Machine Learning: Introduction

Roberto Esposito
esposito@di.unito.it

These slides are based on Peter Flach's “Machine Learning” text book and on the slides made available with it

Teachers

Rosa Meo (meo@di.unito.it)

Phone: 011 6706817
Office hours: by appointment (preferably on Friday morning)
Home page: www.di.unito.it/~meo

Roberto Esposito (roberto.esposito@unito.it)

Phone: 011 6706714
Office hours: by appointment (preferably on Friday morning)
Home page: www.di.unito.it/~esposito

Course Outlook

Text book: Peter Flach. *Machine Learning. Cambridge University Press.*

- Roberto Esposito will cover chapters 1,2,3,7, and 11
- Prof. Meo will cover the rest of the book

Course Outlook — chapters taught by R. Esposito

- ✦ Introduction (~**2h theory**)
  - The ingredients of machine learning (Ch. 1)
- ✦ Tasks: *the problems that can be solved with machine learning* (~**4h theory + 2h lab**)
  - Binary classification (Ch. 2)
  - Beyond binary classification (Ch. 3)
- ✦ Models: *the output of machine learning*
  - Linear models: *least squares, SVM* (Ch. 7) *(~**8h theory + 2h lab**)*
  - Ensemble Learning: *Bagging, Boosting* (Ch. 11) *(~**4h theory + 2h lab**)*

Lab sessions

- Lab sessions at the end of each main topic.
- Lab sessions will be held using scikit-learn and python
- Basic programming skills in python are assumed

Lessons

- 6 lesson hours per week will be held.
- *Scheduling may change to ensure that the course finishes on time. We will keep you informed using the e-learning platform.*
- Lessons will be starting 15 minutes after their scheduled time.
- There will be a break at the end of the first hour.

Exams

Oral exam with both Professors. Each professor shall examine students over the part of the course she/he covered.

Questions

I encourage you to make all questions as soon as you do not understand something. Not only question allows you to not get lost during the lesson, they help me to better clarify difficult points.

I will try to be as clear as possible. Help me in such endeavor by letting me know when difficulties arise.

Slides

These slides are not meant to be a substitute of the suggested book. They are meant to help following the lesson and integrate the book material when necessary.

I shall upload the slide for each lesson on the e-learning platform the day before the lesson. You are free to print them and use during the lesson, but I discourage you to do so. *I firmly believe that taking your own notes help you better understand and learn the topics.*

I encourage you to use the slides only to integrate the notes after the lesson ends.

The ingredients of machine learning

Ingredients of machine learning

- **Tasks**: classification, regression, probability estimation, clustering, …
- **Models**: Linear, Decision Trees, Naive Bayes, Knn, …
- **Features**: numerical, categorical, feature construction, feature selection, ….

Assassinating spam e-mail

SpamAssassin is a widely used open-source spam filter. It calculates a score for an incoming e-mail, based on a number of built-in rules or ‘tests’ in SpamAssassin’s terminology, and adds a ‘junk’ flag and a summary report to the e-mail’s headers if the score is 5 or more.

```
-0.1 RCVD_IN_MXRATE_WL       RBL: MXRate recommends allowing
                             [123.45.6.789 listed in sub.mxrate.net]
0.6 HTML_IMAGE_RATIO_02      BODY: HTML has a low ratio of text to image area
1.2 TVD_FW_GRAPHIC_NAME_MID BODY: TVD_FW_GRAPHIC_NAME_MID
0.0 HTML_MESSAGE             BODY: HTML included in message
0.6 HTML_FONx_FACE_BAD       BODY: HTML font face is not a word
1.4 SARE_GIF_ATTACH          FULL: Email has a inline gif
0.1 BOUNCE_MESSAGE           MTA bounce message
0.1 ANY_BOUNCE_MESSAGE       Message is some kind of bounce message
1.4 AWL                      AWL: From: address is in the auto white-list
```

From left to right you see the score attached to a particular test, the test identifier, and a short description including a reference to the relevant part of the e-mail. As you see, scores for individual tests can be negative (indicating evidence suggesting the e-mail is ham rather than spam) as well as positive. The overall score of 5.3 suggests the e-mail might be spam.

Spam filtering as a *classification* task

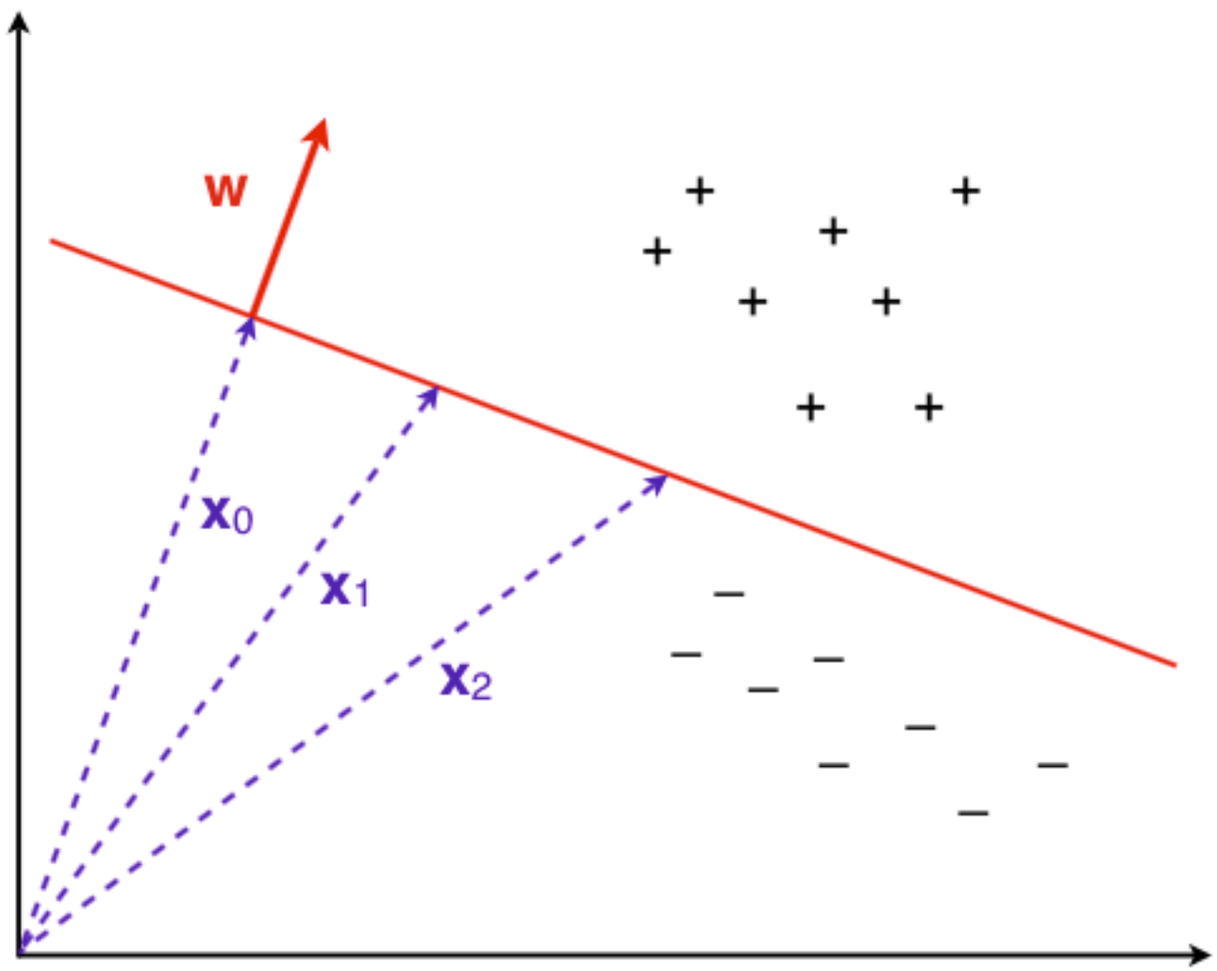
E-mail	x_1	x_2	Spam?	$4x_1 + 4x_2$
1	1	1	1	8
2	0	0	0	0
3	1	0	0	4
4	0	1	0	4

The columns marked x_1 and x_2 indicate the results of two tests on four different e-mails. The fourth column indicates which of the e-mails are spam. The right-most column demonstrates that by thresholding the function $4x_1 + 4x_2$ at 5, we can separate spam from ham.

Linear classification in two dimensions

- The straight line separates the positives from the negatives. It is defined by $\mathbf{w} \cdot \mathbf{x}_i = t$, where $\mathbf{w}$ is a vector perpendicular to the decision boundary and pointing in the direction of the positives.

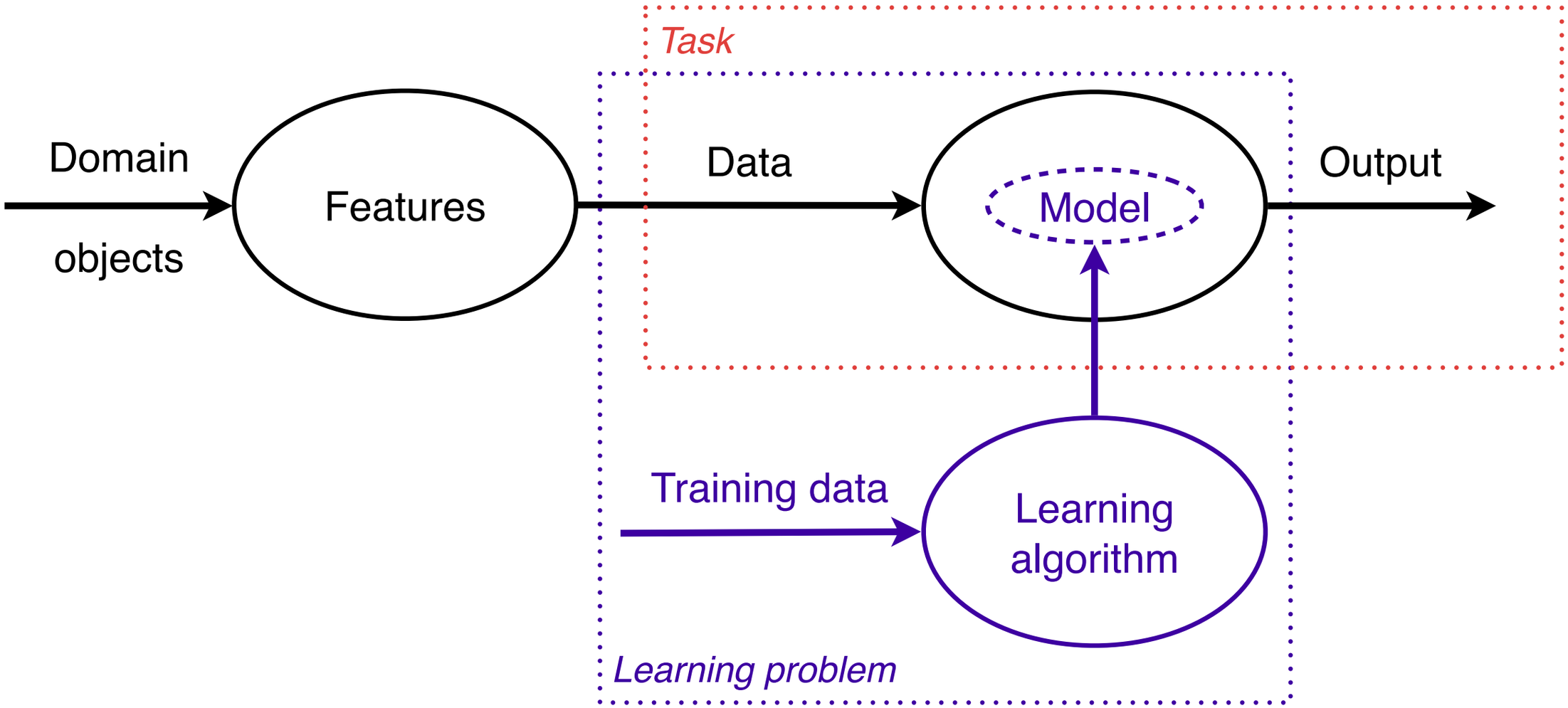
- All points in the plane for which $\mathbf{w} \cdot \mathbf{x}_i \geq t$ are classified as “positive” examples, all the other points are classified as “negative”.



Machine Learning

Machine learning is the *systematic* study of algorithms and systems that *improve* their *knowledge or performance* with *experience*.

How machine learning helps to solve a task



An overview of how machine learning is used to address a given task. A task (red box) requires an appropriate mapping – a model – from data described by features to outputs. Obtaining such a mapping from training data is what constitutes a learning problem (blue box).

Important point to remember

Tasks are *addressed by models*, whereas learning problems are solved by learning algorithms that produce models.

Important point to remember

Machine learning is concerned with using the right ***features*** to build the right ***models*** that achieve the right ***tasks***.

Tasks

Tasks for machine learning

The most common machine learning tasks are **predictive**, in the sense that they concern predicting a target variable from features.

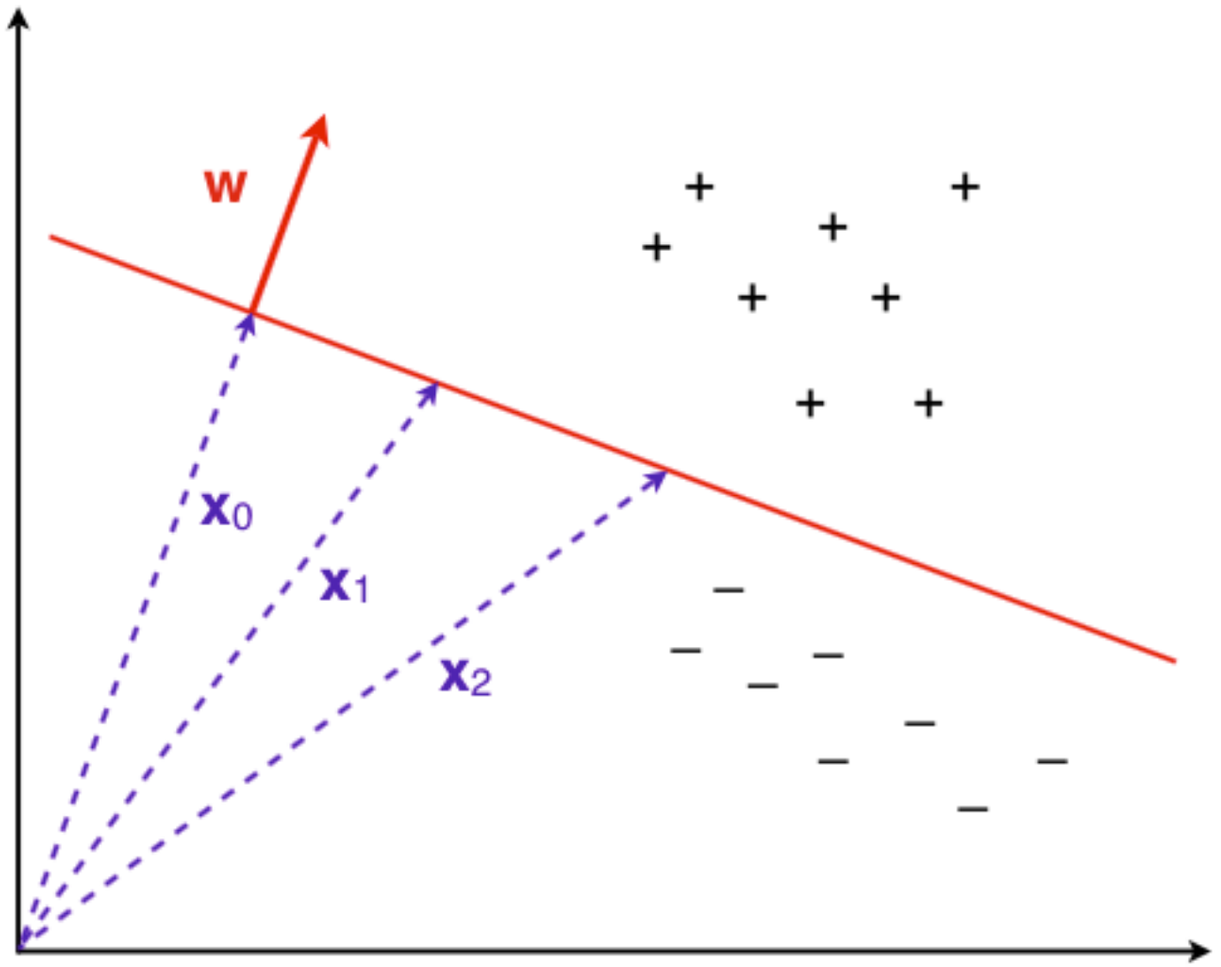
- ✦ Binary and multi-class classification: categorical target
- ✦ Regression: numerical target
- ✦ Clustering: hidden target

