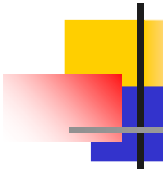
Classification

(1IK172 Introduction to Data Analytics)

Classification

- One of the most frequent task in analytics
  - Without paying attention, we are all the time classifying things
  - We perform a classification task when:
    - Marking a comment as rude or polite
    - Adding someone to our social network
    - Telling our child if an animal in the zoo is a bear, bird, cat etc.
    - Reading numbers from a sheet of paper
- The main difference from Regression is that in classification the target is discrete

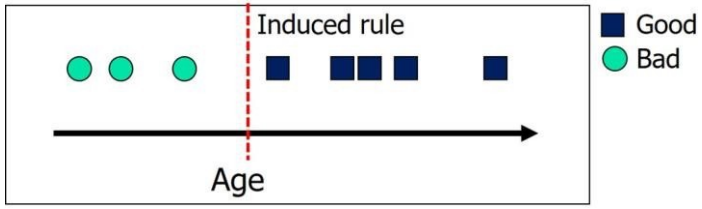


Classification

- Classification Task
  - Predictive task where a label to be assigned to a new, unlabeled, object, given the value of its predictive attributes, is a qualitative value representing a class or category.
  - Classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

Example

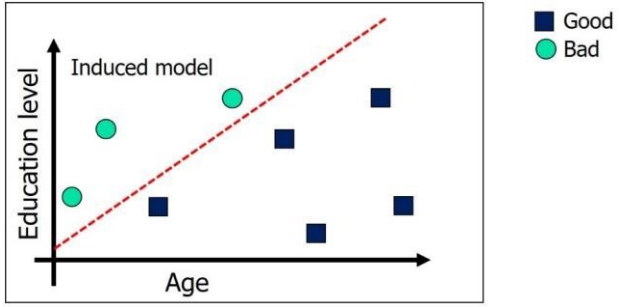
Name	Age	Company
Andrew	51	Good
Bernhard	43	Good
Dennis	82	Good
Eve	23	Bad
Fred	46	Good
Irene	29	Bad
James	42	Good
Lea	38	Good
Mary	31	Bad



If age < 32
Then company is Bad
Else company is Good

Example

Name	Age	Education level	Company
Andrew	51	1.0	Good
Bernhard	43	2.0	Good
Dennis	82	3.0	Good
Eve	23	3.5	Bad
Fred	46	5.0	Good
Irene	29	4.5	Bad
James	42	4.0	Good
Lea	38	5.0	Bad
Mary	31	3.0	Good



If person > decision border
Then company is Bad
Else company is Good

Classification Algorithms

- Dozens of algorithms exist and a lot of them have many variations
- The algorithms can be classified into 4 categories
  - Distance-based algorithms
  - Probability-based algorithms
  - Search-based algorithms
  - Optimization-based algorithms

Classification Algorithms: Distance-based

- **Distance-based algorithms**
  - **K-nearest Neighbor**
  - Case-based Reasoning

Classification Algorithms: Probability-based

- **Probability-based algorithms**
  - Logistic Regression
  - Naïve Bayes

Classification Algorithms: Search-based

- **Search-based algorithms**
  - **Decision Tree**
  - Random Forest

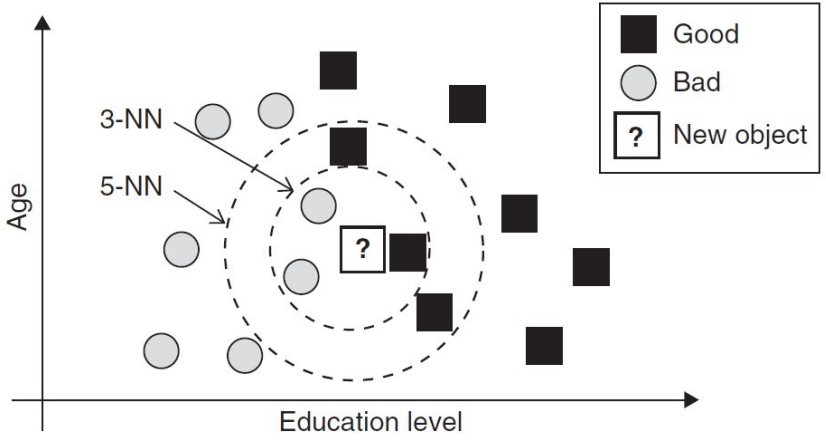
Classification Algorithms: Optimization-based

- **Optimization-based algorithms**
  - **Support Vector Machines**
  - Artificial Neural Networks

K-nearest Neighbor Algorithm

Algorithm K-NN test algorithm.

