

CS 312: Artificial Intelligence Laboratory

Task 6: Machine Learning

Support Vector Machine Classifier

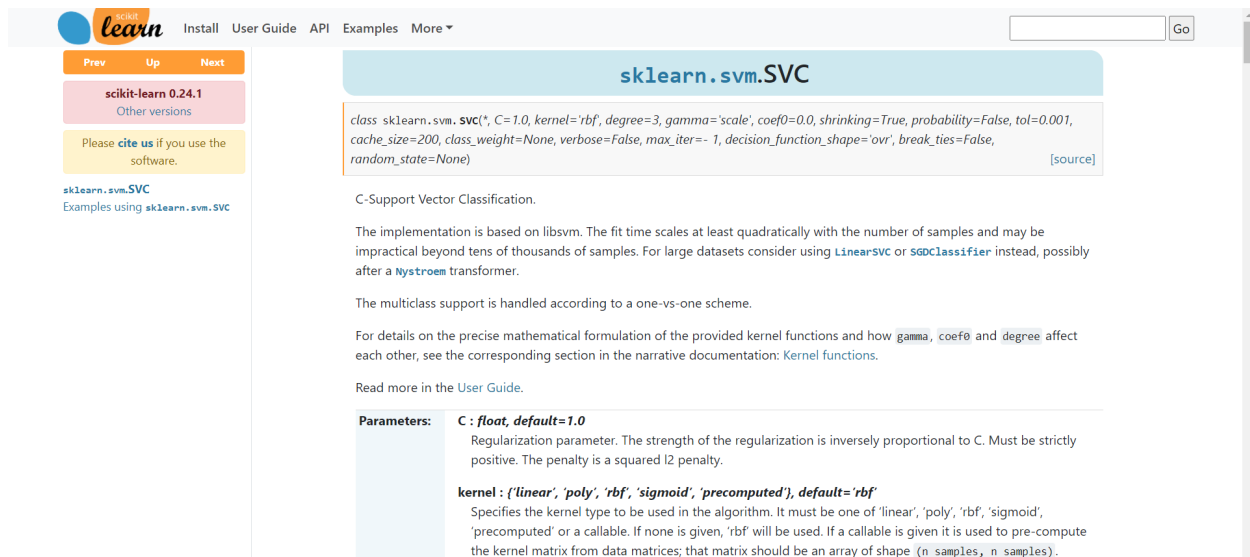
Group 14

Team Members

- 180010029 [Rupesh Kalantre]
- 180010031 [Shagun Bera]

1. Libraries Used

We have used the SVC from scikit learn library



The screenshot shows the scikit-learn documentation page for the `sklearn.svm.SVC` class. The page includes a sidebar with navigation links (Prev, Up, Next) and a search bar. The main content area displays the class name `sklearn.svm.SVC` in a blue header. Below the header, the class signature is shown: `class sklearn.svm.SVC(*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None)`. The page also includes a description of the class as "C-Support Vector Classification", a note about the implementation being based on libsvm, and a list of parameters with their default values and descriptions. The parameters listed are `C` (float, default=1.0), `kernel` (string, default='rbf'), and `degree` (integer, default=3).

sklearn.svm.SVC

```
class sklearn.svm.SVC(*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None)
```

C-Support Vector Classification.

The implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. For large datasets consider using `LinearSVC` or `SGDClassifier` instead, possibly after a `Nystroem` transformer.

The multiclass support is handled according to a one-vs-one scheme.

For details on the precise mathematical formulation of the provided kernel functions and how `gamma`, `coef0` and `degree` affect each other, see the corresponding section in the narrative documentation: Kernel functions.

Read more in the [User Guide](#).

Parameters:

- C :** float, default=1.0
Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.
- kernel :** {'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'}, default='rbf'
Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape (n_samples, n_samples).

2. Details of SVM package used

<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

- Pandas library
 - To take input from file and data manipulation
- StandardScaler
 - For normalization of data and scaling
 - Imported from sklearn.preprocessing
- Train_test_split
 - Splits the data in train and test sets in given proportion
 - Imported from sklearn.model_selection
- SVC (support vector model)
 - Support vector model from sklearn
- .fit() method
 - Fits the model on input data (x_train and y_train)
- .predict() method
 - Predict the output for input data (x_test and y_test)

3. Accuracies for different values of 'C' and different kernels are given in **accuracies.pdf**

Observations :

- RBF kernel performs better than linear and quadratic kernels.
- Linear kernel takes most time for training among quadratic and RBG kernels.
- As we increase the value of 'C', accuracy increases gradually. And after a certain value of 'C', accuracy doesn't change much.