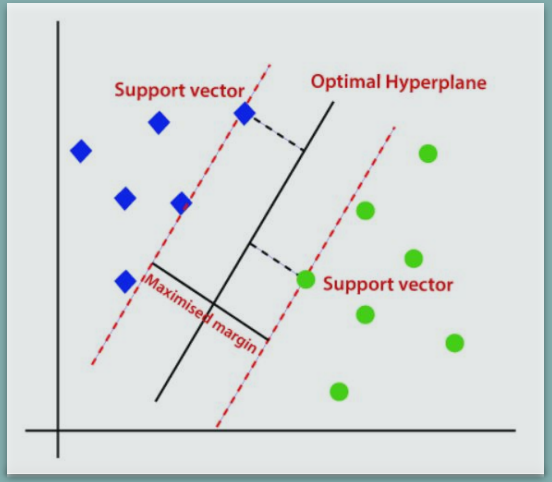
SVM

Support Vector Machines

By Adam Brewer, Robert Stewart, Tarick Mehanna

How the SVM works

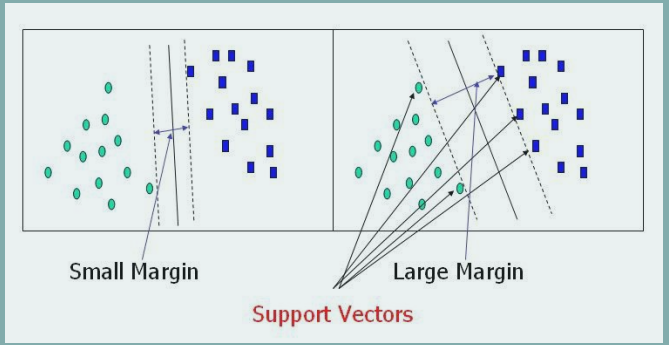


- A classification method.
- Draws a **hyperplane** that separates the points.
    - Hyperplane? It's a 2-D line or 3-D plane that is drawn to try and guarantee all points of one category are on one side of the plane/line and the other points are on the other side.
- Algorithm attempts to **maximize the margin**
    - Margin? The distance from the points in each classified group. We attempt to get as large of a distance from the hyperplane toe each group of points.
- **Kernels!**

Advantages and disadvantages of SVM

Advantages:

- Ideal when we have no idea about the data - regularization
- Handles non-linear data efficiently **(Kernel trick)**
- Stable models
- Works very well with **high dimensional data**



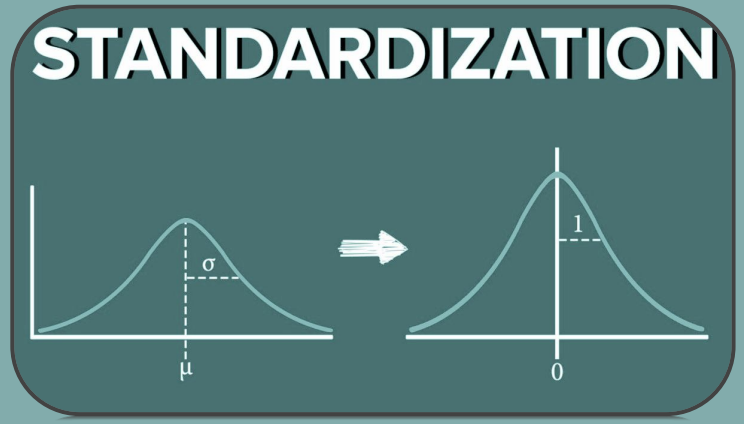
Disadvantages:

- Choosing an **appropriate Kernel function** is relatively difficult
- Performance suffers when the data set has more noise, or overlap among target variables
- Long training time for larger datasets
- Difficult to understand and interpret the **final model, variable weights, and individual impact**

Data processing steps SVM require

- Standard data cleaning procedures
  - Find and handle NaNs and outliers
  - Using feature engineering to look for proper data.

- Be sure to **standardize data!**
  - Working with distances.
  - Things like **StandardScaler()** from sklearn



SVM vs Logistic Regression

While both are classification algorithms, **logistic regression (LR)** models are used to predict the odds in favour of a particular event. SVM models are used to draw hyperplanes that separate data points with the largest margin (distance).

