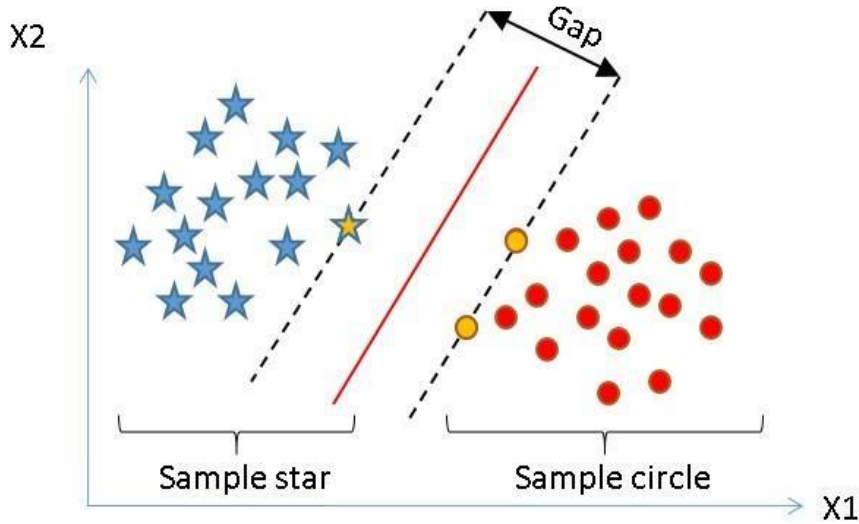


### 6.2.3. Destek Vektör Makineleri

Destek Vektör Makinesi, farklı sınıfları ayırmak ve sınır marjını en üst düzeye çıkarmak için karar sınırlarını bulmaktan sorumludur. Farklı sınıfların sınırları arasındaki boşluklar, çizgi ile çizgiye en yakın noktalar arasındaki (dik) mesafelerdir. DVM'de sınıflar arasındaki sınırlar çok önemlidir.

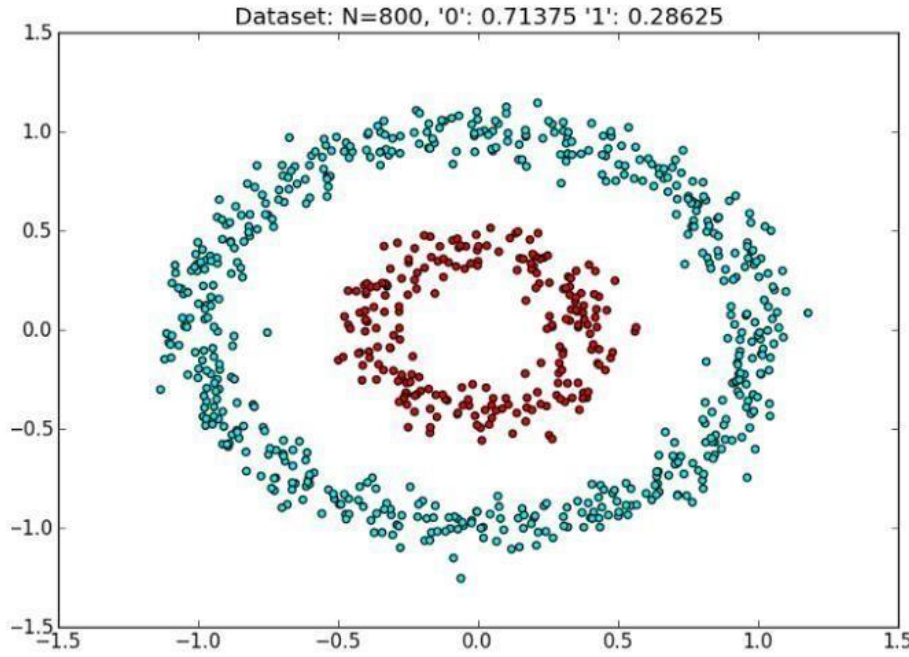
SVM, sınıflandırma için kullanılan denetimli algoritmalar sınıfında yer alır. 2 sınıf bir örnekle başlayalım: Verilen  $X_1$  ve  $X_2$  sınıfları için, 2 sınıfı en iyi, yani minimum hata ile ayıran karar sınırını bulmak istiyoruz. SVM bunu bir "Hiperplane" ile yapar. Şimdi bu hiperdüzlem 2 boyutlu veri olması durumunda tek bir doğru olabilir ve 3 boyutlu veri olması durumunda bir düzlem olabilir.



