University of Tehran

Faculty of New Sciences and Technologies Department

of Mechatronics

Practice of Artificial Neural Networks

By:

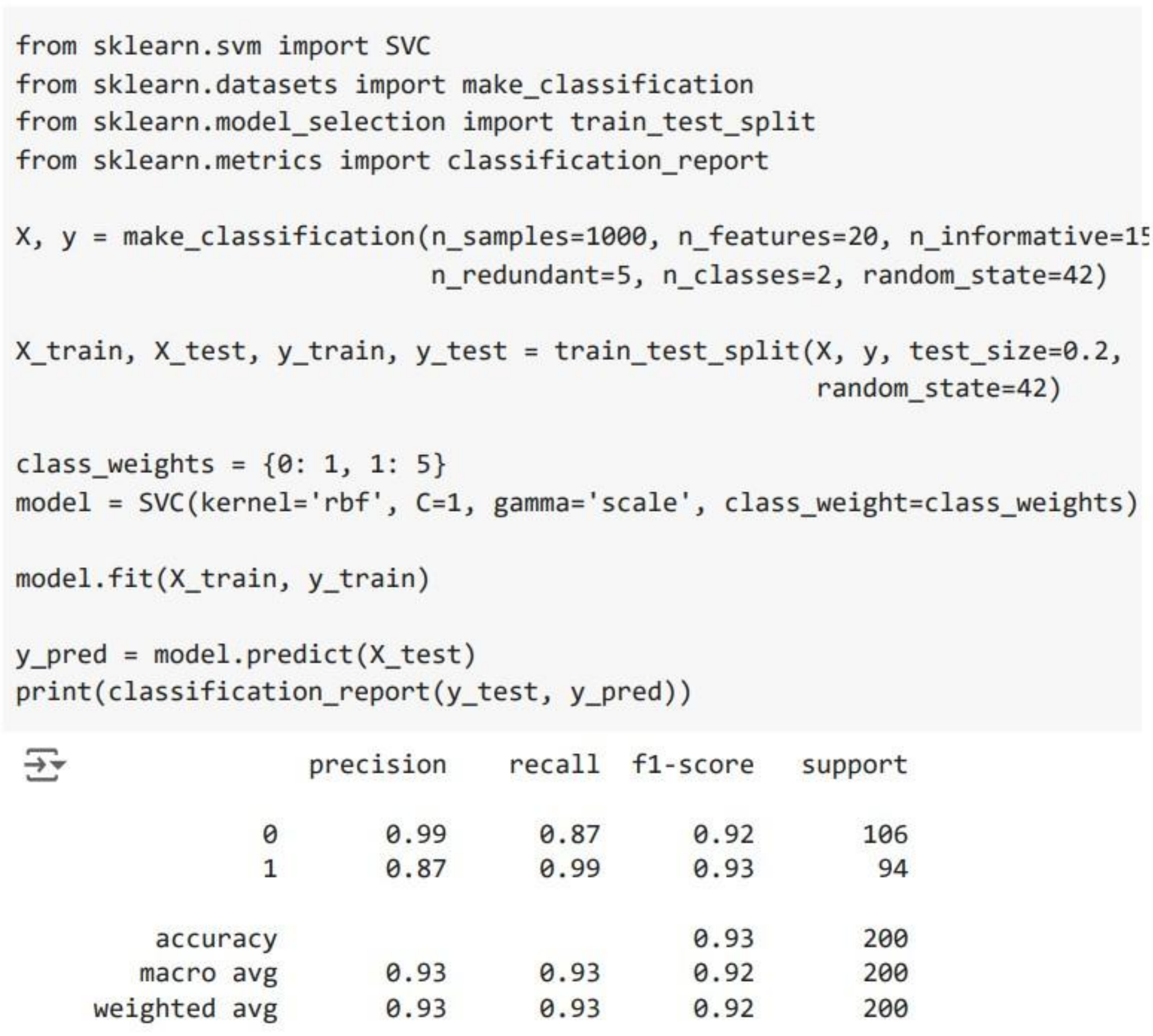
Omid Moradi

Lecturer:

Dr. Masoudnia

Fall 2024

Q1. Suppose we have a nonlinear binary classification problem that we want to solve using a nonlinear Support Vector Machine (SVM). However, there are different risks associated with misclassification for each class, meaning that false positives (FP) and false negatives (FN) have different costs. Modify the formulation of the nonlinear SVM to address this issue. How can it be implemented in Python? (Provide the code and the corresponding hyperparameters).



```
from sklearn.svm import SVC
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

X, y = make_classification(n_samples=1000, n_features=20, n_informative=1
                           n_redundant=5, n_classes=2, random_state=42)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42)

class_weights = {0: 1, 1: 5}
model = SVC(kernel='rbf', C=1, gamma='scale', class_weight=class_weights)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.87      0.92       106
           1       0.87      0.99      0.93        94

    accuracy                           0.93       200
   macro avg       0.93      0.93      0.92       200
weighted avg       0.93      0.93      0.92       200
```

Hyperparameters

1. **kernel='rbf'**: The radial basis function (RBF) kernel is commonly used for nonlinear problems.

