Portfolio Question 3.1.P - Minimising Losses for Decision Trees

Daniel Lawson University of Bristol

Portfolio 3.1.P (v4.0.0)

Portfolio Question 3.1.P - Minimising Losses for Decision Trees

- How do we stop splitting a decision tree?
  - Compare information criteria and cross validation.
  - Can you find examples of when the choice matters?

Decision Tree pruning (Information Criteria)

- Decision trees **overfit** to the data.
- Penalisation is often used:
  - Minimise $\mathcal{L}' = \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) + \alpha|T|$
  - Where $|T|$ is the number of bipartitions in the tree, and $\mathcal{L}$ is a log-loss.
  - All the usual caveats of Information Criteria apply.
  - Tree search is usually performed by a greedy brute force approach:
    - Evaluate $\mathcal{L}'$ for every branch in the tree
    - Choose the sub-tree with the lowest value
    - Repeat until no cuts improve the loss
  - Alternative search approaches exist for large trees

Decision Tree pruning (Cross Validation)

- To avoid the problems with Information Criteria, Cross-validation can be used
- Choose the **sub-tree** that has the best **out of sample** predictive power.
  - With a single left-out dataset (risks overfitting),
  - Or random sets (higher variance)
- Now we have to re-compute the entire model each pruning
- Search is a computational concern

Decision tree notes

- In practice, **bagging** (bootstrapping the data) is important, to prevent overfitting and for smoothing the output
- The choice of feature space directly affects the decisions that are examined. So **PCA** or similar could usefully be applied to obtain "orthogonal" feature space, **reducing depth**.
- There are parameters, e.g. depth/stopping criterion/split rule, which could be chosen by cross-validation.

Discussion

- Does the theory we've seen mean that penalisation and CV are related?
- Which penalisation or information criterion should we use? Why?
- Example where this creates different models?
- When do they make meaningfully different predictions?

Apendix: Examples and code

Fitting a Decision Tree in Python

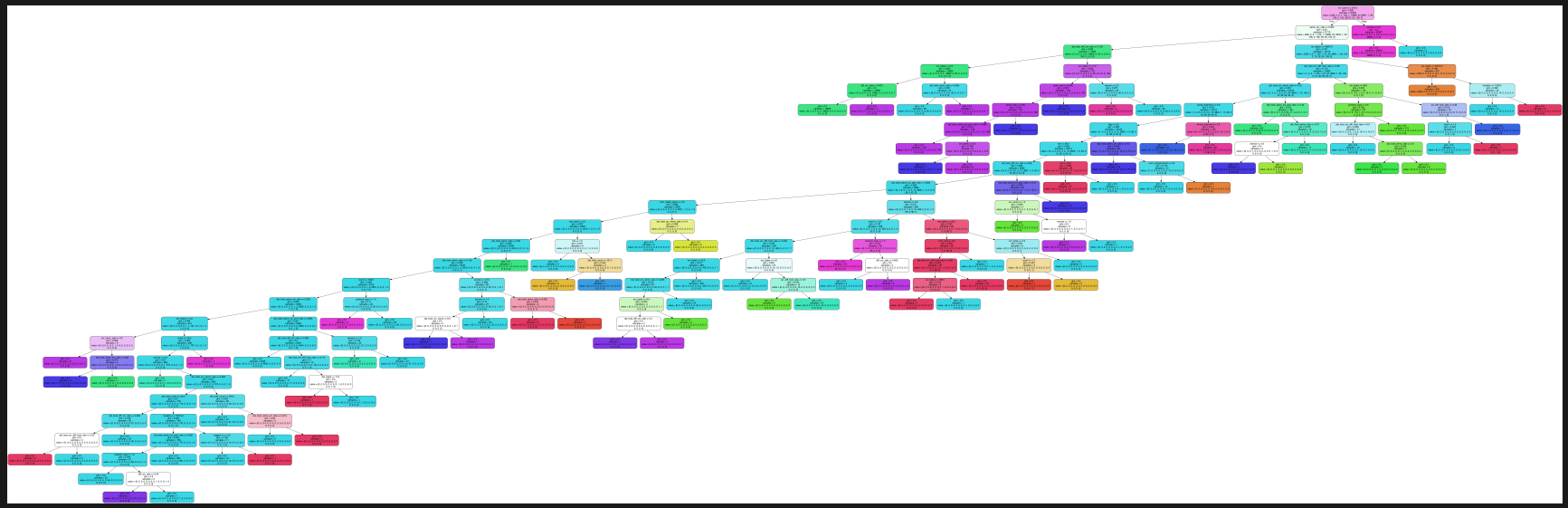
```
from sklearn import tree
cldt = tree.DecisionTreeClassifier()

trained_model_d= cldt.fit(X_train, y_train)
y_pred_d = cldt.predict(X_test)
error_d = zero_one_loss(y_test, y_pred_d)
```

