



# Implementing SVM and Kernel SVM with Python's Scikit-Learn

Last Updated : 20 Jan, 2023

---

In this article we will implement a classification model using Scikit learn implementation for SVM model in [Python](#). Then we will try to understand what is a kernel and how it can help us to achieve better performance by learning non-linear boundaries in the dataset.

## What is a SVM algorithm?

[Support vector machines](#) (SVMs) are a type of supervised learning algorithm that can be used for classification or regression tasks. In simple terms, an SVM constructs a hyperplane or set of hyperplanes in a high-dimensional space, which can be used to separate different classes or to predict continuous variables.

The main idea behind SVMs is to find the hyperplane in the high-dimensional space that has the maximum margin, or the maximum distance between the data points of the different classes. This hyperplane is called the maximum-margin hyperplane, and the data points that are closest to the hyperplane are called support vectors. The position and orientation of the maximum-margin hyperplane can be determined using mathematical optimization techniques.

SVMs have been widely used in many applications, including image and text classification, protein classification, and regression problems. They have several advantages, such as the ability to handle high-dimensional data, the ability to perform non-linear classification using the kernel trick, and the ability to provide probability estimates for classification tasks. However, they can be computationally expensive and may not be suitable for very large datasets.

```
from sklearn import datasets

# Load the Iris dataset
iris = datasets.load_iris()
# We only take the first two
# features for simplicity
X = iris.data[:, :2]
y = iris.target
```

Now we will use SupportVectorClassifier as currently we are dealing with a classification problem.

---

## Python3

```
# Fit the SVM model
model = svm.SVC(kernel='linear')
model.fit(X, y)

# Predict using the SVM model
predictions = model.predict(X)

# Evaluate the predictions
accuracy = model.score(X, y)
print("Accuracy of SVM:", accuracy)
```

### Output :

Accuracy of SVM: 0.82

The above code is an example of using a support vector machine (SVM) model to make predictions on the Iris dataset. The Iris dataset is a well-known dataset in machine learning that contains measurements of various characteristics of

The code first imports the necessary modules and libraries, including the SVM module from Scikit-learn and the Iris dataset from Scikit-learn's datasets module. Then, it loads the Iris dataset and extracts the first two features from each example (sepal length and width), as well as the target labels (the species of the flower).

Next, the code creates an SVM model using the SVC class from Scikit-learn, and specifies that it should use a linear kernel. Then, it trains the model on the data using the `fit()` method, and makes predictions on the same data using the `predict()` method. Finally, it evaluates the predictions by computing the accuracy of the model (the fraction of examples that were predicted correctly) using the `score()` method. The accuracy is then printed to the console.

## What is Kernel SVM?

[Kernel support vector machines](#) (SVMs) are a variant of support vector machines (SVMs) that use kernel functions to find the maximum-margin hyperplane in non-linear classification or regression problems. In simple terms, a kernel function transforms the original data into a higher-dimensional space, where it becomes linearly separable. The maximum-margin hyperplane is then found in this higher-dimensional space using an SVM.

Kernel SVMs have several advantages over regular SVMs. They can handle non-linear classification or regression tasks without having to explicitly perform the data transformation, which can be computationally expensive. They also allow the use of different kernel functions, which can provide different types of non-linear transformations and can be chosen based on the specific characteristics of the data.

The most commonly used kernel functions in kernel SVMs are the linear, polynomial, and radial basis function (RBF) kernels. The linear kernel is used for linear classification or regression tasks, the polynomial kernel can handle non-linear problems, and the [RBF](#) kernel is often used in classification tasks with a large number of features.

datasets. Additionally, choosing the right kernel function and the corresponding hyperparameters can be challenging and requires some domain knowledge and experimentation.

Here is an example of how to implement Support Vector Machines (SVM) and Kernel SVM with Python's Scikit-learn library:

---

## Python3

```
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler

# Dummy function to simulate loading the data from a file

def load_data():
    X_train = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
    X_test = [[10, 11, 12], [13, 14, 15], [16, 17, 18]]
    y_train = [1, 0, 1]
    y_test = [0, 1, 0]
    return X_train, X_test, y_train, y_test

# Load the data and split it into training and test sets
X_train, X_test, \
    y_train, y_test = load_data()
```

Now let's normalize the dataset using the StandardScaler function from the sklearn library this will helps us achieve stable and faster training process of the model.

