SELECTED PREDICTIVE ANALYTICS TECHNIQUES

Lecture

Topics:

- Linear regression
- Logistic regression
- K-nearest neighbors
- Classification tree
- Regression tree
- Random forest

Time: 2 hours

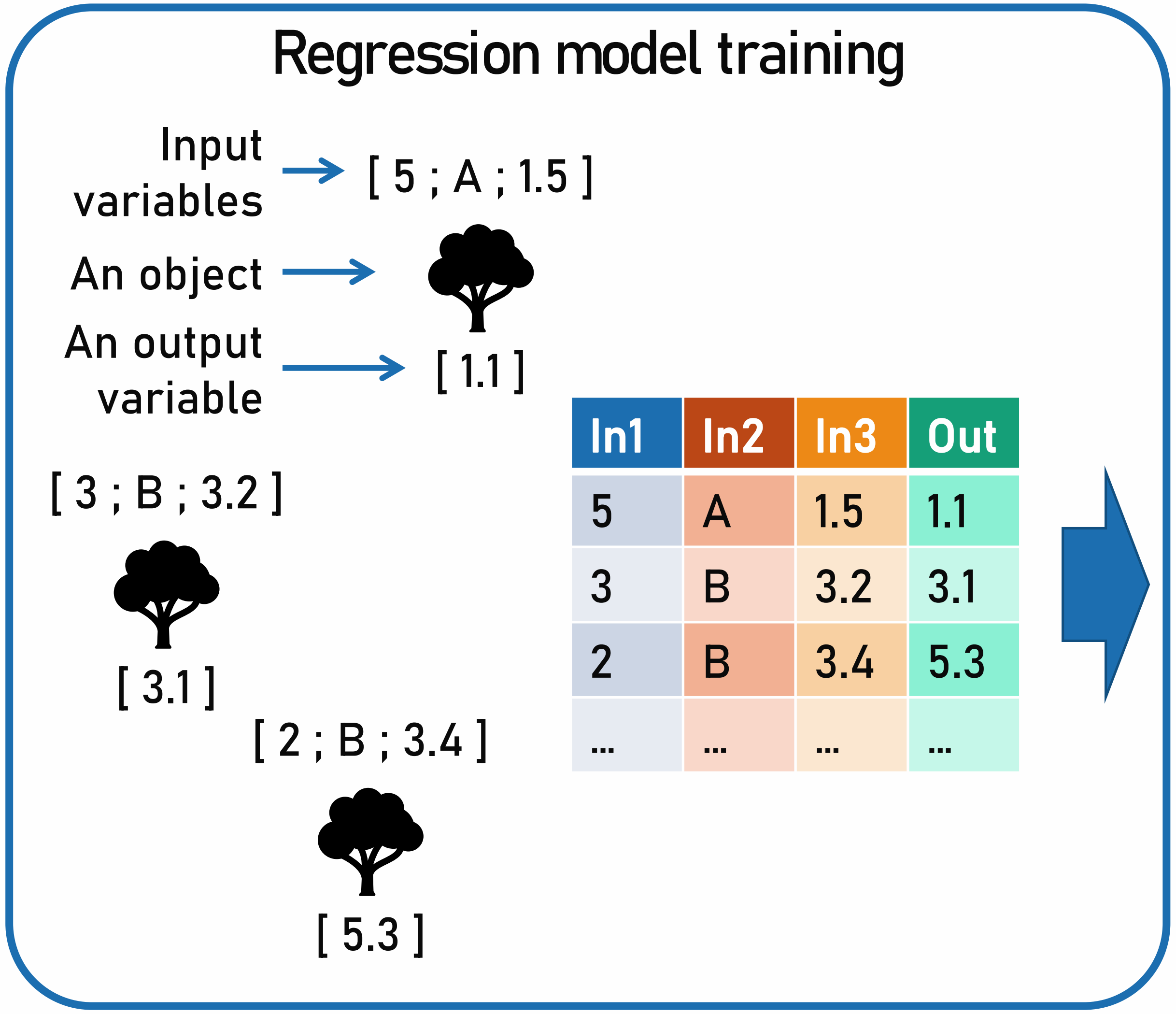
Co-funded by the European Union

TET – The Evolving Textbook
Project no: 2022-1-SI01-KA220-HED-000088975

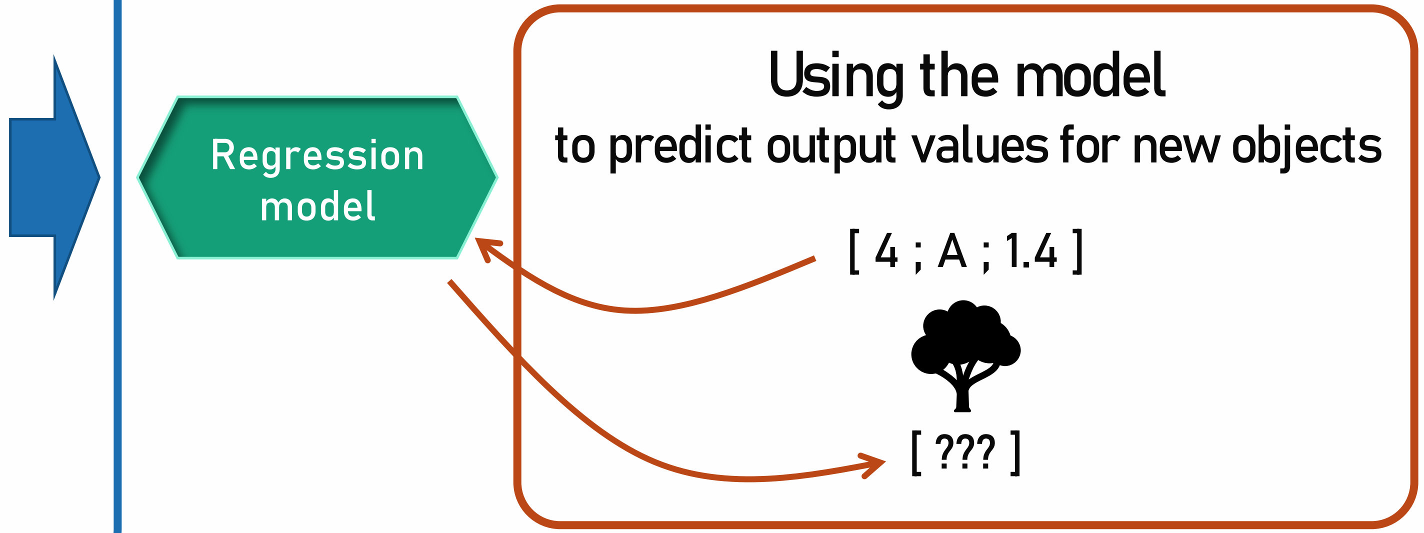
Łukasz Paśko, Rzeszów University of Technology

Regression

Characteristics



In1	In2	In3	Out
5	A	1.5	1.1
3	B	3.2	3.1
2	B	3.4	5.3
...	...	...	...



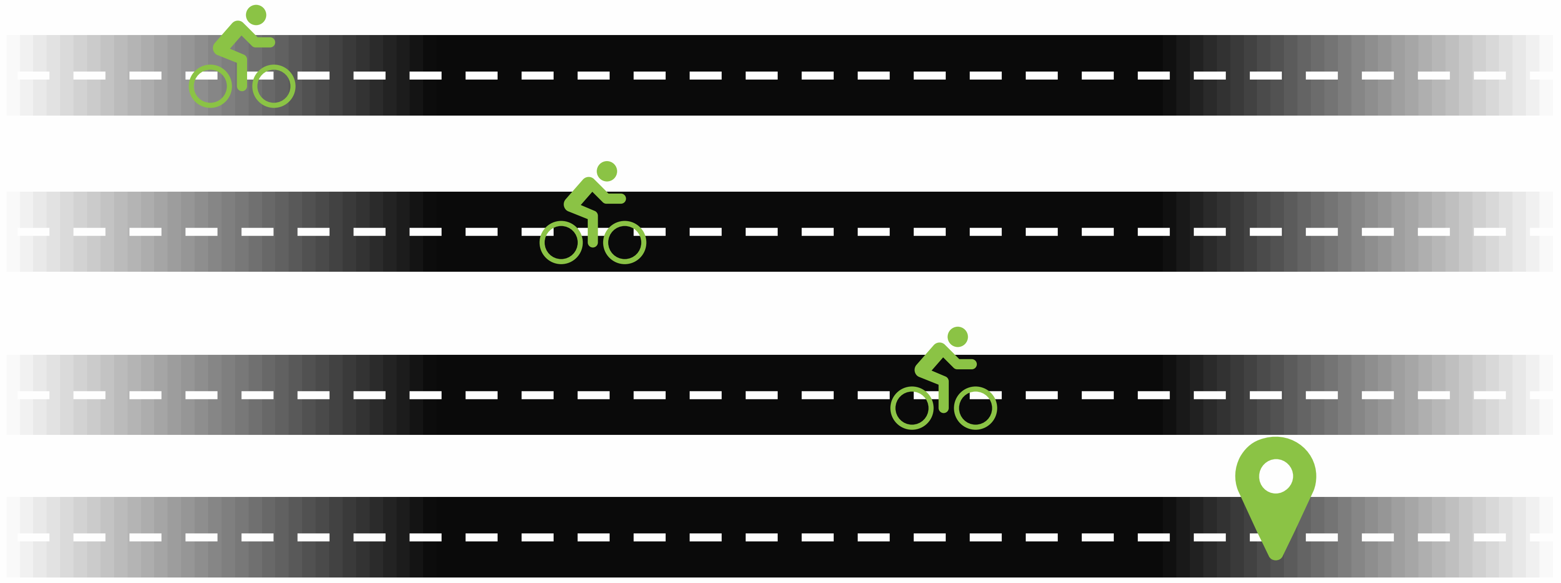
- One of the machine learning methods (supervised learning);
- It requires data on certain objects (e.g. physical, abstract objects, events, states);
- The output value of each object is a number.

Simple linear regression

An example

$$y = a_0 + a_1 x$$

How far will a cyclist travel during his training?



Feature (input variable) Number of energy bars	Result (output variable) Distance
1	10 km
2	20 km
3	30 km
4	?? km

Simple linear regression

An example

Feature (input variable)	Result (output variable)
Number of energy bars	Distance
1	10 km
2	20 km
3	30 km
4	40 km

$$y = a_0 + a_1 x + \varepsilon$$

a_1 – regression coefficient for the input variable x
a_0 – intercept (often denoted as b)

Why in real conditions the relationship between distance and the number of bars may not be perfectly linear?

Training longer/shorter than usual, unfavorable weather, traffic jams…

This randomness is represented by ε.

Simple linear regression

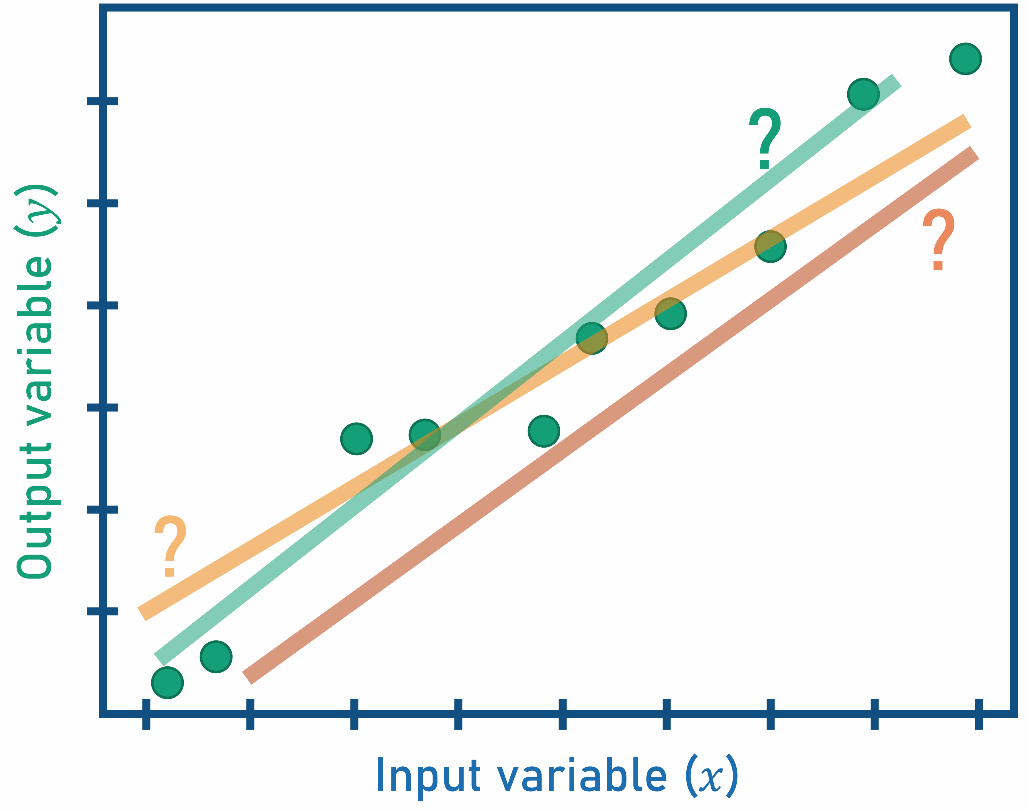
Characteristics

- It allows to express the relationship between two variables;
- It is used to predict continuous values (the output variable is a continuous variable, i.e. it can take on any values within a certain numerical range);
- The values of the output variable are predicted based on the values of one quantitative input variable;
- The relationship between the input and output variable must be approximately linear to enable accurate prediction of the value of the output variable based on the input variable.

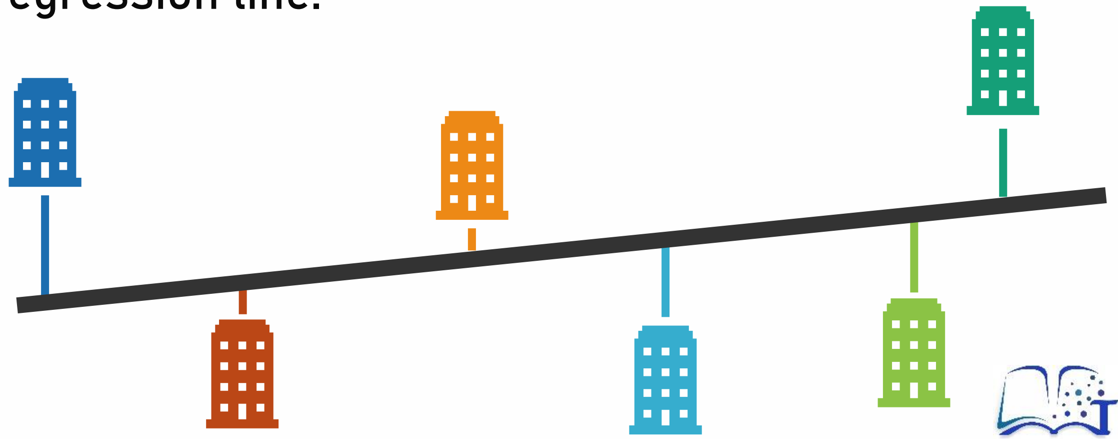
Feature (input variable)	Result (output variable)
Number of energy bars	Distance
1	10 km
2	20 km
3	30 km
4	40 km

Simple linear regression

$$\hat{y} = \overset{?}{\widehat{a_0}} + \overset{?}{\widehat{a_1}}x$$

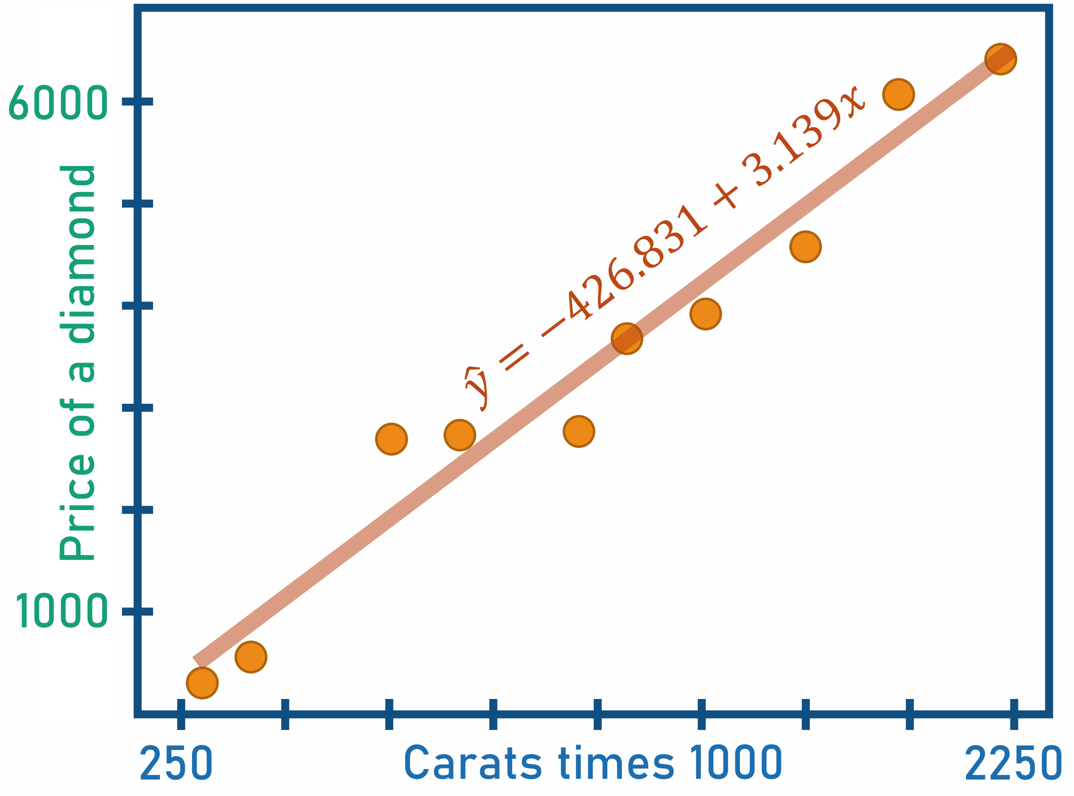


- The goal is to fit a straight line that best reflects the relationship between variables x i y (data fitting).
- The coefficient of the input variable $\widehat{a_1}$ (slope of the regression line) and the intercept $\widehat{a_0}$ (value of y when $x = 0$) must be determined.
- The most popular method: Ordinary Least Squares (OLS).
- OLS minimizes the total distance of all data points from the regression line.



