- ## Support Vector Machine

Support vector machines are simply classifiers that are used to draw a line that best separates different classes of data. Larger dimensions “hyperplane” and the objective is to find a hyperplane which helps us to increase the margin between the classes.

The use of kernel functions allows the user to apply a classifier to data that have no obvious fixed-dimensional vector space representation.

It works on the principle of the margin from the support vectors which are called as ‘hyperplane’.

- ## Support Vector

Support vectors are the data points that lie closest to the decision surface (or hyperplane) .

- ## Difference between One-vs-One and One-vs-All method

The difference is the number of classifiers you have to learn, which strongly correlates with the decision boundary they create.

One vs all strategy involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. This strategy requires the base classifiers to produce a real-valued confidence score for its decision, rather than just a class label; discrete class labels alone can lead to ambiguities, where multiple classes are predicted for a single sample.

In pseudocode, the training algorithm for an OvA learner constructed from a binary classification learner L is as follows:

Inputs:

- L, a learner (training algorithm for binary classifiers)
- samples X
- labels y where $y_i \in \{1, \ldots K\}$ is the label for the sample X_i

Output:

- a list of classifiers f_k for $k \in \{1, \ldots, K\}$

Procedure:

- For each k in {1, …, K}
  - Construct a new label vector z where $z_i = 1$ if $y_i = k$ and $z_i = 0$ otherwise
  - Apply L to X, z to obtain f_k

  Making decisions means applying all classifiers to an unseen sample x and predicting the label k for which the corresponding classifier reports the highest confidence score:

n the one-vs.-one (OvO) reduction, one trains K (K − 1) / 2 binary classifiers for a K-way multiclass problem; each receives the samples of a pair of classes from the original training set, and must learn to distinguish these two classes. At prediction time, a voting scheme is applied: all K (K − 1) / 2 classifiers are applied to an unseen sample and the class that got the highest number of "+1" predictions gets predicted by the combined classifier.

- Dependencies
  https://pypi.org/project/mlxtend/- mlxtend package
  https://pypi.org/project/graphviz/- Graphviz

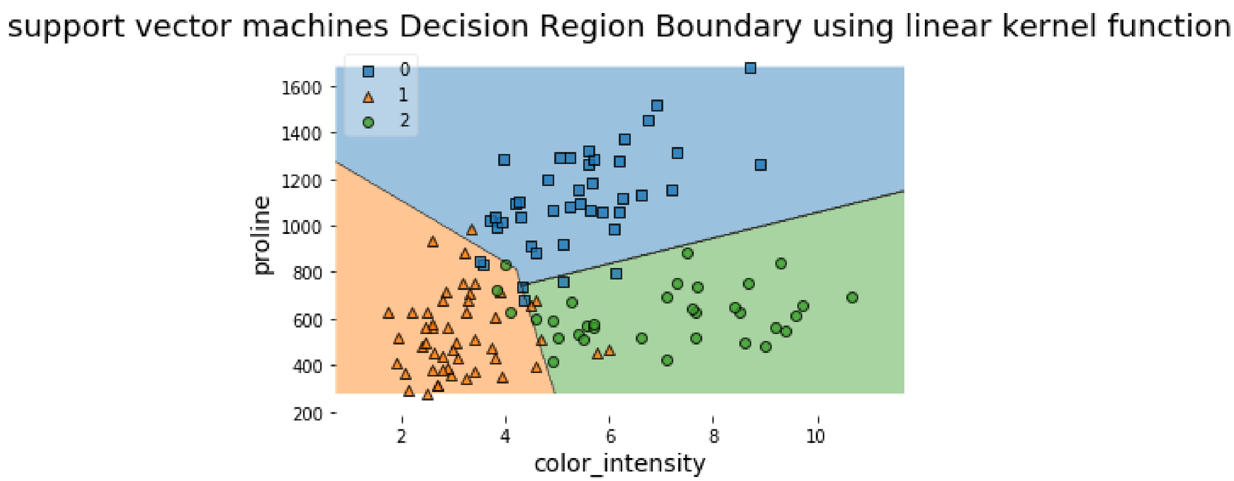
- Criteria for selecting the two attributes.

