Machine Learning Final Project Report
: Predictive Drug Neurotoxicity Screening using RNA-seq data and Machine Learning Models Daniela Quijano

Contents

Abstract

The goal of this project was to be able to predict drug neurotoxicity using RNA Seq data that came from the published study: 'Human pluripotent stem cell-derived neural constructs for predicting neural toxicity' by Schwartz et. Al. This project represented a possible path to assessing the safety of a drug in vitro using neural constructs and RNA seq data. In the paper neural constructs were built to mimic brain tissue that would then be exposed to an array of known toxic and non-toxic chemicals. After 2 and 7 days of chemical exposure Bulk RNA-Seq was conducted in order to assess the types of transcriptional changes that occurred as a result of the toxic chemical exposure.

In order to obtain the data for the project three approaches were taken: GeoQuery database, Recount3 database and e-mailing the authors of the paper. The authors kindly provided the raw data that could be used as input to DESeq Bioconductor package that was used for differential gene expression (DGE) analysis.

The goals for this project included using RNA-seq data from this study to replicate certain aspects of the paper while adding some of my own analysis. Support Vector Machines (SVM) models were used to classify compounds as toxic or non-toxic based on the gene expression data of the 3D neural construct after being exposed to a known toxic or nontoxic chemical. In addition to SVM, K-nearest neighbors (KNN), decision trees (DT) and random forest algorithms (RF) were implemented. Hierarchical Clustering and Dimensionality Analysis in the form of PCA and TSNE were implemented to observe how the data clustered. Lastly, leave one out (LOO) cross validation and K-Fold cross validation were implemented to test different testing and training splits to confirm that the model performance was not an artifact of a random sampling for testing and splitting the data. ROC Curves were also implemented to explore model performance which was 67%, 75%, 58%, and 60% for SVM, KNN, DT and RF models respectively. I was not able to achieve the 90% prediction accuracy that was obtained by Schwartz et. al when implementing SVM models likely due to insufficient hyperparameter tuning.

In general, this work has a lot of potential in the area of diagnostics given that the first two steps of the drug development process are time intensive and essential in lowering risks once drugs are tested in human subjects on step 3 of the process. All of the goals that were set for this project were achieved.

Process & Methods

Section A: Overall Workflow summary

The overall finalized process that was followed for the successful completion of the project is summarized below. This process was very similar to the process that was outlined in the project proposal with some minor changes.

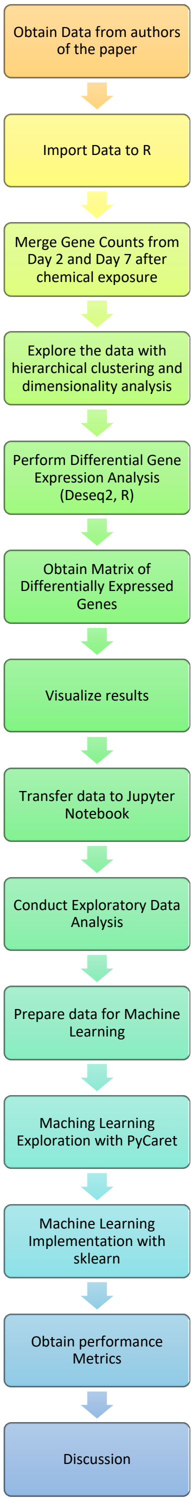

Obtain Data from authors of the paper
Import Data to R
Merge Gene Counts from Day 2 and Day 7 after chemical exposure
Explore the data with hierarchical clustering and dimensionality analysis
Perform Differential Gene Expression Analysis (Deseq2, R)
Obtain Matrix of Differentially Expressed Genes
Visualize results
Transfer data to Jupyter Notebook
Conduct Exploratory Data Analysis
Prepare data for Machine Learning
Maching Learning Exploration with PyCaret
Machine Learning Implementation with sklearn
Obtain performance Metrics
Discussion