---

## Python3

```
# Import the necessary modules and libraries
# Scale the features using standardization
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

# Python3

```
# Create a kernel support vector machine model
ksvm = svm.SVC(kernel='rbf',
                gamma=0.1,
                C=10.0)

# Train the model on the training data
ksvm.fit(X_train, y_train)

# Evaluate the model on the test data
accuracy = ksvm.score(X_test, y_test)
print('Accuracy:', accuracy)
```

## Output :

Accuracy: 0.3333333333333333

## Difference between the SVM and kernel SVM:

Support vector machines (SVMs)	kernel support vector machines (KSVMs)
Support vector machines (SVMs) is supervised learning algorithm that can be used for classification and regression.	kernel support vector machines (KSVMs) are also supervised learning algorithms that can be used for classification and regression.
SVMs are more sensitive to the choice of hyperparameters (such as the regularization parameter and kernel function).	KSVMs are less sensitive and can often be trained using default settings.
SVMs can only handle small to	KSVMs can handle larger datasets due

SVMs are less commonly used in practice than KSVMs.

KSVMs have been shown to perform better on a wider range of tasks and datasets.

[Comment](#)[More info](#)[Advertise with us](#)

## Next Article

Implementing PCA in Python with  
scikit-learn

## Similar Reads

### Implementing SVM and Kernel SVM with Python's Scikit-Learn

In this article we will implement a classification model using Scikit learn implementation for SVM model in Python. Then we will try to understand what is a kernel and how it can help us to achieve better performance...

6 min read

### Implementing PCA in Python with scikit-learn

In this article, we will learn about PCA (Principal Component Analysis) in Python with scikit-learn. Let's start our learning step by step. WHY PCA? When there are many input attributes, it is difficult to visualize the data...

4 min read

### Implementing SVM from Scratch in Python

Support Vector Machines (SVMs) are powerful supervised machine learning algorithms used for classification and regression tasks. They work by finding the optimal hyperplane that separates data points of different...

4 min read

### Save and Load Machine Learning Models in Python with scikit-learn

In this article, let's learn how to save and load your machine learning model in Python with scikit-learn in this tutorial. Once we create a machine learning model, our job doesn't end there. We can save the model to use i...

4 min read

### Feature Selection in Python with Scikit-Learn

Feature selection is a crucial step in the machine learning pipeline. It involves selecting the most important features from your dataset to improve model performance and reduce computational cost. In this article, we

## What is fit() method in Python's Scikit-Learn?

Scikit-Learn, a powerful and versatile Python library, is extensively used for machine learning tasks. It provides simple and efficient tools for data mining and data analysis. Among its many features, the fit() method stand...

4 min read

---

## A beginner's guide to supervised learning with Python

Supervised learning is a foundational concept, and Python provides a robust ecosystem to explore and implement these powerful algorithms. Explore the fundamentals of supervised learning with Python in this...

10 min read

---

## Image processing with Scikit-image in Python

scikit-image is an image processing Python package that works with NumPy arrays which is a collection of algorithms for image processing. Let's discuss how to deal with images in set of information and its...

2 min read

---

## Save classifier to disk in scikit-learn in Python

In this article, we will cover saving a Save classifier to disk in scikit-learn using Python. We always train our models whether they are classifiers, regressors, etc. with the scikit learn library which require a considerable...

3 min read

---

## Implement Machine Learning With Caret In R

In today's society, technological answers to human issues are knocking on the doors of practically all fields of knowledge. Every aspect of this universe's daily operations generates data, and technology solutions base...

8 min read

---



Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate  
Tower, Sector- 136, Noida, Uttar Pradesh  
(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante  
Apartment, Sector 137, Noida, Gautam  
Buddh Nagar. Uttar Pradesh. 201305



Advertise with us

## Company

About Us  
Legal  
Privacy Policy  
In Media  
Contact Us  
Advertise with us  
GFG Corporate Solution  
Placement Training Program  
GeeksforGeeks Community

## DSA

Data Structures  
Algorithms  
DSA for Beginners  
Basic DSA Problems  
DSA Roadmap  
Top 100 DSA Interview Problems  
DSA Roadmap by Sandeep Jain  
All Cheat Sheets

## Web Technologies

HTML  
CSS  
JavaScript  
TypeScript  
ReactJS  
NextJS  
Bootstrap  
Web Design

## Computer Science

Operating Systems  
Computer Network  
Database Management System  
Software Engineering  
Digital Logic Design  
Engineering Maths

## Languages

Python  
Java  
C++  
PHP  
GoLang  
SQL  
R Language  
Android Tutorial  
Tutorials Archive

## Data Science & ML

Data Science With Python  
Data Science For Beginner  
Machine Learning  
ML Maths  
Data Visualisation  
Pandas  
NumPy  
NLP  
Deep Learning

## Python Tutorial

Python Programming Examples  
Python Projects  
Python Tkinter  
Web Scraping  
OpenCV Tutorial  
Python Interview Question  
Django

## DevOps

Git  
Linux  
AWS  
Docker  
Kubernetes  
Azure



High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

### School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

### GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects

---

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved