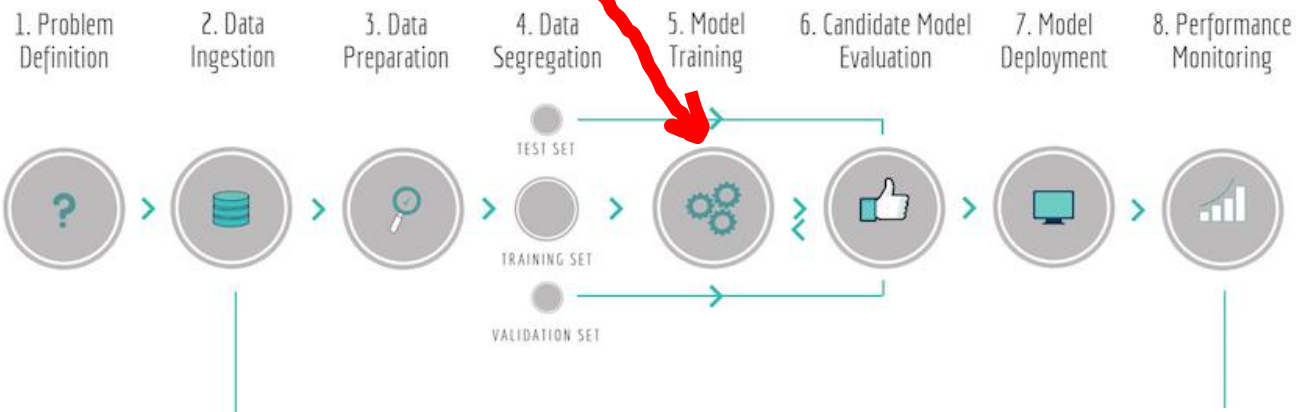
ISLab
Intelligent Synthetic Intelligence Lab
Universidade do Minho
Departamento de
Informática

DADOS e APRENDIZAGEM AUTOMÁTICA

Decision Trees

MESTRADO (integrado) EM ENGENHARIA INFORMÁTICA

Contents

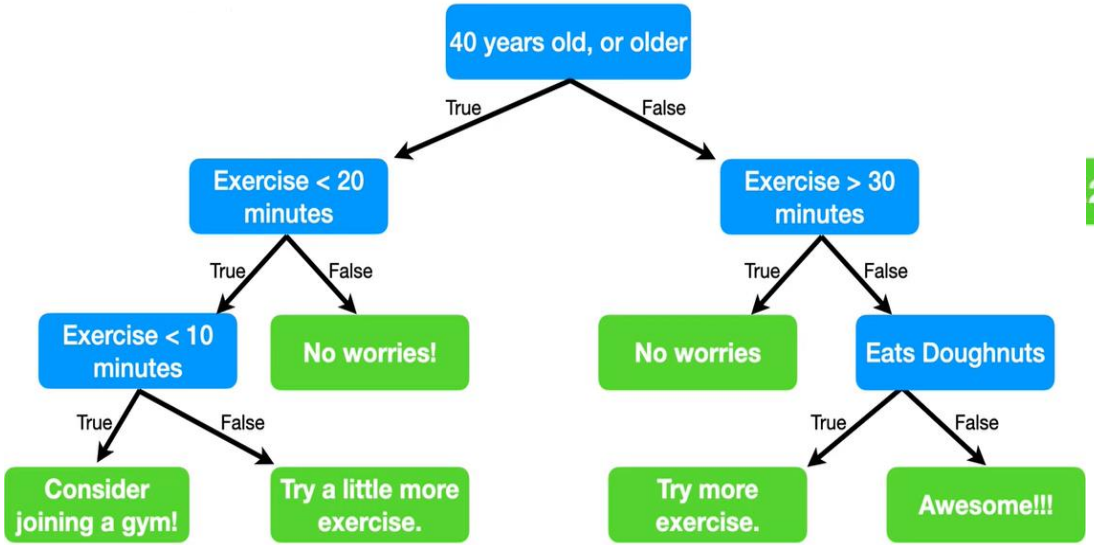


Supervised Learning:

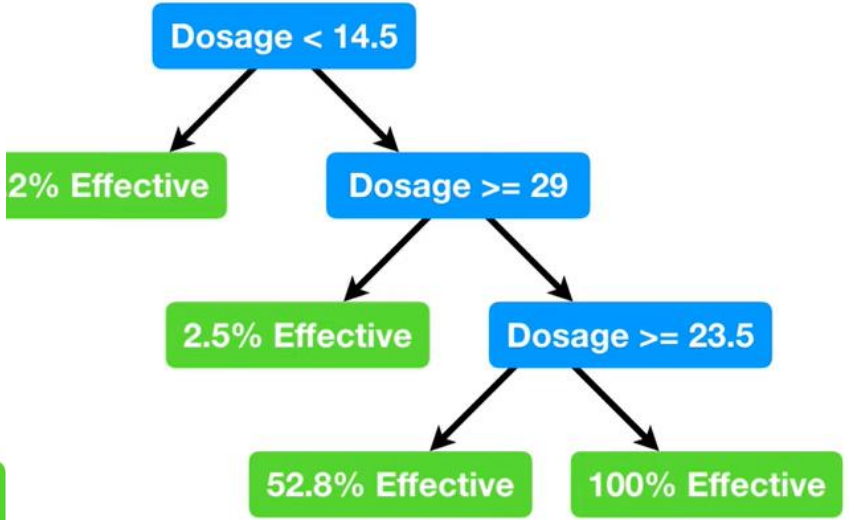
- **Decision Tree**
  - **Classification Tree**
  - **Regression Tree**

