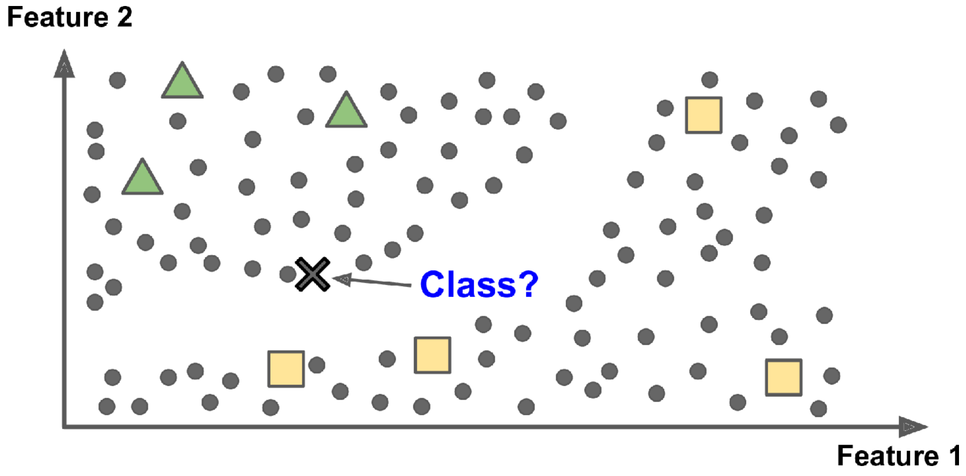
- Association Rule Learning
  - The goal is to dig into large amounts of data and discover interesting relations between attributes.
- Example
  - For a supermarket, running an association rule on the sales logs may reveal that people who purchase barbecue sauce and potato chips also tend to buy steak.
  - Place these items close to each other.

Applications of Unsupervised Learning

1. **Market Segmentation**: Grouping customers based on purchasing behavior without pre-labeled categories.
2. **Social Network Analysis**: Detecting communities or groups within large social networks based on interaction patterns or shared interests.
3. **Recommender Systems**: Recommending products or content by identifying items that are similar to what a user likes, based on purchase history, viewed content, etc.
4. **Medical Imaging**: Grouping similar medical images, like MRI or X-ray scans, to identify patterns or anomalies without labeled data.
5. **Financial Systems**: Detecting abnormal patterns in transaction data, which can be a sign of fraudulent activities.
6. **Real Estate Market Analysis**: Segmenting properties into groups based on features like location, size, and amenities to understand market trends.
7. **Content Aggregation**: Grouping online articles or news stories that cover similar topics or events.

Semi-supervised Learning

- Combination of unsupervised and supervised algorithms
- To deal with partially labeled training data
  - A lot of <u>unlabeled</u> data and some <u>labeled</u> data
- Example
  - Google Photo
  - To recognize who shows up in photos by using family photos



Semi-supervised Learning

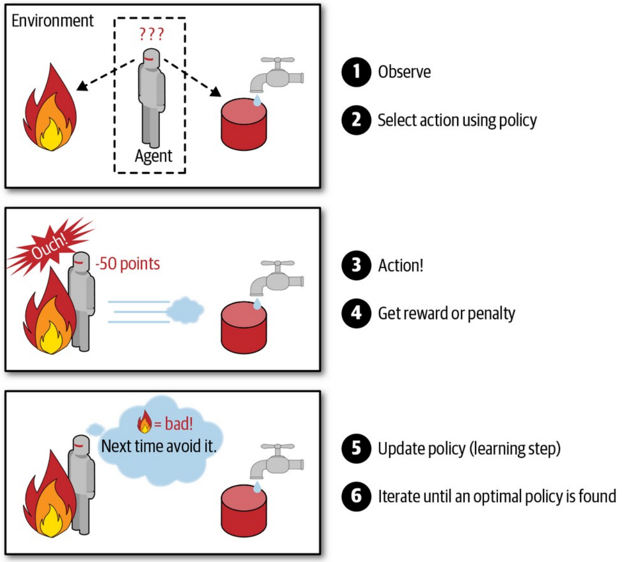
- Algorithm
  - Deep Belief Networks (DBNs) are based on unsupervised components called restricted Boltzmann machines (RBMs) stacked on top of one another.
  - RBMs are trained sequentially in an unsupervised manner, then the whole system is fine-tuned using supervised learning techniques.

