Machine Learning:

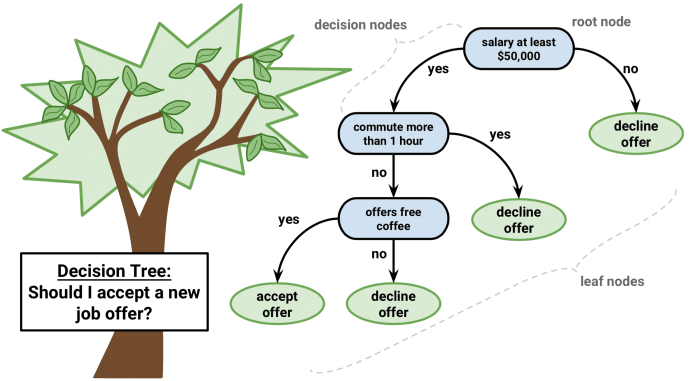
**Superwised Machine Learning:** Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

Supervised learning problems can be further grouped into regression and classification problems.

* **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
* **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

Some popular examples of supervised learning algorithms are:

1. **Linear Regression:** For Regression
2. **Logistic Regression:** For classification
3. Decision Treescovering both **classification and regression**. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making.



In decision trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with record’s attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

We continue comparing our record’s attribute values with other **internal nodes** of the tree until we reach **a leaf node** with predicted class value. As we know how the modeled decision tree can be used to predict the target class or the value. Now let’s understanding how we can create the decision tree model.

It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**.

The primary challenge in the decision tree implementation is to identify which attributes do we need to consider as the root node and each level. Handling this is know the attributes selection. We have different attributes selection measure to identify the attribute which can be considered as the root note at each level.

**The popular attribute selection measures:**

* Information gain
* Gini index

1. **Random forest:** Random forest for classification and regression problems.

The random forest is an **ensemble learning method**, composed of multiple decision trees. By averaging out the impact of several decision trees, random forests tend to improve prediction.

**Ensembled algorithms** are those which combines more than one algorithms of same or different kind for classifying objects. For example, running prediction over Naive Bayes, SVM and Decision Tree and then taking vote for final consideration of class for test object

Random forests tend to shine in scenarios where a model has a large number of features that individually have weak predicative power but much stronger power collectively

1. **Support vector machines** for binary classification problems.

Given a set of points of 2 types in N dimensional place, SVM generates a (N — 1) dimensional hyperplane to separate those points into 2 groups. Say you have some points of 2 types in a paper which are linearly separable. SVM will find a straight line which separates those points into 2 types and situated as far as possible from all those points.

1. **KNN or** [**k-nearest neighbors**](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm) **: KNN** is a **non-parametric, lazy** learning algorithm

Lazy means there is no explicit training phase or it is very minimal

Lack of generalization means that KNN keeps all the training data. To be more exact, all (or most) the training data is needed during the testing phase.

KNN Algorithm is based on **feature similarity**: How closely out-of-sample features resemble our training set determines how we classify a given data point.

An object is **classified** by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. It can also be used for **regression** — output is the value for the object (predicts continuous values). This value is the average (or median) of the values of its k nearest neighbors.

1. **Naive Bayes Classification:** A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. The crux of the classifier is based on the Bayes theorem.

Naive Bayes classifiers have been especially popular for text classification, and are a traditional solution for problems such as **spam detection**.

**Unsuperwised Machine Learning:** Unsupervised learning is where you only have input data (X) and no corresponding output variables.

The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

Some popular examples of unsupervised learning algorithms are:

* k-means for clustering problems.
* Apriori algorithm for association rule learning problems.

**Semi-Supervised Machine Learning**

Problems where you have a large amount of input data (X) and only some of the data is labeled (Y) are called semi-supervised learning problems.

These problems sit in between both supervised and unsupervised learning.

A good example is a photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled.

Many real world machine learning problems fall into this area. This is because it can be expensive or time-consuming to label data as it may require access to domain experts. Whereas unlabeled data is cheap and easy to collect and store.

You can use unsupervised learning techniques to discover and learn the structure in the input variables.

You can also use supervised learning techniques to make best guess predictions for the unlabeled data, feed that data back into the supervised learning algorithm as training data and use the model to make predictions on new unseen data.

**Recommender Machine Learning Algoritms:**

1. KNN( K Nearest neighbors)
2. MF( matrix factorization)
3. AR( Associations Rules)
4. Deep NN

**Keras** is an [open source](https://en.wikipedia.org/wiki/Open-source_software) [neural network](https://en.wikipedia.org/wiki/Artificial_neural_network) library written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)). It is capable of running on top of [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow), [Microsoft Cognitive Toolkit](https://en.wikipedia.org/wiki/Microsoft_Cognitive_Toolkit), or [Theano](https://en.wikipedia.org/wiki/Theano_(software)).[[1]](https://en.wikipedia.org/wiki/Keras#cite_note-1) Designed to enable fast experimentation with [deep neural networks](https://en.wikipedia.org/wiki/Deep_learning), it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System),[[2]](https://en.wikipedia.org/wiki/Keras" \l "cite_note-2) and its primary author and maintainer is François Chollet, a [Google](https://en.wikipedia.org/wiki/Google) engineer.

