Machine Learning

First of all, I going to start the presentation with a small introduction about Machine Learning, Machine Learning is a set of techniques that uses algorithms which iteratively learn from data and enable machine to make and improve predictions and behavior.

Here we can see a simplification of the machine learning flow, we start with the

1. Collecting Data. Having variety, density and volume of relevant data.

2. Preparing Data: Fixing issues with the data set.

3. Training Mode: Choose an appropriate algorithm and representation of data in form of the model. Use the training data-set to train the model.

4. Evaluating the model: test the accuracy and precision of the chosen method.

5. And finally, Improving the performance.

Machine learning algorithms discover patterns in big data. These different algorithms can be classified into two categories based on the way they "learn" about data to make predictions. Those are supervised and unsupervised learning.

---

1. In supervised learning scientist acts as a guide to teach the algorithm what conclusions or predictions it should come up with. In unsupervised learning there is no correct answer there is no teacher, algorithms are left to their own to discover and present the interesting hidden structure in the data.

2. Supervised learning model will use the training data to learn a link between the input and the outputs. Unsupervised learning does not use output data. In unsupervised learning, their won't 'be any labeled prior knowledge, whereas in supervised learning will have access to the labels and will have prior knowledge about the datasets

To understand the supervised learning and unsupervised learning better, let's take real-life examples, probably one of the most used examples to explain this.
The first is Supervised Learning. This is what you probably associate with Machine Learning A model is built by providing a dataset with concrete example. (e.g. this is a duck, this is not a duck).
The second approach, Unsupervised Learning, is used to discover structures within given data. The initial data is not necessarily labeled and the learning uses clustering algorithms in order to group unlabeled data together.

6. Supervised learning algorithms: Support vector machine, Linear and logistics regression, Neural network, Classification trees and random forest etc.

8. Classification and regression area widely used algorithms in supervised learning. Support Vector Machines (SVM) are supervised machine learning models with associated learning algorithms, those can be used for both classification and regression purposes, but mostly used for classification problems.

11.Clustering is widely used in unsupervised learning. Clustering is the task of dividing the data points into number of groups such that same traits points will be together in the form of cluster. There are more many clustering algorithms; few of them are Connectivity models, centroid models, Distribution models and Density models.

10. The main goal of regression algorithms is to predict the discrete or continues value. In some cases, the predicted value can be used to identify the linear relationship between the attributes.

5. Supervised learning: The idea is that training can be generalized and that the model can be used on new data with some accuracy.

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting (“curse of dimensionality”) and also reduce computational costs.

The general LDA approach is very similar to a Principal Component Analysis, but in addition to finding the component axes that maximize the variance of our data (PCA), we are additionally interested in the axes that maximize the separation between multiple classes (LDA).

Suppose that we plotted the relationship between two variables where each color represent a different class.

If we’d like to reduce the number of dimensions down to 1, one approach would be to project everything on to the x-axis.

This is bad because it disregards any useful information provided by the second feature. On the other hand, Linear Discriminant Analysis, or LDA, uses the information from both features to create a new axis and projects the data on to the new axis in such a way as to minimizes the variance and maximizes the distance between the means of the two classes.

1. Compute the ***d***-dimensional mean vectors for the different classes from the dataset.
2. Compute the scatter matrices (in-between-class and within-class scatter matrix).

3. Compute the eigenvectors ($\boldsymbol{e_1, e_2, \ldots, e_d}$) and corresponding eigenvalues ($\boldsymbol{\lambda_1, \lambda_2, \ldots, \lambda_d}$) for the scatter matrices.
4. Sort the eigenvectors by decreasing eigenvalues and choose ***k*** eigenvectors with the largest eigenvalues to form a ***d*** x ***k*** dimensional matrix ***W*** (where every column represents an eigenvector).

5. Use this ***d*** x ***k*** eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: ***Y*** =***X*** x ***W*** (where ***X*** is a ***n*** x ***d***-dimensional matrix representing the ***n*** samples, and ***y*** are the transformed ***n*** x ***k***-dimensional samples in the new subspace).

1. Create an axis that maximizes the distance between the means for the two categories whiile minimizing the scatter.

   The axis was chosen to maximize the distance between the two means (between the two categories) while minimizing the “scatter”

More than 2 categories

How you mesure the distances among the means: First, find the point that is central to all of the data, then measure the distances between a point that is central in each category and the main central point. [d]

Maximaze the distance between each category and the central point while minimizing the scatter for each category

Support Vector Machines (SVM)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, the objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

When we have 2 categories, but no obvious linear classifier that separates them in a nice way, support vector machines work by moving the data into a relatively high dimensional space and finding a relatively high dimensional support vector classifier that can effectively classify the observations.

In order to make the mathematics possible, Support vector machine use something called Kernel Function to systematically find support vector Classifier in higher dimensions

The Kernal Trick reduces the amount of computation required for Support Vector Machines by avoiding the math that transforms the data from low to high dimensions and it makes calculating relationships in the infinite dimensions used by the Radial Kernel possible.

The Regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example.

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Where as high gamma means the points close to plausible line are considered in calculation

And finally last but very important characteristic of SVM classifier. A margin is a separation of line to the closest class points. A good margin is one where this separation is larger for both the classes. Images below gives to visual example of good and bad margin. A good margin allows the points to be in their respective classes without crossing to other class.

In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

1. Start with data in a relatively low dimension
2. Move the data into a higher dimension
3. Find a support vector classifier that separates the higher dimensional data into two groups