Support Vector Machines

in Machine Learning



**Anish Mahapatra**

Contents

[Introduction 4](#_Toc115715020)

[What is Support Vector Machines (SVM) in Machine Learning? 4](#_Toc115715021)

[Examples of SVM 4](#_Toc115715022)

[Why SVMs are used in Machine Learning 6](#_Toc115715023)

[Types of Support Vector Machines Algorithm 7](#_Toc115715024)

[Linear SVM 7](#_Toc115715025)

[Non-Linear SVM 7](#_Toc115715026)

[Hyperplane and Support Vectors in the SVM Algorithm 7](#_Toc115715027)

[How do we find the right hyperplane? 8](#_Toc115715028)

[How does SVM work in Machine Learning? 8](#_Toc115715029)

[Nonlinear data 8](#_Toc115715030)

[The Kernel Trick 8](#_Toc115715031)

[SVM Kernel Functions 8](#_Toc115715032)

[Simple SVM Classifier [Step-by-step] 8](#_Toc115715033)

[1.Create a new classifier 8](#_Toc115715034)

[2. Select how you want to classify your data 8](#_Toc115715035)

[3. Import your training data 8](#_Toc115715036)

[4. Define the tags for your SVM classifier 8](#_Toc115715037)

[5. Tag data to train your classifier 8](#_Toc115715038)

[6. Set your algorithm to SVM 8](#_Toc115715039)

[7. Test your classifier 8](#_Toc115715040)

[8. Integrate the topic classifier 8](#_Toc115715041)

[Applications of Support Vector Machine 8](#_Toc115715042)

[Advantages and Disadvantages of Support Vector Machine 8](#_Toc115715043)

[Conclusion 9](#_Toc115715044)

[Author 9](#_Toc115715045)

# Introduction

Welcome to the blog that is going to help you with Support Vector Machines. Machine Learning has been advancing over the years and so have the questions that are being asked in interviews. The focus has gone from the fundamental mathematics questions on simple linear regression and logisitic regression to more advanced machine learning algorithms. Interviewers tend to put forth more focus on models that you might not have seen as much, but are regularly used in the enterprise world to build and deploy models. One of the more popular models that is computationally acceptable and has great results is Support Vector Machines in machine learning. A model that is used for classification as well as regression, fundamental knowledge on support vector regression and classification should be known by all interviewees. In the following sections, we will go over the svm algorithm or svm machine learning to learn more about how to model works in depth, so that you are able to add support vector machine algorithm to your data science toolkit. In the following section, we will learn more about what is svm in machine learning.

# What is Support Vector Machines (SVM) in Machine Learning?

The svm model, or support vector machine model is a popular set of supervised learning models that is used for regression as well as classification analysis. It is a model that us based on the statstical learning framework and is known for being robust and effective in multiple use-cases. Based on a non-probabilistic binary linear classifier, support vector machine is used for seprating different classes with the help of various kernels that we will discuss in further sections. One of the main reasons that companies are leaning towards support vector mahine models as compares to other models is because support vector machines have significantly higher accuracy that can be leveraged, while using decreased computation from the system. One quick point to note here – SVM applications are generally implemented in the field of classification.

The questions on which kernel to choose while performing minimal computation is a big one, especially when we deal with larger datasets and this is done using something called the “kernel trick” and we will deep-dive into this topic in detail in a later section. Let us first get an intuition of support vector machines by looking at a few examples.

## Examples of SVM

In this section, we will look at a few svm examples. Let us forget all of this complex jargon that you might have read above and look at a solid example.

Q. What is the main goal of a classfication algorithm?

A. The main goal of a classification model in machine learning is to separate out different classes of points in an effective manner in a generalized manner. When doing this in a two dimensional (2-D) plane, it means drawing a straight line so that we are able to linearly separate out two classes of points in a manner that the future points also have a high probability of the points being separated out accurately.

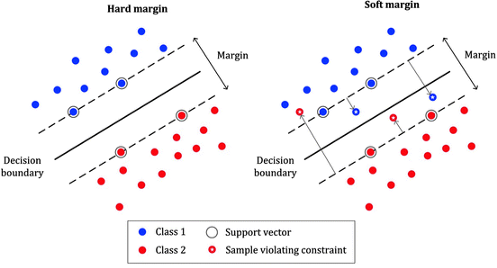
Using the below support vector machine example, we will also introduce some new terminology.