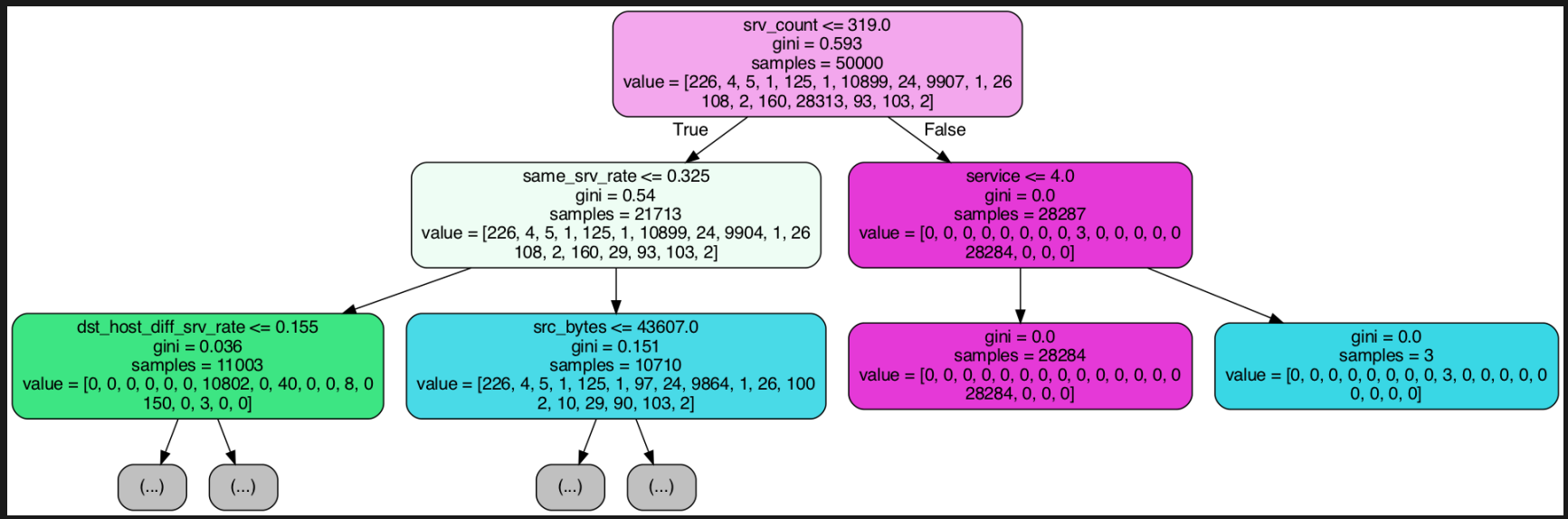
Plot a Decision Tree in Python

```
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(cldt, out_file=dot_data,max_depth=2,
                feature_names=X_train.columns.values,
                filled=True, rounded=True)
graph = pydotplus.graph_from_dot_data(
    dot_data.getvalue())
Image(graph.create_png())
```