Decision Trees

Classification Tree

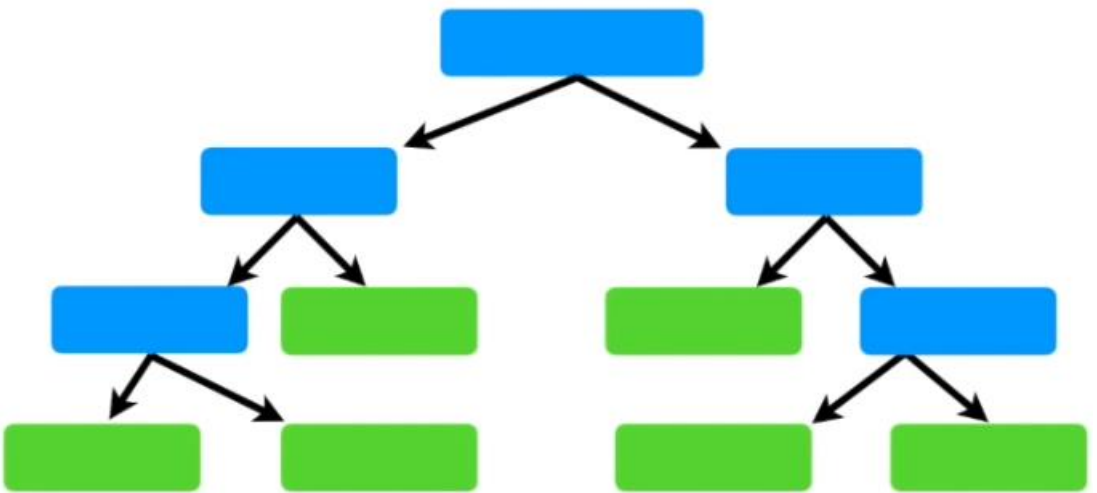


Regression Tree



Classification Trees

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…



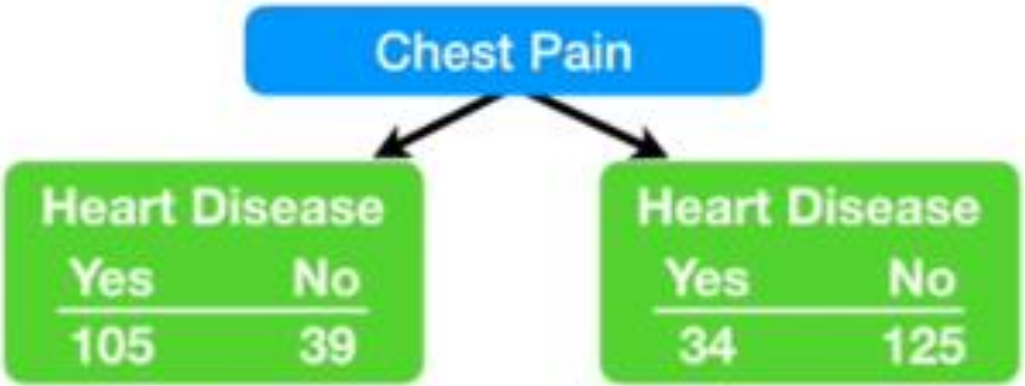
The StatQuest Illustrated Guide to Machine Learning (PDF). (n.d.). https://statquest.gumroad.com/l/wvtmc

Classification Trees

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…

Mesure Impurity:

- Gini Impurity
- Entropy Impurity
- Information Gain



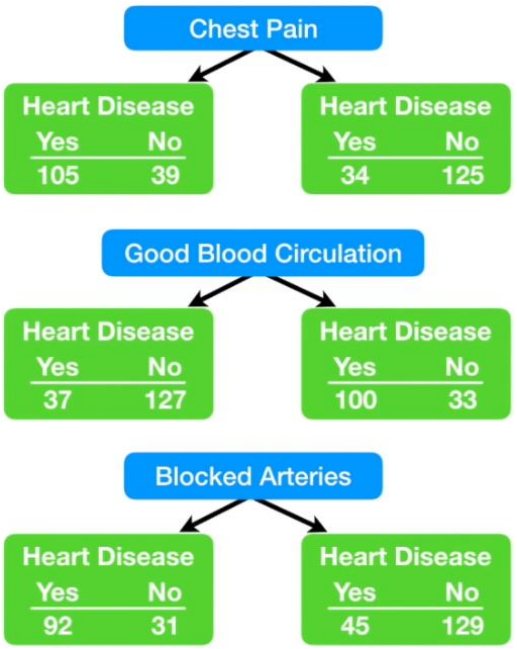
Geni Impurity
$G=1-(P_{yes})^2-(P_{no})^2$
$G = 1-(105/(105+39))^2-(39/(105+39))^2$
$G = 0.395$
= 144

Geni Impurity
$G=1-(P_{yes})^2-(P_{no})^2$
$G = 1-(34/(34+125))^2-(125/(34+125))^2$
$G = 0.336$
#=159

Total Geni Impurity = weighted average of Gini impurities for the Leaves:
TG = 0.395(144/(144+159)) + 0.336(159/(144+159))
TG = 0.364

Classification Trees

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...



