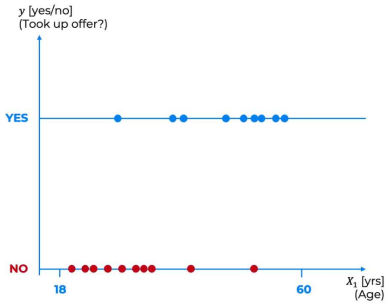
Logistic Regression

Tuesday, December 10, 2024 11:51 AM

Logistic Regression: Predict a categorical dependent variable from a number of independent variables https://www.superdatascience.com/blogs/the-ultimate-guide-to-regression-classification

- Can be a yes/no

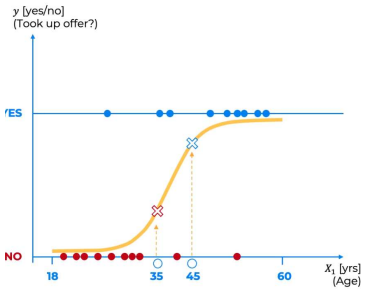


-

$$\ln \frac{p}{1-p} = b_0 + b_1 X_1$$

p is the probability
Sigmoid curve

-



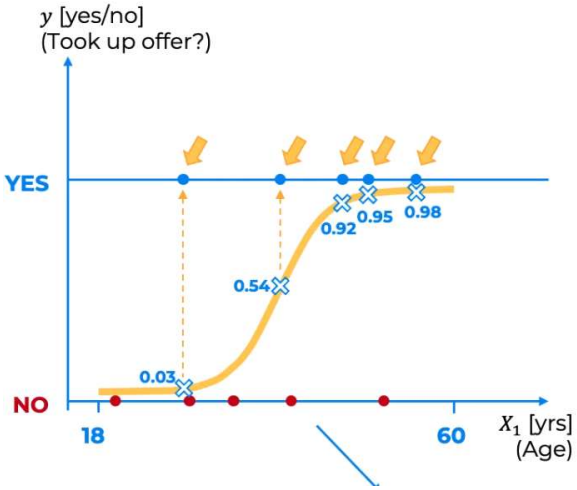
- Gives us probability or someone saying yes or no.
  - 35 is 42% chance
  - 45 is an 81% chance
  - But we want a binary outcome
- Anything above 50% is a yes, and anything below 50% is a no

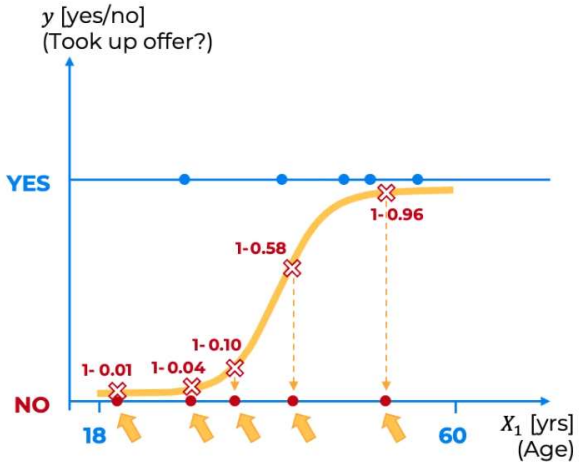
Maximum Likelihood

Tuesday, December 10, 2024 11:57 AM

How to determine if the curve fitting our data is the best curve

- Calculate the maximum likelihood
  - For each point, find the percentage of the change on the logistic regression curve





  - Likelihood is all of those numbers multiplied together

Likelihood = 0.03 x 0.54 x 0.92 x 0.95 x 0.98 x (1 – 0.01) x (1 – 0.04) x (1 – 0.10) x (1 – 0.58) x (1 – 0.96)

Likelihood = **0.00019939**

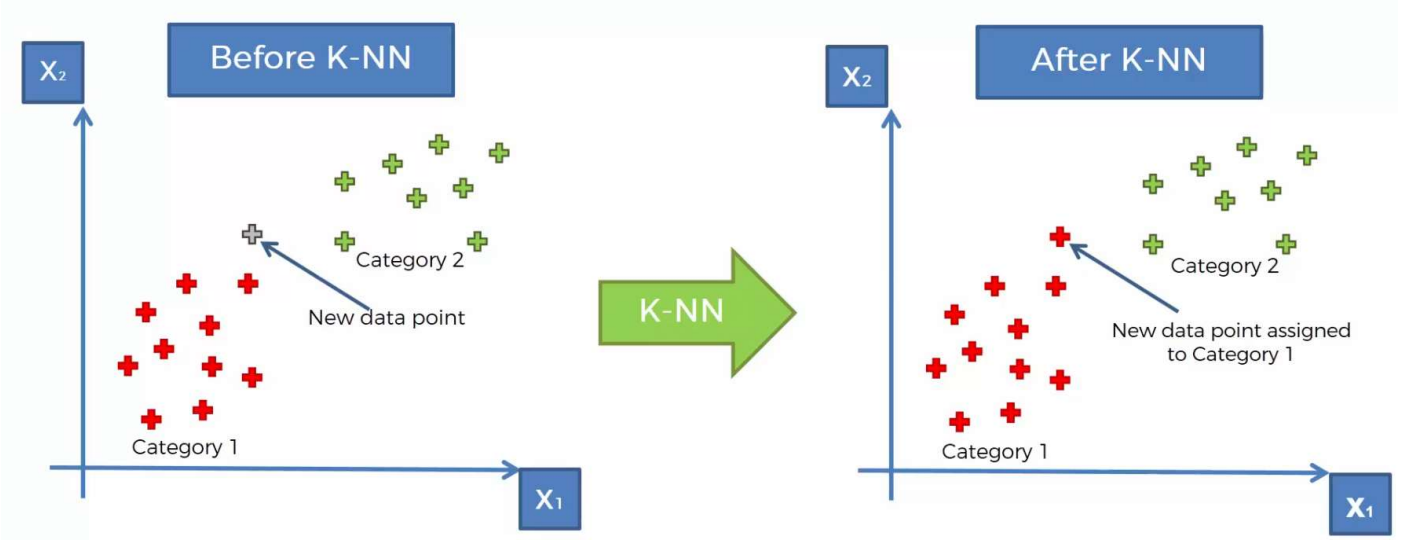
- To find the best fitting curve, we can look at all possible curves
  - As you can guess, the best fitting curve is the MAXIMUM Likelihood