Applications of Semi-supervised Learning

1. **Web Page Classification**: Using both labeled and unlabeled web pages to categorize them into topics.
2. **Bioinformatics**: Predicting protein structures with limited labeled data.
3. **Voice Command Recognition**: Training voice models with a small amount of labeled data and a larger amount of unlabeled data.
4. **Image Annotation**: Predicting labels for images when only a subset of the dataset has annotations.
5. **Document Clustering and Categorization**: Grouping documents into clusters and then using a small labeled subset to name or categorize those clusters.

Reinforcement Learning

- Agent can observe the environment, select and perform actions, and get rewards or penalties in return.
- The main concern is how agents ought to take actions in an environment to maximize the cumulative reward towards the goal, by utilizing policies.
- A *policy* defines what action the agent should choose when it is in a given situation.



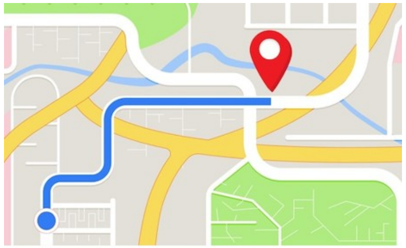
- To optimize the policies incrementally, rather than conventional model training
  - Conventional ML ☾ Train once with Big Data and Predict many times!
  - Reinforcement Learning ☾ Learn Incrementally!

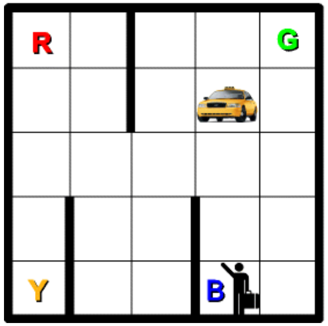
Reinforcement Learning

- To utilize agents
  - To observe the environment
  - To selected and perform actions
  - To get rewards or penalties in return
  - To learn by itself what is the best strategy (i.e. policy)
- Example
  - DeepMind's AlphaGo learns its <u>winning policy</u> by analyzing millions of games, and then playing many games against itself.

Elements of RL System

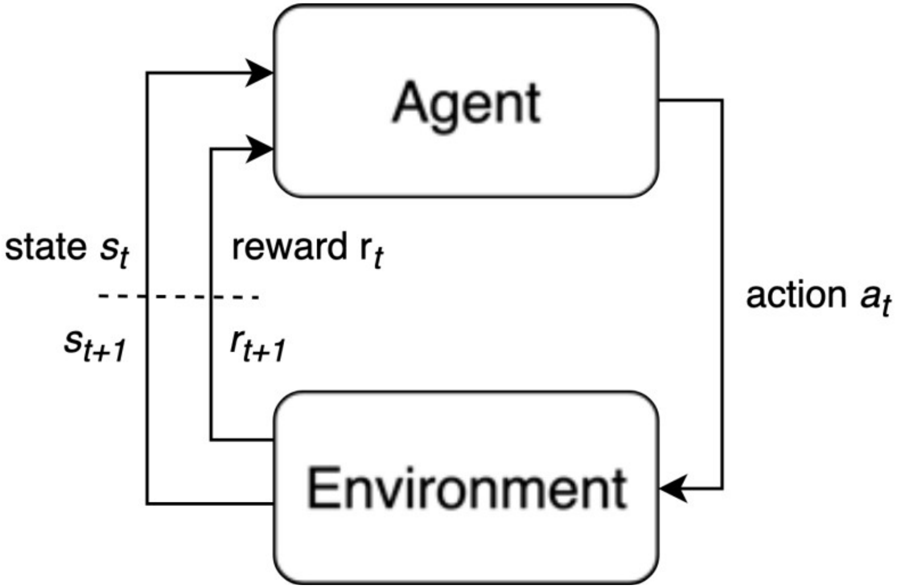
- Agent
  - Object that tries to maximize *rewards* for actions
  - Interacts with the *environment* by observing the *state*
- Goal
  - The milestone that an agent wishes to accomplish
- Environment
  - Representation of the target domain in terms of states
  - Examples
    - Locations of Taxi and Passenger on streets
    - Representation of Go Game board