Destek Vektör Makineleri, hiper düzleme en yakın noktalar olan 'Destek Vektörleri' kavramını kullanır. Yukarıdaki örnekte, kırmızı çizgi, 2 sınıfı (Mavi yıldızlar ve Kırmızı daireler) ayıran karar sınırımızı gösterir ve tireli çizgiler, her iki sınıfın Destek Vektörleri arasında istediğimiz boşluğu, 'Marj'ımızı temsil eder. Sınırlar Önemlidir

Marj, Destek Vektörlerinin (dolayısıyla adı) yardımıyla tanımlanır. Örneğimizde, Sarı yıldızlar ve Sarı daireler, Marjı tanımlayan Destek Vektörleridir. Boşluk ne kadar iyi olursa, sınıflandırıcı o kadar iyi çalışır. Bu nedenle destek vektörleri sınıflandırıcının geliştirilmesinde önemli bir rol oynamaktadır. Test verilerindeki her yeni veri noktası bu Marj'a göre sınıflandırılacaktır. Sağ tarafındaysa Kırmızı daire, aksi halde Mavi yıldız olarak sınıflandırılır.

En iyi yanı, SVM'nin doğrusal olmayan verileri de sınıflandırabilmesidir.



Doğrusal olmayan veriler söz konusu olduğunda işler biraz zorlaşır. Burada SVM 'Çekirdek hilesi' kullanır, doğrusal olmayan verileri daha yüksek boyutlara eşlemek için bir çekirdek işlevi kullanır, böylece doğrusal hale gelir ve orada karar sınırını bulur. Bir Çekirdek işlevi, ister doğrusal ister doğrusal olmayan veri olsun, SVM tarafından her zaman kullanılır, ancak ana işlevi, veriler mevcut haliyle ayrılmaz olduğunda devreye girer. Burada, Çekirdek işlevi, sınıflandırma sorununa boyutlar ekler.

Soruna bağlı olarak, farklı türde Çekirdek işlevleri kullanabilirsiniz:

- Doğrusal
- Polinom
- Radyal Temel Fonksiyonu
- Gauss
- Laplace

... ve daha fazlası. Doğru çekirdek işlevini seçmek, sınıflandırıcıyı oluşturmak için önemlidir.

Destek vektör ağırları olarak da bilinen destek vektör makineleri, sınıflandırma ve regresyon için kullanılan ilgili denetimli öğrenme yöntemleri kümesidir. Her biri iki kategoriden birine ait olarak işaretlenmiş bir dizi eğitim örneği verildiğinde, bir DVM eğitim algoritması yeni bir örneğin bir kategoriye mi yoksa diğerine mi düştüğünü tahmin eden bir model oluşturur. Bir DVM eğitim algoritması olasılık dışı, ikili, doğrusal bir sınıflandırıcıdır, ancak Platt ölçeklendirme gibi yöntemler olasılıklı bir sınıflandırma ayarında DVM'yi kullanmak için mevcuttur. Doğrusal sınıflandırmayı gerçekleştirmenin yanı sıra, DVM'ler, girdilerini yüksek

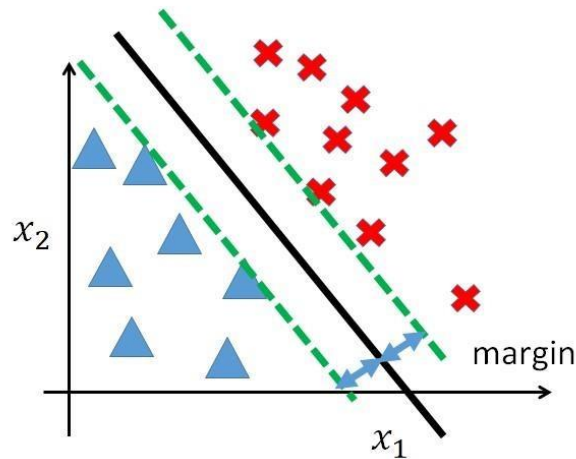
boyutlu özellik alanlarına dolaylı olarak eşleyerek çekirdek hilesi adı verilen kullanarak doğrusal olmayan bir sınıflandırma gerçekleştirebilir.

