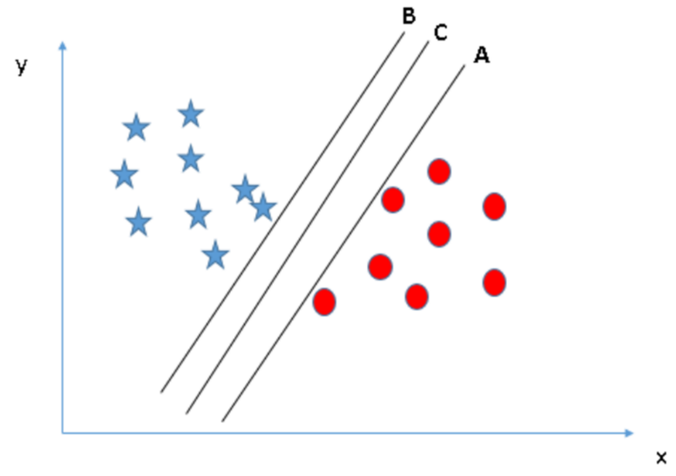
How does it work?

Scenario-2 (Linearly Separable Data)

Identify the right hyper-plane :

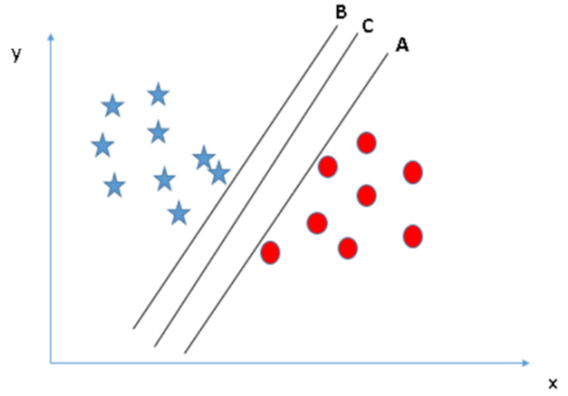
- Here, we have three hyper-planes (A, B and C) and **all** are **segregating the classes** well.
- Now, How can we identify the right hyper-plane?



How does it work?

Scenario-2 (Linearly Separable Data)

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

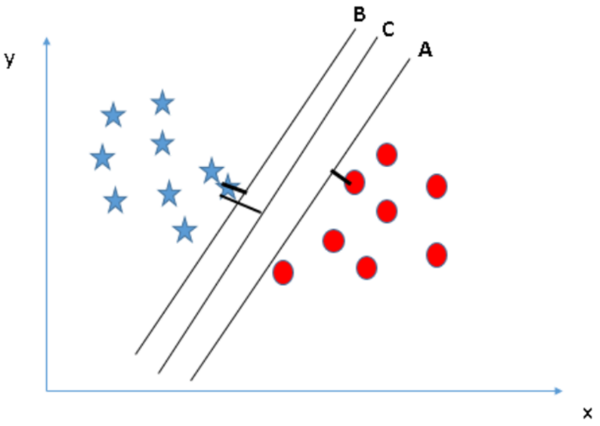


- This distance is called **Margin**.

How does it work?

Scenario-2 (Linearly Separable Data)

- 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.



How does it work?

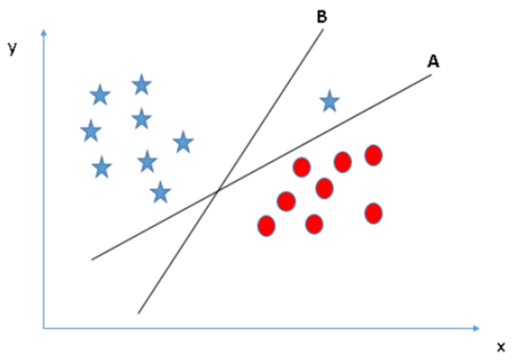
Scenario-2 (Linearly Separable Data)

- Another lightning reason for selecting the hyper-plane with higher margin is **robustness**.
- If we select a hyper-plane having **low margin** then there is **high chance** of **miss-classification**.

How does it work?

