Introduction to Machine Learning with Python

PIKAKSHI MANCHANDA

POST DOCTORAL RESEARCH FELLOW, VISTA AR

@itsPikakshi

p.Manchanda@Exeter.ac.uk

Content Overview

- What is Machine Learning?
- Why Machine Learning is important?
- Examples
- Types of Machine Learning
- How does it work?

Available on github.com/Pikakshi/Advanced_NLP_with_ML

What is Machine Learning?

- Field of study that gives computers the ability to learn without being explicitly programmed -Arthur Samuel, 1959.
  - Wrote a checkers playing program
  - Program learned by observing board positions

- Study of algorithms that improve their performance P at some task T with experience E - Tom Mitchell, 1998.
  - A well-defined learning task is given by <*P, T, E*>.
  - ✓ T: Playing checkers
  - ✓ P: The number (or percentage) of games won
  - ✓ E: Playing against oneself.

Why is it important?

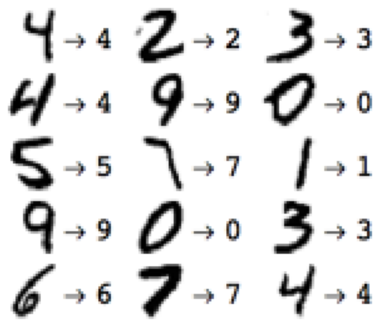
- **Machine learning is a subfield of artificial intelligence**.
- Its goal is to enable computers to learn on their own.
- Machine learning is at the core of AI → it will change every industry and have a massive impact on our day-to-day lives.
- A research report by McKinsey Global Institute(Sep-2018 report) suggests that ‘*Artificial intelligence has the potential to incrementally add 16% or around $13 trillion to the US economy by 2030*’.
- Growing volumes and varieties of data. More and more powerful computational processing. Extensive data storage capabilities. ➔ Better chances at building precise ML models capable of analysing complex and huge quantities of data.

Examples of Machine Learning

- Handwriting Recognition
- Speech Recognition
- Image Tagging
- Fraud Detection
- Virtual Assistants and Chatbots
- Self driving cars
- Stock Market Predictions
- Recommender Systems: Netflix, Spotify, Amazon, etc.
- Text Analysis: Sentiment Analysis, Cluster Analysis, Topic Detection, Entity Recognition, Spam Detection, Document Similarity, ..

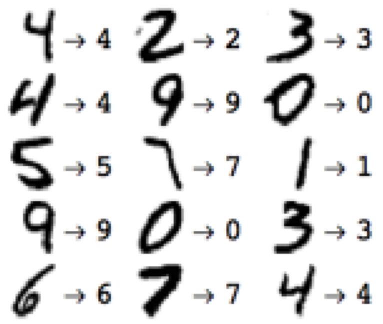
Examples of Machine Learning

- Handwriting Recognition
  - Task T: recognizing and classifying handwritten words within images
  - Performance P: percent of words correctly classified
  - Training experience E: a database of written words with given classification
  - Use of algorithms like *Neural Networks, Support Vector Machines*.



Examples of Machine Learning

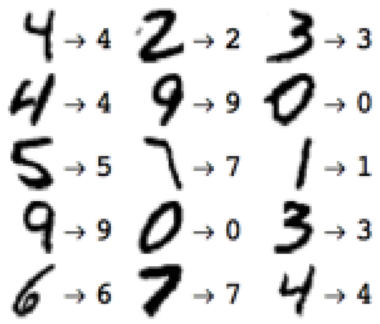
- Handwriting Recognition
  - Task T: recognizing and classifying handwritten words within images
  - Performance P: percent of words correctly classified
  - Training experience E: a database of written words with given classification
  - Use of algorithms like *Neural Networks, Support Vector Machines*.



- Fraud Detection
  - Task T: Recognize presence of fraud among business transactions
  - Performance P: percent of fraudulent payments correctly detected
  - Experience E: a database of records with labelled transactions.
  - Use of algorithms such as *Neural Network, Logistic Regression, Random Forest*.

Examples of Machine Learning

- Handwriting Recognition
  - Task T: recognizing and classifying handwritten words within images
  - Performance P: percent of words correctly classified
  - Training experience E: a database of written words with given classification
  - Use of algorithms like *Neural Networks, Support Vector Machines*.



- Fraud Detection
  - Task T: Recognize presence of fraud among business transactions
  - Performance P: percent of fraudulent payments correctly detected
  - Experience E: a database of records with labelled transactions.
  - Use of algorithms such as *Neural Network, Logistic Regression, Random Forest*.

- How about Facebook's Image Tagging?

- And Sentiment Analysis?

