Assignment 1

Lecturer: Prof. Moshe Sipper, TA: Shachar Schnapp

Spring 2022

Submission guidelines. **Please read and follow carefully**:

- The exercise is submitted in pairs.
- Submit via Moodle.
- The submission should include two separate files:
  1. A Jupyter notebook file that includes your answers for part 1,
  2. A Jupyter notebook file that includes your answers for part 2.
- For questions, use the exercise forum, or if they are not of public interest, send them via the course requests system.
- Grading: part 1 is 65 points and part 2 is 35 points.

Part 1. Your answers for the this part should be included in Jupyter file: `part1.ipynb`.

- Download the `iris.csv` from here. Use pandas.read_csv to load the dataset and run preliminary data analysis on it (features, samples, ranges, scales, variance, and any other information that you find relevant).
- Use `sklearn` function `test_train_split` to split the data to test-set and train-set, for each test-size ratio $r \in [0.1, 0.2, 0.3, ..., 0.9]$, and use skleran `LogisticRegressor` to train logistic regressor on the train-set and evaluate the accuracy on the test-set. Use `matplotlib.pyplot` to plot the accuracy of each r (using r as x axis and accuracy as y axis).
- The 150-sample dataset is completely *balanced* (what does that mean?). Create a sub-dataset from it, of size 110, which is *unbalanced*.
- Implement code for label balancing to take your new (sub)-dataset and generate a new dataset with label balancing. Use the the following pseudo code:
  - First, find l_{min}: the number corresponding to the label that appears the least in the data.
  - Then, for each label in the dataset, randomly select only l_{min} samples and add them to the new dataset.

For the same values of r that you used in the previous part, split the balanced dataset that you created into test-set and train-set, train linear regression on the train-set, and print the accuracy of the fitted (trained) logistic regressor on the test-set.

- Run a naïve k-features selection algorithm that for each set of k features from the data trains a classifier on the train-set, and selects the set of k features that achieved the best accuracy on the test-set. Implement a function that gets `train-set`, `test-set`, and `k` and returns the best k features from the dataset and the accuracy achieved on the test-set. Run the function with $k = 2$ and print the results.

Part 2. You answers for the this part should be included in Jupyter file: `part2.ipynb`.

- Download the `diabetes.csv` from here. Use pandas.read_csv to load the dataset and run preliminary data analysis on it (features, samples, ranges, scales, variance, and any other information that you find relevant).
- Use `sklearn` function `test_train_split` to split the data to test-set and train-set, for each test-size ratio $r \in [0.1, 0.2, 0.3, ..., 0.9]$, and use skleran `LinearRegressor` to train a linear regressor model on the train-set and evaluate the accuracy on the test-set. Use `matplotlib.pyplot` to plot the mean absolute error of each r (using r as x axis and accuracy as y axis).
- Run a naïve k-features selection algorithm (use mean absolute error instead of accuracy) for $k = 2$ and $k = 5$ prints the results.