Homework 2 **CSCE 633**
Due: 11.59pm on October 11, 2021

Instructions for homework submission
a) Please write a brief report and *include your code.*
b) Create a **single pdf** and submit it on **CANVAS**. Please do not submit .zip files or colab notebooks.
c) Please start early :)
d) The maximum grade for this homework, excluding bonus questions, is **10 points** (out of 100 total for the class). There is **1 bonus point**.

Question 1 (3 points)
1-dimensional linear regression: Assume a 1-dimensional linear regression model $y = w_0 + w_1 x$. The residual sum of squares (RSS) of the training data $\mathcal{D}^{train} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ can be written as:

$$RSS(w_0, w_1) = \sum_{n=1}^{N} (y_n - w_0 - w_1 x_n)^2$$

We will estimate the weights w_0, w_1 by minimizing the RSS error.

(a) (1.5 points) Show that minimizing RSS results in the following closed-form expressions:

$$w_1^* = \frac{\sum_{n=1}^{N} x_n y_n - N\left(\frac{1}{N}\sum_{n=1}^{N} x_n\right)\left(\frac{1}{N}\sum_{n=1}^{N} y_n\right)}{\sum_{n=1}^{N} x_n^2 - N\left(\frac{1}{N}\sum_{n=1}^{N} x_n\right)^2}$$

$$w_0^* = \left(\frac{1}{N}\sum_{n=1}^{N} y_n\right) - w_1\left(\frac{1}{N}\sum_{n=1}^{N} x_n\right)$$

Note: Set the partial derivatives $\frac{\vartheta RSS(w_0,w_1)}{\vartheta w_0}$ and $\frac{\vartheta RSS(w_0,w_1)}{\vartheta w_1}$ equal to 0. Then solve a 2×2 system of linear equations with respect to w_0 and w_1.

(b) (1 point) Show that the above expressions for w_0^* and w_1^* are equivalent to the following:

$$w_1^* = \frac{\sum_{n=1}^{N} (x_n - \bar{x})(y_n - \bar{y})}{\sum_{i=1}^{N} (x_n - \bar{x})^2}$$

$$w_0^* = \bar{y} - w_1 \bar{x}$$

where $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$ and $\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n$ are the sample means of input features and outcome values, respectively.

(c) (0.5 point) How would you interpret the above expression in terms of the descriptive statistics (e.g. sample mean, variance, co-variance) of populations $\{x_n\}_{n=1}^N$ and $\{y_n\}_{n=1}^N$?

Question 2: Machine learning with Pokemon GO
Recent studies have found that novel mobile games can lead to increased physical activity. A notable example is Pokemon Go, a mobile game combining the Pokemon world through augmented reality with the real world requiring players to physically move around. Specifically, in the following study, researchers have found that Pokemon Go leads to increased levels of physical activity for the most engaged players!
`https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5174727/`

In this problem, our goal is to predict the combat points of each pokemon in the 2017 Pokemon Go mobile game. Each pokemon has its own unique attributes that can help predicting its combat points. These include:

1. Stamina
2. Attack value
3. Defense value
4. Capture rate
5. Flee rate
6. Spawn chance
7. Primary strength

Inside the "Homework 2" folder on CANVAS you will find the data file (named "hw2_data.csv") that will be used for our experiments. The rows of these files refer to the data samples (i.e., pokemon samples), while the columns denote the name of the pokemon (column 1), its attributes (columns 2-8), and the combat point outcome (column 9). You can *ignore column 1* for the rest of this problem.

(i) (0.5 point) Data exploration: Which are categorical and which are numerical attributes (columns 2-8) of this dataset?

(ii) (0.5 point) Data exploration: Plot 2-D scatter plots and compute the Pearson's correlation coefficient between the numerical attributes and the outcome of interest. Which attributes would be the most predictive of the outcome of combat points?
Note: The Pearson's correlation coefficient is a measure of linear association between two variables. It ranges between -1 and 1, with values closer to 1 indicating high degree of association

between a feature and the outcome. For more details, see this link: `https://en.wikipedia.org/wiki/Pearson_correlation_coefficient`. You can use any available library to compute this metric.