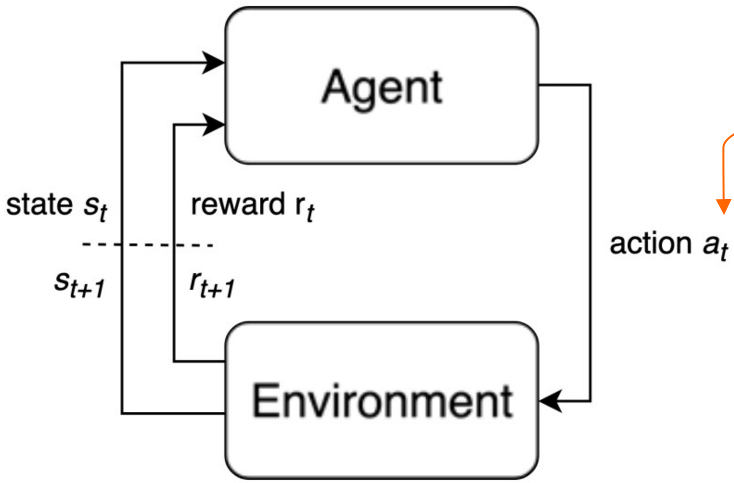
Elements of RL System

- State
  - Representation of the current environment.
  - Examples) State of Go Game, State of Facial Skin Condition
- Action
  - Method that the Agent performs to reach the goal
  - Action causes the change of the state and consequently the environment.
- Reward Function
  - Function that returns a <u>reward value</u> for a performed action
  - The range of a reward value can be:
    - 0..1
    - -1..1



Elements of RL System

- Policy
  - Knowledgebase that captures the rules and patterns of the system.
  - Example
    - For Go Game, the policy contains the game rules and winning strategies.
  - Policy is used to recommend an action for an agent.
    - Can be a machine learning model



Actions

States

	A_1	A_2	...	A_M
S_1	$Q(S_1, A_1)$	$Q(S_1, A_2)$		$Q(S_1, A_M)$
S_2	$Q(S_2, A_1)$	$Q(S_2, A_2)$		$Q(S_2, A_M)$
⋮			⋱	⋮
S_N	$Q(S_N, A_1)$	$Q(S_N, A_2)$	...	$Q(S_N, A_M)$

Policy

Example of RL System

- Alpha Go
  - Agent
    - Software Object, *Go Player*
  - Goal
    - Winning the game by gaining more stones of its color than the other player.
  - Environment
    - Go Game Board with Stones
  - Action
    - Dropping Stone
  - Reward
    - Point for each Action
  - Policy
    - Representation of Knowledge and Strategies

Applications of Reinforcement learning

1. **Game Playing**: Training agents to play and excel at games like Go, chess, and video games.
2. **Robot Navigation**: Allowing robots to learn optimal paths and actions in an environment.
3. **Personalized Recommendations**: Adapting to user preferences and behavior over time to suggest content.
4. **Ad Bidding**: Learning to bid in online advertising to maximize click-through rates.
5. **Self-Driving Cars**: Making driving decisions based on environmental data.

