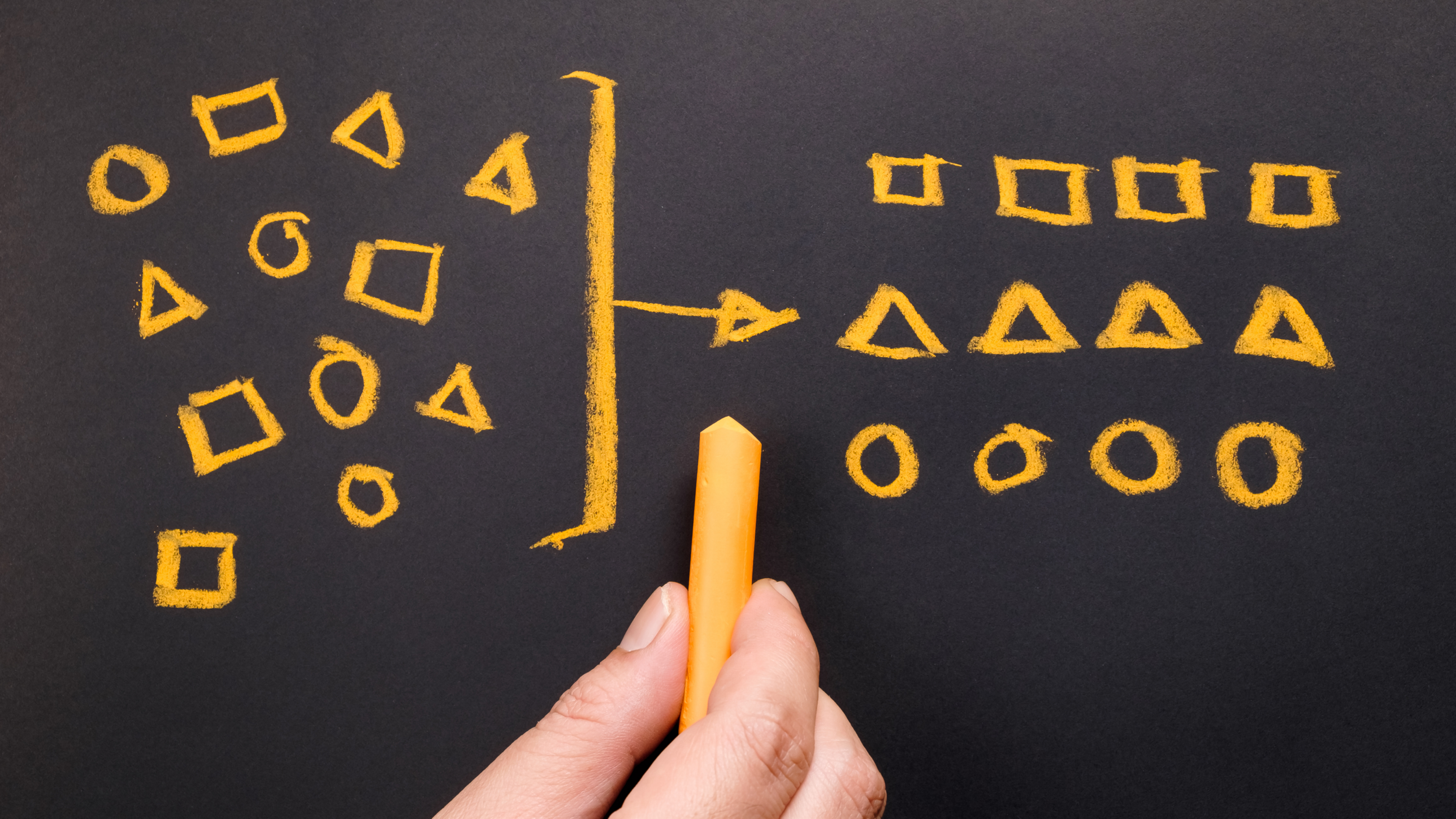
Classifications Algorithms

Xavier Morera

Helping developers understand and work with data

@xmorera www.xaviermorera.com / www.bigdatainc.org



When performing classification, we want to identify an algorithm that separates the data so that every entry, that is data sample, can be correctly categorized

Logistic Regression

Logistic Regression

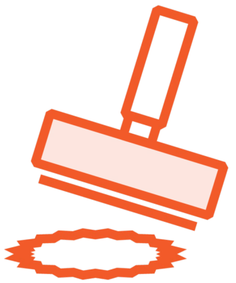
True or false

Cancer or not-cancer

Guilty or innocent

Logistic Regression

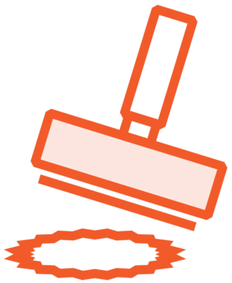
[0,1]



0.75

Logistic Regression

[0,1]



0.75

Logistic regression is not a regression problem

- However, it uses the same algorithm as part of its main algorithm
- Activates the linear regression function to create a probability

Probability lies within 0 and 1

- 0.5 used as boundary to determine the class

$$h_\theta(X) = w_0 + \sum_{j=1}^{m} X_{ij} w_i$$

◂ **Estimate the linear function**

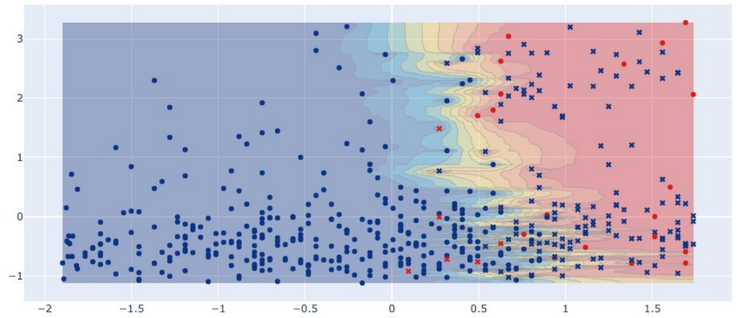
$$\hat{y}_i = \omega(h_\theta(X)) = \frac{1}{1 + e^{h_\theta(X)}}$$

◂ **Activate using the following function**

$$J(\theta) = \frac{1}{m} \left(\sum_{i=1}^{m} -y^{(i)} \cdot log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \cdot log(1 - h_\theta(x^{(i)})) \right)$$

◂ **Cost function for the logistic regression algorithm**

Logistic Regression



Use logistic regression for binary classification

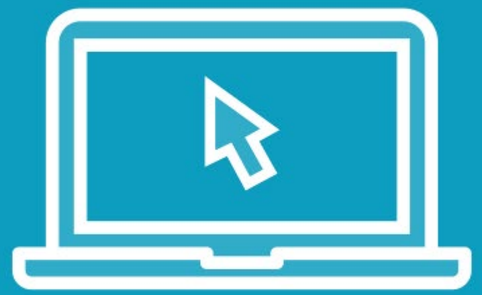
- Can also be used for multi-class classification

Input data must be numerical

Entry features must have a normal distribution for best results

Assumes all variables are independent

Demo



Logistic Regression

Naive Bayes

Naive Bayes

Family of simple probabilistic classifiers based on applying Bayes theorem with strong (naive) independence assumptions between features

How often does the Ace of Spades appear in a deck of cards?

What is the probability of getting the Ace of Spades after we drew the Ace of Hearts from the deck?

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

- Probability that an event A will happen if B occurs first

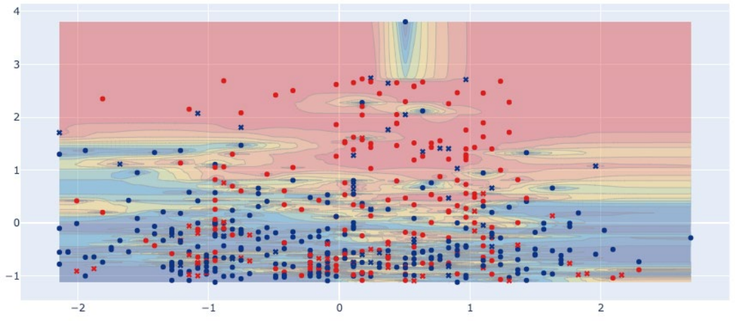
$$P(AceOfHearts|AceofSpades) = \frac{0.00037 \cdot 0.019}{0.019} = 0.00037$$

- Probability of drawing the Ace of Hearts right after the Ace of Spades

$$P(class|data) = \frac{P(data|class) \cdot P(class)}{P(data)}$$

- Reuse the formula to estimate the probability that a class belongs to a data sample

Naive Bayes



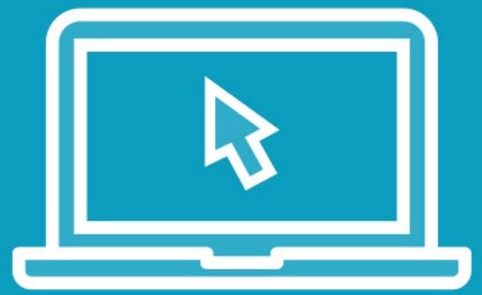
Output data must be categorical or labels