Descriptive tasks are concerned with exploiting underlying structure in the data.

Tasks: predictive

Predictive models

Predictive models are concerned with the problem of infer some knowledge on new problem instances based on past experience



Overfitting

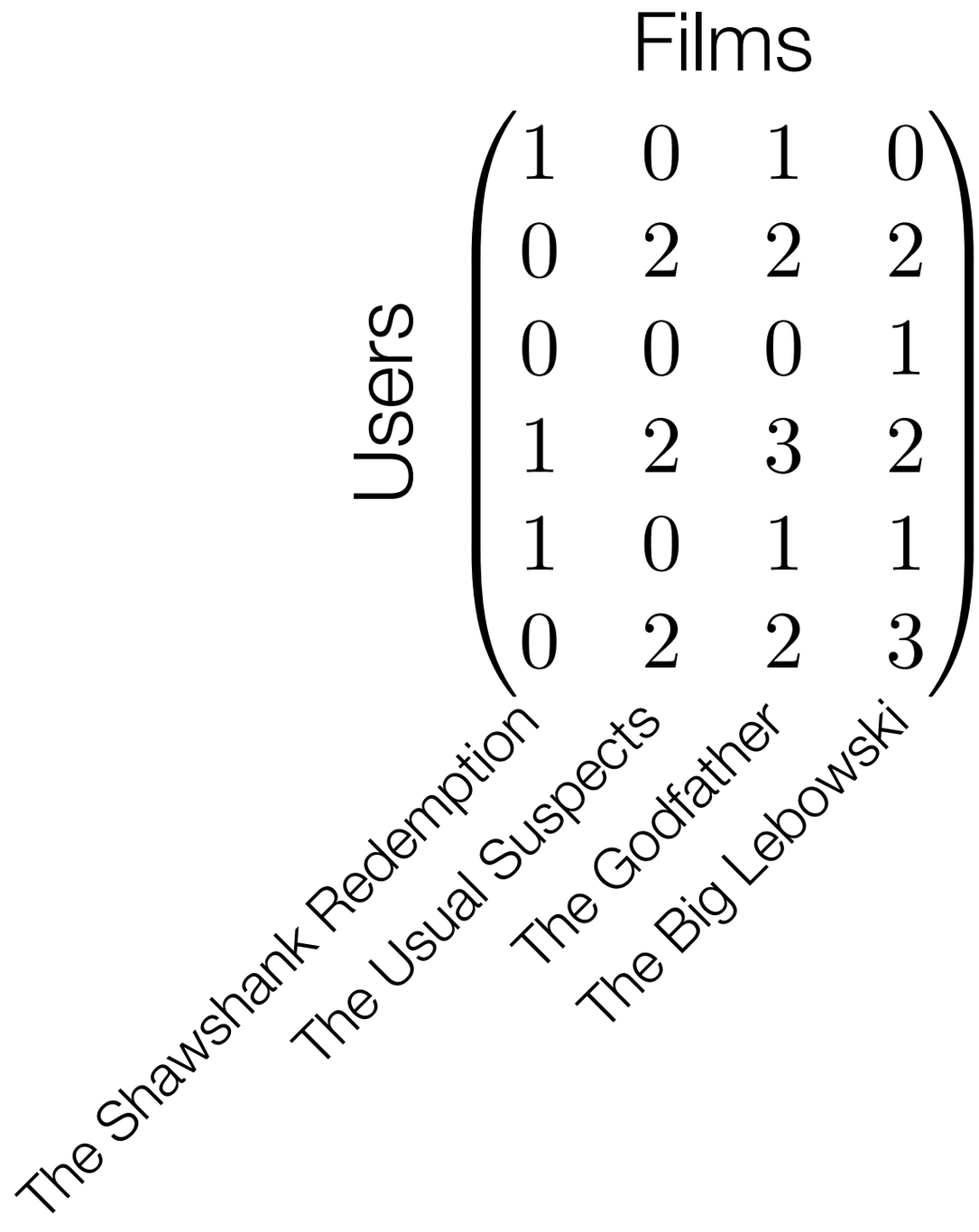
Imagine you are preparing for your Machine Learning exam. Helpfully, the professor has made previous exam papers and their worked answers available online. You begin by trying to answer the questions from previous papers and comparing your answers with the model answers provided.

Unfortunately, you get carried away and spend all your time on memorizing the model answers to all past questions. Now, if the upcoming exam completely consists of past questions, you are certain to do very well. But if the new exam asks different questions about the same material, you would be ill-prepared and get a much lower mark than with a more traditional preparation.

In this case, one could say that you were overfitting the past exam papers and that the knowledge gained didn’t generalize to future exam questions.

Tasks: descriptive

Descriptive task example: predicting users film preferences



Descriptive task example

Genres

Films

Users

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

drama
crime
comedy

The Shawshank Redemption
The Usual Suspects
The Godfather
The Big Lebowski

Machine learning settings

	Predictive model	Descriptive model
Supervised learning	classification, regression	subgroup discovery
Unsupervised learning	predictive clustering	descriptive clustering, association rule discovery

Models

Machine learning models

Machine learning models can be distinguished according to their approach:

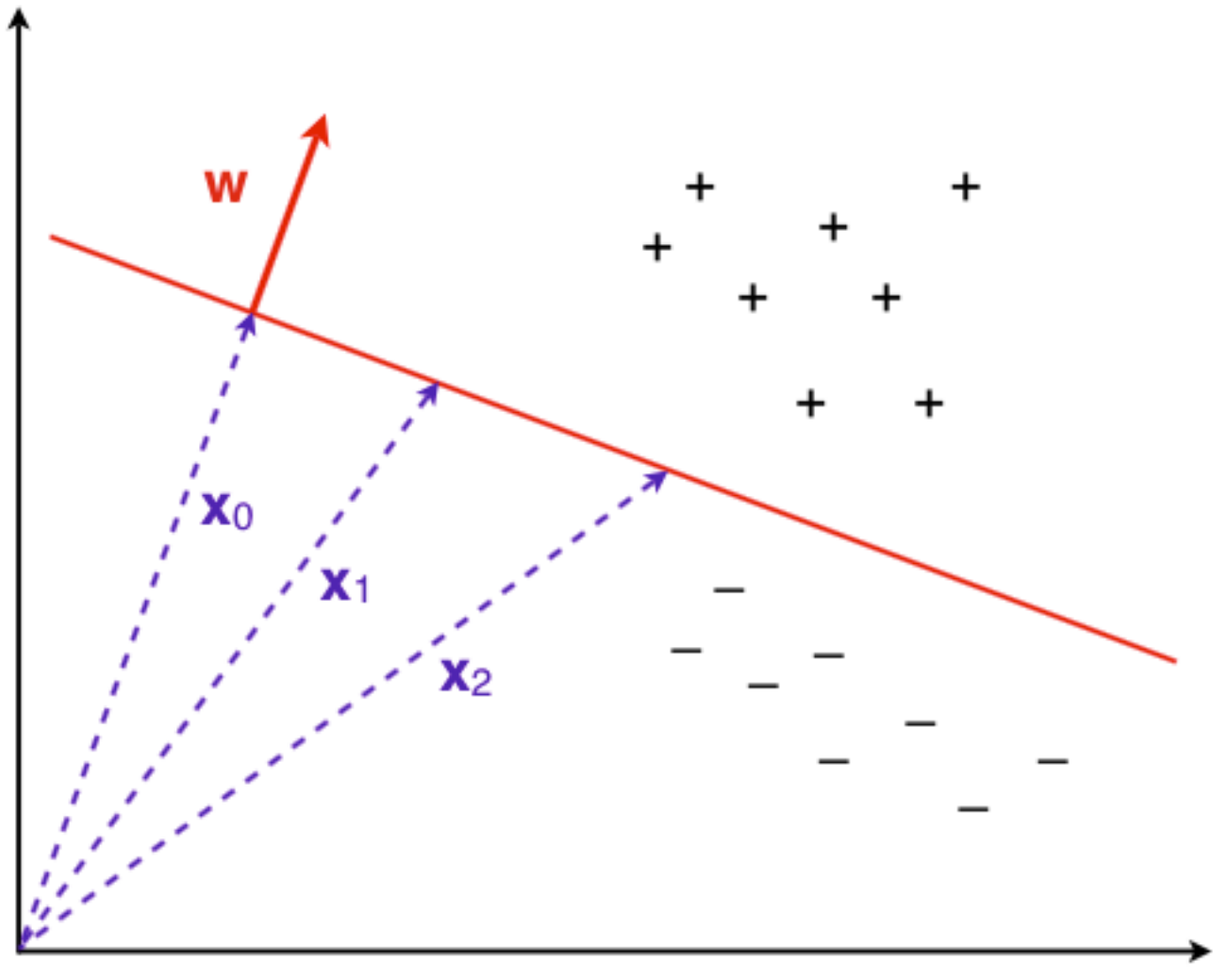
- Geometric models use intuitions from geometry such as separating (hyper-)planes, linear transformations and distance metrics.
- Probabilistic models view learning as a process of reducing uncertainty, modelled by means of probability distributions.
- Logical models are defined in terms of easily interpretable logical expressions.

Models: geometric classifiers

Linear classification in two dimensions

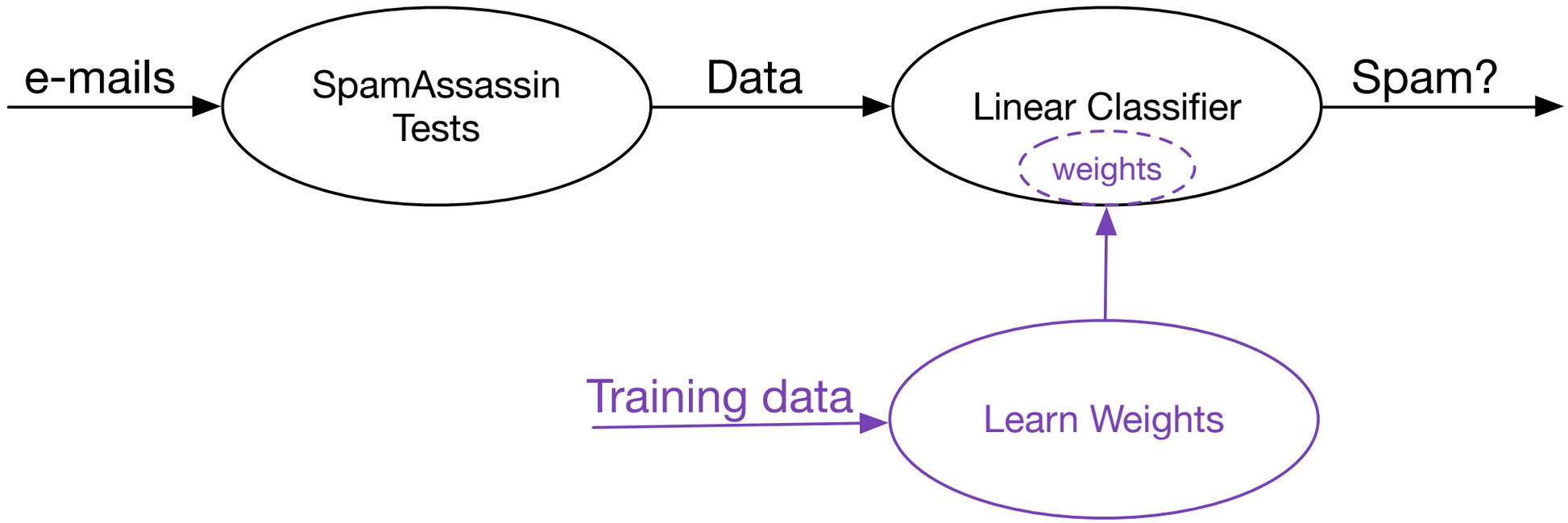
- The straight line separates the positives from the negatives. It is defined by $\boldsymbol{w} \cdot \boldsymbol{x}\mathrm{i} = t$, where $\boldsymbol{w}$ is a vector perpendicular to the decision boundary and pointing in the direction of the positives.

- All points in the plane for which $\boldsymbol{w} \cdot \boldsymbol{x}\mathrm{i} \geq t$ are classified as “positive” examples, all the other points are classified as “negative”.



Machine Learning

Machine learning is the *systematic* study of algorithms and systems that *improve* their *knowledge or performance* with *experience*.