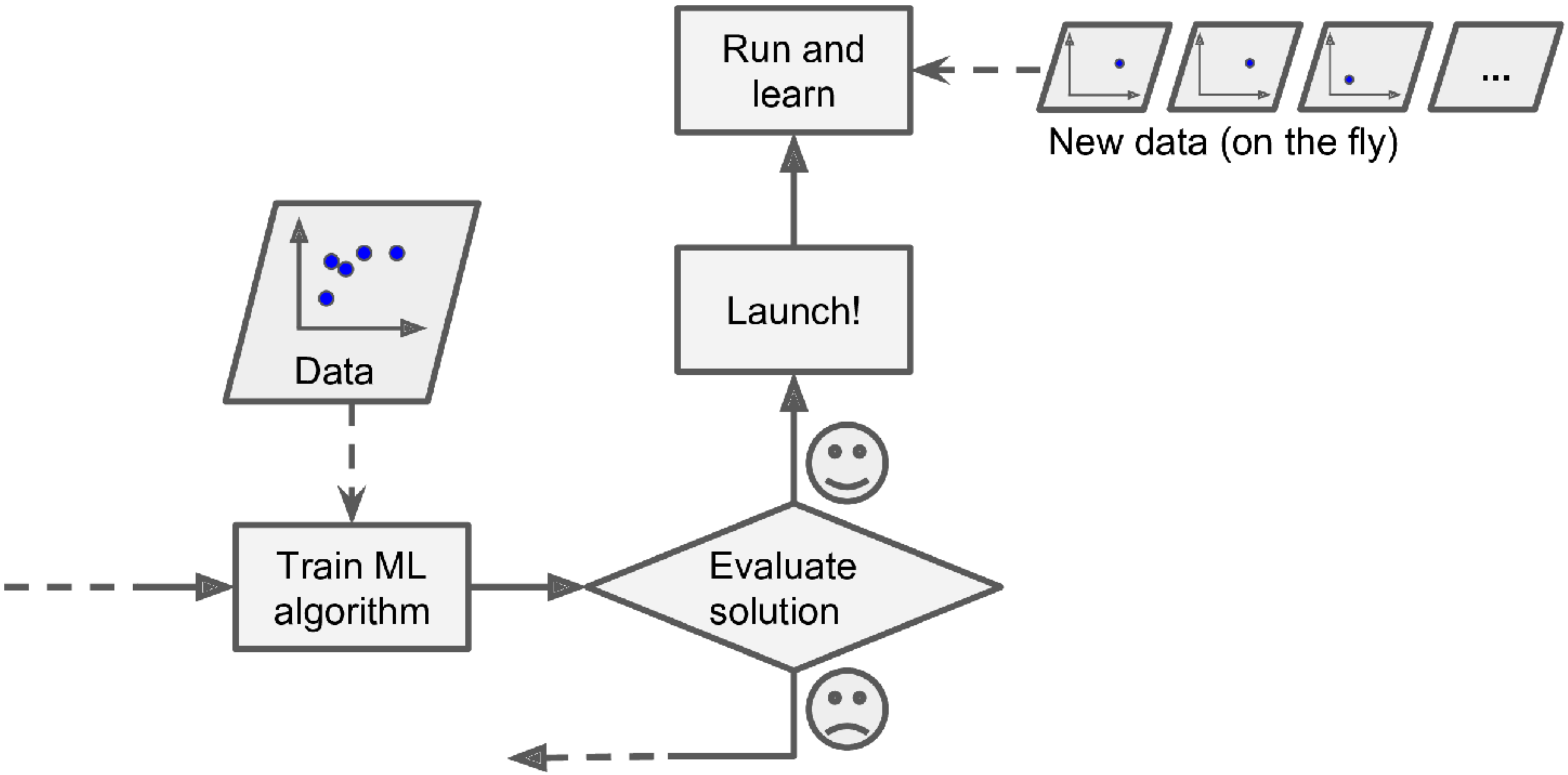
Batch Learning

- Trained with All the available data
  - Requires a high computing resource and time
- Replace the system with new model

