Grading

We've broken the learning objectives down by numerical grade, 1 through 5. You will receive the best grade for which you've met at least one objective plus all fully completed grades below. Please be aware that that means that if you haven't fully met all objectives for getting a 3, then you won't get a 2 or 1 even if you've completed all the objectives for grade 2 or 1. We've structured the objectives so that they build on each other for the most part. So, in order to show that you've learned something for grade 2 then, you'll need to show that you've also met all requirements for grades 3 and 4.

Python libraries allowed:

- Numpy
- Pandas
- Scikit-learn
- Matplotlib
- Scipy

Grade 4 (50% - 62%)

- Set up a programming environment & Jupyter notebook.
- Pick 3 data sets (UCI ML Repository) for the task of classification with different characteristics and load them in the Jupyter notebook. The choice of diverse data sets is important for grading! *Read ahead to the other grades before choosing data sets as certain tasks can influence your choice.*
    - number of samples – small vs. large
    - number of dimensions/features – low vs. high dimensional
    - number of classes – few vs. many classes
- Discuss the characteristics of each data set in detail. Mention the following
    - General characteristics
    - Missing value structure
    - Correlation between numerical features
    - Per feature mention: Number of unique values, number of total values, most common values, feature ranges

Grade 3 (62.5% - 74.5%)

- Write a function that takes the following 3 inputs: (1) the data set and (2) a given data point as row index and (3) k, a positive integer with $k>0$. The function should output the distances and indices of the k nearest neighbors of a given data point. Select only numerical features to be used in determining the nearest neighbors. Choose one data set with more than 6 numerical features with at least one of them having missing values (if the data set is very large, use a random subset of the data but at least 1,000 points)

and your neighborhood function. Using only the output of this function, test and discuss the following points:

- Discuss how you handle missing data?
- Which data points do you consider outliers? Discuss how you determine them (parameters, features used, etc.).
- Determine the nearest neighbor of each point and report the precision score of how often they share the same class out of all points checked.
  - Apply standard scaling to the features prior to the nearest neighbor computation. Discuss why (or why not) this affects the precision.
  - How does this score change with dimensionality (e.g., using random subsets of features)? Discuss its effect.

Grade 2 (75% - 87%)

- Discuss the following characteristics of each data set in detail (for data sets with more than 20 dimensions choose a random subset of their features).
  - Describe characteristics of individual feature distributions. Plot a histogram with matplotlib to show marginal distributions. Discuss the modality structure within individual features (i.e., do you find multiple peaks in numerical features)? If you find a multi-modal structure, discuss it's correlation/relationship to the class labels.
  - Apply the Chi-Squared test of independence to categorical variables (in a pairwise fashion) to test what features are independent. Discuss the results.
- Apply the whitening transform to the chosen data set (see above) before computing nearest neighbors. How does the score change compared to standard scaling? Discuss the results.
- Vary the number of neighbors k (from 1 - 10) when determining the class of a given point (use majority voting). How does the precision score change with k?

Grade 1 (87.5% - 100%)

- Vary both the number of neighbors k and the number of dimensions at the same time in a "grid search"-style fashion (again using the same data set). Report the combination with the highest precision. Do you find a correlation in the precision score between these two parameters? Discuss the results.

- Implement an approach that can deal with heterogeneous data (mix between categorical and numerical features with at least 20% categorical features; choose one data set among the 3, optimally the same you've been using for the previous tasks). Discuss the following:
    - Discuss your treatment of categorical featues.
    - How does the precision score change when adding categorical features to the mix? To solidify your statement make sure to again consider different numbers of neighbors, different dimensions, etc.