URP 6931. Introduction to Urban Analytics

Lecture 11: Supervised learning – diverse classifiers

Instructor: Shenhao Wang
Assistant Professor, Director of Urban AI Lab
Department of Urban and Regional Planning
University of Florida

Part 3. Diverse classifiers

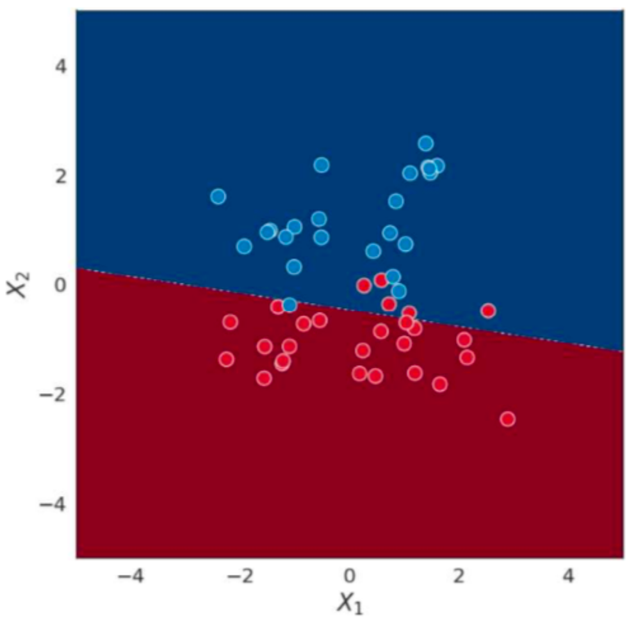
Goal: Familiarize you with the names of the classifiers and gain the intuition of the classifiers' mechanism.

1. K nearest neighbors
2. Support vector machine
3. Kernel methods
4. Naïve Bayes
5. Decision tree
6. Random forest
7. Neural networks

Review

Intuition about classifiers: create a boundary to identify the positive and negative points.

