Data Analysis: Predictive Models

Technological University Dublin Tallaght Campus

Department of Computing

Models

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	**numeric**
Independent variable[S]/ feature[M] type	**categorical**	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

- Models are used for predicting a variable, called
  - *dependent variable* in the context of statistics
  - *target variable* in the context of machine learning
- They are built using properties of data called
  - *independent variables* in the context of statistics
  - *features* in the context of machine learning

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Categorical Naïve Bayes

- A method for estimating the probabilities of target variable values.
- Complete independence among all attributes is rarely possible in real data sets but the predictions made by the Naïve Bayes family of models are nevertheless very good for practical purposes.
- The Naïve Bayes formula for **the probability of a data instance belonging to class** C_j (any target feature value), given a certain combination of attribute values, E_i (consisting of a value for each attribute, i.e. x_{1i}, $x_{2i}...x_{n_A i}$ if there are n_A attributes), is:

$$P(C_j|E_i) = \frac{P(C_j)\prod_{a=1}^{n_A} P(x_{ai}|C_j)}{P(E_i)}$$

where n_A is the number of predictor attributes, x_{ai} is the value of attribute a in instance i, $P(C_j)$ is the proportion of instances in the data set with target value C_j, $P(x_{ai}|C_j)$ is the proportion of instances with target value C_j that have x_{ai} as the value for attribute a and $P(E_i)$ is the proportion of instances in the set that have attribute values equal to E_i.

- The formula above assumes independence among the predictor attributes. As the assumption is often not valid, the model is labelled *Naïve* (despite this the model can work well).
- In practice, deriving $P(E_i)$ from a data set may not be possible since the combination of values E_i may not exist in the data set, especially with many attributes. As we will see from the practical example, this proportion (as an estimation of probability) is not necessary for calculating $P(C_j|E_i)$.
- The model can be used both with numeric and categorical attributes. Probabilities for categorical attributes can be estimated using counts. For numeric attributes probability densities for a suitable distribution (e.g. normal or Poisson) are used.

Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Source: [DM]

First, let's look at the table and see what we can say about whether the game will be played if it is **Sunny**. There are five instances that have the value of **Sunny** for `Outlook` and of those 3 are associated with a **No** for `Play` and 2 with a **Yes**. In terms of probabilities, we can write this as:

$$P(\mathbf{Yes}|\mathbf{Sunny}) = \frac{2}{5} \qquad P(\mathbf{No}|\mathbf{Sunny}) = \frac{3}{5}$$

Based on these values our prediction might be that if it is sunny, the game is more likely not to go ahead than to go ahead. However, we have used only one of the four attributes of the weather availabe to us. It would make more sense to use all data available i.e. all four attributes to make the prediction. What if one day the attribute values were: `Outlook`=**Sunny**, `Temperature`=**Cool**, `Humidity`=**High**, `Windy`=**True** and we wanted to see how likely it was that the game would go ahead? We don't even have a case like that in the table, let alone several cases to use for counting up incidences of **Yes** and **No**. This is where Naïve Bayes comes in:

$$P(\mathbf{Yes}|\mathbf{E}) = \frac{P(\mathbf{E}|\mathbf{Yes})P(\mathbf{Yes})}{P(\mathbf{E})} \qquad P(\mathbf{No}|\mathbf{E}) = \frac{P(\mathbf{E}|\mathbf{No})P(\mathbf{No})}{P(\mathbf{E})} \tag{1}$$

where **E** is used as shorthand for the 'evidence', consisting of the set of values $\{\mathbf{Sunny}, \mathbf{Cool}, \mathbf{High}, \mathbf{True}\}$, hence $P(\mathbf{E}) = P(\mathbf{Sunny}, \mathbf{Cool}, \mathbf{High}, \mathbf{True})$.

The assumption of independence of attributes allows us to simply multiply the individual probabilities, whether conditional or not:

$$P(\mathbf{E}|\mathbf{Yes}) = P(\mathbf{Sunny}|\mathbf{Yes})P(\mathbf{Cool}|\mathbf{Yes})P(\mathbf{High}|\mathbf{Yes})P(\mathbf{True}|\mathbf{Yes}) \qquad P(\mathbf{E}) = P(\mathbf{Sunny})P(\mathbf{Cool})P(\mathbf{High})P(\mathbf{True})$$

and these individual probabilities *can* be estimated using the data in the table.

Finally, because **Yes** and **No** are the only two possible outcomes, $P(\mathbf{Yes}|\mathbf{E}) + P(\mathbf{No}|\mathbf{E}) = P(\mathbf{E})$. From this and equation (1) it follows that

$$\frac{P(\mathbf{E}|\mathbf{Yes})P(\mathbf{Yes})}{P(\mathbf{E})} + \frac{P(\mathbf{E}|\mathbf{No})P(\mathbf{No})}{P(\mathbf{E})} = 1$$

and further that

$$P(E) = P(\mathbf{E}|\mathbf{Yes})P(\mathbf{Yes}) + P(\mathbf{E}|\mathbf{No})P(\mathbf{No})$$

Substituting this into equations (1) we can reduce the probabilities that need to be estimated to the conditional probabilities:

$$P(\mathbf{Yes}|\mathbf{E}) = \frac{P(\mathbf{E}|\mathbf{Yes})P(\mathbf{Yes})}{P(\mathbf{E}|\mathbf{Yes})P(\mathbf{Yes}) + P(\mathbf{E}|\mathbf{No})P(\mathbf{No})} \qquad P(\mathbf{No}|\mathbf{E}) = \frac{P(\mathbf{E}|\mathbf{No})P(\mathbf{No})}{P(\mathbf{E}|\mathbf{Yes})P(\mathbf{Yes}) + P(\mathbf{E}|\mathbf{No})P(\mathbf{No})}$$

Using the counts from the table to estimate the conditional probabilities we get:

$$P(\mathbf{E}|\mathbf{Yes})P(\mathbf{Yes}) = P(\mathbf{Sunny}|\mathbf{Yes})P(\mathbf{Cool}|\mathbf{Yes})P(\mathbf{High}|\mathbf{Yes})P(\mathbf{True}|\mathbf{Yes})P(\mathbf{Yes}) = \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} = 0.0053$$

$$P(\mathbf{E}|\mathbf{No})P(\mathbf{No}) = P(\mathbf{Sunny}|\mathbf{No})P(\mathbf{Cool}|\mathbf{No})P(\mathbf{High}|\mathbf{No})P(\mathbf{True}|\mathbf{No})P(\mathbf{No}) = \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} = 0.0206$$

and the *posterior** probabilities for **Yes** and **No**:

$$P(\mathbf{Yes}|\mathbf{E}) = \frac{0.0053}{0.0053 + 0.0206} = 0.205 \qquad P(\mathbf{No}|\mathbf{E}) = \frac{0.0206}{0.0053 + 0.0206} = 0.795$$

So, with the evidence at hand (**Sunny**, **Cool**, **High** and **Yes**), we can say that it is nearly four times more probable that the game will not be played than that it will.