**SVM'nin (Destek Vektör Makinesi) kullanabileceği iki sınıflandırma yöntemi:**

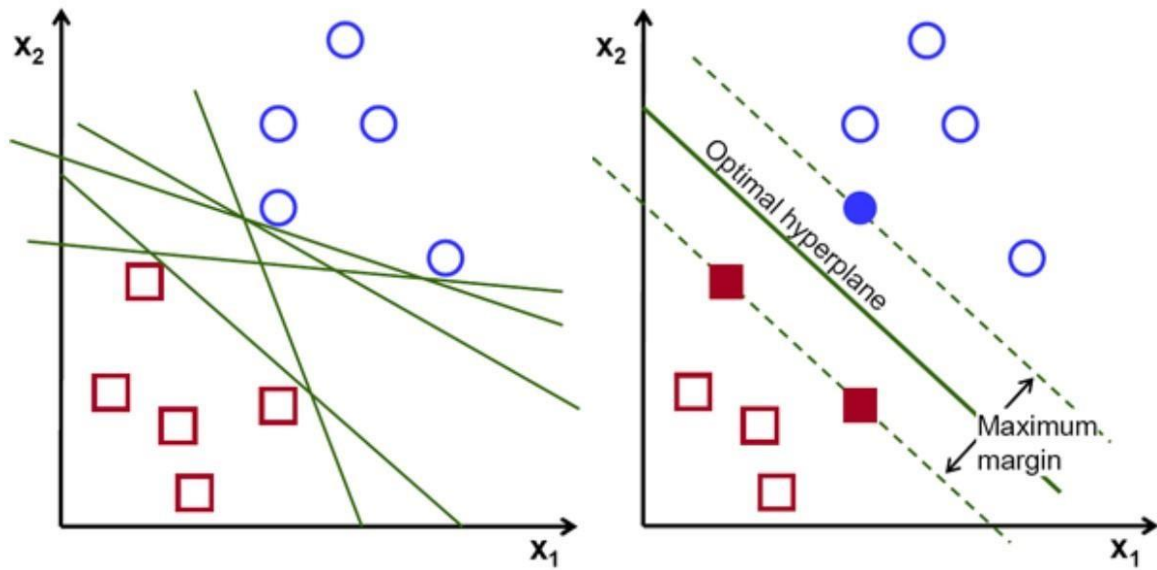
- İkili sınıflandırıcıları birleştirme
- Çok sınıflı öğrenmeyi dahil etmek için ikili programı değiştirme

**Destek Vektör Makinesi algoritması oluşturulurken,**

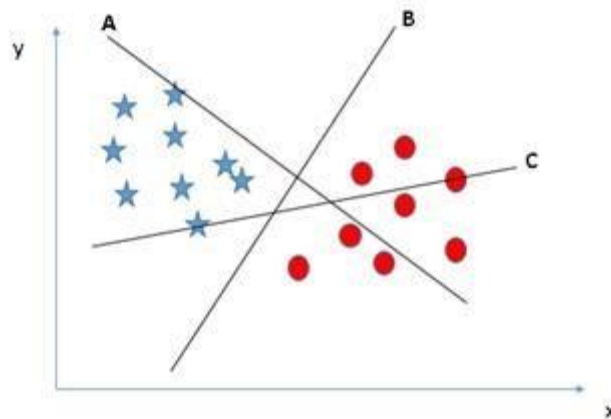
- Sınıflar arasındaki birbirine en yakın ikili seçilir. Bu noktalara destek vektörleri isimleri veriyoruz.
- Destek vektörlerinden geçecek şekilde doğrular çizilir. Bu doğrulara sınır çizgisi adı verilir.
- Her doğruya eşit uzakta çizilen doğruya karar doğrusu adı verilir. Karar doğrusuna hiper düzlem de denir.



- DVM'lerde sınıflar +1 ve -1 olarak etiketlenir. Karar doğrusunun üst kısmında kalan doğru denklemi,  $wx+b=1$ , altında kalan doğru denklemi ise  $wx+b=-1$  olarak belirlenir. Karar doğrusu denklemi ise,  $wx+b=0$  olur.

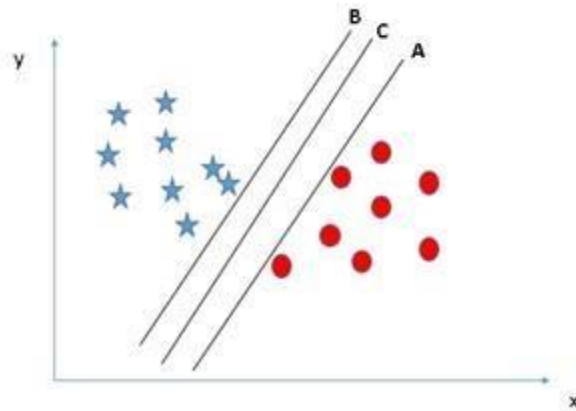


- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B, and C). Now, identify the right hyper-plane to classify stars and circles.