Input data must be numeric and contiguous

Input variables must follow a distribution according to the algorithm

Variables must be independent of each other

Demo



Naive Bayes

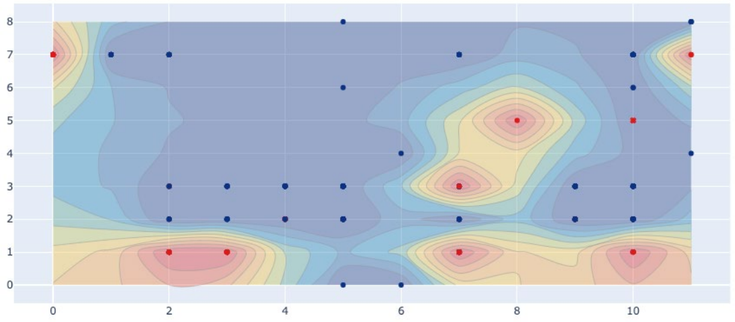
Support Vector Machines

Support Vector Machine

Algorithm that separates data using a line or a hyperplane (many dimensions)

One-vs-all classifier, which works well for binary classification

Support Vector Machine



When performing classification, a set of elements are used to create the best separation

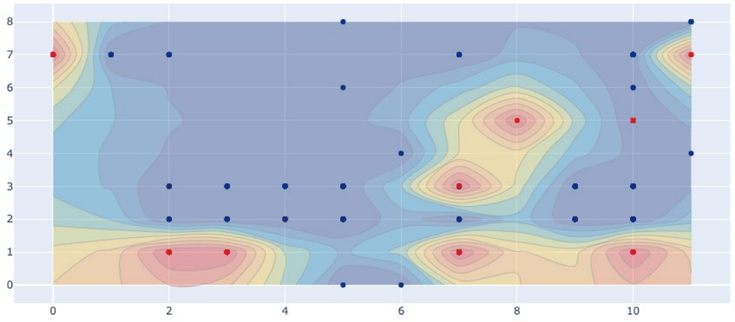
- These are known as support vectors

Separating data linearly is not always practical

Can define how to separate data when performing classification

- Linear, polynomial, or radial

Support Vector Machine



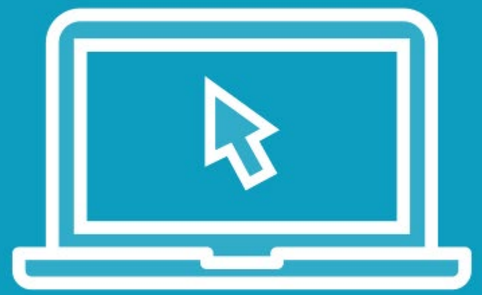
Advantages

- Resistance to outliers
- Allows for different degrees of freedom to fit the training data

Recommendations

- Use SVM as a binary classifier
- Use dimensionality reduction algorithms
- Use grid search with cross-validation
- Test out several kernels and parameter values to test fit and data separability

Demo



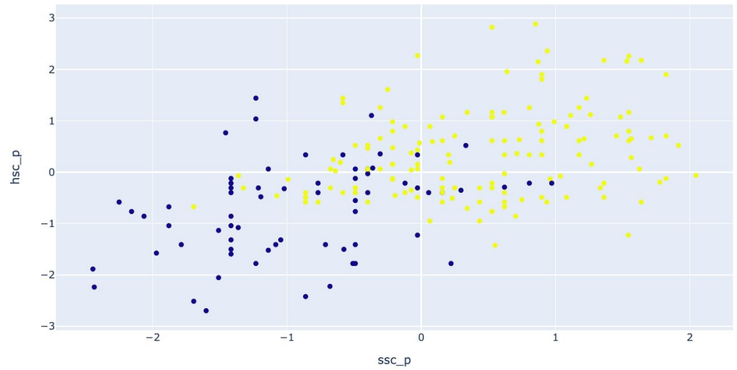
Support Vector Machine

K Nearest Neighbors

All data must be used on every prediction

KNN's drawback

K Nearest Neighbors



KNN uses a distance algorithm to analyze elements

- Euclidean or Manhattan

K = 3 means it analyzes each element with the three closest neighbors

Exponential increase of analysis required when the number of elements grows

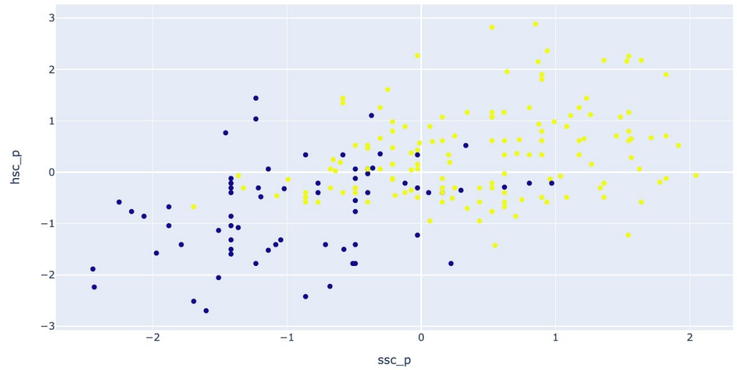
$$\vec{d}_{\text{Euclidean}}\left(\mathbf{a}, \mathbf{b}\right) = \sqrt{\sum_{i=i}^{M}(a_i - b_i)^2}$$