Online Learning

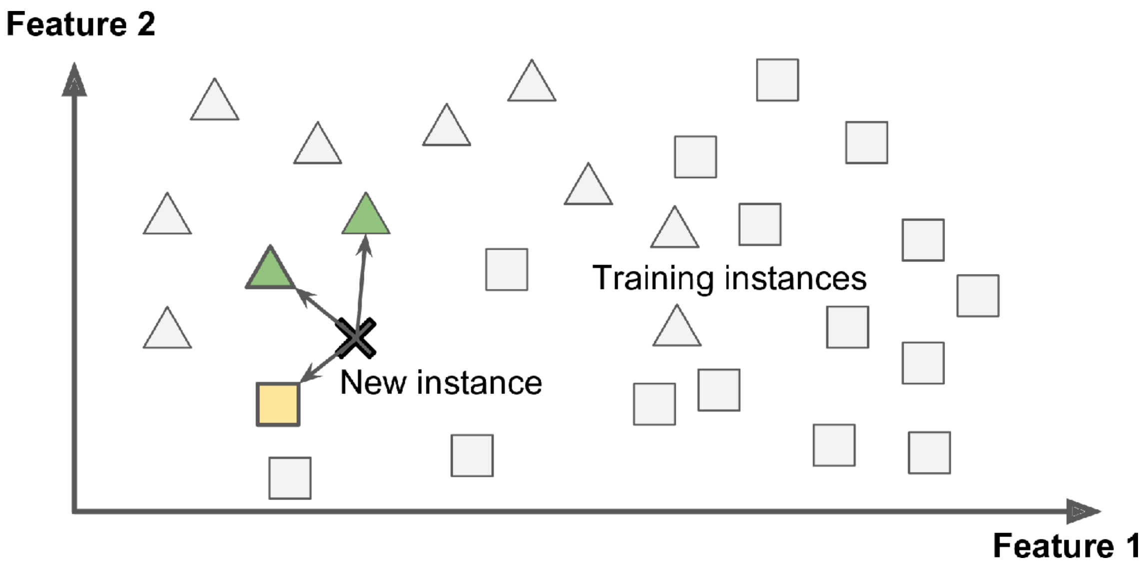
- To train the system incrementally by feeding it data instances sequentially
  - Individually or small groups (i.e. mini-batches)
- Situations
  - When a system receives data as a continuous flow
  - Need to adapt to change rapidly or autonomously
  - Have limited computing resources
- Learning Rate
  - High Rate
  - Low Rate

