Applying Decision Trees: Takeaways

Syntax

- Instantiate the scikit-learn decision tree classifier:

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=1)
```

- Fit data using the decision tree classifier:

```
columns = ["age", "workclass", "education_num", "marital_status"]
clf.fit(income[columns], income["high_income"]
```

- Shuffle the rows of a data frame and re-indexing:

```
income = income.reindex(numpy.random.permutation(income.index))
```

- Calculate the area under curve (AUC) using scikit-learn:

```
from sklearn.metrics import roc_auc_score
error = roc_auc_score(test["high_income"], clf.predict(test[columns])
```

Concepts

- Scikit-learn includes the `DecisionTreeClassifier` class for classification problems, and `DecisionTreeRegressor` for regression problems.
- AUC (area under the curve) ranges from 0 to 1 and is a measure of how accurate our predictions are, which makes it ideal for binary classification. The higher the AUC, the more accurate our predictions. AUC takes in two parameters:
    - `y_true` : true labels.
    - `y_score` : predicted labels.
- Trees overfit when they have too much depth and make overly complex rules that match the training data but aren't able to generalize well to new data. The deeper the tree is, the worse it typically performs on new data.
- Three ways to combat overfitting:
    - "Prune" the tree after we build it to remove unnecessary leaves.
    - Use ensampling to blend the predictions of many trees.
    - Restrict the depth of the tree while we're building it.
- We can restrict tree depth by adding a few parameters when we initialize the `DecisionTreeClassifier` :
    - `max_depth` : Globally restricts how deep the tree can go.
    - `min_samples_split` : The minimum number of rows a node should have before it can be split; if this is set to 2 then nodes with two rows won't be split and will become leaves instead.
    - `min_samples_leaf` : The minimum number of rows a leaf must have.
    - `min_weight_fraction_leaf` : The fraction of input rows a leaf must have.

        - `max_leaf_nodes` : The maximum number of total leaves; this will limit the number of leaf nodes as the tree is being built.
        - However, some parameters aren't compatible. For example, we can't use `max_depth` and `max_leaf_nodes` together.
- Underfitting occurs when our model is too simple to explain the relationships between the variables.
- High bias can cause underfitting while high variance can cause overfitting. We call this bias-variance tradeoff because decreasing one characteristic will usually increase the other. This is a limitation of all machine learning algorithms.
- The main advantages of using decision trees is that decision trees are:
    - Easy to interpret.
    - Relatively fast to fit and make predictions.
    - Able to handle multiple types of data.
    - Able to pick up nonlinearities in data and fairly accurate.
- The most powerful way to reduce decision tree overfitting is to create ensembles of trees. The random forest algorithm is a popular choice for doing this. In cases where prediction accuracy is the most important consideration, random forests usually perform better.

Resources

- DecisionTreeClassifier class documentation
- Area under the curve
- Documentation for roc_auc_score