Logistic Regression

Introduction

- logistic models can modify raw data streams to produce characteristics for various AI and machine learning methods.
- This type of regression is a good choice when modeling binary variables, which happen frequently in real life
- In reality, one of the often employed machine learning techniques for **binary classification issues**, or problems with two class values, includes logistic regression
- These problems include predictions like "*this or that,*" "*yes or no,*" , "*A or B, work or don't work, marry or don't marry, buy a house or ren.*"
- The probability of occurrences can also be estimated using logistic regression, which includes establishing a link between *feature likelihood* and *outcome likelihood*.

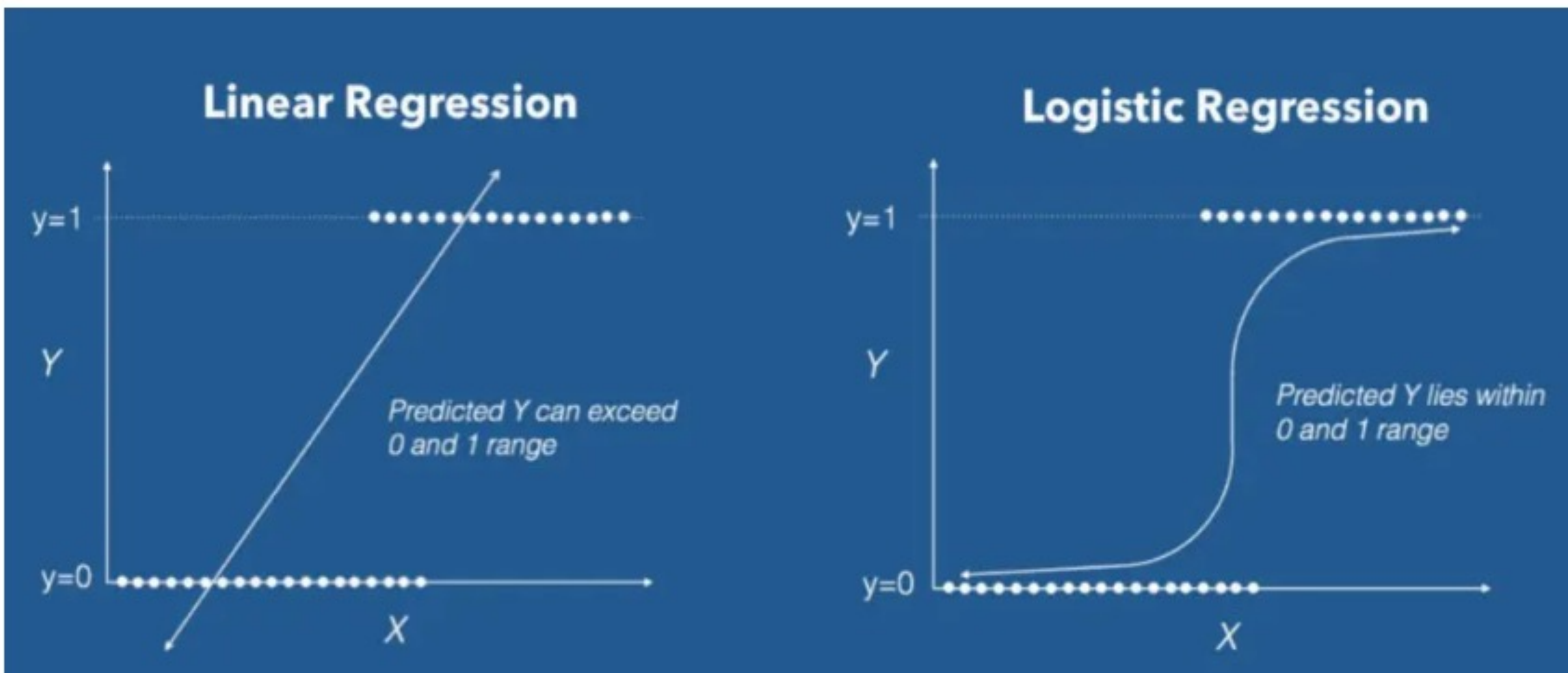
- The logistic regression model is popular, in part, because it gives probabilities between 0 and 1
- For instance
  - it can be applied for categorization by building a model that links the *number of hours of study to the likelihood that a student would pass or fail.*
  - *modeling a risk of credit default*: values closer to 0 indicate a tiny risk, while values closer to 1 mean a very high risk

