Classifying data using Support Vector Machines(SVMs)

Introduction to SVMs: In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

What is Support Vector Machine?

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification, implicitly mapping their inputs into high-dimensional feature spaces.

What does SVM do?

Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. Let you have basic understandings from this article before you proceed further. Here I'll discuss an example about SVM classification of cancer UCI datasets using machine learning tools i.e. scikit-learn compatible with Python.

SVM Hyperparameter Tuning using GridSearchCV

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. However, there are some parameters, known as **Hyperparameters** and those cannot be directly learned. They are commonly chosen by humans based on some intuition or hit and trial before the actual training begins. These parameters exhibit their importance by improving the performance of the model such as its complexity or its learning rate. Models can have many hyper-parameters and finding the best combination of parameters can be treated as a search problem.

SVM also has some hyper-parameters (like what C or gamma values to use) and finding optimal hyper-parameter is a very hard task to solve. But it can be found by just trying all combinations and see what parameters work best. The main idea behind it is to create a grid of hyper-parameters and just try all of their combinations (hence, this method is called **Gridsearch**, But don't worry! we don't have to do it manually because Scikit-learn has this functionality built-in with GridSearchCV.

GridSearchCV takes a dictionary that describes the parameters that could be tried on a model to train it. The grid of parameters is defined as a dictionary, where the keys are the parameters and the values are the settings to be tested.

This article demonstrates how to use the **GridSearchCV** searching method to find optimal hyper-parameters and hence improve the accuracy/prediction results

Using SVM to perform classification on a non-linear dataset

Definition of a hyperplane and SVM classifier:

For a linearly separable dataset having n features (thereby needing n dimensions for representation), a hyperplane is basically an (n – 1) dimensional subspace used for separating the dataset into two sets, each set containing data points belonging to a different class. For example, for a dataset having two features X and Y (therefore lying in a 2-dimensional space), the separating hyperplane is a line (a 1-dimensional subspace). Similarly, for a dataset having 3-dimensions, we have a 2-dimensional separating hyperplane, and so on.

In machine learning, Support Vector Machine (SVM) is a non-probabilistic, linear, binary classifier used for classifying data by learning a hyperplane separating the data.

Classifying a non-linearly separable dataset using a SVM – a linear classifier:

As mentioned above SVM is a linear classifier which learns an (n – 1)-dimensional classifier for classification of data into two classes. However, it can be used for classifying a non-linear dataset. This can be done by projecting the dataset into a higher dimension in which it is linearly separable!

A brief introduction to kernels in machine learning:

In machine learning, a trick known as "kernel trick" is used to learn a linear classifier to classify a non-linear dataset. It transforms the linearly inseparable data into a linearly separable one by projecting it into a higher dimension. A kernel function is applied on each data instance to map the original non-linear data points into some higher dimensional space in which they become linearly separable.