Fig 1. Workflow followed in my analysis

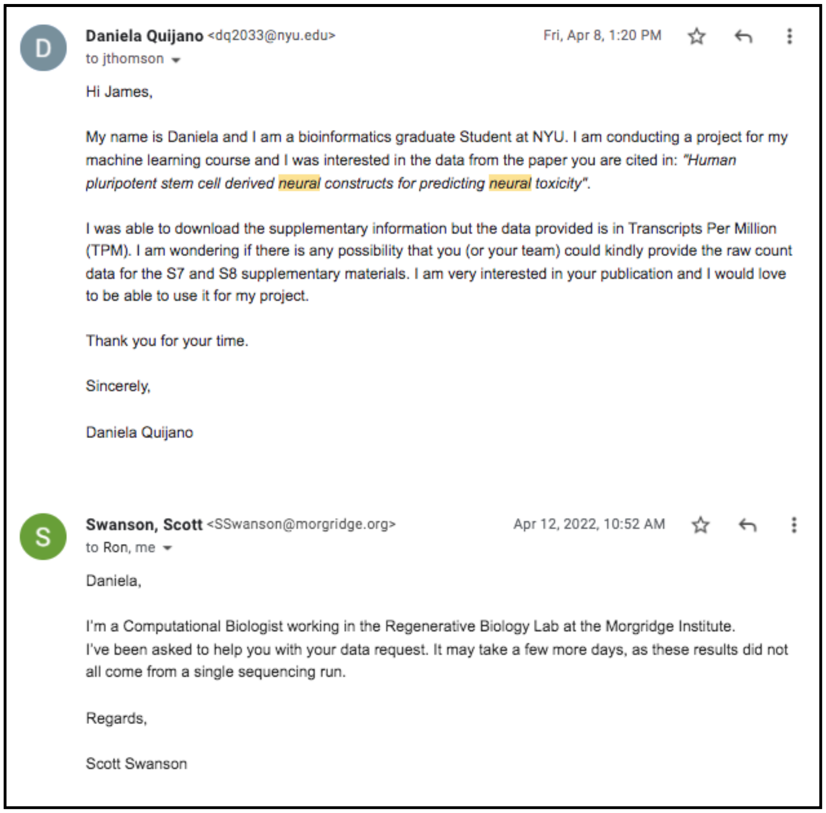
Section B: Data Acquisition & Clean-up

The goal of my data acquisition was to obtain a set of two matrices that represented the gene counts for 140 neural constructs from day 2 and day 7 after exposure to chemicals that are known to be toxic and non-toxic. Day 2 and day 7 count matrices were averaged in order to obtain an average gene expression across the two days for each sample.

There were several options for data retrieval to make sure I was able to get the data for the project: 1. Obtain the data from GEOQuery, 2 Obtain the data from Recount3, 3. E-mail the authors of the paper asking for the raw data. The initial barrier in the execution of the project was the data acquisition.

GEOQuery is described as 'the bridge between GEO and BioConductor.'(4). I was able to download the samples for day 2 and 7 using a combination GEO Accession ID and the SRA number and successfully downloaded the data. I realized there was a problem at the end of this process. The data available on GEO was in Transcripts Per Million (TPM)-a metric to describe the abundance of reads that map to the reference genome. This was a problem because DESeq2 does NOT work with TPM data and MUST get the raw count data. Although I ended up not using this approach, I am still attaching the GEO data acquisition process to the appendix files in this report.

The data from the paper was found in the Recount3 repository. Recount3 stores ' summaries and queries for large-scale RNA-seq expression and splicing.'(9). Using the recount3 bioconductor package I was able to download the raw data of interest but I did not use this data in the end. The last option I had was e-mailing the authors of the paper which was what allowed me to proceed with my project. As seen in Fig.1 below, I emailed the authors of the paper and they kindly provided me with the raw data a couple of days later. I decided to use the data the authors provided. Even though the data the authors provided looked very similar to the data I was able to obtain from recount3, I wanted to be safe and make sure the data I was working with was of the highest quality possible. Both the GEOQuery and Recount3 data retrieval and processing attempts can be found in the attached appendix. Once I was able to combine the day 2 and day 7 count matrices into one representing the gene expression from each day, I was able to begin preliminary exploratory data analysis for downstream processing.