*The *a priori* probability of an outcome is the general probability of that outcome, while the posterior, or *a posteriori* probability is the probability of the outcome based on some information, i.e. 'evidence', in addition to the general probability.

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ **Naïve Bayes based on probability density function e.g. Gaussian [M]**	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Gaussian Naïve Bayes

- In the case of numeric variables, probabilities are estimated not by proportions of instances but by distribution (probability density) function values, resulting in the formula:

$$P(C_j|E_i) = \frac{P(C_j)\prod_{a=1}^{n_A} f_{ja}(x_{ai}|C_j)}{P(E_i)}$$

  which is identical to that for categorical attributes, except for the expression $f_{ja}(x_{ai}|C_j)$ - this represents the value at xai of the probability density function for attribute a among instances belonging to class C_j (each class defines a subset of the data set and the distribution of each numeric variable is estimated for these).

- In the particular case of a normally distributed variable x, the probability density is calculated using the following formula:

$$f_{ja} = \frac{1}{\sigma_{ja}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\bar{x}_{ja}}{\sigma_{ja}}\right)^2}$$

  where $\bar{x}_{ja}$ and σ_{ja} are the mean and the population standard deviation for attribute a among instances belonging to class C_j

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ **categorical Naïve Bayes [M]**	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ **Naïve Bayes based on probability density function e.g. Gaussian [M]**	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Combination Naïve Bayes

- Numeric and categorical variables can be combined as predictor attributes, in which case the following formula applies:

$$P(C_j|E_i) = \frac{P(C_j)\prod_{a=1}^{n_{CA}} P(x_{ai}|C_j)\prod_{a=1}^{n_{NA}} f_{ja}(x_{ai}|C_j)}{P(E_i)}$$

where n_{CA} is the number of categorical attributes and n_{NA} is the number of numeric attributes.

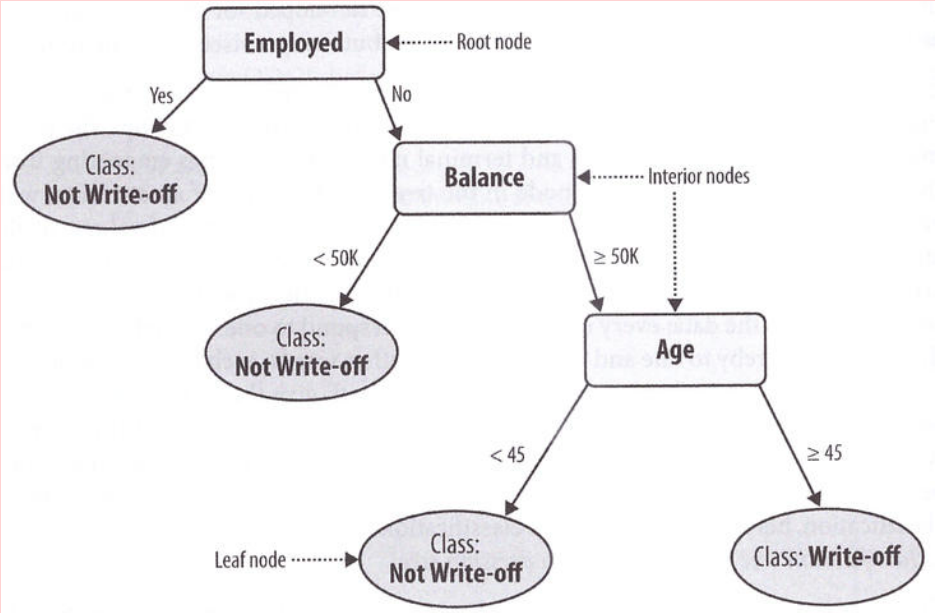
[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ **classification trees: supervised segmentation using impurity measures [M]** ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Classification trees

- A classification tree is built through a **tree learning** process using one of many existing *induction algorithms* (e.g. ID3 or CART).
- The tree learning process is based on the **supervised segmentation** principles discussed in the lecture about relationships between variables. Starting with a tree root node corresponding to the entire set, the data in nodes is segmented and child nodes created to represent the segments.
- Segmentation is performed on the values of the attribute that is **most highly correlated** to the target feature (in the context of the node that is being segmented).
- Tree induction can continue until all attributes are used or leaf subsets are pure, but this is not done in practice because it usually results in *overfitting*.
- Trees are relatively **efficient**, **easy to understand** and **easy to use**.
- In essence, supervised segmentation uses categorical predictors, but **numeric to categorical** conversion boundaries are used as an additional type of **splitting criterion** during segmentation.
- Used in ensemble method *random forest*, where the majority class is chosen from among several different tree predictions.

Examples

The picture below shows a simple classification tree for a target variable that has two values: ***Write-off*** *and* ***Not Write-off****. The attribute* ***Employed*** *is used to split the entire dataset and is hence shown in the root node. As the value of* ***Yes*** *for this attribute is found to indicate the value* ***No Write-off*** *for the target variable, the node representing the subset grouped by value* ***Yes*** *for attribute* ***Employed*** *is a leaf node. Attributes* ***Balance*** *and* ***Age*** *are used to split the dataset further, until the leaf nodes are deemed to be informative enough regarding the value of the target variable (see section on overfitting also).*



Source: [DSB]

