Advanced Institute for Artificial Intelligence

Supervised Classification Methods

Advanced Institute for Artificial Intelligence – AI2

https://advancedinstitute.ai

Decision Trees

Jupyter Notebook: https://bit.ly/3nIlpNE

Credits

- Examples and images taken from:
  - https://www.oreilly.com/library/view/hands-on-machine-learning/9781491962282/
  - https://github.com/ageron/handson-ml
  - https://towardsai.net/p/programming/decision-trees-explained-with-a-practical-example-fe47872d3b53
  - https://geam.paginas.ufsc.br/files/2020/02/decision-tree-ensemble.pdf

Decision Trees

- Versatile Machine Learning algorithms that can perform both classification and regression tasks
- Powerful algorithms, capable of fitting complex datasets
- Fundamental components of Random Forests
  - One of the most powerful Machine Learning algorithms
- **High Explicability Power!**

A quick digression - a word about *explicability*

- White Box approach:
  - Easy to interpret
  - Provide nice and simple classification rules that can even be applied manually if need be (e.g., for flower classification).
- Black Box approach
  - Great predictions
  - “Easily” check the calculations
  - It is usually hard to explain in simple terms why the predictions were made

Example

If a neural network says that a particular person appears on a picture, it is hard to know what contributed to this prediction: the **person’s eyes**? **Her mouth**? Or even **the couch that she was sitting on**?

Laws and Regulations

- EU General Data Protection Regulation (GDPR)
  - "The ethical principles include autonomy, prevention of harm, fairness and explicability"

GDPR

"The data subject should have the **right not to be subject to a decision, which may include a measure**, evaluating personal aspects relating to him or her **which is based solely on automated processing and which produces legal effects** concerning him or her or similarly significantly affects him or her, **such as automatic refusal of an online credit application or e-recruiting practices without any human intervention**."

Laws and Regulations

- USA Equal Credit Opportunity Act
  - "Creditors are required to notify applicants of action taken in certain circumstances, and such notifications must provide specific reasons"

Equal Credit Opportunity Act

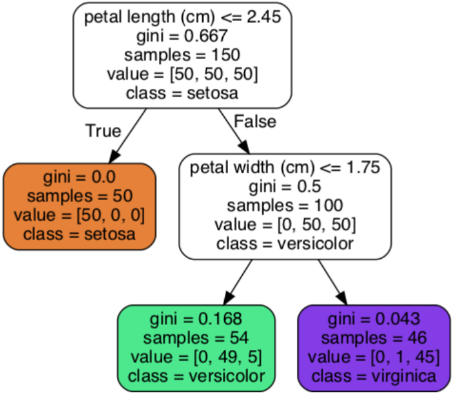
"(2) Statement of specific reasons. The statement of reasons for adverse action required by paragraph (a)(2)(i) of this section must be **specific and indicate the principal reason(s) for the adverse action**. Statements that the adverse action was based on the creditor's **internal standards or policies or that the applicant**, joint applicant, or similar party **failed to achieve a qualifying score** on the creditor's credit scoring system are insufficient."

Building our first Decision Tree

```python
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris()
X = iris.data[:, 2:] # petal length and width
y = iris.target
tree_clf = DecisionTreeClassifier(max_depth=2)
tree_clf.fit(X, y)
```

The Trained Model:



Decision Trees

Making Predictions

- Start at the root node (depth 0, at the top)
- Asks whether the flower's petal length is smaller than 2.45 cm
- If it is, then you move down to the root's left child node (depth 1, left)
  - In this case, it is a leaf node (i.e., it does not have any children nodes)
  - The predicted class for that node and the Decision Tree predicts that your flower is an Iris-Setosa (`class=setosa`).
- The petal length is greater than 2.45 cm
  - move down to the root's right child node (depth 1, right)
  - Ask another question: is the petal width smaller than 1.75 cm?
    - If yes, then your flower is most likely an Iris-Versicolor (depth 2, left)

