Decision Trees

Decision trees are conceptually built from nodes and branches. Where the nodes are the decision point, and branches are the points of data that are separated from the node's decision. When branches that stem from nodes emerge, they carry with them all the data points that were separated from the node. This iterative process will continue till some criteria are met, either running out of data to split, or the information gained from splitting data is not worth the cost of splitting.

Decision tree algorithms are robust, meaning they can be used for regression or classification problems. They are also easy to visualize and usually are represented in an inverted tree.

Random Forests

Random forests are an extension of decision trees. What makes them so unique is that they are made up of many tree-based models. Each tree is unique, created by taking a random sample of data to train on, and using a random set of available features when splitting data at a node. All the models create a prediction, where it is summarized by another model that manages all the trees. These methods are usually called **ensemble models**.

Code Example of Random Forest

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier

df = pd.DataFrame(
    [[1, 2, 0], [3, 4, 1], [5, 6, 0], [7, 8, 1]],
    columns=["num", "amount", "target"]
)

clf = RandomForestClassifier().fit(df[["num", "amount"]], df["target"])

clf.score(df[["num", "amount"]], df["target"])
# 1.0

clf.predict([[1, 2]])
# array([0])
```

Hierarchical Clustering

Hierarchical clustering, while a tree-based model, is not thought about in the context of supervised models. Instead, it builds a hierarchy of clusters based on the distances of features

when compared to each other. This is visualized in the model by clustering data together while branching off dissimilar data. To calculate these distances, you'll use more standardized approaches like Euclidian and Manhattan distances.

Feature Selection

When tree-based models create nodes, they select the best features first to split data. In doing so, they inherently create feature importance rankings by creating the model itself. The higher the node in a tree-based model, the more important that feature is. This is useful not only to learn which models are most important but also which features can be dropped from the dataset since they provide little value to model performance. Smaller data also means faster training and a smaller overall dataset size.

Additional Resources

- If you want to learn more about decision trees, we recommend reading through the scikit-learn documentation on **decision trees**.
- Documentation of random forests from scikit-learn is highly informative, which you can find at **Random Forests**.
- The Wikipedia page for **hierarchical clustering** has a great overview of the methodology.
- Feature selection in Scikit-learn can be found here: **feature selection**.