Comparison of linear regression and logistic regression

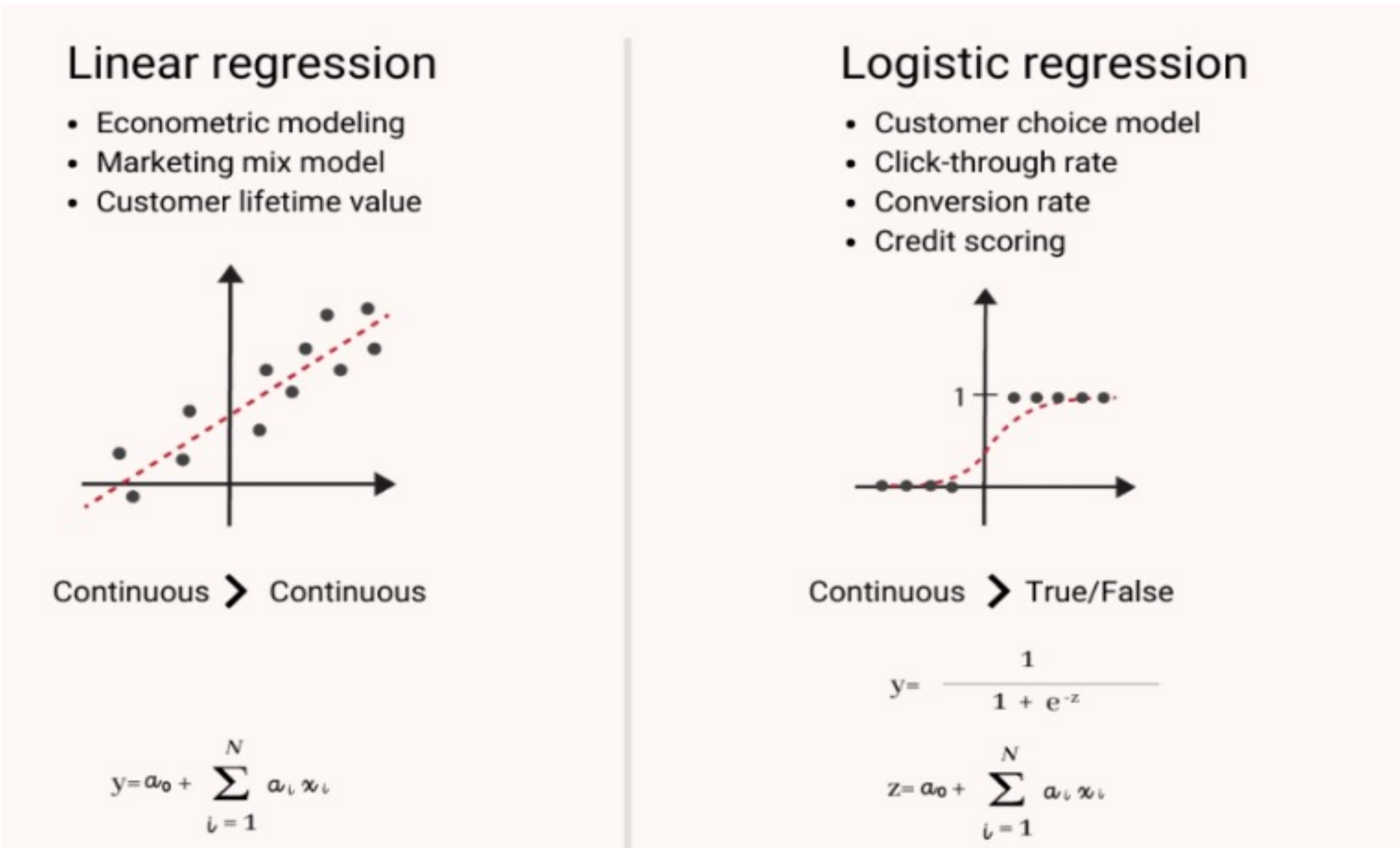
- The primary distinction between logistic and linear regression is that the output of logistic regression is *constant whereas* the output of linear regression is *continuous*.
- The outcome, or dependent variable, in logistic regression has just two *possible values*. However, the output of a linear regression is continuous, which means that there *are an endless number of possible* values for it.
- When the response variable is categorical, such as *yes/no, true/false, and pass/fail, logistic regression* is utilised. When the response variable is continuous, like *hours, height, or weight, linear regression* is utilised

- Logistic regression and linear regression, for instance, can predict various outcomes depending on the information about the amount of time a student spent studying and the results of their exams.