K-NN

Tuesday, December 10, 2024 1:58 PM

K Nearest Neighbors Algorithm

- Given two categories defines, how can we classify where a new data point will lie
- Where the K-NN algorithm assists us



Step 1: Chose the number K of neighbors

- Most common default value is 5

Step 2: Take the K nearest neighbors of the new dat apoint, according to the Euclidean Distance
Step 3: Among these K neighbors, count the number of data points in each category
Step 4: Assign the new data point to the category where you counted the most neighbors

NOT A LINEAR CLASSIFIER

SVM

Tuesday, December 10, 2024 2:49 PM

Support Vector Machine

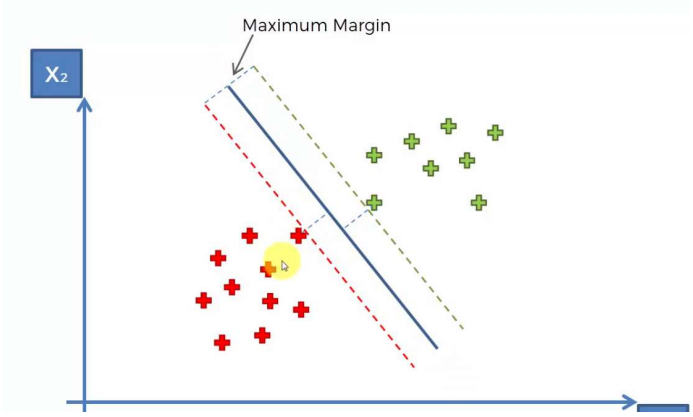
- Originally developed in 1960s and redeveloped in 1990s

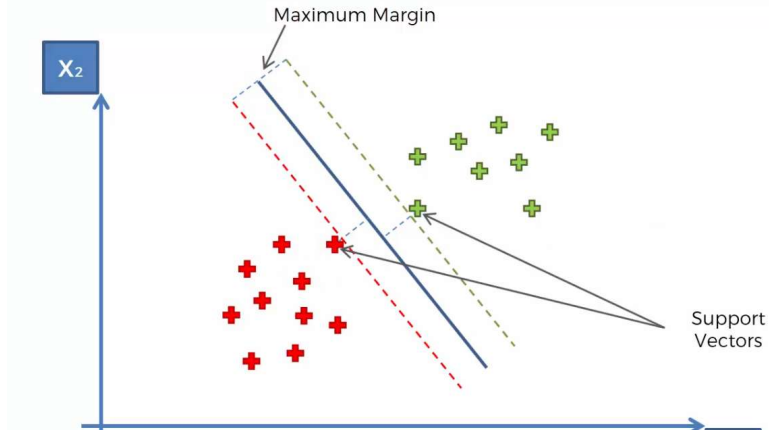
Somewhat different from other machine learning models

There a lot of different lines we can create to separate the classifications

- SVM is used to find the best line to separate the classifications

Maximum Margin

- Line that separates the two classifications, but also had the maximum margin between them
- 

  - The two points that are closes to the margin are called the **Support vectors**
  - 

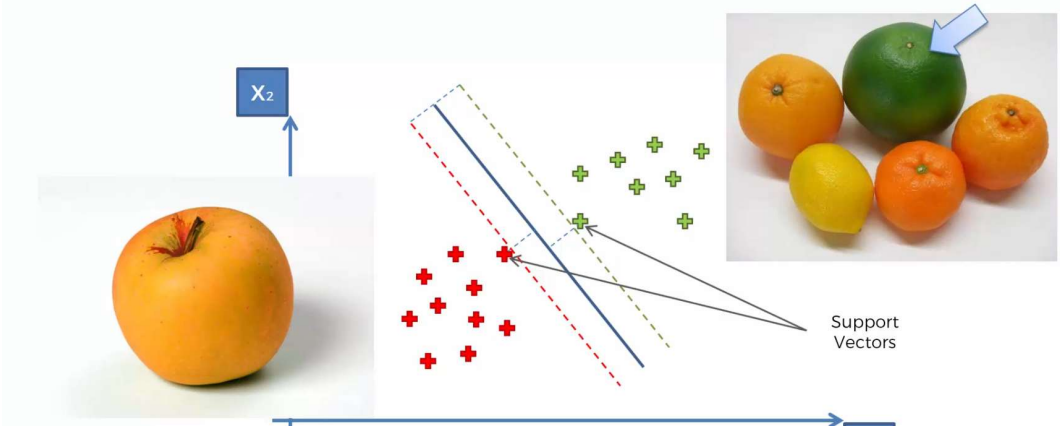
    - When in a multidimensional space, each of those points are a vector
  - Line in the middle is the Maximum Margin Hyperplane (Maximum Margin Classifier)
    - Positive Hyperplane and Negative Hyperplane

We can put a line through the chart, and find the maximum distance between the two points. This is the maximum margin

What's so special about SVMs?

- Ex. How to classify fruits between apples and oranges
  - Instead of looking at the most apple-ly looking apples and orange-ly looking oranges
  - They look at apples that look a lot like oranges, oranges that look a lot like apples
    - These are the support vectors

- The SVM is kinda risky, because it looks at an extreme case and uses that for analysis
-