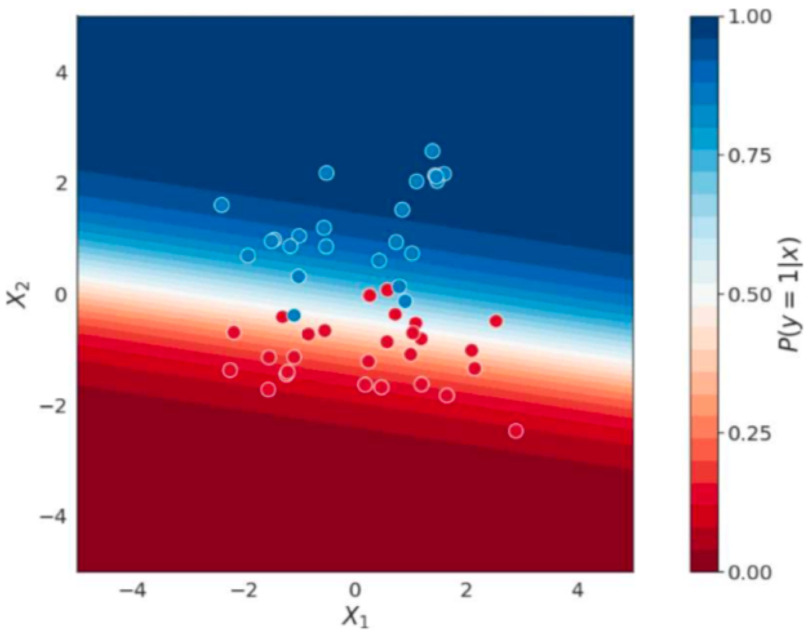
$$X \rightarrow Y$$



Deterministic classifiers

e.g. The person will not use public transit if the price is higher than $1.0

example: decision Tree

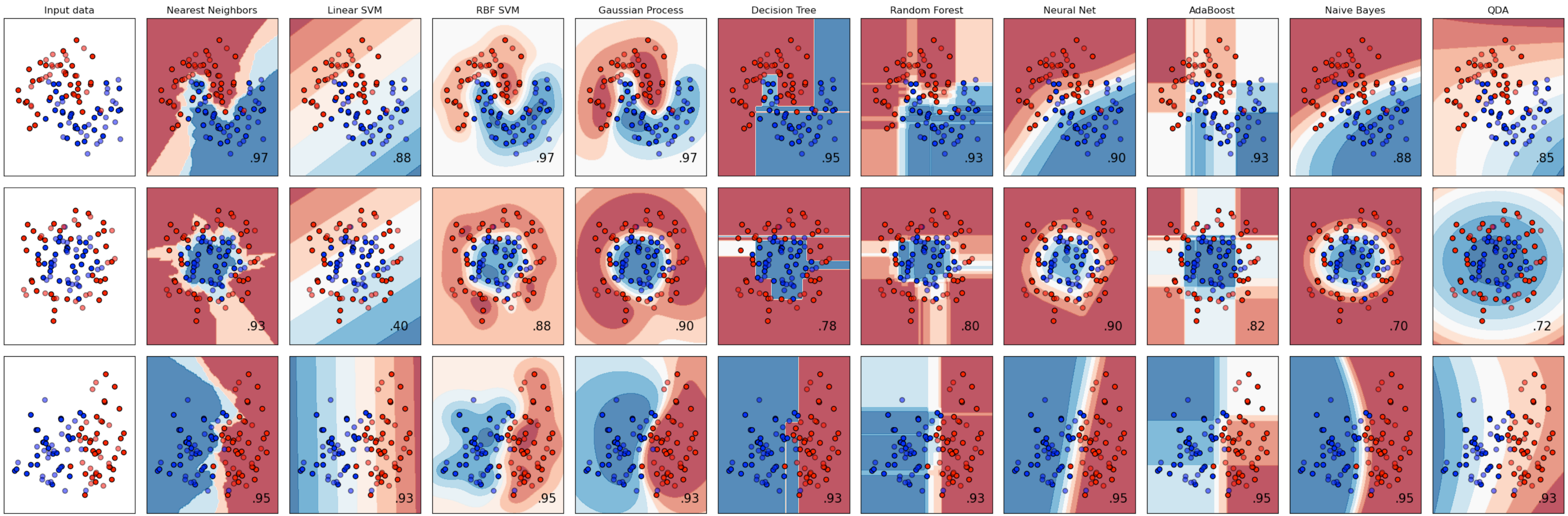


Probabilistic classifiers

e.g. The person has only 20% chance to use public transit if the price is $1.0

example: logistic regression

Logistic Regression

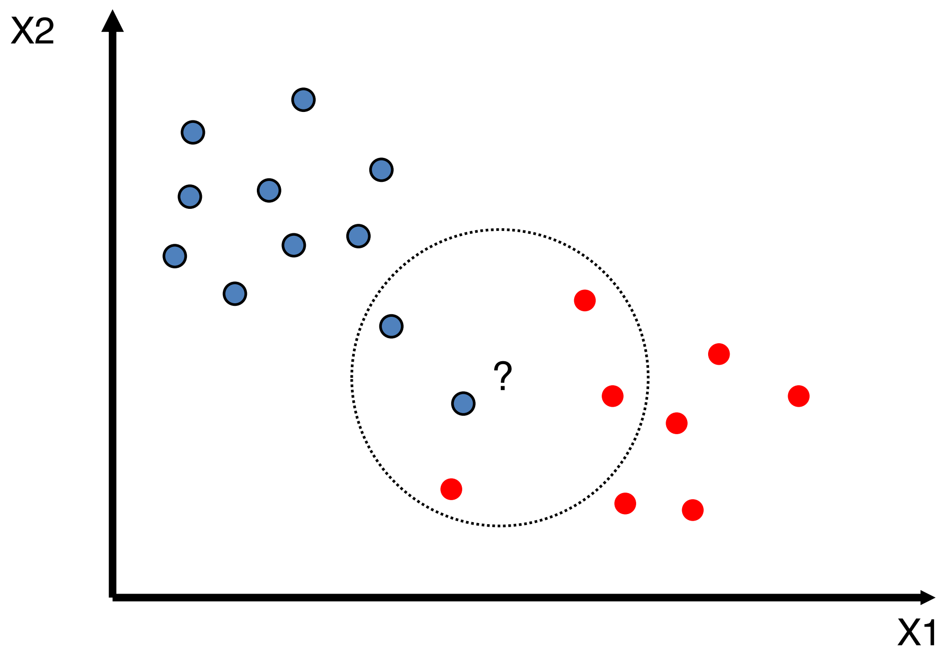


$X \rightarrow Y$

https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

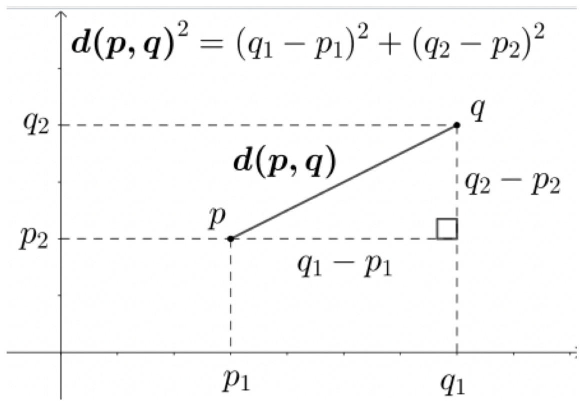
1. K-Nearest Neighborhood

How to assign label to this point?



KNN algorithm

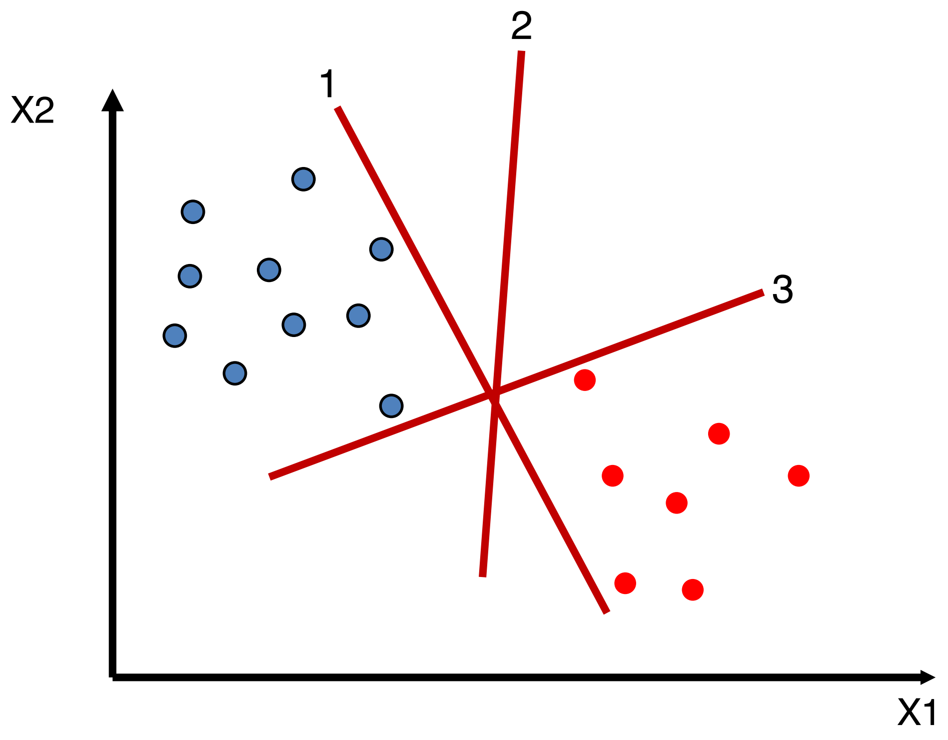
1. Choose the number K of neighbors; Assume K = 5
2. Take K nearest neighbors, using Euclidean distance