Types of Machine Learning

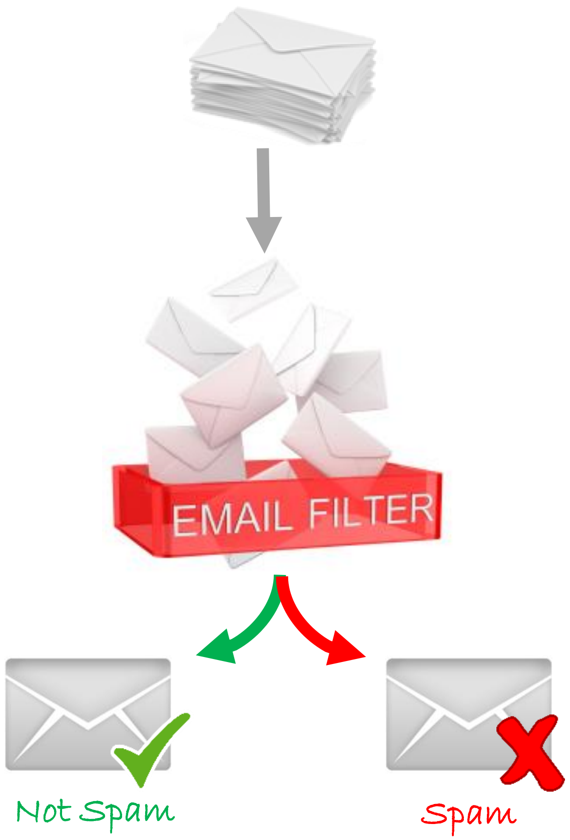
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

1. Supervised Learning

- Task Driven learning.
- Making predictions using data.

1. Supervised Learning

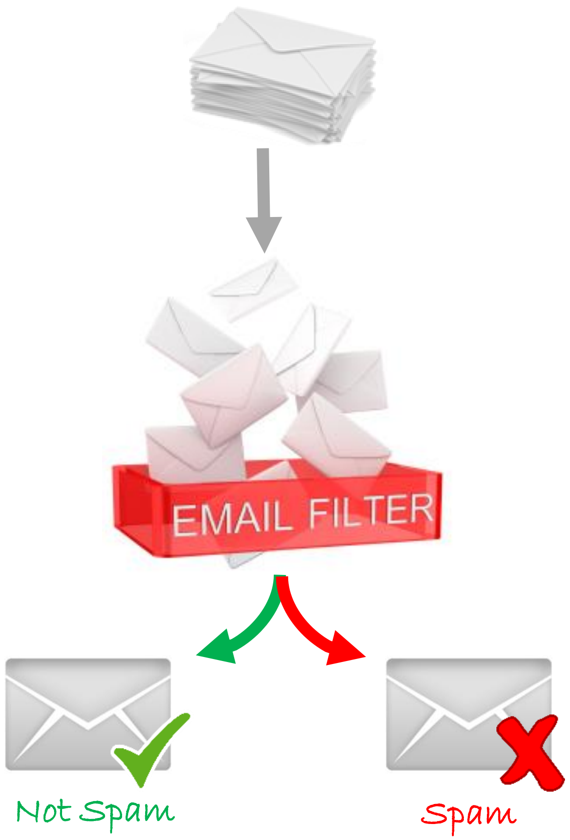
- Task Driven learning.
- Making predictions using data.
- Consider the problem of *email spam detection* -- predicting whether an incoming email is spam or not.



1. Supervised Learning

- Task Driven learning.
- Making predictions using data.
- Consider the problem of *email spam detection* -- predicting whether an incoming email is spam or not.
  - ✓ Task T: Categorize email messages as spam or legitimate.
  - ✓ Performance P: Percentage of email messages correctly classified.
  - ✓ Experience E: Database of emails, some with human-given labels.

➔ Given a dataset with 'right answers', an algorithm learns to produce predictions on never-before-seen data.



Terminology:

- *Label*: Variable we're predicting – usually represented by the variable **y**
- *Features*: Input variables describing data – usually represented by variables $\{x_1, x_2, \ldots, x_n\}$
- *Example*: particular instance of data, **x**
- *Labeled Example*: has **{features,label}**: **(x,y)** – used to train the model
  - Input data with labeled examples form the *training dataset*.
- *Unlabeled Example*: has **{features,?}**: **(x,?)** – used for making predictions on new data
  - Collection of unlabeled examples are the *test dataset* which are used to test the performance of the trained model.
- *Model*: maps examples to predicted labels **y'**

How does it work?

1. Training the Machine Learning Algorithm using **labelled data**.

- The model learns the relationship between attributes of input data and the outcome.
- The goal is to approximate a mapping function which can predict the output variable **(Y)** for a new input data **(x)**, i.e.,

$$Y = f(X)$$

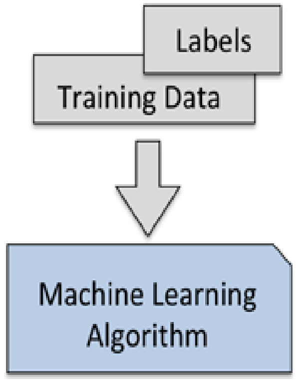
How does it work?

1. Training the Machine Learning Algorithm using **labelled data**.

- The model learns the relationship between attributes of input data and the outcome.
- The goal is to approximate a mapping function which can predict the output variable **(Y)** for a new input data **(x)**, i.e.,

$$Y = f(X)$$

In other words, the algorithm learns by comparing its output with the correct outputs to find errors and then modifies the model accordingly.



How does it work?

