

# Support Vector Regression Tutorial for Machine Learning

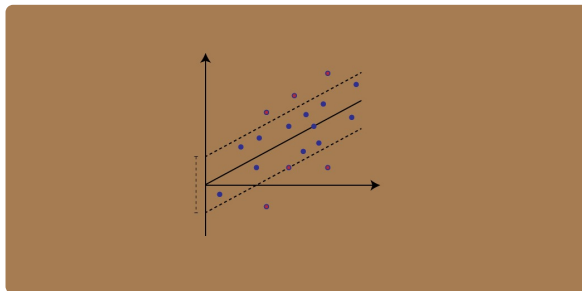


Alakh Sethi

09 Feb, 2024 • 8 min read

Support Vector Machines (SVM) are popularly and widely used for classification problems in machine learning. I've often relied on this not just in machine learning projects but when I want a quick result in a hackathon.

But SVM for regression analysis? I hadn't even considered the possibility for a while! And even now when I bring up "Support Vector Regression" in front of machine learning beginners, I often get a bemused expression. I understand – most courses and experts don't even mention Support Vector Regression (SVR) as a machine learning algorithm.



But SVR has its uses as you'll see in this tutorial. We will first quickly understand what SVM is, before diving into the world of Support Vector Regression in machine learning and how to implement it in Python!

*Note: You can learn about Support Vector Machines and Regression problems in course format here (it's free!):*

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using Analytics

Vidhya, you agree to our [Privacy Policy](#) and [Terms of Use](#).

Accept

- [Fundamentals of Regression Analysis](#)

Here's what we'll cover in this Support Vector Regression tutorial:

- What is a Support Vector Machine (SVM)?
- Hyperparameters of the Support Vector Machine Algorithm
- Introduction to Support Vector Regression (SVR)
- Implementing Support Vector Regression in Python

[Table of contents](#)



◀  
Reading list

## Quiz of the Day #3




Start Quiz

### What is a Support Vector Machine (SVM)?

A Support Vector Machine (SVM) is a supervised [machine learning](#) algorithm used for classification and regression tasks. SVM works by finding a hyperplane in a high-dimensional space that best separates data into different classes. It aims to maximize the margin (the distance between the hyperplane and the nearest data points of each class) while minimizing classification errors. SVM can handle both linear and non-linear classification problems by using various kernel functions. It's widely used in tasks such as image classification, text categorization, and more.


So what exactly is Support Vector [Machine](#) (SVM)? We'll start by understanding SVM in simple terms. Let's say we have a plot of two label classes as shown in the figure below:




17 Apr 2024 - 7:00 PM IST

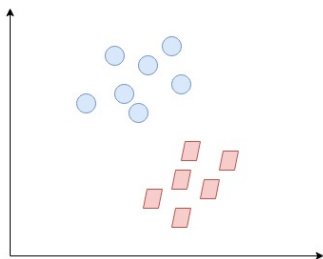
Setup Data Warehousing in Microsoft Fabric

RSVP Now

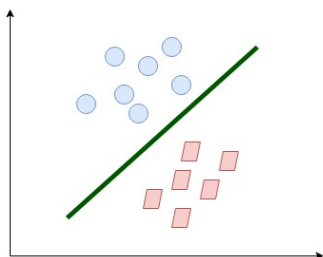


Rajendra Ongole  
Business Intelligence Specialist

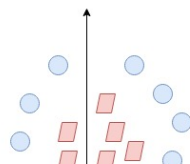


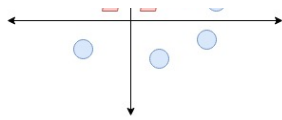


Can you decide what the separating line will be? You might have come up with this:

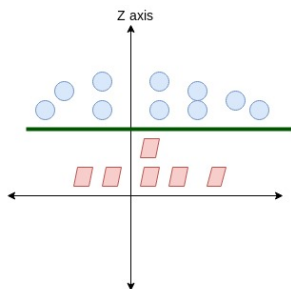


The line fairly separates the classes. This is what SVM essentially does – **simple class separation**. Now, what is the data was like this:





Here, we don't have a simple line separating these two classes. So we'll extend our dimension and introduce a new dimension along the z-axis. We can now separate these two classes:



When we transform this line back to the original plane, it maps to the circular boundary as I've shown here:

This is exactly what SVM does! It tries to find a line/hyperplane (in multidimensional space) that separates these two classes. Then it classifies the new point depending on whether it lies on the positive or negative side of the hyperplane depending on the classes to predict.