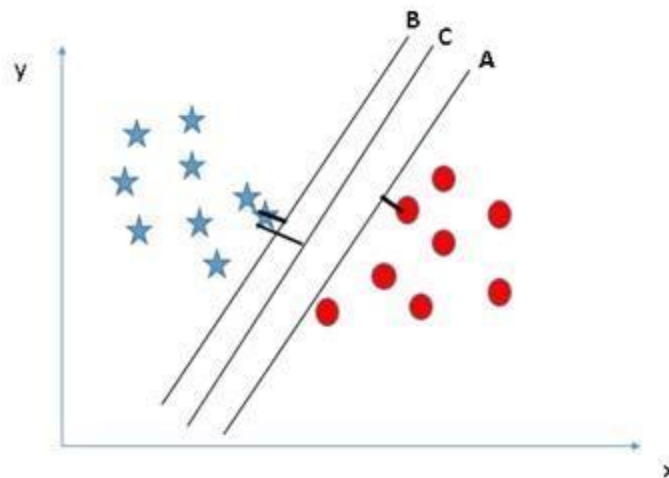
You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B, and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?



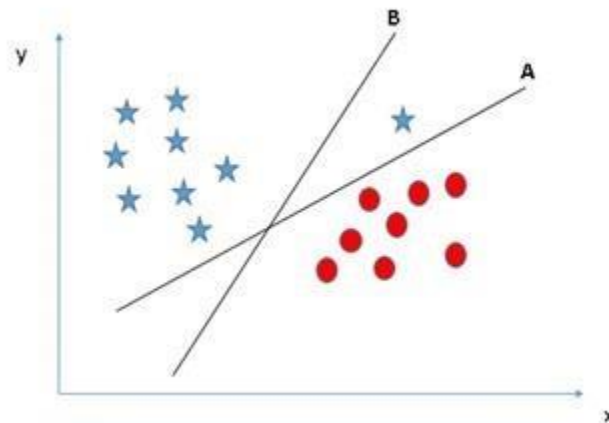
Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**.

Let's look at the below snapshot:



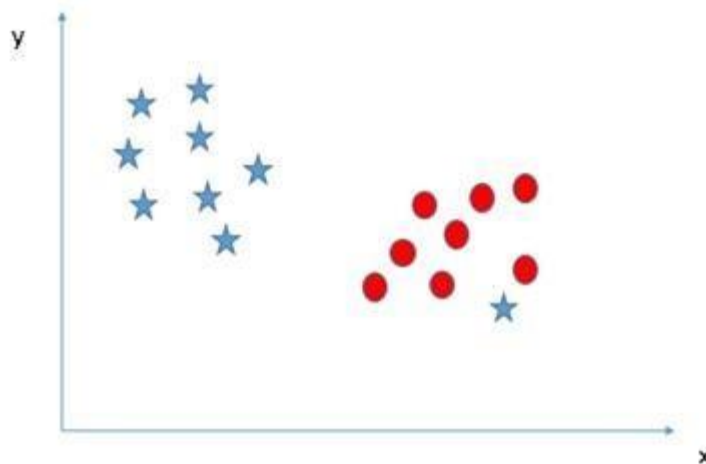
Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyperplane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

- **Identify the right hyper-plane (Scenario-3):** Hint: Use the rules as discussed in previous section to identify the right hyper-plane

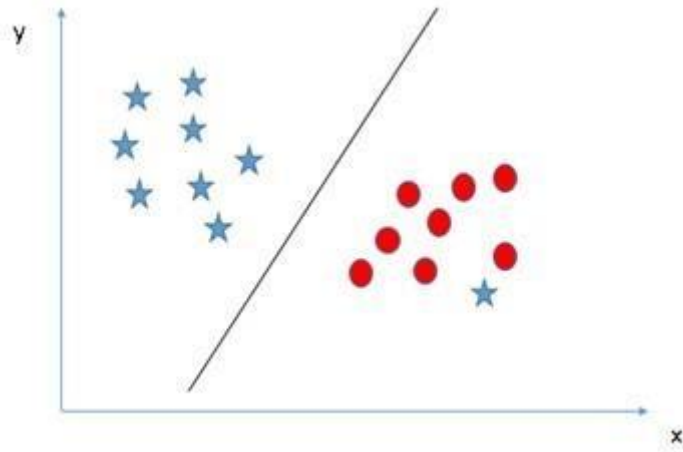


Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A**. But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A**.