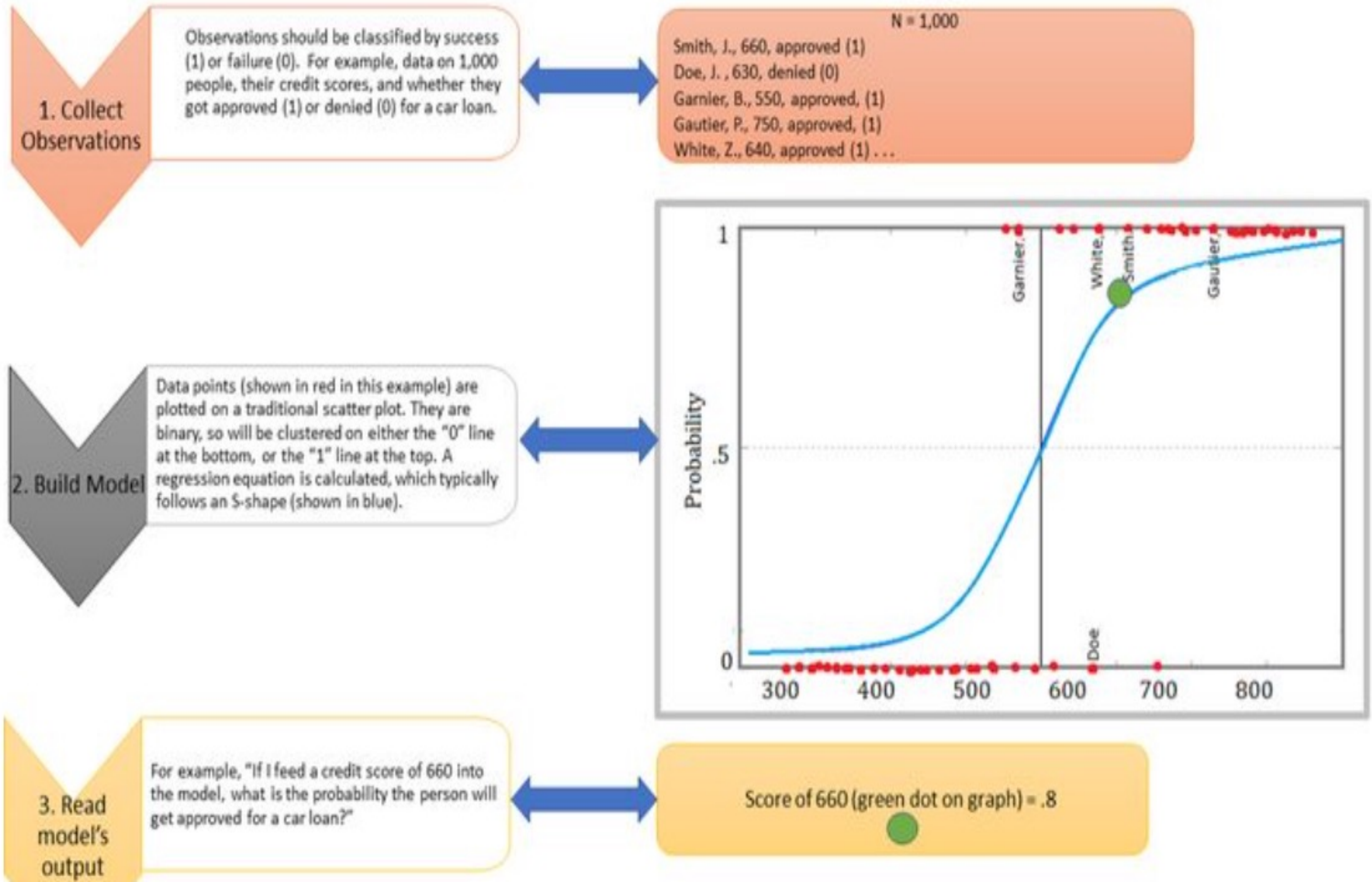
Curve, a visual representation of linear and logistic regression



- An S-shaped curve is revealed using logistic regression. Here, the orientation and **steepness** of the curve are affected by changes in **the regression coefficients**



- The following image shows an example of how one might tailor a logistic model for credit score based risk.