1. Training the Machine Learning Algorithm using **labelled data**.

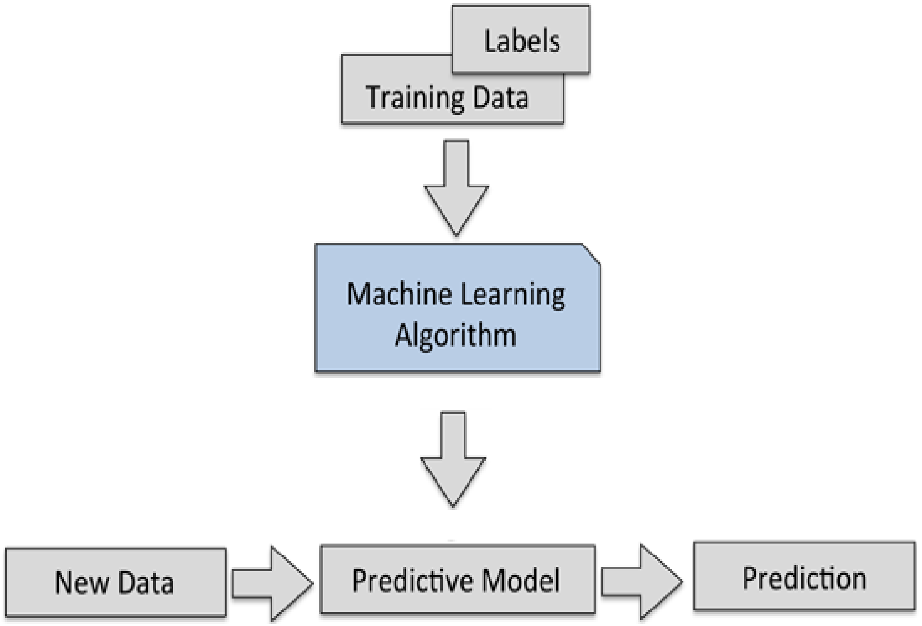
- The model learns the relationship between attributes of input data and the outcome.
- The goal is to approximate a mapping function which can predict the output variable **(Y)** for a new input data **(x)**, i.e.,

$$Y = f(X)$$

In other words, the algorithm learns by comparing its output with the correct outputs to find errors and then modifies the model accordingly.

2. **Predictions** on **new (future) data** for which label is unknown using the trained model to predict **future outputs**.

➔ Predictive Modeling.



Types of Supervised Learning

REGRESSION

- Learn a function ***f(x)*** to predict ***y*** given ***x***, where y is a real-valued continuous output (eg: housing prices, monthly income)
- Continuous means there aren't gaps (discontinuities) in the value that Y can take on.
- Popular algorithms:
  - Linear Regression (simple/MLR)
  - Support Vector Machines
  - Random Forest
  - Neural Network
  - Decision Trees

CLASSIFICATION

- Learn a function ***f(x)*** to predict ***y*** given ***x***, where y is a discrete categorical output (eg: spam/not spam, male/female).
- Discrete means that Y can take on only a finite number of values.
- Popular algorithms:
  - Logistic Regression
  - Naïve Bayes
  - Decision Trees
  - Support Vector Machines (SVM)
  - Random Forest
  - KNN

Regression

- ➢ Consider the problem of predicting housing prices.
- ➢ Features: Input variables that can be used to predict housing prices such as: size ($feet^2$), number of bedrooms, number of floors, age of house (years)
  - Lets consider one input variable (size in sq. ft) → Univariate/Simple Linear Regression
- ➢ **Simple LR**: Finds a linear function (a straight line) that predicts the target variable (y) as a function of the independent variable (x).

Estimated Price

Slide credit: Andrew Ng

Regression

- Consider the problem of predicting housing prices.
- Features: Input variables that can be used to predict housing prices such as: size (feet^2), number of bedrooms, number of floors, age of house (years)
  - Lets consider one input variable (size in sq. ft) → Univariate/Simple Linear Regression
- **Simple LR**: Finds a linear function (a straight line) that predicts the target variable (y) as a function of the independent variable (x).

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

Features/Independent Variables

Target Variable

Slide credit: Andrew Ng

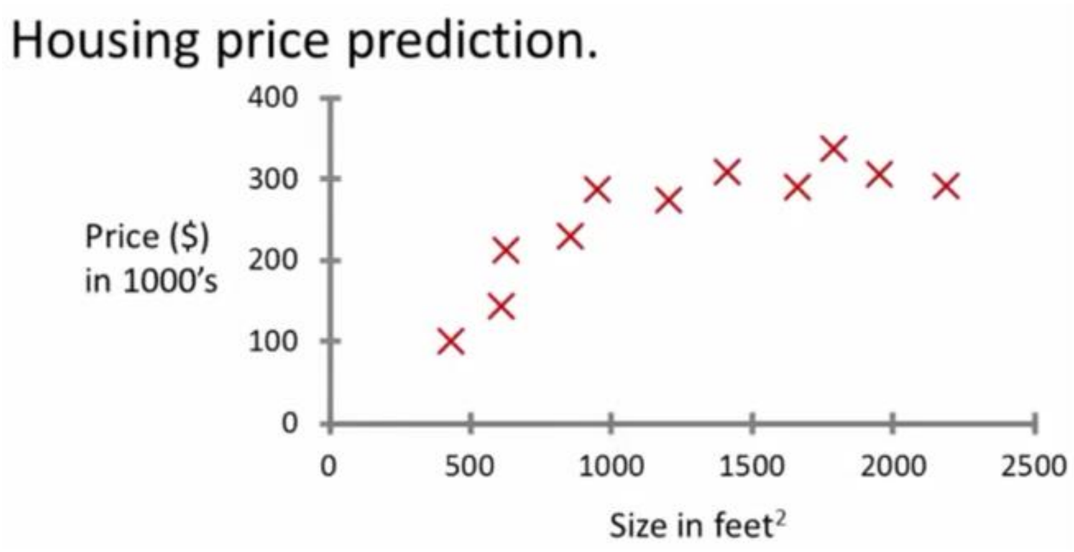
Regression

- Consider the problem of predicting housing prices.
- Features: Input variables that can be used to predict housing prices such as: size (feet^2), number of bedrooms, number of floors, age of house (years)
  - Lets consider one input variable (size in sq. ft) → Univariate/Simple Linear Regression
- **Simple LR**: Finds a linear function (a straight line) that predicts the target variable (y) as a function of the independent variable (x).

