**Centre for ARTIFICIAL INTELLLIGENCE  
TKM COLLEGE OF ENGINEERING, Kollam**

**Semester II**

**Advanced Computing Lab**

**TUTORIAL QUESTION: SET 2 (25.05.21)**

**Instructions**

1. Visit <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
2. Prepare & add solutions next to questions.
3. Save your file name like: ebin\_lab\_tutorial\_2.docx & attach back in ETlab.
4. Be ready to attend Viva (SVM & Following questions) in the next lab hour.
5. Study and present (your own points) relevance of various parameters available in sklearn - SVC function.

Solution:

1. C (float, default = 1.0): – Regularization parameter:

* It is used to control error or margin.
* Strength of regularization is inversely proportional to ‘C’.
* It is a hyperparameter (set before training model) that determines the penalty for misclassifying an observation i.e., how much we want to avoid misclassifying each training example.
* Smaller values of ‘C’ will cause optimizer to look for large margin, reduces error.
* Larger values of ‘C’ look for small margin and error is increased.

1. Kernel (‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ’precomputed’, default = ‘rbf’): -

* It is another tuning parameter.
* It is a mathematical function which takes data as input and transfer it into required form.

1. Degree (int, default = 3): -

* It specifies the degree of polynomial kernel function.
* This is ignored by all other polynomial functions.
* Degree of polynomial means the features or attributes of the data.

1. Gamma (float, default = ‘scale’): -

* It can be ‘scale’ or ‘auto’.
* It represents the kernel coefficient.
* It is used for Gaussian rbf, polynomial and sigmoid kernel.
* For linear kernels ‘C’ hyperparameter is only required.
* Gamma is another hyperparameter, set before training model and decides how much curvature weight we need in a decision boundary.
* High gamma means more curvature and low gamma means less curvature.
* Tuning high and low gamma totally depends on the data.

1. Coef0 (float, default = 0.0): -

* It is used to adjust the independent term in the kernel function.
* It is more significant for polynomial and sigmoid functions.

1. Shrinking (bool, default = True): -

* Shrinking are the parameter estimates to be closer to each other and heuristics are the rules for decision making.
* Shrinking heuristics speeds up the classifier training.
* It tries to identify and remove the bounded elements.

1. Probability (bool, default = False): -

* It is used to calculate probability scores and is like a degree of confidence.
* Predict\_proba () is a function which is used to get possibility for the classes of the data.
* But if the probability is set to false it indicates error.
* It is enabled before calling fit.

1. tol (float, default = 1e-3 i.e., 10^-3): -

* It is a setting for SVM’s tolerance in optimization.
* For an SVM to be valid, all values must be greater than or equal to 0, and at least one value on each side needs to be equal to 0, which will be the support vectors.
* Since it is highly unlikely to get values equal to 0, we set tolerance to allow a bit of change.

1. cache\_size (float, default = 200): -

* It indicates the size of kernel cache in MB.
* The scikit-learn SVM module wraps 2 powerful libraries written in C, libsvm and liblinear.
* When fitting a model, there is a flow of data between Python and 2 external libraries.
* A cache smooths the data exchange operations.
* If cache is too small and we have too many data it becomes too risky.
* If we have good memory, it is good to set a cache size greater than 200MB.

1. Class\_weight (dict or ‘balanced’, default = None): -

* Balanced mode automatically adjusts weights inversely proportional to class frequencies in the input data.
* It given none then all classes are supposed to have weight 1.

1. Verbose (bool, default = false): -

* It enables verbose output for producing detailed logging information.
* It is similar to asking the program about what it is doing all the time.

1. max\_iter (default = -1): -

* It sets the maximum number of iterations for the quadratic programming to cycle and optimize.
* Default value indicates there is no limit.

1. decision\_function\_shape (‘ovr’ and ‘ovo’, default = ‘ovr’): -

* Decides whether to return one-v/s-one (‘ovo’) or one-v/s-rest (‘ovr’).
* This is not required for binary classification.
* SVM is a binary classifier. SVM optimization is really tasked to separate one group from another, so for classifying 3 or more groups we can use the decision function.
* The ’ovr’ separates each group from rest and is slightly unbalanced and ‘ovo’ requires specification of boundary that separates one group from all other groups.

1. break\_ties (bool, default = False): -

* If it is false then all input in the specified area would be classified as one class, but if it is true the tie breaking mechanism will create a non-convex decision boundary in that area.
* It is not enabled if it is set as false by default.

1. random\_state (int, Random state instance or none, default = none): -

* Control’s pseudo random number generation for shuffling data for probability estimates.
* If we don’t specify the random\_state in the code, then every time we execute the code a new random value is generated and the train and test datasets would have different values each time.
* If a fixed value is assigned like 0 or 1 or 42 or any other integer then no matter how many times, we execute the code, the result would be the same i.e., same inputs in train and test datasets.

1. Study and prepare a NOTE (Intro/Definition -2/3 lines, Over-coming any problems? –1/2 lines, Advantages, Disadvantages, Applications, Equation if any) separately on LINEAR, POLY, RBF, SIGMOID kernels.

Solution:

* Kernel functions are set of mathematical functions that takes data as input and transform it into required form.
* It generally transforms the training set of data so that a non-linear decision surface is transformed to linear equation in higher number of dimension spaces.
* Different kernel functions used by different SVM algorithms include linear, polynomial, gaussian, sigmoid, RBF etc.

1. Linear Kernel: -

* It is the simplest kernel function.
* It is given by the inner product of (x, y) plus an optional constant.
* Kernel algorithms using a linear kernel are often equivalent to their non-kernel counterparts i.e., Kernel PCA with linear kernel is same as standard PCA.
* It is given by the equation:

k (x, y) = + c

Advantages: - Faster than other kernel functions. For optimization of linear kernels only ‘C’ hyperparameter is required, but for other kernels both ‘C’ and gamma regularization parameter is required.

1. Polynomial Kernel: -

* It is a non-stationary kernel.
* It is well suited for problems where all the training data is normalized.
* It is given by the equation:

k (x, y) =

* Adjustable parameters are the slope , the constant term c and the polynomial degree d. x and y are vectors in input space.
* It is used in image processing.

1. RBF Kernel: -

* Radial Basis Function kernel or RBF kernel is a popular kernel function used in various kernelized learning algorithms.
* It is commonly used in support vector machine classification.
* The RBF kernel on two samples x and x’ represented as feature vectors in some input space is defined as: - here ‘σ’ is the variance and our hyperparameter  
  and *||*x – x’|| is the Euclidean distance between 2 points x and x’.

k (x, x’) =

1. Sigmoid Kernel: -

* It is applied in neural networks for classification tasks.
* Bipolar sigmoid function is used as an activation function for artificial neurons.
* The 2 adjustable parameters in sigmoid kernels are slope and intercept constant c.

Advantages: -SVM classifier which is applied with sigmoid kernel, has excellent classification accuracy.

Disadvantages: -A sigmoid kernel has complicated structure. It is generally difficult for human experts to interpret and understand how the sigmoid kernel makes its classification decision.

* The equation is given by:

k (x, x’) = tanh (

---Happy Learning--