Models: probabilistic classifiers

A probabilistic classifier

Probabilistic classifiers estimate probabilities from the data and use those estimations to make predictions using a decision rule like the following one:

$$
\begin{aligned}
Y_{\mathrm{MAP}} &= \arg\max_{Y} P(Y|X) \\
&= \arg\max_{Y} \frac{P(X|Y)P(Y)}{P(X)} \\
&= \arg\max_{Y} P(X|Y)P(Y)
\end{aligned}
$$

If priors are unknown or unimportant this can be simplified to:

$$
Y_{\mathrm{ML}} = \arg\max_{Y} P(X|Y)
$$

A probabilistic classifier: an example

Assume that spam e-mail is 30% of the total (that is: P(spam)=0.3) and that the following probabilities have been estimated from past e-mails

X_1=viagra	X_2=lottery	$P(X_1X_2\|$ spam)	$P(X_1X_2\|$ ¬spam)	$P(X_1,X_2)$
0	0	0.033	0.857	0.61
0	1	0.067	0.086	0.08
1	0	0.233	0.043	0.1
1	1	0.667	0.014	0.21

A probabilistic classifier: MAP decision rule

X_1 X_2	P(spam|X_1X_2)	Y_{MAP}
0 0	0.0164	not spam
0 1	0.25	not spam
1 0	0.7	spam
1 1	0.95	spam

$$P(\text{spam}|X_1X_2) \geq 0.5$$

$$\arg \max_{Y \in \{\text{spam}, \neg\text{spam}\}} P(X_1X_2|Y)P(Y)$$

X_1 X_2	P(X_1X_2|spam)P(spam)	P(X_1X_2|¬spam)P(¬spam)	Y_{MAP}
0 0	0.033 * 0.3 = 0.0099	0.857 * 0.7 = **0.5999**	not spam
0 1	0.067 * 0.3 = 0.0201	0.086 * 0.7 = **0.0602**	not spam
1 0	0.233 * 0.3 = **0.0699**	0.043 * 0.7 = 0.0301	spam
1 1	0.667 * 0.3 = **0.2001**	0.014 * 0.7 = 0.0098	spam

A probabilistic classifier: MAP vs ML decision rules

X_1 X_2	$P(X_1X_2\|spam)P(spam)$	$P(X_1X_2\|\neg spam)P(\neg spam)$	Y_{MAP}
0 0	0.033 * 0.3 = 0.0099	0.857 * 0.7 = **0.5999**	not spam
0 1	0.067 * 0.3 = 0.0201	0.086 * 0.7 = **0.0602**	not spam
1 0	0.233 * 0.3 = **0.0699**	0.043 * 0.7 = 0.0301	spam
1 1	0.667 * 0.3 = **0.2001**	0.014 * 0.7 = 0.0098	spam

X_1 X_2	$P(X_1X_2\|spam)$	$P(X_1X_2\|\neg spam)$	Y_{ML}
0 0	0.033	**0.857**	not spam
0 1	0.067	**0.086**	not spam
1 0	**0.233**	0.043	spam
1 1	**0.667**	0.014	spam

A probabilistic classifier: Naive Bayes Assumption

If we assume that X1 and X2 are independent given spam, we can now simplify the decision function and limit ourselves to the estimation of the following:

X_1	$P(X_1\|spam)$	$P(X_1\|\neg spam)$
0	0.1	0.943
1	0.9	0.057

X_2	$P(X_2\|spam)$	$P(X_2\|\neg spam)$
0	0.27	0.9
1	0.73	0.1

A probabilistic classifier: an example

X_1	$P(X_1\|spam)$	$P(X_1\|\neg spam)$
0	0.1	0.943
1	0.9	0.057

X_2	$P(X_2\|spam)$	$P(X_2\|\neg spam)$
0	0.27	0.9
1	0.73	0.1

$$\arg \max_{Y \in \{\text{spam}, \neg\text{spam}\}} P(X_1|Y)P(X_2|Y)P(Y)$$

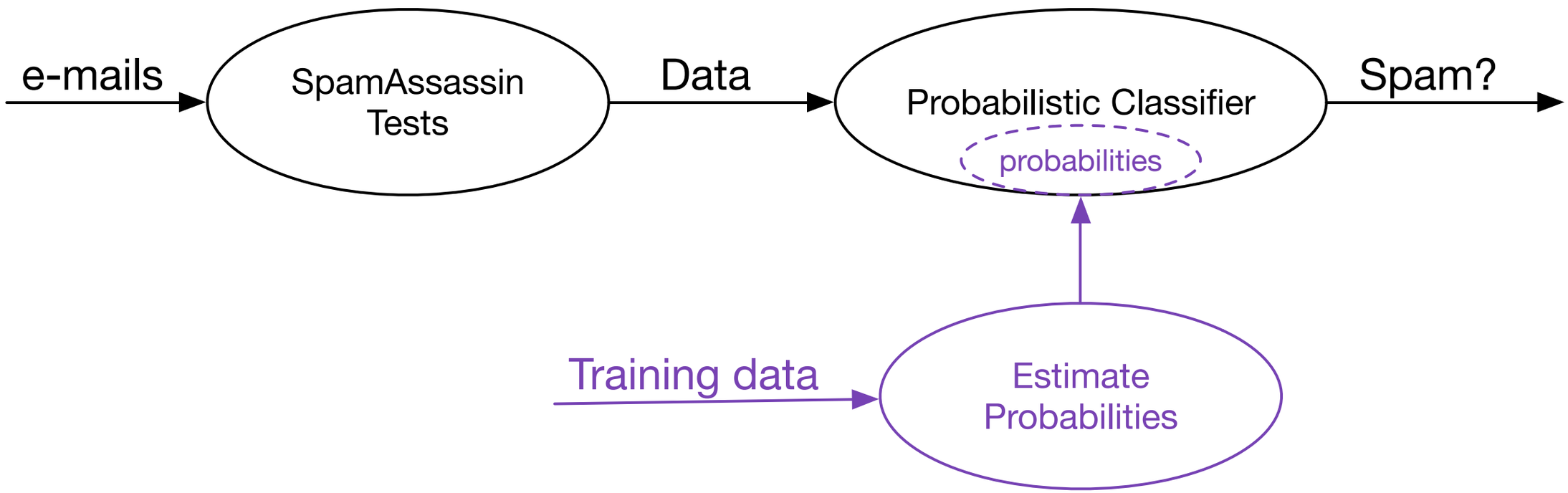
X_1 X_2	$P(X_1\|spam)P(X_2\|spam)P(spam)$	$P(X_1\|\neg spam)P(X_2\|\neg spam)P(\neg spam)$	Y_{MAP}
0 0	0.1 * 0.27 * 0.3 = 0.0081	0.943 * 0.9 * 0.7 = **0.594**	not spam
0 1	0.1 * 0.73 * 0.3 = 0.0219	0.943 * 0.1 * 0.7 = **0.066**	not spam
1 0	0.9 * 0.27 * 0.3 = **0.0729**	0.057 * 0.9 * 0.7 = 0.036	spam
1 1	0.9 * 0.73 * 0.3 = **0.1971**	0.057 * 0.1 * 0.7 = 0.004	spam

Probabilistic Classifier examples: Joint Distribution

In all examples, the following joint distribution has been used to calculate the reported conditional and marginal probabilities.

Viagra	Lottery	Spam	Probability
0	0	0	0.6
0	0	1	0.01
0	1	0	0.06
0	1	1	0.02
1	0	0	0.03
1	0	1	0.07
1	1	0	0.01
1	1	1	0.2