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

Features/Independent Variables

Target Variable



Regression

➢ Consider the problem of predicting housing prices.

➢ Features: Input variables that can be used to predict housing prices such as: size (feet^2), number of bedrooms, number of floors, age of house (years)

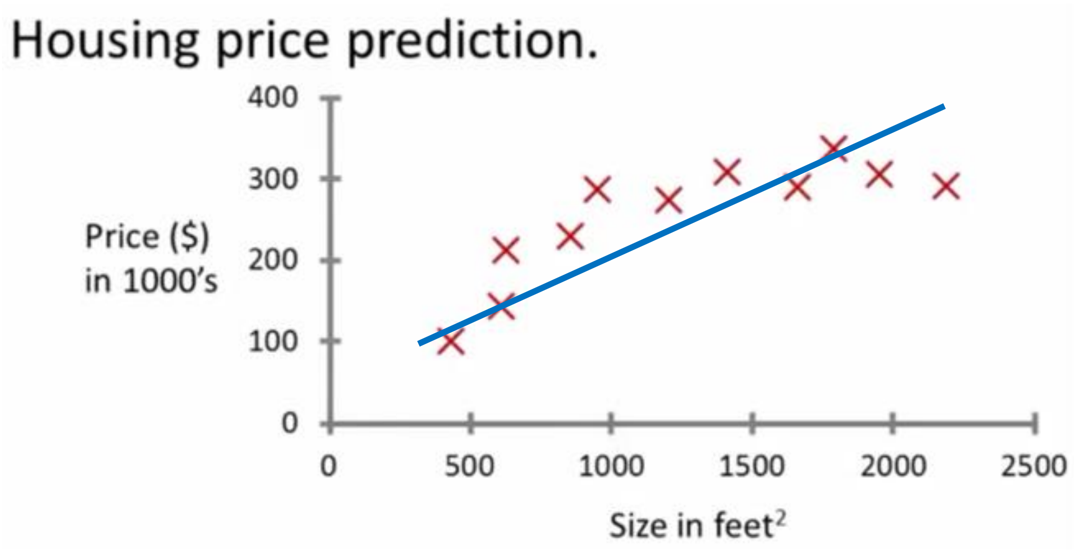
- Lets consider one input variable (size in sq. ft) → Univariate/Simple Linear Regression

➢ **Simple LR**: Finds a linear function (a straight line) that predicts the target variable (y) as a function of the independent variable (x).

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Features/Independent Variables

Target Variable



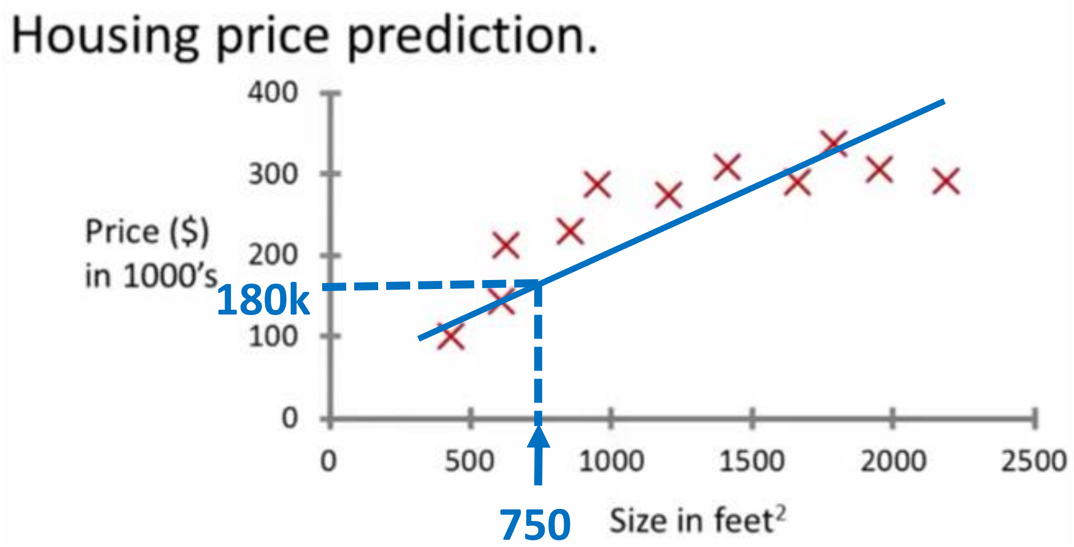
Regression

- Consider the problem of predicting housing prices.
- Features: Input variables that can be used to predict housing prices such as: size (feet^2), number of bedrooms, number of floors, age of house (years)
  - Lets consider one input variable (size in sq. ft) → Univariate/Simple Linear Regression
- **Simple LR**: Finds a linear function (a straight line) that predicts the target variable (y) as a function of the independent variable (x).