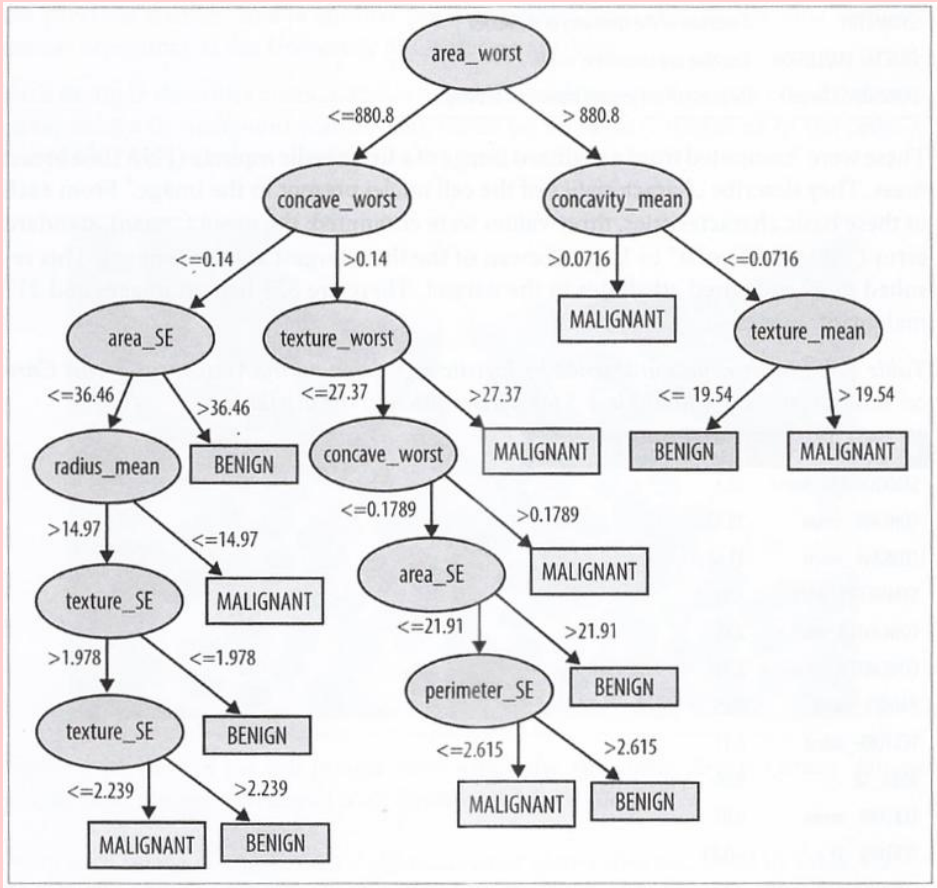


Figure 4-12. Decision tree learned from the Wisconsin Breast Cancer dataset.

Source: [DSB]

The picture above shows a classification tree for a real dataset relating to breast tissue scans examined for cancer. The target variable has values ***MALIGNANT*** *and* ***BENIGN****. The attributes are numeric, categorized into ranges that produce a good predictive model.*

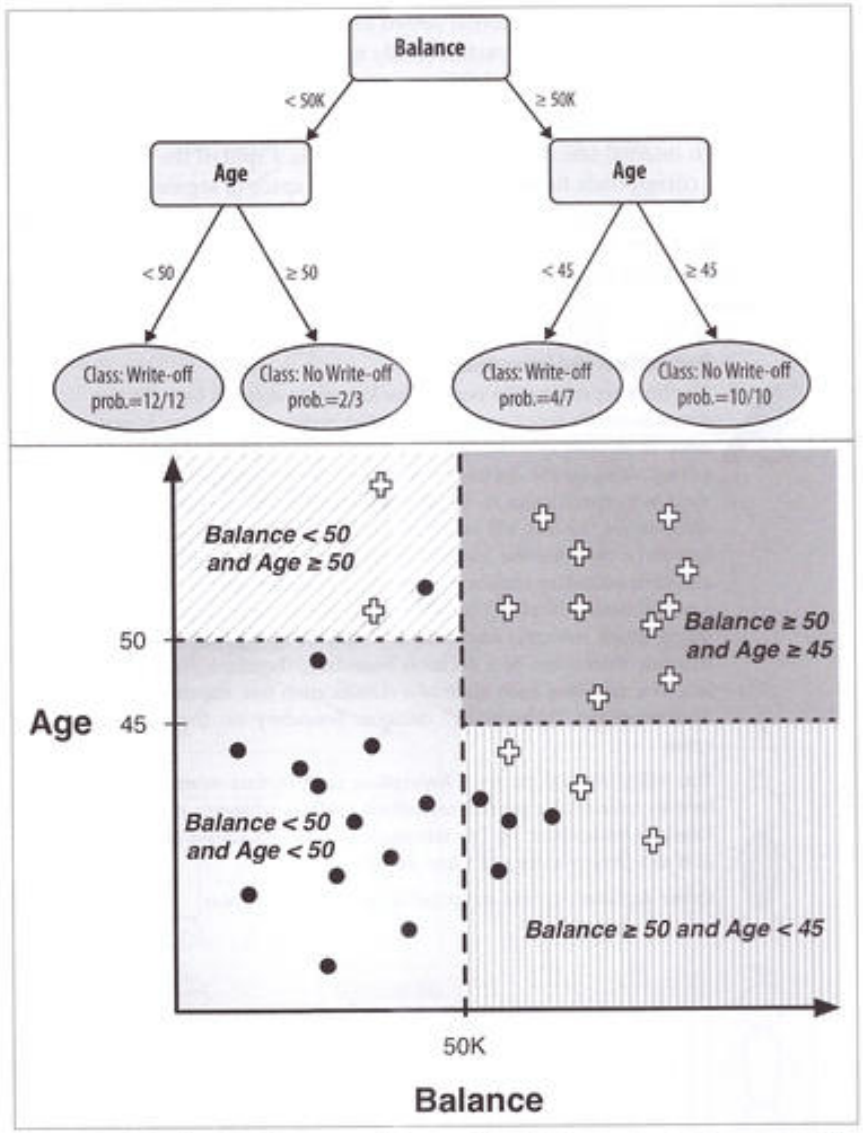


Figure 3-15. A classification tree and the partitions it imposes in instance space. The black dots correspond to instances of the class Write-off, the plus signs correspond to instances of class non-Write-off. The shading shows how the tree leaves correspond to segments of the population in instance space.