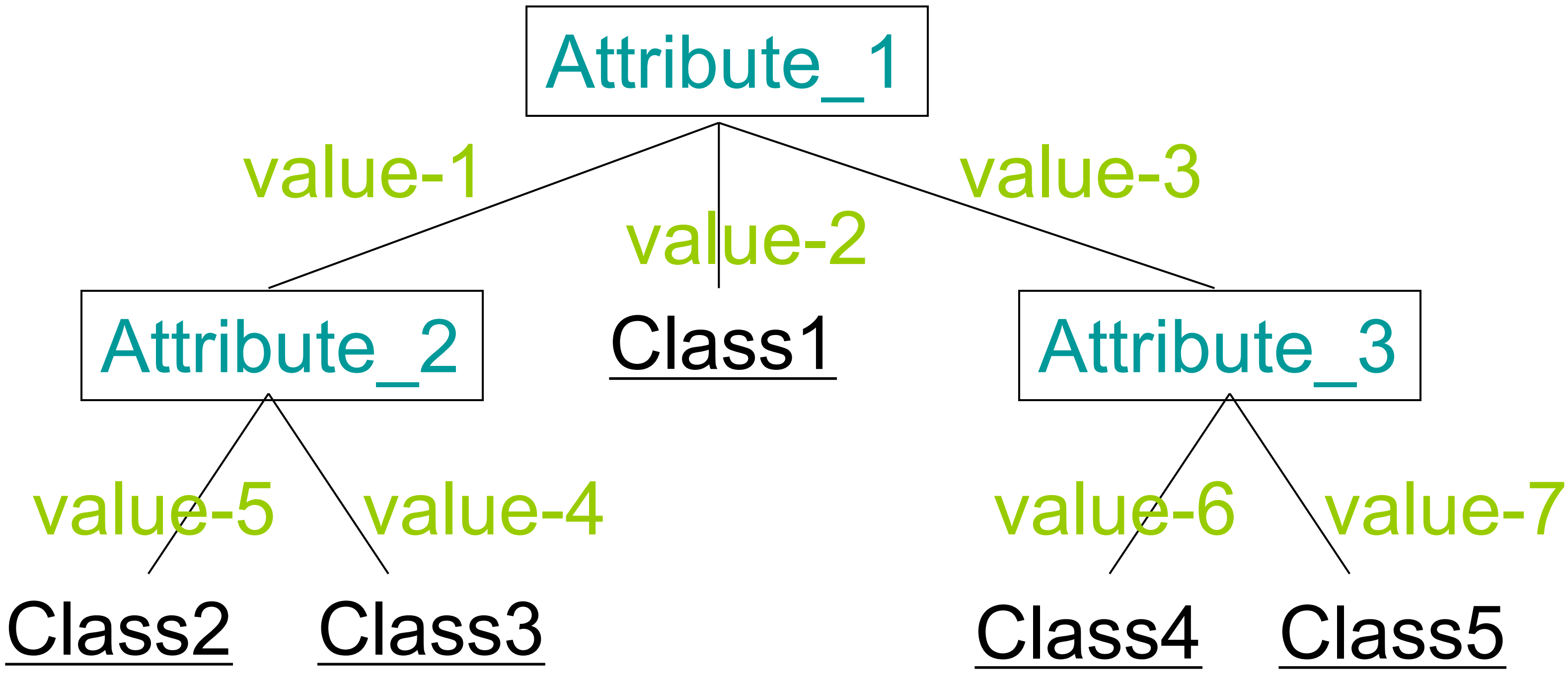
Decision tree classifier

Decision tree classifier

- *Decision tree* performs classification by constructing a tree based on training instances with leaves having class labels.
  - The tree is traversed for each test instance to find a leaf, and the class of the leaf is the predicted class.
  - This is a directed knowledge discovery in the sense that there is a specific field whose value we want to predict.
- Widely used learning method. It has been applied to:
  - classify medical patients based on the disease,
  - loan applicant by likelihood of payment.