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

Features/Independent Variables

Target Variable



Slide credit: Andrew Ng

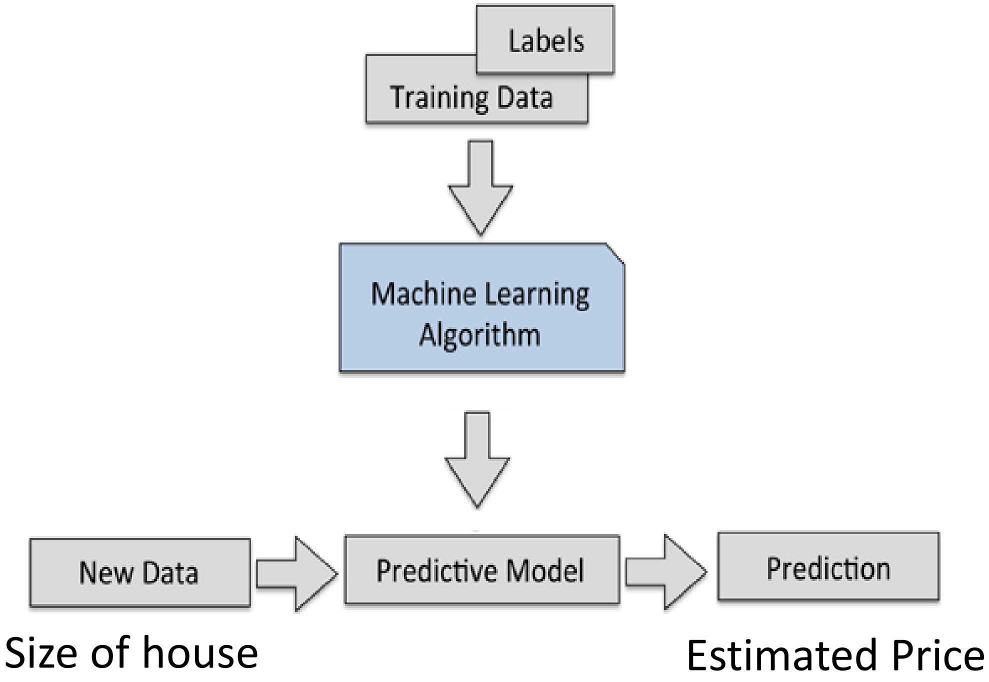
Regression

- ➢ Consider the problem of predicting housing prices.
- ➢ Features: Input variables that can be used to predict housing prices such as: size (feet^2), number of bedrooms, number of floors, age of house (years)
  - Lets consider one input variable (size in sq. ft) → Univariate/Simple Linear Regression
- ➢ **Simple LR**: Finds a linear function (a straight line) that predicts the target variable (y) as a function of the independent variable (x).

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Features/Independent Variables

Target Variable



Regression

- Consider the problem of predicting housing prices.
- Features: Input variables that can be used to predict housing prices such as: size (feet^2), number of bedrooms, number of floors, age of house (years)
  - Consider multiple input variables → Multiple Linear Regression (MLR)
- **MLR**: Model a linear function that predicts the target variable as a function of the independent variables.

Regression

- Consider the problem of predicting housing prices.
- Features: Input variables that can be used to predict housing prices such as: size (feet^2), number of bedrooms, number of floors, age of house (years)
  - Consider multiple input variables → Multiple Linear Regression (MLR)
- **MLR**: Model a linear function that predicts the target variable as a function of the independent variables.

Size (feet^2)	Number of bedrooms	Number of floors	... Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

Features/Independent Variables

Target Variable

Regression

- Consider the problem of predicting housing prices.
- Features: Input variables that can be used to predict housing prices such as: size (feet^2), number of bedrooms, number of floors, age of house (years)
  - Consider multiple input variables → Multiple Linear Regression (MLR)
- **MLR**: Model a linear function that predicts the target variable as a function of the independent variables.

n features

Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	... Age of home (years) x_n	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

m training examples

Features/Independent Variables

Target Variable

Types of Supervised Learning

REGRESSION

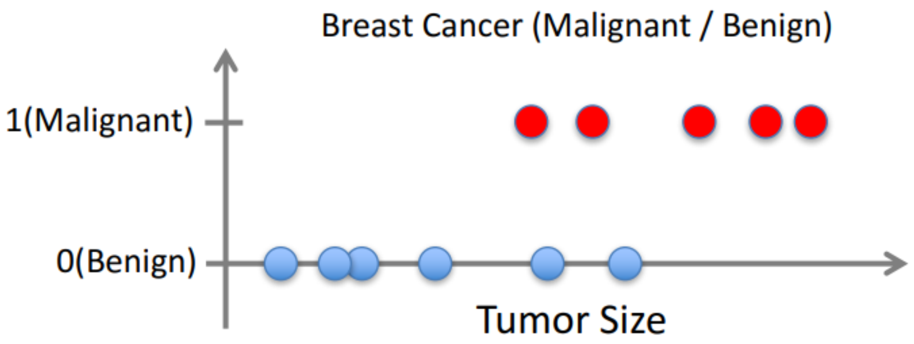
- Learn a function ***f(x)*** to predict ***y*** given ***x***, where y is a real-valued continuous output (eg: housing prices, monthly income)
- Continuous means there aren't gaps (discontinuities) in the value that Y can take on.
- Popular algorithms:
  - Linear Regression (simple/MLR)
  - Support Vector Machines
  - Random Forest
  - Neural Network
  - Decision Trees