$G_{ChestPain} = 0.364$

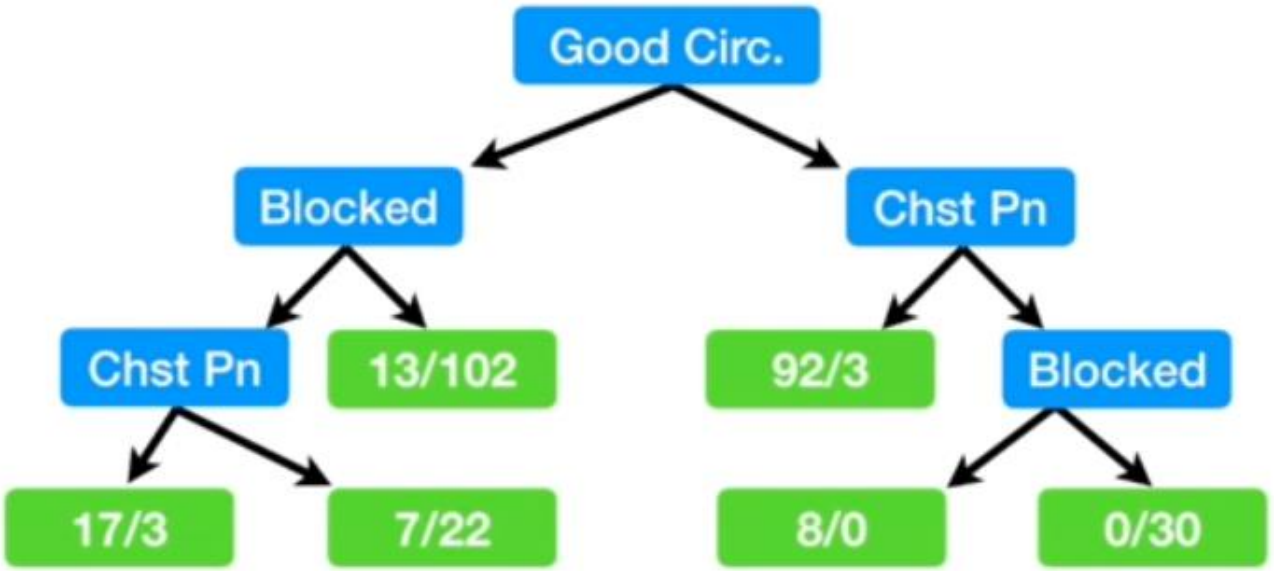
$G_{BloodCirculation} = 0.360$

$G_{BlockedArteries} = 0.381$

Method:
1 – calculate the Geni Impurity scores
2 – If node has lowest score do not separate any more and becomes leaf
3 – If separating becomes an improvement than choose the separation with the lowest impurity score

Classification Trees

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...



Method:
1 – calculate the Gini Impurity scores
2 – If node has lowest score do not separate any more and becomes leaf
3 – If separating becomes an improvement than choose the separation with the lowest impurity score

ISLab
Synthetic Intelligence Lab
Universidade do Minho
Departamento de Informática

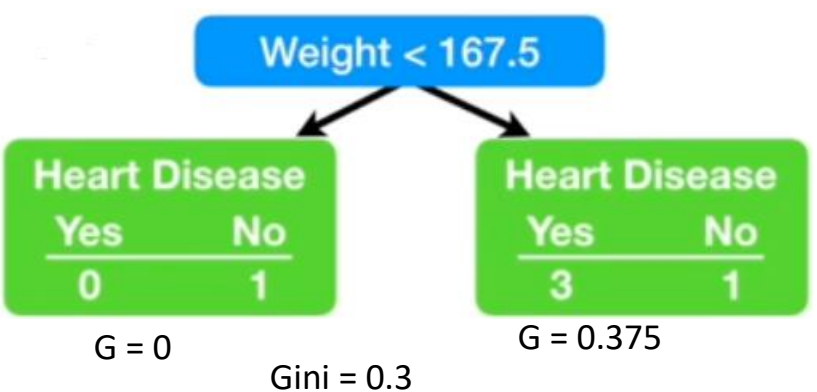
Classification Trees – continuous attributes