Online Learning



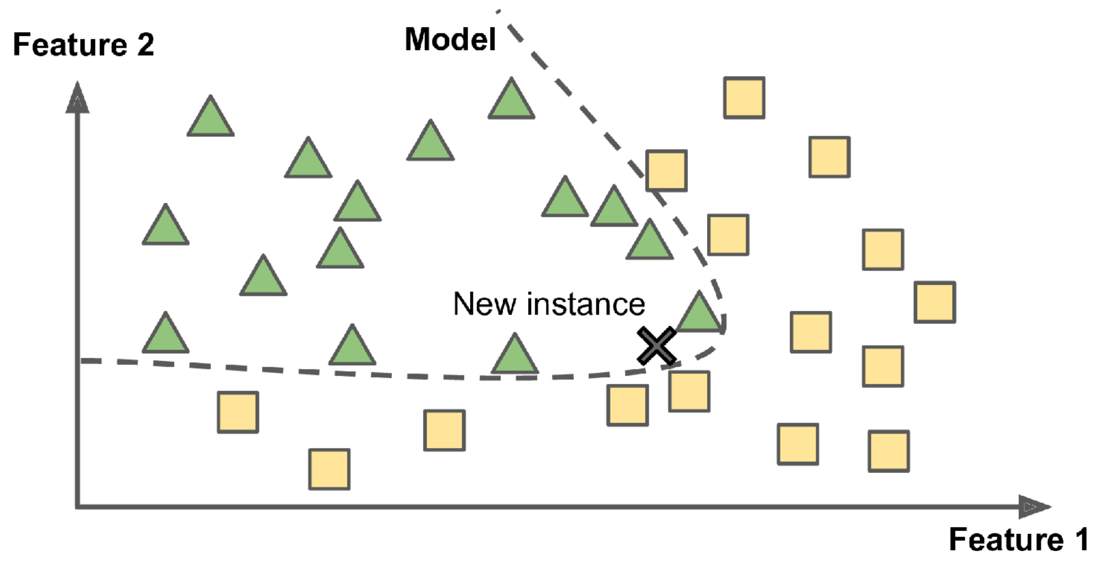
Instance-based Learning

- One of the 2 Approaches to generalization
- To compare all the new data points to known data points and generalize to new cases using a similarity measure
  - Measure of Similarity



Model-based Learning

- To build a model of the training data and use the model to make predictions



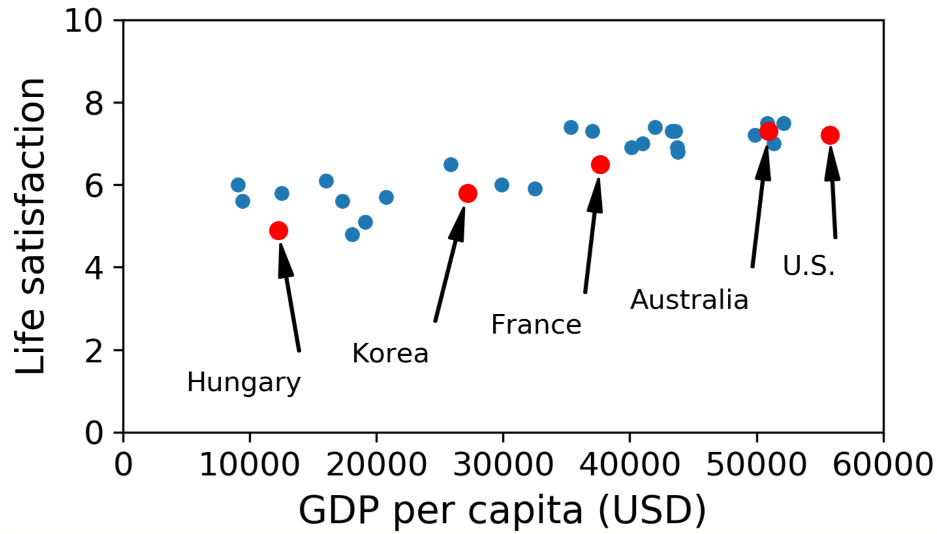
Model-based Learning

- Typical Process
  - Studying the train data
  - Selecting a model
  - Training the model on the training data
    - Learning algorithm searched for the model parameter values that minimize a cost function
  - Applying the model to make predictions on new cases

Model-based Learning (3)

- Example
  - Does money make people happy?
  - Dataset
    - Better Life Index data from the OECD
    - GDP per Capita from IMF
    - Join the tables and sort by GDP per capita.

