Weekly meeting 5

Dr. Doina Bein
Thursday, June 22, 10:30am-12pm

Surveys to be completed

To be done today, before starting research:

CIC-PCUBED Pre-event survey:

https://fullerton.qualtrics.com/jfe/form/SV_6YIVSkC6hLxbunA

Project 1: Data Science

What you need to do: topics & objectives

Objective 1: Learn Python using some textbook or some online courses such as (https://www.codecademy.com/learn/learn-python). Shared by Stephanie Pocci: Learn Python in a couple hours. This YouTuber does a very beginner-friendly crash course about the capabilities of Python and its uses. Here is the link: https://www.youtube.com/watch?v=rfscVS0vtbw

Objective 2: Learn how to use Jupyter Notebook. Start here http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html

Objective 3: For data science, find a suitable dataset and start training some neural network using with Google tensorflow.

Logistics for all students

- Who is participating: list of current research students and their availability
- Research will be conducted virtually during the week with in-person meetings throughout the week
- Zoom meetings for me to teach new topics and for you to participate in open discussions
- Support:
  - If needed, you can meet me
    Zoom: Mon, Tu, Wed from 8:30-10:25 am
    IN PERSON: Mon, Tu, Wed from 8:30-9:30 am, Thursday 8:30-10am or by email
  - CIC-PCUBED peer mentor: (tentative) availability

Logistics for all students (contd.)

- Make a copy of this GDoc Work schedule, share the Gdoc copy with me, and maintain it weekly and daily; due at the end of Week 2
- Before the end of week 3, make a copy and maintain your Proposed work by individual or teams of up to three; due by the end of Week 3
- Complete your availability here; try to have it consistent over the 7 weeks such that it will be easy to partner in the project
- Group projects: to be decided; sample list here
- Oral or poster presentations: tentatively scheduled for Friday, July 28, from 8:30am-12:30 pm and if needed, from 1:30-4 pm

Please checkout:

- Other websites and ebooks
- Websites with free datasets
- More resources on selected topics
- If you find good, free resources, please share it by email or during weekly meetings
- Next meeting: I will lecture on ZOOM on Data Science: Friday, June 23, from 10:30am-12pm

Progress on Learning Python

- Free course: https://www.codecademy.com/learn/learn-python
- Free course: https://www.kaggle.com/learn/python
- Youtube video (about 4 hours): https://www.youtube.com/watch?v=rfscVS0vtbw

Data Science

Supervised Learning

- In supervised learning, we are given a labeled *training set* $Z = \{(x_i , y_i)\}$ with $y_i \in \pm1$ (the ground truth labeled data) and the task is to learn a *classifier* so that we can classify new unlabeled observations of a *testing set* $Q = \{x_i'\}_i$.
- When the training set has only two classes, we deal with *binary classification*, otherwise it is a multi-class classification problem.
- Statistical learning assumes that both the training set and the testing set are independently and identically sampled for an arbitrary but fixed unknown distribution.
- Our focus: learn a target function that can be used to predict the values of a discrete class attribute
- The task is commonly called: Supervised learning, classification, or inductive learning.

The data and the goal

(https://www.cs.uic.edu/~liub/teach/cs583-fall-06/CS583-supervised-learning.ppt)

- Data: A set of data records (also called datapoints, examples, instances or cases) described by
  - *k* attributes: A_1, A_2, … A_k.
  - a class: Each example is labelled with a pre-defined class.
- Goal: To learn a classification model from the data that can be used to predict the classes of new (future, or test) cases/instances.

Supervised vs. Unsupervised Learning

(https://www.cs.uic.edu/~liub/teach/cs583-fall-06/CS583-supervised-learning.ppt)

