Aadithya Sekar
Muhammad Shabbeer Ghauri
BIOL 419
Course Project Final Report

Abstract:

The nature of our course project was an investigation into the nature of classifying categorical response data by means of using a panel of numerical data. Specifically, the categorical response/dependent variable involved was "mouse class", with each of the eight classes corresponding to a combination of trisomy 21 or control physiological condition, either administered with the pharmacological learning rescue drug memantine or administered with saline, and either subject to one of two forms of contextual fear conditioning. The independent, numerical variable that was used to classify/discriminate between the eight classes of mice consisted of protein expression data for 77 proteins of the murine cerebral cortex. The entire dataset consisted of 1080 mice samples (1080 rows). With respect to the mice classes more specifically, the eight classes are designated as c-SC-m, c-SC-s, c-CS-m, c-CS-s, t-SC-m, t-SC-s, t-CS-m, and t-CS-s. The letter "c" / "t" indicates control vs. trisomy 21, "CS" / "SC" indicates context-shock vs shock-context, and "s" / "m" indicates saline-administered vs menantine-administered.

Specifically, we sought to find an optimal classifier model for the classification of mice based on their cerebral protein expression. The models were chosen primarily with respect to their suitability to categorical response data, and were manipulated with different parameters in order to more thoroughly compare the classifier models to get a more rigorous cross-validated accuracy.

Following this, we sought to answer which sets of proteins within the dataset are most "discriminant" between the different classes. For this purpose, two methods of feature selection were utilized--Recursive Feature Elimination (RFE) and Extra Trees Classifier (ETC). The nature and utility of these two methods is explained in greater detail in the "Methods" section below.

The most successful classifier model, SVM, had a cross-validated accuracy of ~0.8 after several rounds of manipulation of the parameters that feed into the model.

In addition to this, it was possible to narrow down the panel of 77 proteins to a set of 11 that are the most important for classification of the 8 classes of mice. The significance of this particular number, and the particular proteins involved is discussed in the "Results" section below.

Introduction:

Trisomy 21, primarily referred to as Down Syndrome (DS) is a genetic disorder caused by the presence of a third chromosome 21. It typically manifests as physical growth delays, facial feature changes and mild to moderate intellectual difficulties. Neuropsychological evaluations, such as certain genres of IQ tests, reveal a young adult's learning ability comparative to that of a 8- or 9-year-old child, of course varying widely. In order to further explore the neurophysiology of learning difficulty associated with DS, the mouse is often used as a model organism. A rather widely used means of understanding learning in mice is context fear conditioning, in which the subject is either first exposed to a stimulus, then a shock, or a shock first, and then a stimulus (the former is noted as "CS", and the latter as "SC" in this dataset.) In order to understand the interaction between this learning process, and DS, as well as to improve the quality of drugs used in improving learning, a "learning rescue" drug is used as well. In this paper, we will analyze a dataset that was produced as part of one such study on the "protein dynamics of failed and rescued learning" in order to understand how, if at all, protein expression can be used to accurately differentiate between specific classes of control and test mice, in order to further understand the protein interactions that play a key role in regulating learning in not only mice, but in humans as well.

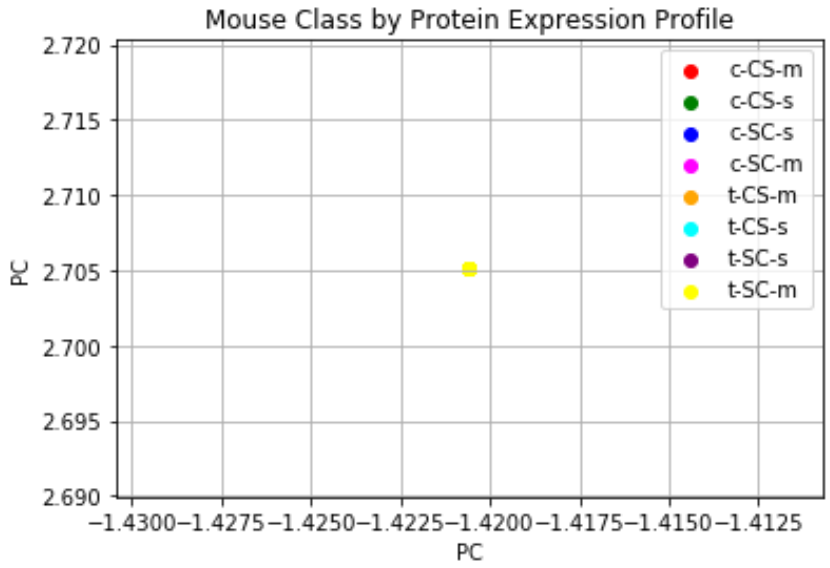
Methods:

In order to pursue the question of modeling, it was first necessary to perform a PCA on the data. Prior to that, the data was rudimentarily visualized using two histograms of two proteins, displaying the frequencies corresponding to their expression levels. As an initial visualization, it was a convenient way of understanding the differential expression of the proteins. Specifically, the histograms each displayed the expression of one protein respectively over the range of all mice samples. In this case, the proteins pPKCG_N and pMTOR_N were chosen for the completeness of their data, as the dataset was incomplete.

Thus, it was necessary to deal with the missing values in the dataset prior to attempting to perform PCA. In order to do this, the dataset was first partitioned into the protein expression values and the mice class values (the entirety of the missing values were found in the protein expression portion of the dataset). The missing values found in the dataset were first replaced with 'NaN's, and subsequently replaced with the mean along the axis. While this allowed us a method to work around the missing values in the dataset, it is possible that it unintentionally introduced bias to the results by introducing "new" values that could have swayed the accuracy of classification in one direction or the other.

Following this, PCA was performed with the protein expression dataset after processing for the missing 'NaN' values (in the code, this is displayed as 'processed_protein'). A plot displaying the change in "Fraction Variance Explained" vs "rth component" was made with an approximate value of 90% of the variance in the data (protein expression data) explained by the first 6 components.

However, upon attempting to graph the mouse class data on the dimensions of any two PCs, an error was incurred in which only one of the 1080 given mice sample points was plotted. The cause of this error is not known; curiously, the only point that is plotted is uniformly one sample of the t-SC-m type of mice (trisomy 21, shock-context, menantine-administered). The reason for this peculiarity is also unknown.



In order to allow for cross-validated classifier models, a test group and train group were defined, with parameters of the sample size, N, and the fraction of the sample that is designated as test (this technique was sourced from HW #5). A value of 0.3 was used for the initial cross validated comparison. After a function was made to calculate the cross validated accuracy of any model, with the parameters of model type and specifications, explanatory dataset, response dataset, the number of principal components, the fraction that is reserved for test, and the number of repetitions, it was run for an arbitrary set of values for those parameters, and then run through a *for loop* in order to attain more comprehensive comparisons. The set of bar graphs comparing the cross-validated accuracies of the different models is shown below.

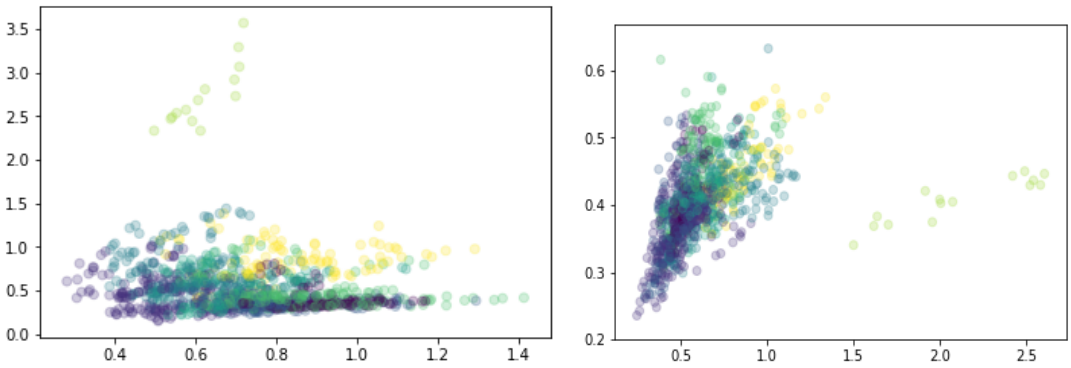
After performing the above cross-validated comparisons of a groups of classifier models at a time, it was necessary to find the subset of the protein data that was most important for the task of discriminating between the 8 classes of mice. Thus feature selection was done.

