# Lab 3: Comparing Classifiers

**This lab is due at 11:59 PM ET on Sunday, September 30, 2018.** Late submissions will be penalized as described in the syllabus.

You may work individually or in groups of two. If working in a group, only one student needs to submit the group's work via Carmen. (Each individual student still must submit his/her own individual component separately.) As with previous labs, you'll do all your work in a Jupyter notebook. This lab is worth 100 points.

In this lab, you'll use the Scikit-Learn[1] library, one of numerous SciKits (SciPy toolkits) for scientific computation in Python. Scikit-Learn is specifically focused on machine learning and data mining. You'll use several different Scikit-Learn classifiers to try to predict the grape variety of wines based on 13 feature attributes.

## 0. Get the data

Download the "Wine" dataset from the UCI Machine Learning Repository[2]. (Note this is not the same as "Wine Quality" dataset.) The dataset contains 178 rows and 14 columns.

## 1. Preliminary data analysis (15 points)

Discuss any observed patterns or data quality issues, such outliers, missing values, and attribute correlations. Visualize those that are interesting or important. Include a dataset description and summary statistics, as always. Describe your observations.

## 2. Data preprocessing (5 points)

Determine whether any transformations or data preprocessing (e.g. feature subset selection, scaling, feature creation) is needed. Apply the transformations (or don't). Describe and justify your decisions—i.e. what did you do and why?

## 3. Choose an evaluation approach (15 points)

Split your data into two subsets, one for testing and the other for training and validation. Set the test dataset aside and do not use it again until part 5.

We discussed several approaches for model evaluation, such as holdout, random subsampling, bootstrap, cross-validation. Choose one. (Refer to the Scikit-Learn "Splitter Classes" API docs.) Report and justify your choice—e.g. what are the advantages of this approach? You will use the same method when evaluating all six models in the next section. For example, if you choose 5-fold cross-validation, then you must use 5-fold cross-validation on the training/validation set for all six models. (This does NOT mean you have to use the same exact folds every time, however.)

[1] Scikit-Learn API reference: scikit-learn.org/stable/modules/classes.html

[2] UCI ML Repo: archive.ics.uci.edu/ml

## 4. Build and evaluate six classifiers (40 points)

In this part, you will build models using each of the following Scikit-Learn classifiers:

- k-Nearest Neighbors **(sklearn.neighbors.KNeighborsClassifier)**
- Decision Tree **(tree.DecisionTreeClassifier)**
- Naïve Bayes **(sklearn.naive_bayes.GaussianNB)**
- Artificial Neural Network **(sklearn.neural_network.MLPClassifier)**
- Support Vector Machine **(svm.LinearSVC)**
- Ensemble Classifier (**ensemble.RandomForestClassifier**)

For each classifier, you must:

- Identify the most important or interesting tunable parameters and the values used, even if you took the defaults.
- Fit and evaluate the model using whatever evaluation method you chose in part 3.
- Report performance statistics—e.g. accuracy, F-measure, AUC.
- Include at least one non-tabular visualization—i.e. plot or chart—of your results.
- Be sure to use ONLY the training and validation sets for this part!

What is your preferred classifier? (Before you can answer this question, you need to have found a set of satisfactory parameters for each classifier.) Justify your choice by comparing results to the other classifiers. What are the pros and cons of your preferred classifier?

## 5. Make predictions and evaluate performance on the test set (15 points)

For this part, you'll return to the test set that you set aside in part 3. Make predictions on the test set using the parameters you settled on in part 4. Compare these results to those found in part 4. Did all six models generalize well? For those that did not, explain why.

## 6. Individual and Exceptional work (10 points)

For your individual and exceptional work, you may choose one of the following topics:

- Use a hyper-parameter optimizer (e.g. **model_selection.GridSearchCV**) to determine optimal parameters for one of the six models, then re-run that classifier. Report any differences in performance.
- Plot ROC curves for at least two of the classifiers and compare the results. Are they consistent with what you observed? Explain.
- Experiment with different weights and/or distance metrics for the **KNeighborsClassifier**. Try at least three different configurations and report your findings. Which, if any, performed better than your "off the shelf" results?
- Try to find another classifier from the Scikit-Learn library that performs better than your best classifier. Based on what you can learn from the Scikit-Learn API documentation, explain why this new classifier worked better than the others.

Alternatively, you may explore a topic of your choosing.

## 7. Submit your work

Be sure to report any problems or bugs you could not resolve. Save your notebook using descriptive filename, such as **lab3_burkhardt_5.ipynb**. You may split your work into multiple notebooks if you wish but be sure to provide consistent and meaningful filenames. Submit your notebook(s) via Carmen, under Lab 3. Do not submit any data files.