◂ **Euclidean distance**

$$\vec{d}\left(\mathbf{a}, \mathbf{b}\right) = \sum_{i=1}^{M} \mid a_i - b_i \mid$$

◂ **Manhattan distance**

K Nearest Neighbors



KNN is non-parametric

Works well for multi-class settings

Can be used for classification or regression

Possible to select the distance algorithm

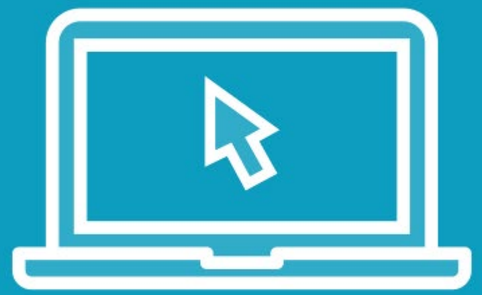
Slow with large datasets

No recipe for optimal K

Very sensitive to outliers

Does not perform well with imbalanced datasets

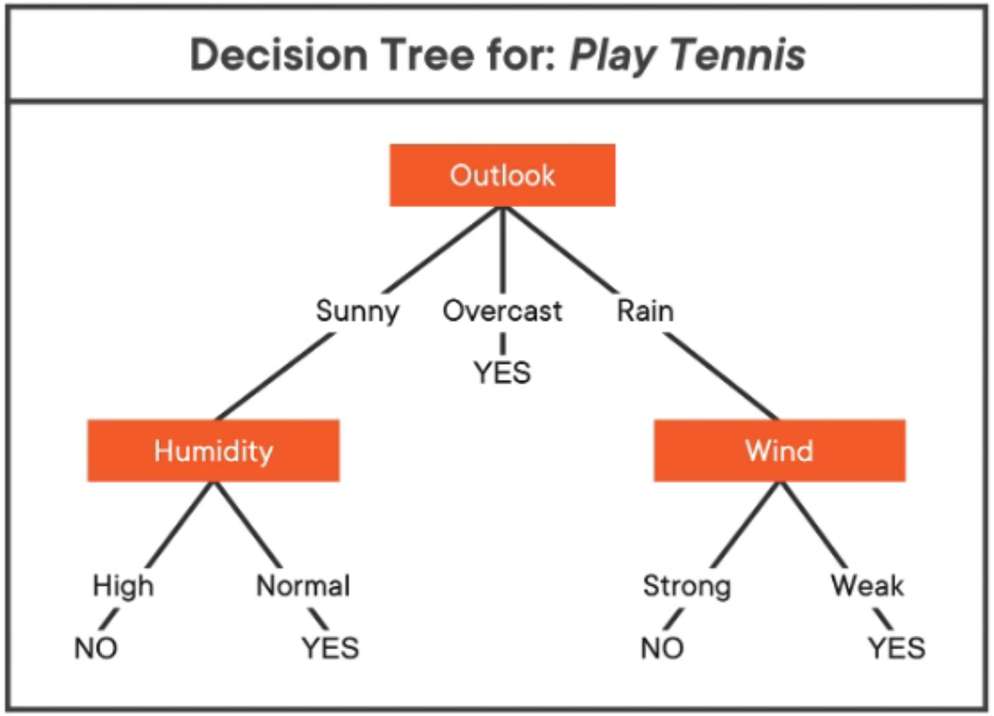
Demo



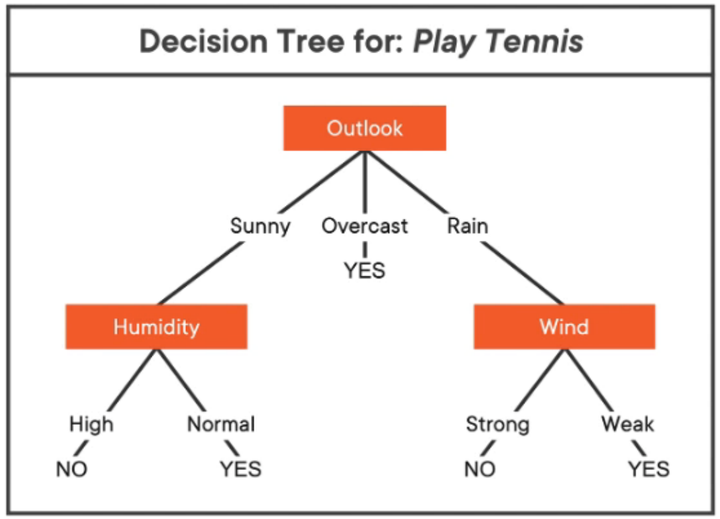
K Nearest Neighbors

Decision Trees and Random Forests

Decision Tree



Decision Trees



How to determine split condition?