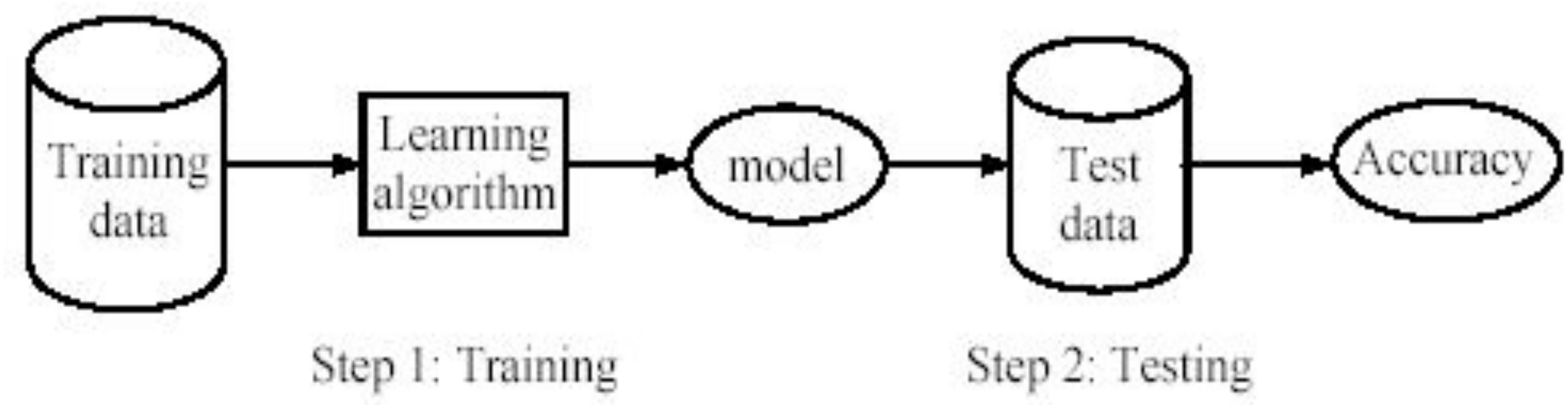
- Supervised learning: classification is seen as supervised learning from examples.
  - Supervision: The data (observations, measurements, etc.) are labeled with pre-defined classes. It is like that a "teacher" gives the classes (supervision).
  - Test data are classified into these classes too.
- Unsupervised learning (clustering)
  - Class labels of the data are unknown
  - Given a set of data, the task is to establish the existence of classes or clusters in the data

Supervised learning process: two steps

(https://www.cs.uic.edu/~liub/teach/cs583-fall-06/CS583-supervised-learning.ppt)

- Learning (training): Learn a model using the training data
- Testing: Test the model using unseen test data to assess the model accuracy

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$



What do we mean by learning?

(https://www.cs.uic.edu/~liub/teach/cs583-fall-06/CS583-supervised-learning.ppt)

- Given
  - a data set *D*,
  - a task *T*, and
  - a performance measure *M*,