	Weight	Heart Disease
Lowest	155	No
	180	Yes
	190	No
	220	Yes
Highest	225	Yes

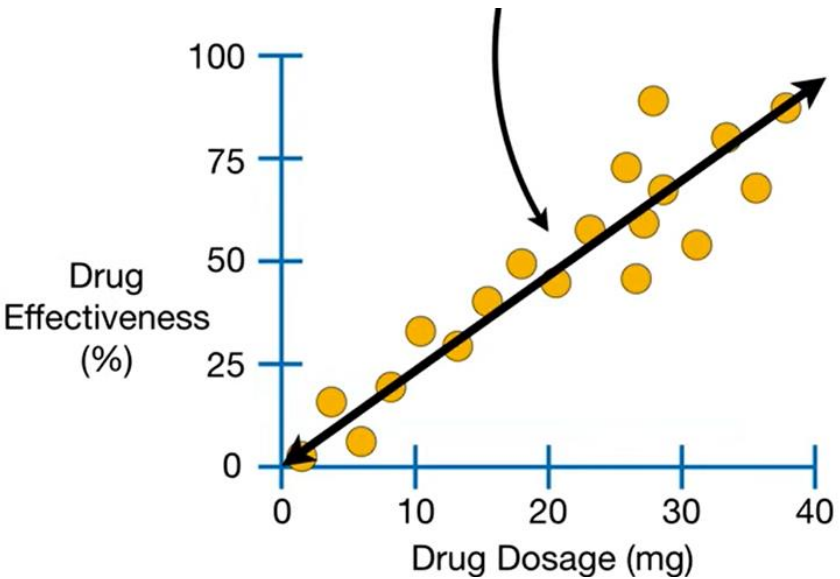
Weight	Heart Disease
155	No
167.5	
180	Yes
185	
190	No
205	
220	Yes
222.5	
225	Yes

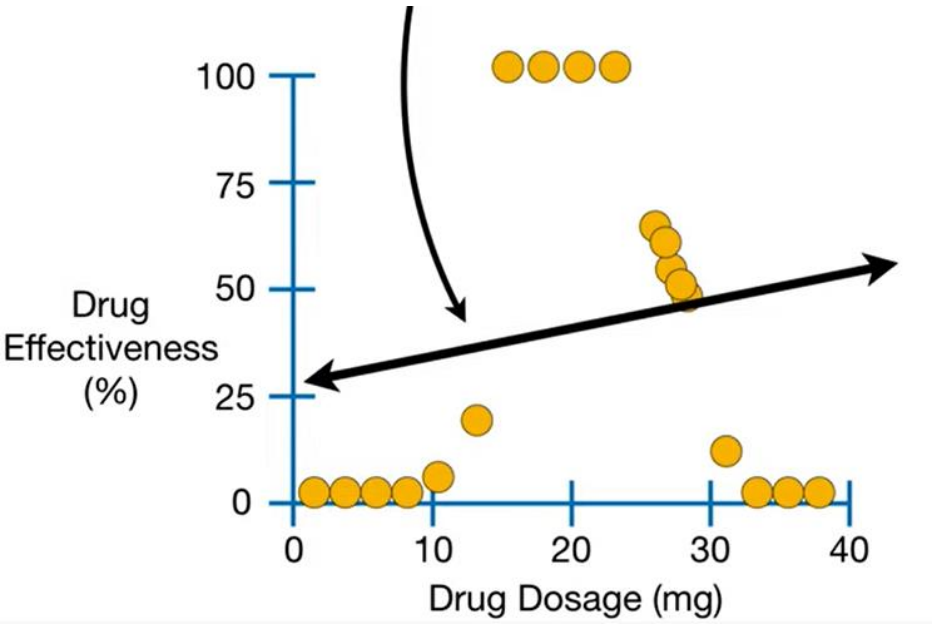
Weight	Heart Disease	
155	No	
167.5		Gini = 0.30
180	Yes	
185		Gini = 0.47
190	No	
205		Gini = 0.27
220	Yes	
222.5		Gini = 0.40
225	Yes	

Method (**continuous attributes**):
1 – sort the attribute lowest to highest
2 – calculate the average value for all adjacent attributes
3 – calculate the impurity values for each average value
4 – if separating becomes an improvement than choose the separation with the lowest impurity score