- Gini coefficient or information entropy

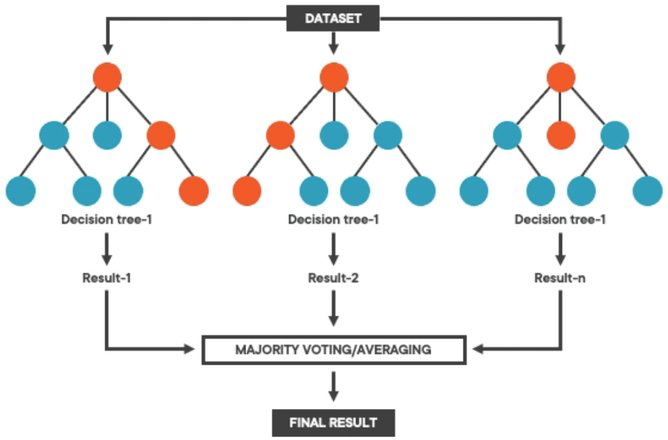
Which element shows lowest value in the formula

- Highest inequality

This element is considered the split condition

A single tree can be transformed into multiple trees

Random Forests



Random forest is a decision tree

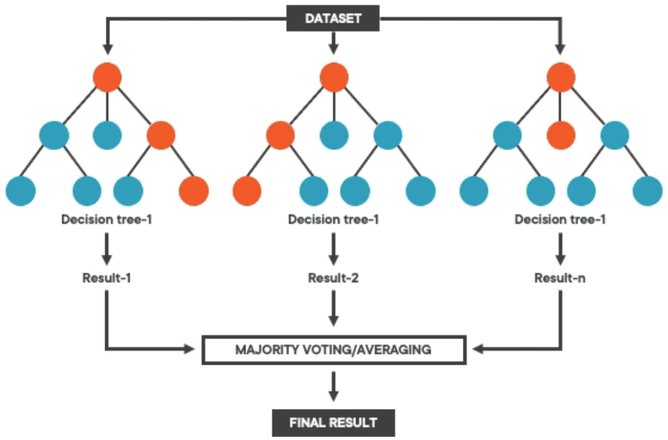
- That has been generated n-times

Results are gathered from all decision trees

Each decision tree "votes" to get the results

Makes the random forest a strong classifier

Decision Trees and Random Forests



When to use Gini or Entropy?

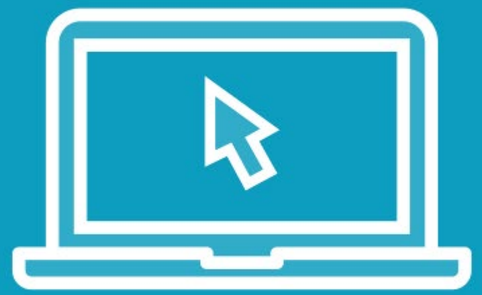
- Gini usually recommended because it is computationally inexpensive

Using a random forest on a small dataset might be overkill

Random forests take more time to train

Both decision trees and random forests can be used for regression and classification