  a computer system is said to **learn** from *D* to perform the task *T* if after learning the system's performance on *T* improves as measured by *M*.
- In other words, the learned model helps the system to perform *T* better as compared to no learning.

Topics in Data Science

- Supervised machine learning approach is where the learning algorithm is first trained with data and labels, and later the accuracy is evaluated on training set without labels.
- Supervised learning requires that data is labeled, before it used for training the classifier; this process of labeling is highly expensive and time consuming
- k-NN, SVM, Decision Tree, Random Forest, and Neural Networks
    - An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible (Wikipedia)
    - Decision Trees are flow-chart like structures that lets you classify input data points or predict output given an input.
    - A Random Forest is a robust approach to implement large number of decision trees and then ensemble their outputs.

Decision tree

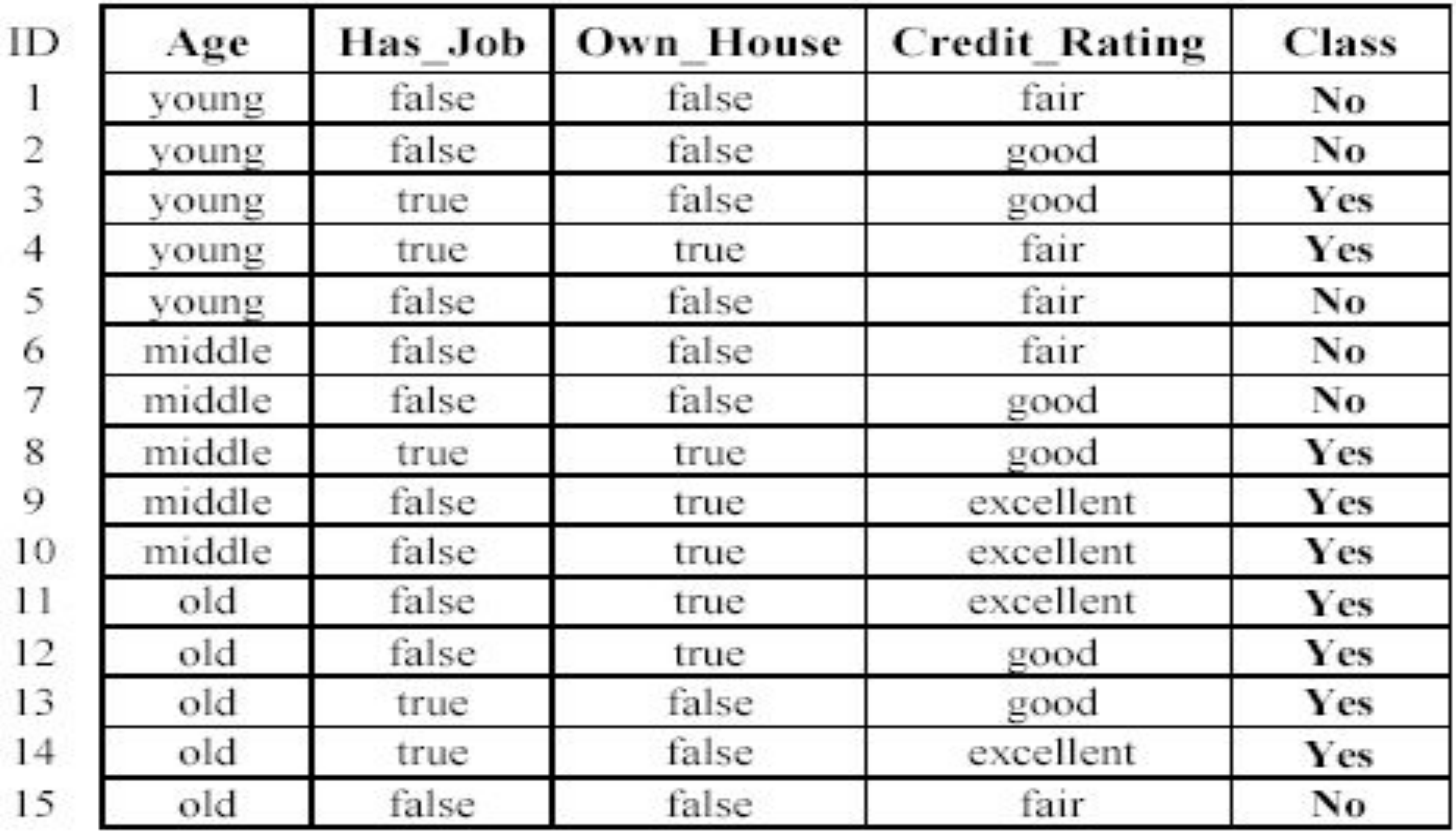
(https://www.cs.uic.edu/~liub/teach/cs583-fall-06/CS583-supervised-learning.ppt)