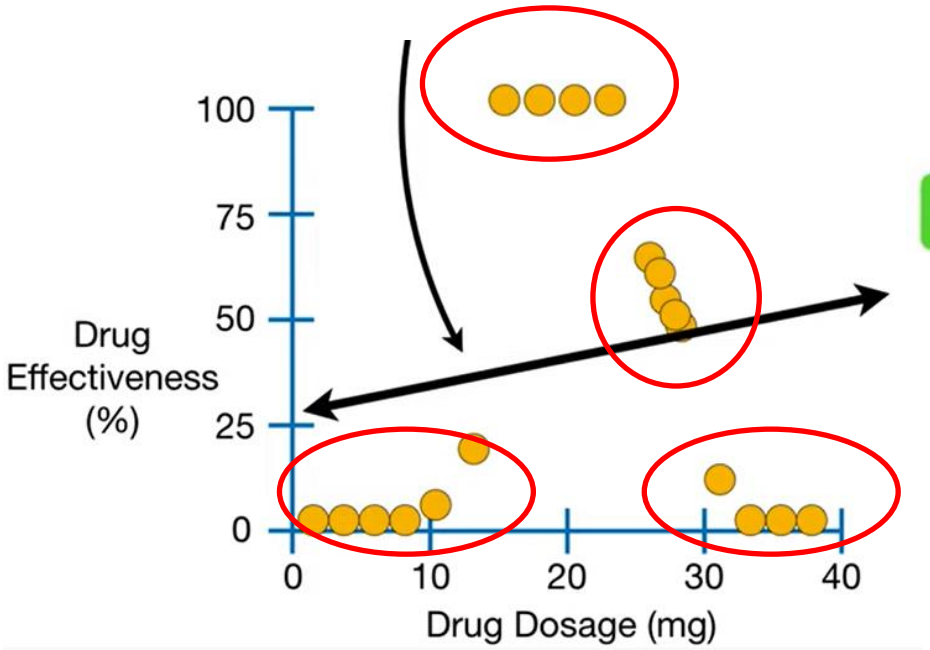


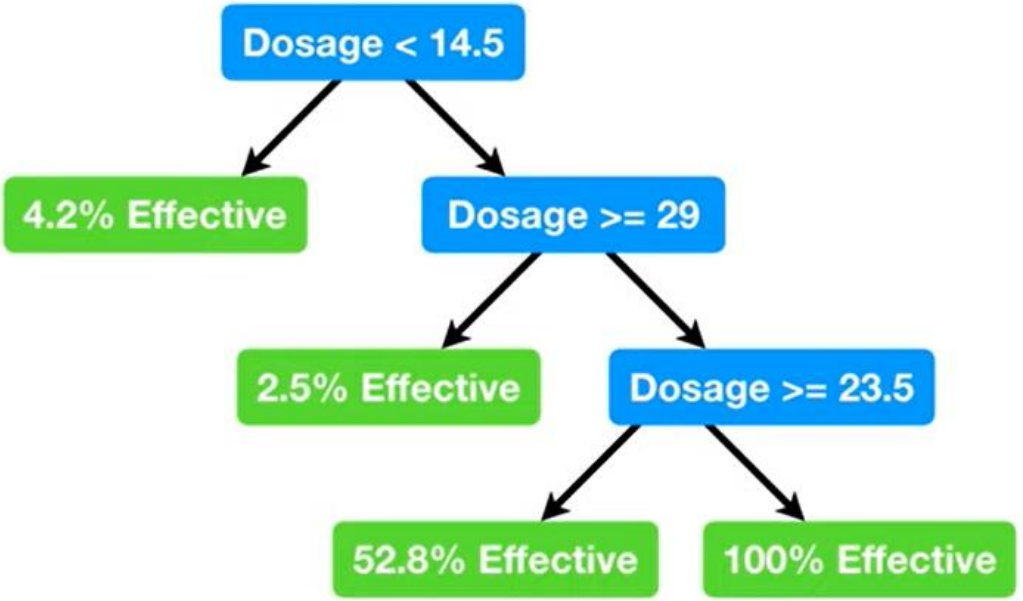
Regression Trees



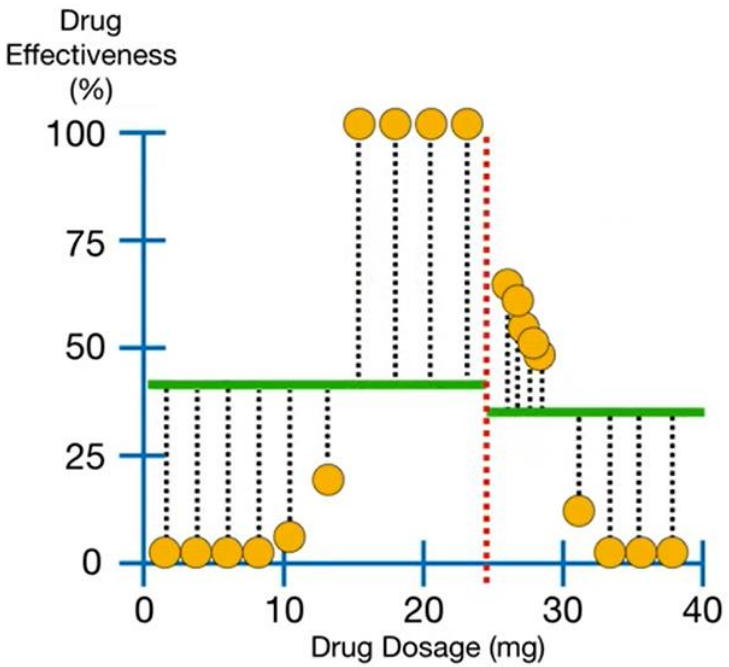


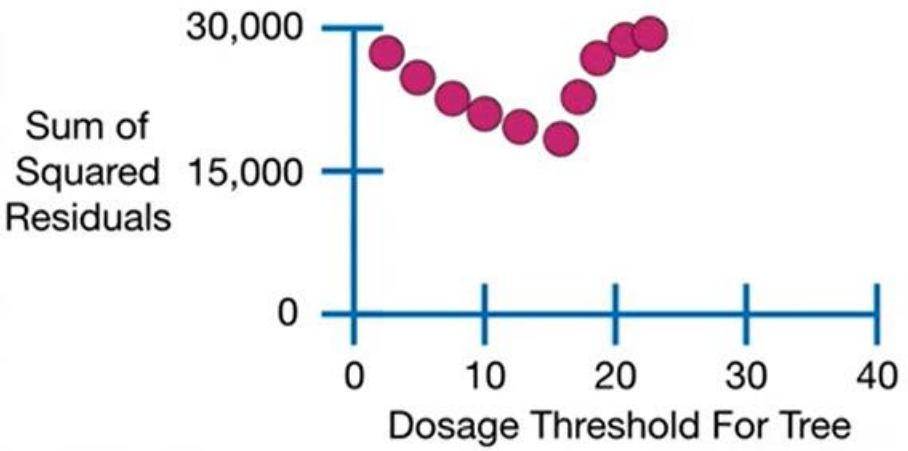
Regression Trees





Regression Trees





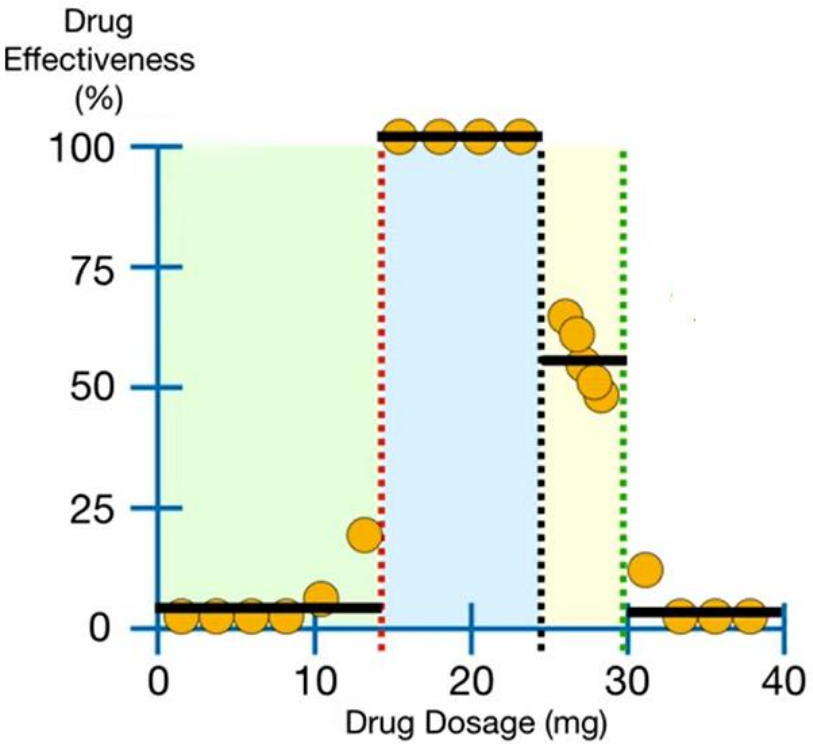
Method

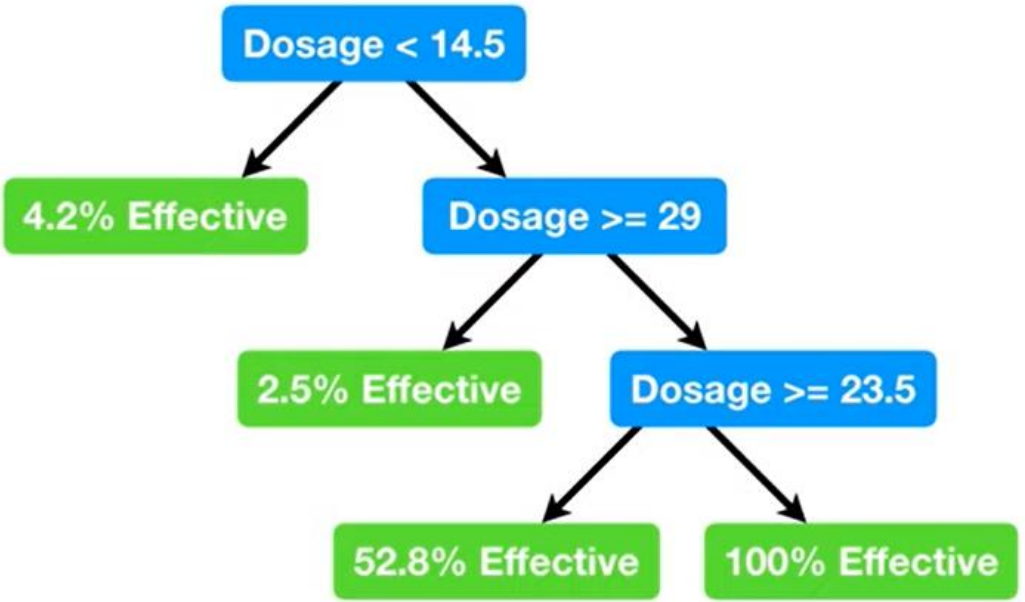
1 - For each possible threshold calculate the average of the left and right samples and calculate the sum of the square error for each sample

2 – select the threshold with the minimum sum of squared errors for a branch

3 – when number of samples less then predefined value (e.g. 20) then it is a leaf with value equal to the average of samples