Simple linear regression

An example



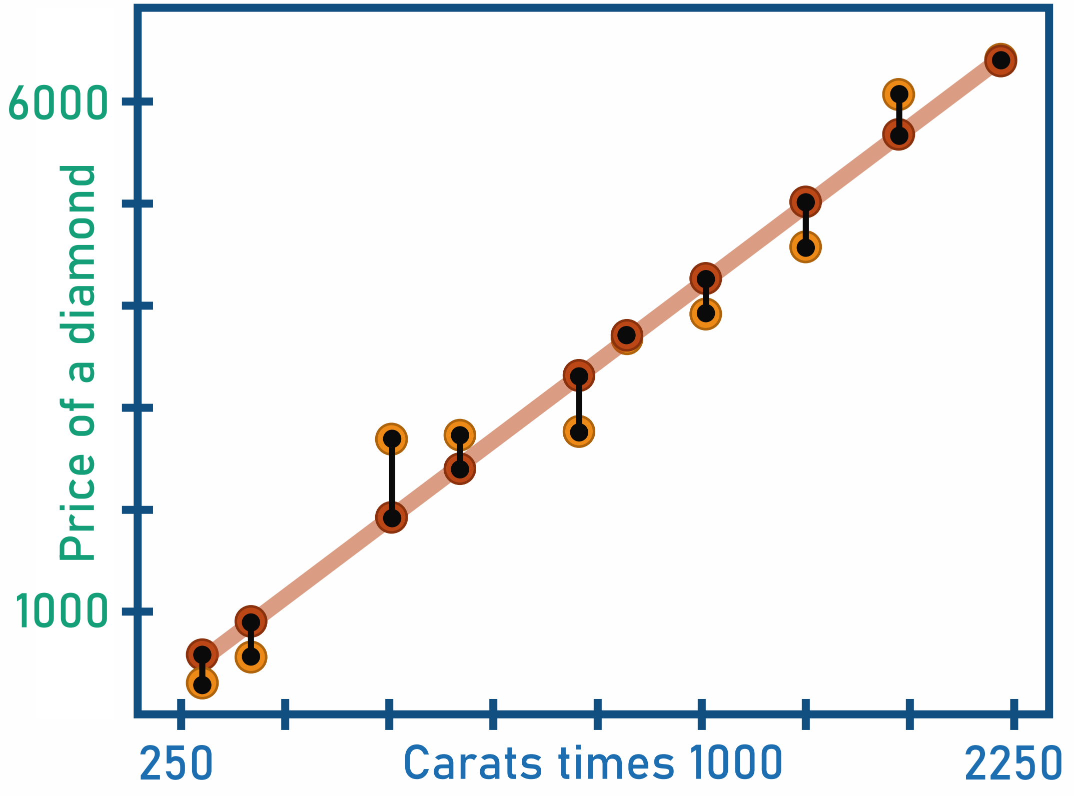
We want to find the relationship between the price of diamonds (y) and the number of carats of the diamond, with carats multiplied by 1000 (x).

The obtained regression line determines the estimated value of the diamond price ($\hat{y}$) for any value of the input variable (x), e.g. for a diamond that weighs 0.3 carats:

$$\hat{y} = -426.831 + 3.139 \cdot 0.3 \cdot 1000 \approx 515$$

Simple linear regression

Determining model fitting errors

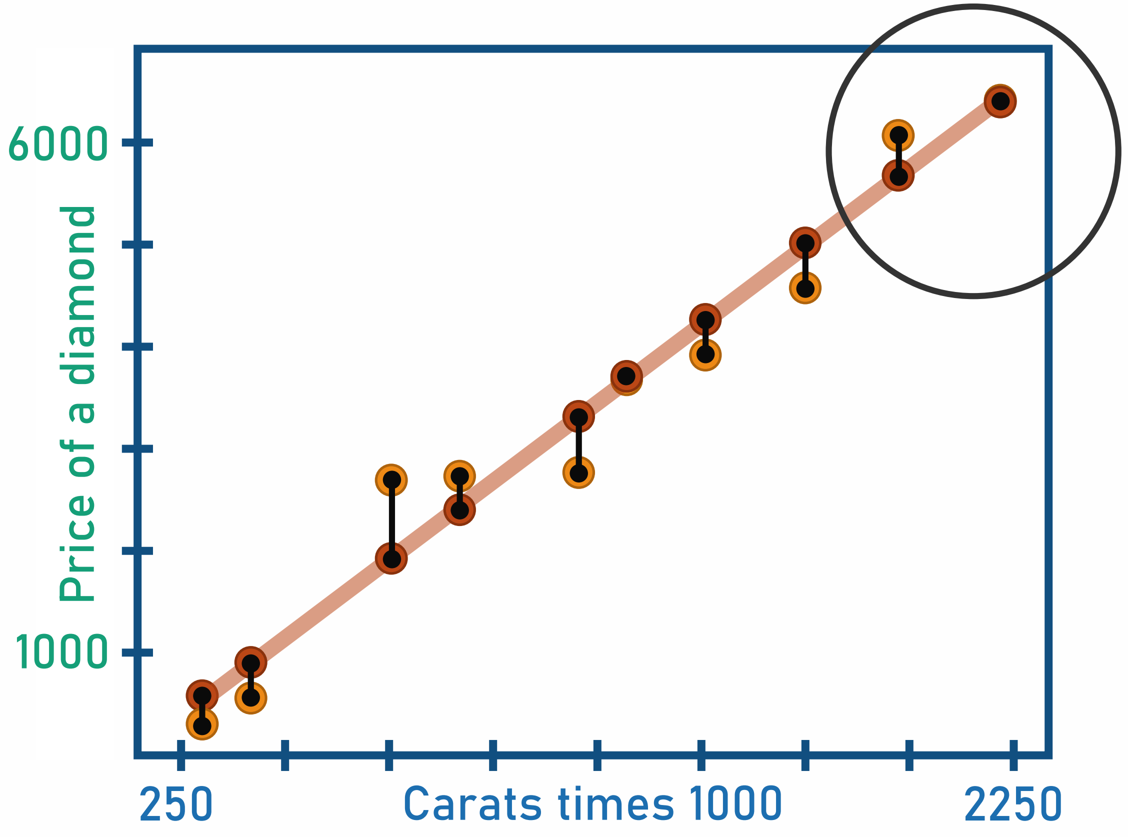


x y $\hat{y} = -426.831 + 3.139x$

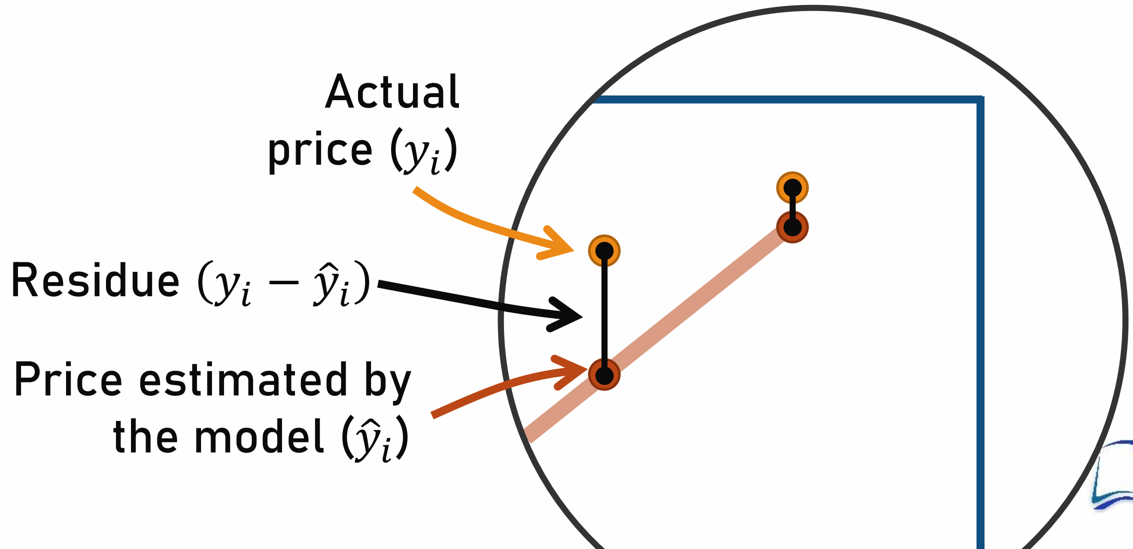
Carats *1000	Price	Estimated price	Residuals $y - \hat{y}$
300	339	514.9	-175.9
410	561	860.2	-299.2
750	2760	1927.4	832.6
910	2763	2429.7	333.3
1200	2809	3340.0	-531.0
1310	3697	3685.3	11.7
1500	4022	4281.7	-259.7
1740	4677	5035.0	-358.0
1960	6147	5725.6	421.4
2210	6535	6510.4	24.6

Simple linear regression

The aim of the regression task



- When determining the course of a simple regression line, we minimize the residual sum of squares (RSS).
- Residuals (black lines on the graph) are the differences between the actual value (y_i) and the value predicted by the model ($\hat{y}_i$).
- RSS acts as a cost function: $RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$



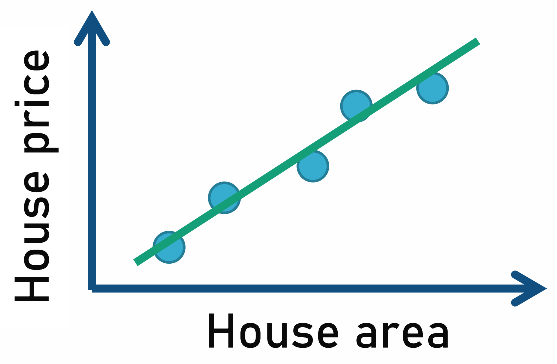
Actual price (y_i)
Residue ($y_i - \hat{y}_i$)
Price estimated by the model ($\hat{y}_i$)

Types of linear regression

Simple linear regression

One input and one output variable; the relationship between variables should be approximately linear

$$\hat{y} = a_0 + a_1 x$$

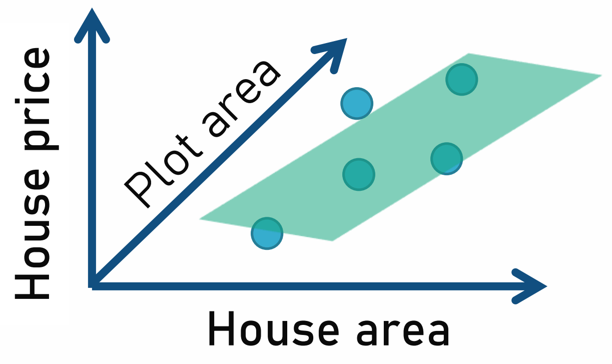


House area (x)	House price (y)
300	499999
200	400000
100	310000
150	367999

Multiple regression

More than one input variable and one output variable; the relationship between input and output variables should be approximately linear

$$\hat{y} = a_0 + a_1 x_1 + a_2 x_2$$



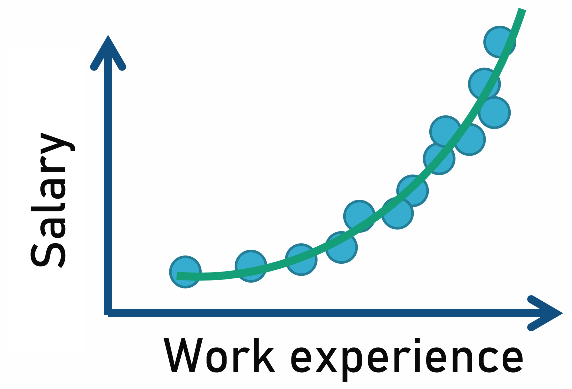
House area (x_1)	Plot area (x_2)	House price (y)
300	45	499999
200	30	400000
100	15.5	310000
150	9.8	367999

Types of linear regression

Polynomial regression

At least one input variable is nonlinearly correlated with the output variable

$$\hat{y} = a_0 + a_1 x + a_2 x^2$$



Work experience (x_1)	Work experience^2 (x_1^2)	Salary (y)
1	1	3000
2	4	3100
4	16	3400
6	36	3900

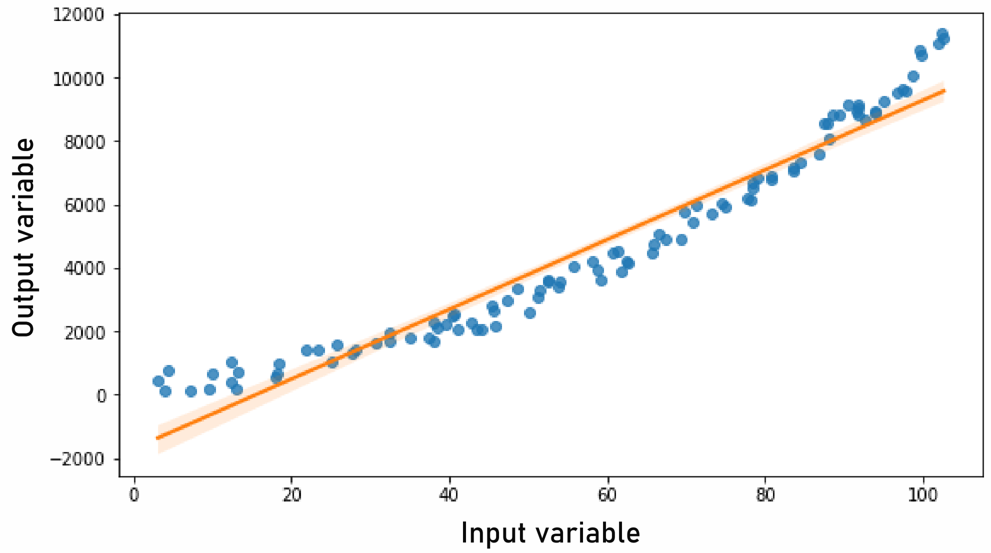
- Polynomial regression is a special case of linear regression with transformations;
- Polynomial regression **is linear regression in the transformed feature space**. However, in the original feature space, the relationship between the input variables and the output y is nonlinear.

Types of linear regression

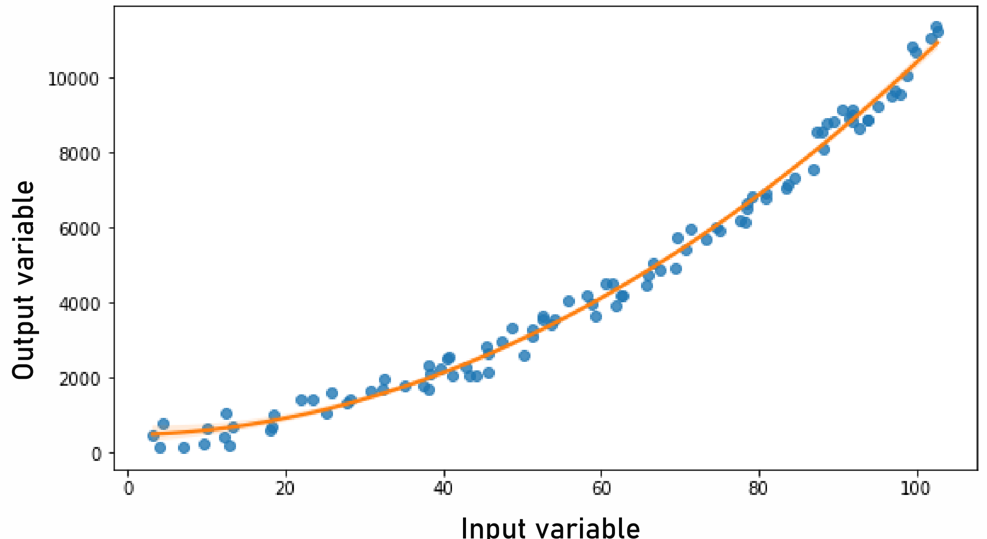
Polynomial regression

- It is used when a given input variable is nonlinearly correlated with the output variable.
- Steps in polynomial regression:
  - We transform the input data, extending it with additional features: x, x^2, x^3, …, x^d;
  - Then we apply classical linear regression on this transformed data; in this way, the linear model maps the nonlinear dependencies between the input and the output.
- The model becomes a linear combination of the input variables and their transformations; the transformations in this case are the result of exponentiating the values of the original input variables.