(iii) (0.5 point) Data exploration: Plot 2-D scatter plots and compute the Pearson's correlation coefficient between the numerical attributes themselves. Which variables are the most correlated to each other?

(iv) (1 point) Pre-processing of categorical variables: Categorical variables require special attention because usually they cannot be the input of regression models as they are. A potential way to treat categorical variables is to simply convert each value of the variable to a separate number. However, this might impute non-existent relative associations between the values, which might not always be representative of the data (e.g., if we assign "1" to the value "green" and "2" to the value "red", the regression algorithm will assume that "red" is greater than "green," which is not necessarily the case). For this reason, we can use a "one hot encoding" to represent categorical variables. According to this, we will create a binary column for each category of the categorical variable, which will take a value of 1 if the sample belongs to that category, and 0 otherwise. For each categorical variable of the problem, count the number of different values and **implement** the one hot encoding. For the remaining of the problem, you will be working with the one hot encoding of the categorical variables.
Note: You can find more information on different types of pre-processing categorical variables in the following links:

`https://pbpython.com/categorical-encoding.html`

`https://stats.idre.ucla.edu/spss/faq/coding-systems-for-categorical-variables-in-regression-analysis-2/`

(v) (1.5 points) Predicting combat points: The goal of this question is to predict the combat points using the numerical attributes, as well as the categorical attributes that were pre-processed with the one hot encoding process. **Implement** a linear regression model using the ordinary least squares (OLS) solution. How many parameters does the model have? To test your model, randomly split the data into 5 folds and use a 5-fold cross-validation. For each fold compute the square root of the residual sum of squares error (RSS) between the actual and predicted outcome variable. Also compute the average square roof of the RSS over all folds.
Hint: You will build the data matrix $\mathbf{X} \in \mathcal{R}^{N_{train} \times D}$, whose rows correspond to the training samples $\mathbf{x_1}, \ldots, \mathbf{x_{N_{train}}} \in \mathcal{R}^{D \times 1}$ and columns to the D features (including the constant 1 for the intercept): $\mathbf{X} = \begin{bmatrix} 1, \mathbf{x_1^T} \\ \vdots \\ 1, \mathbf{x_N^T} \end{bmatrix} \in \mathcal{R}^{N_{train} \times D}$. Then use the ordinary least squares solution that we learned in class: $\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.
Note: You can use libraries for matrix operations and random sampling, but please implement the linear regression algorithm, the 5-fold cross-validation process, and the RSS error computation.

(vi) (1 point) Predicting combat points: Repeat the same experiment as in question (v), but instead of linear regression, **implement** linear regression with $l2$-norm regularization. Experiment and report your results with different values of the regularization term λ.
Note: You can use libraries for matrix operations and random sampling, but please implement the regularized linear regression algorithm, the 5-fold cross-validation process, and the RSS error computation.
Note: Use the same sample split as in question **(v)** for better comparison between regularized

and non-regularized regression.

(vii) (Bonus, 0.5 point) Based on your findings from questions (ii) and (iii), use linear regression and experiment with different feature combinations. Report your results.

(vii) (Bonus, 0.5 point) Use linear regression with $l1$-norm regularization. Report your results. How do the estimated regression weights change when using the $l1$-norm regularization compared to the $l2$-norm regularization?
Note: You can use an existing library for running linear regression with $l1$-norm regularization.

(viii) (1 point) Use the sample mean of the outcome to binarize the data. Run a logistic regression model to classify between low and high combat points. To evaluate the model, randomly split 80% of data into training and 20% into testing. Report the accuracy of the classifier on the test data.
Note: ou can use an existing library for running logistic regression from the available libraries. You can use the *sklearn.linear_model.LogisticRegression* function setting the 'penalty' parameter to 'none'.

(ix) (1 point) Run a logistic regression model with regularization to classify between low and high combat points. Use the same training and testing split as in question (viii). Find the optimal regularization term using a 5-fold cross-validation on the training data. Use the regularization term that provided the best results from the cross-validation, and evaluate the regularized logistic regression on the test data. Report the final accuracy on the test data, as well as the best hyperparameter.
Note: You can use a built-in *sklearn.linear_model.LogisticRegression* function for the logistic regression from the available libraries.