*Figure: Classification using support vector machine model* [*Source*](https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm)

Let us understand sme simple terminology:

* **Hyperplane**: Similar to how a line can separate out points in a two dimensional space, a hyperplane is the plane that an separate out points in a n-dimensional space
* **Positive** **Hyperplane**: The dotted line that we see in the figure that is situated in the poitive region is called the positive hyperplane. The positive hyperplane passes through the first point in the positive space.
* **Negative** **Hyperplane**: The dotted line that we see in the figure that is situated in the negative region is called the negative hyperplane. The negative hyperplane passes through the first point in the negative space.
* **Hard** **Margin**: A hard margin indicates that the svm model is trying to work extremely well on the dataset and can cause overfitting. This is generally used in linearly separable data, generally only in linearly separable data.
* **Soft** **Margin**: The soft margin indicats that the model is flexible in terms of fitting the dataset and so, will not cause overfitting. This is used in most cases when the data is not linearly separable. It allows some extent of miscalssification to make the model fit better on the test dataset.
* **Maximum Margin Hyperplane**: The decision boundary (indiated in the above figure as a solid line) is the decision boundary based on which the points are bifurcated.



*Figure: Soft Margin versus Hard Margin (*[*Source*](https://medium.com/swlh/the-support-vector-machine-basic-concept-a5106bd3cc5f#:~:text=SVM%20has%20a%20special%20way,finds%20the%20optimal%20separating%20hyperplane.)*)*

The idea behind selecting the decision boundary is that the larger the margin (difference between positive hyperplane and negative hyperplane), the lesser the generalization error as when we have smaller margins with the decision boundaries, it tends to lead to overfitting.

Besides this simple, yet effective example, support vector mahine is used to perform more complex use-cases such as categorization of text, classification of images and even face detection!

## Why SVMs are used in Machine Learning

The two main reasons why support vector machines are used in machine learning are:

* **Relatively high accuracy**: One of the main advantages of support vector macine is that as compared to more fundamental algorithms, it has a much higher relative accuracy. This means that when deploying the model in the real-world, we see better results from the machine learning models implemented.
* **Minimal Computation time**: Due to the “kernel trick”, the computation time of svm support vector machines is greatly reduced, which means that as data scientists, we are able to get beter results out in a reduced time, while utilizing lesses resources. This is a win-win, as we are able to get the better results, without affecting hardware utilization costs and even at a faster time!

In the next section, we will take a deep dive intor the types of support vector machine algorithms.

# Types of Support Vector Machines Algorithm

In this section, we will understand more about the types of SVM, based on the kind of data that we use. This is more specific to classification as that is the primary use-ase for support vector machines.

## Linear SVM

The linear support vector machine algorithm is used when we have linearly separable data. In simple language, if we have a dataset that **can** be classified into two groups using a simple straight line, we call it linearly separable data, and the classifier used for this is known as Linear SVM classifier.

## Non-Linear SVM

The non-linear support vector machine algorithm is used when we have non-linearly separable data. In simple language, if we have a dataset that **cannot** be classified into two groups using a simple straight line, we call it non-linear separable data, and the classifier used for this is known as Non-Linear SVM classifier.

# Hyperplane and Support Vectors in the SVM Algorithm

In this section, we will discuss more about hyperplane and support vectors in svm.

## Hyperplane

When given a set of points, there can be multiple ways to separate the classes in an n-dimensional space. The way that SVM works, it transforms the lower dimensional data into higher dimensional data and then separates out the points. There are multiple ways to separate out the data, these can be called decision boundaries, however, the main idea behind svm classification is to find the best possible decision boundary. The hyperplane is the optimal, generalized and best fit boundary for the support vector machine classifer.

For instance, in a two dimensional space, like we discussed in our example, the hyperplane will be s straight line, where as, if the data exists in a three dimensional space, then the hyperplane will exist in two dimensions. A good rule of thumb is that for a n-dimensional space, the hyperplane will generally have a n-1 dimension.