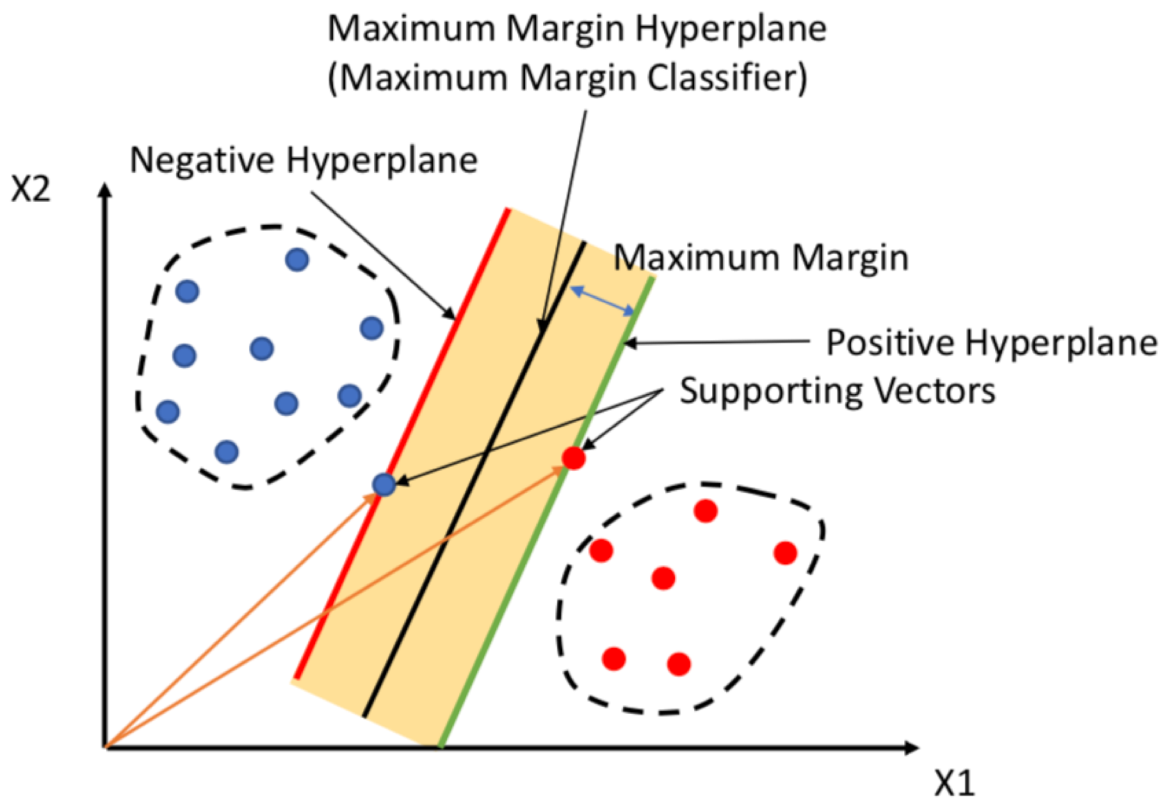
3. Count the number of data points
4. Assign the new label based on the majority vote.

2. Support Vector Machine

What about the prediction accuracy of the following classifiers? Which one do you prefer?

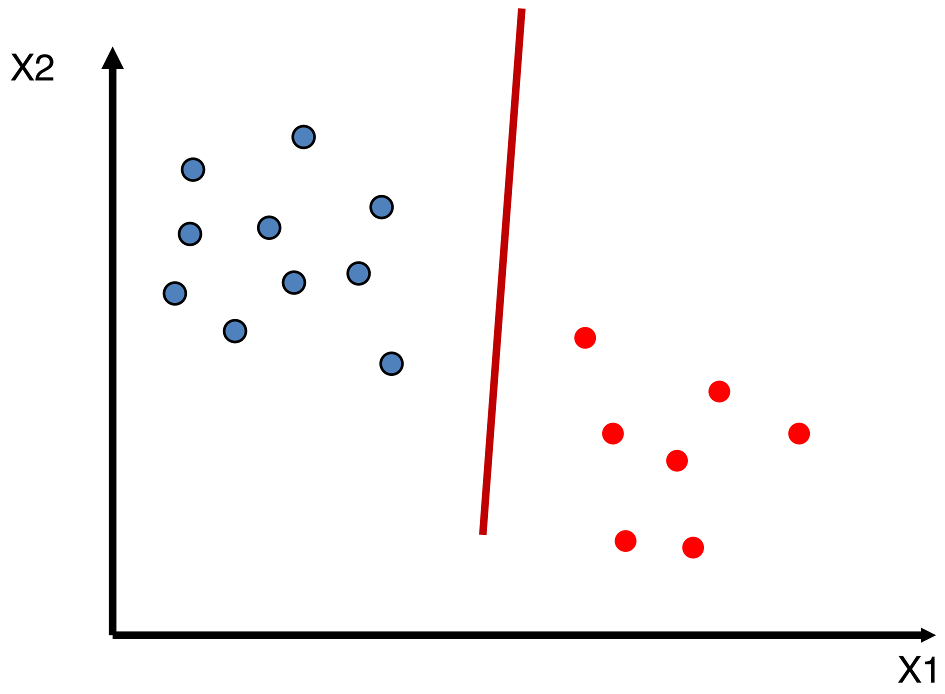


Find the classifier with the largest "buffer"