- Decision tree learning is one of the most widely used techniques for classification.
    - Its classification accuracy is competitive with other methods, and
    - it is very efficient.
- The classification model is a tree, called decision tree.
- C4.5 by Ross Quinlan is perhaps the best known system. It can be downloaded from the Web.

The loan data

(https://www.cs.uic.edu/~liub/teach/cs583-fall-06/CS583-supervised-learning.ppt)

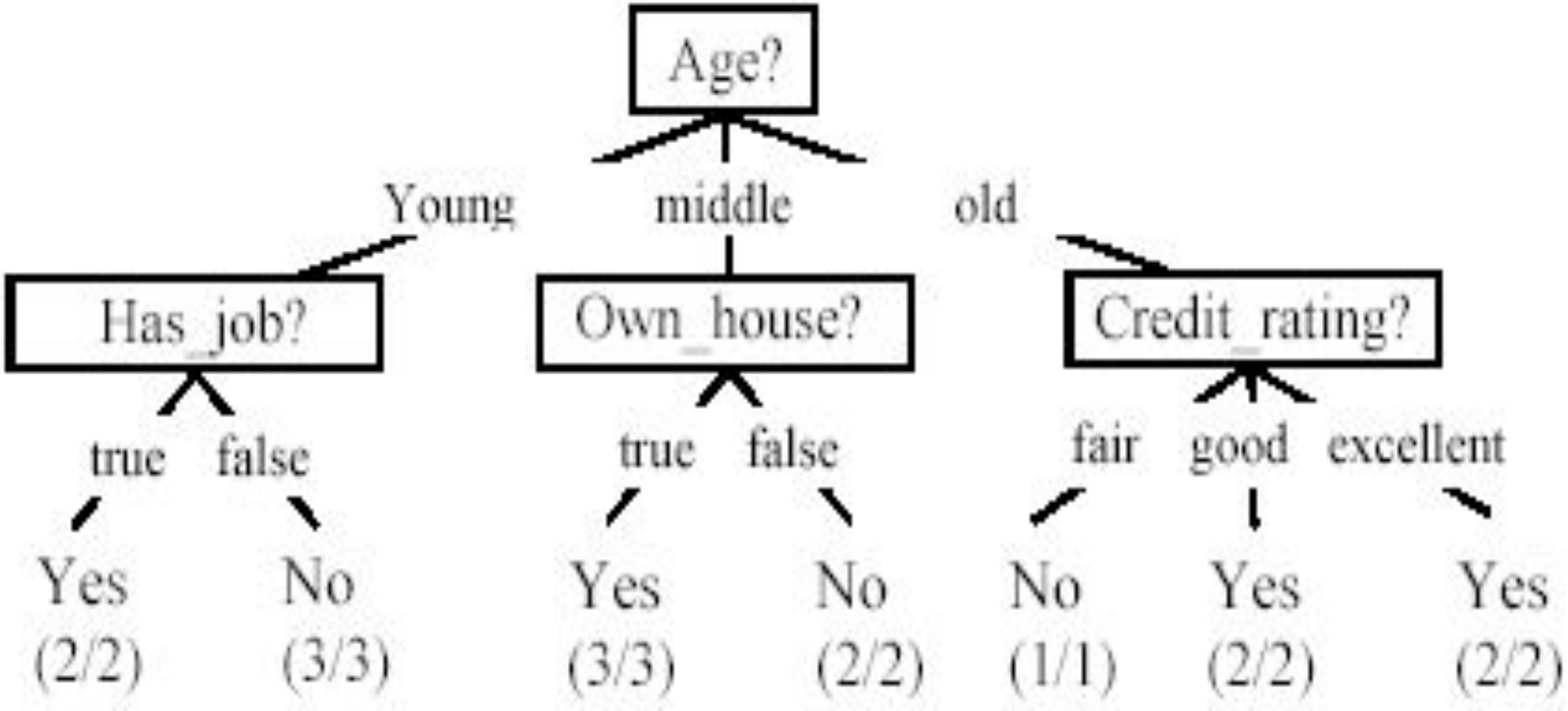
Approved or not



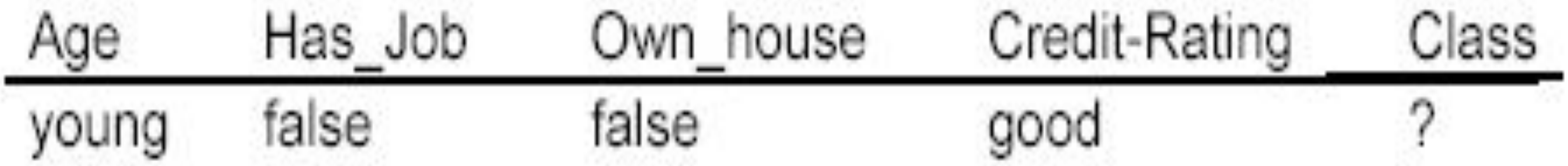
ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

A decision tree from the loan data

- Decision nodes and leaf nodes (classes)