## Hyperparameters of the Support Vector Machine (SVM) Algorithm

There are a few important parameters of [SVM](#) that you should be aware of before proceeding further:

- Kernel:** A kernel helps us find a hyperplane in the higher dimensional space without increasing the computational cost. Usually, the computational cost will increase if the dimension of the data increases. This increase in dimension is required when we are unable to find a separating hyperplane in a given dimension and are required to move in a higher dimension:
- Hyperplane:** This is basically a separating line between two data classes in SVM. But in Support Vector Regression, this is the line that will be used to predict the continuous output
- Decision Boundary:** A decision boundary can be thought of as a demarcation line (for simplification) on one side of which lie positive examples and on the other side lie the negative examples. On this very line, the examples may be classified as

either positive or negative. This same concept of SVM will be applied in Support Vector Regression as well

To understand SVM from scratch, I recommend this tutorial: [Understanding Support Vector Machine\(SVM\) algorithm from examples.](#)

## Introduction to Support Vector Regression (SVR)

Support Vector [Regression](#) (SVR) is a type of machine learning algorithm used for regression analysis. The goal of SVR is 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 seeks to find a hyperplane that best fits the data points in a continuous space. This is achieved by mapping the input variables to a high-dimensional 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 can handle non-linear relationships between the input variables and the target variable by using a kernel function to map the data to a higher-dimensional space. This makes it a powerful tool for regression tasks where there may be complex relationships between the input variables and the target variable.

Support Vector Regression (SVR) uses the same principle as SVM, but for regression problems. Let's spend a few minutes understanding the idea behind SVR.

### The Idea Behind Support Vector Regression

The problem of [regression](#) is to find a function that approximates mapping from an input domain to real numbers on the basis of a training sample. So let's now dive deep and understand how SVR works actually.

Consider these two red lines as the decision boundary and the green line as the hyperplane. **Our objective, when we are moving on with SVR, is to basically consider the points that are within the decision boundary line.** Our best fit line is the hyperplane that has a maximum number of points.

The first thing that we'll understand is what is the decision boundary (the danger red line above!). Consider these lines as being at any distance, say 'a', from the hyperplane. So, these are the lines that we draw at distance '+a' and '-a' from the hyperplane. This 'a' in the text is basically referred to as epsilon.

Assuming that the equation of the hyperplane is as follows:

$Y = wx + b$  (equation of hyperplane)

Then the equations of decision boundary become:

$wx + b = +a$

$wx + b = -a$

Thus, any hyperplane that satisfies our SVR should satisfy:

$-a < Y - wx + b < +a$

Our main aim here is to decide a decision boundary at 'a' distance from the original hyperplane such that data points closest to the hyperplane or the support vectors are within that boundary line.

Hence, we are going to take only those points that are within the decision boundary and have the least error rate, or are within the Margin of Tolerance. This gives us a better fitting model.

## Implementing Support Vector Regression (SVR) in Python

Time to put on our coding hats! In this section, we'll understand the use of Support Vector [Regression](#) with the help of a dataset. Here, we have to predict the salary of an employee given a few independent variables. A classic HR analytics project!

### Step 1: Importing the libraries

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
```

svr1.py hosted with ❤ by GitHub

[view raw](#)

### Step 2: Reading the dataset

```
1 dataset = pd.read_csv('Position_Salaries.csv')
2 X = dataset.iloc[:, 1:2].values
3 y = dataset.iloc[:, 2].values
```

sv2.py hosted with ❤ by GitHub

[view raw](#)

### Step 3: Feature Scaling

A real-world dataset contains features that vary in magnitudes, units, and range. I would suggest performing normalization when the scale of a feature is irrelevant or misleading.

Feature Scaling basically helps to normalize the data within a particular range. Normally several common class types contain the feature scaling function so that they make feature scaling automatically. However, the SVR class is not a commonly used class type so we should perform feature scaling using Python.

