61011 Lab - Week 4

David Wong and Christopher Yau

October 2020

1 Introduction

The targets for this week are listed below. At a minimum, you should aim to complete Level 1. The Level 2 and 3 tasks are significantly trickier than in previous weeks and are designed to stretch those with slightly more experience. The purpose of this lab is to examine the performance of some popular decision tree methods, and to appreciate the importance of the different hyperparameters.

During (or before) the live Wednesday lab sessions, you may request help by entering your name on the following spreadsheet:
https://docs.google.com/spreadsheets/d/1Ux0iZjqvF1MPciMzm1w_o5DviDSpTyg1PkiPKdtRBfA/edit?usp=sharing.

When you join the weekly blackboard collaborate session, a teaching assistant will move you to a break out room to discuss your issue.

Level 1

On a data set of your choice:

- Build an ID3 decision tree with a maximum depth of 10, and a minimum number of examples of 5. Remember to do a train/test split. Repeat this with a minimum number of examples of 500. What is the divergence in test error? Plot a graph of the training error and testing error, as you change this minimum number.
- Repeat the previous task, using gini impurity as the loss function for deciding how to split a tree. Are there many differences in the final tree? What about computation time?
- Construct an ROC curve for a logistic regression classifier using the inbuilt functions in sklearn.

Level 2

- Implement an ROC curve for a logistic regression classifier from scratch. You can use sklearn to generate and learn the model, but should write the rest yourself.
- using the result from the previous task, write a function to calculate the AUROC. You will need to consider how to estimate an area under a curve using the triangle/ trapezium rule.
- Compare how well the decision tree does against other models we've built so far. How sensitive is it to the parameter settings, compared to the other models? How does it react to datasets with large numbers of examples versus small numbers? What about data with large numbers of features versus small numbers?
- Implement your own decision tree. You can use SKlearn to calculate the mutual information for a feature x and the labels. The rest of the recursive algorithm you'll have to implement yourself.

Level 3

Write a test framework where you can supply a list of models and each model will be evaluated via a cross fold validation (also possibly part of the supplied arguments to the test harness), across all those datasets. Be sure to take account of any sources of variance (e.g. random initialisation of the logistic regression).