Daniela Quijano <dq2033@nyu.edu> Fri, Apr 8, 1:20 PM
to jthomson

Hi James,

My name is Daniela and I am a bioinformatics graduate Student at NYU. I am conducting a project for my machine learning course and I was interested in the data from the paper you are cited in: *"Human pluripotent stem cell derived neural constructs for predicting neural toxicity"*.

I was able to download the supplementary information but the data provided is in Transcripts Per Million (TPM). I am wondering if there is any possibility that you (or your team) could kindly provide the raw count data for the S7 and S8 supplementary materials. I am very interested in your publication and I would love to be able to use it for my project.

Thank you for your time.

Sincerely,

Daniela Quijano

Swanson, Scott <SSwanson@morgridge.org> Apr 12, 2022, 10:52 AM
to Ron, me

Daniela,

I'm a Computational Biologist working in the Regenerative Biology Lab at the Morgridge Institute.
I've been asked to help you with your data request. It may take a few more days, as these results did not all come from a single sequencing run.

Regards,

Scott Swanson

Fig. 2 Emails showing the exchange with the authors of the paper to obtain raw data.

Results & Discussion

Section A: Differential Gene Expression Analysis

The next step in my data pre-processing was the differential gene expression analysis. The package DESeq2 by Michael Love et. Al was used in order to process the raw counts. DESeq2 relies on binomial generalized linear models and dispersion estimates in order to generate statistical analysis reports on genes that are significantly expressed when performing comparisons between sample groups. The number of genes in the raw count matrix represent the number of statistical tests that were performed. DESEq2 relies on the wald test to output the likelihood that the difference in expression between genes in different groups is statistically significant. The design matrix in this case was very simple: a matrix of two columns where row names were sample names and a column titled "condition" indicated whether a sample was toxic or nontoxic. This design matrix was used for the DESeq2 analysis to be able to compare the dispersions and log fold changes of genes in each group. The final product from the differential gene expression analysis was a list of 118 differentially expressed genes amongst the 140 samples.

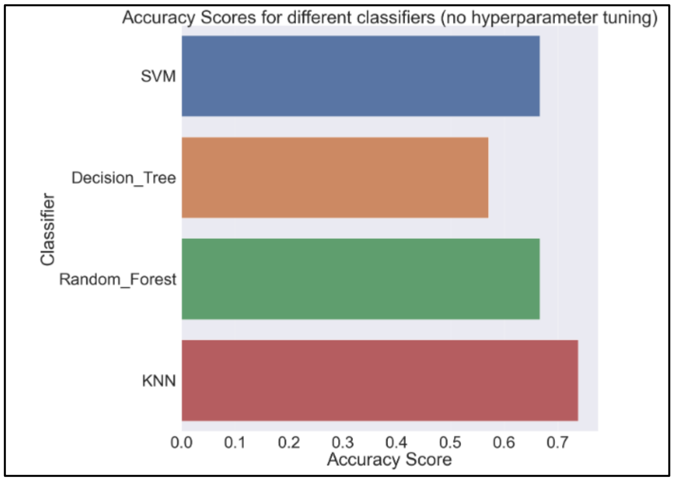


Fig 3. Performance Summary of Machine Learning Classifiers that were implemented in my analysis.

Section B: Machine Learning Discussion

The task of binning RNA-seq data as having originally come from a compound that was exposed to a toxic chemical or not is a classification problem. The performance of the support vector machines classifier relies on the linear separability of the classes that one is trying to separate. The hyper planes made up of support vectors attempt to maximize the margin between the groups in the data. After implementing cross validation through grid search (GridSearchCV()), the kernel that best fit the data was the radial basis function or simply, 'rbf' kernel. A large factor in the performance of an SVM classifier is the kernel that is used since the success of SVM relies on the ability of the hyperplanes to separate features. The kernel trick can be described as 'a method where a Non Linear data is projected onto a higher dimension space so as to make it easier to classify the data where it could be linearly divided by a plane.'(10).