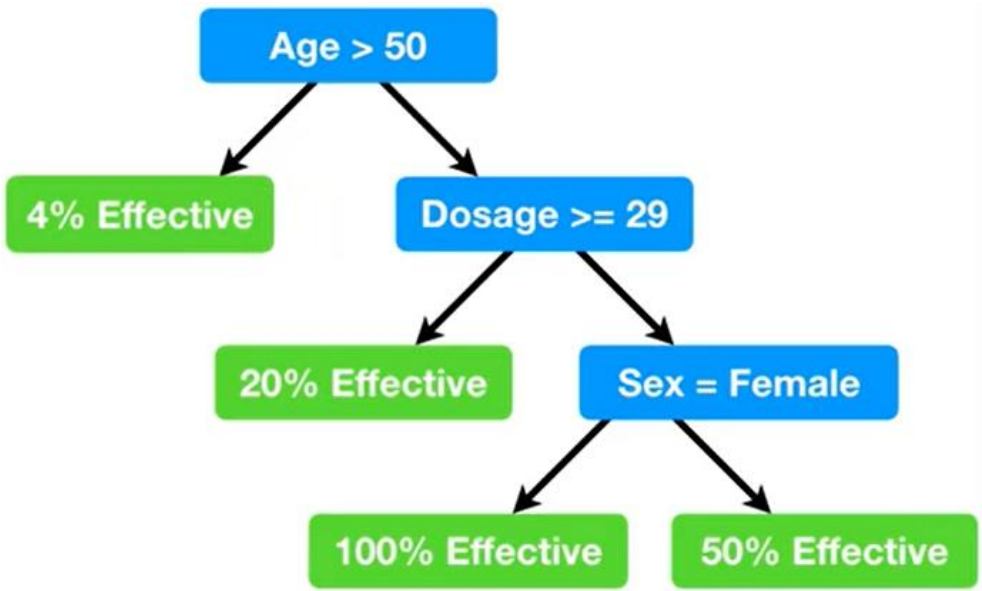
Regression Trees



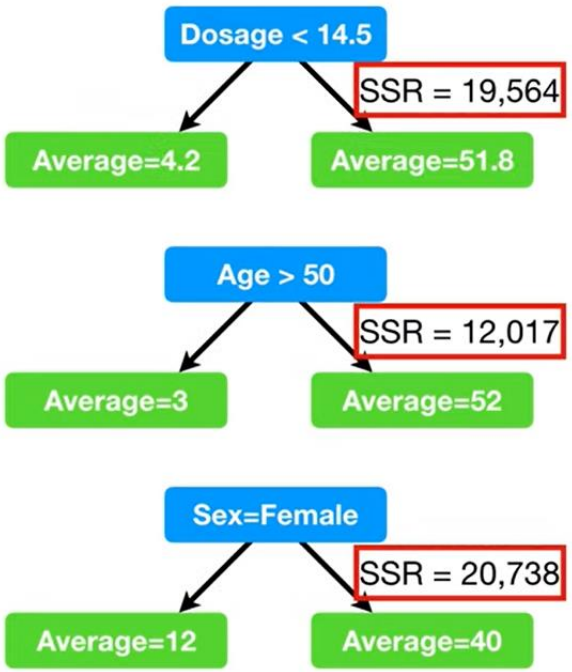


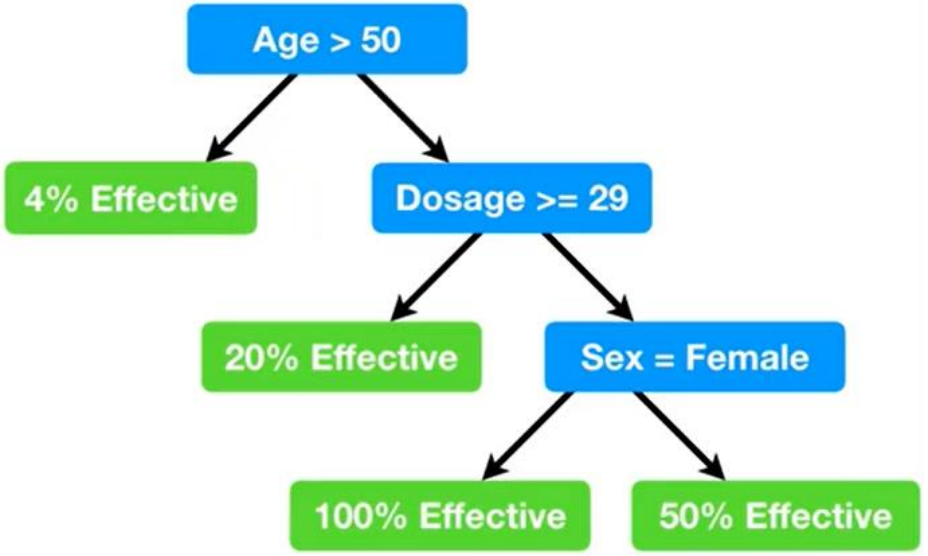
Regression Trees – multiple attributes

Dosage	Age	Sex	Etc.	Drug Effect.
10	25	Female	…	98
20	73	Male	…	0
35	54	Female	…	100
5	12	Male	…	44
etc…	etc…	etc…	etc…	etc…



Regression Trees – multiple attributes





Method

1 – calculate the minimum of the sum of squared errors for each attribute

2 – select the attribute and threshold with the minimum sum of squared errors for a branch

3 – when number of samples less then predefined value (e.g. 20) then it is a leaf with value equal to the average of samples

Decision Trees – prunning

A decision tree will always overfit the training data if we allow it to grow to its max depth

```
full_tree = DecisionTreeClassifier(random_state=2020)
full_tree.fit(X_train, y_train)
```

```
print(full_tree.get_depth())
print(full_tree.get_n_leaves())
```

Pre-prunnig (early stopping):

- **min_sample_split** is the minimum no. of sample required for a split.

```
min_samples_split_grid_search = GridSearchCV(
    estimator=DecisionTreeClassifier(random_state=2020),
    scoring=make_scorer(accuracy_score),
    param_grid=ParameterGrid(
        {"min_samples_split":[[min_samples_split] for min_samples_split in np.arange(EPS, 1, 0.025)]}),
)
```