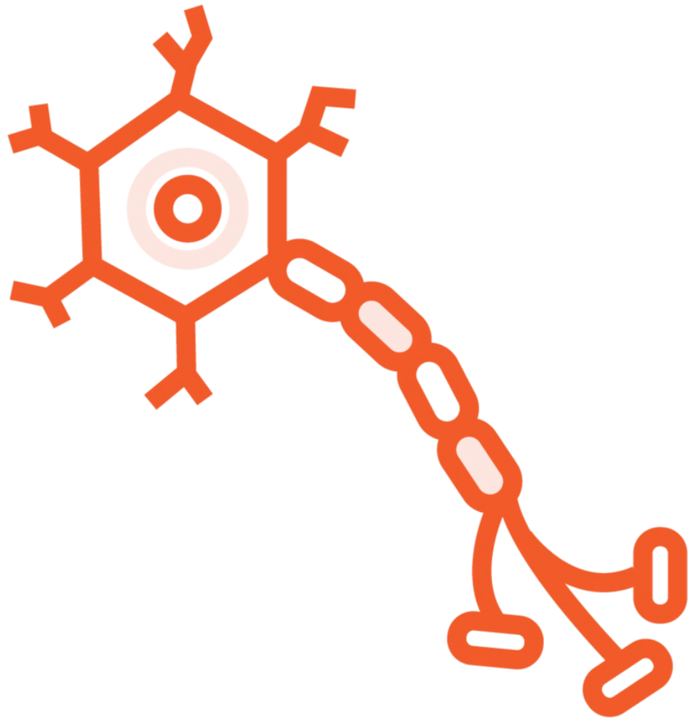
Demo



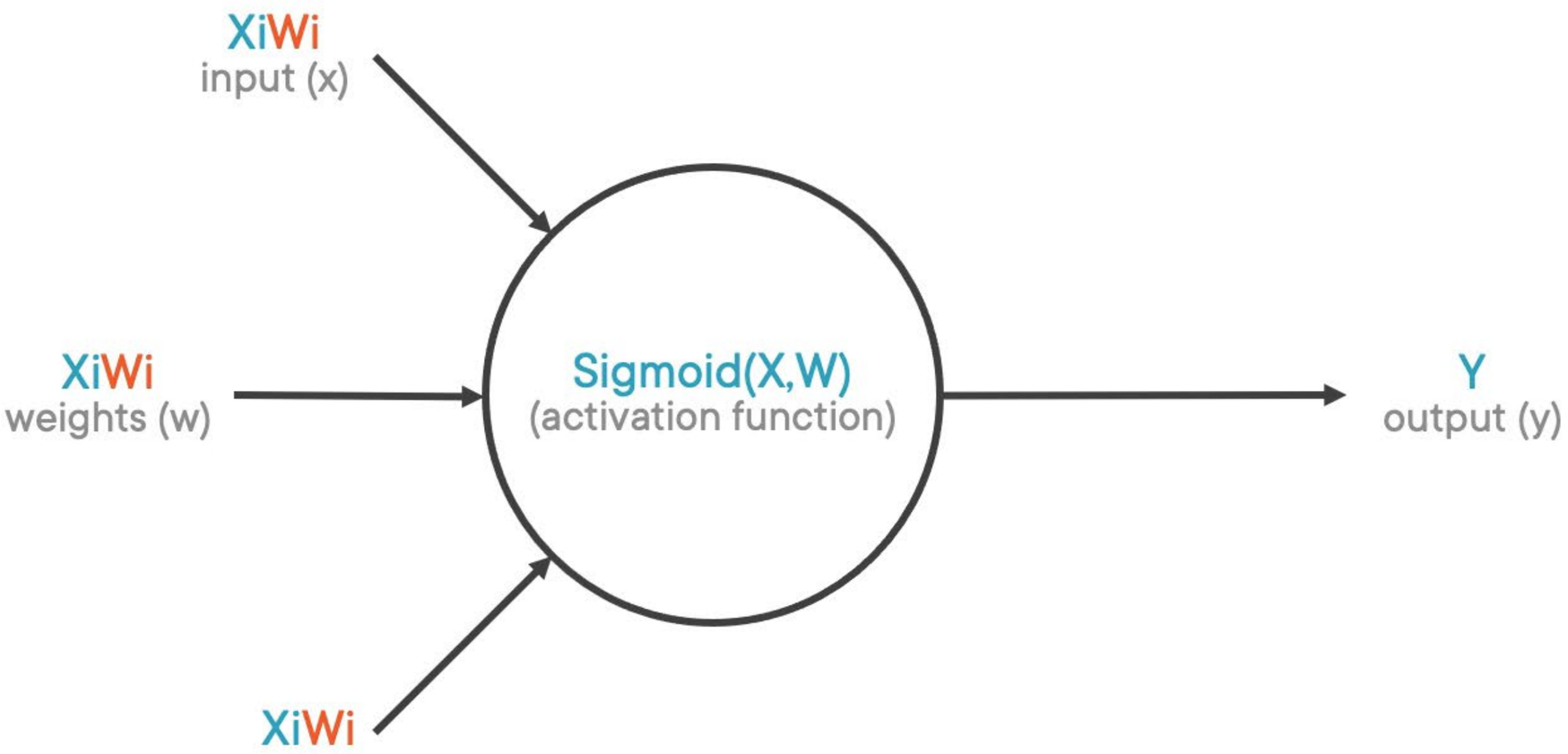
Decision Trees and Random Forests

Neural Networks

Neural Networks



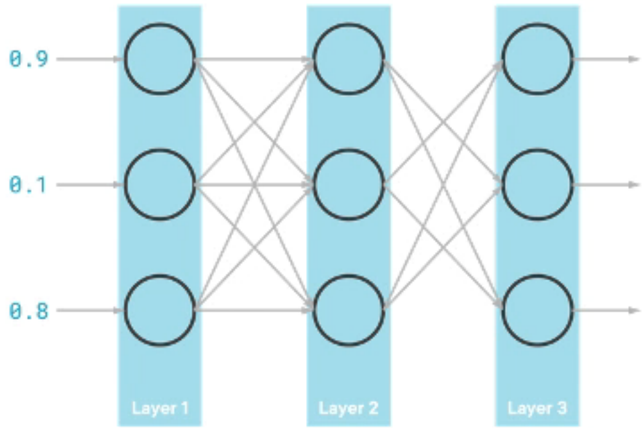
Neural Networks



Deep Learning

$$oL2 = \begin{bmatrix} 0.761 \\ 0.603 \\ 0.650 \end{bmatrix}$$

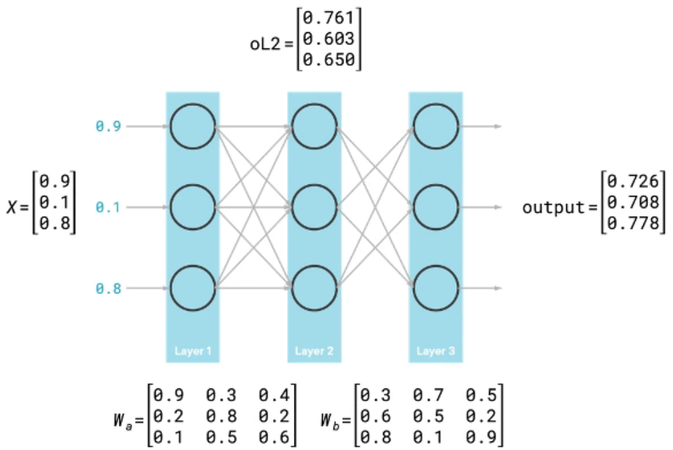
$$X = \begin{bmatrix} 0.9 \\ 0.1 \\ 0.8 \end{bmatrix}$$



$$output = \begin{bmatrix} 0.726 \\ 0.708 \\ 0.778 \end{bmatrix}$$

$$W_a = \begin{bmatrix} 0.9 & 0.3 & 0.4 \\ 0.2 & 0.8 & 0.2 \\ 0.1 & 0.5 & 0.6 \end{bmatrix} \quad W_b = \begin{bmatrix} 0.3 & 0.7 & 0.5 \\ 0.6 & 0.5 & 0.2 \\ 0.8 & 0.1 & 0.9 \end{bmatrix}$$

Deep Learning Architectures



Neural networks are parametric

No recipe on how many layers

- Use trial and error

Algorithms used by Deep Learning neural networks

- Forward propagation and back propagation

Back propagation adjusts weights to reduce error

Epoch
000,000

Learning rate: 0.03

Activation: Tanh

Regularization: None

Regularization rate: 0

Problem type: Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

X_1X_2

$\sin(X_1)$

$\sin(X_2)$

2 HIDDEN LAYERS