Source: [DSB]

Segment visualization

*The picture shows a classification tree using two attributes: **Balance** and **Age**. With two attributes, it is possible to show the data in a two dimensional coordinate system, together with the lines along which the data set is split. The splitting for this particular set could be improved, for example if the age categorization boundary were changed to 55 for Balance < 50K, the two data subsets for Balance < 50K would be pure.*

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	**numeric**
Independent variable[S]/ feature[M] type	**categorical**	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ **regression trees: supervised segmentation using variance measures [M]**
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Regression trees

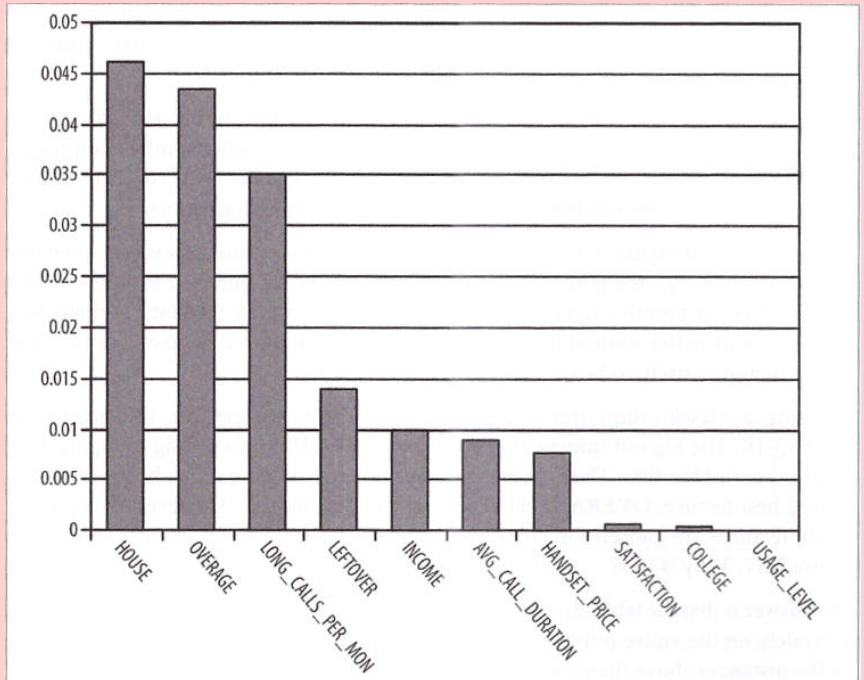
- Regression trees are built for *continuous numeric target variables*
- Predictions made by a regression tree model are in the form of a numeric value
- Instead of measures of impurity, regression tree building algorithms use *variance* to define splitting criteria
- Variance is used in the context of a test such as ANOVA, which evaluates splits based on variance within and between subsets: the most informative split is that with the greatest difference between inter-subset and intra-subset variance

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ **classification trees: supervised segmentation using impurity measures [M]** ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Probability estimation trees

- Probability estimation trees are classification trees where outcomes are presented as probabilities, leaving it to the user to consume this information for themselves. For example, if a leaf node of the tree contains 9 instances belonging to class `positive` and 1 belonging to the class `negative` and an unknown instance ends up in this leaf in the classification process, a probability estimation tree would present the user with the outcome `P(positive)=0.9, P(negative)=0.1` rather than classifying the instance as `positive`.
- There are applications where probability estimation is more appropriate than classification. For example, if a class has an overall low probability, it might be useful to rank segments by these low probabilities (e.g. 0.01, 0.008, 0.005) and act on a number of those with the highest probability. If a company has a budget to spend on customer retention, they may want to rank all the customers by probability of leaving and offer incentives to the number of highest ranking potential leavers that the budget allows.

Examples



Rank	Info. gain	Attribute name
1	0.0461	HOUSE
2	0.0436	OVERAGE
3	0.0350	LONG_CALLS_PER_MON
4	0.0136	LEFTOVER
5	0.0101	INCOME
6	0.0089	AVG_CALL_DURATION
7	0.0076	HANDSET_PRICE
8	0.0003	SATISFACTION
9	0.000	COLLEGE
10	0.000	USAGE_LEVEL

Figure 3-17. Churn attributes from Table 3-2 ranked by information gain.

Source: [DSB]

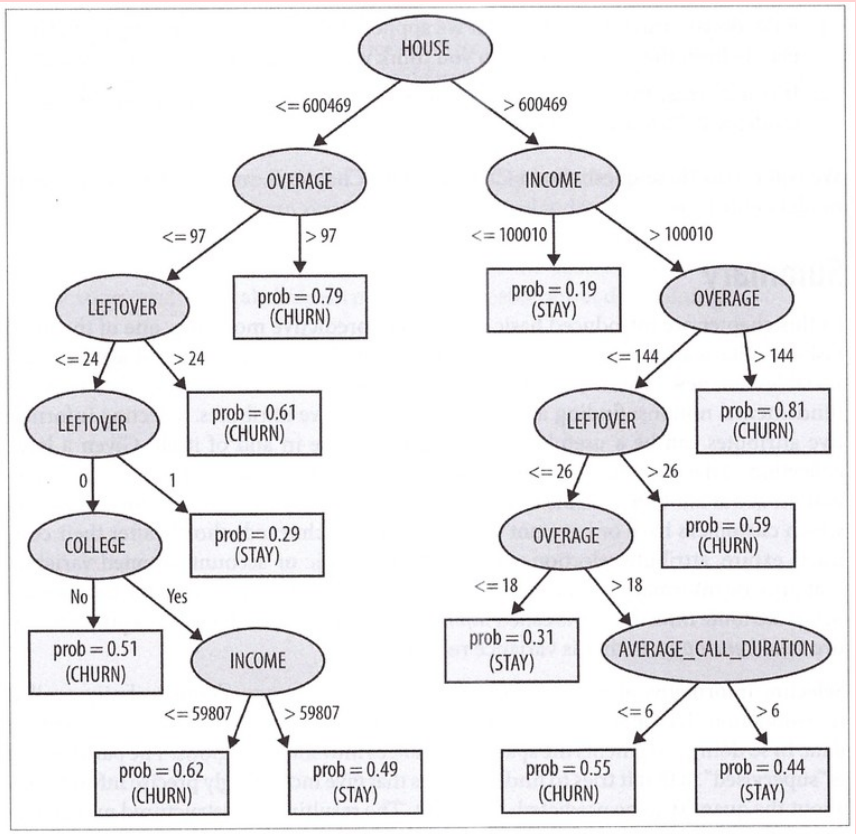


Figure 3-18. Classification tree learned from the cellular phone churn data. Rectangular leaves correspond to segments of the population, defined by the path from the root at the top. Probabilities at the leaves are the estimated probabilities of churning for the corresponding segment; in parentheses are shown the classifications resulting from applying a decision threshold of 0.5 to the probabilities (i.e., are the individuals in the segment more likely to CHURN or to STAY?).