Scenario-3 (Linearly Separable Data)

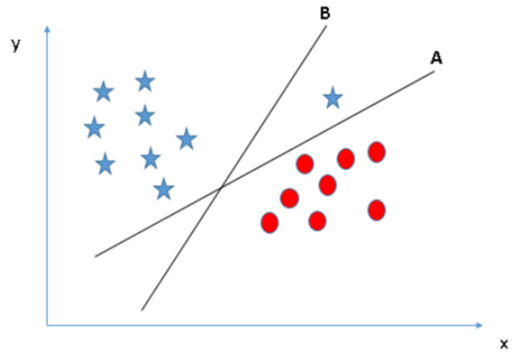
Identify the right hyper-plane :



How does it work?

Scenario-3 (Linearly Separable Data)

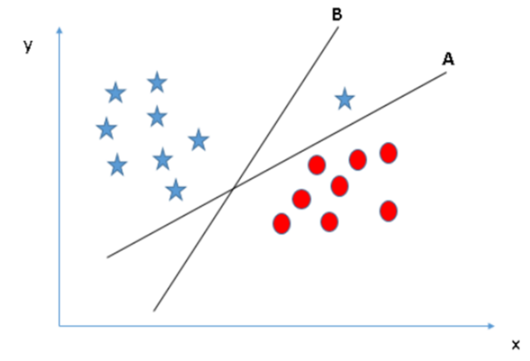
- Some of you may have **selected** hyper-plane **B as it has higher margin compared to A**.



How does it work?

Scenario-3 (Linearly Separable Data)

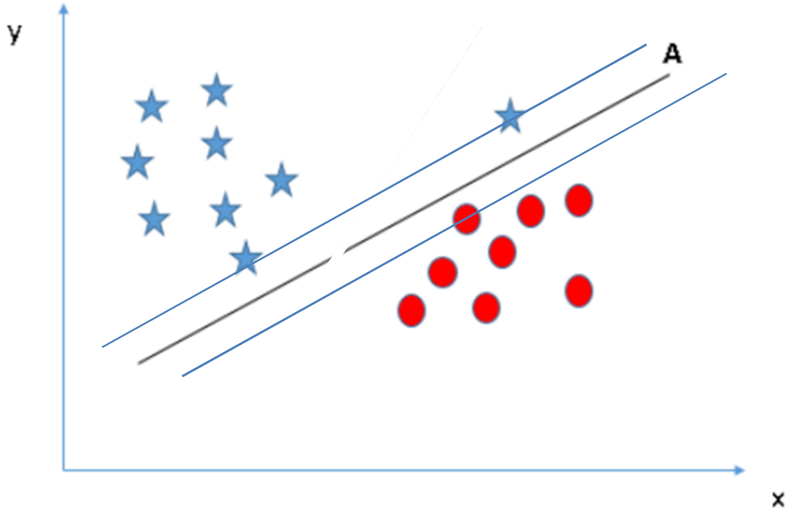
- SVM selects the hyper-plane which classifies the **classes accurately** prior to **maximizing** margin.



- Here, hyper-plane **B** has a **classification error** and A has **classified all correctly**. Therefore, the right hyper-plane is **A**.

Problem with line A

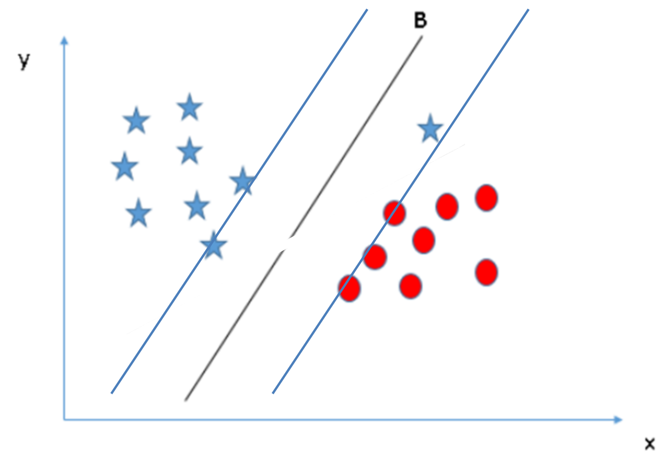
- If we select a hyper-plane having **low margin** then there is **high chance** of **miss-classification** on unseen data.
- The model is **overfitting**.
- Also the model is **less confident** while classifying data points.



Benefit of choosing Line B

- **High confidence** while classifying.
- **Generalizes well** on **unseen data**.
- **Less over-fitting**.