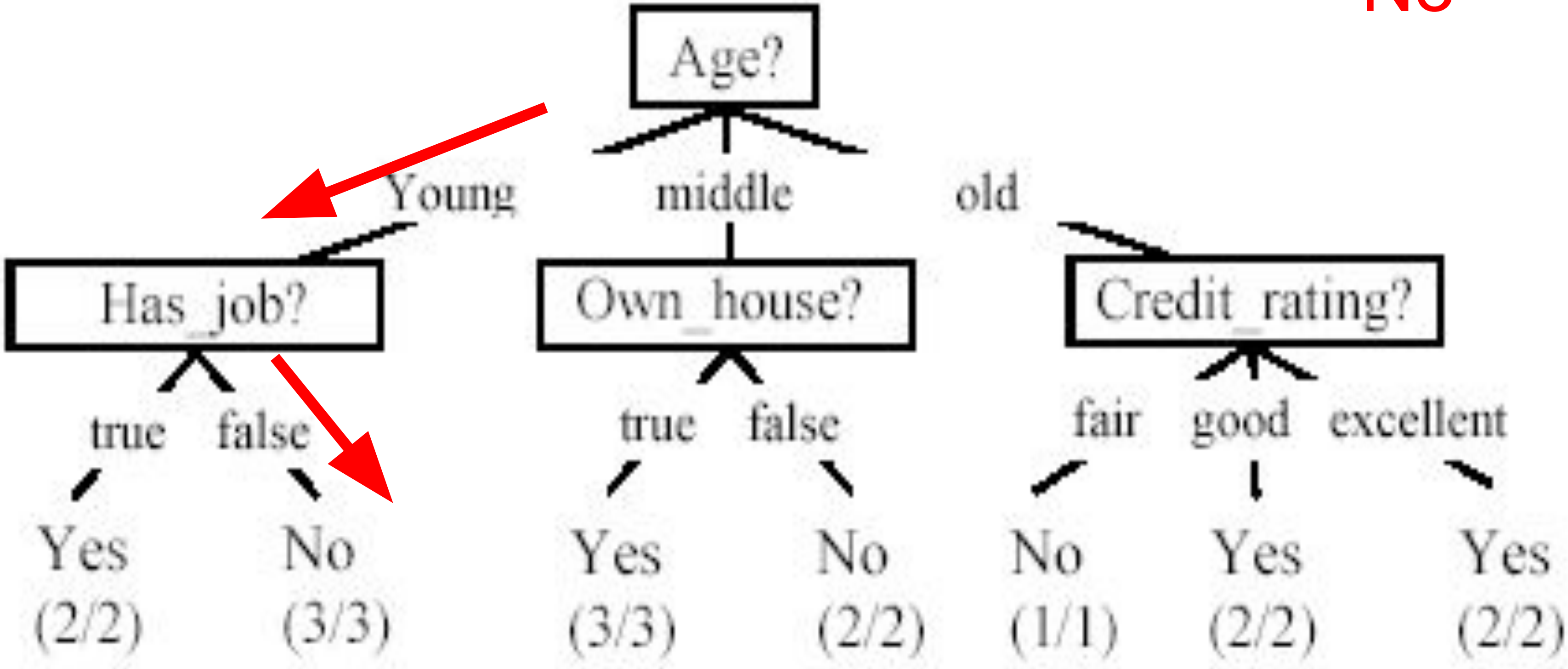


Use the decision tree



Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

No



Random Forest (Breiman 2001)

(hanj.cs.illinois.edu/bk3/bk3_slides/08ClassBasic.ppt)

- Random Forest:
  - Each classifier in the ensemble is a *decision tree* classifier and is generated using a random selection of attributes at each node to determine the split
  - During classification, each tree votes and the most popular class is returned
- Two methods to construct Random Forest:
  - Forest-RI (*random input selection*): Randomly select, at each node, F attributes as candidates for the split at the node. The CART methodology is used to grow the trees to maximum size
  - Forest-RC (*random linear combinations*): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)
- More robust to errors and outliers
- Insensitive to the number of attributes selected for consideration at each split

SUPPORT VECTOR REGRESSOR

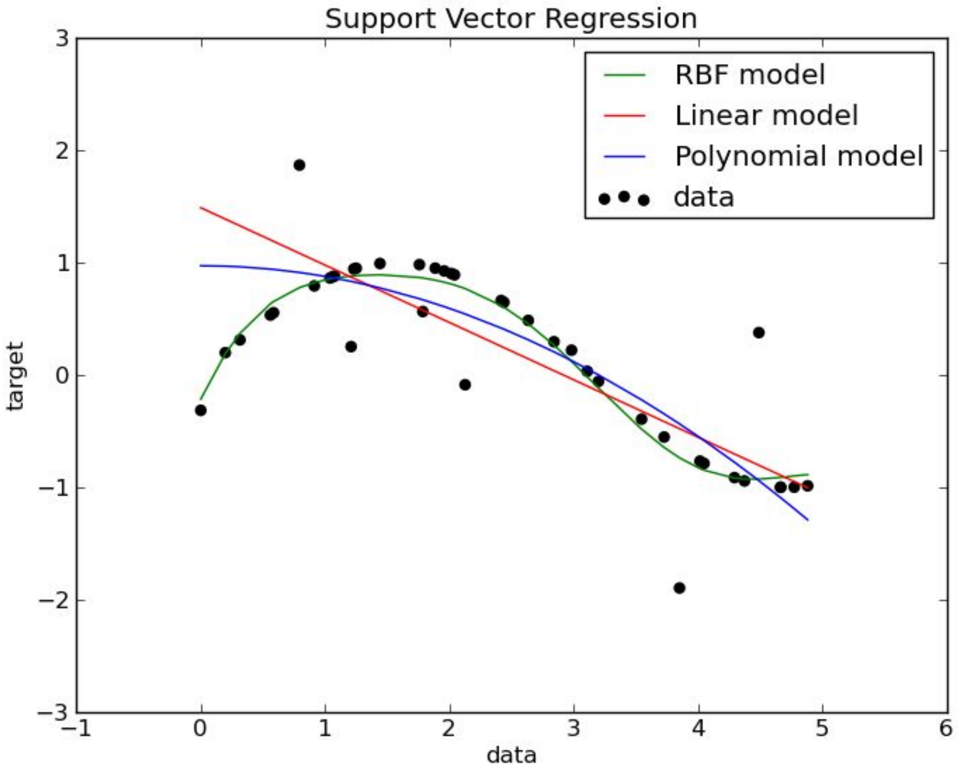


Figure 2. Support Vector Regressor. Adapted from "Support Vector Regression (SVR) using RBF kernel," by scikits-learn developers. Retrieved from http://scikit-learn.sourceforge.net/0.5/auto_examples/svm/plot_svm_regression.html. Copyright 2010 by scikits.learn developers.

