Project 5 Analysis - K.D. Gulko

Data

Madelon is an artificial dataset containing data points grouped in 32 clusters placed on the vertices of a five dimensional hypercube and randomly labeled +1 or -1. The five dimensions constitute 5 informative features. 15 linear combinations of those features were added to form a set of 20 (redundant) informative features. A number of distractor features called 'probes' having no predictive power were added. The order of the features and patterns were randomized.

Problem Statement

For this unsupervised learning problem, my task is to test and select the best model to reduce noise and determine the most salient features in this binary classification dataset.

Solution Statement

I will develop a binary classification model and attempt to augment its performance using several feature selection techniques to find the best model.

Metric

I will be using accuracy by comparing coefficients and test scores.

Benchmark

I will use as a benchmark the mean accuracy from a naive logistic regression with a C value of 1000. This model had a 52% accuracy.

Steps and Evaluation

In step 1-benchmarking, I loaded in the Madelon dataset, performed a train-test split, fit the training data with a standard scalar, then transformed the training data and the test data. After which, I ran it through a naive logistic regression (and high C of 1000) and returned the train and test scores of 0.504 and 0.787 respectively, providing a benchmark test score. These scores revealed that this model is extremely overfit.

In step 2-identify_features_l1_penalty, I loaded in the Madelon dataset, performed a train-test split, fit the training data with a standard scalar, then transformed the training data and the test data. After which, I ran it through a logistic regression with an L1 penalty. This model successfully identified 13 salient features and the number of co-variables were reduced to 13. The train and test scores a little better and the train scores indicated a better bias variance trade off at .616 and .633 respectively.

In step 3-build model, I loaded in the Madelon dataset, performed a train-test split, fit the training data with a standard scalar, then transformed the training data and the test data. I then fit the training data with a SelectKBest transformer, then transformed the training and test data. 10 features were left after running SelectKBest. After which, I ran a GridsearchCV on Logistic Regression and KNeighborClassifier to determine the best fit for each model type. Although Logistic Regression did provide a score above the benchmark, it was only minimally better with a test score of .526. The KNeighborClassifier GridSearchCV, however, provided a much better train and test score with .84 and .91 scores respectively. These scores also indicated that the model is the best fit while neither over-fitting or under-fitting.

The best fit was created using KNeighborClassifier with a SelectKBest transformer. Although the number of co-variables remaining after running step 2 and step 3 were similar, the actual co-variables were different.