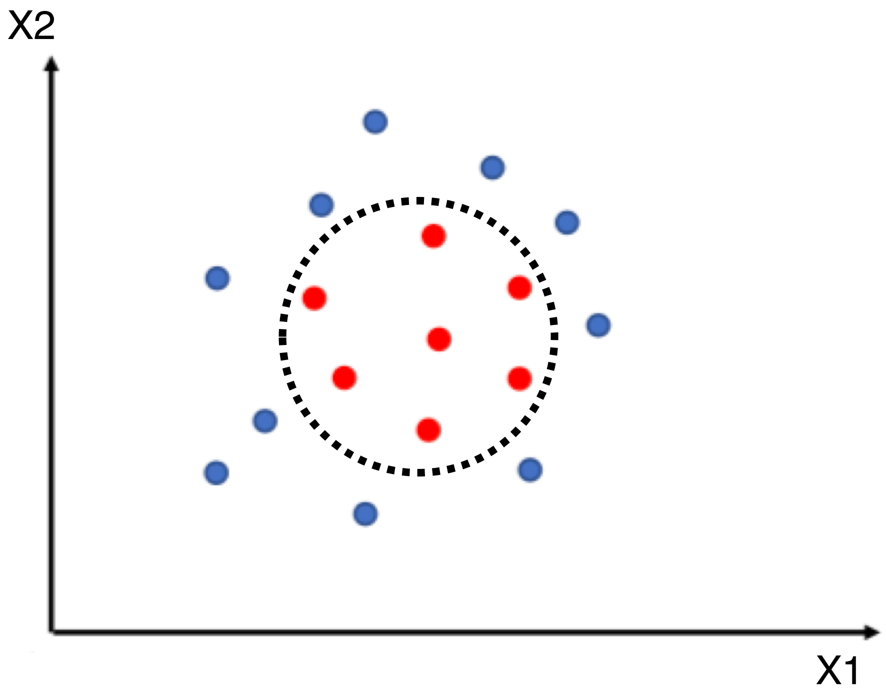


3. Kernel Methods

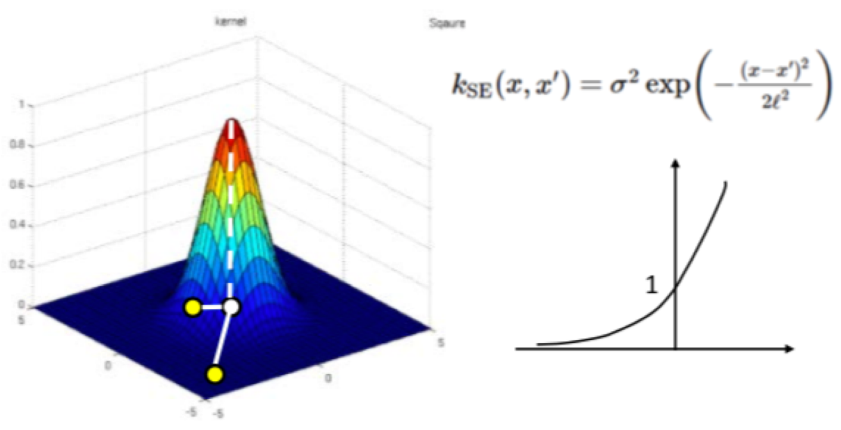
Linearly separable data



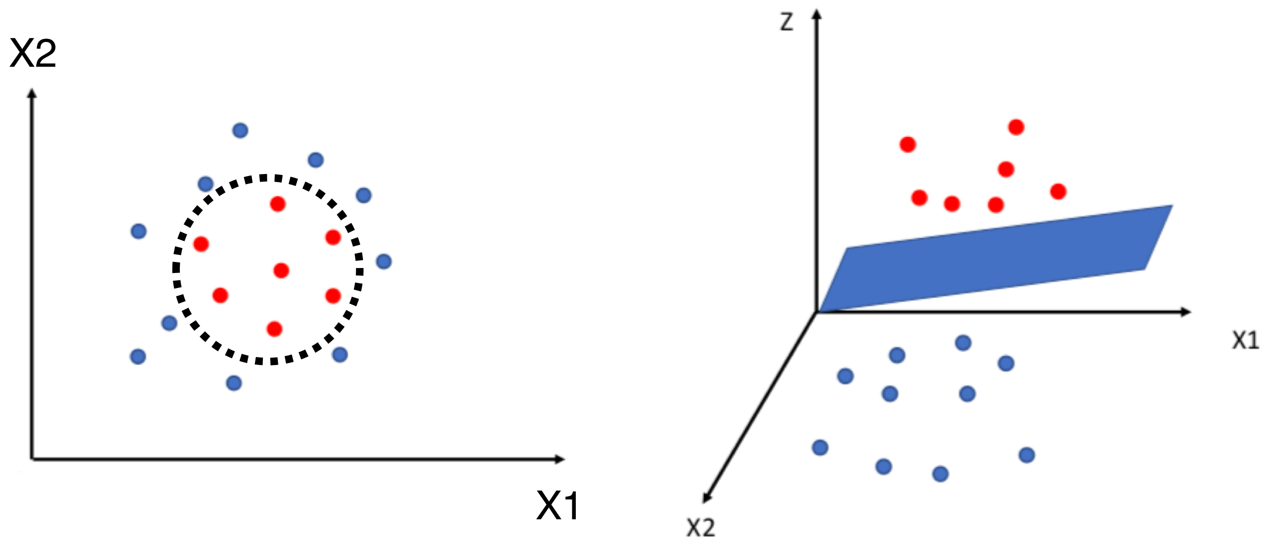
Data is not linearly separable



3. Kernel Methods: Gaussian Radial Basis Function (RBF) Kernel

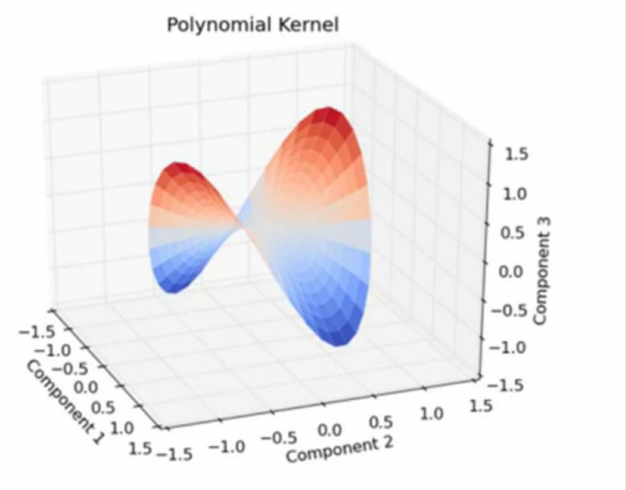


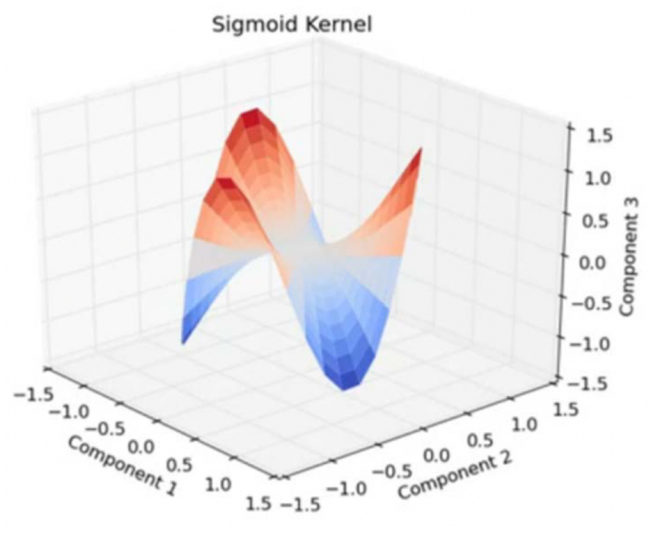
Gaussian Kernel

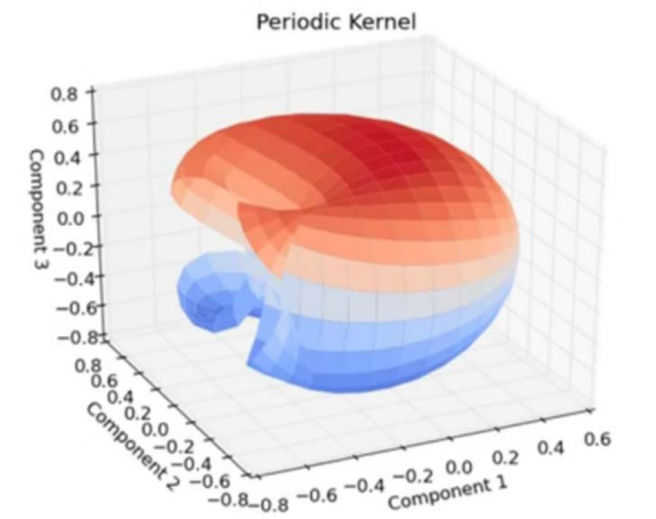


Adding another dimension z with the Gaussian Kernel to (perfectly) separate the data