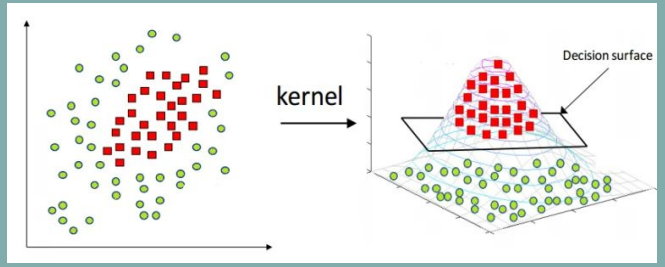
- LR chooses a decision boundary based on a more statistical approach, SVM chooses the decision boundary that maximizes the margin
- SVM also works with unstructured or semi structured data (texts / images)
- LR is more vulnerable to overfitting and outliers than SVM

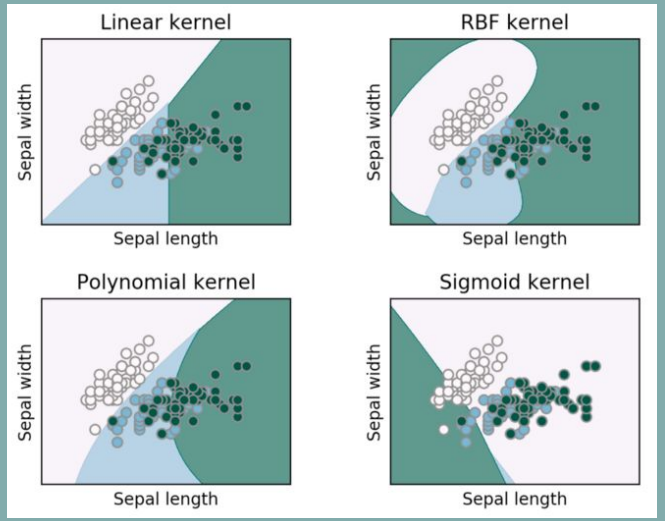
SVM is the only linear model that can **classify data that is not linearly separable!**

SVM Hyperparameters

Kernel | (Types - linear, poly, rbf, sigmoid, precomputed)

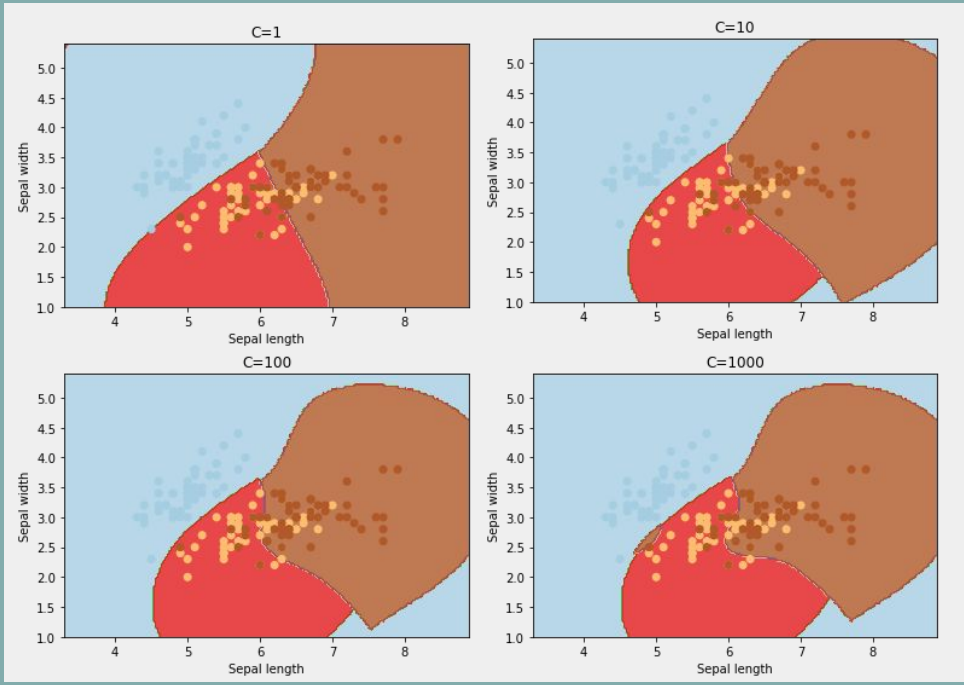
The main function of the kernel is to transform the given dataset input data into the required form. There are various types of functions such as **linear, polynomial, and radial basis function (RBF)**. Polynomial and RBF are useful for non-linear hyperplane. Polynomial and RBF kernels compute the separation line in the higher dimension. This transformation can lead to more accurate classifiers. Some less used kernels are **sigmoid** kernels which act similarly to logistic regression, and **precomputed** kernels where the kernel is an nxn matrix of scalars known as a Kernel matrix or a Gram matrix.