Fitting a linear model



Fitting a polynomial model



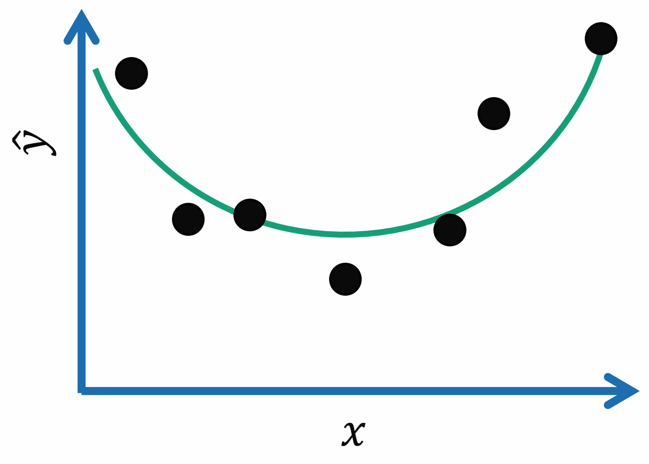
Types of linear regression

Polynomial regression

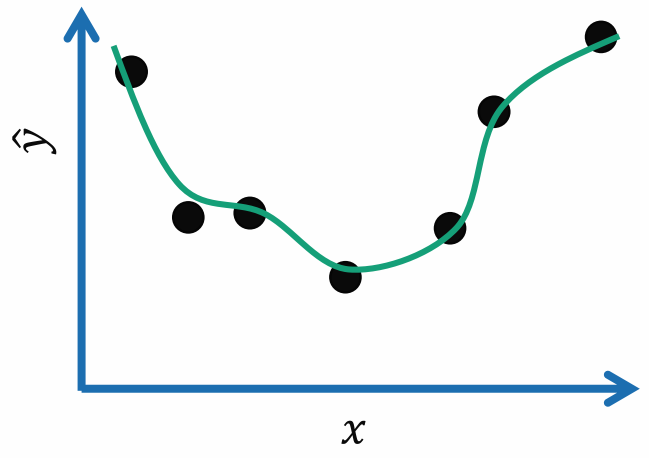
To what extent should a feature be expanded?

- If the data has a simple, nonlinear relationship between input and output (e.g. parabolic), then a low degree (2 or 3) may be sufficient.
- For more complex relationships, higher degrees can be considered, but with caution to avoid overfitting.
- A degree that is too low = underfitting the model.
  A degree that is too high = overfitting the model.
- To choose the right polynomial degree:
  - Use different polynomial degrees starting with low ones;
  - Each time, test the model on a test set or using cross-validation, paying attention to metrics (e.g. mean squared error, R-squared);
  - Choose the lowest degree possible, but that still provides sufficient model accuracy.

$$\hat{y} = a_0 + a_1 x + a_2 x^2$$



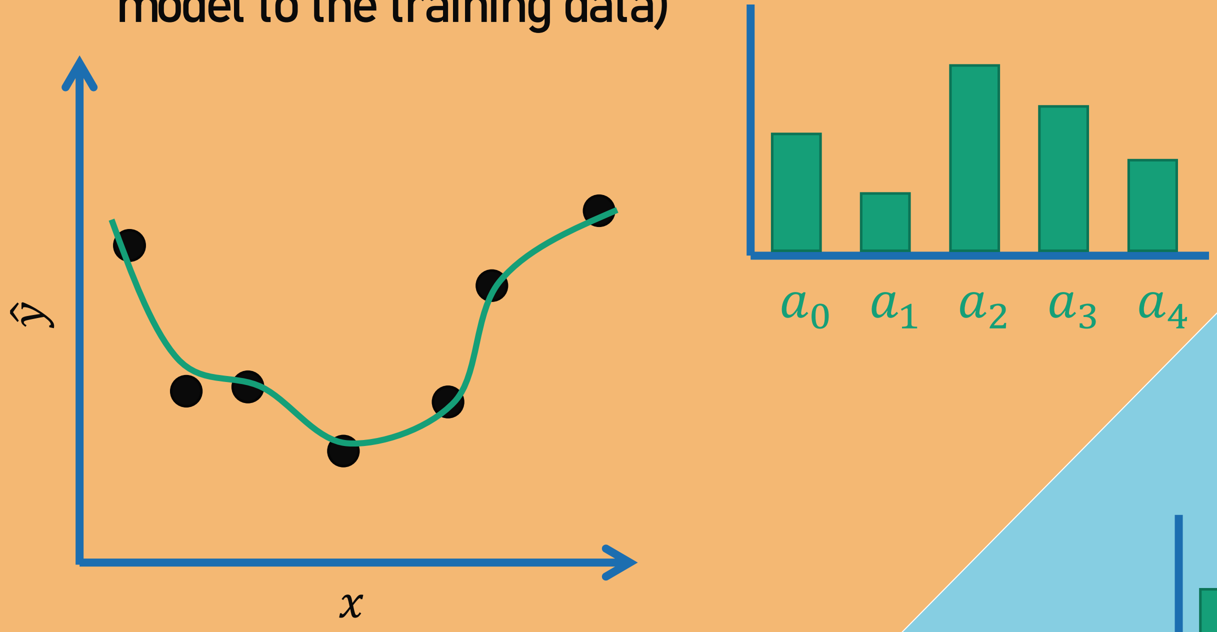
$$\hat{y} = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$$



Regularization in regression

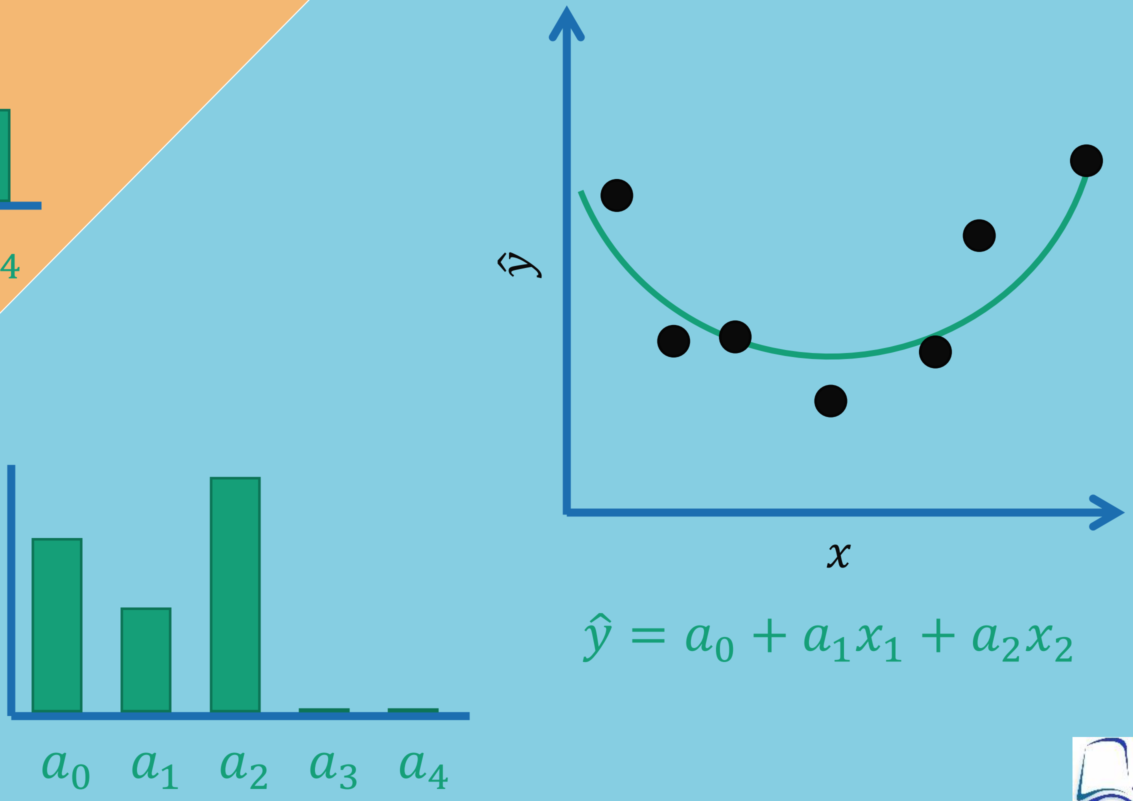
An example of regularization in linear regression

Regression without regularization (overfitting the model to the training data)



$$\hat{y} = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4$$

Regression with regularization (proper model fit, redundant features removed)



$$\hat{y} = a_0 + a_1x_1 + a_2x_2$$

Regularization in regression

Comparison of regularization approaches

Regularization involves adding a component to the loss function that acts as a penalty for fitting the model too closely to the training data (overfitting).

	Ridge regression	Lasso (Least Absolute Shrinkage and Selection Operator)	Elastic Net
Type of regularization	Adds a penalty for the sum of squared coefficients (L_2)	Adds a penalty for the sum of the absolute values of the coefficients (L_1)	Combination of L_1 and L_2
Impact on regression model coefficients	Reduces the values of the coefficients, but does not reset them, so no feature will be eliminated	Can reset coefficients to zero; features with zero coefficients are eliminated from the model (feature selection)	Reduces coefficients and can assign them a value of zero
Usefulness	When all features are somewhat important for predicting the output variable; a good choice to start with	Large number of features; sparse data; suspicion of non-significant features	Controlling the balance between eliminating features and reducing their impact

Regularization in regression

Tips

The λ parameter controls the regularization strength

All types:

- When $\lambda = 0$, we get an ordinary linear regression model;
- The optimal value of λ is selected using cross-validation.

Ridge:

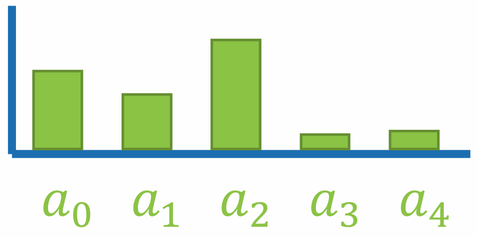
- When λ is very large, the model coefficients approach zero and the regression line runs along the mean of the data set (underfitted model);
- The model coefficients do not reach zero, but they can have values close to zero, which may be enough to avoid overfitting the model.

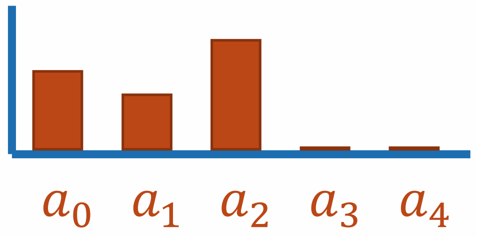
LASSO:

- When λ is very large, the least significant coefficients of the model approach zero;
- Can be used to select the most important features (irrelevant features will receive a coefficient value of zero).

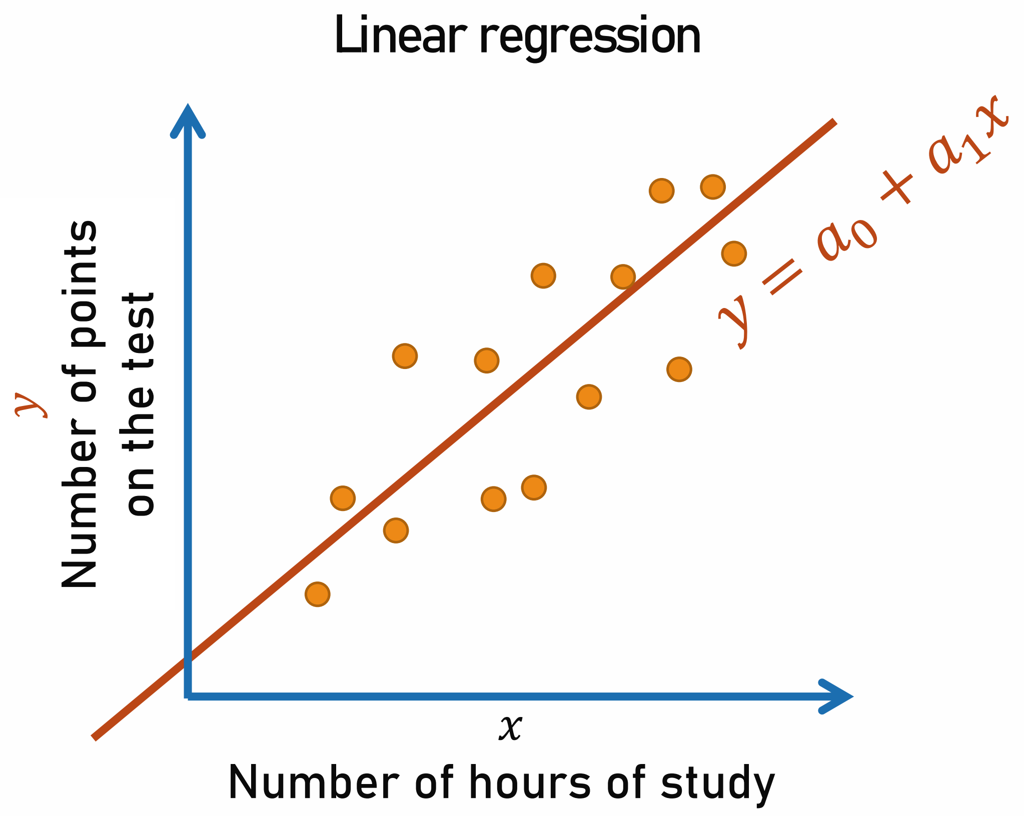
Elastic Net:

- The parameter r is the proportion between the two types of regression:
  - When $r = 0$, the elastic net method is equivalent to ridge regression;
  - When $r = 1$, the elastic net method behaves like the LASSO method.





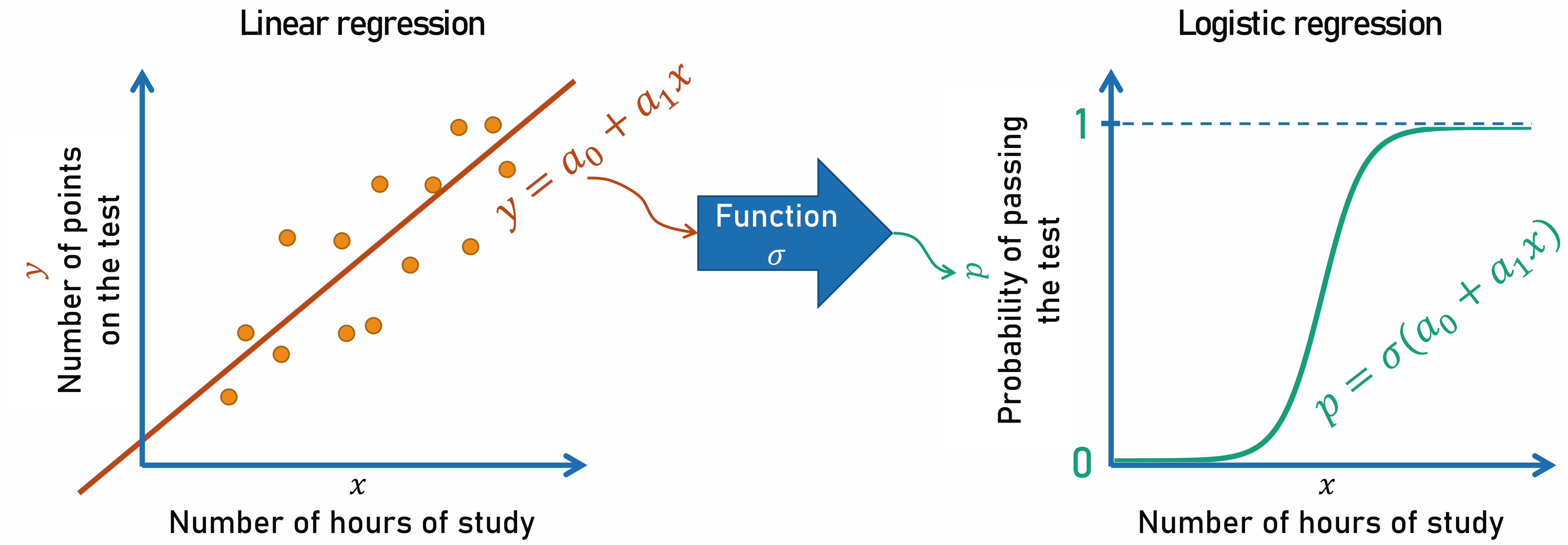
Linear regression vs. Logistic regression



Using linear regression, we can determine the relationship between input variables and the output variable (in the example: the relationship between the number of hours of studying and the number of points scored on a test).

