Exercise 2 –
Virus Test Challenge – Data Preparation

At last, we are about to start the virus test challenge. Like any other machine learning project, our first step is data preparation. By the end of this task, you should have 3x2 data sets, as follows:

- Split the data to – train (50-75%), validation (25-15%), and test (25-10%)
- For each set –
    - Keep a copy of the raw data
        - These datasets will serve as references, in case you would like to re-examine (and possibly reconsider) the transformations you've made.
    - Prepare the data for use
        - Clean, scaled, transformed, without missing values, while keeping only the relevant attributes

Wet Mandatory Assignment

Write a Python script that will execute the following

1. Load the Virus Test Challenge data from the virus_hw2.csv file
    - Can be found at the "The Virus Test Challenge" section in the course site
2. Identify and set the correct type of each attribute
    - It is advised to do it with Pandas
    - The "TestResultsCode" attribute serves as the label
3. Perform the following data preparation actions only on the training set and apply the resulted transformations to both the test and the validation sets *
    - Imputation
        - Filling up missing values
    - Data Cleansing
        - Outlier detection
        - Type/Value modification – Use mainly for Nominal attributes
    - Normalization (scaling)
        - Use with care, do not destroy attribute dependencies
    - Feature Selection
        - Use at least one filter method and one wrapper method
4. Split the data to train, test, validation sets
5. Save the 3x2 data sets in CSV files

* label-dependent methods (such as impute by class mean) may **not** be applied on the test set! Since the test set simulates a case where you don't know the label. For the same reason, statistics (such as mean/median) should not be calculated using data from the test set. Whether to apply such transformation on the validation set is your choice.

Dry Mandatory Assignment

1. In Tutorial 02 we learned that given a feature space R^d, a vector $w \in R^d$ defines a homogeneous halfspace using a classification function:

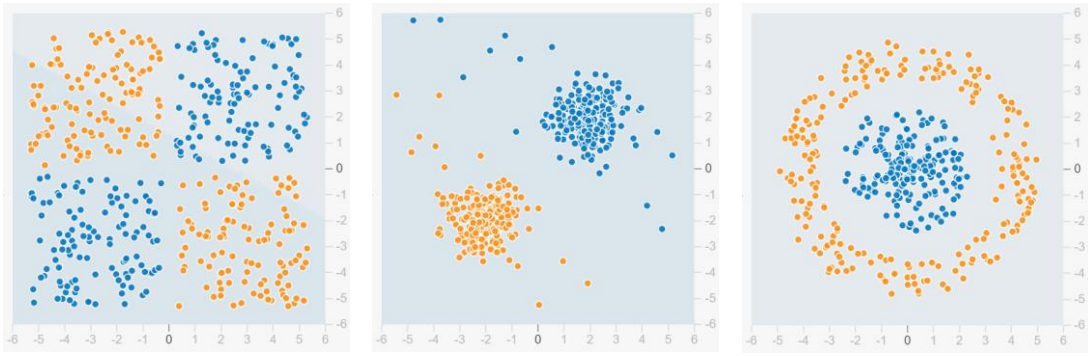
$$f(x) = sgn(\langle x, w \rangle)$$

where x can be any vector in the feature space.
Show how to generalize this function to the non-homogeneous case.
Specifically, find a function $g: R^d \to R^{d+1}$ such that a vector $z \in R^{d+1}$ will define a **non**-homogeneous halfspace in R^d using:

$$f(x) = sgn(\langle g(x), z \rangle)$$

2. Following are 3 datasets in the R^2 feature space with 2 classes.

(A) (B) (C)



For each dataset above, choose all models that can perfectly fit it (0 training error) from the following list:

i. Perceptron
ii. kNN with $k = 1$ *(here assume a point isn't considered a neighbor of itself)*
iii. kNN with $k = 3$ *(here assume a point isn't considered a neighbor of itself)*
iv. Decision tree with at most 4 leaves