Decision Trees

- Tree where internal nodes are simple decision rules on one or more attributes and leaf nodes are predicted class labels;
- Given an instance of an object or situation, which is specified by a set of properties, the tree returns a "yes" or "no" decision about that instance.



Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical
  - Examples are partitioned recursively based on selected attributes
  - Optimal attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

Attribute Selection Measure

Technically, we would like the resulting groups to be as *pure* as possible –*homogeneous with respect to the target variable*. If there is at least one member of the group that has a different value for the target variable than the rest of the group, then the group is impure.

- Information gain
  - Select the attribute with the highest information gain
    - First, compute the disorder using Entropy; the expected information needed to classify objects into classes
    - Second, measure the Information Gain; to calculate by how much the **disorder of a set would reduce** by knowing the value of a particular attribute.
  - In information gain measure we want:-
    - large Gain
    - same as: small average disorder created

Entropy

- The **Entropy** measures the disorder of a set S containing a total of *n* examples of which n_+ are positive and n_- are negative and it is given by:

$$D(n_+, n_-) = -\frac{n_+}{n}\log_2\frac{n_+}{n} - \frac{n_-}{n}\log_2\frac{n_-}{n} = Entropy(S)$$

- Some useful properties of the Entropy:
  - D(n,m) = D(m,n)
  - D(0,m) = 0
    - *D(S)=0* means that all the examples in *S* have the same class
  - D(m,m) = 1
    - *D(S)=1* means that half the examples in *S* are of one class and half are the opposite class

Information Gain

- The Information Gain measures the expected reduction in entropy due to splitting on an attribute A

$$GAIN_{split} = Entropy(S) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, **p** is split into **k** partitions;

n_i is number of records in partition i

Information Gain: Measures Reduction in Entropy achieved because of the split.

- Choose the split that achieves most reduction (maximizes GAIN)

Example: Decision Tree for "buy computer or not". Use the training Dataset given below to construct decision tree

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Attribute Selection by Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- E(P, N) = E(9, 5) =0.940
- Compute the entropy for

