Joshua Mitchell
CS 7312 A1 Report

Implementation Details

Software and Packages

This project used a Python stack:

- **pandas** for dataframe management
- **scikit-learn** for model selection, metrics, training, and evaluation
- **matplotlib** for plotting
- **os** for file and path navigation and retrieval

Design, execution, and presentation were done with **Jupyter Notebooks**.

Parameters

Data Cleaning

Dataset selection was done by choosing “Categorical” datasets with reasonable amounts of samples (50 - 30,000) and attributes (2 - 25). Once the dataset selection was complete, all incomputable values (i.e. “?”, spaces, etc) were replaced with np.NaN’s:

```
> data = data.replace('?', np.NaN)
```

At which point, any columns with NaN's were dropped from the dataset:

```
> data = data.dropna(axis=1)
```

Finally, all categorical entries were mapped to a discrete value:

```
> data = data.apply(lambda x: pd.factorize(x)[0])
```

Modeling

The first column of every dataset was chosen to be the label:

```
> y_col = 0
> X = data.drop(y_col, 1)
> y = data[y_col]
```

The StratifiedKFold module from scikit-learn was used for validation:

```
> skf = StratifiedKFold(n_splits=n_splits)
> data_split = skf.split(X, y)
```

The DecisionTreeClassifier and ExtraTreeClassifier were used for modelling the entropy based and random based trees, respectively:

```
> model = DecisionTreeClassifier(criterion='entropy')
> model2 = ExtraTreeClassifier(criterion='entropy', max_features=1)
```

Per the documentation, when max_features is set to 1, this acts as a random decision tree.

To ascertain the depth of the tree, a custom algorithm was used to recursively walk the nodes.

Screenshots

Here are some previews of the notebook used to do these experiments. Alternatively, a full PDF copy of the notebook can be found bundled with this report, named **A1 Notebook.pdf.**

In [13]:
```
import pandas as pd
import numpy as np
import sklearn as sk
import matplotlib.pyplot as plt
from os import listdir
from os.path import isfile, join
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.tree import DecisionTreeClassifier, ExtraTreeClassifier

%matplotlib inline
```

In [117]:
```
DATASET_PATH = "datasets/"
datasets = [f for f in listdir(DATASET_PATH) if isfile(join(DATASET_PATH, f)) and f != '.DS_Store']
datasets
```

Out[117]:
```
['agaricus-lepiota.data.txt',
 'primary-tumor.data.txt',
 'hayes-roth.data.txt',
 'monks-2.test.txt',
 'nursery.data.txt',
 'lymphography.data.txt',
 'soybean-large.data.txt',
 'car.data.txt',
 'SPECT.test.txt',
 'adult+stretch.data.txt',
 'balance-scale.data.txt']
```

```
def get_info_for_data(dataset_name, data_dict_list, n_splits=10):
    data_dict = {}

    data = pd.read_csv(DATASET_PATH + dataset_name, header = None)
    data = data.replace('?', np.NaN)
    data = data.dropna(axis=1)
    data = data.apply(lambda x: pd.factorize(x)[0])

    data_dict = {"name": dataset_name, "num_instances": data.shape[0], "num_attributes": data.shape[1], "num_classes":

    print()
    print(len(data_dict_list))
    print(dataset_name)
    print(data.shape)
    print()
    # print(data.head())
    # print()

    # number of attributes, Random average height, and Random average accuracy

    y_col = 0
    X = data.drop(y_col, 1)
    y = data[y_col]
    skf = StratifiedKFold(n_splits=n_splits)
    accuracies = []
    tree_depths = []
    accuracies2 = []
    tree_depths2 = []
    data_split = skf.split(X, y)
    # try:
    for train, test in data_split:
```

```
{'name': 'balance-scale.data.txt', 'num_instances': 625, 'num_attributes': 5, 'num_classes': 23, 'C4.5 Avg Accuracy':
0.67144158209732, 'C4.5 Avg Height': 12.6, 'Random Avg Accuracy': 0.6784309133489461, 'Random Avg Height': 14.5}
```

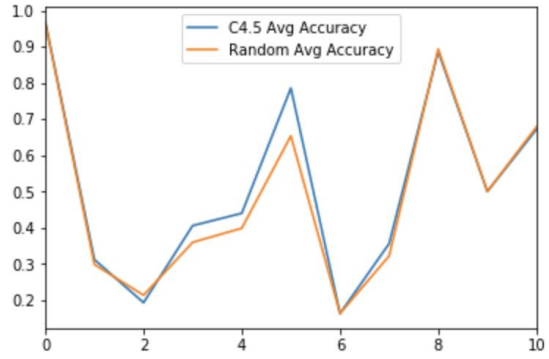
:
```
# Use a table to present your experiment results. The table should include the following
df = pd.DataFrame(data_dict_list)
df = df[['name', 'num_instances', 'num_attributes', 'num_classes', 'C4.5 Avg Accuracy', 'C4.5 Avg Height', 'Random Avg
df
```