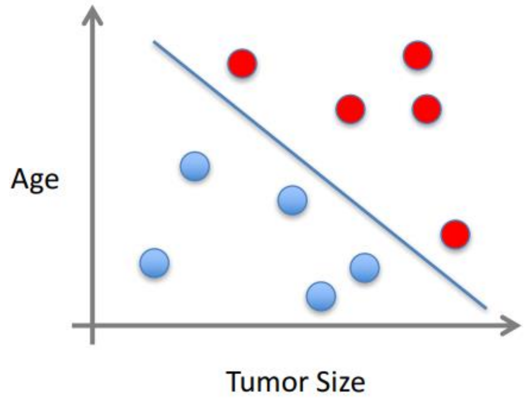
CLASSIFICATION

- Learn a function ***f(x)*** to predict ***y*** given ***x***, where y is a discrete categorical output (eg: spam/not spam, male/female).
- Discrete means that Y can take on only a finite number of values.
- Popular algorithms:
  - Logistic Regression
  - Naïve Bayes
  - Decision Trees
  - Support Vector Machines (SVM)
  - Random Forest
  - KNN (K Nearest Neighbour)

Classification

- ➢ Binary Classification: Spam/no spam, cancer/no cancer
  - ▪ Using one input variable
  - ▪ Using more than one input variable





- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

…

- ➢ Multi-class Classification: Handwritten Digit Recognition (0 to 9), Cancer stage (0, 1, 2, 3)

Decision Trees for Classification

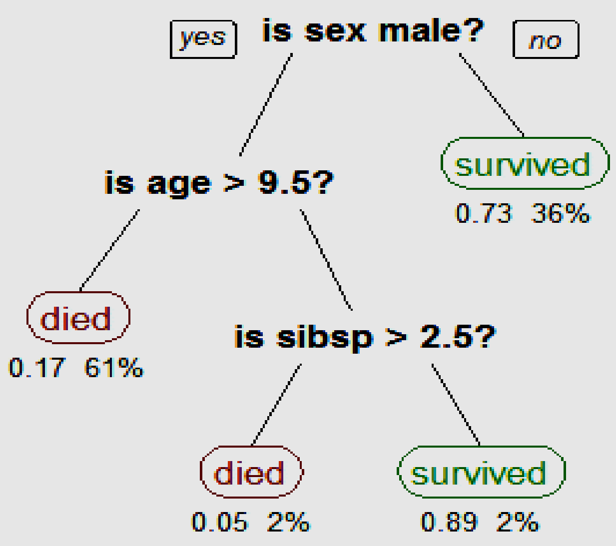


Image taken from wikipedia

- Uses a tree-like model for decisions.
- Visually and explicitly represents decisions and decision-making.
- Drawn upside down with the root at the top.
- Consider an example of the titanic dataset for predicting whether a passenger will survive or not (y).
  - Features (x_1, x_2, .., x_n): gender, age, and number of spouses or children aboard
  - Condition/**internal node** based on which the tree splits into branches/ **edges**.
  - End of the branch that doesn't split anymore is the decision/**leaf**, in this case, whether the passenger died or survived, represented as red and green text respectively.
  - From the tree it is seen that if you were (*i*) a female or (*ii*) a male younger than 9.5 years with less than 2.5 siblings, your survival chances were good.
  - The figures under the leaves show the probability of survival and the percentage of observations in the leaf.

Making it work

A Machine Learning project has a series of well known steps:

- Define the problem
- Load data
- Evaluate Algorithms
- Make Predictions

Machine Learning in Python

Installing the libraries

- *pandas:* Library for pre-processing data and convert data into Data Frames (similar to database tables).
- *SciPy:* Library used for scientific and technical computing.
- *NumPy:* Library for mathematical and scientific computing library for Python
- *matplotlib:* Library for visualising data and results.
- *Scikit-learn:* provides a consistent interface to ML models and covers libraries like *NumPy, SciPy* and *matplotlib*
- *Keras:* Library that encapsulates complex Deep Learning frameworks.

Installing scikit-learn: 2 options:

1. Install the library with the dependencies (NumPy and SciPy)
   - `pip install scikit-learn`
   - `pip install numpy`
   - `pip install scipy`
2. Install the Anaconda Distribution of Python
   - Getting started manual available here for Windows/Linux/macOS.
   - `conda install scikit-learn`

Supervised Learning (Classification) in Python

✓ Lets consider a *multi-class classification problem* – classification of iris flowers using the famous iris dataset. The dataset has:

- 4 attributes/input variables: sepal length, sepal width, petal length, petal width (in cm) ➔ n= 4
- 150 rows/training examples ➔ m = 150
- Target classes (3 species of 3 different types of Iris flowers): Iris Setosa, Iris Versicolour, Iris Virginica

- Problem definition: Predict the species of an iris flower given its sepal and petal measurements.

- Step-by-step tutorial provided here.