The aim is to create a hyperplane that has the highest possible margin, so as to create a geenralized model. This indicates that there will be the maximum distance between data points.

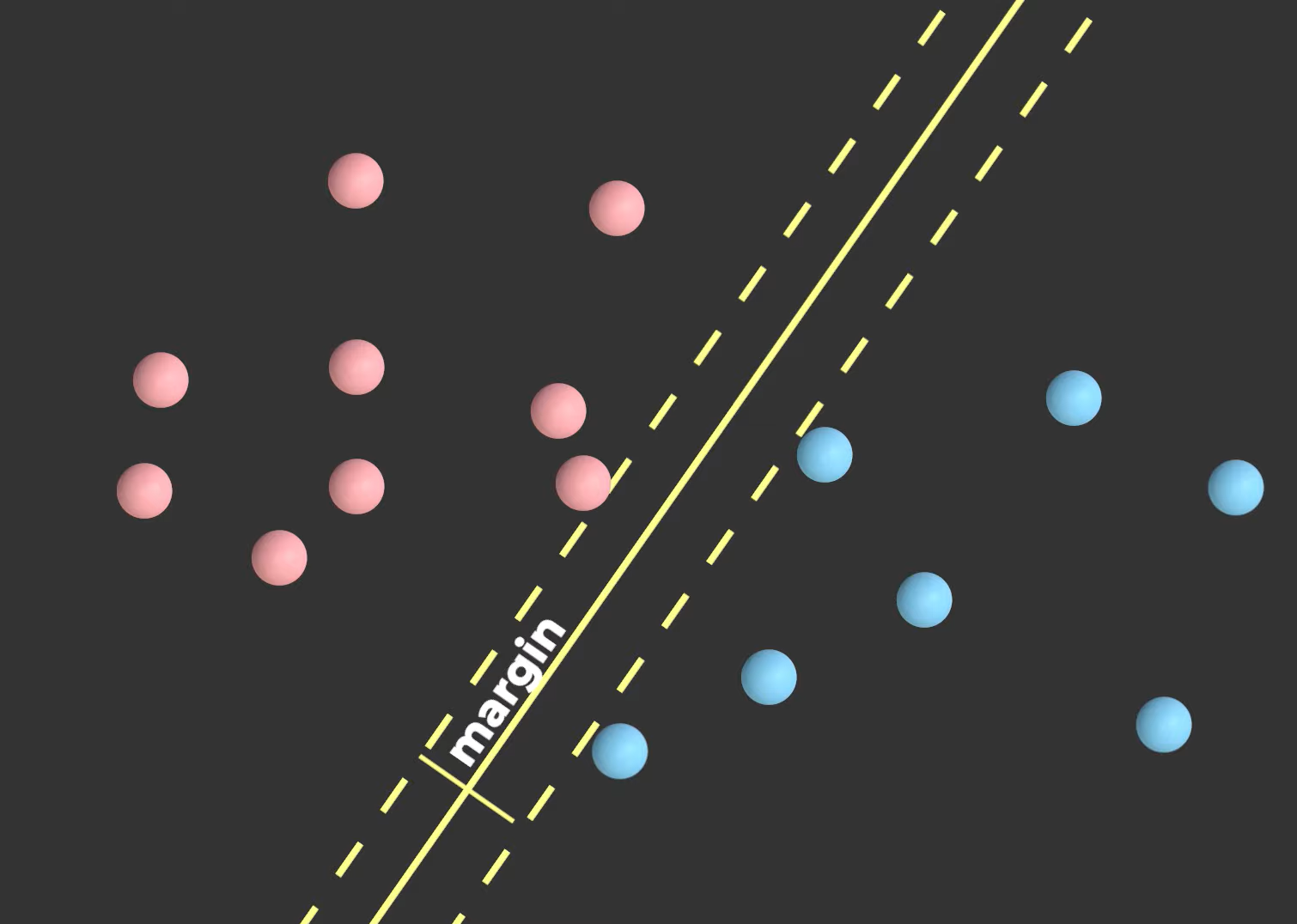
## Support Vectors

The term support vector indicates that we have supporting vectors to the main hyperplane. If we have maximum distance between the support vectors, it is an indication of best fit. So, support vectors are the vectors that pass through the closest points to the hyperplane and affect the overall position of the hyperplane.