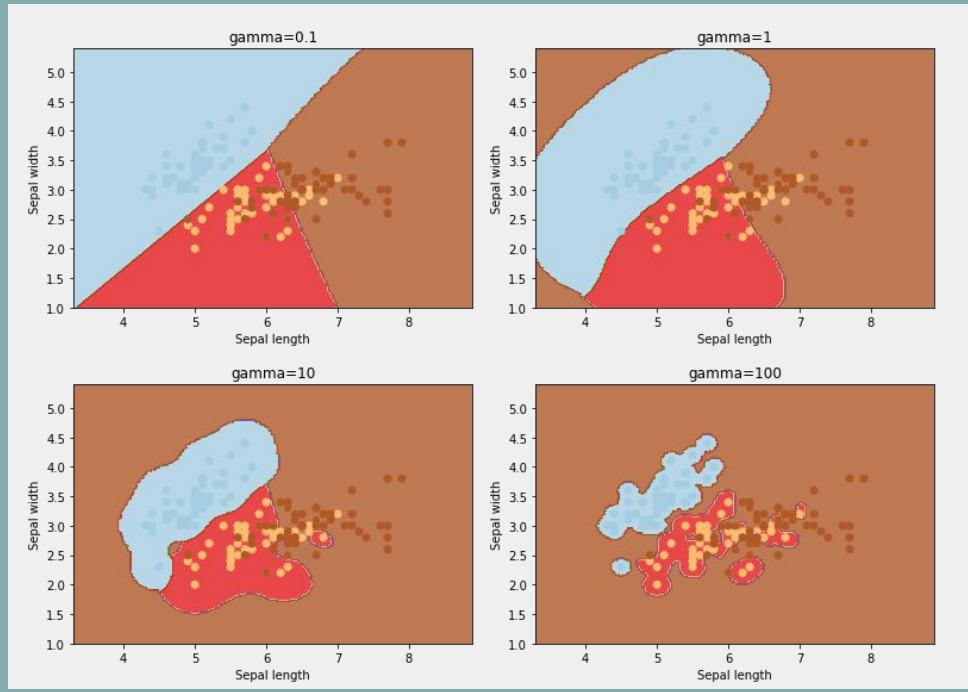
Regularization (C)

Regularization parameter in python's Scikit-learn **C parameter** used to maintain regularization. Here C is the **penalty parameter**, which represents misclassification or error term. The misclassification or error term tells the SVM optimization how much error is bearable. This is how you can control the trade-off between decision boundary and misclassification term. A **smaller value of C** creates a **small-margin hyperplane** and a **larger value of C** creates a **larger-margin hyperplane**. The C parameter default is 1.0 but it can be changed to any positive value, however, **increasing C values can lead to overfitting**.



Gamma

Gamma hyperparameter only available for kernels **polynomial, RBF, and sigmoid. Gamma helps classify how strong of a fit we use.** Gamma's default is 'scale' equal to 1 / (n_features * X.var()) as value of gamma. Gamma can also be set to 'auto' equal to 1 / n_features, or a positive float value. A **lower value of Gamma** will **loosely fit** the training dataset, whereas a **higher value of gamma** will **exactly fit** the training dataset, which causes overfitting. In other words, you can say a low value of gamma considers only nearby points in calculating the separation line, while the a value of gamma considers all the data points in the calculation of the separation line. **Increasing gamma leads to overfitting** as the classifier tries to perfectly fit the training data.



Appendix

Code Demos & Technical Breakdowns of SVM:

Advantages and Disadvantages of SVM: [Very simple bullet point breakdown]
https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/

Link with Demo and Lower level Explanation:
https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python

Coding Demo showing how all hyperparameters are used:
https://www.kaggle.com/code/maxl11/svm-demo/notebook

Link with High level Explanation and LR Comparison: [potentially too light on useful info?]
https://medium.com/axum-labs/logistic-regression-vs-support-vector-machines-svm-c335610a3d16

Code along using sklearn.svm.SVC with visuals:
https://www.analyticsvidhya.com/blog/2021/04/insight-into-svm-support-vector-machine-along-with-code/

Advantages of SVM: [Has some mathematical details]
https://iq.opengenus.org/advantages-of-svm/

Low Level Explanation, C and Gamma Parameters:
https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167

Hyperparameter Tuning Code demo with GridSearchCV:
https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/

Link with Implementation and Higher level Explanation:
https://scikit-learn.org/stable/modules/svm.html

Low Level Explanation, Includes Some Data Cleaning Details:
https://towardsdatascience.com/svm-implementation-from-scratch-python-2db2fc52e5c2

Hypertuning parameters code along:
https://medium.com/all-things-ai/in-depth-parameter-tuning-for-svc-758215394769

Appendix

Video Tutorials:

General technical and abstract breakdown on SVM: (2 min)
https://www.youtube.com/watch?v=_YPScrckx28

Mathematical/technical Breakdown with fantastic visuals: (15 min)
https://www.youtube.com/watch?v=ny1iZ5A8ilA

Hyper parameter kernel and gamma breakdown on SVM: (3 min)
https://www.youtube.com/watch?v=_YPScrckx28

SVM technical breakdown & code along demo: (23 min)
https://www.youtube.com/watch?v=FB5EdxAGxQg