Homework 2 **CSCE 633**
Due: 5pm on February 26, 2020
Submission: HRBB 401A (Prafulla's office) between 12-5pm on 2/26

Instructions for homework submission

a) Please write a brief report. At the end of the report or at the end of each answer (whatever works best), please include your code. Make sure that your code is commented clearly. Print the report, including the code.
c) **Staple all your answers and hand them out in paper in class or in the instructor's office.**
d) Please start early :)
e) The maximum grade for this homework, excluding bonus questions, is **10 points** (out of 100 total for the class). There are **4 bonus points**.

Question 1

Predicting urban traffic: We would like to predict the slowness in traffic in the streets of the city of San Paolo in Brazil. We use the Behavior of the urban traffic of the city of Sao Paulo in Brazil Data Set in the following UCI Machine Learning Repository: `https://archive.ics.uci.edu/ml/datasets/Behavior+of+the+urban+traffic+of+the+city+of+Sao+Paulo+in+Brazil`.
The database was created with records of behavior of the urban traffic from December 14, 2009 to December 18, 2009. The traffic was registered from 7:00 to 20:00 between Monday to Friday every 30 minutes.
Inside "Homework 2" folder on Piazza you can find two files including the train and test data (named "hw2_ question1_train.csv" and "hw2_ question1_test.csv") for our experiments. The rows of these files refer to the data samples, while the columns denote the features (columns 1-17) and the traffic slowness outcome (column 18), as described bellow:

1. Hour
2. Immobilized bus
3. Broken Truck
4. Vehicle excess
5. Accident victim
6. Running over
7. Fire Vehicles
8. Occurrence involving freight
9. Incident involving dangerous freight
10. Lack of electricity
11. Fire
12. Point of flooding
13. Manifestations

14. Defect in the network of trolleybuses
15. Tree on the road
16. Semaphore off
17. Intermittent Semaphore
18. Slowness in traffic (%) (this is the **outcome**)

(i) (1 point) Data exploration: Plot the histograms of the features and the outcome of interest.

(ii) (1 point) Data exploration: Compute the Pearson's correlation coefficient between each of the features and the outcome of interest.
Note: In order to make the Hour feature an integer, you can discretize it to include 27 values from 1 to 27, e.g., 7.00 is 1, 7.30 is 2, 8.00 is 3, etc.
Note: The Pearson's correlation coefficient is a measure of linear association between two variables. It ranges between 0 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) (4 points) Using the train data, **implement** a linear regression model using the ordinary least squares (OLS) solution. How many parameters does this model have? Please provide your code as the answer to this problem.
Hint: You will build the data matrix $\mathbf{X} \in \mathcal{R}^{N_{train} \times 18}$, whose rows correspond to the training samples $\mathbf{x_1}, \ldots, \mathbf{x_{N_{train}}} \in \mathcal{R}^{18 \times 1}$ and columns to the features (including the constant 1 for the intercept): $\mathbf{X} = \begin{bmatrix} 1,\ \mathbf{x_1}^T \\ \vdots \\ 1,\ \mathbf{x_N}^T \end{bmatrix} \in \mathcal{R}^{N_{train} \times 18}$. 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.

(iv) (1 point) Test your model on the test data and compute the Pearson's correlation coefficient and the residual sum of squares error (RSS) between the actual and predicted outcome variable. **Note:** You can use any available library for this.

(v) (Bonus, 2 points) Experiment with different feature combinations and report your findings. What do you observe?

(vi) (3 points) Use the mean sample value of the outcome to binarize the data. Run a logistic regression model to classify between low and high traffic. Report the accuracy of the classifier on the test data.

(vii) (Bonus, 2 points) Run a logistic regression model with regularization to classify between low and high traffic. Perform a cross-validation on the train set to find the optimal hyperparameter value for the regularization term. Use the best model that yielded from the cross-validation process and run it on the test data. Report the final accuracy on the test data.