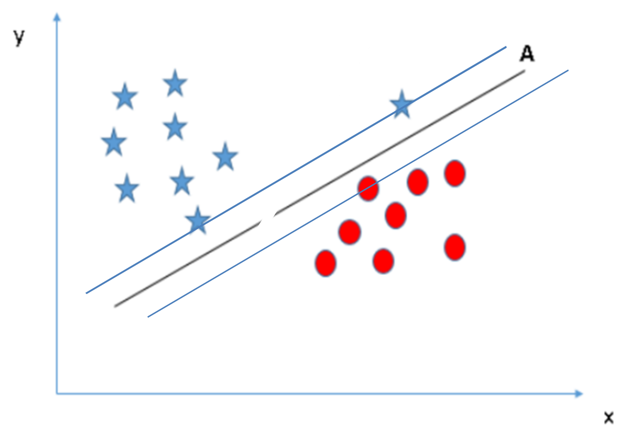
- But how to **make** the **model** **choose Line B** instead of **Line A**



Regularization Parameter C

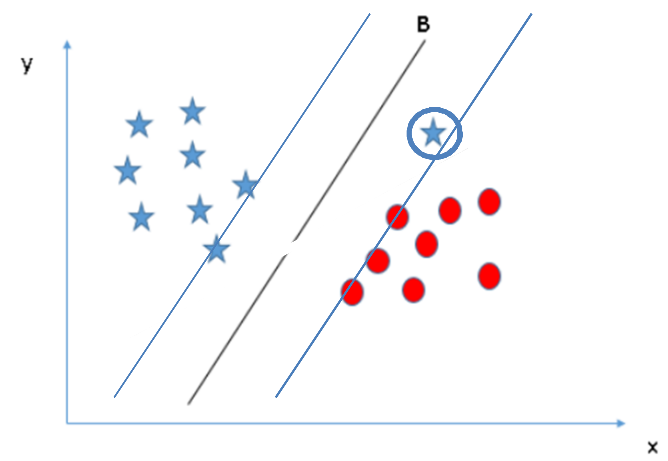
- Used **to control no. of mistakes** that a **model** is **allowed to make**.

- If **C value is high**, **model is not allowed** to **make any mistakes**, sensitive to outliers also – **Hard Margin SVM**

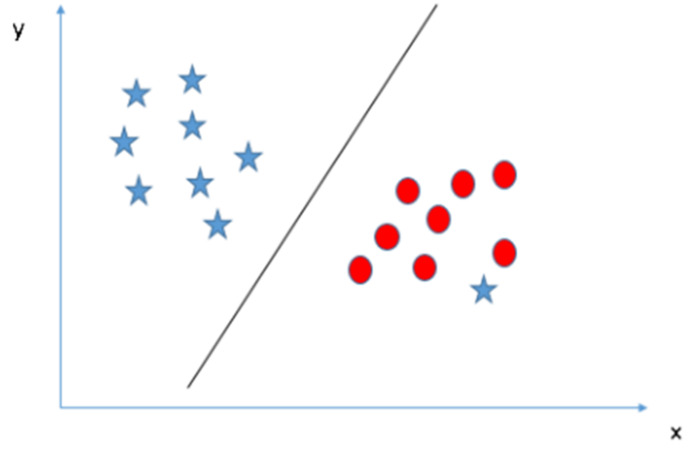


Regularization Parameter C

- If **C value is low**, **model can make some mistakes**.
- Not affected by outliers.
- Generalizes well on unseen data – **Soft Margin SVM**



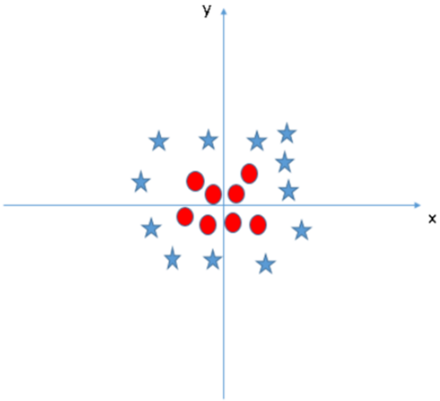
Soft Margin SVM



How does it work?