Linear regression vs. Logistic regression

Linear regression will allow you to determine the expected number of points for a given number of hours of study, but it will not estimate the probability of passing or failing a test – we will use logistic regression for that.



Linear regression vs. Logistic regression

General case

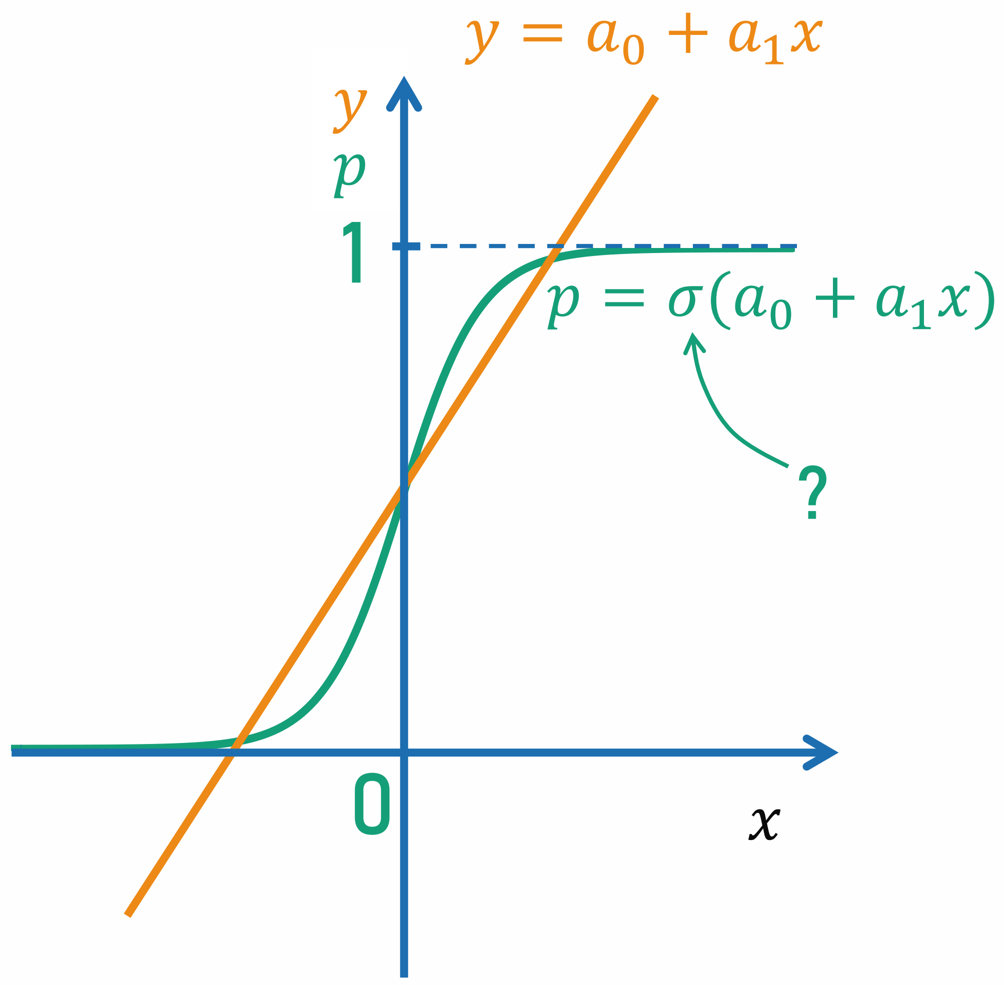
Linear regression model:

$$h(\boldsymbol{x}; \boldsymbol{a}, a_0) := a_0 + a_1 x_1 + a_2 x_2 + \dots + a_m x_m$$

Logistic regression model:

$$h(\boldsymbol{x}; \boldsymbol{a}, a_0) := \sigma(a_0 + a_1 x_1 + a_2 x_2 + \dots + a_m x_m)$$

- In logistic regression, we introduce the function σ;
- The function σ should be selected so that it returns results from the range 0 to 1, which we will interpret as the probability of the object described by features $x_1, x_2, \dots, x_m$ belonging to one of the classes (the so-called positive class).



Logistic regression

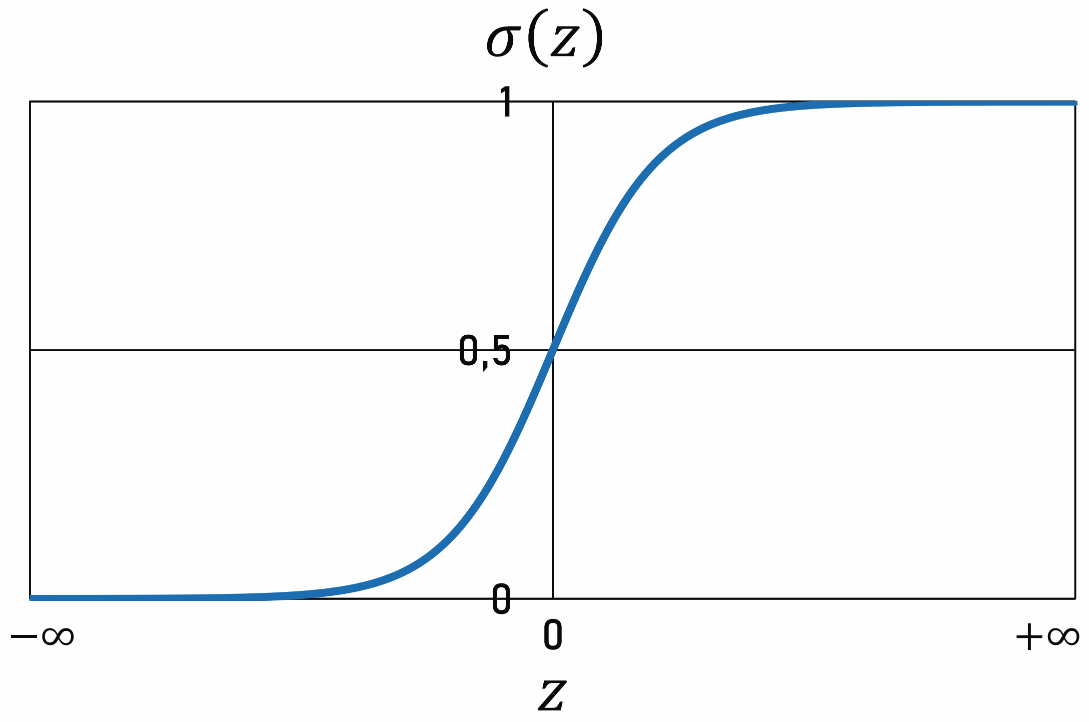
Logistic function

- The function σ is a logistic function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- Essential properties of the logistic function:
  - Horizontal asymptote at height 0,
  - Horizontal asymptote at height 1,
  - $\sigma(0) = 0{,}5$.
- Therefore, the logistic regression model has the form:

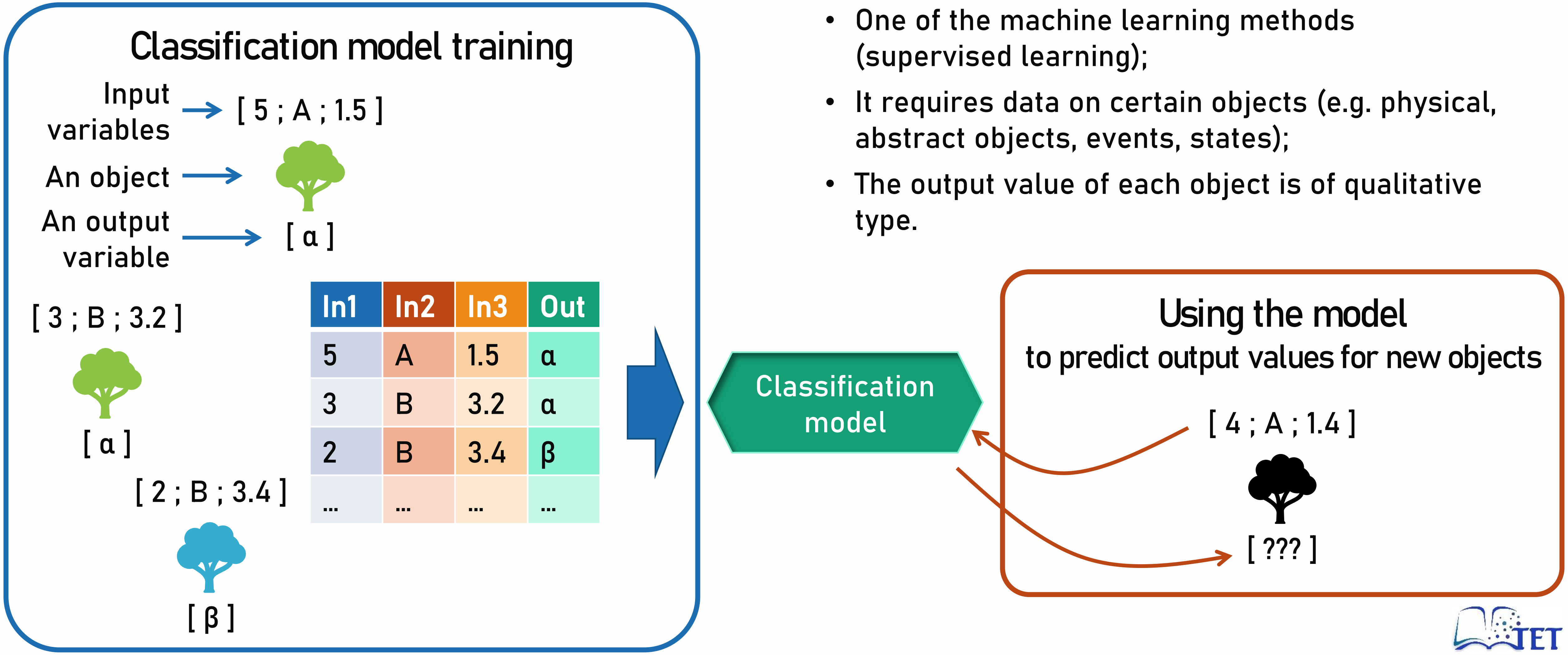
$$h(\boldsymbol{x}; \boldsymbol{a}, a_0) := \frac{1}{1 + e^{-(a_0 + a_1 x_1 + \cdots + a_m x_m)}}$$



Classification

Characteristics

- One of the machine learning methods (supervised learning);
- It requires data on certain objects (e.g. physical, abstract objects, events, states);
- The output value of each object is of qualitative type.

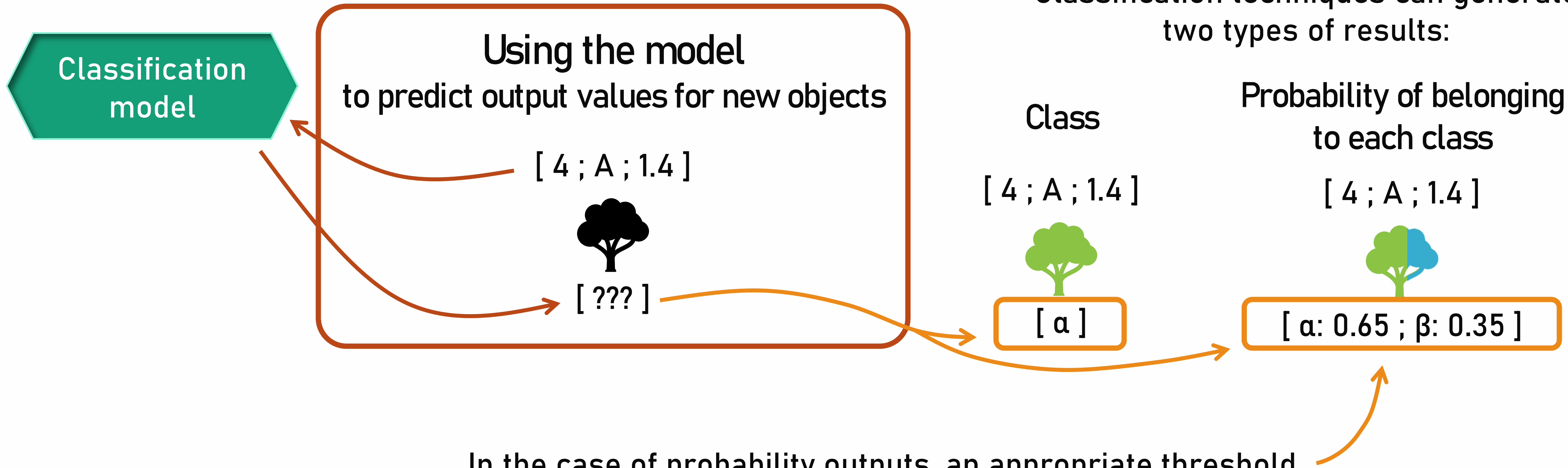


In1	In2	In3	Out
5	A	1.5	α
3	B	3.2	α
2	B	3.4	β
…	…	…	…

Classification

Results we can get

Classification techniques can generate two types of results:



In the case of probability outputs, an appropriate threshold should be established for each class, above which objects will be assigned to a given class.

Positive and negative class

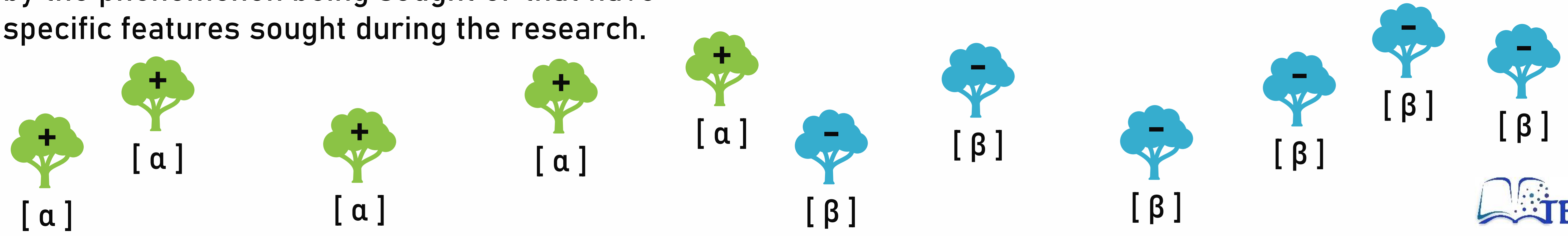
Characteristics

Positive class

- In other words, the distinguished class, marked with the number 1;
- A class that is particularly interesting from the point of view of the prediction task and the research being conducted;
- It expresses the presence of certain specific conditions that we are trying to identify during the study;
- It designates a group of objects that are affected by the phenomenon being sought or that have specific features sought during the research.

Negative class

- In other words, the normal class, marked with the number 0;
- A class in which the phenomenon being sought does not occur, and conditions important from the point of view of the research being conducted do not occur;
- Complementary to the positive class;
- Denotes a group of objects that do not belong to the positive class.