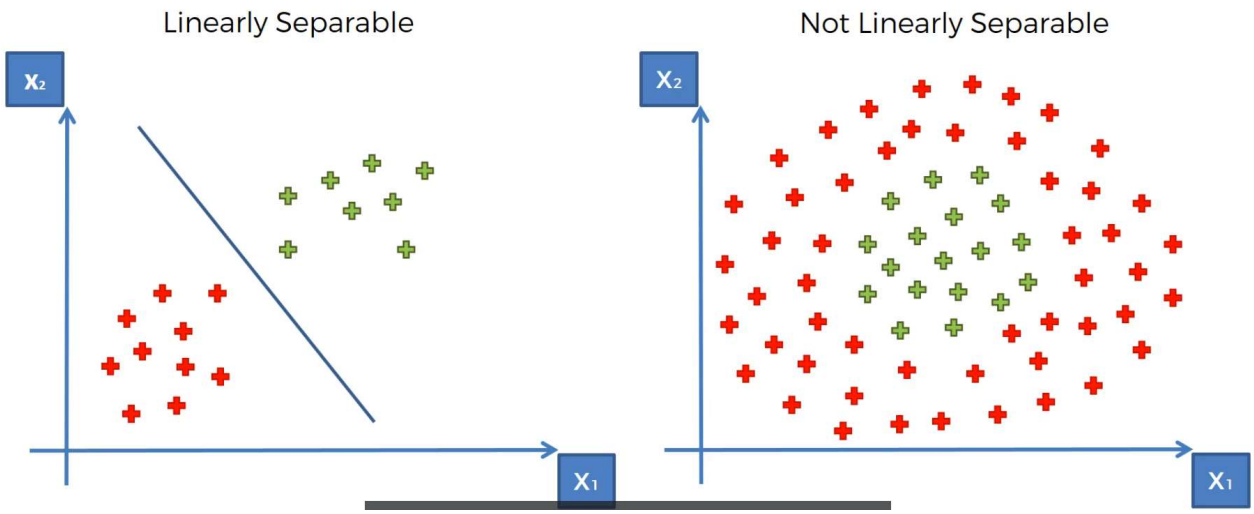
Kernel SVM

Tuesday, December 10, 2024 7:40 PM

Kernel Support Vector Machine

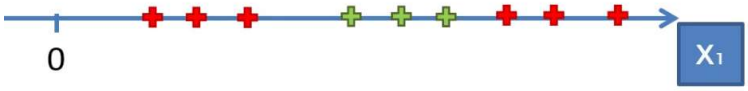
- When there are no direct boundaries
- Where you cannot separate the points like the SVM would do

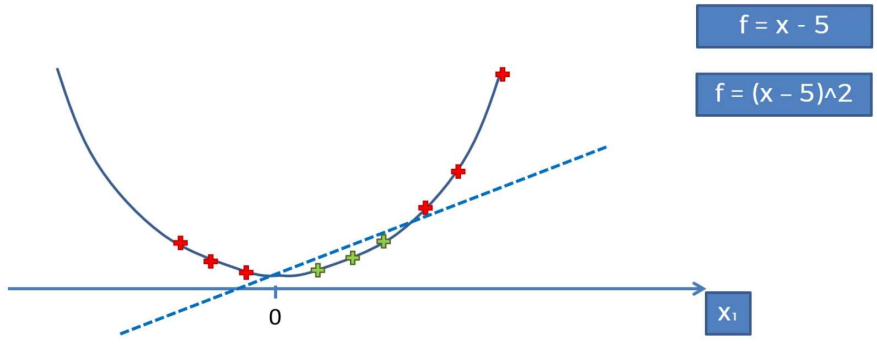
This is because the points are not linearly separable



- Assumption that data is separable

A Higher-Dimensional Space

- One Dimensional Data Set
  - Points are non-linearly separable
  - 

  - We are going to make a higher dimension
    - First move the data back a certain amount (lets say 5)
      - F = x - 5
    - Now we want to square that
      - F = (x-5)^2
      - This will create a parabola thing on our graph
    - We can then plot the points onto the parabola
    - Now it is linearly separable
  - 

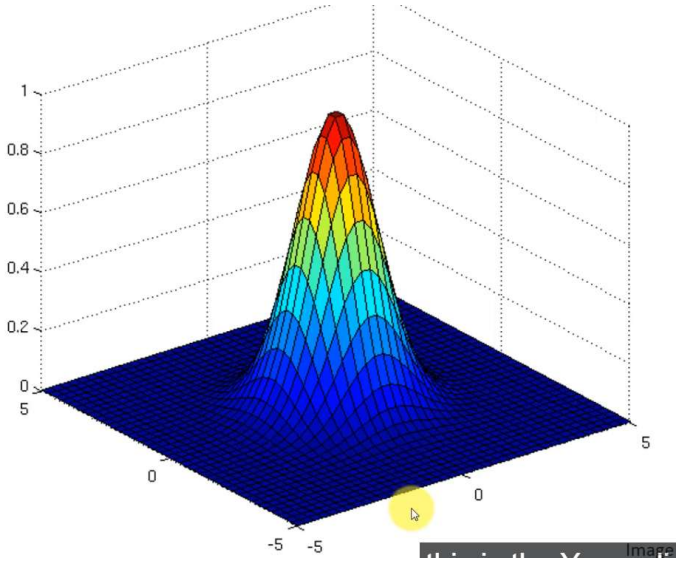
Mapping to a higher dimensional space can be highly compute-intensive

- Explore a different approach (kernel trick)

Kernel Trick **(Gaussian RBF Kernel)**

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$

-



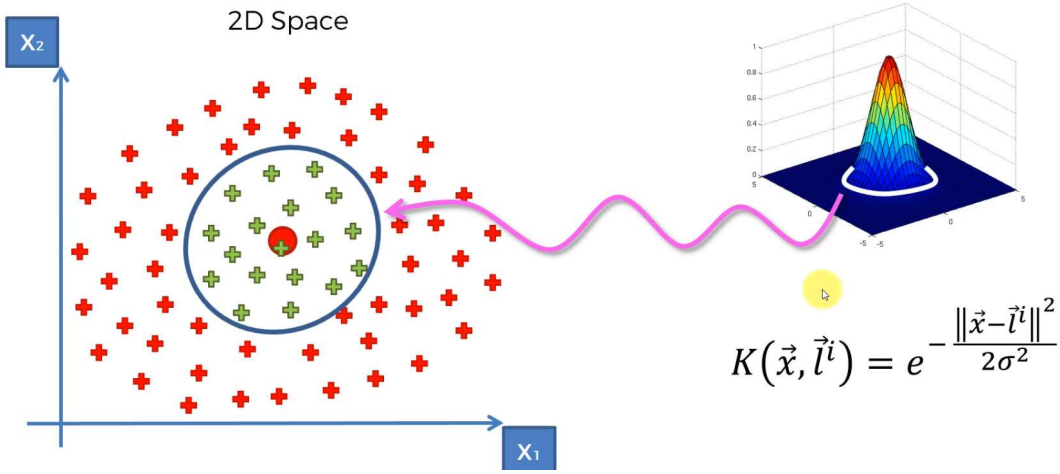
- The landmark is the middle of the plane
- The vertical represents the result of the formula

- When you are far away from the landmark (zero) then you will get a very small number from the equation
- If you are closer to the landmark then you will get a larger number

We are going to use this kernel function to build our data

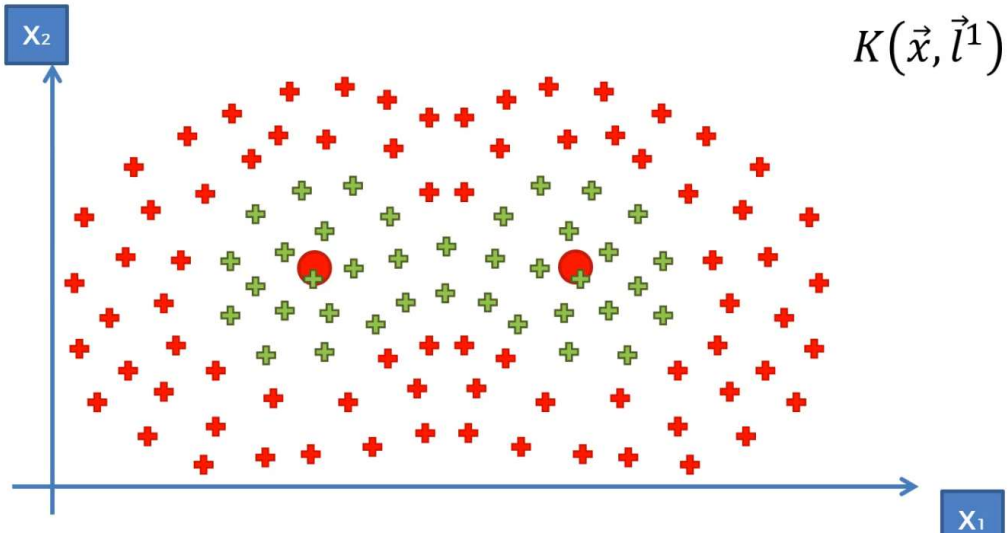
- Take the landmark and put it in the middle of the data

-



- Sigmas role defines the circumference of the circle
  - We need to find the right sigma

You can also take two kernel functions and add them up

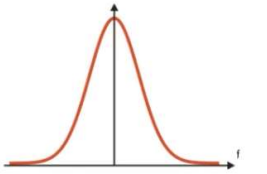


$$K(\vec{x}, \vec{l}^1) + K(\vec{x}, \vec{l}^2)$$

(Simplified Formula)

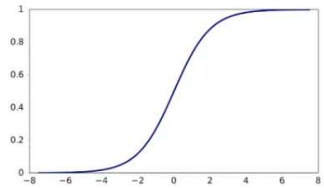
We can make this non-linear boundary without needing to go into a higher dimensional space

Types of Kernel Functions



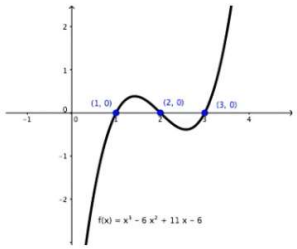
Gaussian RBF Kernel

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$



Sigmoid Kernel

$$K(X, Y) = \tanh(\gamma \cdot X^T Y + r)$$



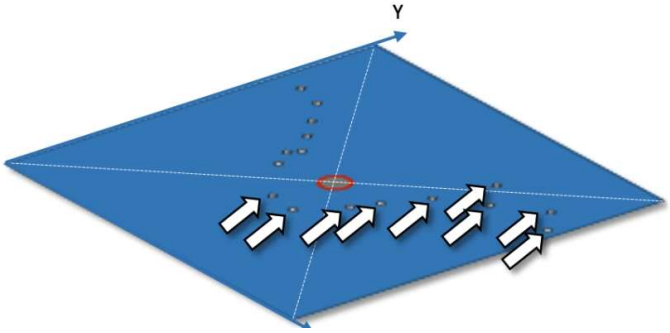
Polynomial Kernel

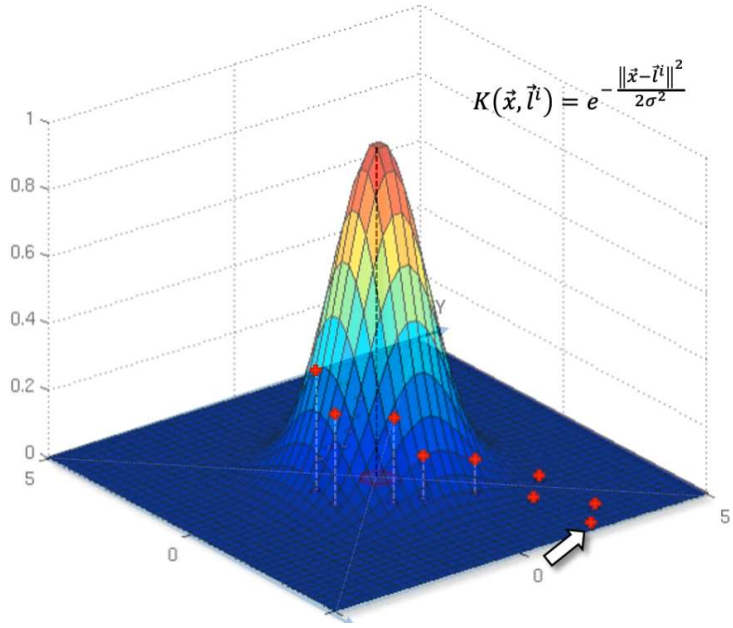
$$K(X, Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$$

Non Linear SVR

Wednesday, December 11, 2024 1:36 PM

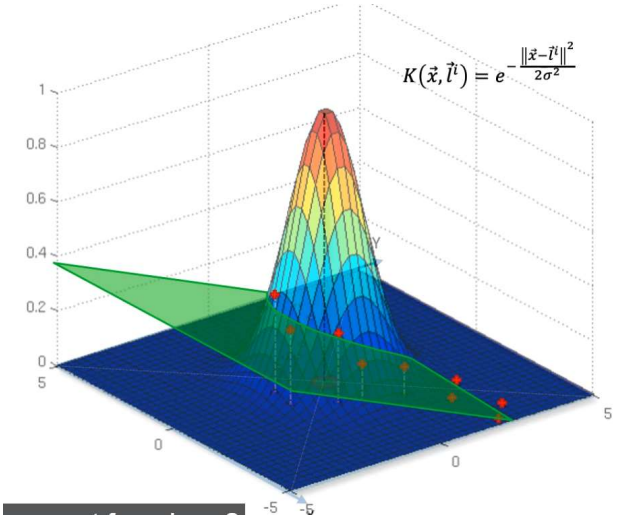
How to we build a non-linear SVR that can fit a graph like this



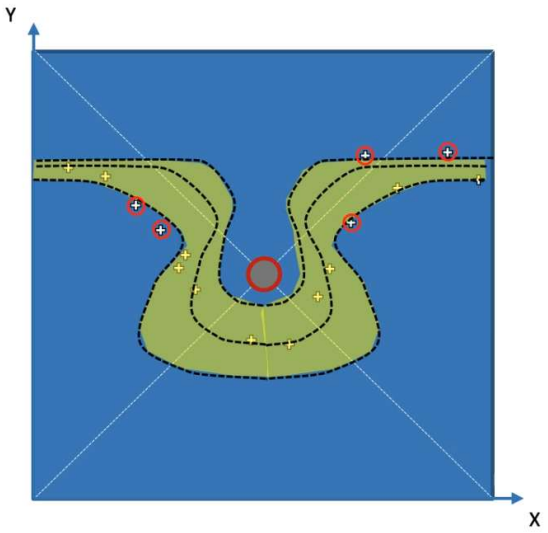


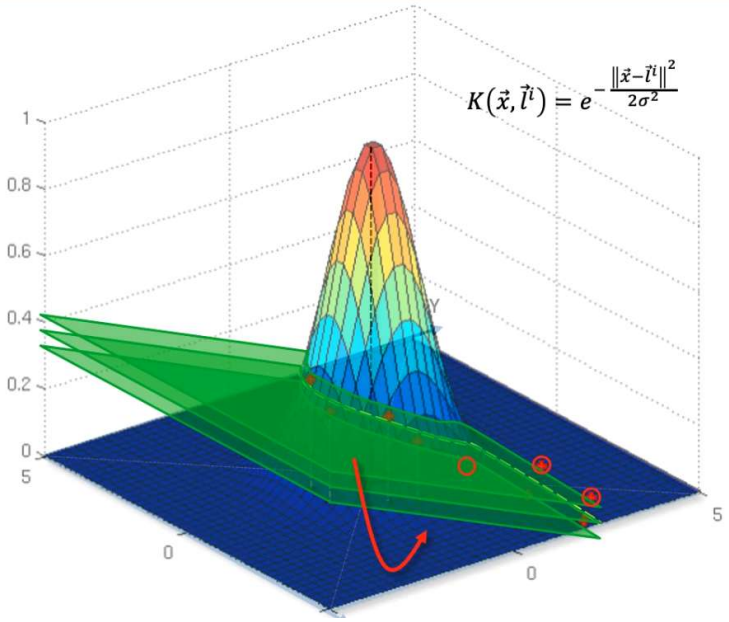
Now that we are in three dimensions we can run a linear model

- Hyperplane



- Where does the hyperplane intersect the RBF
- This is the line we can apply in the 2D dimension





In reality we don't have to go into the third dimension to get the projections

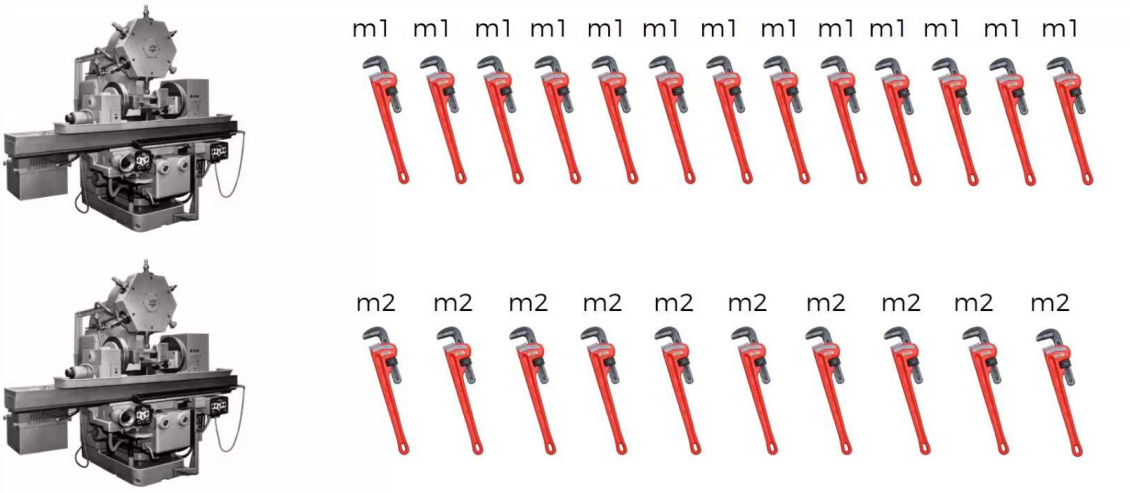
Bayes Theorem

Wednesday, December 11, 2024 2:00 PM

For this example we are going to be talking about a wrench (or a spanner which is what he is calling it)

Two machines that produce spanners, both with different characteristics but they produce the same amount of spanners

- The spanners are marked/tagged so we know which machine they came from



Now they are in a pile and we need to go through the defective spanner

- What is the probability that a random spanner from machine 2 is defective

Bayes Theorem

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Machine 1 produces 30 wrenches per hour
Machine 2 produces 20 wrenches per hour
We can see from all produced parts, 1% are defective
Of all defective parts 50% came from machine 1, and 50% came from machine 2
Q:

- What is the probability that a part produces by machine 2 is defective?