Initially, LASSO was attempted as that was the recommended method to use for the purpose of feature selection; however, realizing that this method would not work for a categorical response variable like ours (i.e., class of mice), it was necessary to use a method that worked for a dataset containing a categorical response variable. For this purpose, logistic regression was used as a model within the method of RFE. This was then followed by performance of ETC. RFE involves the successive removal of attributes and the building of models based on those attributes that are not removed. In this way, RFE essentially assesses which features (in the case of this dataset, proteins) are most important in yielding the response

variable (mice class)--this method provides a ranking of the 77 proteins by importance. ETC provides the weighted importance of each protein, and thus is provided as a fraction.

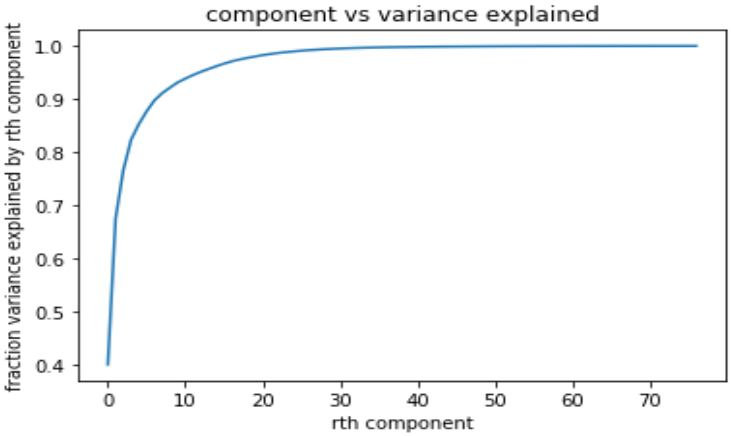
In the above RFE method, the 11 most important proteins were chosen in order to compare with the conclusions of a research group that identified 11 proteins that were discriminant between the mouse classes in the same dataset (http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0129126). However, the method used by the research team involved the creation of a Self-Organizing Map (SOM), or an artificial neural network. Thus the results of our different learning methods would be compared.

After ranking the proteins in terms of their "relative importance", we desired to see if the protein expression data itself would cluster into groups. With respect to the protein expression data itself, there is no outright "ground truth" as it were--thus, we sought to conduct unsupervised learning on the set of protein expression data by means of clustering, as our response (dependent) variable is still categorical. For this purpose, the unsupervised learning method of clustering, specifically k-means clustering was utilized. K-means clustering was done as we so that we might be able to visually see that the expression values themselves cluster into 8 clusters, corresponding to each of the eight mice classes. However, this was unsuccessful in producing distinct clusters, as may be seen in the images below for proteins indexed 15 and 10 (left) and proteins indexed 1 and 30 (right).