1: INPUT D_{train}, the training set
2: INPUT D_{test}, the test set
3: INPUT d, the distance measure
4: INPUT x_i objects in the test set
5: INPUT K, the number of neighbors
6: INPUT n, the number of objects in the test set
7: **for all** object x_i in D_{test} **do**
8: **for all** object x_j in D_{test} **do**
9: Find the k objects from D_{train} closest to x_i according to the chosen distance measure d
10: Assign x_i the class label most frequent in the k closest objects



K-nearest Neighbor Algorithm

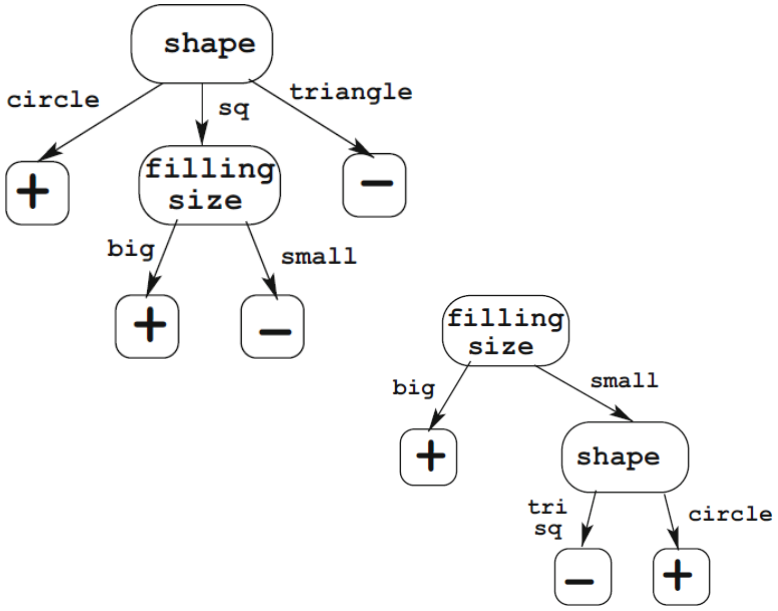
- **Pros**
  - Its simplicity
  - Good predictive power in several problems
  - It is inherently incremental
- **Cons**
  - k-NN can take a long time to classify a new object
  - The use of only local information to classify new objects
  - Sensitive to the presence of irrelevant attributes and outliers
  - **Predictive quantitative attributes need to be normalized**

Decision Tree

What is it?

- A technique to create easily interpretable flowchart-like models
- The new (classifiable) object starts at the root node
- At each node, the object travels down based on the value of **one** of it's attribute
- The problem space is split by the node along the axis of the attribute
- Leaf nodes are output nodes, at each leaf node we have an assigned output value

Example



Example	crust size	shape	filling size	Class
*e*1	big	circle	small	**pos**
*e*2	small	circle	small	**pos**
*e*3	big	square	small	**neg**
*e*4	big	triangle	small	**neg**
*e*5	big	square	big	**pos**
*e*6	small	square	small	**neg**
*e*7	small	square	big	**pos**
*e*8	big	circle	big	**pos**