If we pick up a wrench in a pile there is a .6 chance of it being from machine 1

- P(mach1) = 30/50 = 0.6

If we pick up a wrench from a pile there is a .4 chance of it being from machine 2

- P(mach2) = 20/50 = 0.4

P(Defect) = .01 or 1%

P(Mach1 | Defect) = 50%

- Likelihood of a part from machine 1 given that it is defective

P(Mach2 | Defect) = 50%

Q:

- P(Defect | Mach2)
- What is the probability that is a part is defective given its from machine 2

$$P(Defect \mid Mach2) = \frac{P(Mach2 \mid Defect) * P(Defect)}{P(Mach2)}$$

$$P(Defect \mid Mach2) = \frac{.5 * .01}{0.4} = 0.0125 = 1.25\%$$

Example:

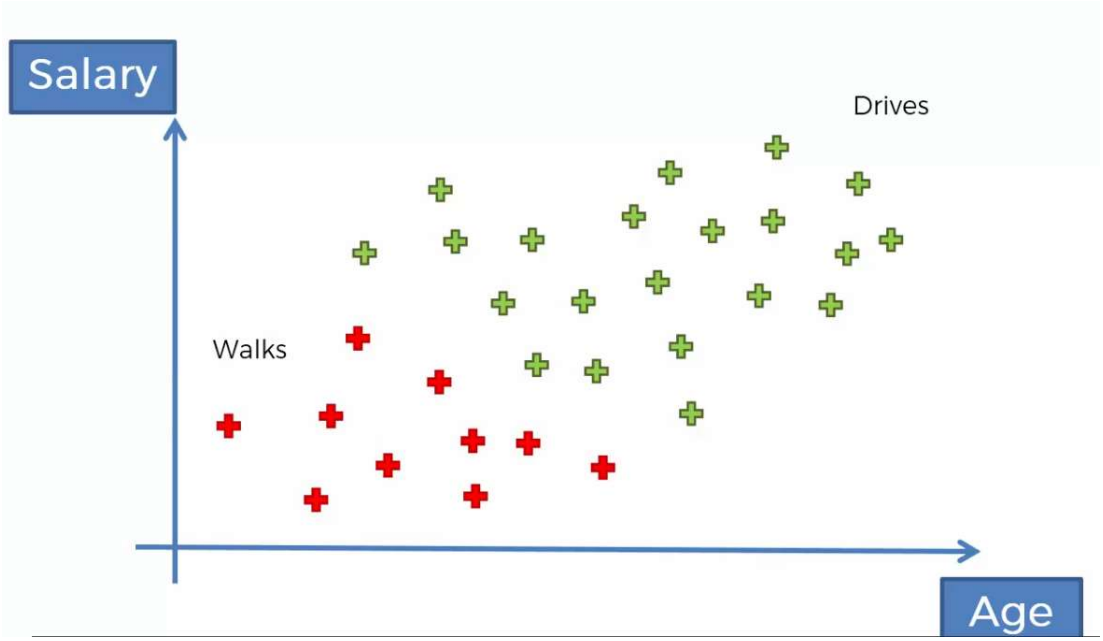
- 1000 wrenches
- 400 came from Mach2
- 1% have a defect = 10
- of them 50% came from Mach2 = 5
- % defective parts from Mach2 = 5/400 = 1.25%

- This is the exact same process as the equation

Naïve Bayes Classifier

- How to apply it to create a machine learning algorithm

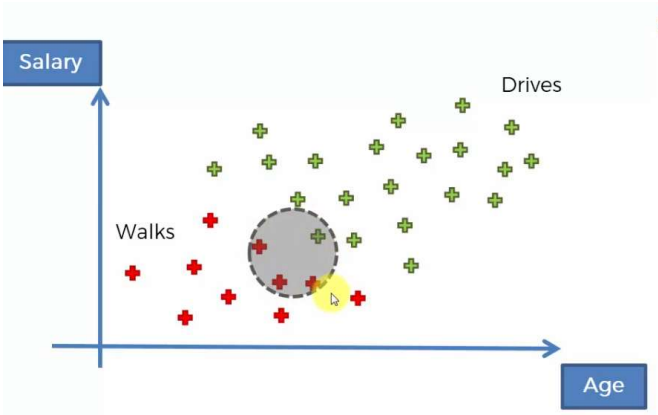
-



Plan of Attack

- Apply the Bayes Theorem Twice
  - Find the probability that a person walks given its features
    - In our example age and salary
    - $$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

      #4 Posterior Probability; #3 Likelihood; #1 Prior Probability; #2 Marginal Likelihood
  - Find the probability that a person will drive given its features
    - $$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

      #4 Posterior Probability; #3 Likelihood; #1 Prior Probability; #2 Marginal Likelihood
- P(Walks | X) vs P(Drives | X)
- 

- Anyone within the circle is deemed to be similar to X
- P(X) = Num of Similar Observation / Total Observations
  - P(X) = 4 / 30
- P(X | Walks) = Num of Similar Observations who walk / Total Number of Walkers
  - P(X | Walks) = 3 / 10

Now do this again with step 2

P(Walks | X) vs P(Drives | X)
- We calculated 0.75 vs 0.25
- Therefore P(Walks | X) > P(Drives | X)

Naïve Bayes
- Why 'Naïve'
  - Requires independence assumptions
  - These assumptions may not be correct. This is why it is naïve

- P(X)
    - Likelihood that a randomly selected point in the data set will have similar features of X
    - $P(X) = \frac{Number\ of\ similar\ observations}{Total\ Observations}$
- More than 2 classes
    - When we have 2 classes, the probability will always add up to 1.0
    - So when we have more than 2 classes each of the probabilities will add up to 1.0

