CMSC726 Programming Assignment 1: Decision Trees, K-NN and inductive bias

In this programming assignment you will implement the Decision Tree method and experiment with it and K-nearest neighbors. You will use k-nn code and the spam dataset from the knn practical session of September 7. Questions are shown in this color. Please address everything marked TODO below.

Start early, and ask questions often.

Part I: The decision tree algorithm

Implement the decision tree algorithm as presented in CIML, including treedepth as a hyper-parameter. Recall that features in the spam dataset are continuous, so you need to figure out how to deal with those. Please discuss below how you did that. Also, Insert the code here, e.g.:

TODO: Discuss how you dealt with continues features

TODO: add source code

```python
class DecisionTree:
    def __init__(self, traindat, depth):
        # etc...
```

Extend your implementation to use information gain as scoring function. Insert your code for entropy gain here:

TODO: add code for information gain here

Apply your code to the spam dataset. Analyze the result. Things you should discuss: 1. The effect of tree depth on accuracy, 2. Which features seem to be important for classification.

In [ ]: # TODO: insert analysis here

Part II: Extending k-nn

Implement the weighted distance idea discussed in CIML for k-nearest neighbors. Feel free to extend the code linked to above. Please provide your modified code here:

TODO: insert weighted distance k-nn code here

Compare your new weighted k-nn algorithm to the original algorithm provided using the spam dataset.

In [ ]: # TODO: insert analysis here

Part III: Compare k-nn and decision tree performance on the spam dataset

Discuss performance and any other suitable characteristics of k-nn and decision tree using the spam dataset as
Part IV: Feature weights in k-nn

As discussed in class k-nn uses all features equally. In this dataset we know not all features are equally informative. The last part of your homework is to propose a method for weighting features in k-nn.

Design and implement a method to weight features in the k-nn classifier

Hint: you can base weights on scoring functions you have used in decision tree building

TODO: Discuss the method you have designed and insert source code here:

Compare the performance of your new k-nn classifier to the other classifiers you have defined so far on the spam dataset

Handing-in

Use the class handin system to handin your work. You have to turn in this IPython notebook (after editing of course) PA01.ipynb. There are two other (optional) files you can turn in pa01.tar.gz and pa01.zip which you can use to turn in any source code you used.