4 neurons

2 neurons

*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

*This is the output from one **neuron**. Hover to see it larger.*

OUTPUT

Test loss 0.510

Training loss 0.502

-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6

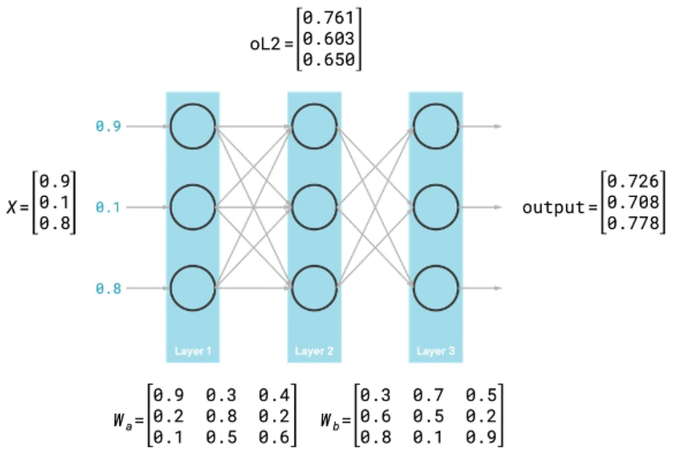
Colors shows data, neuron and weight values.

-1 0 1

Show test data

Discretize output

Neural Networks: Considerations



No recipe for the number of layers

Can use pre-trained neural networks

Creating a deep learning network is easy

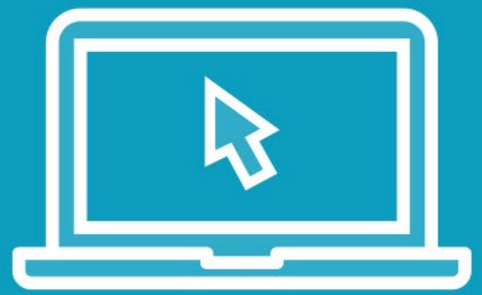
- Tuning it is not

Normalized values highly recommended

Requires a lot of data

Takes a lot of time to train

Demo



Neural Networks

Convolutional Neural Networks

kaggle

Search

Create

Home

Competitions

Datasets

Code

Discussions

Courses

More

Recently Viewed

Dogs-Breeds

Stanford Dogs Dataset

Abstract Art Gallery

Anime Faces - Week 5

notebook7f064bddf8

View Active Events

Playground Prediction Competition

Dogs vs. Cats

Create an algorithm to distinguish dogs from cats

Kaggle · 213 teams · 8 years ago

Overview Data Code Discussion Leaderboard Rules ...