2. **C=1**: The regularization parameter, controlling the trade-off between maximizing the margin and minimizing the classification error.
3. **gamma='scale'**: The kernel coefficient; 'scale' is a good starting point.
4. **class_weight**: Dictionary specifying the weights for each class. Adjust values to reflect the relative costs of false positives and false negatives.

Q2. Support vector-based method can also be applied to regression problems. Derive the formulation for Support Vector Regression (SVR).

Support Vector Regression (SVR) is a machine learning algorithm used for regression analysis. SVR Model in Machine Learning aims to find a function that approximates the relationship between the input variables and a continuous target variable while minimizing the prediction error.

Unlike Support Vector Machines (SVMs) used for classification tasks, SVR Model seeks a hyperplane that best fits the data points in a continuous space. This is achieved by mapping the input variables to a highdimensional feature space and finding the hyperplane that maximizes the margin (distance) between the hyperplane and the closest data points, while also minimizing the prediction error.

SVR Model can handle non-linear relationships between the input and target variables by using a kernel function to map the data to a higher-dimensional space. This makes it a powerful tool for regression tasks where complex relationships may exist.

Support Vector Regression (SVR) uses the same principle as SVM but for regression problems.

Formulation of SVR:

Given a training dataset$\{(x_I, y_i)\}^{n}_{\{i=1\}}$, where $x_i \in R^d$ represents the input features and $y_i \in R$ denotes the target values, SVR seeks a function $f(x) = w^T\varphi(x) + b$ that deviates from the actual target y_i by a value no greater than ε for each training point x_i, while maintaining the function's flatness.

Objective:

Minimize the norm of the weight vector w to ensure flatness of f(x):

$$\frac{1}{2}\ ||w||^2$$

Constraints:

Introduce slack variables ξ_i and ξ_i^* to handle deviations larger than ε: y_i

$$- (w^T\varphi(x_i) + b) \leq \varepsilon + \xi_i$$

$$(w^T\varphi(x_i) + b) - y_i \le \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \ge 0$$

Optimization Problem:

Combine the objective and constraints into a single optimization problem:

$$\min \left(\frac{1}{2} \quad ||w||^2 + C \sum^n \quad (\xi_i + \xi_i^*)\right)$$
$$\{i=1\} \{w,b,\xi,\xi^*\} \quad .$$

subject to:

$$y_i - (w^T\varphi(x_i) + b) \le \varepsilon + \xi_i$$
$$(w^T\varphi(x_i) + b) - y_i \le \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \ge 0$$

Here, C is a regularization parameter that determines the trade-off between the model's complexity (flatness) and the amount up to which deviations larger than ε are tolerated.

Kernel Trick:

To handle nonlinear relationships, SVR employs the kernel trick, mapping input features into a higherdimensional space where a linear regression can be performed. This is achieved by replacing the dot product $w^T\varphi(x)$ with a kernel function $K(x_I, x_j)$, allowing the algorithm to operate in the original input space without explicitly computing the mapping $\varphi(x)$.

Dual Formulation:

The primal problem can be transformed into its dual form, which is more computationally efficient, especially when dealing with kernels. The dual formulation involves Lagrange multipliers and leads to the following optimization problem:

$$\max_{\{\alpha,\alpha^*\}} . -\frac{1}{2}\sum_{\{i=1\}}^{n}\sum_{\{j=1\}}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K_{(x_i,\, x_j)} - \varepsilon \sum_{\{i=1\}}^{n}(\alpha_i + \alpha_i^*) + \sum_{\{i=1\}}^{n} y_i(\alpha_i - \alpha_i^*)$$

subject to:

$$\sum_{\{i=1\}}^{n}(\alpha_i - \alpha_i^*) = 0$$
$$0 \le \alpha_i, \alpha_i^* \le C$$

Prediction Function:

Once the optimization is solved, the prediction for a new input x is given by:

$$f(x) = \sum_{\{i=1\}}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + b$$

This formulation allows SVR to perform regression tasks effectively, even in high-dimensional feature spaces, by leveraging the kernel trick and maintaining a balance between model complexity and prediction accuracy.