For selection of the attributes we are implanting chi square test. chi-squared test assumes that a null hypothesis and an another hypothesis. If its less than 0.05 then we reject the hypothesis as null hypothesis

- SVM- Linear :
  They are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. 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 (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). 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. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

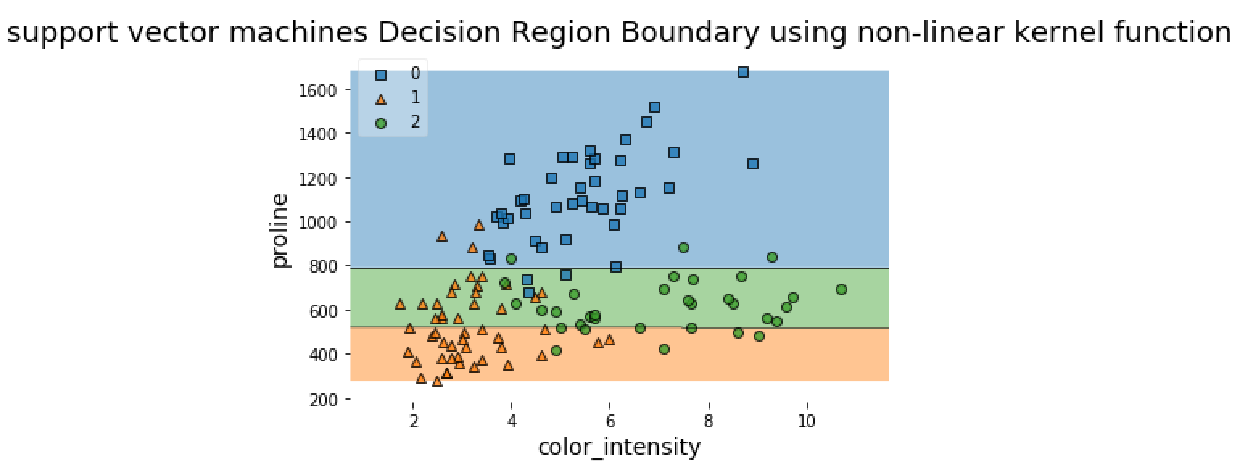
2D projection



- SVM Non-Linear(Round Bassein Function)

Perform binary classification using non-linear SVC with RBF kernel. The target to predict is a XOR of the inputs.

2DProjection

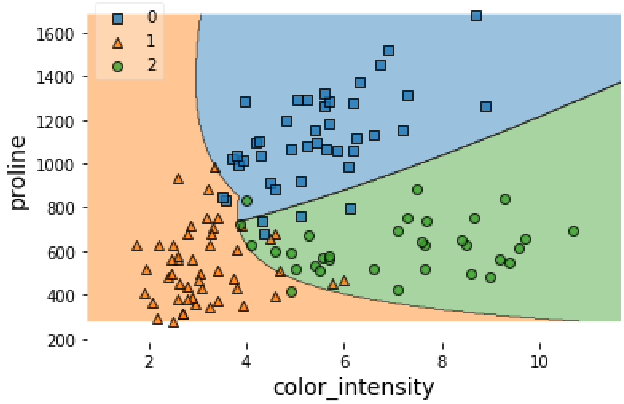


- SVM- Polynomial (Degree-3)

The **polynomial kernel** is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

2D- Projection

support vector machines Decision Region Boundary using polynomial kernel function



- **Observation:**

Linear support vector machine gives higher precision, fscore and recall values than the non linear RBF and polynomial.

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

In-General SVMs are great for:

- Small to medium data sets perform only.
- When the data set has more noise i.e. overlapping more of data, SVM is very bad in terms of handling it.
- It is quite memory efficient and also extremely fast prediction.

An important point to note is that the SVM doesn't directly provide probability estimates, these are calculated using an expensive cross-validation in sklearn implementation.

- **References**

http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
https://rasbt.github.io/mlxtend/
http://scikit-learn.org/stable/modules/svm.html
https://en.wikipedia.org/wiki/Support_vector_machine
https://en.wikipedia.org/wiki/Polynomial_kernel