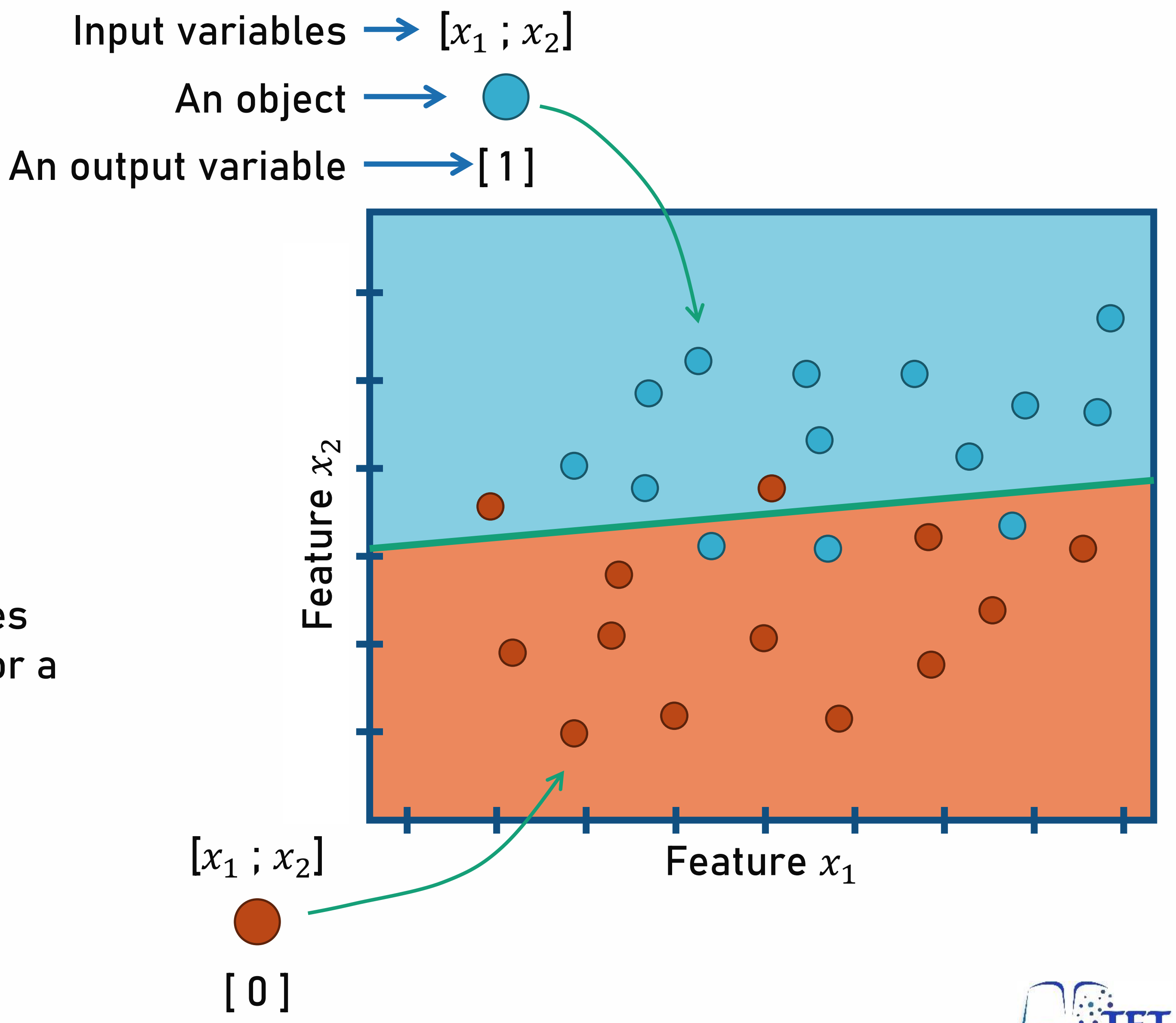
Positive and negative class

Examples	Positive class	Negative class
A model for predicting machine failure	failure	non-failure
A model for recognizing unwanted emails	spam	non-spam
A model for recognizing suspicious transactions in bank accounts	fraud	non-fraud
A model for predicting which customer will default on a loan	default	non-default
A model to predict which customer will switch to a competitor	churn	non-churn
A model for recognizing photos with a cat	cat	not a cat

Logistic regression

Characteristics

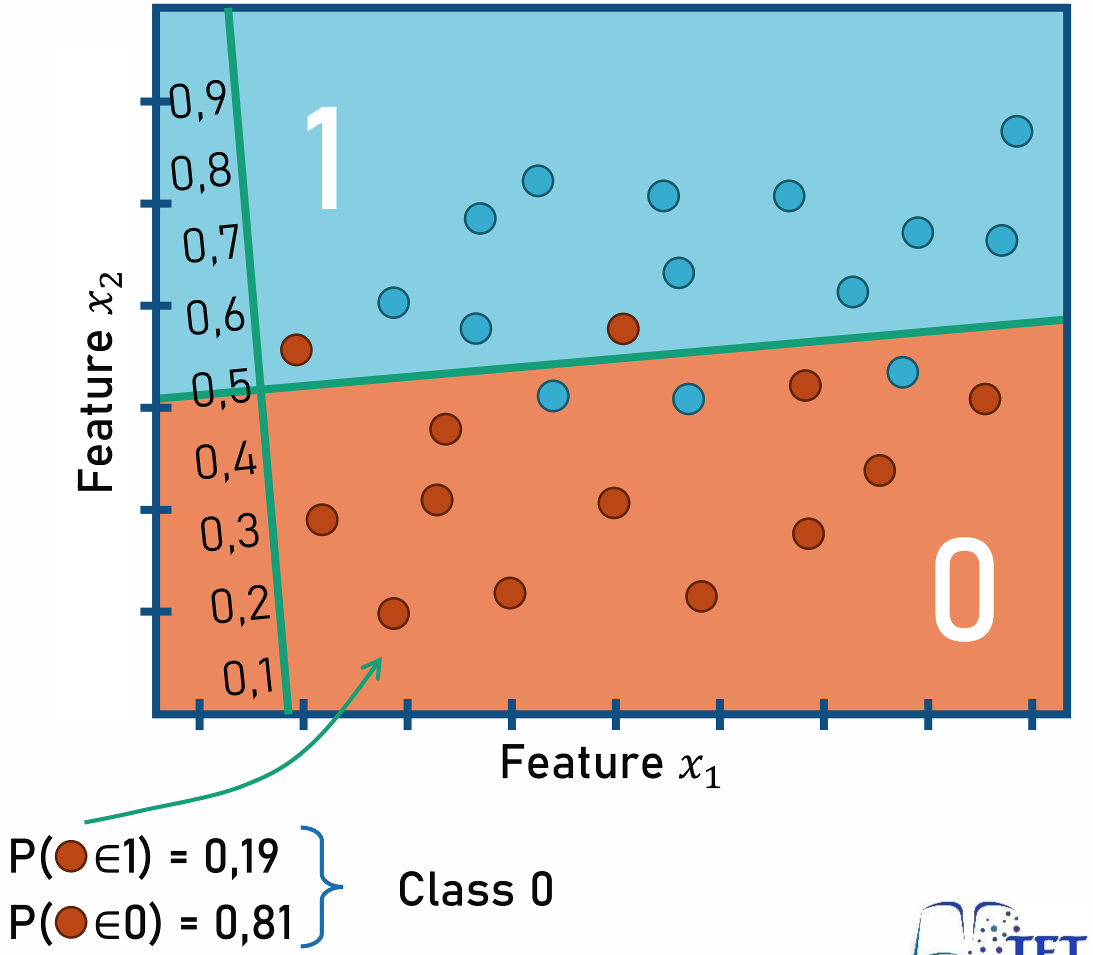
- One of the basic machine learning techniques for classification (despite the fact that the name suggests that the technique is used in the regression problem, i.e. to predict the value of a continuous variable).
- It assigns a probability of belonging to a positive class to each object.
- The boundary of the division of objects into classes is a straight line (in the case of two dimensions) or a hyperplane (in the case of >2 dimensions), so it works well for problems in which the classes are linearly separable.



Logistic regression

Characteristics

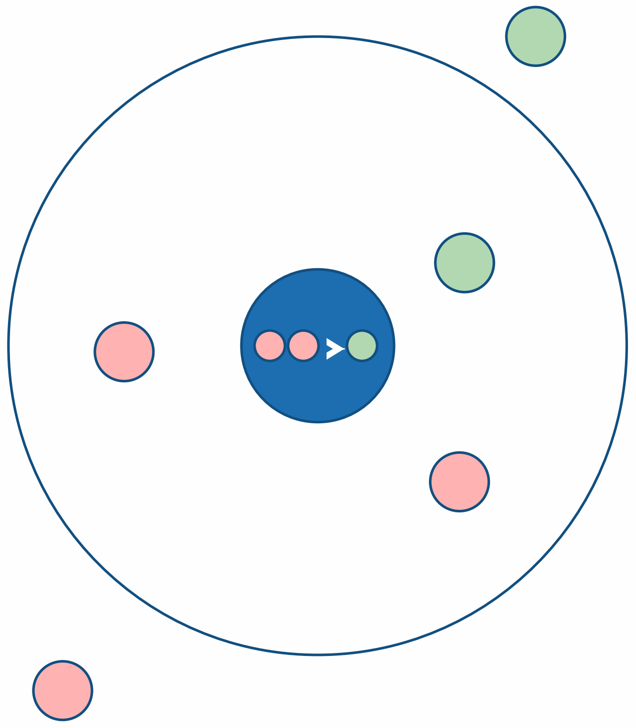
- In the case of two-class problems, the probability of belonging to the negative class = 1 – the probability of belonging to the positive class.
- The probability of belonging to classes can be converted to classes after setting a partition threshold, which is usually 0.5. Objects with a probability higher than the threshold are assigned to the positive class (1), and the remaining objects – to the negative class (0).
- The closer the object is to the class boundary, the greater the uncertainty of the classification result.
- The values of the input variables (features describing the objects) should be of quantitative type. If they have a qualitative type, the type of the input variables should be converted.



K-nearest neighbors

Characteristics

- It is a data-driven technique, which means that it does not create a model that describes the relationship between the input variables and the output variable (it does not create a discriminant function);
- It is a so-called lazy learning technique, because it does not create and train a model;
- The training data set serves as a "memory" based on which new objects are classified;
- It requires that all features of objects (variables) have quantitative values (qualitative features must be transformed).

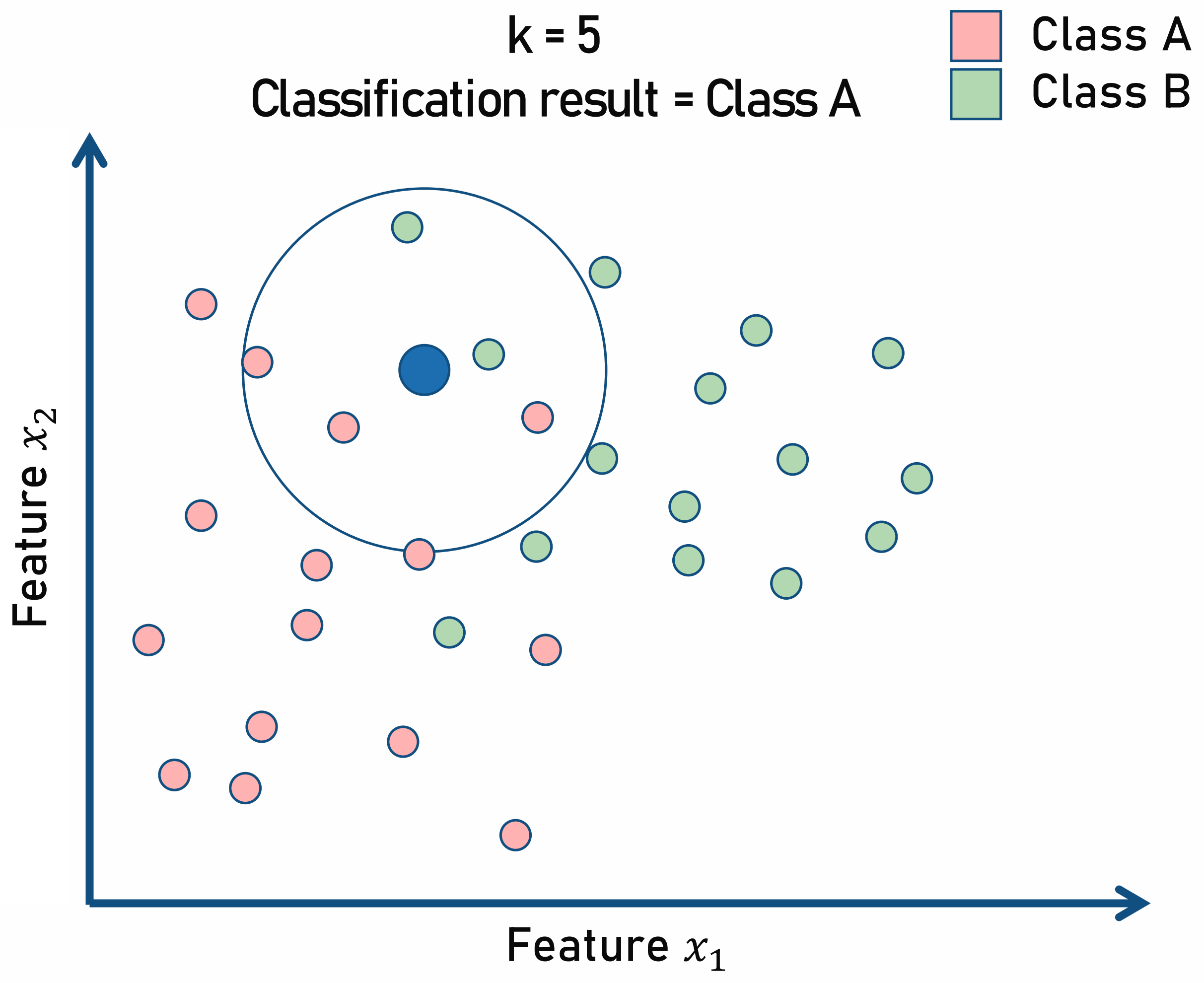


K-nearest neighbors

An example

For an object that we want to classify, we need to:

1. Calculate the distances between this object and every other object in the data set;
2. Find the k objects that are closest to the object being classified;
3. Assign to the object being classified the class that occurs most frequently among the k closest objects.



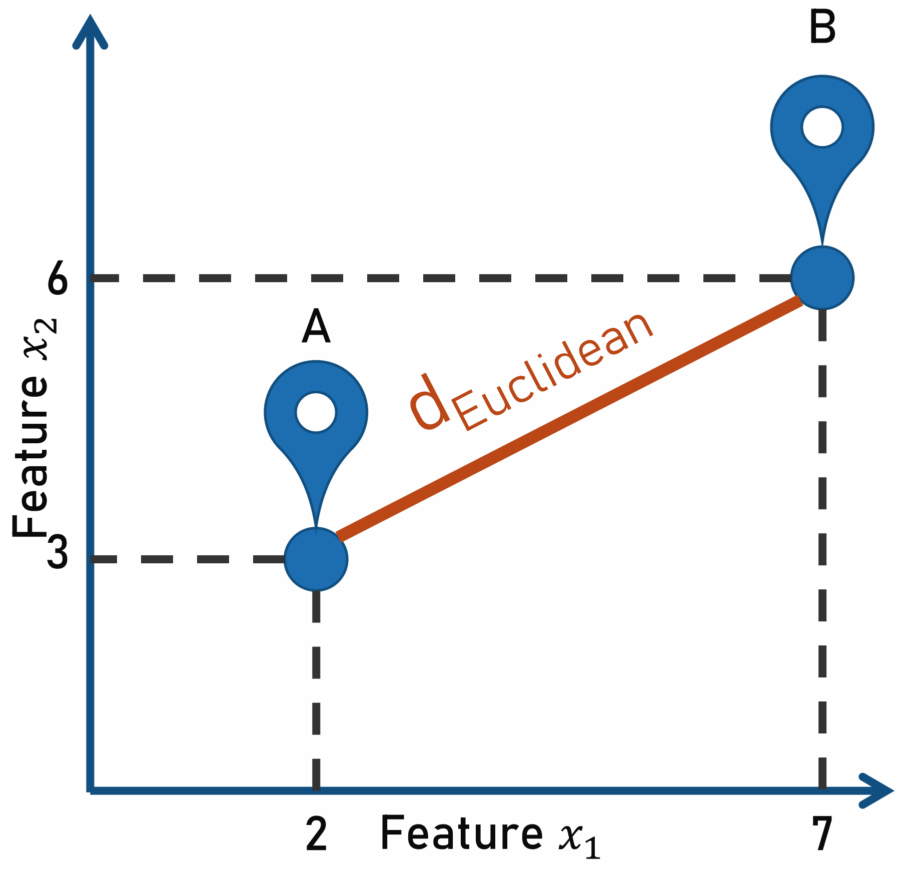
K-nearest neighbors

How to measure the dissimilarity of objects?

- The dissimilarity of objects is expressed by a measure of the distance between objects in the m-dimensional feature space;
- Euclidean distance measure:

$$d(A,B) = \sqrt{\left(x_{1_A} - x_{1_B}\right)^2 + \cdots + \left(x_{m_A} - x_{m_B}\right)^2}$$

- There are other distance measures (e.g., Manhattan, Minkowski, Mahalanobis, Chebyshev), but the Euclidean distance is most often used in the k-nearest neighbors method.



$$d(A,B) = \sqrt{(2-7)^2 + (3-6)^2} = \sqrt{34} \approx 5{,}83$$

K-nearest neighbors

Selection of k

- There is no universal way to select k that will work in every data set – the selection of k depends on the characteristics of the data.
- Selecting different values of k and checking the quality of the obtained classification results on test data or using cross-validation can help determine the best value of k.