## How do we find the right hyperplane?

Now, we come to a great question, how do we actually find the right hyperplane? Let us try to visualize and understand the two ways that we find the right hyperplane.

* **Maximize the margin between support vectors**The recommended way to find the right hyper-plane is by maximizing the distance between the support vectors. Below, we visualize what this will look like in a two-dimensional space, this can also be done in a n-dimensional space, but it will be difficult for us to visualize.

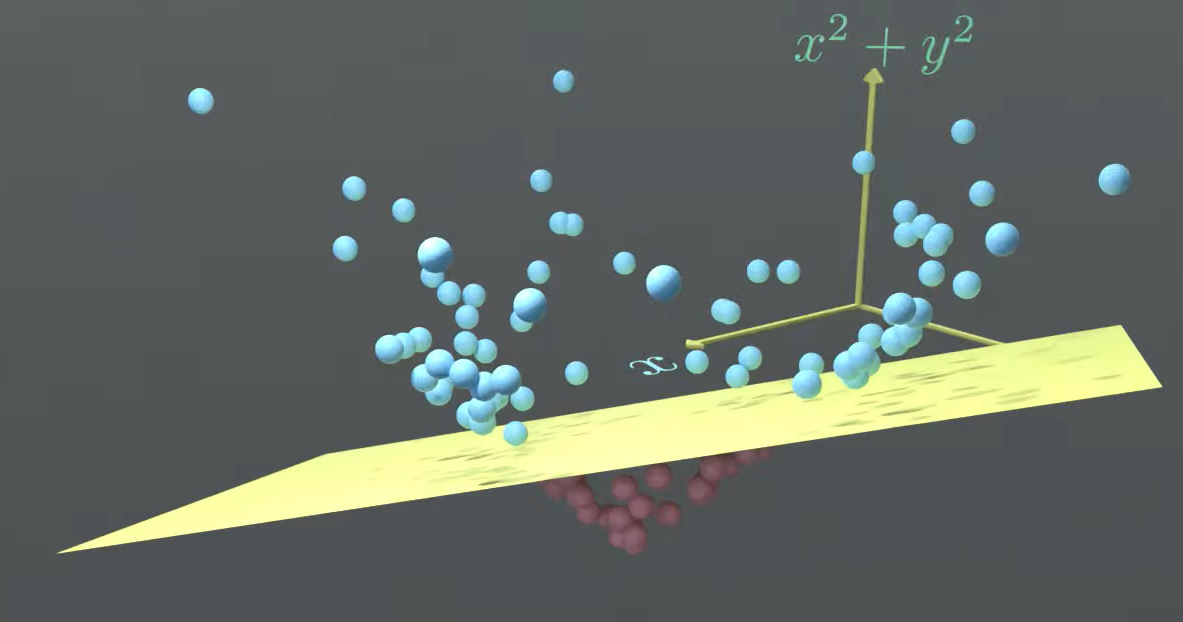


*Maximize the margin between the support vectors (*[*Source*](https://www.youtube.com/watch?v=_YPScrckx28)*)*

* **Transform lower dimensional data into higher dimensional data**When we transform lower dimensional data into higher dimensional data, with the help of new features created, it separated out the points in a higher dimension and we can then pass a hyperplane with more efficiency to segragate out the data.



*Two-dimensional view of the data on the X and Y axes (*[*Source*](https://www.youtube.com/watch?v=_YPScrckx28)*)*

*******Projecting the data in a three dimensional space (*[*Source*](https://www.youtube.com/watch?v=_YPScrckx28)*)*

This is done with the help of the following steps:

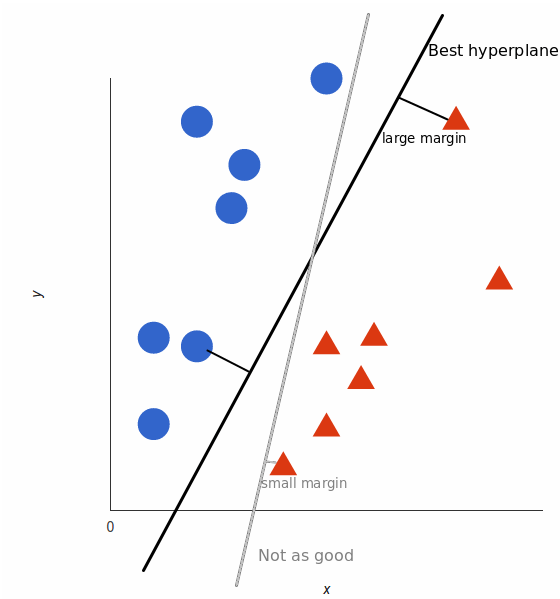
1. Augment the data with some non-linear features that are computed using the existing features
2. Find the separating hyperplane in the higher dimensional space
3. Project the points back to the original space

# How does SVM work in Machine Learning?

SVM works based on the principle of maximizing the distance between the support vectors, this ensures that we have the maximum margin possible between points, thus, giving us a generalized model. The aim of support vector machine classification is to maximize the margin between the support vectors. Let us go over the ways that SVM works in machine learning.

## Linearly separable data

We use kernels in support vector machines. SVM kernels are functions based on which we can transform the data so that it is easier to fit a hyperplane to better segregate the points.

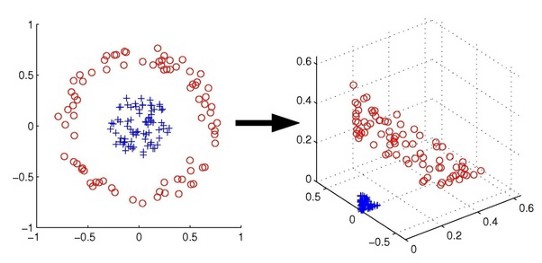


*Linearly separable data (*[*Source*](https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/)*)*

Linearly separable points generally consist of the points that can be separated by a simple straight line. The line has to have the largest margin possible between the closest points to form a geenralized svm model.

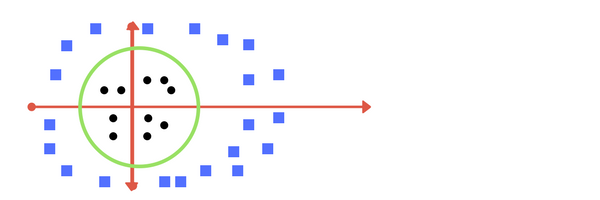
## Nonlinear data

Non-linear data is data that cannot be separated via a simple straight line. We can separate out the classes by mapping the data into a higher dimensional space, such that we are able to classify the points. Here, we use derived higher dimensional features from the dataset itself. For instance, with a dataset that is present in the X and Y axis, we will use features such as X2 , Y2, and XY to make a higher dimensional model, project the data, make the hyperplane and then revert the data to it’s original space.



*Figure: A two dimensional data depicted as three dimensional data (*[*Source*](https://medium.com/swlh/the-support-vector-machine-basic-concept-a5106bd3cc5f#:~:text=SVM%20has%20a%20special%20way,finds%20the%20optimal%20separating%20hyperplane.)*)*

This is done using a clever trick that we will discuss in the next section. In the end, the figure will end up looking like the below figure, which separates out the two classes in the same original space.



*Figure: Separating the points out by mapping the   
three-dimensional space to a two-dimensional space (*[*Source*](https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72)*)*

## The Kernel Trick

The kernel trick is the “superpower” of support vector machines. Essentially, support vector machine uses something called kernels, which is a function based on which the points can be segregated. The points that are no-linearly separable are projects to a higher dimensional space.

