Lab Three: Extending Logistic Regression

Due Mar 4 by 11:59pm **Points** 10 **Submitting** a file upload

Lab Assignment Three: Extending Logistic Regression

In this lab, you will compare the performance of logistic regression optimization programmed in scikit-learn and via your own implementation. You will also modify the optimization procedure for logistic regression.

This report is worth 10% of the final grade. Please upload a report (**one per team**) with all code used, visualizations, and text in a rendered Jupyter notebook. Any visualizations that cannot be embedded in the notebook, please provide screenshots of the output. The results should be reproducible using your report. Please carefully describe every assumption and every step in your report.

Dataset Selection

Select a dataset identically to the way you selected for the lab one (i.e., table data). You are not required to use the same dataset that you used in the past, but you are encouraged. You must identify a classification task from the dataset that contains **three or more classes to predict**. That is it cannot be a binary classification; it must be multi-class prediction.

Grading Rubric

- Preparation and Overview (**30 points tota**l)
  - [**20 points**] Explain the task and what business-case or use-case it is designed to solve (or designed to investigate). Detail exactly what the classification task is and what parties would be interested in the results. For example, would the model be deployed or use mostly for offline analysis?
  - [**5 points**] (*mostly the same processes as from previous labs*) Define and prepare your class variables. Use proper variable representations (int, float, one-hot, etc.). Use pre-processing methods (as needed) for dimensionality reduction, scaling, etc. Remove variables that are not needed/useful for the analysis. Describe the final dataset that is used for classification/regression (include a description of any newly formed variables you created).
  - [**5 points**] Divide you data into training and testing data using an 80% training and 20% testing split. Use the cross validation modules that are part of scikit-learn. **Argue "for" or "against" splitting your data using an 80/20 split. That is, why is the 80/20 split appropriate (or not) for your dataset?**
- Modeling (**50 points total**)
  - The implementation of logistic regression must be written only from the examples given to you by the instructor. No credit will be assigned to teams that copy implementations from another source, regardless of if the code is properly cited.
  - [**20 points**] Create a custom, one-versus-all logistic regression classifier using numpy and scipy to optimize. Use object oriented conventions identical to scikit-learn. You should start with the template

developed by the instructor in the course. You should add the following functionality to the logistic regression classifier:

- Ability to choose optimization technique when class is instantiated: either steepest descent, stochastic gradient descent, or Newton's method.
- Update the gradient calculation to include a customizable regularization term (either using no regularization, L1 regularization, L2 regularization, or both L1 and L2 regularization). Associate a cost with the regularization term, "C", that can be adjusted when the class is instantiated.

- [**15 points**] Train your classifier to achieve good generalization performance. That is, adjust the **optimization technique** and the value of the **regularization term "C"** to achieve the best performance on your test set. Visualize the performance of the classifier versus the parameters you investigated. Is your method of selecting parameters justified? That is, do you think there is any "data snooping" involved with this method of selecting parameters?
- [**15 points**] Compare the performance of your "best" logistic regression optimization procedure to the procedure used in scikit-learn. Visualize the performance differences in terms of training time and classification performance. **Discuss the results**.

- Deployment (**10 points total**)
  - Which implementation of logistic regression would you advise be used in a deployed machine learning model, your implementation or scikit-learn (or other third party)? Why?
- Exceptional Work (**10 points total**)
  - You have free reign to provide additional analyses. **One idea**: Update the code to use either "one-versus-all" or "one-versus-one" extensions of binary to multi-class classification.
  - One idea (**required for 7000 level students**): Implement an optimization technique for logistic regression using mean square error as your objective function (instead of binary entropy). Your solution should be able to solve the binary logistic regression problem in one gradient update step.