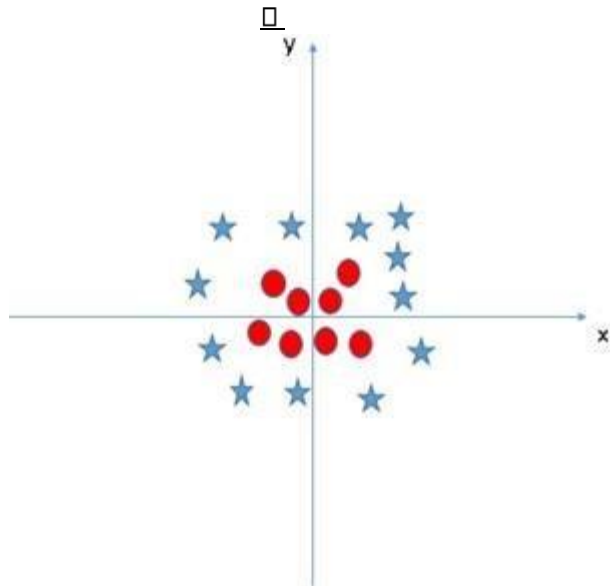
- **Can we classify two classes (Scenario-4)?:** Below, I am unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.



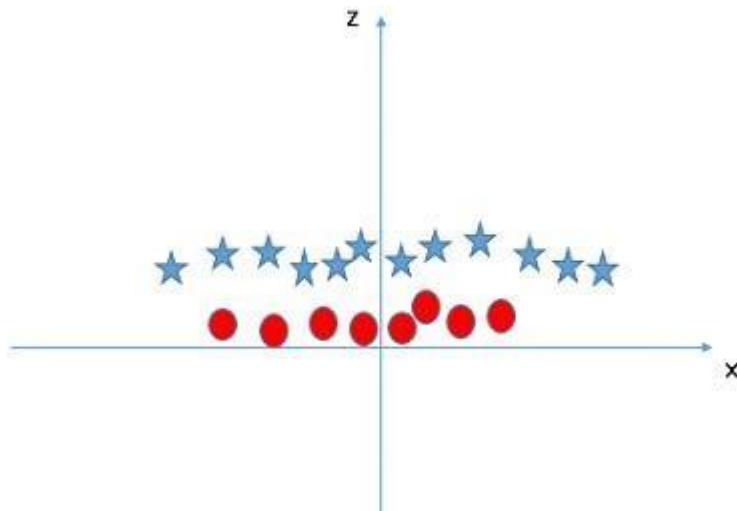
As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



- **Find the hyper-plane to segregate to classes (Scenario-5):** In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.



- SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature  $z = x^2 + y^2$ . Now, let's plot the data points on axis x and z:

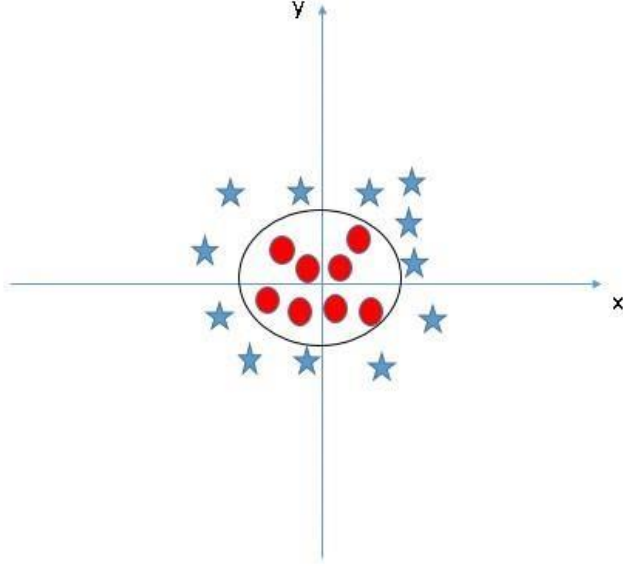


In above plot, points to consider are:

- All values for  $z$  would be positive always because  $z$  is the squared sum of both  $x$  and  $y$
- In the original plot, red circles appear close to the origin of  $x$  and  $y$  axes, leading to lower value of  $z$  and star relatively away from the origin result to higher value of  $z$ .

In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, the SVM algorithm has a technique called the [kernel trick](#). The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in nonlinear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined. When we look at the hyper-plane in original input space it looks like a circle:





**Örnek:**

İki sınıf arasında dairesel bir halka boşluğu bulunmaktadır. Halkanın iç yarı çapı 5cm, dış yarı çapı 10cm dir. Bir nokta halkanın içinde ve dış yarı çapına 1cm mesafede ise hangi yorum doğrudur?

Bu değer halkanın dışına yakın olan sınıfa aittir.