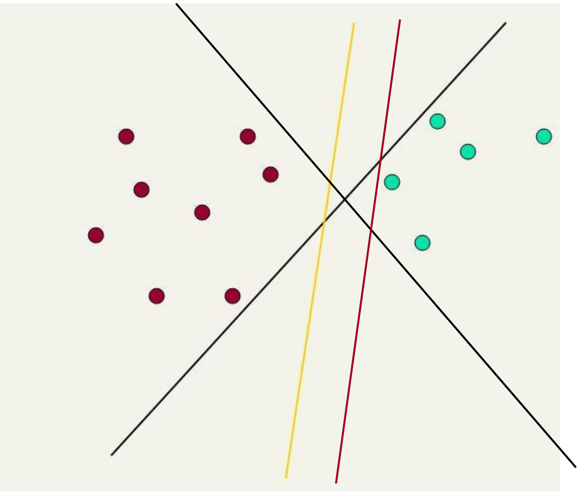
Support Vector Machines

(slides taken from https://web.stanford.edu/class/cs276/handouts/lecture14-SVMs.ppt)

Linear classifiers: Which Hyperplane?

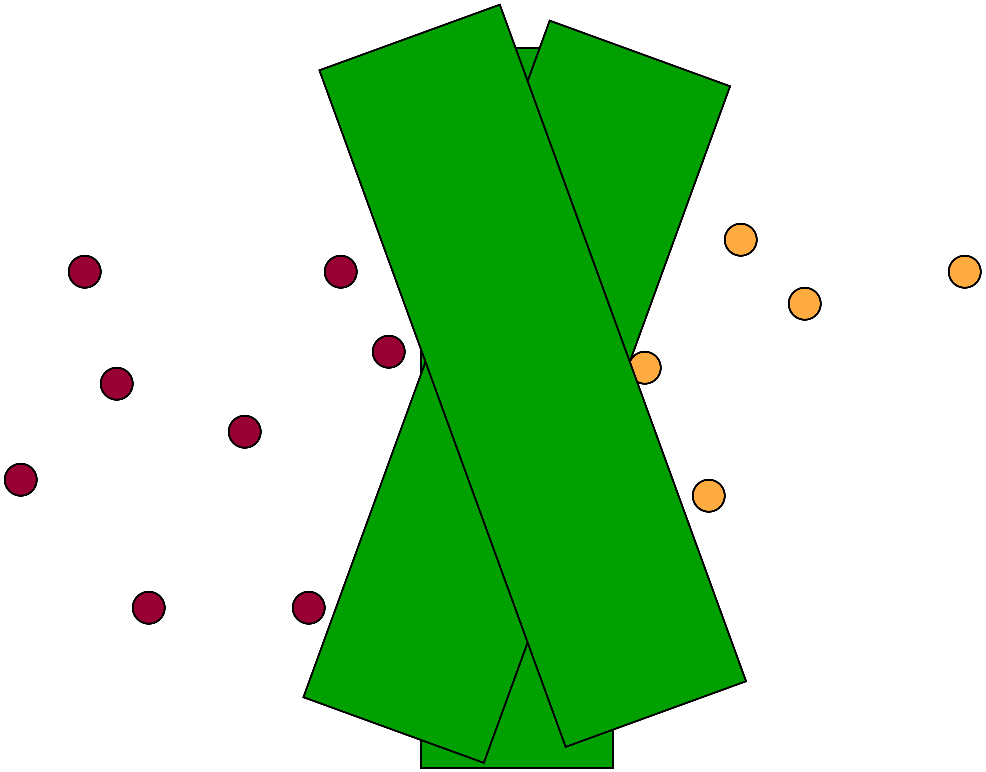
- Lots of possible solutions for *a, b, c.*
- Some methods find a separating hyperplane, but not the optimal one [according to some criterion of expected goodness]
  - E.g., perceptron
- Support Vector Machine (SVM) finds an optimal* solution.
  - Maximizes the distance between the hyperplane and the “difficult points” close to decision boundary
  - One intuition: if there are no points near the decision surface, then there are no very uncertain classification decisions