Source: [DSB]

*A probability estimation tree for target variable with values **CHURN** and **STAY**, regarding the likelyhood of the customers of a mobile phone company to leave. The picture on the left lists the attributes that are used in the tree shown in the picture on the right.*

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ **linear classifiers** ✧ **logistic regression [S]** ✧ **support vector machines [M]** ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Linear classifiers

- A linear classifier as a predictive model has numeric inputs and a categorical value as output
- A linear classifier is most easily illustrated in a two-dimensional cartesian coordinate system, where data instances labelled by the value of the categorical target variable are shown at the points defined by their attribute values. The linear classifier is a straight line that splits the entire space into two parts, each containing as homogeneous as possible a data subset with regard to the target variable (see examples below).
- This idea can be extended to any number of dimensions, corresponding to the same number of attributes, with the dividing artefact being an $(n-1)$-dimensional plane, where n is the number of attributes (e.g. when the number of attributes is $n = 2$, the dividing artefact is a 1-dimensional 'plane' i.e. a line).
- The form of the general linear model is:

$$f(x) = w_0 + w_1 x_1 + w_2 x_2 ... + w_n x_n$$

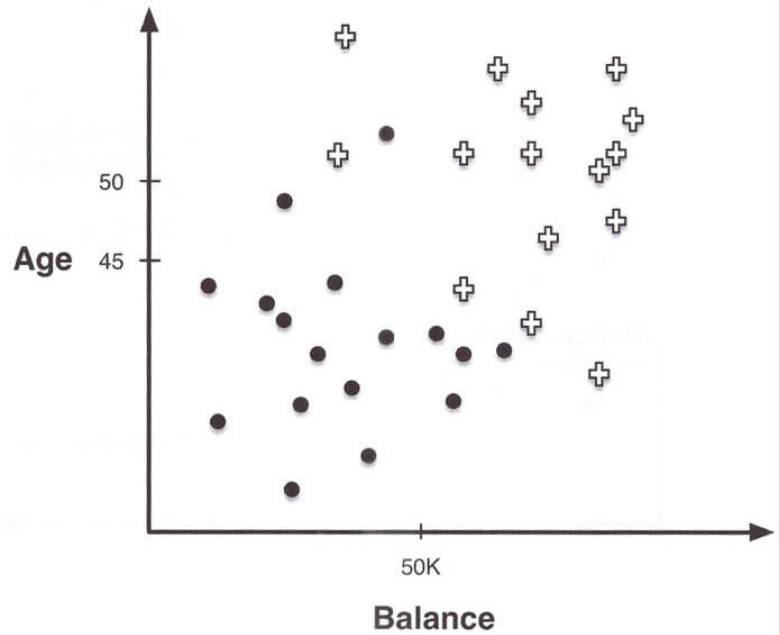
where n is the number of variables, x_i are the attributes and w_i are the linear function parameters. The function $f(x)$ is called the *linear discriminant function*.

- The aim of linear classification is to find the geometrical artefact that best divides the attribute space with respect to the target classes (target variable values). This line, plane or hyperplane (the term used for a plane with more than 2 dimensions) is expressed as

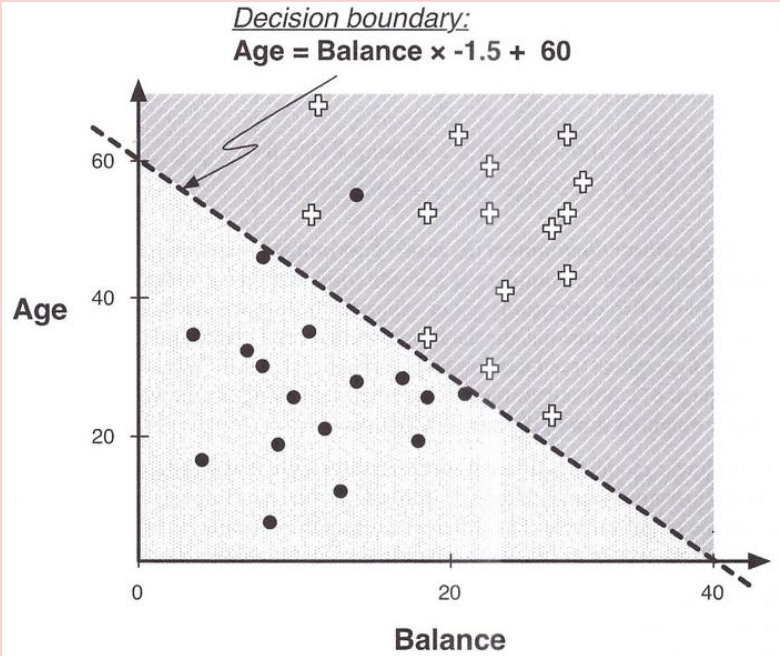
$$f(x) = 0$$

  and its parameters (weights) w_i are chosen by a process of optimization, or learning, with an exact objective that depends on the method used.
- Once the weights are determined i.e. learned, the linear discriminant function can be used to assign new data instances to classes, with $f(x) > 0$ placing data instance x in one class and $f(x) \leq 0$ in the other.
- The weights, w_i, can be interpreted as a measure of importance of the attibutes to which they apply, provided the attribute values were normalised preceding the weight learning (optimization) process.
- The most common linear classifier methods are:
  - Logistic regression
  - Support Vector Machines

Examples



Source: [DSB]



Source: [DSB]

The picture on the left shows the same data set as that shown for segment visualisation, but without the splitting lines. A boundary line is shown alongside its linear formula using attributes ***Balance*** *and* ***Age****.*

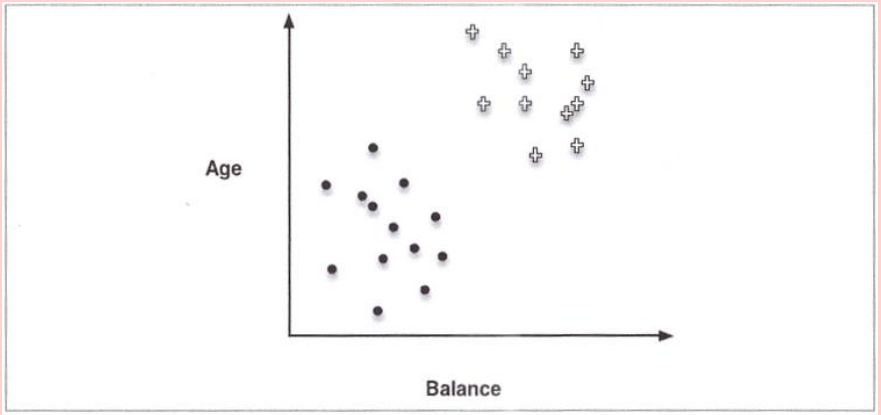


Figure 4-4. A basic instance space in two dimensions containing points of two classes.