Scenario-4 (Linearly Inseparable Data)

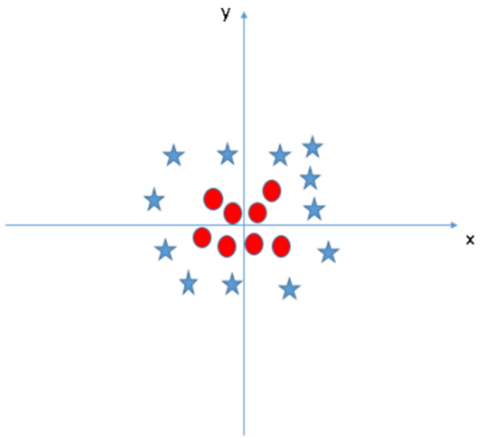
Find the hyper-plane to segregate to classes :



How does it work?

Scenario-4 (Linearly Inseparable Data)

- In this scenario, we **can't** have **linear hyperplane** between the two classes, so how does SVM classify these two classes?



- Till now, we have only looked at the linear hyper-plane.

How does it work?

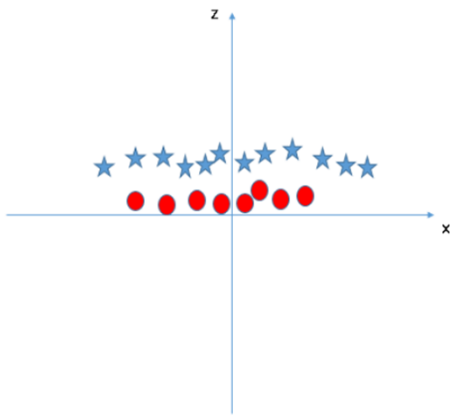
Scenario-4 (Linearly Inseparable Data)

- SVM can solve this problem. Easily!
- It solves this problem by introducing additional feature.
- Here, we will add a new feature $\mathbf{z = x^2 + y^2}$.

How does it work?

Scenario-4 (Linearly Inseparable Data)

- Now, let’s plot the data points on axis x and z:

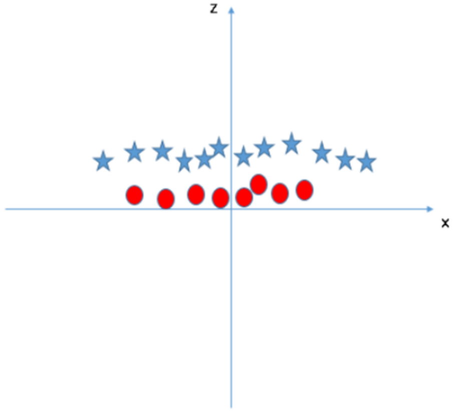


How does it work?

Scenario-4 (Linearly Inseparable Data)

In above plot, points to consider are:

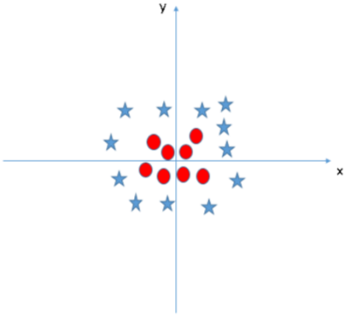
- All values for **z** would be **positive always** because z is the squared sum of both x and y.

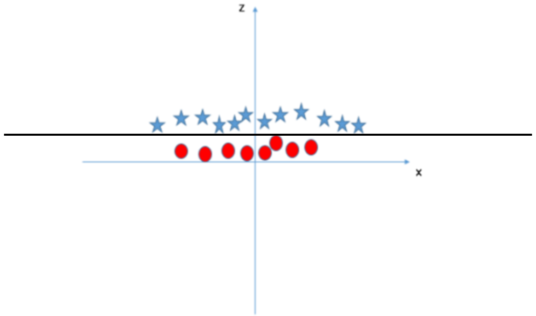


How does it work?

Scenario-4 (Linearly Inseparable Data)

- 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.





Agenda

- What is SVM?
- Ideology behind SVM
- Intuition Development
- Terminologies used in SVM
- How does it work?
- What is Kernel trick?
- Types of kernel
  - Polynomial Kernel
  - Gaussian RBF Kernel
- Support Vector Regression
- Pros and Cons of SVM
- Data preparation for SVM
- Use Case - House Prices

What is Kernel trick?

- Coming to the **major part** of the SVM for which it is most famous, the **kernel trick**.

- The kernel computes in a way such that when you **project the 2-D data into a 3-D** space, the data points **close to the center** of the data gets to the top and those **far away from center** gets into the **bottom** of the 3-D space.