age	p_i	n_i	$E(p_i, n_i)$
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

$$E(age) = \frac{5}{14}E(2,3) + \frac{4}{14}E(4,0) + \frac{5}{14}E(3,2) = 0.69$$

Hence

$$Gain(age) = E(p,n) - E(age)$$

Similarly

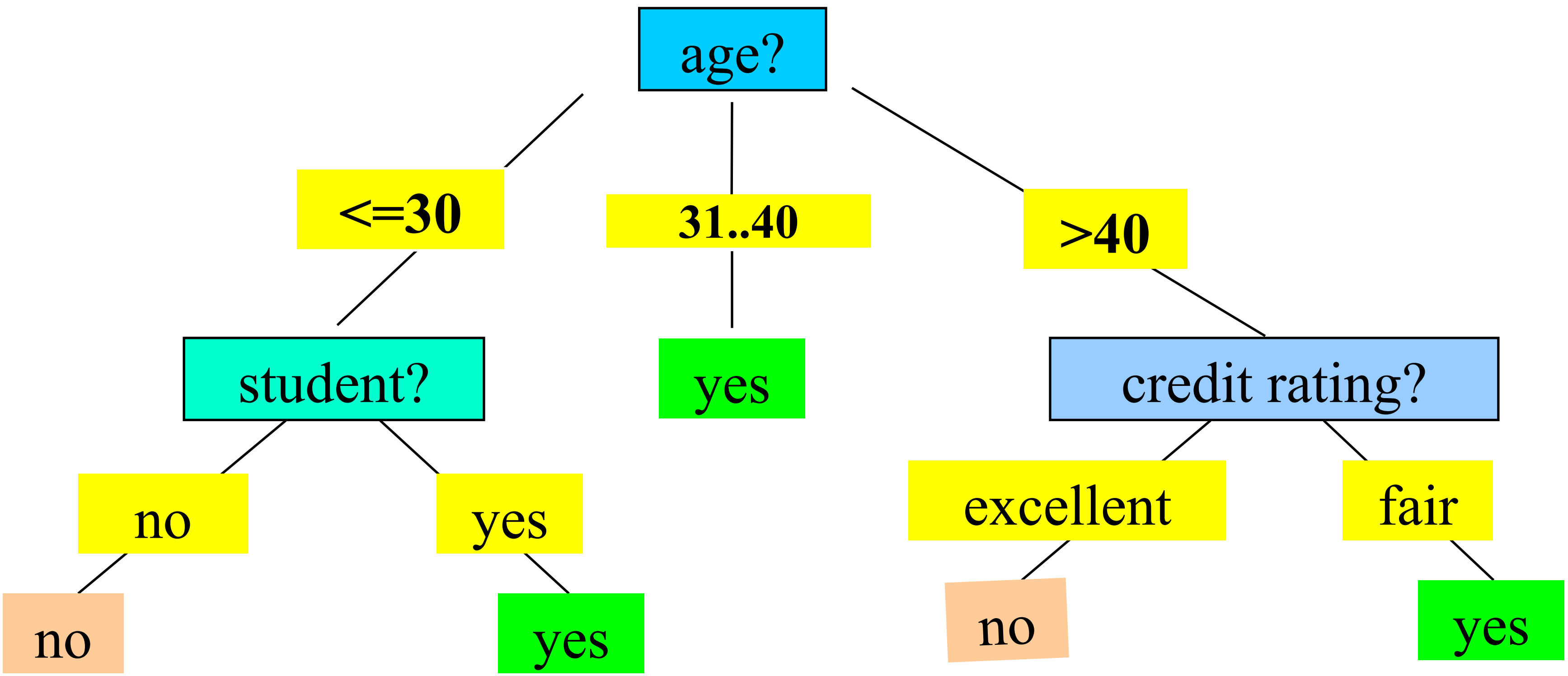
$$Gain(age) = 0.25$$

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

Output: A Decision Tree for “*buys_computer*”



<u>Classification Rules</u>

IF *age* = “<=30” & *student* = “*no*” THEN *buys_computer* = “*no*”
IF *age* = “<=30” & *student* = “*yes*” THEN *buys_computer* = “*yes*”
IF *age* = “31…40” THEN *buys_computer* = “*yes*”
IF *age* = “>40” & *credit_rating* = “*excellent*” THEN *buys_computer* = “*yes*”
IF *age* = “>40” & *credit_rating* = “*fair*” THEN *buys_computer* = “*no*”

Assignment: The problem of "Sunburn"

- You want to predict whether another person is likely to get sunburned if he is back to the beach. How can you do this?
- Data Collected: predict based on the observed properties of the people

Name	Hair	Height	Weight	Lotion	Result
Sarah	Blonde	Average	Light	No	Sunburned
Dana	Blonde	Tall	Average	Yes	*None*
Alex	Brown	Short	Average	Yes	*None*
Annie	Blonde	Short	Average	No	Sunburned
Emily	Red	Average	Heavy	No	Sunburned
Pete	Brown	Tall	Heavy	No	*None*
John	Brown	Average	Heavy	No	*None*
Kate	Blonde	Short	Light	Yes	*None*

Exercise: ‘is the customer Good, *Doubtful or Poor?’*

Customer ID	Debt	Income	Marital Status	Risk
Abel	High	High	Married	Good
Ben	Low	High	Married	Doubtful
Candy	Medium	Low	Unmarried	Poor
Dale	High	Low	Married	Poor
Ellen	High	Low	Married	Poor
Fred	High	Low	Married	Poor
George	Low	High	Unmarried	Doubtful
Harry	Low	Medium	Married	Doubtful
Igor	Low	High	Married	Good
Jack	High	High	Married	Doubtful
Kate	Low	Low	Married	Poor
Lane	Medium	High	Unmarried	Good
Mary	High	Low	Unmarried	Poor
Nancy	Low	Medium	Unmarried	Doubtful
Othello	Medium	High	Unmarried	Good

Ensembling decision trees

1. Bagging

- Bootstrap aggregating, or bagging, is an algorithm which applies bootstrapping to machine learning problems
- Bootstrapping is a statistical procedure that creates multiple datasets from the existing one by sampling data with replacement.
- Bootstrapping can be used to measure the properties of a model, such as bias and variance
- In general, a bagging algorithm follows these steps
  1. We generate new training sets from input training data by sampling with replacement
  2. For each generated training set, we fit a new model
  3. We combine the results of the models by averaging or majority voting
- As you can imagine, bagging can reduce the chance of overfitting

