Wonyoung Lee

Group Assignment 1 Report: Clinical Readmission

The dataset we were given was a table, or matrix, where each row indicated a patient and each column was a feature such as age or mean number of platelets. There were 68 features including the label which is whether the patients were readmitted (label 1) or not (label 0). Our task was to make a machine learning model that would classify whether a patient should be readmitted or not.

Our group used Logistic Regression as the machine learning model of this project. We chose this amongst other models because of its efficiency in calculation. Unlike other models such as Random Forest, Logistic Regression does not calculate each and every weight for each individual feature. Rather, it initializes the weights randomly and "nudges" the weights in the direction of less cross-entropy loss using the Stochastic Gradient Descent. Also while other models such as the k Nearest-Neighbor are very space heavy, or memory heavy, the Logistic Regression is much less harsh on memory space.

For this specific task, we decided to use oversampling on the minority class data because we observed a big class imbalance. There were about 10 times more label 0 patients than label 1 patients. Therefore we implemented an oversampling function which replicated the minority data z times where z is the integer closest to yet smaller than the factor of the majority and minority data (z = number of majority data / number of minority data). More specifically, the oversampling function takes in the training set and gives back a modified dataset where the minority data is replicated so that the number is similar to the majority data. We used the numpy "repeat" function for this function.

With the training set which was split from the full dataset (the split ratio was 20:80 = test : training), we ran the Logistic Regression program including SGD on the mini batches. We changed the number of batches and epochs in the beginning but settled on 1200 for batch size and 500 for number of epochs. We tried out different learning rates and figured out that the F1 score was best at 0.001 and 0.0015 learning rate.

We tried to have the PCA or Principal Component Analysis to work in our model but it did not perform well when we deleted the column of class label from the dataset. Therefore, for further improvement, we would have to improve our PCA function to work better without the class label information. Also, we could use methods to prevent overfitting such as Cross-Validation or Regularization. Regularization would especially help since it is a process

that adds noise (Gaussian Noise) to the training set, and especially since Logistic Regression depends heavily on the weights of the features.

In this program, Josh took part in programming the PCA, Annie took part in programming the SGD, and I took part in programming the oversampling method.