Decision Trees

- A node's samples attribute counts how many training instances it applies to
  - For example, 100 training instances have a petal length greater than 2.45 cm (depth 1, right)
- The value attribute: how many training instances of each class this node applies to:
  - For example, the bottom-right node applies to 0 Iris-Setosa, 1 IrisVersicolor, and 45 Iris-Virginica
- A node's gini attribute measures its impurity:
  - A node is "pure" (gini=0) if all training instances it applies to belong to the same class
  - For example, since the depth-1 left node applies only to Iris-Setosa training instances, it is pure and its gini score is 0.

Decision Trees

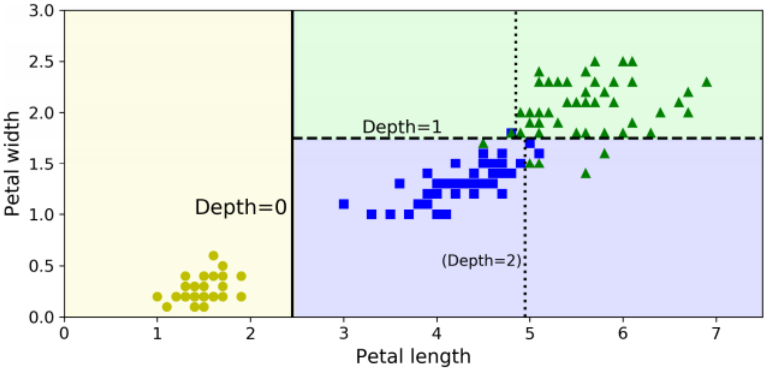
- The Gini Impurity Measure

$$\boxed{G_i = 1 - \sum_{k=1}^{n} (P_{i,k})^2}$$

where $P_{i,k}$ is the ratio of class k instances among the training instances in the i-th node

- For Example, the depth-2 right leaf:
  - $G = 1 - {\frac{0}{46}}^{2} - {\frac{1}{46}}^{2} - {\frac{45}{46}}^{2} \approx 0.043$

Decision Boundaries



Decision Boundaries

- The thick vertical line represents the decision boundary of the root node (depth 0): petal length $= 2.45$ cm
- Since the left area is pure (only Iris-Setosa), it cannot be split any further
- The right area is still impure, so the depth-1 right node splits it at petal width $= 1.75$ cm
- Since max_depth was set to 2, the Decision Tree stops
- If you set max_depth to 3, then the two depth-2 nodes would each add another decision boundary (represented by the dotted lines).

Estimating Class Probabilities

- Traverse the tree to find the leaf node for this instance, and then return the ratio of training instances of class k in this node

Example

Suppose you have found a flower whose petals are **5cm long** and **1.5cm wide**.

The corresponding leaf node is the **depth-2 left node**, so the Decision Tree should output the following probabilities: **0%** for Setosa (0/54), **90.7%** for Versicolor (49/54), and **9.3%** for Virginica (5/54)

Classification And Regression Tree (CART) algorithm

- Repeat until reach the given depth or convergence
  - Searches for the pair (k, tk) that produces the purest subsets (weighted by their size).
  - Splits the training set in two subsets using a single feature k and a threshold tk

The optimization function:

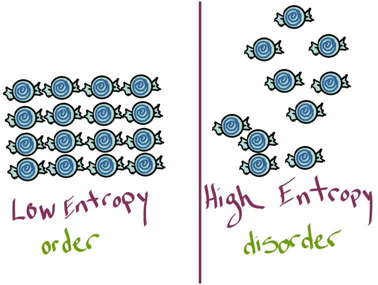
$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

where $G_{left/right}$ measures the impurity of the left/right subset, and

$m_{left/right}$ is the number of instances in the left/right subset;

How about using Entropy instead of Gini?