In addition to standard neural networks, Keras has support for [convolutional](https://en.wikipedia.org/wiki/Convolutional_neural_networks) and [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks)

**TensorFlow** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library](https://en.wikipedia.org/wiki/Library_(computing)) for [dataflow programming](https://en.wikipedia.org/wiki/Dataflow_programming) across a range of tasks. It is a symbolic math library, and is also used for [**machine learning**](https://en.wikipedia.org/wiki/Machine_learning) **applications such as** [**neural networks**](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

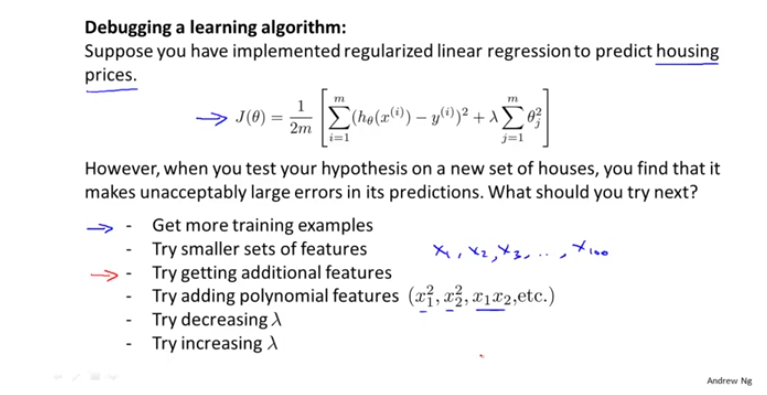
TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

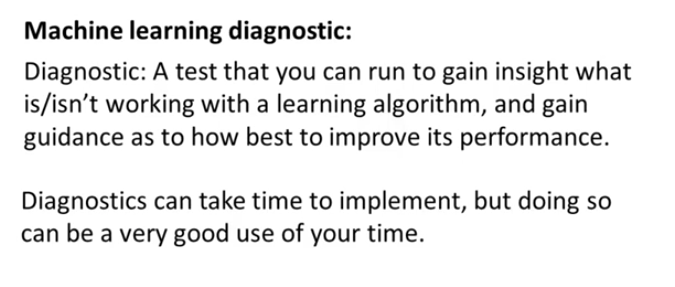
The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as [*tensors*](https://en.wikipedia.org/wiki/Tensor).

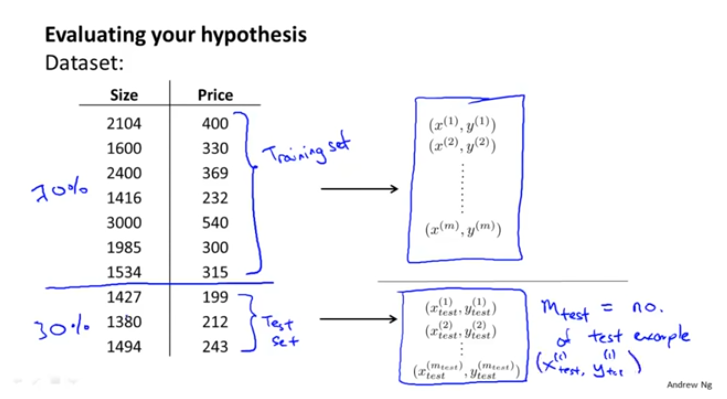
Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, [TPUs](https://en.wikipedia.org/wiki/Tensor_processing_unit)), and from desktops to clusters of servers to mobile and edge devices.

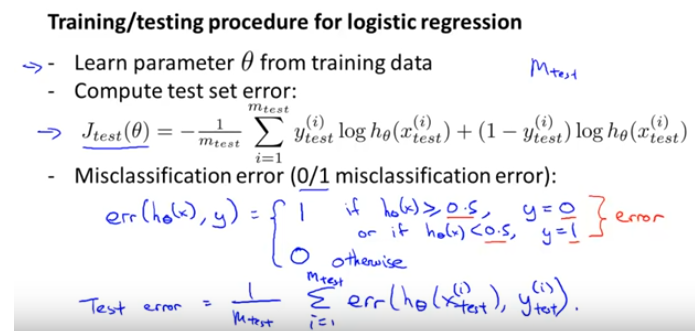
**Microsoft Cognitive Toolkit:** previously known as **CNTK** and sometimes styled as **The Microsoft Cognitive Toolkit**, is a [deep learning](https://en.wikipedia.org/wiki/Deep_learning) [framework](https://en.wikipedia.org/wiki/Software_framework) developed by [Microsoft Research](https://en.wikipedia.org/wiki/Microsoft_Research). Microsoft Cognitive Toolkit describes [neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) as a series of computational steps via a [directed graph](https://en.wikipedia.org/wiki/Directed_graph).

**Theano** is a Python library and optimizing compiler for manipulating and evaluating mathematical expressions, especially matrix-valued ones.[[2]](https://en.wikipedia.org/wiki/Theano_(software)#cite_note-2) In Theano, computations are expressed using a [NumPy](https://en.wikipedia.org/wiki/NumPy)-esque syntax and [compiled](https://en.wikipedia.org/wiki/Compiler) to run efficiently on either CPU or [GPU](https://en.wikipedia.org/wiki/General-purpose_computing_on_graphics_processing_units) architectures.





****



Randomly shuffle the data to select 70% and 30% of the data to avoid any data patterns.