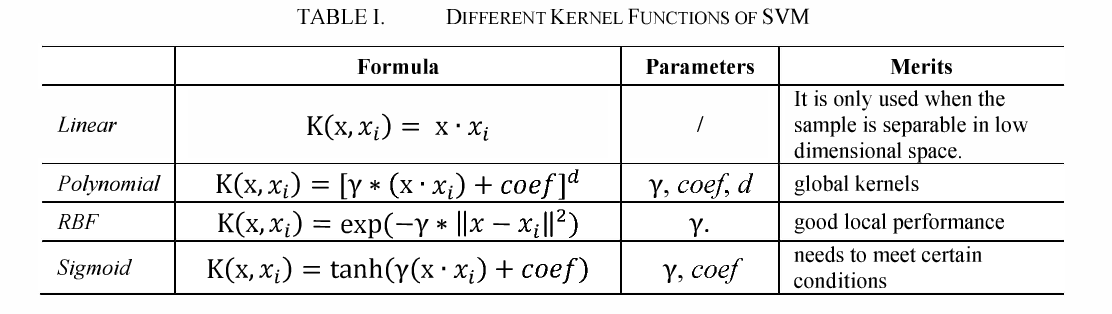
**Q. So, what is the “trick” here?**

A. SVM represents the non-linear data points in a fashion where the data points seem to be transformed, and then finds the hyperplane. However, in reality, the points actually remain the same and they have not been transformed.

This trick is the reason that the seeming transformation of the points from a lower to higher dimension is known as the kernel trick.

# SVM Kernel Functions

We have been talking about svm kernels for a while now, let us briefly go over some of the important kernel functions that help transform the data to pass hyperplanes to segregate the data. All of the neat tricks we talk about is math, the transformations of the data are performed using linear algebra. We are going to go into a little bit of mathematics now, as this will help give you an intuition of the kernel.

*Mathematical formulas of the kernel functions (*[*Source*](https://www.semanticscholar.org/paper/The-optimization-of-kernel-function-and-its-for-SVM-Ren-Hu/3a92a26a66efba1849fa95c900114b9d129467ac)*)*

* **Linear Kernel** **Function**  
  The linear kernel is primarily leveraged for linearly separable data. It is used for points that have a linear relationship.
* **Polynomial Kernel Function**   
  The polynomial kernel function is used by leveraging the dot product and transforming the data to a n-dimension. This helps represent the data with a higher dimension leveraging newly transformed data points.
* **RBF (Radial basis Function)**This is one of the most common and widely used functions as a kernel, which behaves similar to a weighted nearest neighbour model. It can transform the given data in infinitte dimensions and then leverage the weighted nearest neighbour model to ientify the observations that has the highest influence on the new data point for the classification. The ‘Radial’ function in RBF can either be Laplace or Gaussian. We can decide this based on the ‘Gamma**’** hyperparameter**.**
* **Sigmoid function**The sigmoid function is found in use-cases such as neural networks, where it is used as an activation function (Tanh). It is also known as the hyperbolic tangent function and has certain use-cases where it is able to segregate the data better.

That was support vector machine explained. We have now learnt about the various kernels that are used in support vector machine functions. Next, we will go over the svm classifier python code.

# Simple SVM Classifier [Step-by-step]

In this section, we will look over the svm implementation in Python. We will quickly go over an example of Python code to see support vector machines in action.

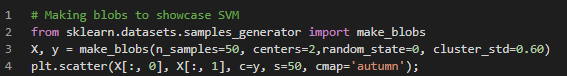
## Import the required libraries

The support vector machine can be used from the SVC python library, which stands for support vector classifier. It is a supervised learning algorithm with is used to perform classification and can be founf in sci-kit learn. We can look at two use-cases of a dataset that has a linear and non-linear distribution for this python showcase. We will make a blob to showcase the data in the next section.