Machine Learning



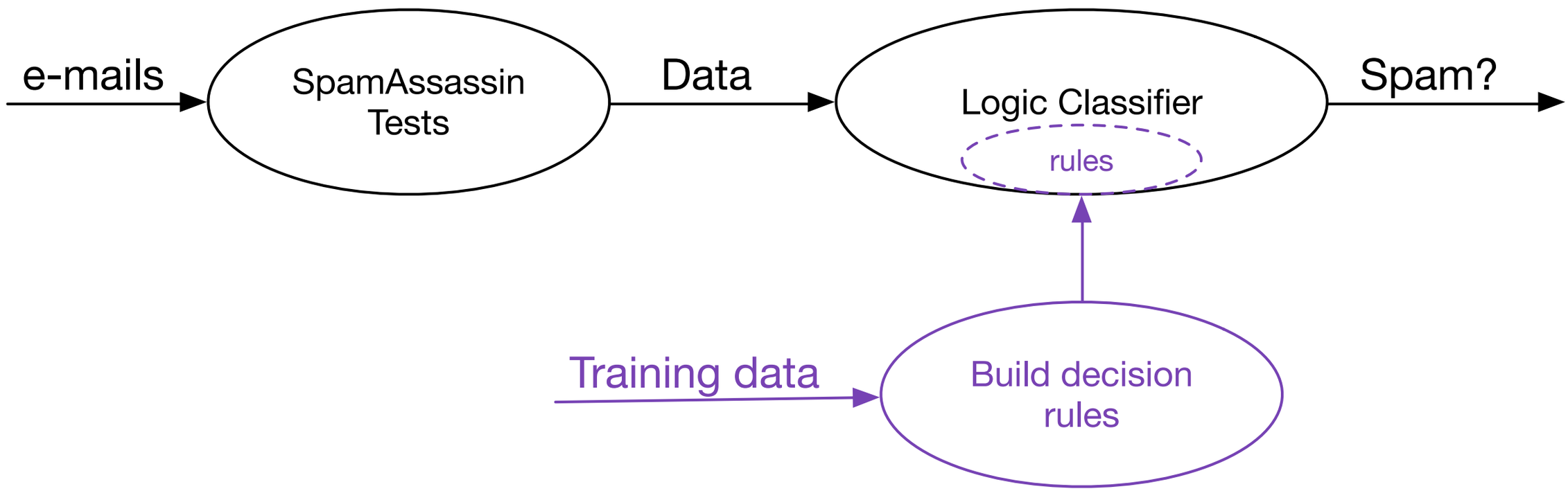
Models: logic classifiers

Logic Classifier: an example

- if the e-mail contains the word ‘Viagra’ then estimate the odds of spam as 4:1;
- otherwise, if it contains the phrase ‘blue pill’ then estimate the odds of spam as 3:1;
- otherwise, estimate the odds of spam as 1:6.

The first rule covers all e-mails containing the word ‘Viagra’, regardless of whether they contain the phrase ‘blue pill’, so no overcounting occurs. The second rule only covers e-mails containing the phrase ‘blue pill’ but not the word ‘Viagra’, by virtue of the ‘otherwise’ clause. The third rule covers all remaining e-mails: those which neither contain neither ‘Viagra’ nor ‘blue pill’.

Machine Learning

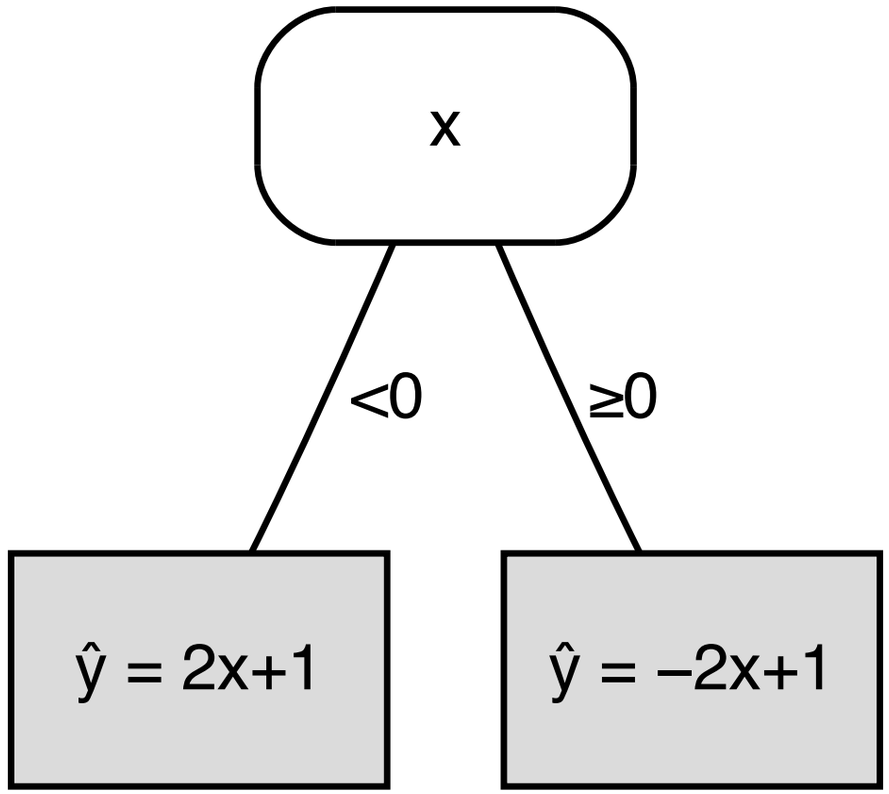


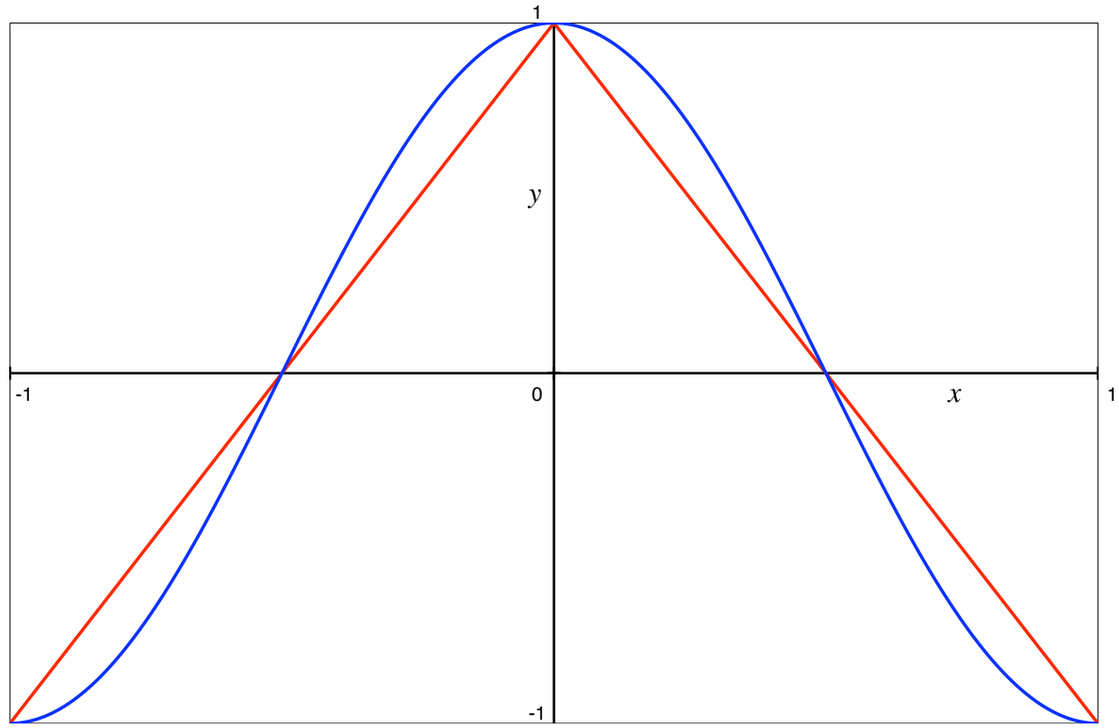
Features

Two uses of features

Suppose we want to approximate $y = cos\ \pi x$ on the interval $-1 \leq x \leq 1$. A linear approximation is not much use here, since the best fit would be $y = 0$. However, if we split the x-axis in two intervals $-1 \leq x < 0$ and $0 \leq x \leq 1$, we could find reasonable linear approximations on each interval. We can achieve this by using x both as a splitting feature and as a regression variable.