3. Other Kernels







https://datafreakankur.com/machine-learning-kernel-functions-3d-visualization/

4. Naïve Bayes Classifier

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

P(A): prior probability

P(B|A): conditional likelihood

P(B): marginal likelihood

P(A|B): posterior probability

Intuition: New information B updates our prior belief A towards P(A|B)

4. Naïve Bayes Classifier

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

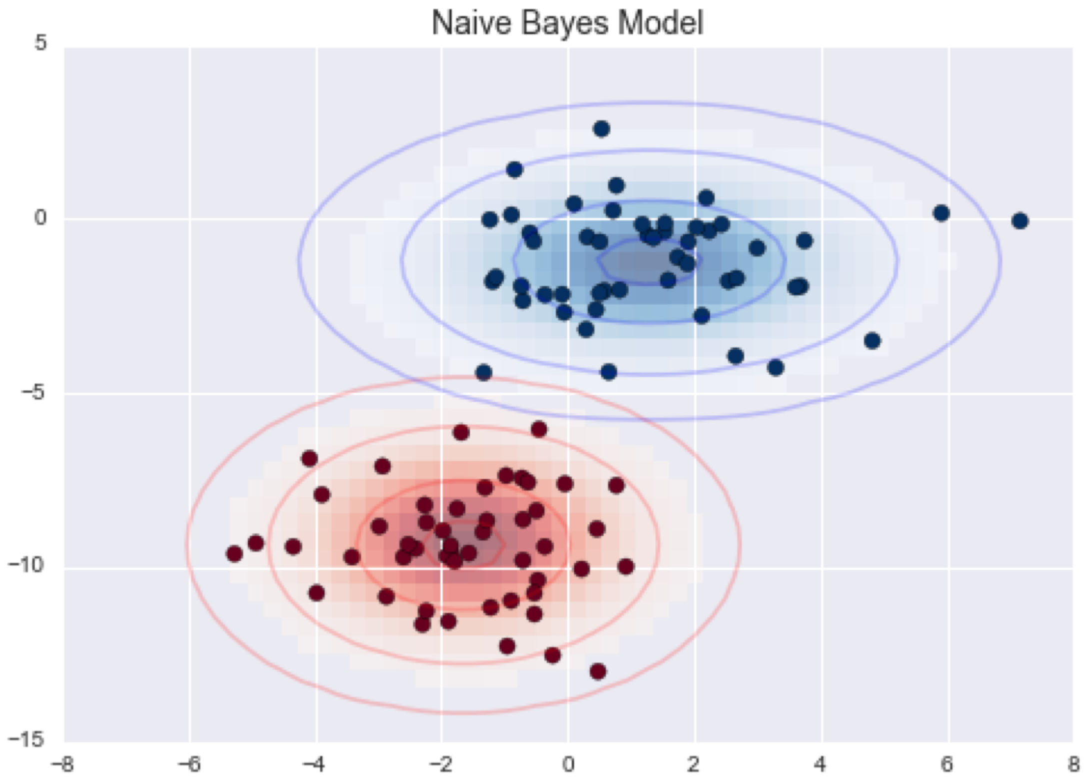
X: inputs; y: labels (0, 1).

When $\{x_1, x_2, \ldots, x_N\}$ are i.i.d., $P(X|y) = P(x_1|y)P(x_2|y) \ldots P(x_N|y)$

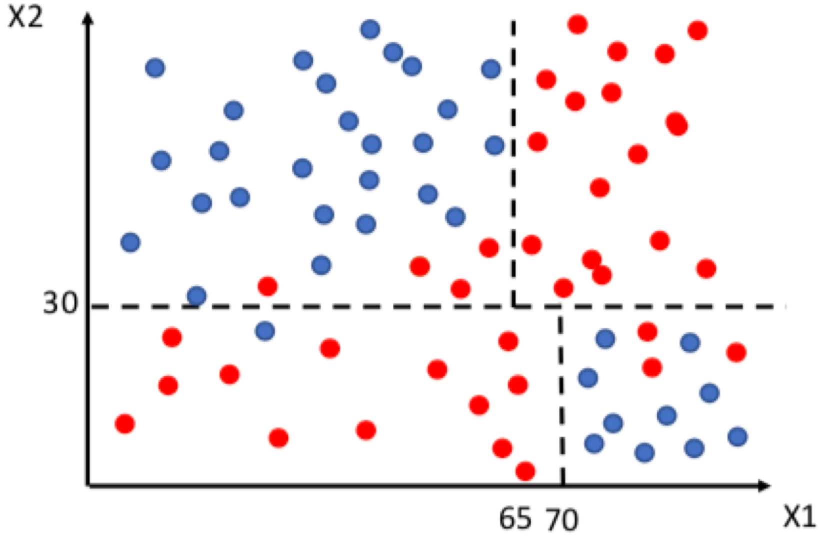
A typical assumption is Gaussian assumption:

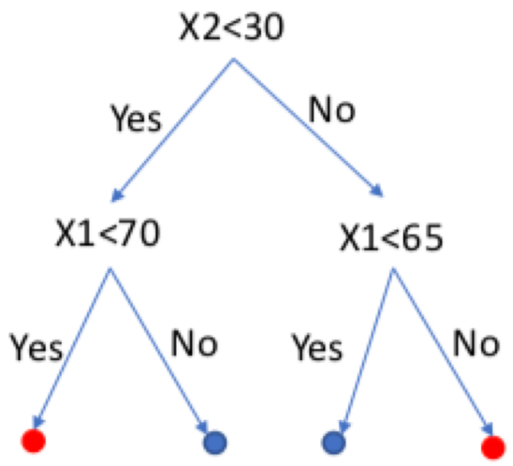
$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{\left(x_i - \mu_y\right)^2}{2\sigma_y^2}\right)$$