Overview

Description

Prizes

Evaluation

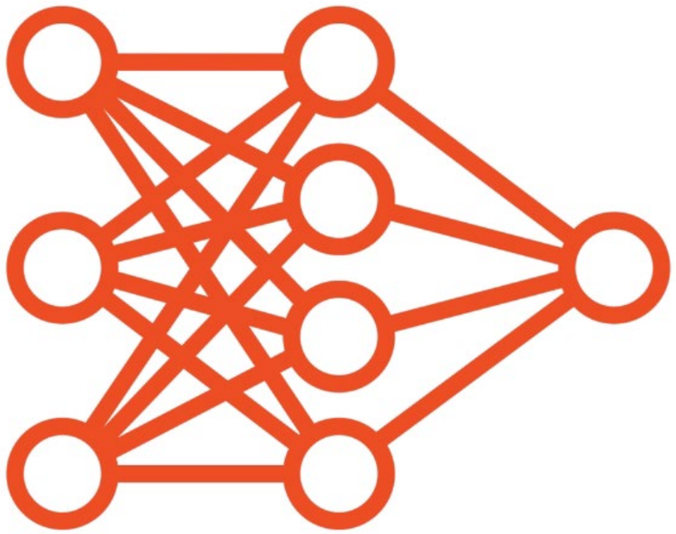
Winners

In this competition, you'll write an algorithm to classify whether images contain either a dog or a cat. This is easy for humans, dogs, and cats. Your computer will find it a bit more difficult.

Deep Blue beat Kasparov at chess in 1997.

300 * 300 * 3 = 270 000

Convolutional Layers



Convolutional networks are neural networks or deep networks with additional layers

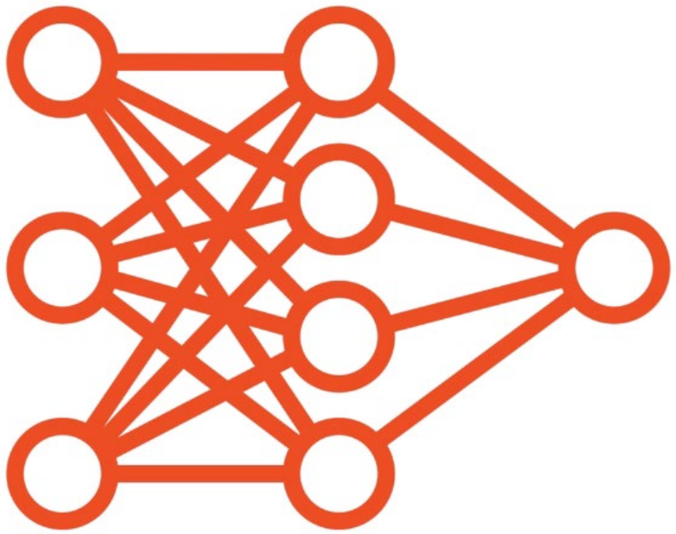
These layers reduce data size (image size)

Feature engineering performed by selecting the essential elements of an image

Convolutional layer extracts features

Pooling layer shrinks the image without losing quality

Using Convolutional Networks



Need a lot of data

Take time to train

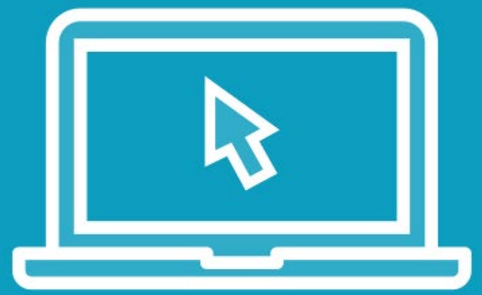
Transfer learning available

- Can use pre-trained models

Can perform data augmentation

Tuning CNNs is a difficult task

Demo



Convolutional Networks

Takeaway

Logistic regression typically used for binary classification

Naive Bayes is a probabilistic classifier that applies Bayes theorem with strong independence assumptions between the features

Support Vector Machine separates date using a line or hyperplane with many dimensions

Takeaway

KNN uses a distance algorithm to classify an element according to its neighbors

A decision tree allows predicting outcomes based on particular conditions

Neural networks help group data according to similarities

Convolutional neural networks is a class of neural networks that extract features and can analyze using a smaller image without sacrificing quality