1 Individual Assignment

Instructions: *This exercise should be done "by hand", that is, not using Python. All necessary calculations should be included in the submission, as well as brief explanations of what you do.*

The training data set in Table 1 on the next page provides a summary of the traffic conditions experienced on a major road across several days, times and weather conditions.

1. Create a full classification tree that estimates the traffic based on the day of the week, time of the day and the weather condition. Use the entropy as a purity measure to guide your splitting choices.

2. Construct a confusion matrix for the performance of your tree on the training data. What is the misclassification rate, and what is the sensitivity and specificity (assuming that 'yes' is the 'important' class)?

3. Construct the confusion matrix that results if your algorithm is applied to the test set in Table 2 on page 3.

day	**weather**	**time**	**traffic**
weekday	sunny	1pm	no
weekday	rainy	1pm	yes
weekday	sunny	8am	no
weekday	sunny	1pm	no
weekday	rainy	1pm	yes
weekday	sunny	8am	no
weekend	sunny	8am	yes
weekend	sunny	1pm	yes
weekday	sunny	8am	no
weekday	sunny	1pm	no
weekday	sunny	1pm	no
weekend	rainy	1pm	yes
weekday	rainy	1pm	yes
weekday	sunny	8am	no
weekday	sunny	1pm	no
weekend	sunny	1pm	yes
weekday	rainy	8am	yes
weekday	sunny	8am	no
weekday	sunny	8am	no
weekday	sunny	1pm	no
weekday	sunny	8am	yes
weekend	rainy	8am	no
weekday	sunny	1pm	no
weekday	rainy	8am	yes
weekday	sunny	8am	yes

Table 1: Training set for the individual assignment.

day	**weather**	**time**	**traffic**
weekend	rainy	8am	no
weekday	sunny	8am	yes
weekend	sunny	1pm	yes
weekday	sunny	8am	no
weekend	sunny	1pm	yes
weekday	rainy	8am	no
weekday	sunny	8am	yes
weekday	sunny	1pm	no
weekday	sunny	1pm	no
weekday	sunny	1pm	no
weekend	rainy	8am	yes
weekday	sunny	8am	yes
weekday	sunny	1pm	no
weekday	rainy	1pm	yes
weekday	sunny	1pm	no

Table 2: Test set for the individual assignment.

2 Individual Assignment

In this exercise, we try to predict defaults in student loan applications. To this end:

1. Load the data set `loandata.csv` into Python.
2. The data set contains some categorical predictors. Sklearn, which you should use for this exercise, can only handle numerical predictors. Translate the categorical predictors into numerical predictors. (You may want to look into the pandas function `get_dummies`.)
3. Shuffle the data set and split it into 50% training data, 25% validation data and 25% test data.
4. Calculate the accuracy of the naïve benchmark (majority predictor) on the validation set.
5. Train a decision tree using the default settings, and calculate the accuracy of this tree on the training and the validation set. What do you think of this classifier?
6. Retry the previous step using different maximum depths for the tree. (Look at the `max_depth` parameter.) Plot the accuracy on the training data as well as on the validation data as a function of the tree depth.
7. Choose the most appropriate tree depth and justify your choice. What do you think of this classifier?
8. Retrain the best classifier using all the samples from both the training and the validation set. Estimate the out-of-sample accuracy of this classifier on the test set. Discuss what you observe.
9. Retrain the best classifier on *all* samples (including the test set) and describe the tree that you obtain (*e.g.* via visualisation, or via textual description).