Low k value	High k value
Well-structured data	Poorly visible class structure (blurred boundaries between classes)
Low data noise	High data noise

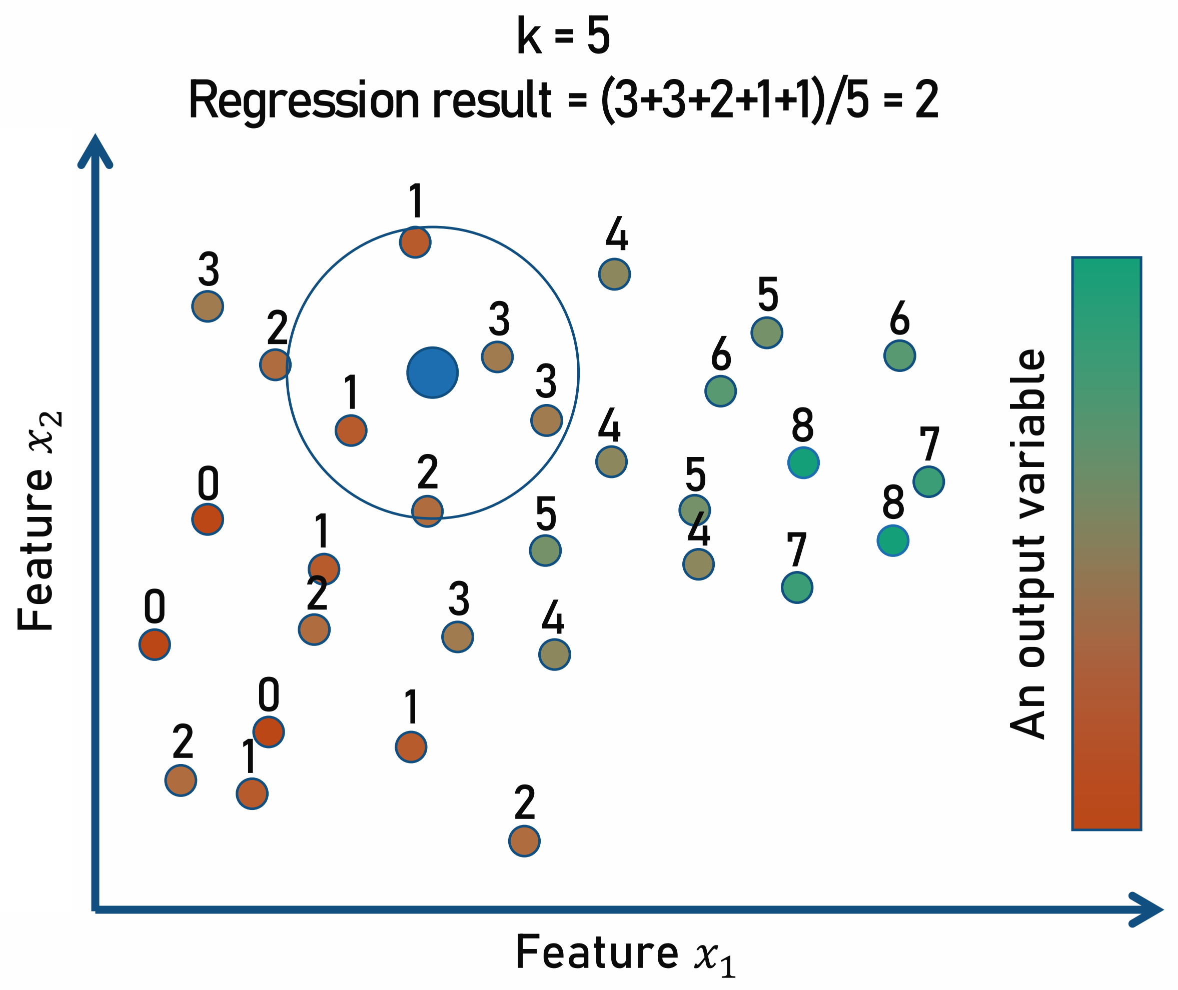
K-nearest neighbors

Application in regression

The k-nearest neighbors technique can also be used when predicting quantitative values.

For an object for which we want to predict a numerical value, we should:

1. Calculate the distances between this object and every other object in the data set;
2. Find the k objects that are closest to this object;
3. The result of the regression will be the average value of the output variable of all k nearest neighbors of this object.



Classification tree

Creating a decision node

Goal: to determine the node that best divides the data set into subsets – the best division means that the subsets will be as pure as possible (homogeneous in terms of classes).

STEP 1: calculate the impurity measure of the data set – the Gini index is most often used (the lower the index value, the lower the impurity):

$$Gini = 1 - \sum_{i=1}^{K} p_i^2$$

where p_i is the percentage of objects belonging to class i in the data set.

$$Gini = 1 - \underset{\overset{\uparrow}{\text{Percentage of class : A}}}{\left(\frac{7}{15}\right)^2} - \underset{\overset{\uparrow}{\text{B}}}{\left(\frac{8}{15}\right)^2} \approx 0{,}498$$

Data on bank customers

age	income	club membe rship	class
25	40	1	A
30	50	0	A
45	80	1	B
22	35	1	B
36	70	0	B
50	100	0	B
28	42	1	A
33	55	0	A
48	85	1	B
21	30	1	A
39	75	0	B
41	90	0	A
26	45	1	A
35	60	1	B
47	96	0	B

Classification tree

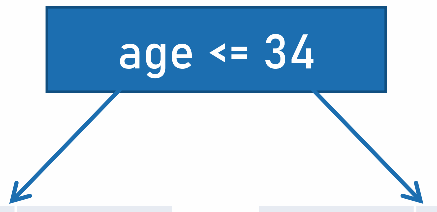
Creating a decision node

STEP 2: testing possible splits – for each input variable we test all possible split points. We will choose the split that minimizes the Gini index value.

Left branch: $Gini = 1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2 \approx 0{,}245$

Right branch: $Gini = 1 - \left(\frac{1}{8}\right)^2 - \left(\frac{7}{8}\right)^2 \approx 0{,}219$

An example of division



age	income	club membership	class
25	40	1	A
30	50	0	A
22	35	1	B
28	42	1	A
33	55	0	A
21	30	1	A
26	45	1	A

age	income	club membership	class
45	80	1	B
36	70	0	B
50	100	0	B
48	85	1	B
39	75	0	B
41	90	0	A
35	60	1	B
47	96	0	B

Classification tree

Creating a decision node

STEP 3: Choosing the best partition. We choose the split that gives the smallest $Gini_{split}$ value. This will give us the purest subsets in both branches.

$$Gini_{split} = \frac{7}{15} \cdot 0{,}245 + \frac{8}{15} \cdot 0{,}219 \approx 0{,}231$$

We perform all 3 steps of the procedure on the two obtained subsets (left and right branches).

The tree construction will stop when all nodes are clean (leaf) or when stopping conditions are reached.

An example of division

age <= 34

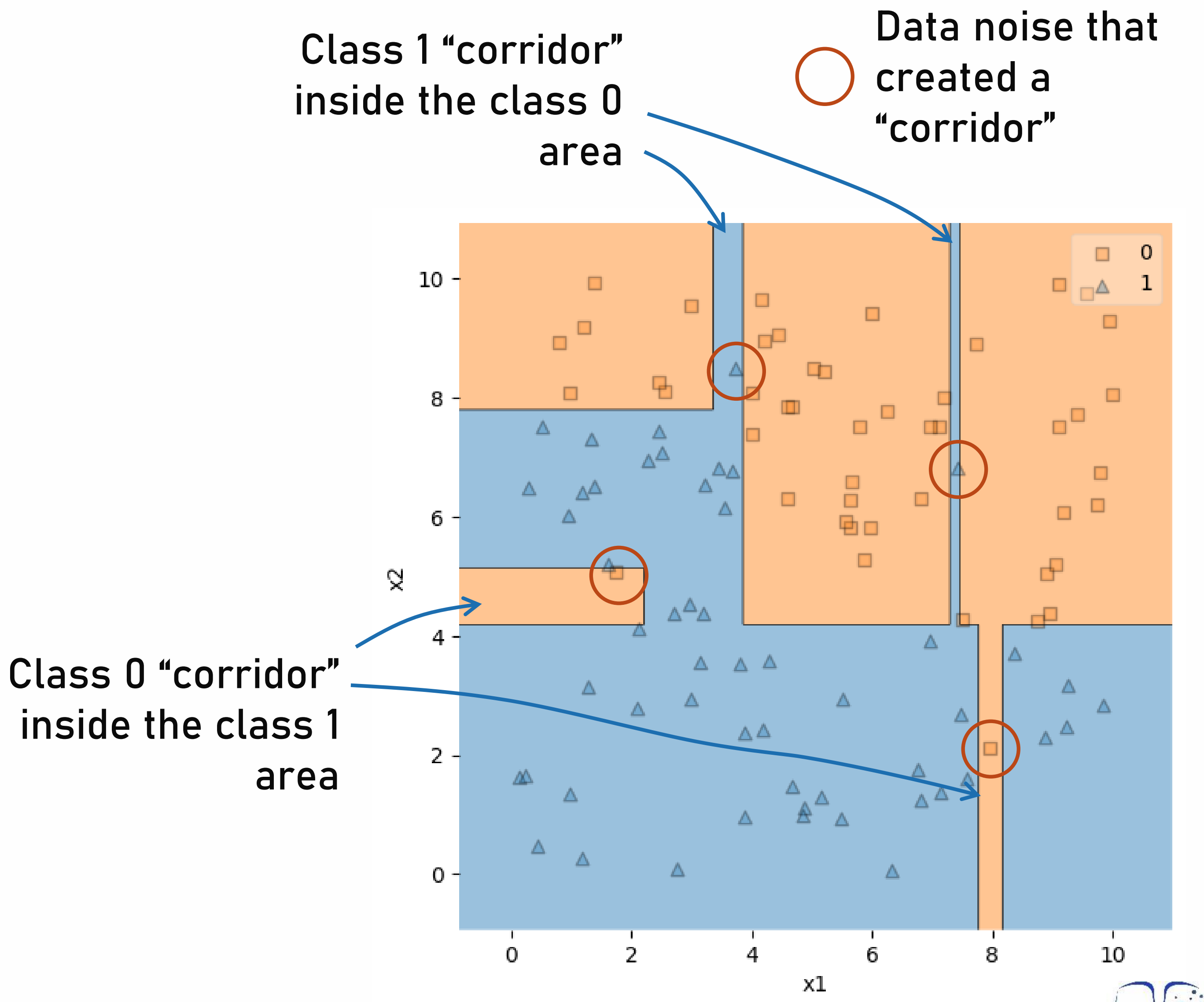
age	income	club membership	class
25	40	1	A
30	50	0	A
22	35	1	B
28	42	1	A
33	55	0	A
21	30	1	A
26	45	1	A

age	income	club membership	class
45	80	1	B
36	70	0	B
50	100	0	B
48	85	1	B
39	75	0	B
41	90	0	A
35	60	1	B
47	96	0	B

Classification tree

Classification tree overfitting

- The algorithm that builds a decision tree is a greedy algorithm – an algorithm set with default hyperparameters will run until it creates a tree that is maximally adapted to the training data (it assigns all objects to maximally pure leaves).
- As a result, the classification tree also adapts to the noise in the data, which can be observed in the form of "corridors" on the graph showing the boundaries established by the classification tree.

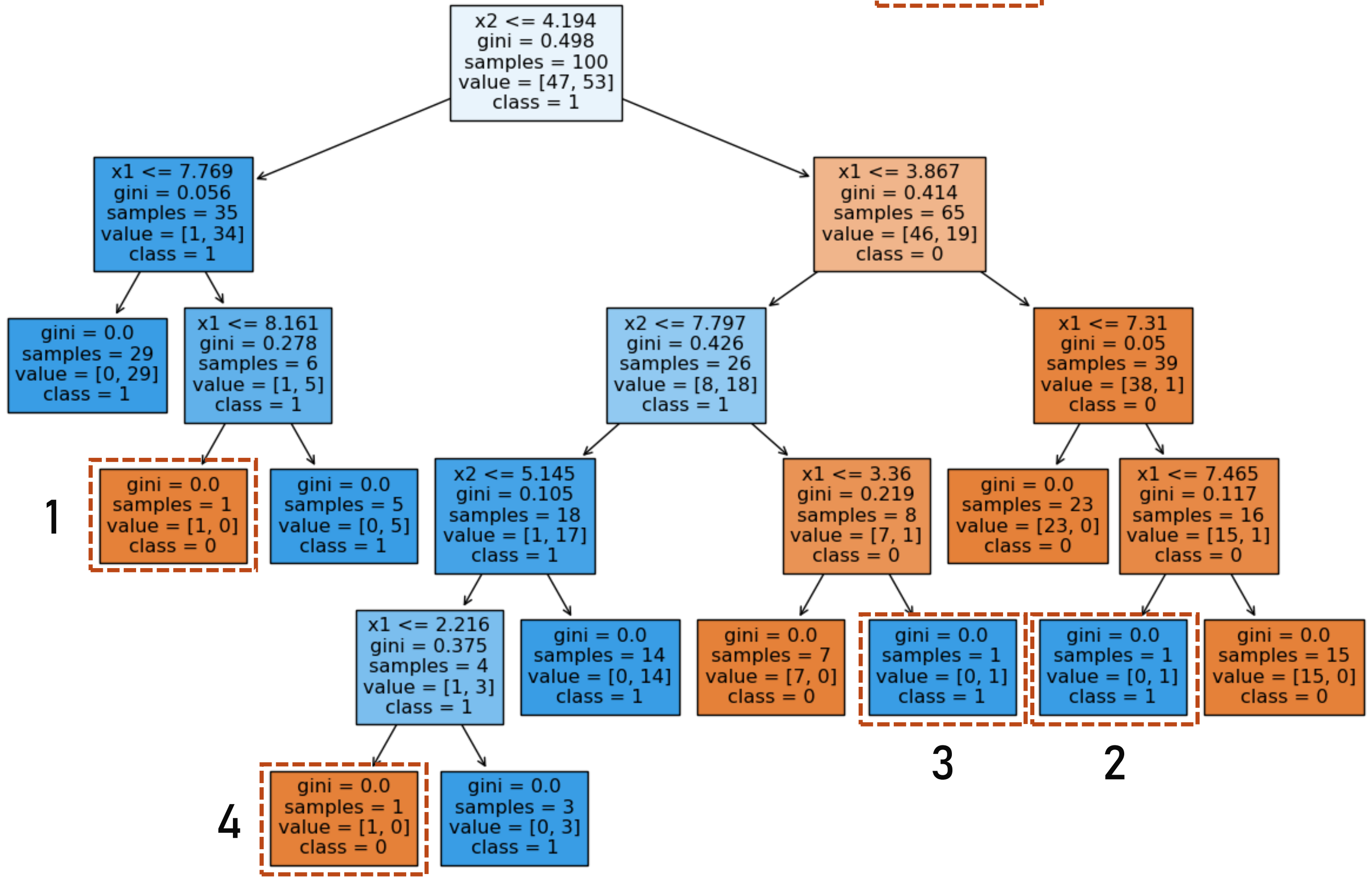


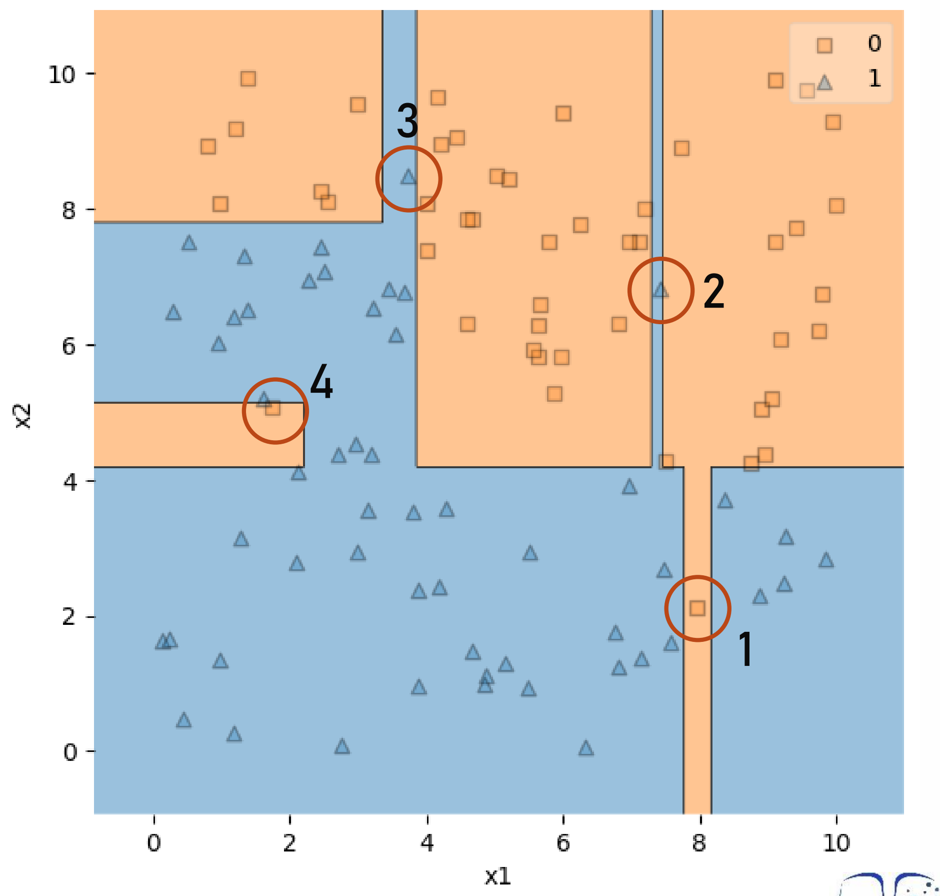
Classification tree

Classification tree overfitting

▭ Leaf containing noise in the data

○ Data noise that created a "corridor"