This line represents the decision boundary: $ax + by - c = 0$



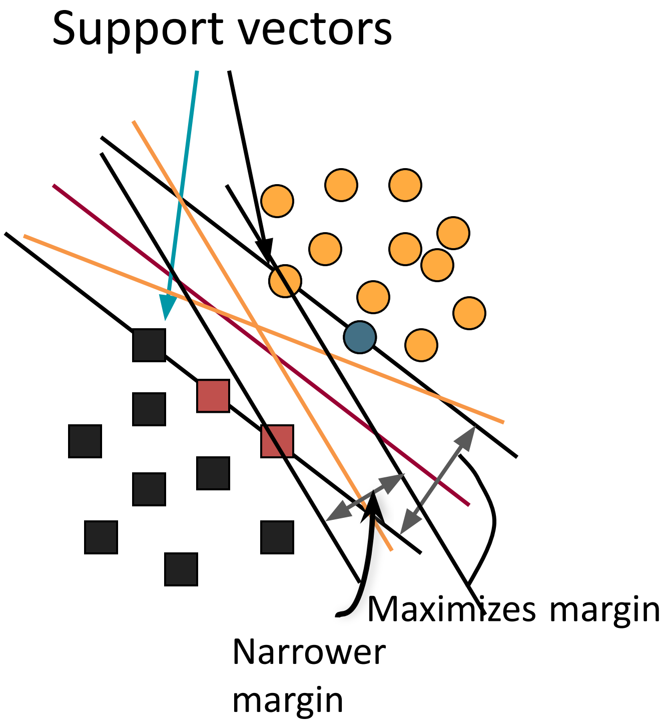
Another intuition

- If you have to place a fat separator between classes, you have less choices, and so the capacity of the model has been decreased



Support Vector Machine (SVM)

- SVMs maximize the *margin* around the separating hyperplane.
    - A.k.a. large margin classifiers
- The decision function is fully specified by a subset of training samples, *the support vectors*.
- Solving SVMs is a *quadratic programming* problem
- Seen by many as the most successful current text classification method*



Support vectors

Maximizes margin

Narrower margin

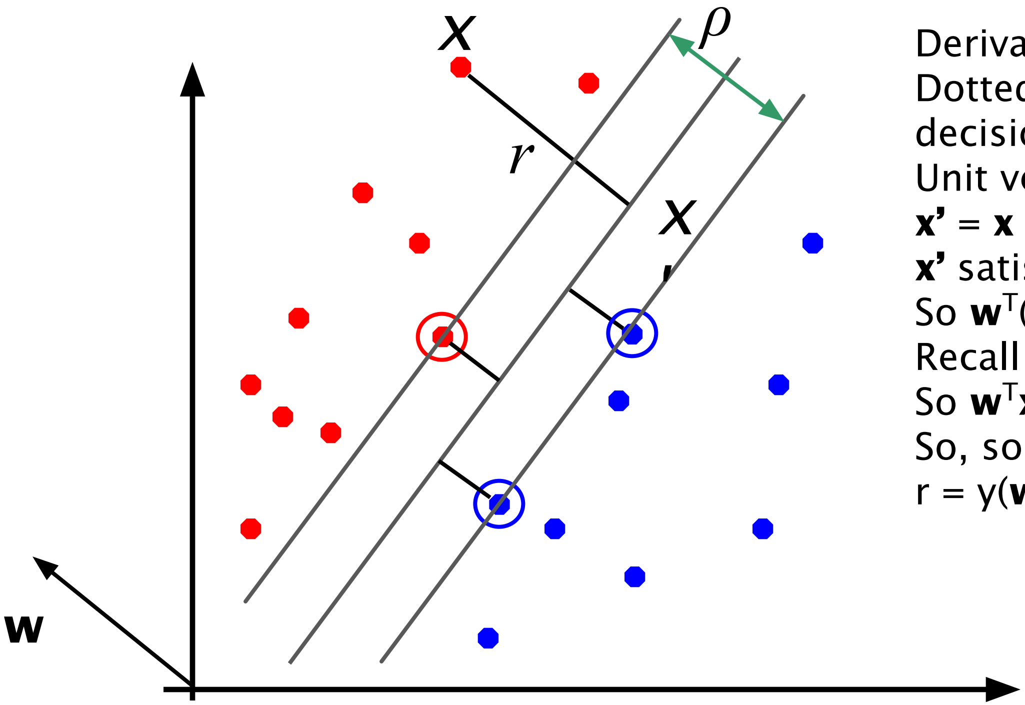
*but other discriminative methods often perform very similarly

Maximum Margin: Formalization

- **w**: decision hyperplane normal vector
- $\mathbf{x}_i$: data point *i*
- y_i: class of data point *i* (+1 or -1)   NB: Not 1/0
- Classifier is: $f(\mathbf{x}_i) = \text{sign}(\mathbf{w}^T\mathbf{x}_i + b)$
- Functional margin of $\mathbf{x}_i$ is: $y_i\,(\mathbf{w}^T\mathbf{x}_i + b)$
  - But note that we can increase this margin simply by scaling **w**, **b**….
- Functional margin of dataset is twice the minimum functional margin for any point
  - The factor of 2 comes from measuring the whole width of the margin

Geometric Margin

- Distance from example to the separator is $r = y\frac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are ***support vectors***.
- ***Margin*** *ρ* of the separator is the width of separation between support vectors of classes.



