Charlotte DAUMAL
Clémence KIEHL

December 23, 2022

Machine Learning Mini Project - Report

Introduction

The aim of our mini project was to predict as accurately as possible the experimental conditions (KAT5, CBP or eGFP) under which gene expression levels were measured in multiple cells. To achieve our goal we started with some cleaning and visualization of our training data. Afterwards we implemented a linear method, called the Logistic Classifier, as well as some non-linear methods, such as Neural Network Classifier, Random Forest Classifier and Simple KNN Classifier.

Visualization and Cleaning Steps

As the training and test datasets have significant dimensions (5000 rows and 32 285 columns), we directly considered cleaning it to better identify potential clusters later on and reduce our code's running time.
First, we compared the input variables' dimensions of the training and test sets and noticed they were identical so we did not remove any column during this step. Regarding the handling of missing data, we decided to drop all the rows with missing values. Even if this command did not suppress any data, we decided to keep it as a precaution.
We then computed the standard deviation of each column of the training dataset and stored the indices of the columns having a standard deviation equal to zero. This indices' storing allowed us to remove later on the same columns in the test dataset as in the training dataset. Furthermore, we identified the pairs of correlated columns in the training dataset and chose to keep only one column of each pair. For the same reason as the previous cleaning step, we stored the indices of the deleted columns.
Afterwards, we observed by fitting a PCA machine on the training dataset that nearly all the variance of the data could be explained using only 4508 PCA predictors instead of using the initial ones. Therefore, we transformed both datasets using a PCA machine with a maximum out-dimension of 5000 to proceed to denoising. The models all use this cleaned data as it produced either the same or better results when testing and allowed a much faster hyperparameters' tuning (except with the Simple KNN Classifier).
To visualize our data, we went for a PCA plot in 2D that could show us potential clusters in the data. Unfortunately, one cannot distinguish clear clusters. Additionally, we projected our data using PCA onto the plane spanned by the directions along the largest variance of the data and visualized those directions through loading vectors. However, as we have significant data, this method did not display any clear results that we could exploit.

Linear Methods

Logistic classification

After finishing the data cleaning steps and visualization, we decided to start with the simplest model: Logistic Classification. We only tested Lasso regularization and we decided to focus on tuning the hyperparameter lambda. We proceeded with small intervals between 0 and 1 because of the long running time tuning implied. We finally managed to find an optimal value of 9.501e-9 for lambda. Our highest AUC of **0.91585** was achieved with the Logistic Classifier when using Lasso regularization.

Non-Linear Methods

Random Forest Classifier
We chose to implement a Random Forest classifier method with tuning. For the hyperparameter tuning we used the XGBoostClassifier() method. We proceeded to the tuning of the eta, max_depth and min_child_weight hyperparameters to optimize our data's fitting. To reduce running time, we proceeded to tuning with a PCA denoised training data with only 100 predictors. We finally settled for the eta at 0.1, max_depth at 6 and min_child_weight at 0.5. The highest AUC we obtained with this method was **0.84304**.

K-Nearest Neighbors
The second nonlinear method we chose to explore was KNN Classifier as it is a simple method that can produce surprisingly good results at the cost of taking a long time for predictions. Unfortunately, we rapidly realized that our data being significant, running a such method was too time consuming. Thus, we decided to use the SimpleKNNClassifier() defined manually by a code that we were given. After tuning the hyperparameter K, we found an optimal value of 4. With this value for K, we reached the highest AUC of **0.63754** with the SimpleKNNClassifier().

Neural Network
We also created a Neural Network, as it produces highly accurate predictions. We did not want to try to tune the parameters as it was too time and resources consuming. We tested many values manually for several parameters such as n_hidden, batch_size or epochs to better fit our data while saving time. The highest AUC we achieved with this method is **0.88834**.

Conclusion

In closing, the highest AUC we achieved was 0.91585, thanks to the L1 regularization linear method. The second highest AUC was obtained with the Neural Network method, with a score of 0.88834. Thus, we decided to keep those two methods for our graded submission in the Kaggle competition.
In the future, focusing on the visualization of the data could be interesting to find promising features for the Logistic Classifier.