Source: [DSB]

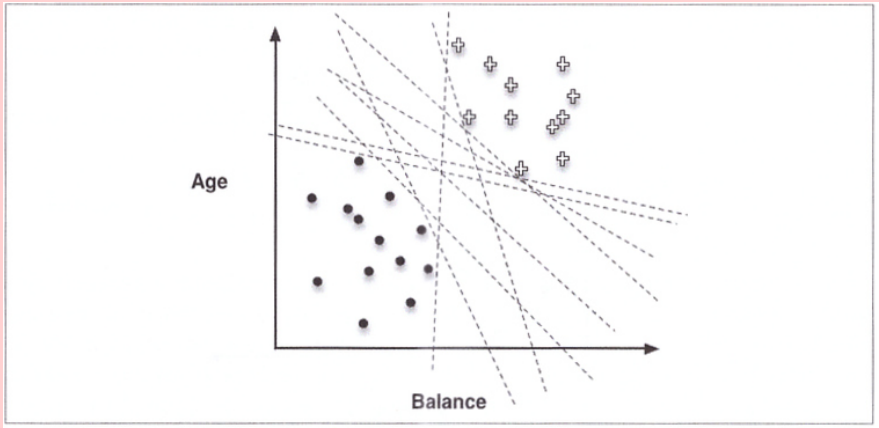


Figure 4-5. Many different possible linear boundaries can separate the two groups of points of Figure 4-4.

Source: [DSB]

The picture on the left shows another dataset in the two-attribute space, while the one on the right illustrates a few of the infinite number of ways in which the linear discriminant could be defined.

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ **logistic regression [S]** ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Logistic regression

- Logistic regression is, in spite of its name, in fact a classification model. Its input attibutes are numerical and the output is either a category or a probability of belonging to a category.

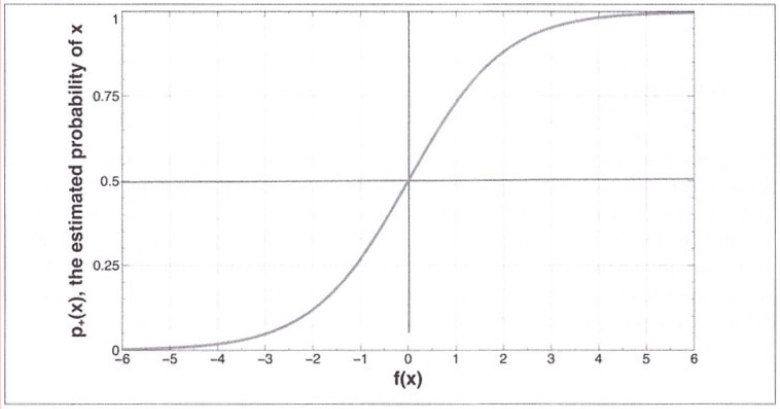


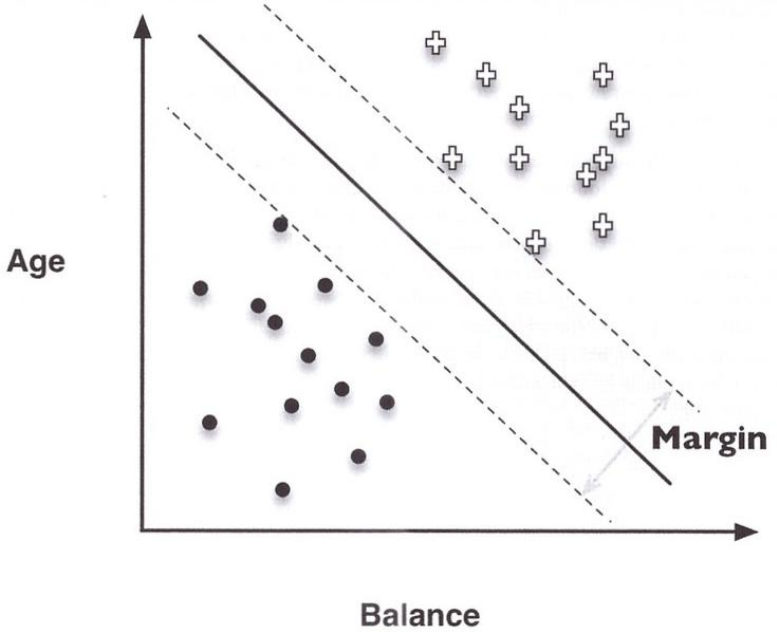
Figure 4-10. Logistic regression's estimate of class probability as a function of f(x), (i.e., the distance from the separating boundary). This curve is called a "sigmoid" curve because of its "S" shape, which squeezes the probabilities into their correct range (between zero and one).

Source: [DSB]

- The model is fitted using the following sigmoid function, which is a transformation of the linear discriminant: $\boxed{p_+(x) = \frac{1}{1 + e^{-f(x)}}}$
- Fitting consists of maximising an aggregation of $p_+(x)$ (or $1 - p_+(x)$ if the instance is classified as not positive) by modifying the weights (w_i) in an iterative process.
- This fitting approach maximises the aggregate distance of the instances from the discriminant plane.
- The value of $p_+(x)$ is in the range [0,1] but is not a probability. However, once the model is fitted, calculating the sigmoid function on unseen data can easily be used for classification (over 0.5 results in classification as '$+$' and below 0.5 as 'not $+$').

