Applied Machine Learning
Exercise 3

Prof. Dr. Amr Alanwar

1.1 Data Preprocessing (3 Points)

1.1.1 Datasets

You are required to pre-process given datasets.
Airfare and demand: `https://users.stat.ufl.edu/~winner/data/airq402.dat`
Wine Quality: `http://archive.ics.uci.edu/ml/datasets/Wine+Quality`

Tasks:

1. Convert any non-numeric values to numeric values. For example, replace a country name with an integer value or, more appropriately, use hot-one encoding. [Hint: use hashmap (dict) or `pandas.get_dummies`]. Please explain your solution.
2. Handle sparse data/missing values.
3. Split the data into a training set (80%) and a test set (20%).

1.2 Linear Regression with Gradient Descent (7 Points)

Part A (4 Points): Implement Linear Regression with Gradient Descent In this part, you are required to implement a linear regression algorithm with gradient descent.

For each dataset provided above:

- A set of training data $D_{\text{train}} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})\}$, where $x \in \mathbb{R}^M$, $y \in \mathbb{R}$, N is the number of training examples, and M is the number of features.
- The Linear Regression model is given as $\hat{y}_n = \sum_{m=1}^{M} \beta_m x_m^n$.
- The Least Square Loss Function is given as $l(x, y) = \sum_{n=1}^{N} (y_n - \hat{y}_n)^2$.
- Minimize the loss function $l(x, y)$ using the Gradient Descent algorithm. Implement (`learn-linregGD` and `minimize-GD`). Choose $i_{\max}$ between 100 to 1000.
- Choose three suitable values for the step length $\alpha > 0$. For each step length:
  - In each iteration of the `minimize-GD` algorithm, calculate $|f(x_{i-1}) - f(x_i)|$ and, at the end of learning, plot it against the iteration number i. Explain the graph.
  - In each iteration step, also calculate the RMSE on the test set RMSE $= \sqrt{\frac{\sum_{q=1}^{T} (y_{\text{test}}^q - \hat{y}^q)^2}{T}}$ and, at the end of learning, plot it against the iteration number i. Explain the graph.

Part B (3 Points): Step Length for Gradient Descent This task is based on Part A. Implement two algorithms: `steplength-armijo` and `steplength-bolddriver` given below. For each step length algorithm:

- In each iteration of the `minimize-GD` algorithm, calculate $|f(x_{i-1}) - f(x_i)|$ and, at the end of learning, plot it against iteration number i. Explain the graph.
- In each iteration step, also calculate the RMSE on the test set and plot it against iteration number i. Explain the graph.

Compare different step length algorithms. Compare the RMSE graphs of `steplength-armijo` and `steplength-bolddriver` with the three fixed step lengths. Explain your graph.

1.3 ANNEX

- You can use `numpy` or `scipy` for linear algebra operations.
- Use `pandas` for data reading and processing.
- Use `matplotlib` for plotting.
- Do not use any machine learning library (e.g., scikit-learn) to solve the problem; doing so will result in zero points for the task.

Algorithm 1 Learn Linear Regression via Loss Minimization: GD

procedure LEARN-LINREG-GD($\mathcal{D}^{\text{train}} := \{(x_1, y_1), \ldots, (x_N, y_N)\}$, α, $i_{\max} \in \mathbb{N}$, $\epsilon \in \mathbb{R}^+$)
  $X := (x_1, x_2, \ldots, x_N)^T$
  $y := (y_1, y_2, \ldots, y_N)^T$
  $\hat{\beta}_0 := (0, \ldots, 0)$
  $\hat{\beta} := \text{Minimize-GD}\left(f(\hat{\beta}) := (y - X\hat{\beta})^T(y - X\hat{\beta}), \hat{\beta}_0, \alpha, i_{\max}, \epsilon\right)$
  return $\hat{\beta}$
end procedure

Algorithm 2 Gradient Descent: MINIMIZE-GD

procedure MINIMIZE-GD($f : \mathbb{R}^N \to \mathbb{R}$, $x_0 \in \mathbb{R}^N$, α, $i_{\max} \in \mathbb{N}$, $\epsilon \in \mathbb{R}^+$)
  for $i = 1, \ldots, i_{\max}$ **do**
    $d := -\frac{\partial f}{\partial x}(x_{i-1})$
    $\alpha_i := \alpha(f, x_{i-1}, d)$
    $x_i := x_{i-1} + \alpha_i \cdot d$
    if $f(x_{i-1}) - f(x_i) < \epsilon$ **then**
      return x_i
    end if
  end for
  error "not converged in $i_{\max}$ iterations"
end procedure

- x_0 : start value
- α : step length function, e.g., STEPLENGTH-ARMIJO (with fixed δ).
- $i_{\max}$: maximal number of iterations
- ϵ : minimum stepwise improvement

Algorithm 3 Armijo Step Length: STEPLENGTH-ARMIJO

procedure STEPLENGTH-ARMIJO($f : \mathbb{R}^N \to \mathbb{R}, x \in \mathbb{R}^N, d \in \mathbb{R}^N, \delta \in (0, 1)$)
  $\alpha \leftarrow 1$
  while $f(x) - f(x + \alpha d) < \alpha \delta d^T d$ **do**
    $\alpha \leftarrow \alpha/2$
  end while
  return α
end procedure

- x : last position.
- d : descend directions
- δ : minimum steepness

Algorithm 4 Bold Driver Step Length: StepLength-BoldDriver

procedure StepLength-BoldDriver($f : \mathbb{R}^N \to \mathbb{R}$, $x \in \mathbb{R}^N$, $d \in \mathbb{R}^N$, α^{old}, α^+, $\alpha^- \in (0,1)$)
    $\alpha := \alpha^{\text{old}}\alpha^+$
    while $f(x) - f(x + \alpha d) \leq 0$ **do**
        $\alpha = \alpha\alpha^-$
    end while
    return α
end procedure

- α^{old} : last step length
- α^+ : step length increase factor, e.g., 1.1.
- α^- : step length decrease factor, e.g., 0.5.