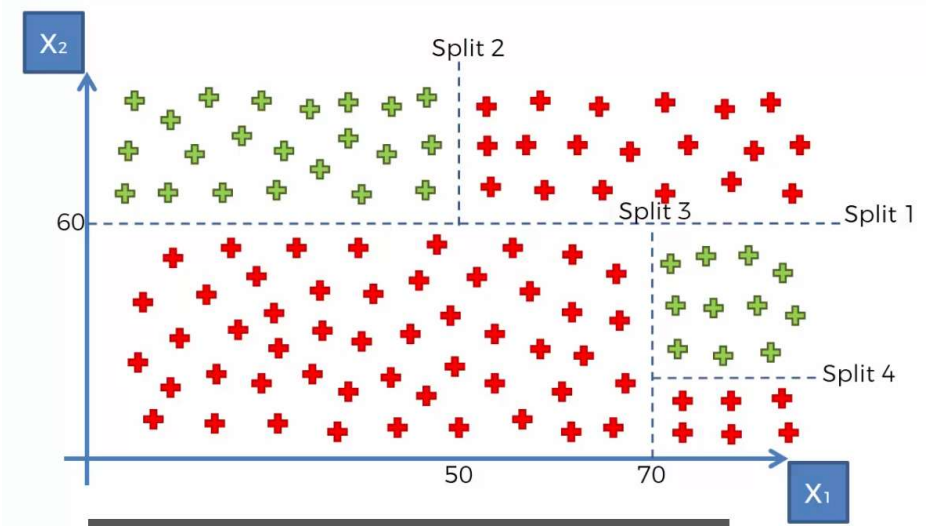
Decision Trees Classification

Wednesday, December 11, 2024 3:43 PM

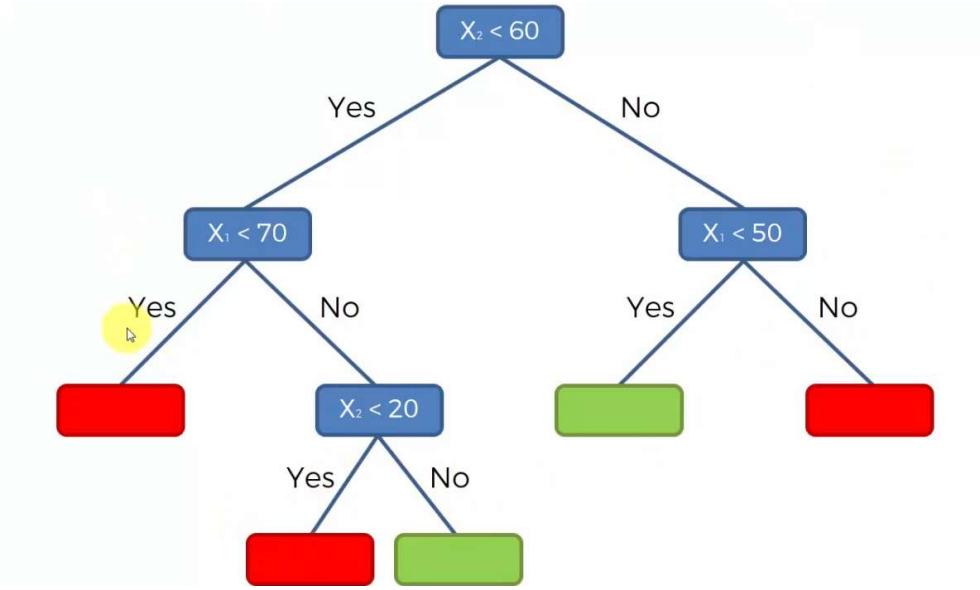
CART

- Classification and Regression Trees

Cut up the graph into multiple parts

- 

- How does the algorithm know where to put the split
  - The algorithm tries to maximize the number of points within a category
  - Minimize Entropy



If the decision tree is very large, sometimes the algorithm will get to a certain node, and find the probability of either green or red

Decision Trees are old and started to die off

- Until they were reborn with upgrades
  - Random Forest
  - Gradient Boosting

Random Forest Classification

Wednesday, December 11, 2024 6:02 PM

Ensemble Learning

- Putting multiple machine learning algorithms together
    - Leverages multiple machine learning algorithms

Run an algorithm multiple times

STEP 1: Pick at random K data points from the training set
STEP 2: Build the Decision Tree associated to these K data points
STEP 3: Choose the number Ntrees off trees you want to build and repeat STEPS 1 and 2
STEP 4: For a new data point, make each one of your Ntrees predict the category to which the data points belongs, and assign the new data point to the category that wins the majority vote

Start with one tree, and then another, and then another

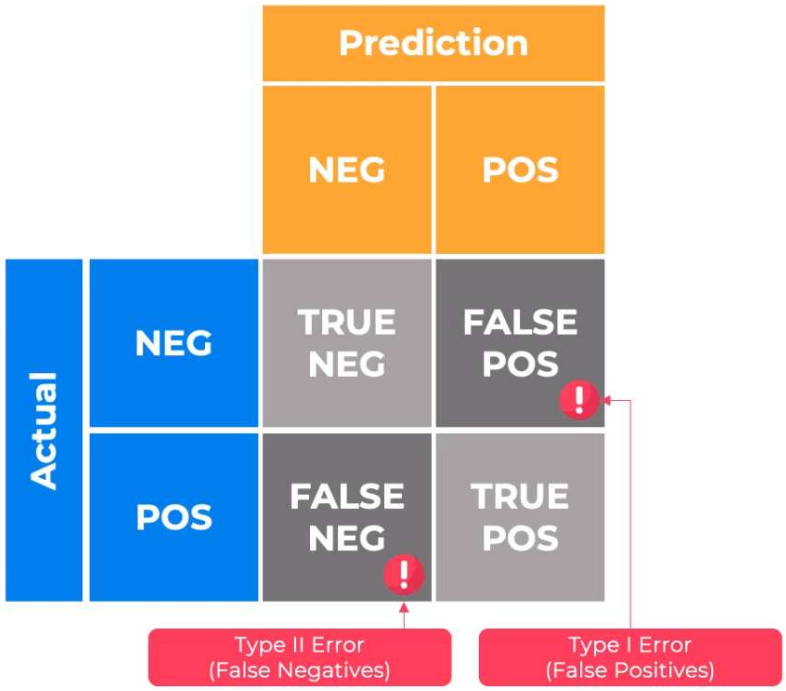
- Each tree is built off a random subset of data
- Leveraging the power of numbers