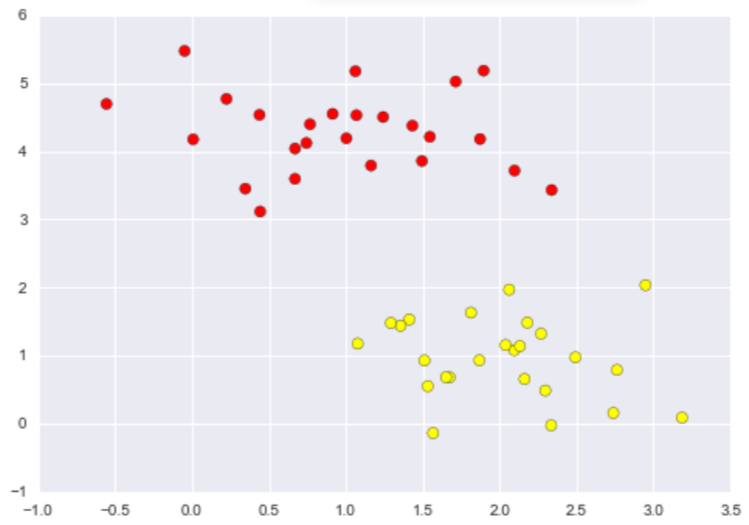


## Import the required dataset

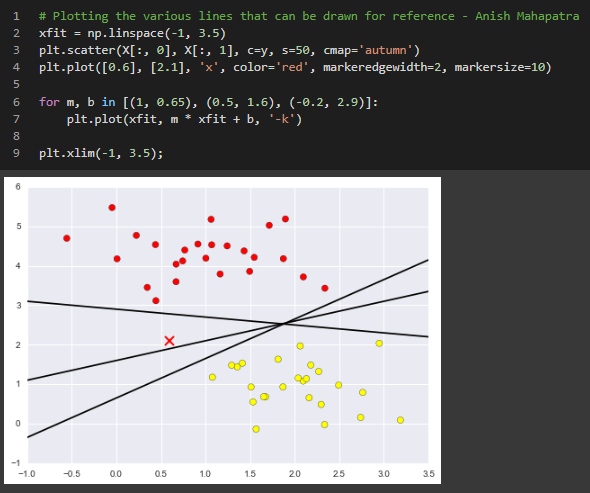
Generally, in this case, we should import the required dataset, perform the necessary pre-processing steps, and then analyze and visualize the data. Here, in this case, we will geenrate two blobs to showcase the power of support vector machines, using kernels that we have discussed in previous sections.



And, our dataset will look something like this, where we would like to showcase the linear separation of the data.



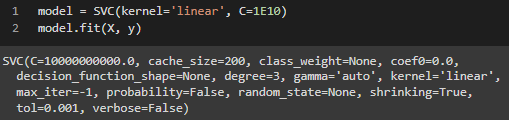
Now, when if were to used a linear discriminative classifier, we would attempt to find an optimal straight line betwee the two sets of data such that we are able to segregate the datasets. There are various lines that can be drawn to segregate the datasets.



Confused about which one to choose? Remember what we discussed in the previous sections? Our aim is to **maximize the margin**. In the next section, we will discuss exactly this.

## Maximize the margin

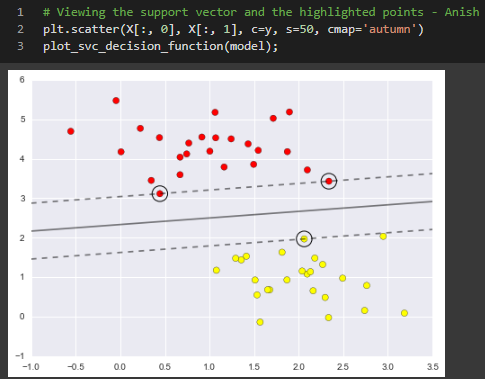
Now, we need to fit the linear support vector machine so that we are able to plot the optimal hyperplane to get the best fit model. In this case, we will be using the linear kernel, as the points in the X and Y axis have a linear relationship.



We use the linear kernel within the support vector classifier (svc) from the Sciket learn package to segregate the datasets appropriately. The aim of the dividing decision boundary is to maximize the margin between the different groups of points. Some of the points touch the line and are indicated separately. These points are critial, and are known as support vectors, they are stored in the support\_vectors\_ attribut of the function.