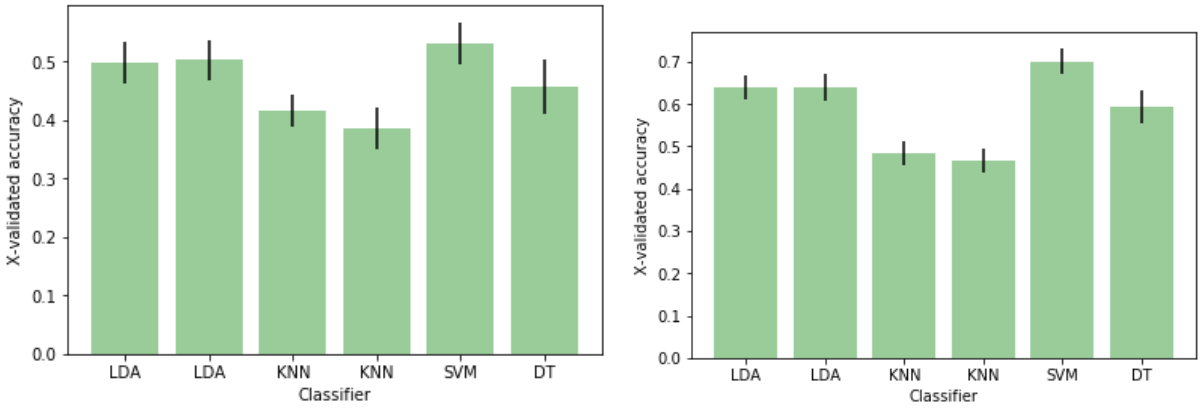


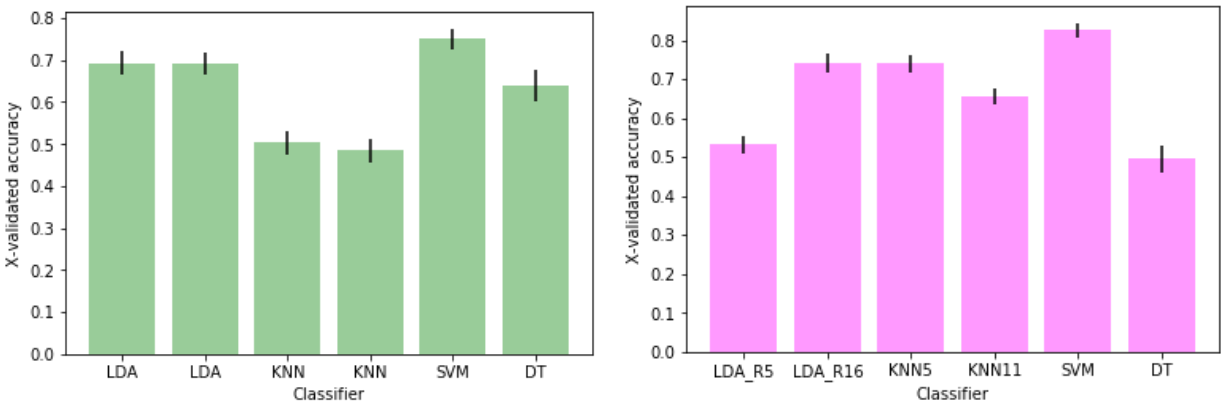
Results and Conclusions:

As mentioned above, the "Fraction Variance Explained" vs "rth component" plot gave a small value for the number of components, 6, necessary for 90% of the data to be explained. However, this result of the PCA did not translate properly to plotting of the mouse class data in two dimensions using the principal components as the axes. The reason for this remains unknown.



With respect to the results of the RFE and ETC methods used to find the most "important" proteins in differentiating between the different classes, the 11 proteins found were given indexed in the array processed_protein as follows: "[1, 10, 15, 28, 30, 32, 42, 48, 65, 73, 76]". Using their point of incidence in the *for loop*, (i.e., the earlier it appears in the *for loop*, the more important the protein), they are ranked from most important to least important as follows: [(32-SODI-N), (48-S6_N), (1-ITSN1_N), (42-pP70S6_N), (65-Ubiquitin_N), (76-CaNA_N), (15-pNR2A_N), (30-APP_N), (10-pERK_N), (73-H3AcK18_N), (28-TRKA_N)]. The numbers preceding the hyphens indicate the index at which these proteins are found in the processed_protein array, as mentioned above. The 11 proteins that were found to be "discriminant" in the study utilizing SOM are BRAF, CaNA, CDK5, DYRK1A, GFAP, ITSN1, pERK, pGSK3B, pNUMB, S6, and SOD1. Thus, in total, 5 proteins are shared between the two classification/learning methods, while 6 proteins are not. This likely corresponds to a discrepancy in the learning method itself.





[Arrangement of bar charts: top left, i=3; top right, i=12; bottom left, i=15;, bottom left, i=16]

As shown in 3 of the bar charts provided (those that were created by a *for loop*, in green), the accuracy of all the models gradually increased with increasing value of the parameters used (the bar charts were plotted with the use of a *for loop* that incremented by 1 and had a range from 1 to 16, non-inclusive). In each of the total 15 bar charts that were plotted, however, the mean cross-validated accuracy of the linear SVM model was greater than that of the consistently next-highest model by accuracy, LDA (which is only once lower than KNN, see bar chart in magenta), perhaps indicating that linear SVM is more consistently of a higher cross-validated accuracy as a model than the other models over a series of changing parameters.

Contributions:

Both authors contributed equally to different aspects of the completed project.

References:

http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0119491

https://en.wikipedia.org/wiki/Down_syndrome

https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Imputer.html

http://blog.datadive.net/selecting-good-features-part-iv-stability-selection-rfe-and-everything-side-by-side/

https://www.ncbi.nlm.nih.gov/books/NBK5223/

http://machinelearningmastery.com/feature-selection-in-python-with-scikit-learn/

http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html