Intuition about the Naïve Bayes



5. Decision Tree





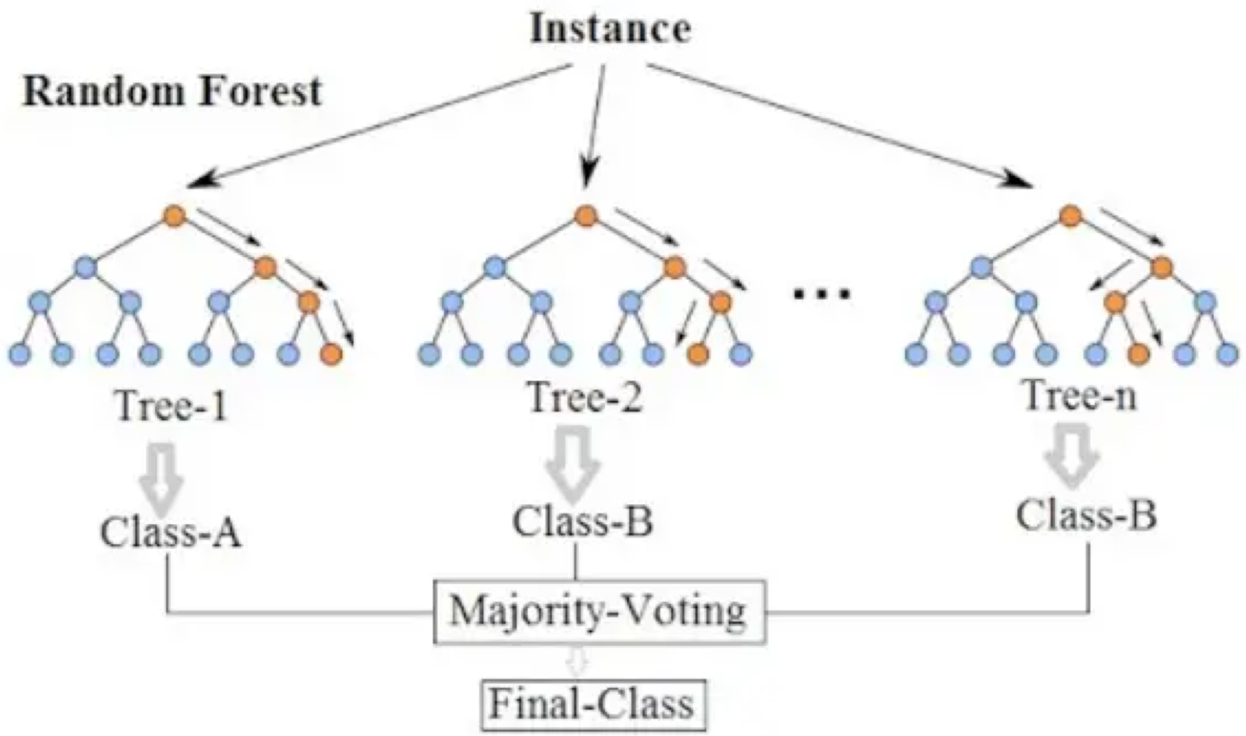
Horizontal & vertical boundaries separating the space

Tree logic for classification

e.g. If price is higher than $10, I will not use this service.
e.g. once I am older than 70 years old, I will retire.

6. Random Forest

Ensemble of decision trees.

