CSE 590-04 **Introduction to Machine Learning** Spring 2022

Mid-Term Exam 1
2/26 12:00pm – 2/28 5:00pm

Problem #1 (15 points)

Create a 2-dimensional data set with 30 samples that has the following properties

a) Samples should belong to 2 classes (15 samples per class)
b) Using a Logistic Regression classifier, all samples from both classes can be correctly classified
c) Using a K-NN classifier, with K=3, two samples from each class will always be misclassified. The remaining 26 can be classified correctly.

Generate a scatter plot of your data. Use a different color/symbol for each class. **Indicate the 4 samples that cannot be classified correctly using the KNN and explain the reasons.**

Note: This data should be generated manually and you do not need to run any code on it

Problem #2 (15 points)

Create a 2-dimensional data set with 30 samples that has the following properties

a) Samples should belong to 2 classes (15 samples per class)
b) All samples can be correctly classified using a decision tree classifier with ***only 2 levels***
c) The data cannot be perfectly classified using a linear classifier.

If it is not possible to generate such data, explain why. Otherwise, generate a scatter plot of your data using a different color/symbol for each class. **Indicate the samples that cannot be classified correctly using a linear classifier.**

Display your 2-level decision tree indicating the **feature/threshold used at each non-leaf node and the number of samples at each leaf node.**

Note: This data should be generated manually and you do not need to run any code on it

Problem #3 (25 points)

For this problem, you need to use the built-in sklearn ***California Housing*** dataset. You can load this data using

```
from sklearn.datasets import fetch_california_housing
cal_housing = fetch_california_housing()
```

Divide the data into training and test sets using `train_test_split` and `random_state=38`

The goal is to experiment with few **regression** algorithms and compare their performance on this data.

a) Build and train a LASSO Regression model. Vary the constraint **parameter** α and analyze the results by identifying cases of **overfitting** and **underfitting**. Select the optimal value of α and justify your choice.
b) Build and train a Decision Tree regression model. Vary the **pruning parameter** and analyze the results by identifying cases of **overfitting** and **underfitting**. Select the optimal pruning and justify your choice.
c) Compare the **accuracy** of the 2 methods and the **relevant features** identified by each method and comment on the results.

Problem #4 (45 points)

For this problem, you need to use the built-in sklearn ***digits*** dataset. You can load this data using

Sklearn.datasets.load_digits (*, n_class=10,return_X_y=False, as_frame=False)

Divide the data into training and test sets using `train_test_split` and `random_state=0`

The goal is to train a Random Forest classifier and optimize its performance on this data.

a) Identify the **most important parameters** that affect the performance of the Random Forest classifier and **outline your experimental design** (using 4-fold cross validation) to learn the optimal values for these parameters.

b) Analyze the results of the classifier using its optimal parameters and comment on its generalization capability.

c) **Visualize** and **explain** the relevant features identified by the Random Forest classifier.

- Create a white 8x8 image that represents the original 64 features. Map each identified relevant feature to this 2D image and display it using a grey scale that reflects its importance (e.g. 0 ➔most relevant feature and 255 ➔ least relevant feature).

d) Identify one misclassified sample from each class (if they exist). Visualize each misclassified sample as an 8x8 image, and use its nearest neighbors and the learned important features to explain why it was misclassified.

Hint: for examples on how to read this data and visualize it, check

https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html#sphx-glr-auto-examples-classification-plot-digits-classification-py