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ **support vector machines [M]** ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Support Vector Machines



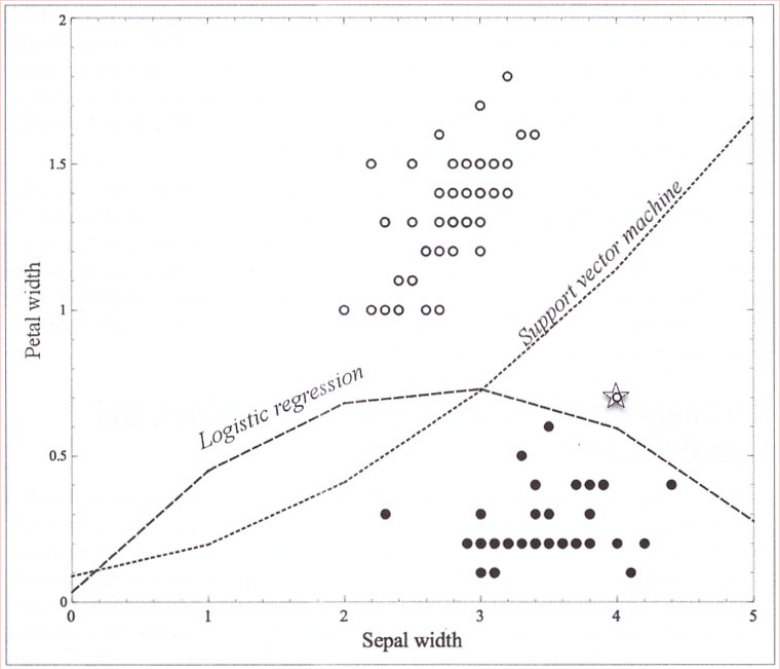
Source: [DSB]

- Support Vector Machines are another type of linear classifier.
- The boundary line (or plane or hyperplane) is determined by finding the widest possible strip of space (delineated by two parallel lines) between instances belonging to one target class and those belonging to the other. The boundary is drawn along the middle of the strip.
- If the data instances cannot be cleanly separated in this manner, a strip is still defined, with the optimization including 'penalties' to account for data instances 'on the wrong side' of the boundary

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ **non-linear classifiers [M]** ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Non-linear classifiers

Classification methods may use non-linear functions (where the attribute values contribute with a power other than 1). In this case the boundary line for classification is not straight.



The picture shows data being classified using non-linear logistic regression and a non-linear support vector machine.

Source: [MSD]

Statistical regression

- Statistical regression models can be linear and non-linear, single variable and multi-variate.
- As a prediction model, regression takes numeric inputs and produces numeric outputs.
- A regression model is *fitted* using known data, then used for the prediction of the dependent variable in data where it is not known.
- We will be looking more closely at linear regression only, but the principles can be extended to the other types of regression.

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ **statistical regression (linear and non-linear, simple and multivariate) [S]**

Linear regression

- A useful linear regression model can be generated if there is a linear relationship between the independent and dependent variable.
- The mathematical model for linear regression is the same linear function as used for classification:

$$f(x) = w_0 + w_1 x_1 + w_2 x_2 ... + w_{n-1} x_{n-1}$$

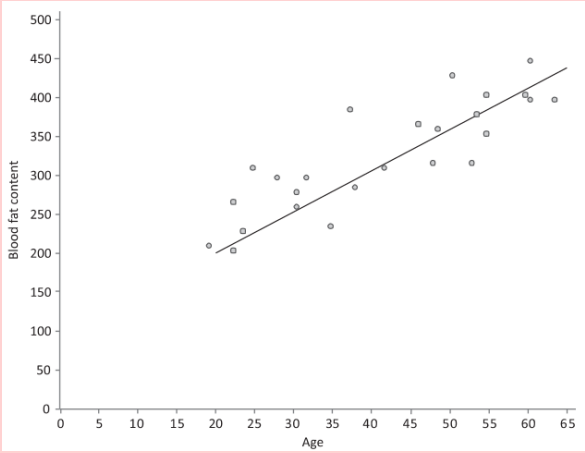
  where x_1, $x_2 ... x_n$ are the independent variables and w_0, w_1, $w_2 ...$ w_n are the coefficients of the linear function.
- With regression, $f(x)$ corresponds to the dependent variable and the coefficients are chosen (i.e. the model fitted) on the principle of minimizing the average distance of the existing data observations from the model line (or plane or hyperplane).
- The model *represents the relationship* between the independent variables on one side and the dependent variable on the other: given a new observation, the dependent variable can be deduced from the independent variables by using the model. In the case of classification the purpose of the line was to *divide the space* and hence define subsets of the data.

Fitting a linear regression model

- A linear regression model with one independent variable can be fitted using a couple of formulae.
- The regression line formula is:

$$y = a + bx$$

where y is the dependent variable, x is the independent variable and a and b are the coefficients of the linear function.



The picture shows an example of some data with one independent variable together with the fitted regression line.

Source: [MSD]

- The coefficient b, which is in face the slope of the line, is calculated as follows:

$$b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

  where x_i and y_i are the independent and dependent variable values for the ith observation, n is the number of observations and $\bar{x}$ and $\bar{y}$ are the means of the independent and dependent variable, respectively.

- The coefficient a can be calculated by substituting the variables in the equasion by the means:

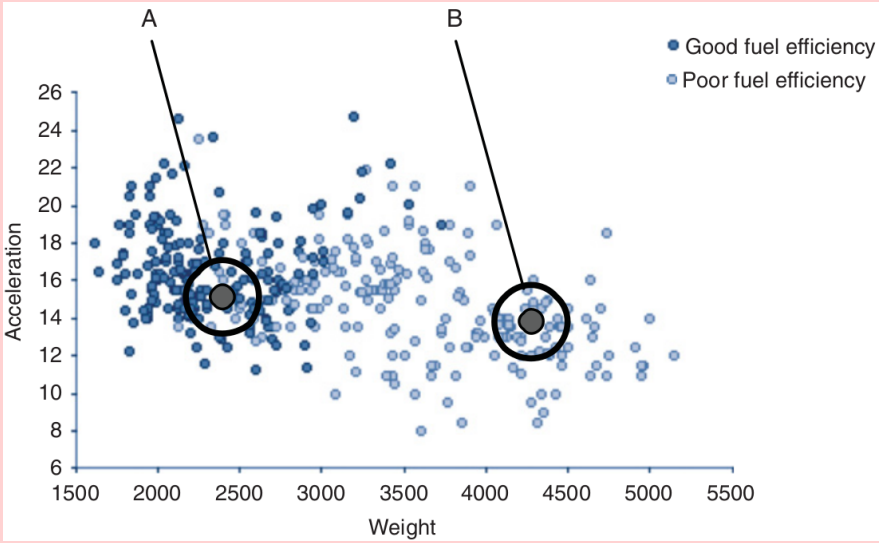
$$a = \bar{y} - b \times \bar{x}$$

- Fitting a model with more than one independent variable is more complicated and is generally done using statistics packages or programming language methods.

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

k-Nearest Neighbours (k-NN)

- This method can be used with **any types of attributes and target variables**.
- It is an **algorithm** for determining a target variable's value for a data instance, given a labelled data set (one in which the target variable's value is known for all instances).