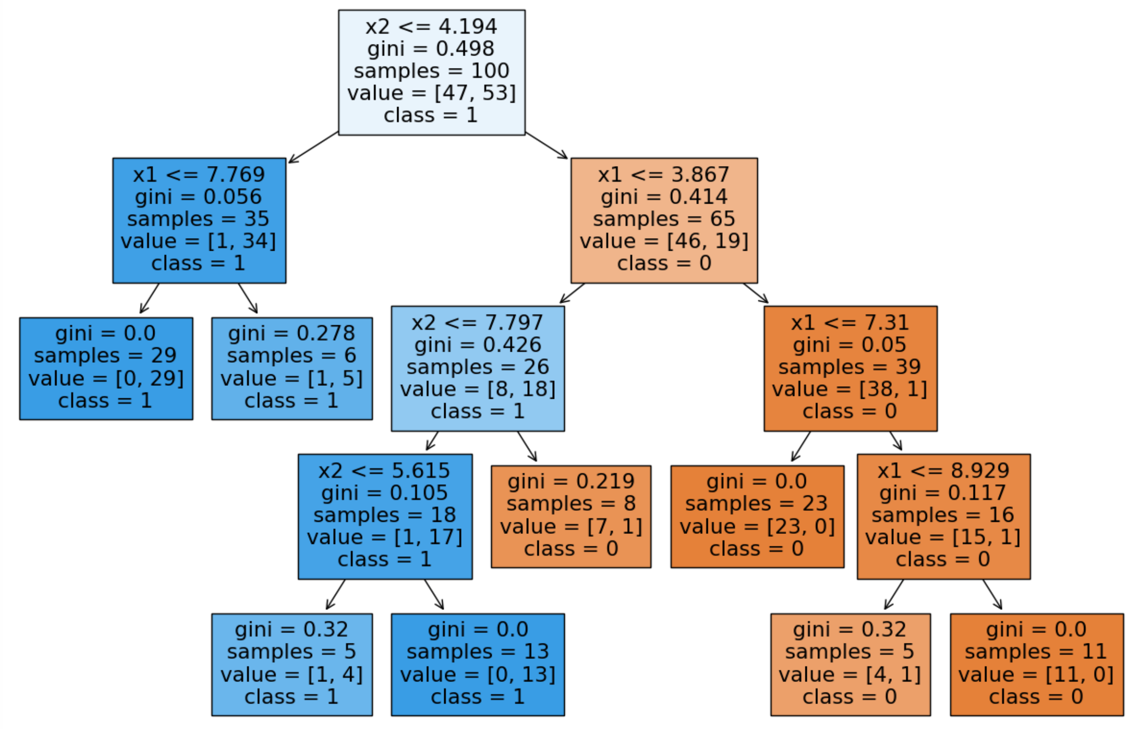


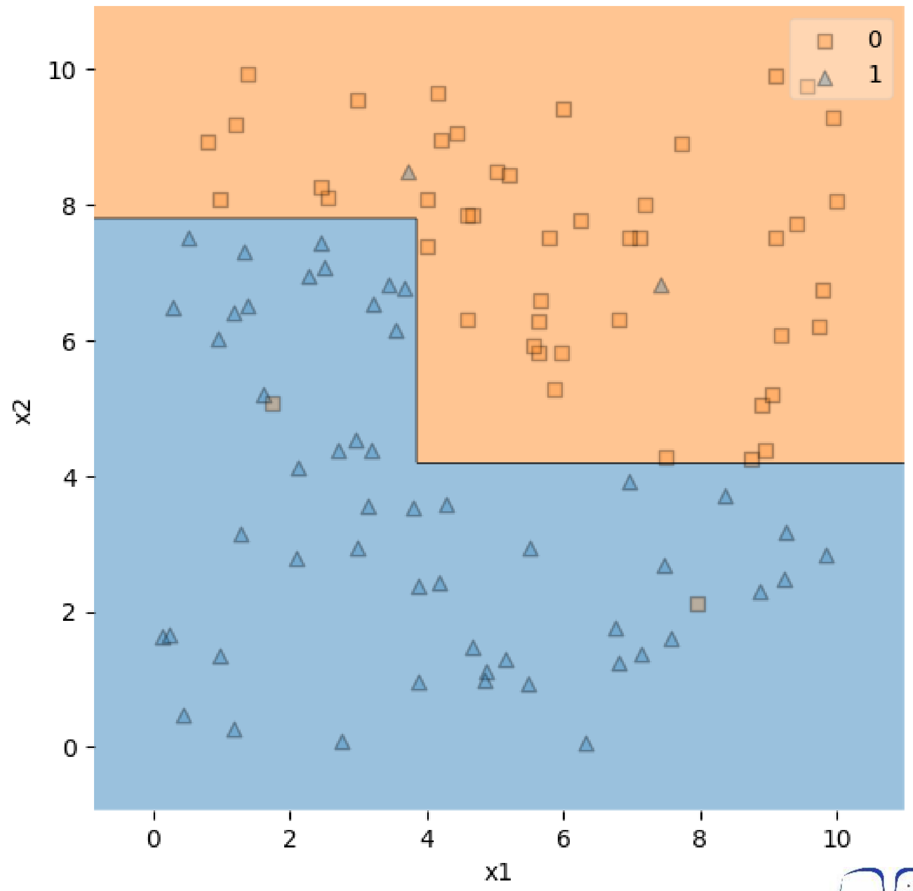


Classification tree

Classification tree regularization

The solution to the problem of overfitting a tree can be appropriate tree regularization (pruning), using appropriate hyperparameter values.





Classification tree

Regularization during tree creation

Hyperparameters used to limit tree growth during tree creation (names taken from the DecisionTreeClassfier – Python sklearn.tree library):

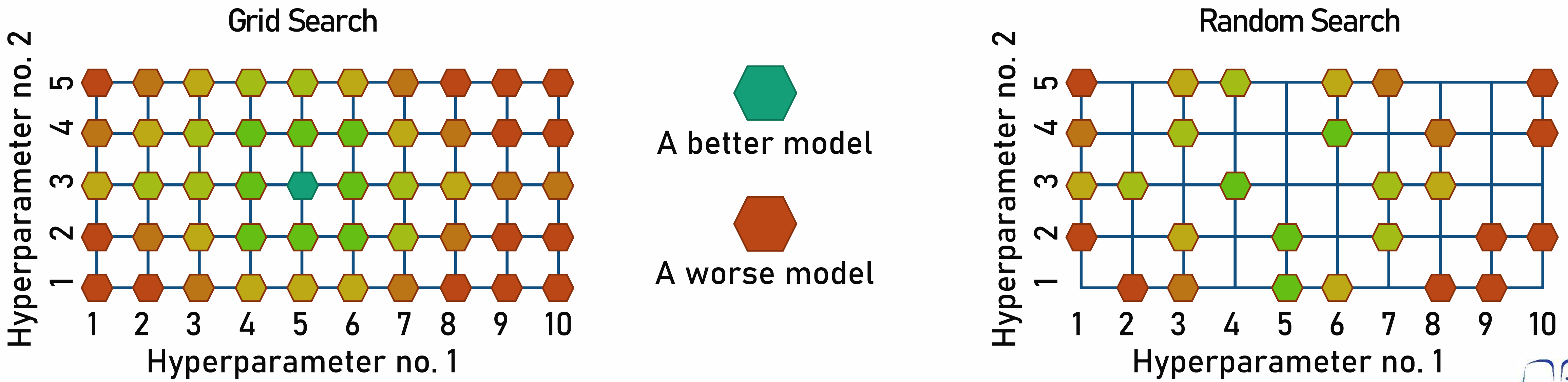
- max_depth - maximum depth (height) of the tree;
- min_samples_split - minimum number of objects that must be in a decision node so that it can be split into lower-level nodes;
- min_samples_leaf - minimum number of objects that must be in each leaf;
- min_weight_fraction_leaf - as in min_samples_leaf, but here the fraction of the total number of samples is taken into account;
- max_leaf_nodes - maximum number of leaves.

The size of the tree can also be indirectly influenced by the max_features hyperparameter, which specifies the number of features (input variables) taken into account when creating each decision node.

Classification tree

Selection of hyperparameters

- Grid Search Cross Validation allows us to use cross validation to search for the best combination of hyperparameters for a machine learning model.
- With a large number of parameters and their values, it is necessary to create and test a great many models, which is time-consuming and may require high computing power. In such situations, random search of parameter combinations (Random Search Cross Validation) is often used.



Regression tree

Creating a decision node

Goal: to determine the node that best divides the data set into subsets – the best partition means that the subsets will best reduce the mean squared error of prediction.

STEP 1: Calculate the Mean Square Error (MSE) of the parent node:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2$$

where: y_i is the value of the output variable of a given object, and $\hat{y}$ is the prediction result treated as the average value of the output variable of all objects.

$$MSE = \frac{1}{15}\sum_{i=1}^{15}(y_i - 150)^2 = 2613.33$$

Data on bank customers

age	income	club membership	deposit amount
25	40	1	150
30	50	0	140
45	80	1	210
22	35	1	80
36	70	0	190
50	100	0	210
28	42	1	100
33	55	0	160
48	85	1	200
21	30	1	30
39	75	0	160
41	90	0	180
26	45	1	90
35	60	1	170
47	96	0	180

Regression tree

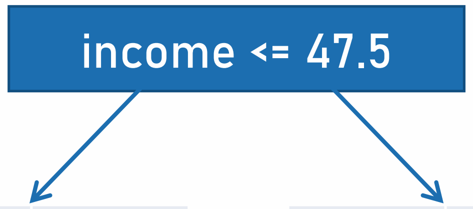
Creating a decision node

STEP 2: testing possible splits – for each input variable we test all possible split points. Each split gives two branches, and for each of them we calculate MSE.

Left branch: $$MSE = \frac{1}{5}\sum_{i=1}^{5}(y_i - 90)^2 = 1480$$

Right branch: $$MSE = \frac{1}{10}\sum_{i=1}^{10}(y_i - 180)^2 = 480$$

An example of division



age	income	club membership	deposit amount
25	40	1	150
22	35	1	80
28	42	1	100
21	30	1	30
26	45	1	90

age	income	club membership	deposit amount
30	50	0	140
45	80	1	210
36	70	0	190
50	100	0	210
33	55	0	160
48	85	1	200
39	75	0	160
41	90	0	180
35	60	1	170
47	96	0	180

Regression tree

Creating a decision node

STEP 3: Choosing the best partition. We choose the partition that best reduces MSE :

$$MSE\ reduction = MSE_P - \left(\frac{n_L}{n_P} \cdot MSE_L + \frac{n_R}{n_P} \cdot MSE_R\right)$$

where: P denotes the parent node, L – the left node, R – the right node.

$$MSE\ reduction = 2613.33 - \left(\frac{5}{15} \cdot 1480 + \frac{10}{15} \cdot 480\right) \approx$$
$$\approx 1800$$

We perform all 3 steps of the procedure on the two obtained subsets (left and right branches).

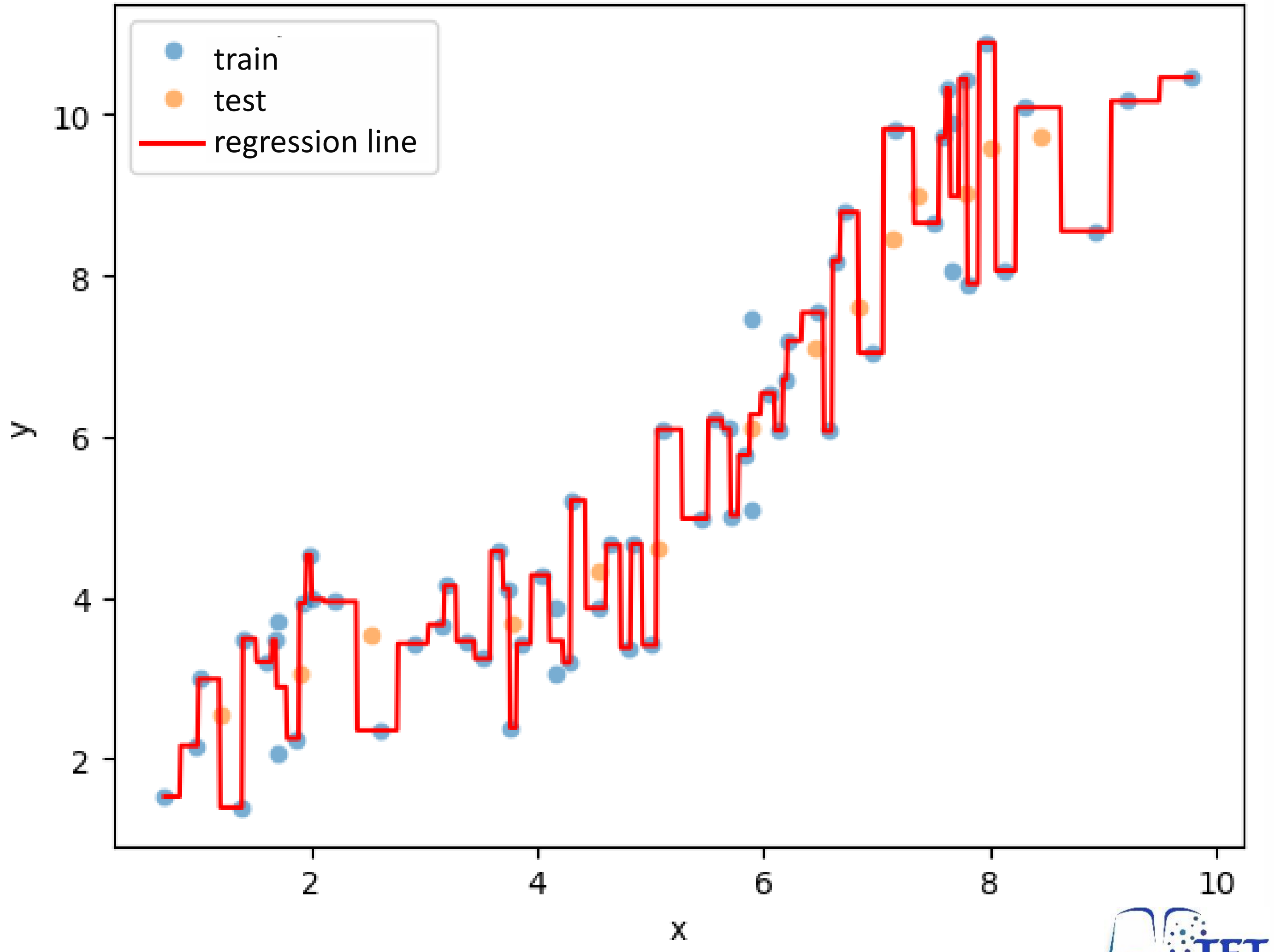
The tree construction will end when all nodes have the minimum achievable MSE value or when stopping conditions are reached.

Regression tree

Regression tree overfitting

- The algorithm that builds a decision tree is a greedy algorithm – an algorithm set with default hyperparameters will run until it creates a tree that is maximally fitted to the training data (assigns all objects to leaves with the minimum MSE value).
- As a result, the regression tree adapts to every object in the training data set (also to the noise in the data).
- Large fluctuations in the regression line suggest that the model is over-fitted to the data.

