HOMEWORK 4

Problem 1.

Consider the following loss function $L(\beta)$ where $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ are the observations in our dataset. x_i is a d-dimensional input vector, i.e., there are d features in our dataset. So, x_{ij} corresponds to the j^{th} feature in the i^{th} observation. y_i corresponds to the outcome variable for observation i. λ is some scalar constant.

$$L(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{d} \beta_j^2$$

i. Can you estimate the parameters β analytically as we did for linear regression? If yes, derive the analytical expression. If no, justify why not.

ii. What is the gradient $\nabla L(\beta)$?

iii. Write down the update step for gradient descent.

iv. Write down the pseudo-code of a stochastic gradient descent algorithm to estimate the parameters β.

Problem 2.[1]

From the nltk[2] corpus (or elsewhere), import a dataset containing tweets that are pre-labeled and classified as denoting a positive sentiment or a negative sentiment. After appropriately segregating the dataset into a training and test set, you will build a logistic regression sentiment classifier. Feel free to explore and create your own features which can be used as inputs to the logistic regression model. Feel free to also explore appropriate pre-processing steps to handle the tweet dataset.

i. Once you have come up with a list of features, clearly explain what features you are using in your logistic regression model, and write down the model mathematically.

[1] This is an open-ended and a slightly more challenging problem. Points will be awarded on the basis of completeness, scope, and clarity of the solution. Submit your best attempt even if incomplete.
[2] https://www.nltk.org/howto/corpus.html

ii. Write the likelihood function for your logistic regression model obtained in (i).

iii. Train the logistic regression classifier using a black-box implementation and evaluate its performance on the test dataset.

iv. Also train the logistic regression classifier by minimizing the negative log-likelihood function using a numerical optimization procedure: gradient descent or stochastic gradient descent. Compare with the coefficients obtained in step (iii).