3. Logistic regression requires solving the following optimization problem:

$$w^* = arg\ arg \sum_{(x_i,y_i)\in D} ln\ ln \frac{1}{1+e^{-y_i\langle x_i,w\rangle}}$$

$$= arg\ arg \sum_{(x_i,y_i)\in D} ln\ ln\ \sigma(y_i\langle x_i,w\rangle) \quad \text{ـــ} = l(x_i,y_i,w)$$

It is often solved using stochastic gradient ascent, where the iterative step is:

$$w^{(k+1)} = w^{(k)} + \mu \nabla_w l\left(x_i, y_i, w^{(k)}\right)$$

3.1. Compute the partial derivative $\frac{\partial}{\partial w_j} l(x_i, y_i, w)$ for $j = 1, \dots, d$.
Hint: Use the partial derivative $\frac{\partial}{\partial z}\sigma(z) = \sigma(z)\big(1 - \sigma(z)\big)$.

3.2. Using the partial derivative you computed, write the gradient vector $\nabla_w l(x_i, y_i, w)$.
Hint: the gradient takes the form $\nabla_w l(x_i, y_i, w) = g(x_i, y_i, w) \cdot x_i$ for some function $g: R^d \times R \times R^d \to R$.

Non-Mandatory (Bonus) Assignments (15%)

The following list includes additional, non-mandatory, assignments. You are highly encouraged to do at least some of them. Each functional implementation of ANY assignment will get a bonus, but more importantly – it will help you in getting better results

A. Identify the role of the attributes with respect to the classes, i.e. the "TestResultsCode" label.
   - Provide an explanation (in the submitted documentation).
     The "role" being a relation (or lack of relation) that you found between the features and the label.
   - Implement and use feature selection algorithm, and use it on the dataset
     - Provide a python file with your own implementation
     - Calling a package that includes Relief doesn't count … 😤
   - Compare and discuss pros and cons of the feature selection capabilities of the algorithm you chose vs the other methods you know

Please submit

1. The list of selected features in selected_columns.csv file. A feature should be marked as "selected" if it passes your feature selection procedures. If you convert a feature to one-hot and use at least one of its features, it counts as "selected". Don't choose more than 18 features.
2. The Python files you made
3. A PDF document that explains your process and any significant decision/insight you would like to share. You may also include unsuccessful attempts of ideas you had. Be concise in your document. Plots and tables are a useful method to achieve this. Submission of any iPython Notebooks **isn't** allowed.

4. The PDF should also contain the answers to the "dry" questions.

Comments

- The three "prepared" data sets that were generated by your process should NOT include ANY missing value
- The minimal set of useful attributes includes less than half of the original attributes, and the set is not unique
    - Namely, there are redundancies among the useful attributes
- It is difficult (might be even impossible) for you at this point to identify and select exactly the set of useful attributes
    - Our grading policy will favor approximating this set as close as possible, but will "punish" any loss of a useful attribute
    - This means that you should employ a conservative approach!
        - Recall that it is difficult to recover, at later stages, from the loss of relevant features at the data preparation stage
    - Your grade for this exercise is largely determined by the features you identify (and those you decide to disregard)
- Keep plotting and looking at the data in various forms and aspects
    - You'll be amazed how much insights can be gained from visualization
- The leading Python Packages that you should use are Pandas, Scikit-learn, and Matplotlib. These packages include a rich set of very useful examples
    - You are highly encouraged to look at it, "steal" some ideas, and shorten your implementation time
    - Moreover, It is permissible, and actually recommended, to look for additional Python packages and implementations, and either use explicitly or borrow ideas
- We highly recommend that you implement the bonus exercises, they can increase your grade well beyond 100.
- Use of Python 3 is mandatory.

This exercise must be submitted in pairs!

- **You should submit only one copy but remember to document who are the contributors**
- **"No Couples Swapping" during the semester**
    - **At least not without a formal approval**