Regression result using a tree maximally fitted to the training data (61 leaves)

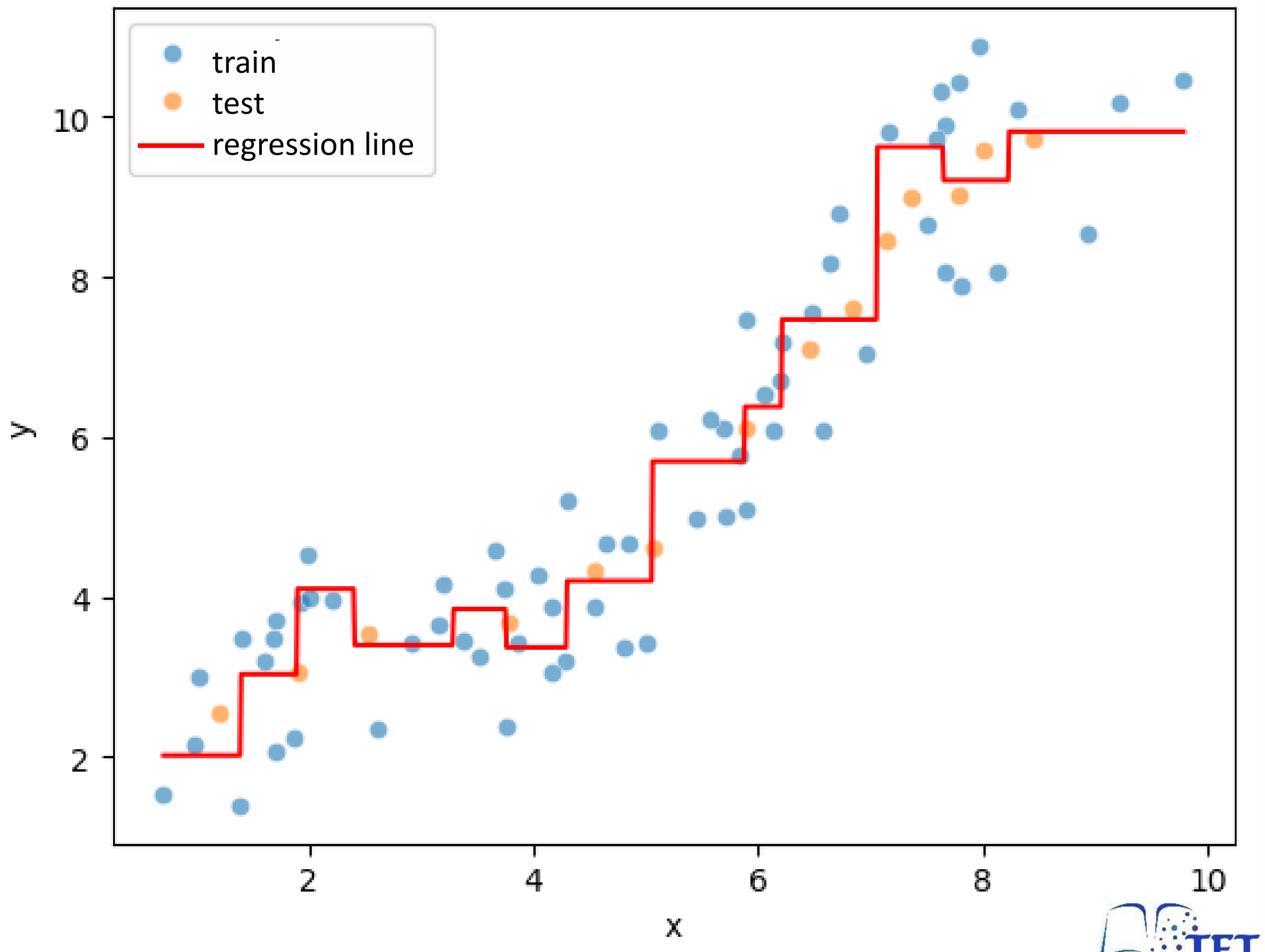


Regression tree

Regression tree regularization

- Regularizing a tree, which involves pruning it (reducing the number of decision nodes and leaves), will reduce the fit to the training data, but may make it easier to find a relationship between the output variable and the input variable (or input variables).
- This may result in improved prediction quality on the test set.
- The hyperparameters used to regularize a regression tree are the same as those used for a classification tree.

Regression result using a regularized tree (13 leaves)

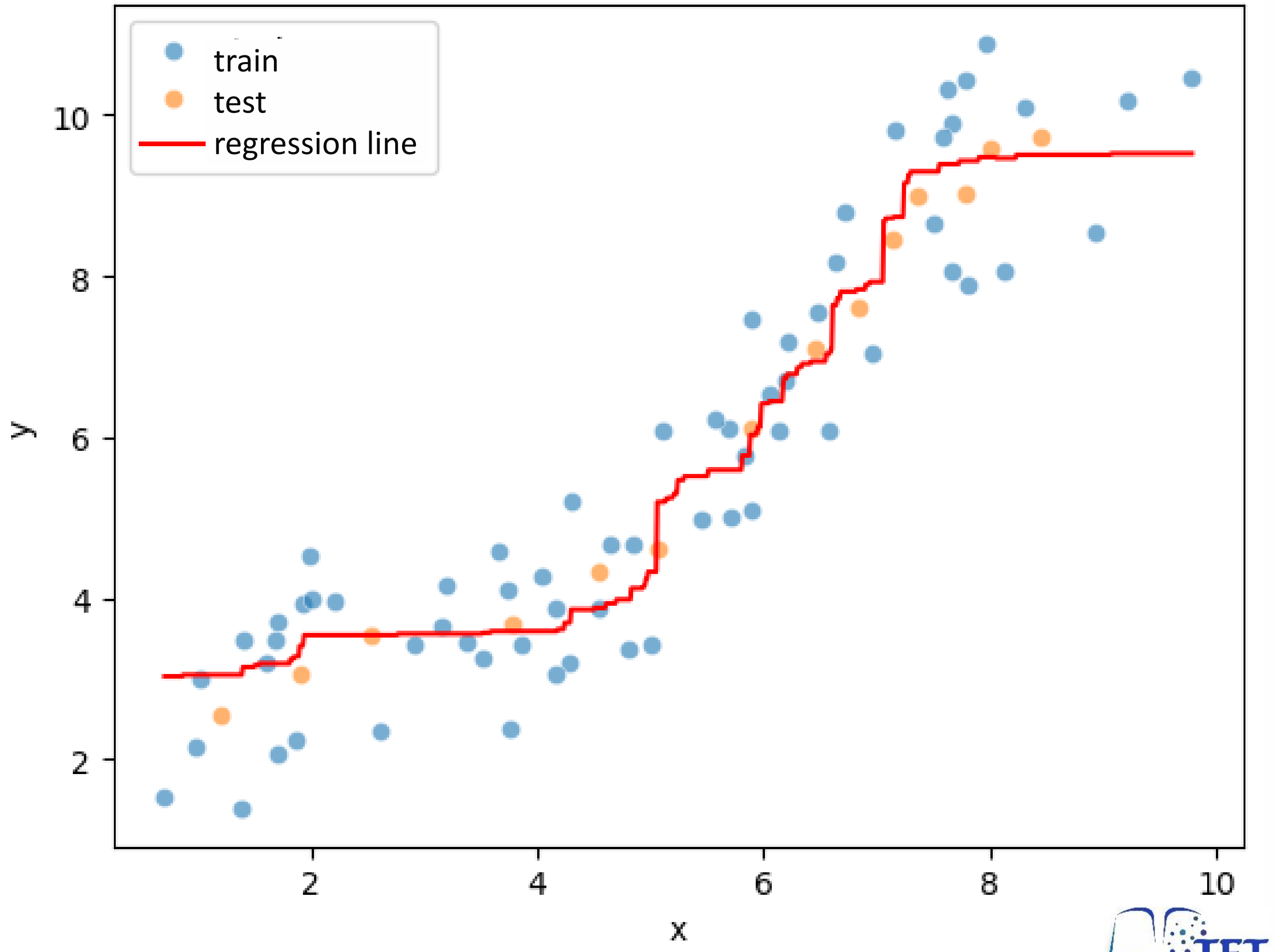


Random forest

Characteristics

- Random forest is an ensemble model based on many simple (non-extensive) decision trees.
- The key hyperparameter is the number of trees in the model (n_estimators) – the more trees, the greater the stability of the prediction results, but the longer the computation time.
- In regression, the prediction result of the random forest model is the average of the prediction results of all trees included in the forest.

Random forest regression result
(100 trees with maximum depth of 2)



Random forest

Bootstrap Aggregating (bagging)

Bagging is a technique for creating ensemble models that consists of two stages:

- **Stage 1:** random subsets of data are created (bootstrap samples) and a different model (decision tree) is trained on each of them.
- **Stage 2:** combining the prediction results of the individual models in the ensemble.

In random forests, the bagging procedure is additionally enriched with a random selection of features as in the case of the max_features hyperparameter.

Advantages of bagging in random forests:

- **Variance reduction:** individual trees can have high variance; the forest, by combining the prediction results of individual trees, eliminates random errors resulting from differences in training samples.
- **Overfitting reduction:** trees that create the forest are trained on random subsets, which reduces the risk of the random forest fitting to noise in the data.

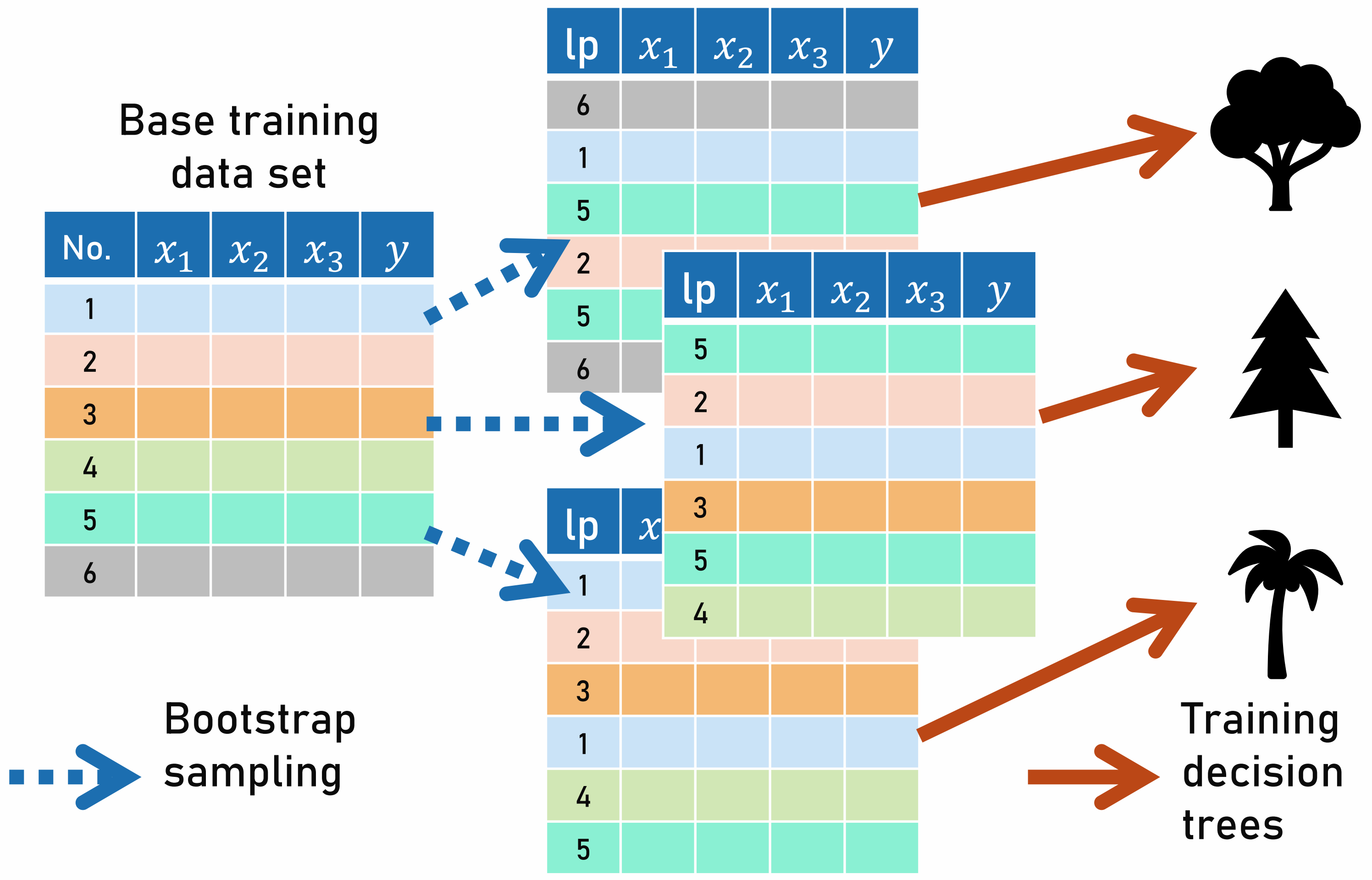
Random forest

Bootstrap Aggregating (bagging): stage 1

Bootstrap stage

An example for
`n_estimators = 3`

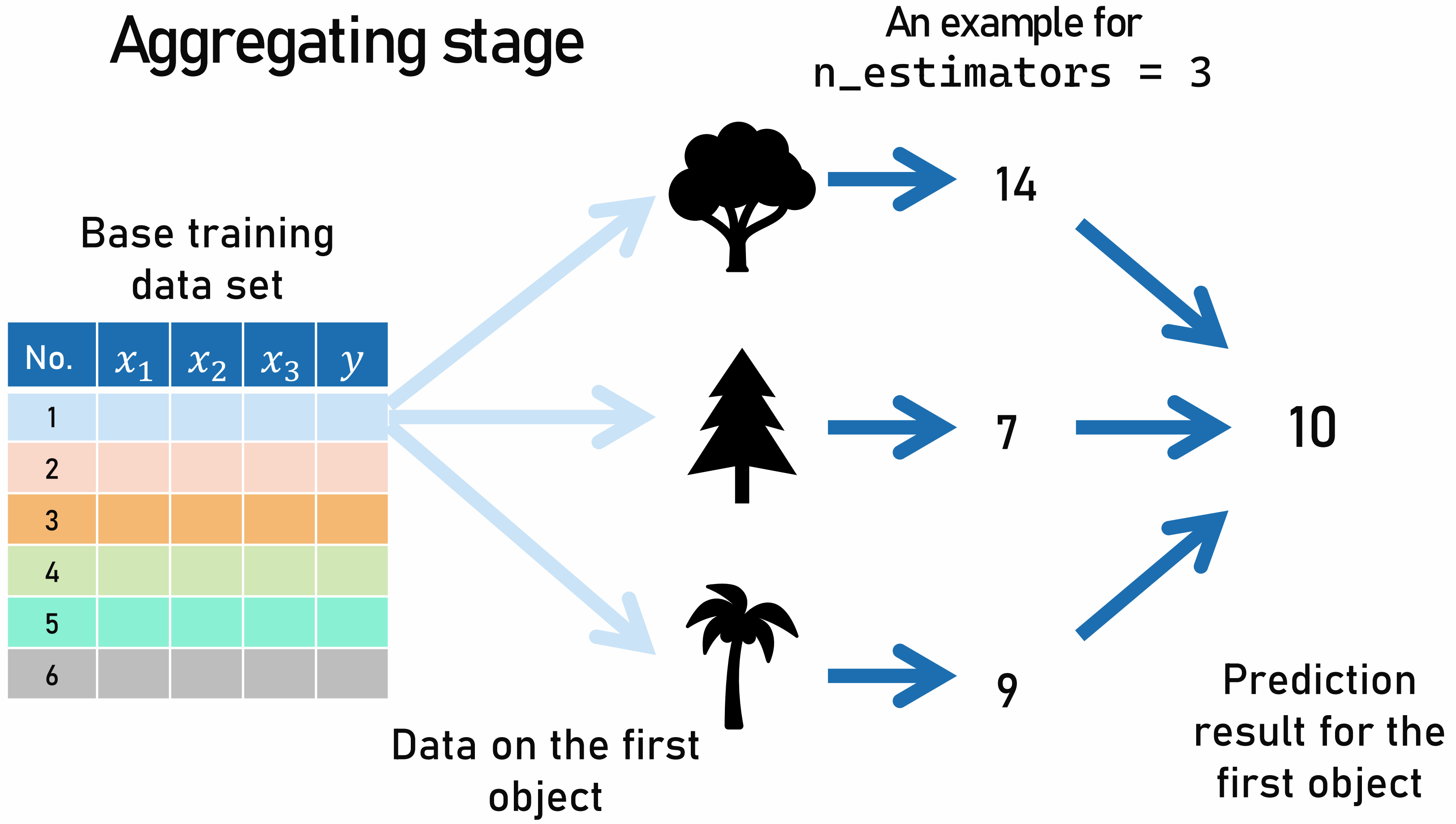
- Creating random training sets (as many sets as there are trees in the random forest);
- Each set is drawn with replacement;
- The size of each random set is equal to the size of the base training data set;
- A different decision tree is trained on each random set.



Random forest

Bootstrap Aggregating (bagging): stage 2

- Trained models calculate a prediction for each object in the data set;
- Calculating the prediction for a given object:
  - **In regression:** the average of the predictions of all models.
  - **In classification:** majority voting – each model predicts the class of a given object, and the majority class is selected as the final prediction.



No.	x_1	x_2	x_3	y
1				
2				
3				
4				
5				
6				

Thank you for your attention!

TET