```
1 from sklearn.preprocessing import StandardScaler
2 sc_X = StandardScaler()
3 sc_y = StandardScaler()
4 X = sc_X.fit_transform(X)
5 y = sc_y.fit_transform(y)
```

svr3.py hosted with ❤ by GitHub

[view raw](#)

## Step 4: Fitting SVR to the dataset

```
1 from sklearn.svm import SVR
2 regressor = SVR(kernel = 'rbf')
3 regressor.fit(X, y)
```

svr4.py hosted with ❤ by GitHub

[view raw](#)

Kernel is the most important feature. There are many types of kernels – linear, Gaussian, etc. Each is used depending on the dataset. To learn more about this, read this: [Support Vector Machine \(SVM\) in Python and R](#)

## Step 5. Predicting a new result

```
1 y_pred = regressor.predict(6.5)
2 y_pred = sc_y.inverse_transform(y_pred)
```

svr5.py hosted with ❤ by GitHub

[view raw](#)

So, the prediction for y\_pred(6, 5) will be 170,370.

## Step 6. Visualizing the SVR results (for higher resolution and smoother curve)

```
1 X_grid = np.arange(min(X), max(X), 0.01) #this step required because data is feature
2 X_grid = X_grid.reshape((len(X_grid), 1))
3 plt.scatter(X, y, color = 'red')
4 plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
5 plt.title('Truth or Bluff (SVR)')
6 plt.xlabel('Position level')
7 plt.ylabel('Salary')
8 plt.show()
```

svr6.py hosted with ❤ by GitHub

[view raw](#)

This is what we get as output- the best fit line that has a maximum number of points. Quite accurate!

## Conclusion

We can think of [Support Vector Regression](#) as the counterpart of SVM for regression problems. SVR acknowledges the presence of non-linearity in the data and provides a proficient prediction model.

I would love to hear your thoughts and ideas around using SVR for [regression](#) analysis. Connect with me in the comments section below and let's ideate!

regression

Support Vector Machine

support vector machine regression

SVM

svm kernel

svm regression

SVR





Alakh Sethi  
09 Feb 2024

Aspiring Data Scientist with a passion to play and wrangle with data and get insights from it to help the community know the upcoming trends and products for their better future. With an ambition to develop product used by millions which makes their life easier and better.

Algorithm

Beginner

Machine Learning

Python

Regression

## Frequently Asked Questions

Q1. What is the difference between SVM and SVR?



A. SVM (Support Vector Machines) is a classification algorithm that separates data points into different classes with a hyperplane while minimizing the misclassification error. On the other hand, SVR (Support Vector Regression) is a regression algorithm that finds a hyperplane that best fits data points in a continuous space while minimizing the prediction error. SVM is used for categorical target variables, while SVR is used for continuous target variables.

Q2. What is SVM regression?



Q3. What are the applications of SVM regression?



## Responses From Readers

What are your thoughts?...

Submit reply

Rahul Dev

19 Sep, 2020

Thanks for the article, it gave an intuitive understanding about SVR. It would be really helpful if you could also include the dataset used for the demonstration.



Venkat

16 Nov, 2020

The code is completely irrelevant to the dataset shown in the picture. Also, this code is from a Udemy course by Kiril Ermenko. At least give them the credit when you have plagiarized the code and content of the tutorial from elsewhere.




Junior Mukenze

02 Nov, 2022

Thank you for this article. It is very clear and helpful. However, I have

# Related Courses



• 24 Lessons

★ 5

Fundamentals of Regression Analysis

FREE

Machine Learning

Supervised ML

Enroll now

## Write for us →

Write, captivate, and earn accolades and rewards for your work

- ✓

Reach a Global Audience
- ✓

Cash In on Your Knowledge
- ✓

Get Expert Feedback
- ✓

Join a Thriving Community
- ✓

Build Your Brand & Audience
- ✓

Level Up Your Data Science Game



Prateek Majumder  
68

Company	Discover	Learn	Engage	Contribute	Enterprise
About Us	Blogs	Free courses	Community	Contribute & win	Our offerings
Contact Us	Expert session	Learning path	Hackathons	Become a speaker	Case studies
Careers	Podcasts	BlackBelt program	Events	Become a mentor	Industry report
	Comprehensive Guides	Gen AI	Daily challenges	Become an instructor	quexto.ai

Download App