Example from *An Introduction to Machine Learning* by *Miroslav Kubat*

Decision Tree

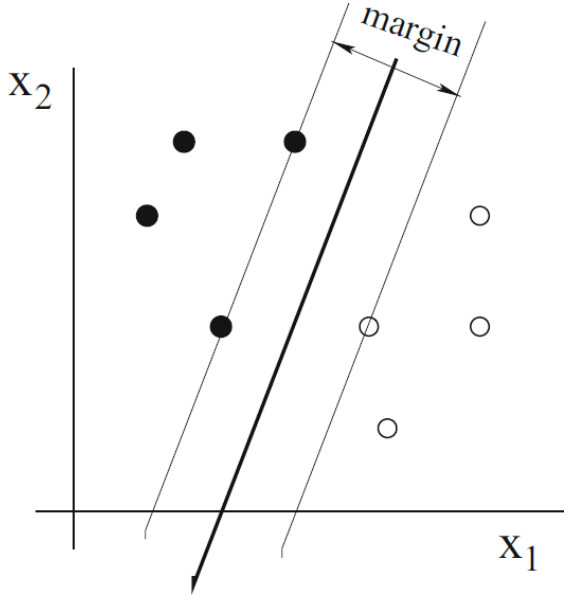
- **Pros**
  - Its simple and interpretable as flowchart or a set of rules
  - Very robust: Can handle outliers and missing values, no need to normalize, does not care about attribute correlation (all thanks to handling one attribute at a node)
- **Cons**
  - Fails at complex models where attribute interrelations are important
  - Can only split along an axis
  - Only able to learn $x_i <= a$ rules, where x_i is a predictive attribute and a is a constant

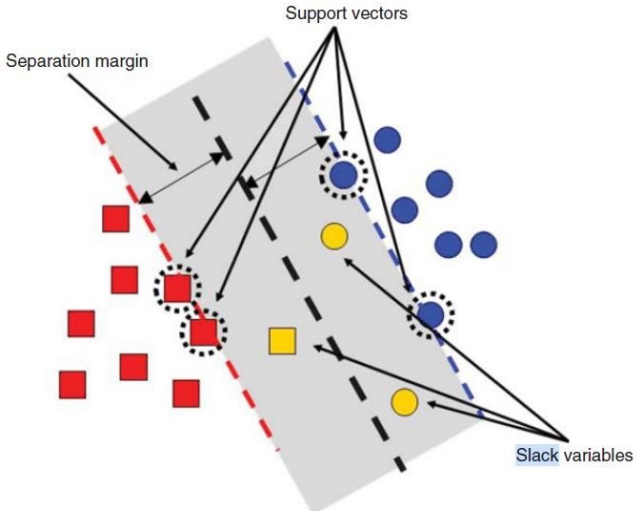
Support Vector Machine

What is it?

- A technique allowing us to create good generalizing models that separate the problem space
- Unlike logistic regression the model clearly decides the class label instead of probabilistic result
- Unlike Neural Networks we find the most optimal solution to split the data by finding the line that maximizes the margin with respect to the support vectors
- Introduces the **kernel trick** to transform data into linearly separable representation

Support Vector Machine



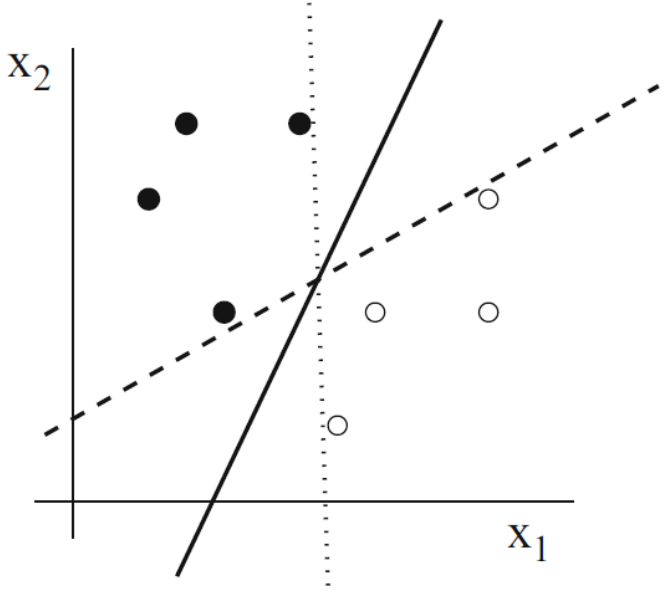


Illustration

- The thick line is the class separator, the thin lines are the support vectors for each class
- The class separator is the best fit for maximizing margin size
- We can allow some slack variables inside the margin zone to increase margin size

