CSE 537 HW5

Student Names:
Ali Selman Aydin (110616791)
Emaad Ahmed Manzoor (110622899)

1. Decision tree classifier has been implemented in the “decision tree” folder. You can run the classifier with **python dtree.py**.

This script runs K-fold cross-validation (k=3) on the provided credit-worthiness dataset. The performance statistics are as below.

	ACCURACY	PRECISION	RECALL	F1 SCORE
FOLD 0	0.72	0.81	0.82	0.82
FOLD 1	0.84	0.79	0.66	0.72
FOLD 2	0.86	0.79	0.91	0.84
AVERAGE	**0.80**	**0.80**	**0.80**	**0.80**

To avoid overfitting, the tree is pruned by limiting its depth (d=3) and using the maximum label distribution at the leaf nodes to assign the final prediction label.

For training examples with missing feature values, we do not include them in the conditional entropy calculation.

For test examples with missing feature values, at the internal nodes of the tree for those features, we move the test example down to the most likely child (the one having the largest number of data samples associated with it).

2. Naive Bayes classifier has been implemented in the “Naive Bayes” folder. You can run the classifier by running **do_classify.py** .

The total runtime is 5.41 seconds on my computer. The classifier achieved an overall classification accuracy of **77.3%**.

Class-wise precision, recall and F1 statistics are presented below:

Digit	0	1	2	3	4	5	6	7	8	9
Precision	0.927	0.817	0.865	0.689	0.773	0.723	0.805	0.859	0.741	0.615
Recall	0.844	0.953	0.748	0.800	0.794	0.652	0.769	0.745	0.612	0.800
F1	0.883	0.880	0.802	0.741	0.783	0.686	0.787	0.798	0.670	0.696

Table 1: Precision, recall and F1 statistics for all classes.

The confusion matrix is presented below. Note that the rows are actual classes, and the columns are the predicted classes:

```
[76, 0, 1, 0, 0, 6, 4, 0, 3, 0]
[0, 103, 0, 0, 0, 2, 1, 0, 2, 0]
[1, 3, 77, 7, 1, 0, 6, 2, 5, 1]
[0, 2, 0, 80, 0, 2, 2, 6, 3, 5]
[0, 1, 0, 0, 85, 0, 2, 1, 2, 16]
[2, 2, 1, 14, 3, 60, 1, 1, 2, 6]
[0, 4, 4, 0, 6, 5, 70, 0, 2, 0]
[0, 7, 3, 0, 3, 0, 0, 79, 2, 12]
[2, 3, 3, 12, 2, 6, 1, 1, 63, 10]
[1, 1, 0, 3, 10, 2, 0, 2, 1, 80]
```