Supervised Learning (Regression) in Python

Problem Definition: Regression problem for *Product Sales Prediction* using an advertising dataset. The dataset has the following features:

- TV: advertising dollars spent on TV for a single product in a given market (in thousands of dollars)
- Radio: advertising dollars spent on Radio
- Newspaper: advertising dollars spent on Newspaper

$\rightarrow$ n = 3

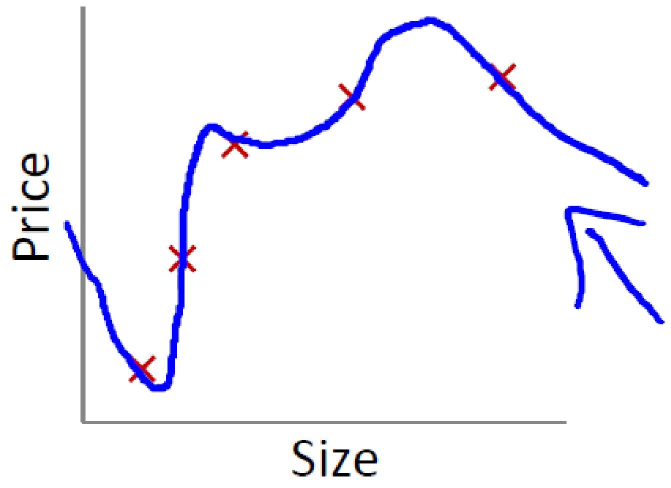
- m = 200 training examples
- Target variable is continuous valued which is why this is a **regression problem**.

- Step-by-step tutorial provided here.

Problems faced

1. **Overfitting**: Learning a function that perfectly explains the training data that the model learned from, but doesn't generalize well to unseen data.

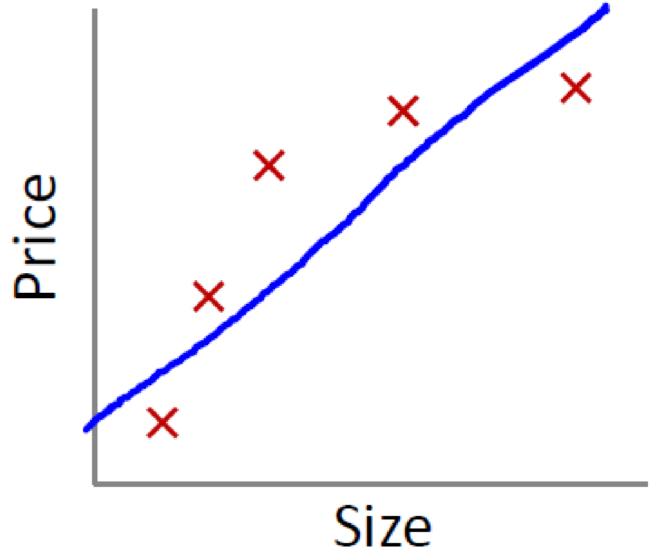
- Happens when a model overlearns from the training data to the point that it starts picking up idiosyncrasies that aren't representative of patterns in the real world.
- Leads to **high variance**.
- **Variance**: how much your model's test error changes on variation in training data. Reflects the model's sensitivity to the idiosyncrasies of the data set it was trained on.



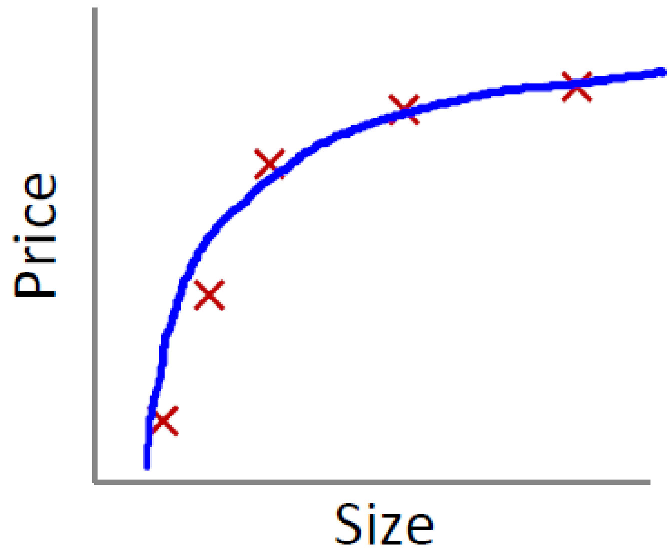
2. Underfitting: Model is not complex enough to capture the underlying trend in the data.

- Leads to **high bias**.
- Bias: Amount of error introduced by approximating real-world phenomena with a simplified model.

- For a good ML model → low bias, low variance.