Two uses of features

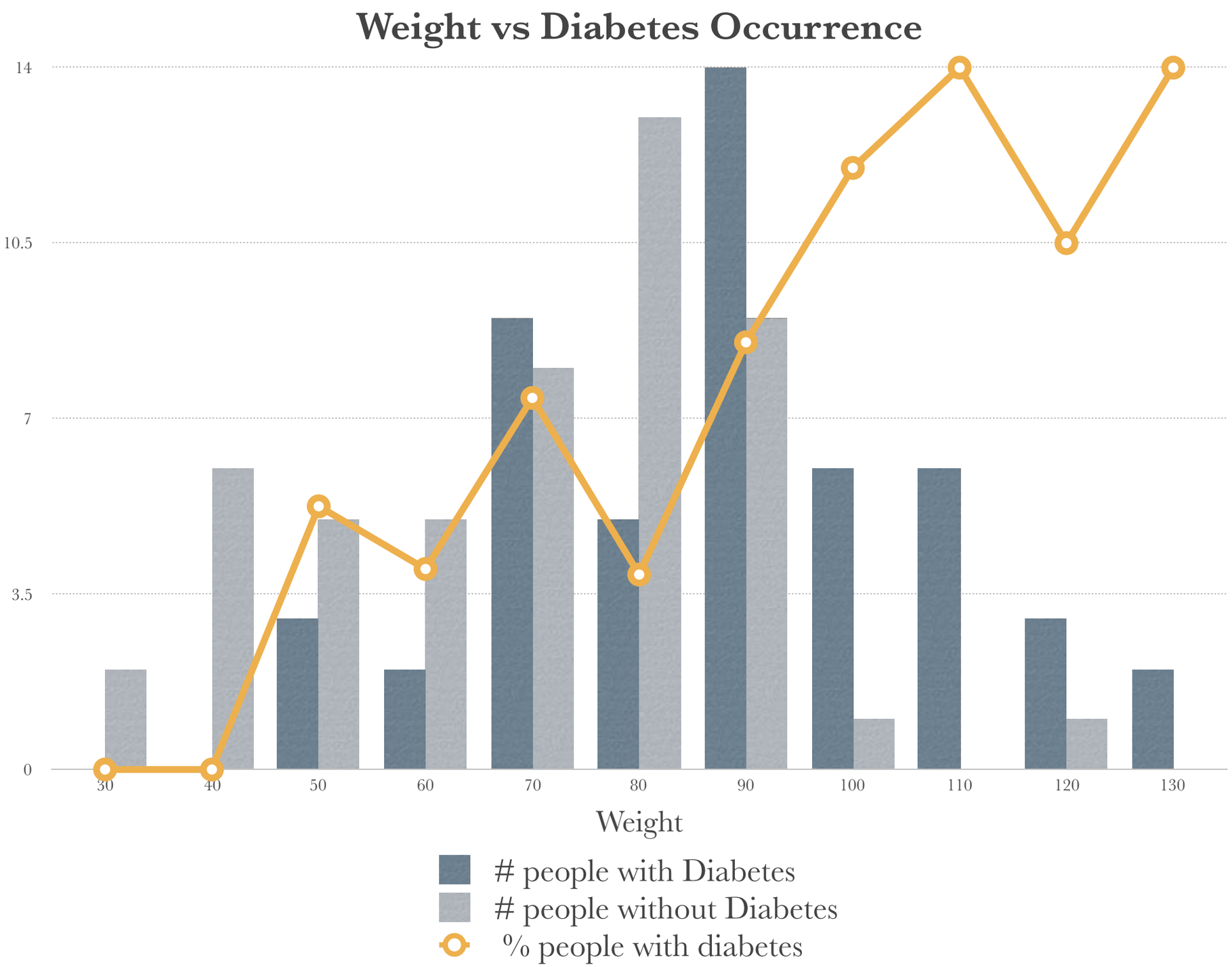




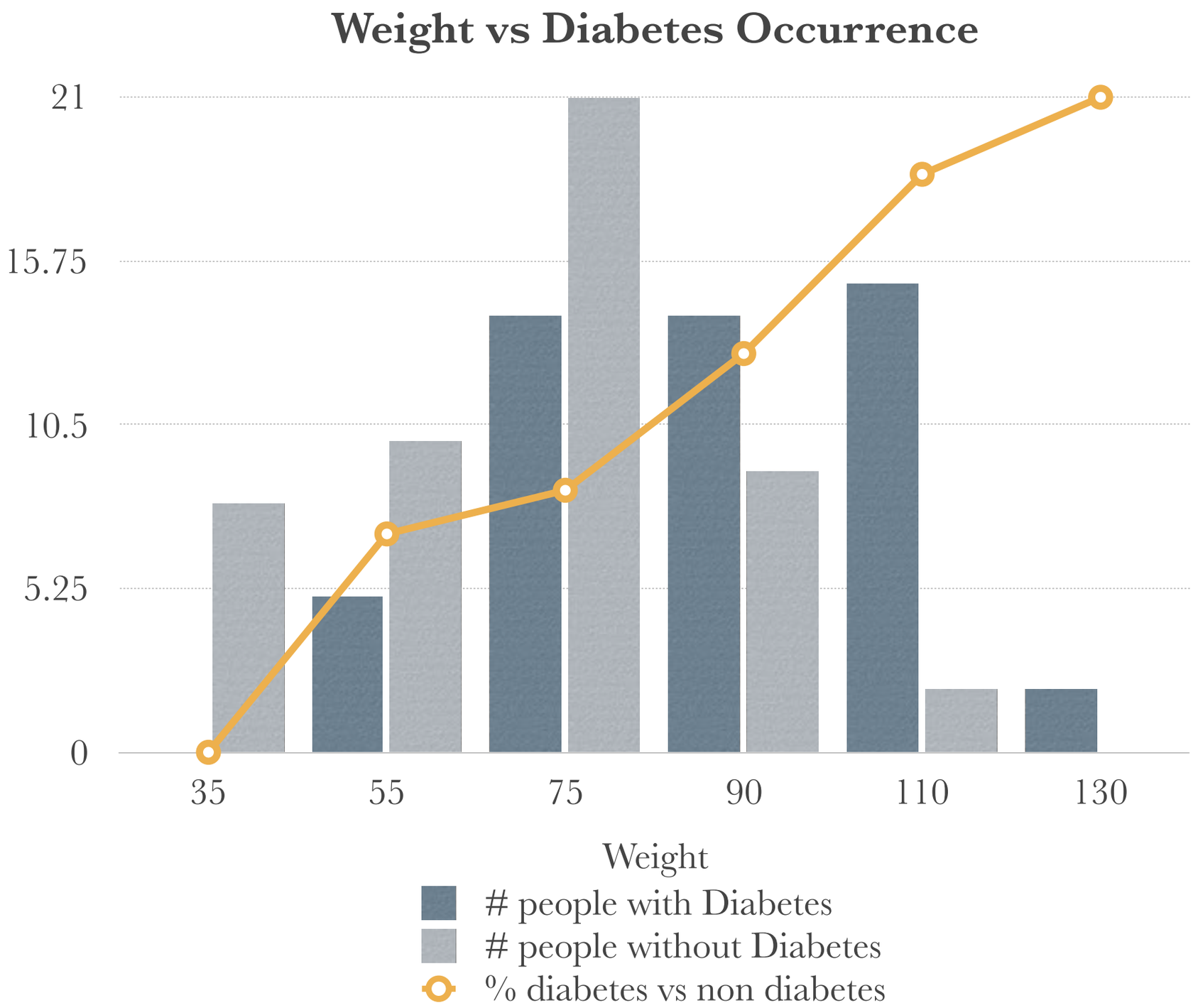
(left) A regression tree combining a one-split feature tree with linear regression models in the leaves. Notice how x is used as both a ***splitting*** feature and a ***regression*** variable.