- **min_sample_leaf** on the other hand is basically the minimum no. of sample required to be a leaf

```
min_samples_leaf_grid_search = GridSearchCV(
    estimator=DecisionTreeClassifier(random_state=2020),
    scoring=make_scorer(accuracy_score),
    param_grid=ParameterGrid(
        {"min_samples_leaf": [ [min_samples_leaf] for min_samples_leaf in np.arange(0.000001, 0.5, 0.025)]}),
)
```

Decision Trees – prunning

A decision tree will always overfit the training data if we allow it to grow to its max depth

Post-prunnig (after perfect training):

- Assign a maximum depth to a tree

```
full_tree = DecisionTreeClassifier(random_state=2020)
full_tree.fit(X_train, y_train)
```

```
print(full_tree.get_depth())
print(full_tree.get_n_leaves())
```

```
max_depth_grid_search = GridSearchCV(
    estimator=DecisionTreeClassifier(random_state=2020),
    scoring=make_scorer(accuracy_score),
    param_grid=ParameterGrid(
        {"max_depth": [[max_depth] for max_depth in range(1, max_depth + 1)]}),
)
```

ISLab
Synthetic Intelligence Lab
Universidade do Minho
Departamento de Informática

Decision Trees – prunning

A decision tree will always overfit the training data if we allow it to grow to its max depth

Post-prunnig (after perfect training):

- Assign a maximum depth to a tree

- Pruning starts with an unpruned tree, takes a sequence of subtrees (pruned trees), and picks the best one through cross-validation.
- **Cost complexity pruning** generates a series of trees where cost complexity measure for sub-tree T_t is:

$$R_{\alpha}(T_t) = R(T_t) + \alpha |T_t| \quad \text{or} \quad \text{TreeScore}_t = SSR + \alpha |T_t|$$

where: R((T) - Total training error of leaf nodes

|T| — The number of leaf nodes

α — complexity parameter

cost_complexity_pruning_path returns the effective alphas and the corresponding total leaf impurities at each step of the pruning process.

```
ccp_alphas = full_tree.cost_complexity_pruning_path(X_train, y_train)["ccp_alphas"]
ccp_alpha_grid_search = GridSearchCV(
    estimator=DecisionTreeClassifier(random_state=42),
    scoring=make_scorer(accuracy_score),
    param_grid=ParameterGrid({"ccp_alpha": [[alpha] for alpha in ccp_alphas]}),
)
```

https://online.stat.psu.edu/stat508/lesson/11/11.8/11.8.2

Algorithms

Algorithm ID3: Iterative Dichotomiser 3
https://iq.opengenus.org/id3-algorithm/

Developed by Ross Quinlan;
Builds the decision tree from root to leafs;
selects the best attribute using maximum Information Gain (IG) or minimum Entropy (H)

Algorithm C4.5
https://towardsdatascience.com/what-is-the-c4-5-algorithm-and-how-does-it-work-2b971a9e7db0

Improvements to algoritmo ID3, Also developed by Ross Quinlan;
Manipulates continuas and discrete attributes;
Handles missing values; very well;
Supports weighted attributes ;
Allows for pruning the tree (goes back 1 iteration in the tree and removes branches that contribute less or not at all to the solution by replacing them with leaves)

Algorithm J48
https://www.cs.waikato.ac.nz/ml/weka)

Open source Implementation of the C4.5 algorithm in JAVA within the WEKA framework;
WEKA: Waikato Environment for Knowledge Analysis;

Algorithm CART:

Classification and Regression Tree
Introduced by Breiman, practically in parallel with Ross Quinlan's ID3;
A same algorithm that shares the similarities of classification and regression models;

Algorithm CHAID: Chi-square Automatic Interaction Detection

It operates data separation in multi-level mode, while CART uses binary modes for this division;
Suitable for large datasets;
Often used in marketing studies for market segmentation;

Conclusions

Strengths:

- **Simple configuration** (doesn't have too many configuration parameters);
- Compared to other algorithms decision trees requires **less effort for data preparation** during pre-processing.
- A decision tree does **not require normalization** of data.
- A decision tree does **not require scaling** of data as well.
- **Missing values in the data also do NOT affect** the process of building a decision tree to any considerable extent.
- A Decision tree model is **very intuitive and easy to explain** to technical teams as well as stakeholders.

Weaknesses:

- **Inadequate** for problems characterized by **many interactions between attributes**;
- Does not avoid **replicas** of subtrees;
- A **small change in the data** can cause a **large change in the structure** of the decision tree causing instability.
- For a Decision tree sometimes calculation can go far **more complex** compared to other algorithms.
- Decision tree often involves **higher time to train** the model.

References

- *The StatQuest Illustrated Guide to Machine Learning (PDF)*. (n.d.). https://statquest.gumroad.com/l/wvtmc
- Ross Quinlan (1986), “Induction of decision trees”, Machine Learning, 1(1):81-106 (http://hunch.net/~coms-4771/quinlan.pdf)
- Ross Quinlan (1993), “C4.5 Programs for Machine Learning”, Morgan Kaufmann
- Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984), “Classification and regression trees”, Monterey, CA
- Pang Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar (2018)“Introduction to Data Mining”, ISBN 9780133128901