Source: [MSD]

*In the picture **fuel efficiency** (possible values: **Good** and **Bad**) is determined for two cars, based on existing fuel efficiency data as related to weight and acceleration, using k-NN.*

- The distance is calculated between the new data instance and the instances in the dataset, based on the values of the attributes, which can be numeric (distance is Euclidean) or categorical (some measure of distance e.g. Jaccard).
- The target variable of the new instance is calculated based on the target variable values of the closest k instances (e.g. as mean if numeric or mode if categorical).
- Determining the best value for k is also part of the algorithm.

[S] statistics [M] machine learning		Type of dependent variable[S] / target feature[M]	
		categorical	numeric
Independent variable[S]/ feature[M] type	categorical	✦ classification trees: supervised segmentation using impurity measures [M] ✦ categorical Naïve Bayes [M]	✦ regression trees: supervised segmentation using variance measures [M]
	numeric	✦ linear classifiers ✧ logistic regression [S] ✧ support vector machines [M] ✦ non-linear classifiers [M] ✦ Naïve Bayes based on probability density function e.g. Gaussian [M]	✦ statistical regression (linear and non-linear, simple and multivariate) [S]

Artificial neural networks

- Artificial neural networks (ANNs) are computational models that loosely emulate the workings of a brain.
- ANNs can be designed for any kind of input and output variable types.
- They are non-linear models, with parameters (weights) that need to be fitted to data.
- The structure of an ANN consists of nodes and connections between those nodes, used to model data:
  - a node performs a parameter-controlled operation, to the value that enters it, despatching a new value further into the network
  - the connections (forward or feedback) define the flow of information between nodes
  - the network is trained on data with known outputs and in the process it adjusts its parameters to best reflect the relationship between the inputs and the output
  - once trained the network based on training data, then predicts for previously unseen input

- ANN types:
  - multi-layer perceptrons (MLPs) - general type suitable for classification and regression with structured data
  - convolutional neural networks (CNNs) - specialised for image categorisation
  - recurrent neural networks (RNNs) - for working with sequences; particularly successfull are Long Short-Term Memory (LSTM) models

Overfitting

- *Overfitting* is the application of model-building procedures to an extent that causes the created model to include properties specific to the training data, in addition to those actually pertinent to a general model.
- For example:
  - a classification tree can be refined until all the leaf nodes are pure, but in most cases such a tree is over-fitted
  - with a mathematical function overfitting may occur through the addition of too many attributes: more attributes means more dimensions and a better fit, but to the detriment of general applicability
- Overfitting can be avoided by using *holdout* data to test that the accurracy of a model when used on new data matches, or is close to, the accuracy of the model when used on training data.

Examples

The optimal complexity of a particular type of model built with some data set can be determined with the help of a ***fitting graph****. A fitting graph plots the accuracy of models of increasing complexity (e.g. represented by the number of nodes if a model is a decision tree), both for the training set and for a test set. As the complexity increases, so does the accuracy for both the training and test (holdout) set, until a critical complexity value at which the accuracy for the two sets diverges. It is for complexity higher than this critical value that the model is overfitted.*

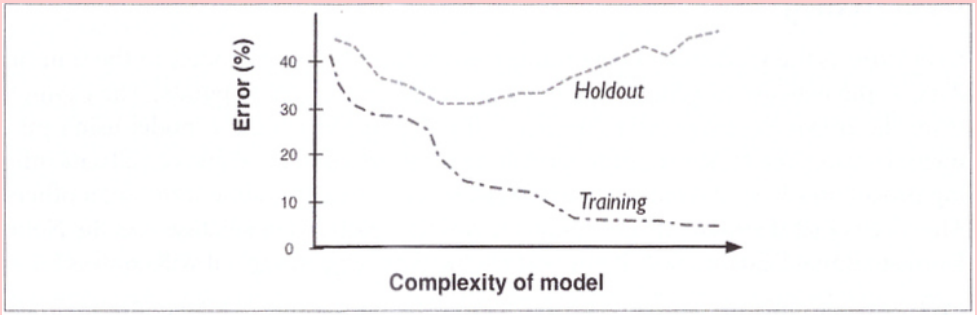


Figure 5-1. A typical fitting graph. Each point on a curve represents an accuracy estimation of a model with a specified complexity (as indicated on the horizontal axis). Accuracy estimates on training data and testing data vary differently based on how complex we allow a model to be. When the model is not allowed to be complex enough, it is not very accurate. As the models get too complex, they look very accurate on the training data, but in fact are overfitting—the training accuracy diverges from the holdout (generalization) accuracy.

Source: [DSB]

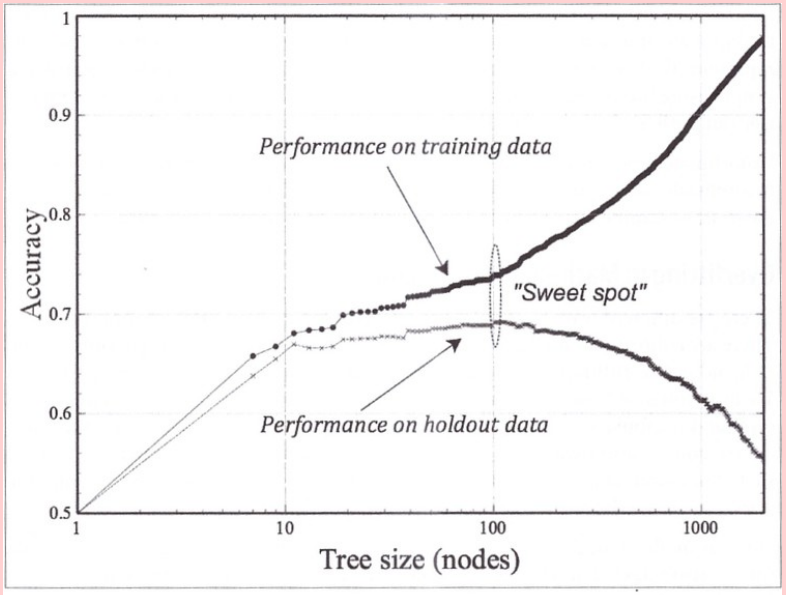


Figure 5-3. A typical fitting graph for tree induction.

Source: [DSB]

References

The pictures in this presentation were taken from the following books. The source for each picture is cited beside it.

[DSB] *Data Science for Business: What you need to know about data mining and data-analytic thinking*, by Foster Provost and Tom Fawcett, O'Reilly Media, 2013.

[MSD] *Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining*, by Glenn J. Myatt and Wayne P. Johnson, John Wiley & Sons, 2014.