Derivation of finding *r:*
Dotted line **x'**−**x** is perpendicular to decision boundary so parallel to **w.**
Unit vector is **w**/|**w**|, so line is r**w**/|**w**|.
x' = **x** – yr**w**/|**w**|.
x' satisfies $\mathbf{w}^T\mathbf{x'}$+b = 0.
So $\mathbf{w}^T$(**x** –yr**w**/|**w**|) + b = 0
Recall that |**w**| = sqrt($\mathbf{w}^T\mathbf{w}$).
So $\mathbf{w}^T\mathbf{x}$ –yr|**w**| + b = 0
So, solving for r gives:
r = y($\mathbf{w}^T\mathbf{x}$ + b)/|**w**|

Linear SVM Mathematically

The linearly separable case

- Assume that all data is at least distance 1 from the hyperplane, then the following two constraints follow for a training set $\{(\mathbf{x_i}, y_i)\}$

$$\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b \geq 1 \quad \text{if } y_i = 1$$
$$\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b \leq -1 \quad \text{if } y_i = -1$$

- For support vectors, the inequality becomes an equality
- Then, since each example's distance from the hyperplane is

$$r = y\frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$$

- The margin is:

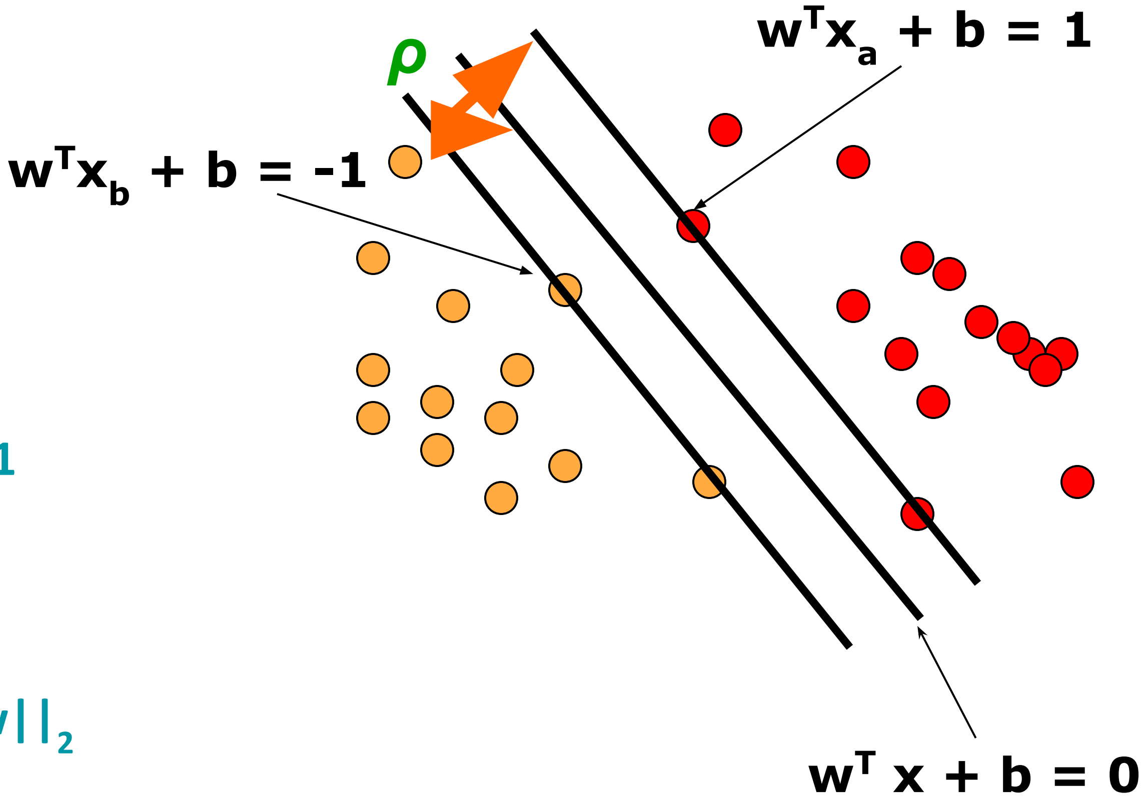
$$\rho = \frac{2}{\|\mathbf{w}\|}$$

Linear Support Vector Machine (SVM)

- **Hyperplane**
  $\mathbf{w}^T \mathbf{x} + b = 0$

- **Extra scale constraint**:
  $\min_{i=1,\ldots,n} |\mathbf{w}^T x_i + b| = 1$

- This implies:
  $w^T(x_a - x_b) = 2$
  $\rho = \|x_a - x_b\|_2 = 2/\|w\|_2$


ρ
$\mathbf{w}^T\mathbf{x}_a + \mathbf{b} = \mathbf{1}$
$\mathbf{w}^T\mathbf{x}_b + \mathbf{b} = \mathbf{-1}$
$\mathbf{w}^T \mathbf{x} + \mathbf{b} = \mathbf{0}$