Confusion Matrix & Accuracy

Thursday, December 12, 2024 1:50 PM

Type 1 Error
- False Positive

Type 2 Error
- False Negative



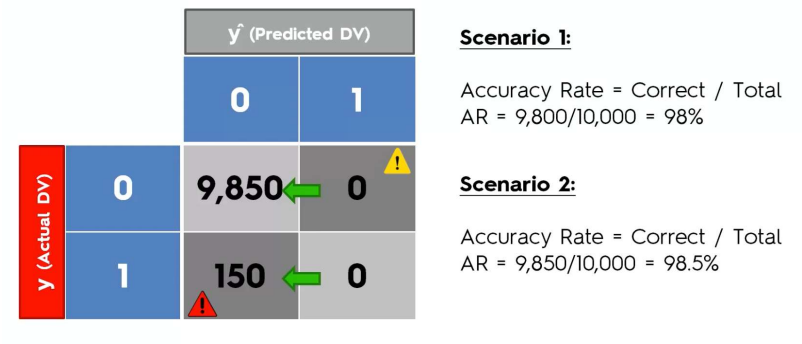
Accuracy rate and Error Rate

$$AR = \frac{Correct}{Total} = \frac{TN + TP}{Total}$$

$$ER = \frac{Incorrect}{Total} = \frac{FP + FN}{Total}$$

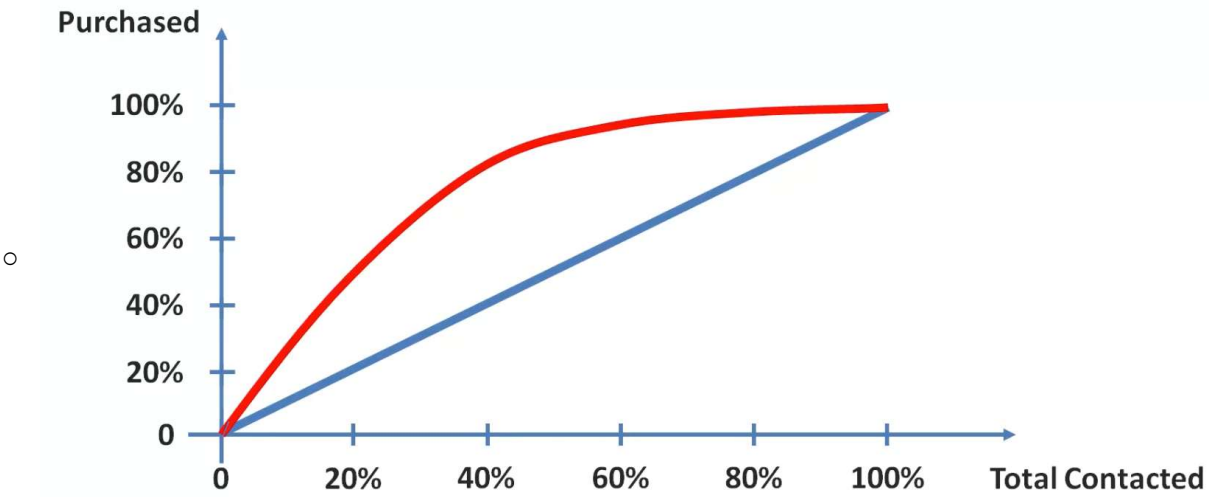
False Positives and False Negatives
- We agreed earlier anything below the 50% line will be one prediction while above 50% is the other prediction
- This may not always be the case
  - You can think of it as type 1 can be a warning (False Positive)
  - You can think of it as type 2 as true danger (False Negative)

Accuracy Paradox
- When you have such a high accuracy rate, and you abandon the model, you have a higher accuracy rate without the model

- 

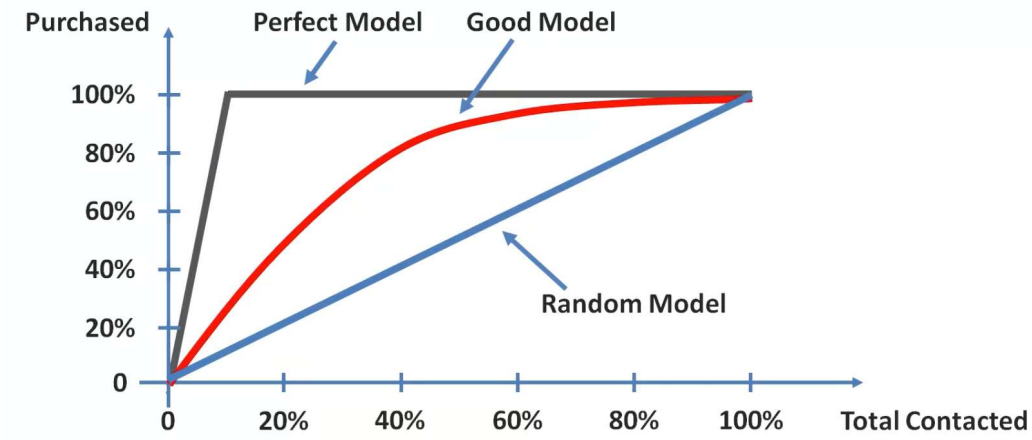
- Don't always base judgement based on accuracy rate

CAP Curve

- Cumulative Accuracy Profile
- Instead of sending newsletters randomly, why not send them to customers who have a record of buying from the newsletter
    - This will give us a better result of customers buying
    - 

    - Red is the better result, while blue was sending to everyone
- How much gain you get while changing your model

CAP curve analysis

- 

- What can we derive from this?
    - Calculate the perfect ratio: Area under the perfect model and above random model (Ap)

    - Calculate the area under the red line (Ar)
    - AR = Ar / Ap

- We can also look at the 50% on the good model
    - This is about 90% on this mode
- X < 60% Terrible
- 60% < X < 70% Poor
- 70% < X < 80% Good
- 80% < X < 90% Very Good
- 90% < X < 100% Too Good

ROC

- Receiver Operating Characteristic
- NOT THE SAME THING AS CAP