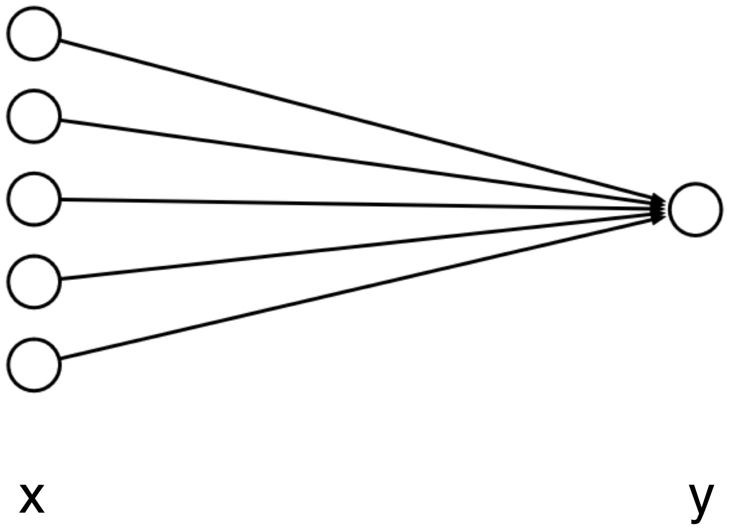


https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d

7. From linear regression to neural networks

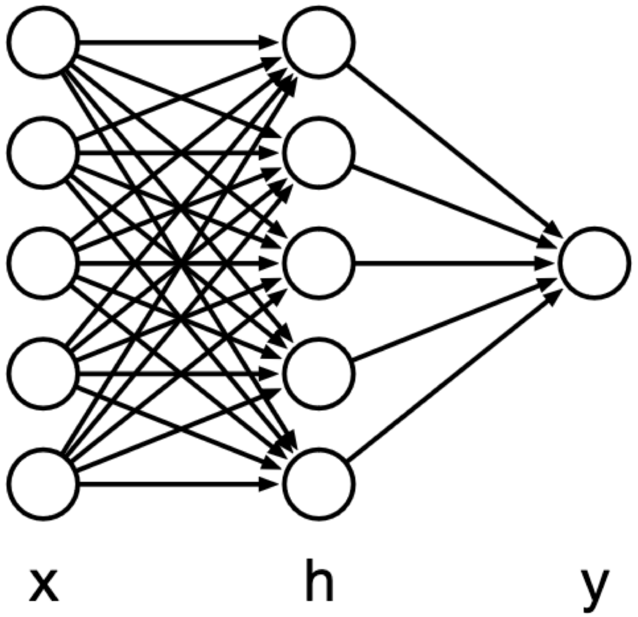
Linear regression

$$y_i = \beta_0 + \beta_1 x_{1i} + \ \ldots \beta_k x_{ki} + u$$



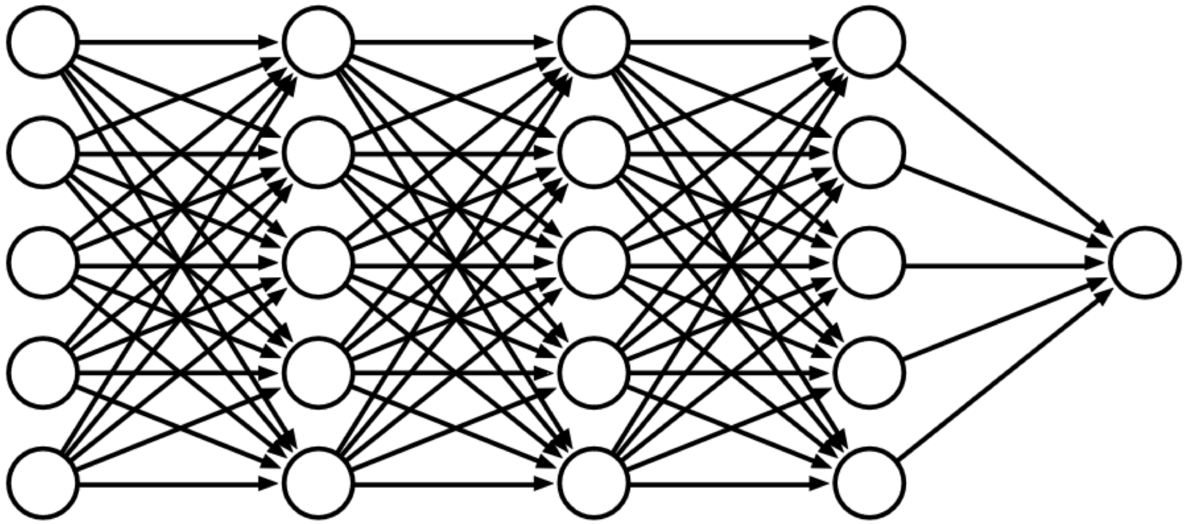
Neural networks

- Add an intermediate ("hidden") layer of processing the inputs. e.g. transforming 10 input features to 1M features (|h| = 1M).
- Intuition: providing more flexibility to fit x to y.



