Assignment 1

Applied Machine Learning

1. [20 pts] Define each of the following machine learning terms **in your own words**:
   i. the training dataset, testing dataset, and validation dataset
   ii. ground truth, label
   iii. pre-processing, feature, numerical, nominal
   iv. decision surface
   v. model validation, accuracy, cross-validation
   vi. parameters, hyperparameters, overfit

2. [20 pts] Pick the **Iris dataset** of the Scikit-learn datasets for classification which is included in the library (i.e. the dataset can be loaded with `datasets.load_`) and find out the following:
   i. the number of data points
   ii. the number of features and their types
   iii. the number and name of categories (i.e. the `target` field)
   iv. the mean (or mode if nominal) of the first two features

   Next, locate the Wine dataset, load and explore it answering above questions once more.

3. [20 pts] Use the following code piece to display Iris dataset feature pairs:

```
import numpy as np
import seaborn as sns; sns.set(style="ticks", color_codes=True)
import sklearn.datasets
import pandas as pd
iris = sklearn.datasets.load_iris()
iris_df = pd.DataFrame(
    data= np.c_[iris.data, [iris.target_names[v] for v in iris.target]],
    columns= iris.feature_names + ['species'])
cols = iris_df.columns.drop('species')
iris_df[cols] = iris_df[cols].apply(pd.to_numeric)
g = sns.pairplot(iris_df, hue='species')
```

   From the plots, which feature(s) shows the most promising separation power for machine learning?

   Now plot the features of the Wine dataset in question 2. When there are too many features, it is possible to switch the dataset or update your code (pandas `Dataframe` line) to look at low number of features at a time.

4. [20 pts] Consider the **Iris dataset**, refer to the plots in the previous question and discuss/think/outline an unsupervised approach to group the dataset into non-overlapping clusters. Answer the following questions:
   i. Which features would you use?
   ii. Are three clusters obvious from the plots?
   iii. What about four clusters? Roughly mark them manually (i.e. specify their ranges) on a few plots if possible or specify their ranges.

iv. For this problem, is there any relation between classification and clustering since the labels are already given?

5. [20 pts] Using the `scikit-learn` class descriptions for Naive Bayes and decision trees, classify the Iris dataset in question 3. Your code should be very similar to that in the Module 1 Jupyter notebook. In classification with SVM section, the dataset is divided into two portions, one for training and the other for testing. Make sure you use the same input data for the Naive Bayes classifier, and the decision tree classifier.

   Answer the following questions:

   i. Which classifier has the highest performance?
   ii. Does more training help? Test this by increasing the training dataset size from let's say 1% training (with the remaining 99% testing), and then 5% training, 10% training, etc.
   iii. Will the performance plateau? Show it on a plot.

   Hints

   - You need to repeat train-test with different samples to collect statistics (mean/stdev), e.g. 10 times for each experimental point. Plotting each experimental data point with mean and stdev helps reasoning, and answering how much training is sufficient.
   - Use `sklearn.model_selection.train_test_split` and make sure the shuffle is on. Why?