- The concept originated in thermodynamics as a measure of molecular disorder:
  - Entropy approaches zero when molecules are still and well ordered
  - Shannon's information theory, where it measures the average information content of a message:
    - Entropy is zero when all messages are identical



Decision Trees

The Entropy Function

$$H_i = -\sum_{k=1}^{n} P_{i,k} log_2(P_{i,k}) \ with \ P_{i,k} \neq 0$$

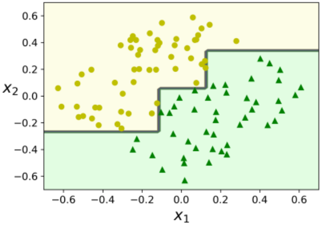
- Again, $P_{i,k}$ is the ratio of class k instances among the training instances in the i-th node, i.e.,

$$P_{i,k} = \frac{Number \ of \ instances \ of \ the \ k \ class \ at \ the \ i-th \ node}{Number \ of \ examples \ at \ the \ i-th \ node}$$

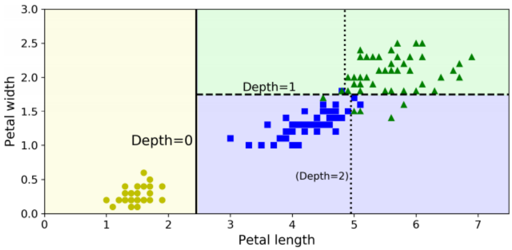
For the depth-2 right leaf: $-\frac{0}{46} log_2(\frac{0}{46}) - \frac{1}{46} log_2(\frac{1}{46}) - \frac{45}{46} log_2(\frac{45}{46}) \approx 0.152$

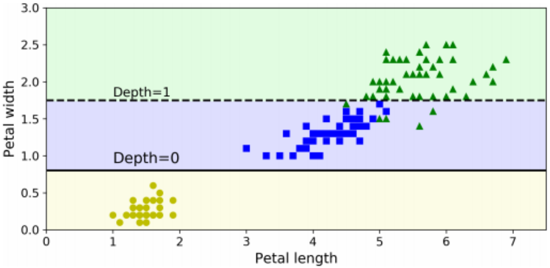
Decision Trees

- Advantages:
  - simple to understand and interpret, easy to use, versatile, and powerful
- Disadvantages:
  - orthogonal decision boundaries (all splits are perpendicular to an axis)



- Disadvantages:
  - very sensitive to small variations in the training data
    - For example, if you just remove the widest Iris Versicolor (the one with petals 4.8 cm long and 1.8 cm wide) and train a new Decision Tree, you may get a model that it looks very different from the previous one.





Random Forests

Jupyter Notebook: https://bit.ly/3nIlpNE

Ensemble

- **Wisdom of the crowd**
  - aggregated answer is better than an expert's answer
- Aggregating the predictions such as classifiers, you will often get better predictions than with the best individual predictor.
- A group of predictors is called an ensemble
  - **Ensemble Learning**
- Example:
  - Train a group of Decision Tree classifiers, each on a subset of the training set
  - Obtain the predictions of all individual trees, then predict the class that gets the most votes

Bagging

- Train a set of classifiers on different random subsets of the training set
  - Use a replacement strategy for sampling, thus allowing duplicates in each classifier training set
- Aggregate the predictions of all predictors:
  - typically the statistical mode:the most frequent prediction
- Predictors can all be trained in parallel, via different CPU cores or even different servers.
- Predictions can also be made in parallel

Random Forests

- Ensemble of Decision Trees, generally trained via the bagging method
- Search for the best feature among a random subset of features instead of finding the best overall
  - Greater tree diversity

Building our first Random Forest

```python
from sklearn.ensemble import RandomForestClassifier

rnd_clf = RandomForestClassifier(n_estimators=500,
                                 max_leaf_nodes=16,
                                 n_jobs=-1)

rnd_clf.fit(X_train, y_train)

y_pred_rf = rnd_clf.predict(X_test)
```