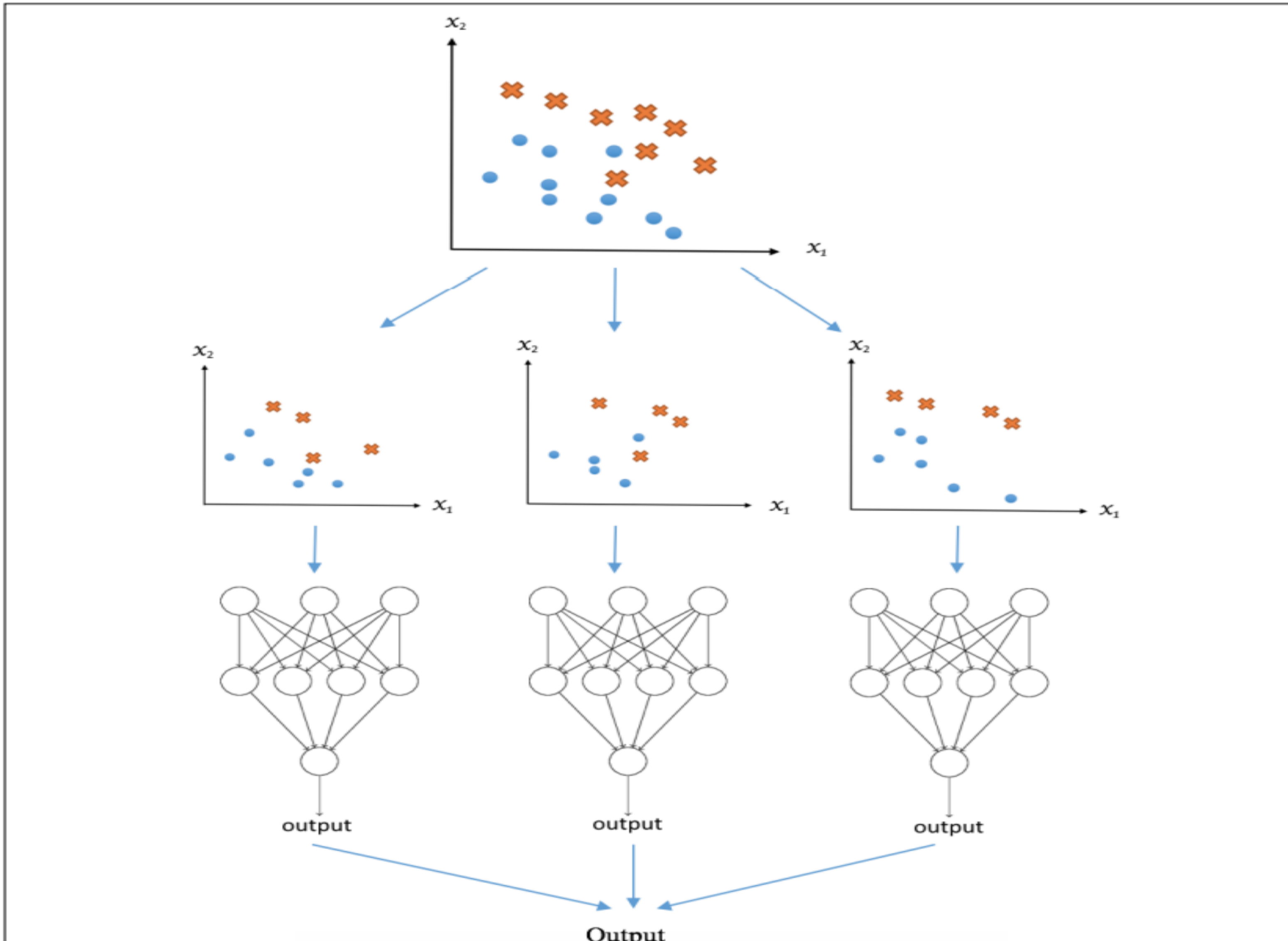


Figure 1.11: Workflow of bagging for classification

- In general, different sets of training samples are randomly drawn with replacement from the original training data;
- each resulting set is used to fit an individual classification model.
- The results of these separately trained models are then combined together through a majority vote to make the final decision