## Fit the support vector machine classifier

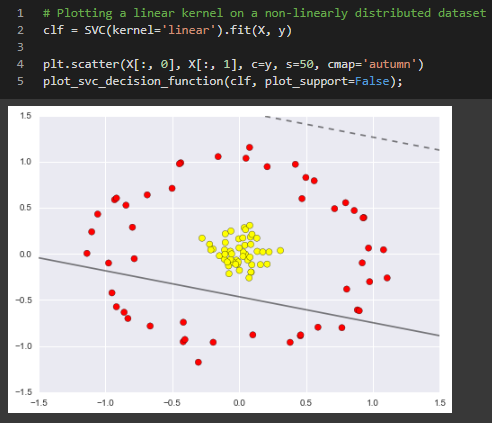
Based on hyperparameter tuning, it is to be decided on what the best possible model for the given dataset will be. We notice the support vectors here and the position of the dividing straight line (called hyperplane for n-dimensional data) will change based on how the margins can be maximized.



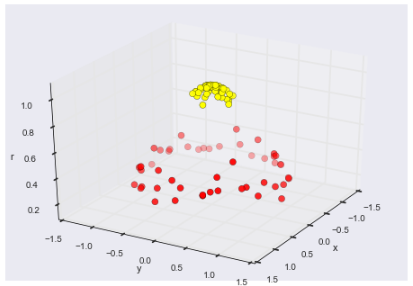
Based on the parameters and the number of rows in the train and test data, the positiion and accuracy of the svm model will vary.

## Decide the kernel type based on the data distribution type

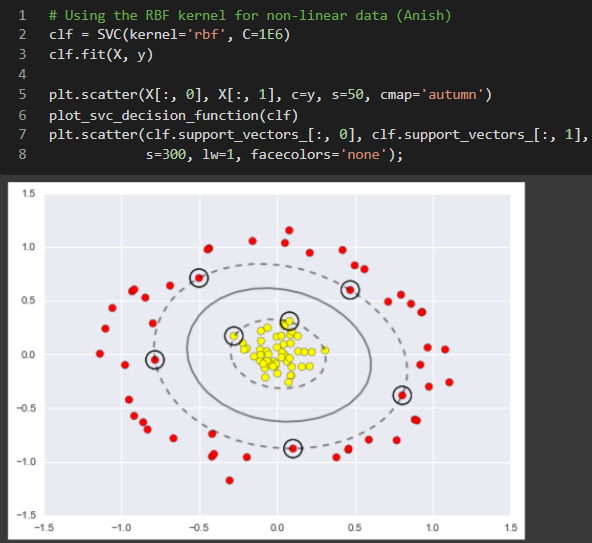
Based on the distribution of the data, it is possible to also have a non-linear dataset distribution that can be solved using other kernels. For instance, if we were to use a linear kernel on a non-linearly distributed dataset, we would see a plot that looks similar to the following.



Whereas, if we were to project and transform the two dimensional data onto a three dimensional space, it would look like the following.



Here, in this case, if we used the rbf kernel, the plot would look like the below image, where we have successfully segregated and mapped the hyperplace back to the original points.



In this section, we successfully went over some simple Python code to generate relavent datasets and showcased how support vector machines can be used to generate fairly accurate models with minimal computation using the kernel trick. In the next section, we will go over some of the applications of support vector machines.

# Applications of Support Vector Machine

In this section, we will go over some of the use-cases of support vector machines.

* Email classification
* Face Detection
* Text categorization
* Handwriting recognition
* Bioinformatics

[Source 1](https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/#:~:text=SVMs%20are%20used%20in%20applications,linear%20and%20non%2Dlinear%20data.), [Source 2](https://data-flair.training/blogs/applications-of-svm/)

# Advantages and Disadvantages of Support Vector Machine

**Pros**

Effective on datasets with multiple features, like financial or medical data.

Effective in cases where number of features is greater than the number of data points.

Uses a subset of training points in the decision function called support vectors which makes it memory efficient.

Different kernel functions can be specified for the decision function. You can use common kernels, but it's also possible to specify custom kernels.

**Cons**

If the number of features is a lot bigger than the number of data points, avoiding over-fitting when choosing kernel functions and regularization term is crucial.

SVMs don't directly provide probability estimates. Those are calculated using an expensive five-fold cross-validation.

Works best on small sample sets because of its high training time.

# Conclusion

# Author

**Anish Mahapatra**   
*Senior Data Scientist*   
LinkedIn: [https://www.linkedin.com/in/anishmahapatra](about:blank) ([LinkedIn](about:blank))   
Email: anishmahapatra01@gmail.com