Support Vector Machine

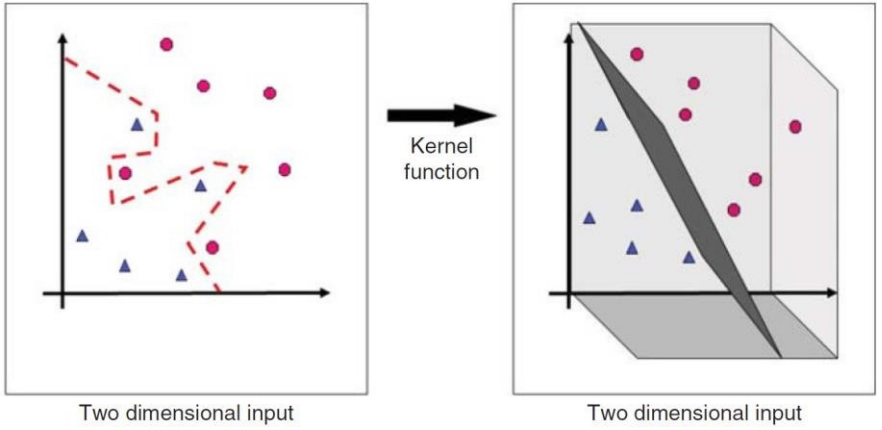
- Out of the many possible separators SVM will find the most optimal one
- Greater margin means better generalized model



Support Vector Machine

Kernel function

- A technique to increase dimensionality in order to transform a non-linear problem into a linear one
- There are also more advanced kernels that can solve non-linear problems these are Radial Basis Function (RBF) and Polynomial kernel



Support Vector Machine

- **Pros**
  - Not random, same results achieved between runs (deterministic)
  - Good performance in many problems
  - Good theoretical foundations
- **Cons**
  - Very sensitive to hyperparameter values
  - Sensitive to outliers, magnitude difference between variables (needs normalization)
  - Training time grows at least quadratically with increased training samples

Measuring predictive performance

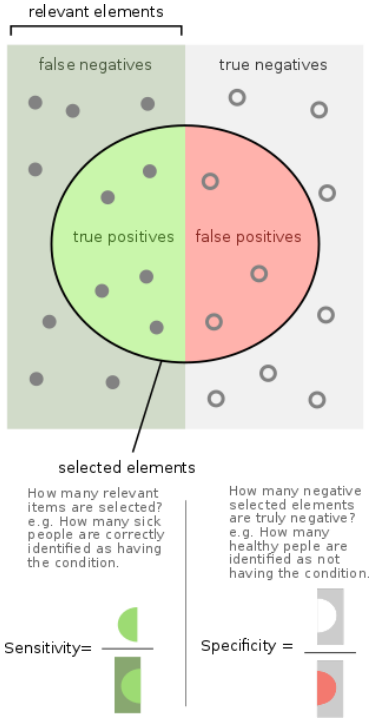
- Assess predictive performance of a classification model
  - How frequent the predicted labels are the true class labels?
  - Model predictive performance must be better than predicting in the majority class
    - Class with the largest number of objects

Measuring predictive performance

- Confusion matrix reports the predictive performance of a binary classifier
  - True class
    - Positive class
    - Negative class
  - Predicted class
  - Each cell contains the count
  - Can be easily extended to multiclass problems

		True class: p	True class: n
Predicted class	P	True positives (TP)	False positives (FP)
	N	False negatives (FN)	True negatives (TN)

Measuring predictive performance



$$\frac{FP}{FP+TN}$$ False positive rate (FPR) = 1-TNR

$$\frac{FN}{TP+FN}$$ False negative rate (FNR) = 1-TPR

$$\frac{TP}{TP+FN}$$ True positive rate (TPR), also known as recall or sensitivity

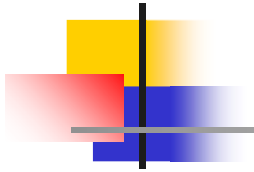
$$\frac{TN}{TN+FP}$$ True negative rate (TNR), also known as specificity

$$\frac{TP}{TP+FP}$$ Positive predictive value (PPV), also known as precision

$$\frac{TN}{TN+FN}$$ Negative predictive value (NPV)

$$\frac{TP+TN}{TP+TN+FP+FN}$$ Accuracy

$$\frac{2}{1/\,precision+1/\,recall}$$ F1-measure



Thank You!