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

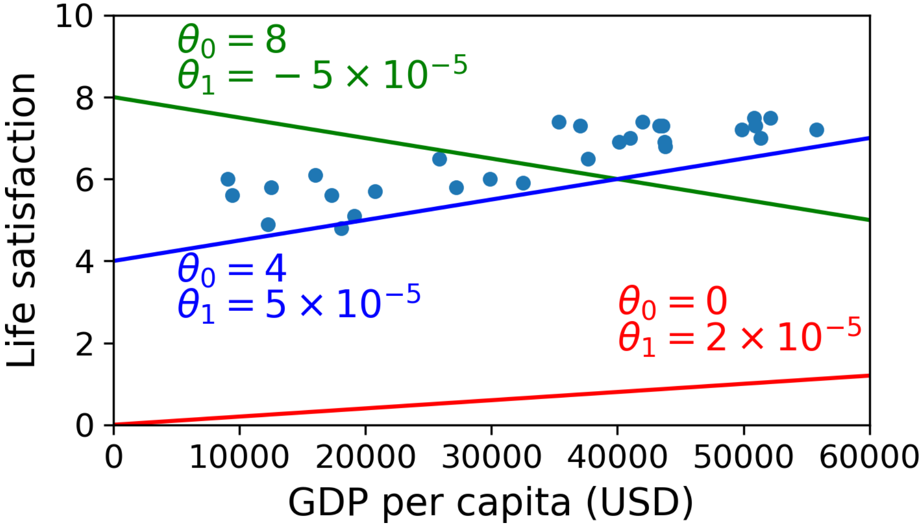


Model-based Learning (4)

- Observe the Trend
  - Life satisfaction goes up more or less linearly as the country's GDP per capita increases.

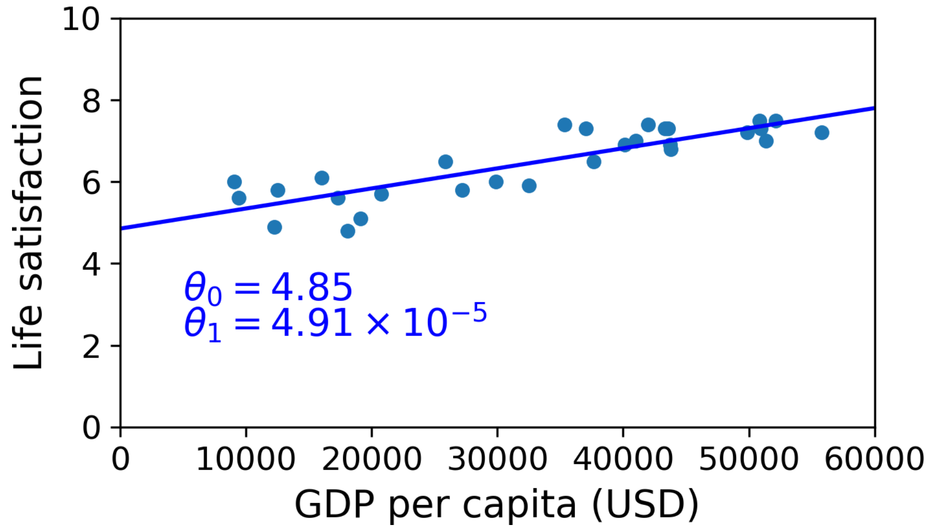
$$life\_satisfaction = \theta_0 + \theta_1 \times GDP\_per\_capita$$

- Model Selection
  - Select a linear model of life satisfaction with just one attribute, *GDP per capita*.



Model-based Learning (5)

- To determine the values of the parameters, specify a performance measure.
  - Either define a utility function (or fitness function) that measures how good your model is,
  - or define a cost function that measures how bad it is.
    - Such as linear model



Model-based Learning (6)

- Training and running a linear model using Scikit-Learn

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import sklearn.linear_model

# Load the data
oecd_bli = pd.read_csv("oecd_bli_2015.csv", thousands=',')
gdp_per_capita = pd.read_csv("gdp_per_capita.csv",thousands=',',delimiter='₩t',
                             encoding='latin1', na_values="n/a")


# Prepare the data
country_stats = prepare_country_stats(oecd_bli, gdp_per_capita)
X = np.c_[country_stats["GDP per capita"]]
y = np.c_[country_stats["Life satisfaction"]]

# Visualize the data
country_stats.plot(kind='scatter', x="GDP per capita", y='Life satisfaction')
plt.show()

# Select a linear model
model = sklearn.linear_model.LinearRegression()

# Train the model
model.fit(X, y)

# Make a prediction for Cyprus
X_new = [[22587]]  # Cyprus's GDP per capita
print(model.predict(X_new)) # outputs [[ 5.96242338]]
```

References

- Russel and Norvig, Artificial Intelligence: A Modern Approach, 3rd edition, Prentice Hall, 2010.