2. Random forest

- Tree bagging, reduces the high variance that a decision tree model suffers from and, hence, performs better than a single tree.
- However, in some cases, where **one or more features are strong** indicators, individual trees are constructed largely based on these features and, as a result, become highly correlated.
- Aggregating multiple correlated trees will not make much difference.
- To force each tree to become uncorrelated, random forest only considers **a random subset of the features** when searching for the best splitting point at each node.
- Individual trees are now trained based on different sequential sets of features, which guarantees more diversity and better performance.
- Random forest is a variant of the tree bagging model with additional **feature-based bagging.**

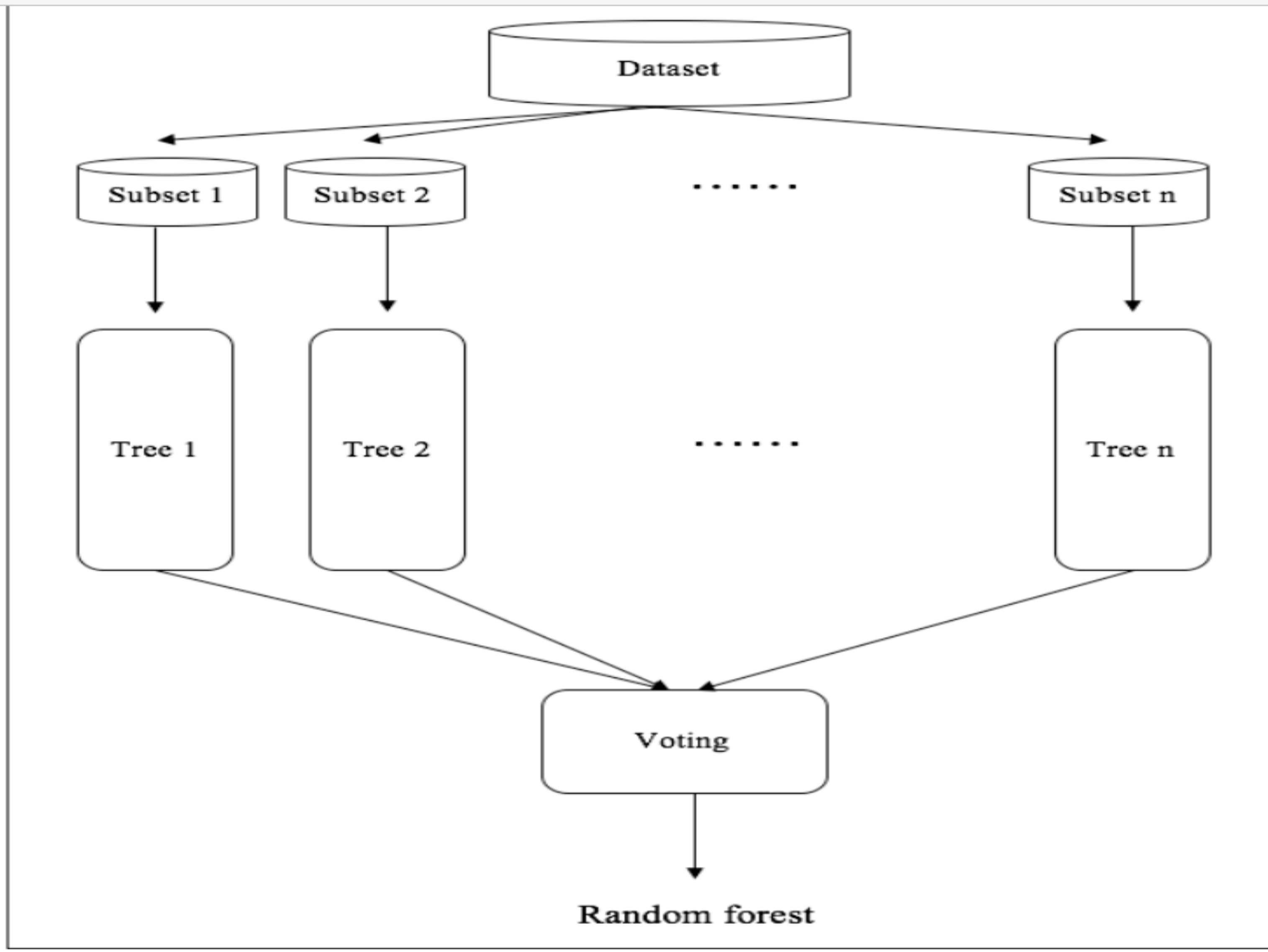


Figure 4.14: The random forest workflow