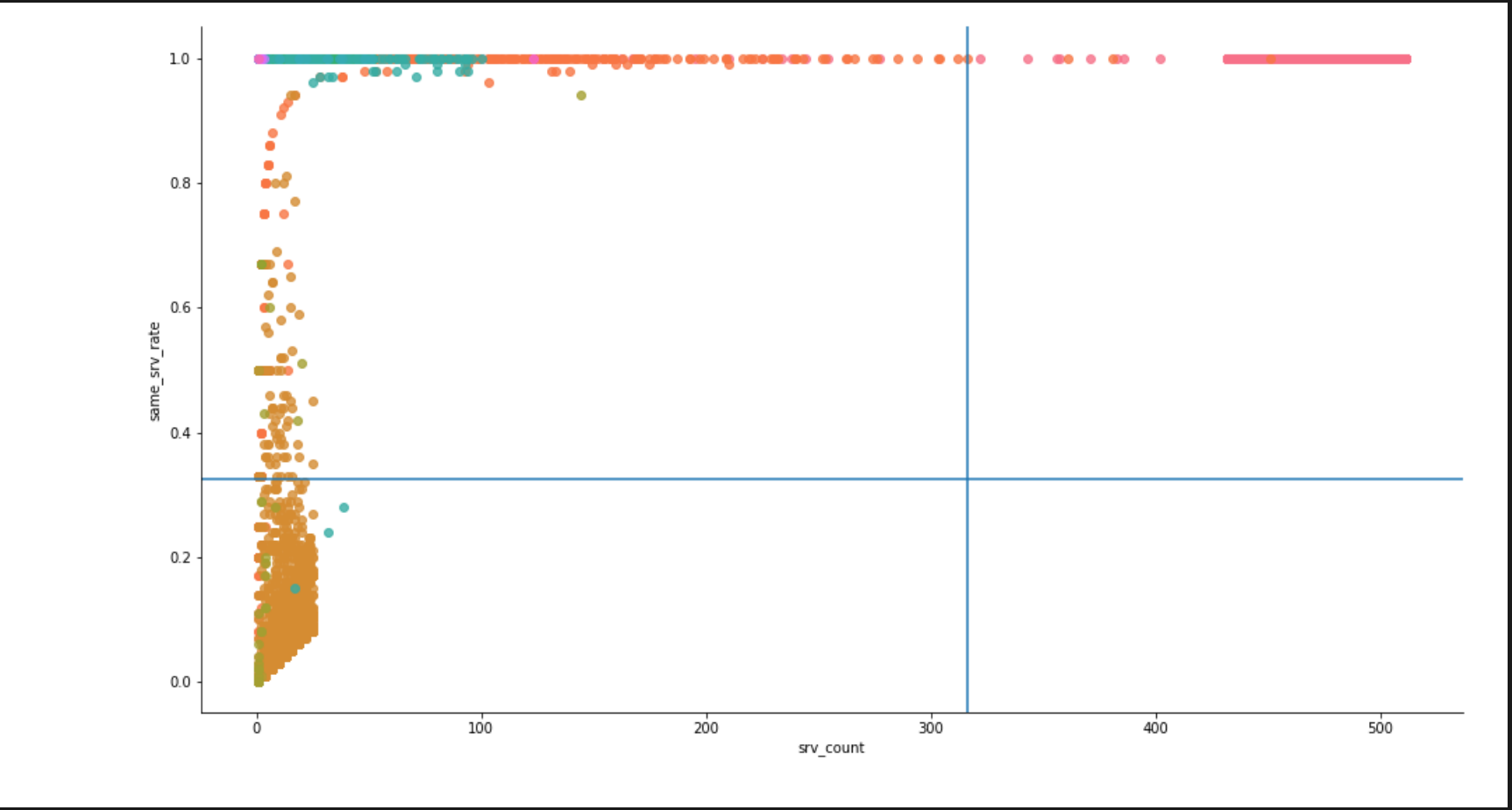
Decision Tree (whole)



Decision Tree (root)



Decision Tree in feature space (1)



Decision Tree in feature space (2)