Simran Soin and Mousta Taenchiki

Prof. Linda Sellie

CS-UY 4563

1 December, 2020

Introduction:

The given dataset provides basic patient information (age, sex, blood pressure, cholesterol levels, and sodium to potassium ratio). Given this information, patients are given one of five different possible drugs. Using machine learning, the aim is to classify which drug a patient should receive, given their information, based on the examples given in the dataset. The learning models that can be applied to achieve this are logistic regression, support vector machines (SVM's), and neural networks.

Unsupervised Analysis:

Since there were five different features in addition to the target feature, the best approach to visualizing the data to see if there were any major trends was to graph several univariate plots. The first set of plots depicts each of the features against the target feature (drug type). The second set of plots is a histogram for each feature to see what the distribution is like within the feature. Based on the first set of graphs, blood pressure, sodium to potassium, and cholesterol showed to have distinct patterns per drug classification. Therefore, they were the chosen features for individual clustering. Using the KMeans clustering algorithm, three models were trained to predict the five drug classifications. The three models were trained using age in addition to blood pressure, cholesterol, and sodium to potassium ratio respectively. When compared to the actual drug type clusters, blood pressure and cholesterol proved to be fairly inaccurate features for classification. However, the sodium to potassium ratio had a much higher accuracy in predicting

general trends for drug classification. Combined with another feature (such as age, in this model), sodium to potassium ratio could prove to be an influential feature in determining classification. One thing to note might be that the dataset provides five independent variables, which could potentially lead to overfitting. Selecting more influential features — such as sodium to potassium ratio — might lead to better results for predicting general trends.

Supervised Analysis:

Logistic regression with L1 (lasso) regularization gave a maximum testing accuracy of 0.4 when c = 0.0001. Logistic regression with L2 (ridge) regularization gave a maximum testing accuracy of 0.4 when c = 0.0001. Logistic Regression using polynomial feature transformation with L1 (lasso) regularization gave a maximum testing accuracy of 0.4 when c = 0.0001. Logistic Regression using polynomial feature transformation with L2 (ridge) regularization gave a maximum testing accuracy of 0.4 when c = 0.0001. SVM's using a linear kernel gave a maximum testing accuracy of 0.978571 when c = 10. SVM's using a radial basis function (RBF) kernel gave a maximum testing accuracy of 0.707143 when c = 10. SVM's using a polynomial kernel gave a maximum testing accuracy of 0.500000 for all values of c. Neural networks gave a maximum testing accuracy of 0.98 when $\lambda = 0.001$.

Table of Results:

Model	**Training Accuracy**	**Testing Accuracy**
Logistic Regression with Lasso Regularization	c= 0.0001 0.47333 c= 0.001 0.47333 c= 0.01	c= 0.0001 0.4 c= 0.001 0.4 c= 0.01

	0.47333 c= 0.1 0.375733 c= 1 0.319467 c= 10 0.316444	0.4 c= 0.1 0.346 c= 1 0.27 c= 10 0.2756
Logistic Regression with Ridge Regularization	c= 0.0001 0.47333 c= 0.001 0.47333 c= 0.01 0.453422 c= 0.1 0.357244 c= 1 0.324889 c= 10 0.319289	c= 0.0001 0.4 c= 0.001 0.4 c= 0.01 0.384 c= 0.1 0.3116 c= 1 0.288 c= 10 0.27
Logistic Regression with Polynomial Feature Transformation (Lasso)	c= 0.0001 0.47333 c= 0.001 0.47333 c= 0.01 0.47333 c= 0.1 0.356089 c= 1 0.316444 c= 10 0.316444	c= 0.0001 0.4 c= 0.001 0.4 c= 0.01 0.4 c= 0.1 0.3292 c= 1 0.2812 c= 10 0.2756

Logistic Regression with Polynomial Feature Transformation (Ridge)	c= 0.0001 0.47333 c= 0.001 0.47333 c= 0.01 0.434933 c= 0.1 0.320756 c= 1 0.317867 c= 10 0.316444	c= 0.0001 0.4 c= 0.001 0.4 c= 0.01 0.374 c= 0.1 0.2976 c= 1 0.2756 c= 10 0.2756
SVM's using Linear Kernel	c = 0.0001 0.483333 c = 0.001 0.750000 c = 0.01 0.750000 c = 0.1 0.783333 c = 1 1.000000 c = 10 1.000000	c = 0.0001 0.442857 c = 0.001 0.657143 c = 0.01 0.692857 c = 0.1 0.685714 c = 1 0.950000 c = 10 0.978571
SVM's using Radial Basis Function (RBF) Kernel	c = 0.0001 0.483333 c = 0.001 0.483333 c = 0.01 0.483333 c = 0.1	c = 0.0001 0.442857 c = 0.001 0.442857 c = 0.01 0.442857 c = 0.1

	0.483333 c = 1 0.633333 c = 10 0.766667	0.442857 c = 1 0.600000 c = 10 0.707143
SVM's using Polynomial Kernel	Range: c = 0.00001 to c = .001 0.633333 0.633333 0.633333 0.633333 0.633333 0.633333	Range: c = 0.00001 to c = .001 0.500000 0.500000 0.500000 0.500000 0.500000 0.500000
Neural Networks	$\lambda = 0.001$ 1.0 $\lambda = 0.01$ 1.0 $\lambda = 0.1$ 1.0 $\lambda = 1$ 1.0 $\lambda = 10$ $0.926\overline{6}$	$\lambda = 0.001$ 0.98 $\lambda = 0.01$ 0.98 $\lambda = 0.1$ 0.98 $\lambda = 1$ 0.92 $\lambda = 10$ 0.74

Why:

Overall, looking at the results from each of the models, one general trend is the similarities between training and testing accuracies. Although the actual accuracy might differ between models, the lack of discrepancy between training and testing indicates low variance that leads to similar results in both subsets of the data.

Comparing each of the models, logistic regression seems to have the lowest accuracies, regardless of c values or regularization techniques. Therefore the model likely has high bias and is underfitting the dataset.

Looking at SVM's using a polynomial kernel, we got the same accuracy over the whole range of c values used. Contrasting the linear kernel SVM with the radial basis function kernel SVM, both had higher accuracies with large c values. The c parameter in SVM's indicates how large the margin is (a lower c value indicates a larger margin, a lower variance, and higher bias, while a larger c value indicates a smaller margin and a higher variance, and consequently a higher chance for overfitting). Given this information, we can conclude that the margin is likely small and that the model is overfitting.

Since neural networks are fit entirely on the given datasets, with smaller datasets or too many iterations, they are more likely to overfit. Although the neural network yielded the highest accuracy out of all of the models tested, it is unlikely to hold true for the greater population. This is especially true since our given dataset was only 200 rows. While the SVM might overfit for large c parameters, this is still less likely to overfit than a neural network because the hyperplane does not adapt to individual data points, in the way that the neural network does. Therefore the SVM with a linear kernel would appear to be most promising in predicting drug classifications for the greater population.