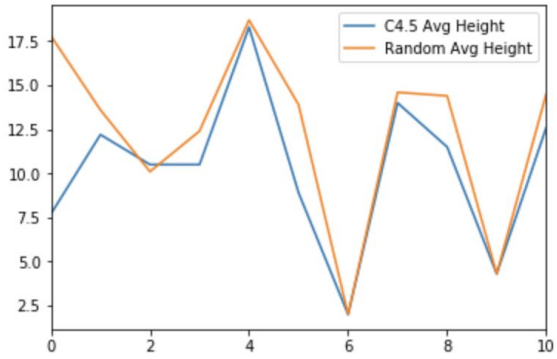
:

	name	num_instances	num_attributes	num_classes	C4.5 Avg Accuracy	C4.5 Avg Height	Random Avg Accuracy	Random Avg Height
0	agaricus-lepiota.data.txt	8124	22	114	0.968512	7.7	0.965679	17.8
1	primary-tumor.data.txt	339	13	46	0.311928	12.2	0.298264	13.6
2	hayes-roth.data.txt	133	5	19	0.193077	10.5	0.213718	10.1
3	monks-2.test.txt	433	7	23	0.405761	10.5	0.359619	12.4
4	nursery.data.txt	12960	9	32	0.439815	18.3	0.398611	18.7
5	lymphography.data.txt	148	19	63	0.785516	8.9	0.653393	13.9
6	soybean-large.data.txt	307	2	21	0.163250	2.0	0.163250	2.0
7	car.data.txt	1728	7	25	0.356052	14.0	0.321987	14.6
8	SPECT.test.txt	187	23	46	0.886725	11.5	0.893129	14.4
9	adult+stretch.data.txt	20	5	10	0.500000	4.3	0.500000	4.3
10	balance-scale.data.txt	625	5	23	0.671442	12.6	0.678431	14.5

Experiment Results

Plots





Tables

	name	num_instances	num_attributes	num_classes	C4.5 Avg Accuracy	C4.5 Avg Height	Random Avg Accuracy	Random Avg Height
0	agaricus-lepiota.data.txt	8124	22	114	0.968512	7.7	0.965679	17.8
1	primary-tumor.data.txt	339	13	46	0.311928	12.2	0.298264	13.6
2	hayes-roth.data.txt	133	5	19	0.193077	10.5	0.213718	10.1
3	monks-2.test.txt	433	7	23	0.405761	10.5	0.359619	12.4
4	nursery.data.txt	12960	9	32	0.439815	18.3	0.398611	18.7
5	lymphography.data.txt	148	19	63	0.785516	8.9	0.653393	13.9
6	soybean-large.data.txt	307	2	21	0.163250	2.0	0.163250	2.0
7	car.data.txt	1728	7	25	0.356052	14.0	0.321987	14.6
8	SPECT.test.txt	187	23	46	0.886725	11.5	0.893129	14.4
9	adult+stretch.data.txt	20	5	10	0.500000	4.3	0.500000	4.3
10	balance-scale.data.txt	625	5	23	0.671442	12.6	0.678431	14.5

	num_instances	num_attributes	num_classes	C4.5 Avg Accuracy	C4.5 Avg Height	Random Avg Accuracy	Random Avg Height
num_instances	1.000000	0.190845	0.393868	0.206282	0.484793	0.211138	0.609317
num_attributes	0.190845	1.000000	0.821264	0.812544	0.127505	0.810171	0.605260
num_classes	0.393868	0.821264	1.000000	0.708877	-0.025014	0.707456	0.586990
C4.5 Avg Accuracy	0.206282	0.812544	0.708877	1.000000	0.048677	0.984489	0.570920
C4.5 Avg Height	0.484793	0.127505	-0.025014	0.048677	1.000000	0.012857	0.770623
Random Avg Accuracy	0.211138	0.810171	0.707456	0.984489	0.012857	1.000000	0.532796
Random Avg Height	0.609317	0.605260	0.586990	0.570920	0.770623	0.532796	1.000000

Summative Analysis

We expected to see, in most cases, that accuracy for the entropy based decision tree model would be higher than that of the random decision tree. This proved to be true, but not by a large margin. In fact, the random tree beat the entropy tree in 3 out of 11 datasets.

Similarly, we expected to see tree depth be larger for random trees. The data supports this. 10 out of 11 datasets had an average entropy tree depth lower than the average random tree depth. Only in 1 dataset did a random tree had a shorter average max depth.

We expected tree complexity to increase as the dataset complexity increased. The data supports this for random trees (where a correlation of 0.58 or greater occurs between average tree height and number of instances, attributes, and classes), but not for entropy-based decision trees (where number of attributes and classes have a -0.02 and 0.13 correlation coefficient with C4.5 average height). It is reasonable to suggest that this is due to the greedy choice and logarithmic scaling of entropy-based decision trees with respect to the number of samples (where as random trees don't benefit from the extra information that comes from more data).

However, the correlation between random tree accuracy and entropy tree accuracy is incredibly high: 0.98. Due to this evidence and the wide range of accuracies for each model (from 0.21 to 0.99), it is fair to suggest that some feature engineering and perhaps a different model is needed for each particular dataset.