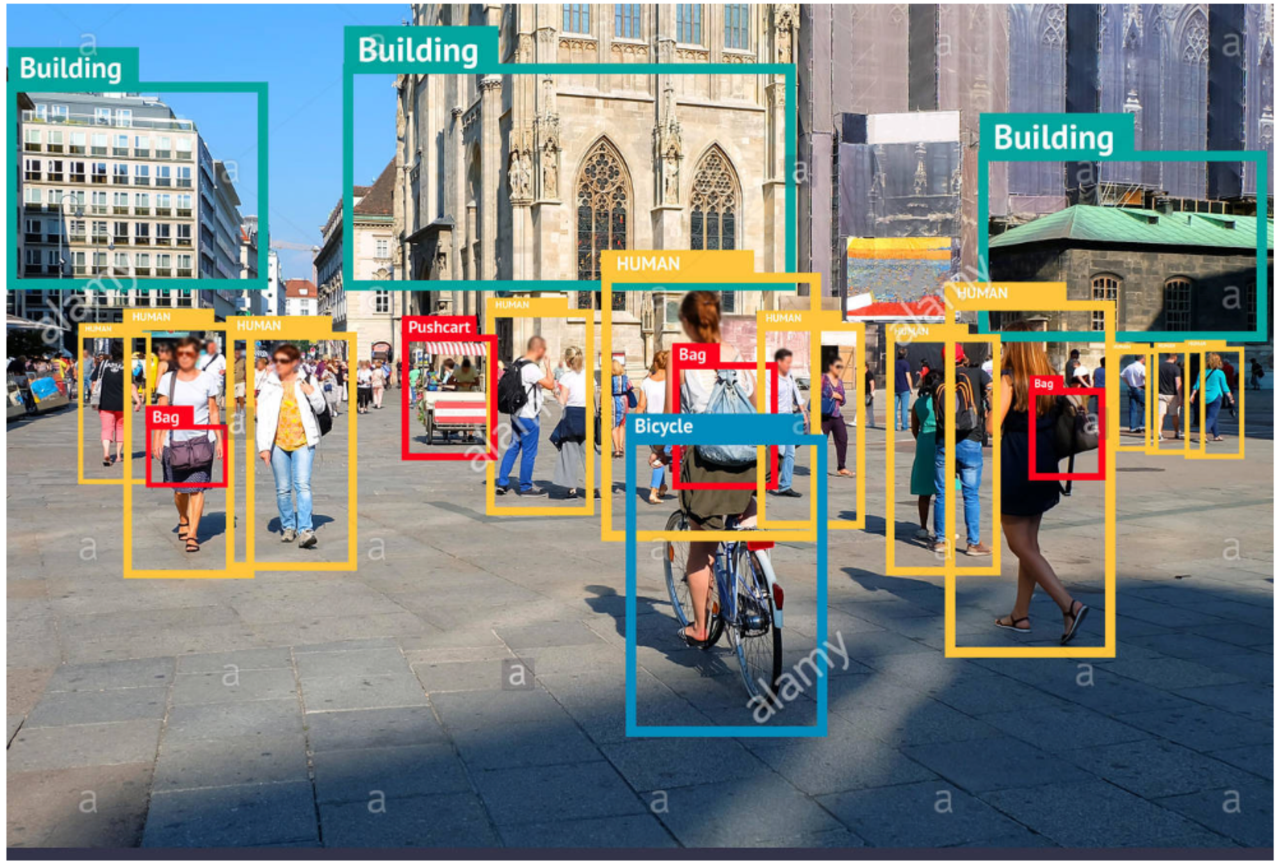
7. Deep learning and neural networks

A lot of layers in neural networks = Deep learning



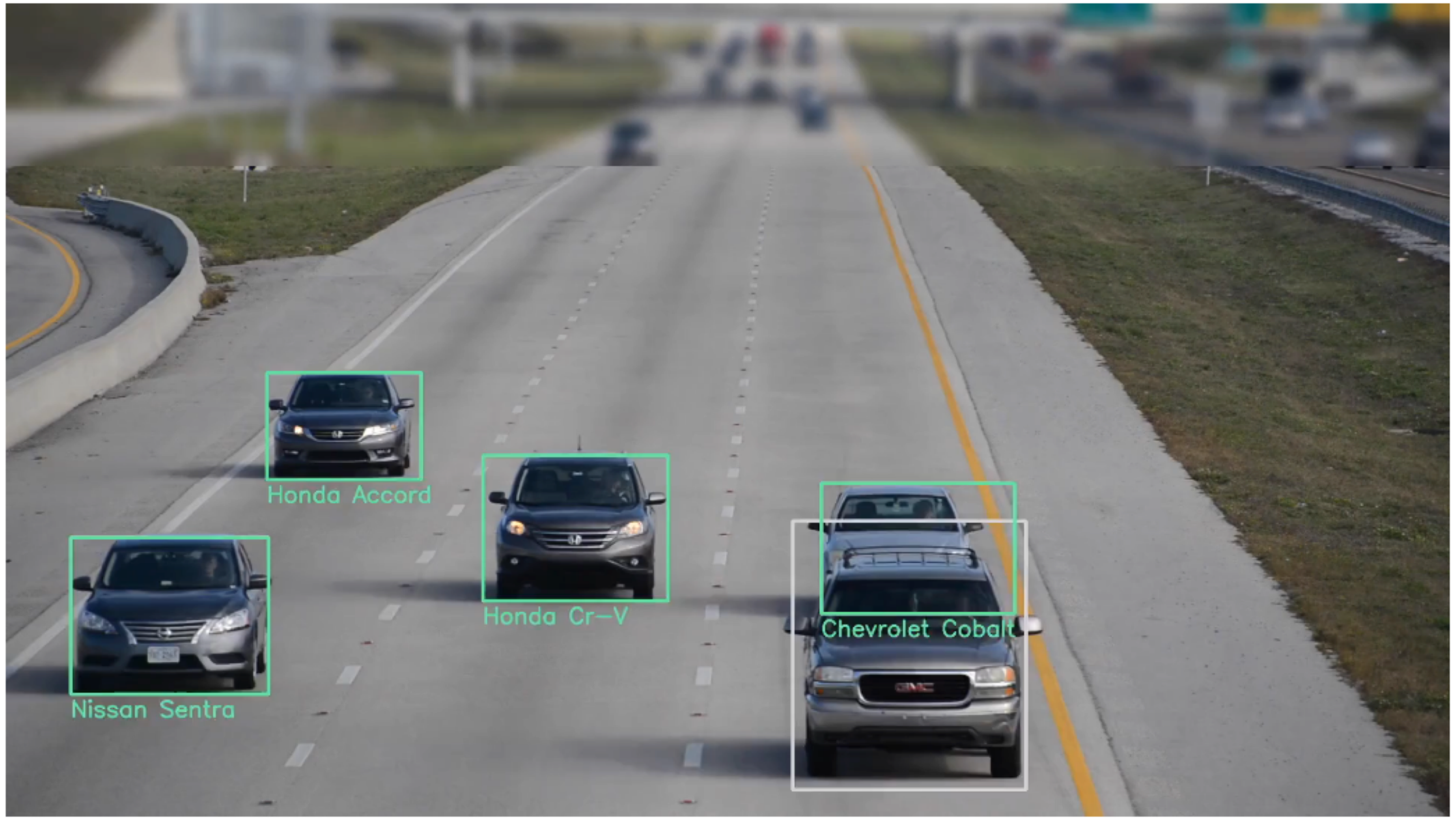
Deep learning is taking over the world…

Example. object detection for urban scenes



Deep learning is taking over the world…

Example. object detection on highways



Deep learning is taking over the world…

Example. beating human GO champions

AlphaGo

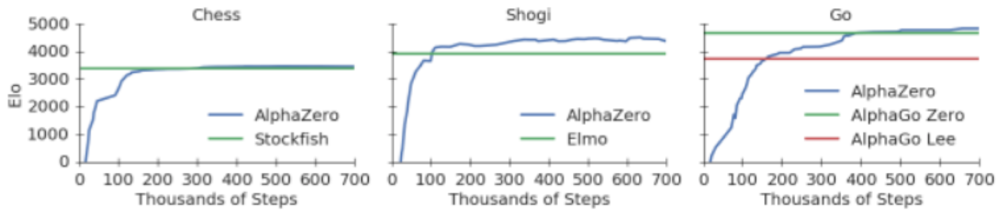


Figure 1: Training *AlphaZero* for 700,000 steps. Elo ratings were computed from evaluation games between different players when given one second per move. **a** Performance of *AlphaZero* in chess, compared to 2016 TCEC world-champion program *Stockfish*. **b** Performance of *AlphaZero* in shogi, compared to 2017 CSA world-champion program *Elmo*. **c** Performance of *AlphaZero* in Go, compared to *AlphaGo Lee* and *AlphaGo Zero* (20 block / 3 day) (*29*).

Deep learning is taking over the world…

What is the underlying model of ChatGPT?

I cannot conclude for sure, but with 99% level of confidence, it is DL...

Pros and Cons of Classifiers

Classification models	Pros	Cons
Logistic regression	Probabilistic approach, gives information about statistical significance	Assumption of logistic function
KNN	Simple to understand, fast and efficient	Need to choose the number of neighbors K
SVM	not biased by outliers, not sensitive to overfitting	Not appropriate for nonlinear problems
Kernel methods	High performance on nonlinear problems, not biased by outliers	Complex models, not the best choice for large number of features
Naïve Bayes	Efficient, and probabilistic approach	Strong assumption on the conditional distribution
DT	Interpretable; works on both linear and nonlinear problems	Easily overfit, poor results on small data sets
RF	Powerful and high performance on linear and nonlinear problems	Difficulty in interpretation; overfitting can easily occur.
DNN	High performance and versatility in processing all data structures	Difficulty in interpretation; and computationally expensive

@SuperDataScience

How to **use** the classifiers?

1. Urban Applications

As long as y is a discrete binary variable, you could use ANY of the classifiers.

1. What attract people to Florida? y: moving to Florida vs. other places.
2. Does a new public transit line support economic development? y: economic growth vs. decline
3. Do the public transit lines serve the low-income people? y: low vs. high transit ridership
4. Will people use the new energy-efficient vehicle? y: adoption of new vehicles or not
5. Do the existing zoning codes mitigate economic opportunities? y: economic growth vs. decline
6. etc.

How to **use** the classifiers?

2. Typically run all the classifiers to compare predictive performance without much interpretation

Example in practice: y - adoption of new technology {0, 1}

1. Splitting the training and testing sets
2. Choosing classifiers, e.g., KNN, SVM, SVM (with kernels), NB, DT, RF, NN, etc.
3. Training classifiers with the training set and evaluating them with the testing set.
4. Choosing the model with the highest predictive performance.
5. (optional) occasionally discuss why a specific classifier achieves the highest performance.

Comment: this practice even applies to the state-of-the-art ML/DL research work. e.g. classifier vs. performance table