(right) The function $y = cos\ \pi x$ on the interval $-1 \leq x \leq 1$, and the piecewise linear approximation achieved by the regression tree.

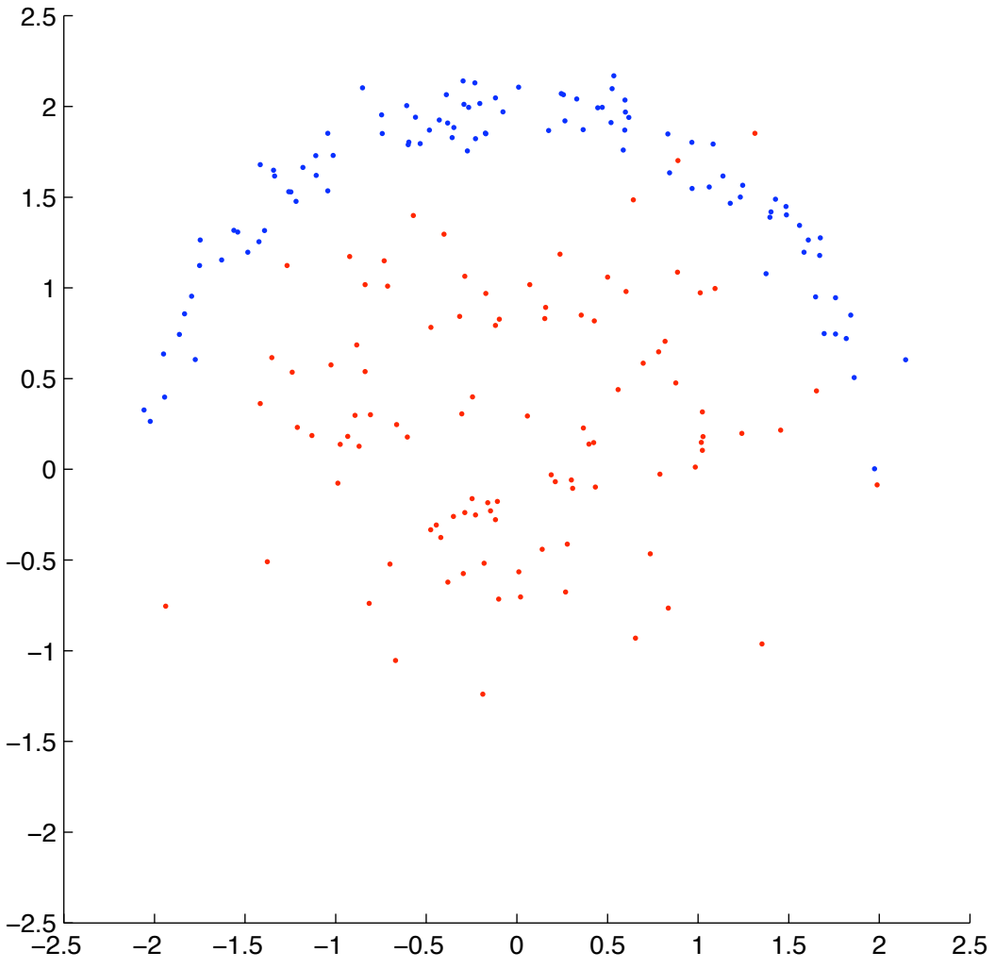
Class sensitive discretization

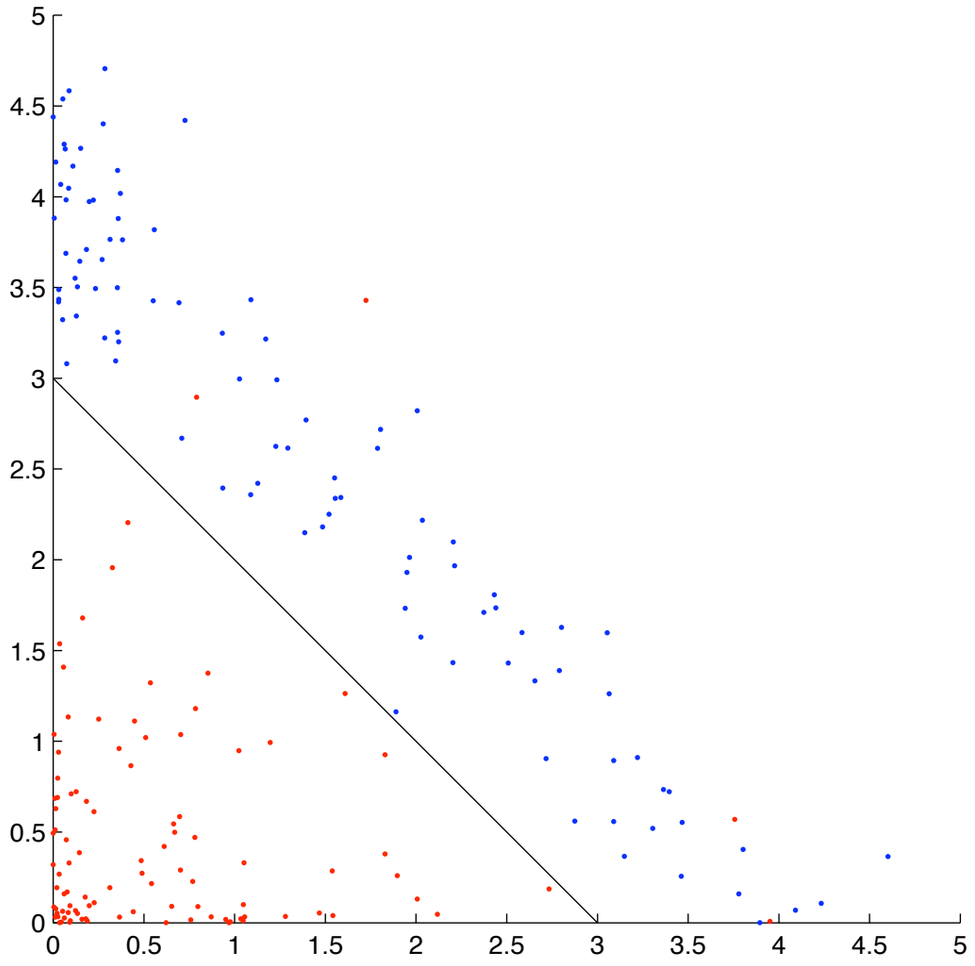


Class-sensitive discretisation



Mapping into new spaces





(left) A linear classifier would perform poorly on this data. **(right)** By transforming the original (x, y) data into $(x',y') = (x^2,y^2)$, the data becomes more ‘linear’, and a linear decision boundary $x' + y' = 3$ separates the data fairly well. In the original space this corresponds to a circle with radius $\sqrt{3}$ around the origin.

The kernel trick

Let us consider two data points:

$$\mathbf{x}_1 = (x_1, y_1) \qquad \mathbf{x}_2 = (x_2, y_2)$$

and consider the following mapping to a three-dimensional feature space

$$(x, y) \rightarrow (x^2, y^2)$$

the points in the feature space corresponding to $\mathbf{x}_1$ and $\mathbf{x}_2$ are:

$$\mathbf{x}'_1 = (x_1^2, y_1^2) \qquad \mathbf{x}'_2 = (x_2^2, y_2^2)$$

and their dot product is:

$$\mathbf{x}'_1 \cdot \mathbf{x}'_2 = x_1^2 x_2^2 + y_1^2 y_2^2$$

That is: we can evaluate the dot product in feature space without actually constructing the feature vectors!

The kernel trick