Underfit → High bias

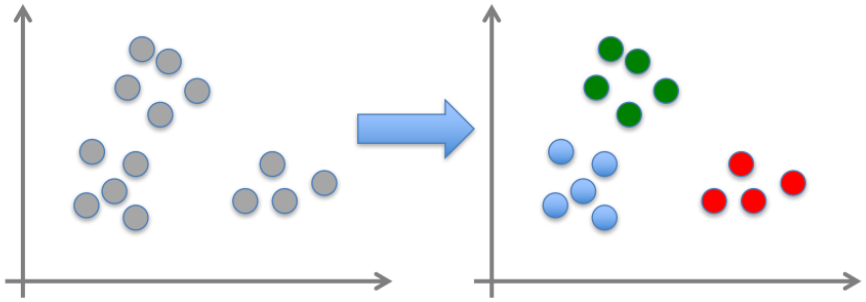


Just Right

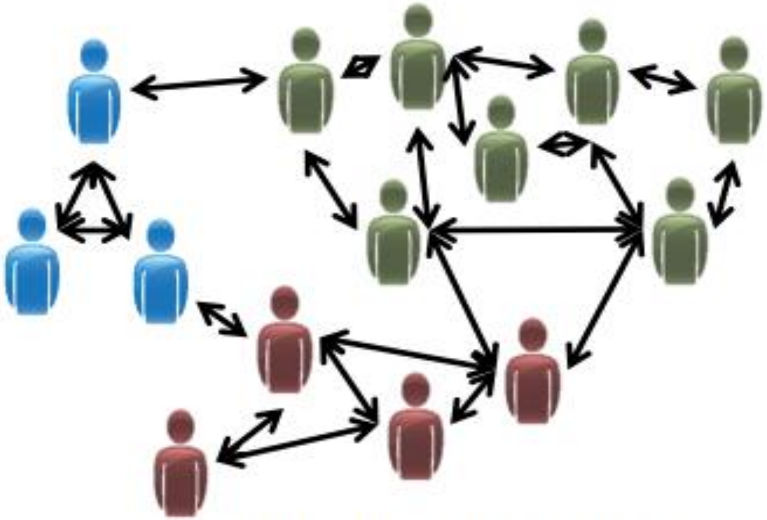
2. Unsupervised Learning

- Data driven learning.
- Extracting structure from data.
- Consider the problem of *market segmentation*.
  - Given a dataset of characteristics and purchasing behaviour of shoppers
  - Unsupervised Learning Task: Segment shoppers into groups or clusters exhibiting similar behaviour
  - No right or wrongs about number of clusters that can be found, which shopper belongs to which cluster or how to describe a cluster.

➔ Given an unlabeled dataset $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$: an algorithm finds hidden structures in the data.



Organize computing clusters



Social network analysis

Market segmentation

Astronomical data analysis

Types of Unsupervised Learning

CLUSTERING

- Discovering inherent groupings or clusters in data. For instance, market segmentation.
- Popular clustering algorithms:
  - K-means
  - Hierarchical clustering
  - KNN (K Nearest Neighbour)
  - Principle Component Analysis

ASSOCIATION

- Association rules establish associations amongst data objects, for instance, in large databases.
- These rules can be used where you want to describe large portions of your data, such as people that buy X also tend to buy Y.
- For instance, people that buy a new home most likely buy new furniture → Market Basket Analysis.
- Popular algorithms for extracting association rules:
  - Apriori algorithm
  - FP-Growth algorithm

Questions

Which learning algorithm (supervised or unsupervised) should be applied for the following problems?

- Given a set of news articles on the web, group them into set of articles about the same story.
- Given a database of customer data, automatically discover market segments and group customers into different market segments.
- Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not.
- Given an inventory of identical items, predict how many of items will sell over next 3 months.

Slide credit: [Andrew Ng]

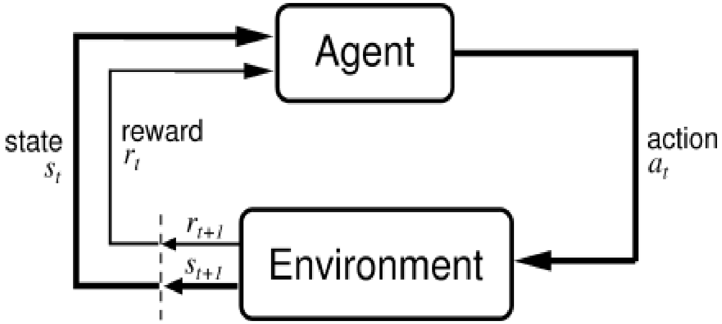
3. Reinforcement Learning

- Learning what to do and how to map **situations** to **actions**.
- The **agent** is not told which action to take, but instead must discover which action will yield the maximum **reward (goal)**. The end result is to maximize the numerical reward signal.
- Algorithm learns to react to an **environment**.
- For instance: a toddler learning to walk, self driving cars,..

Slide credit: [Sutton & Barto]

3. Reinforcement Learning

- Learning what to do and how to map **situations** to **actions**.
- The **agent** is not told which action to take, but instead must discover which action will yield the maximum **reward (goal)**. The end result is to maximize the numerical reward signal.
- Algorithm learns to react to an **environment**.
- For instance: a toddler learning to walk, self driving cars,..



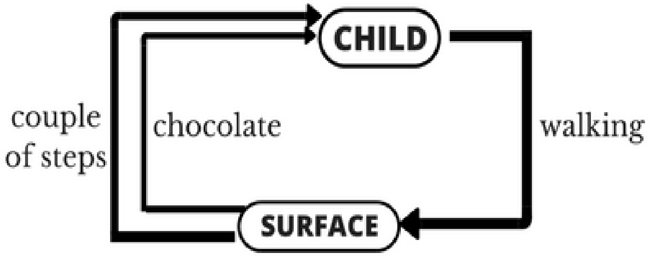
Agent and environment interact at discrete time steps : $t = 0, 1, 2, K$

Agent observes state at step t: $s_t \in S$

produces action at step t: $a_t \in A(s_t)$

gets resulting reward : $r_{t+1} \in \Re$

and resulting next state : s_{t+1}



Slide credit: Sutton & Barto

Semi-supervised Learning

- Represent a middle ground between supervised and unsupervised ML algorithms.
- Huge amount of input data and only some of it is labeled.
- Many real-world ML problems fall in this category since it can be expensive and time-consuming to label all data.

Reading Material

Books:

- Machine Learning Yearning by Andrew Ng --- a book in progress
- Machine Learning by Tom Mitchell
- The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani and Jerome Friedman

Online courses/articles:

- Machine Learning – Stanford University (online course) at Coursera.
- Tutorials available on DataCamp.
- Practical Machine Learning Video Series by PythonProgramming.net
- Machine Learning Mastery by Jason Brownlee – online reading material/crash course.