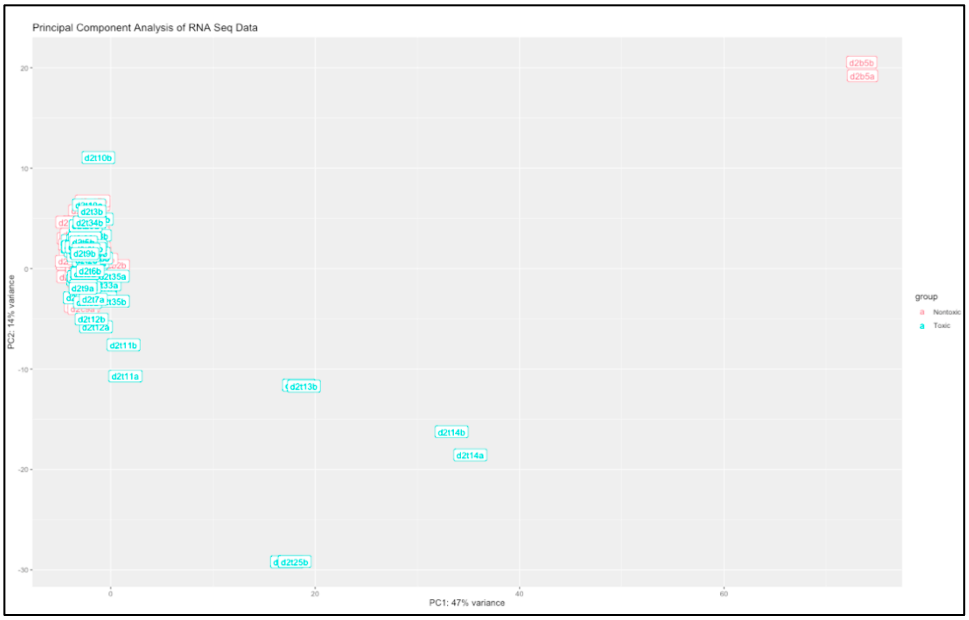


Fig 4. Principal Component analysis. This PCA shows that there are a few samples that did not cluster as expected (see samples on top right)

The ROC curve that was produced from the SVM implementation in my analysis looks similar to the curve that was

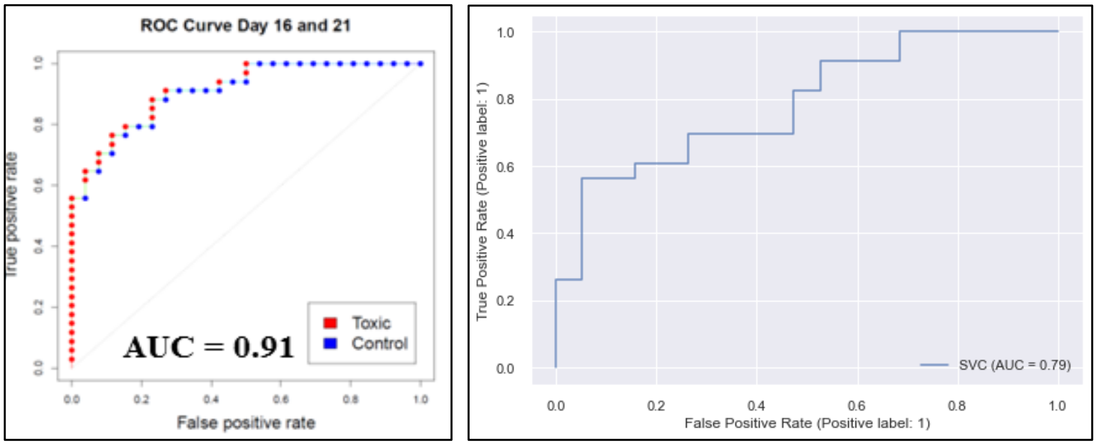


Fig 5. ROC Curve from Schwartz. Et. al paper (data source) and ROC curve produced by my own analysis

produced by Schwartz et. Al (Fig. 4). My curve's false positive rate increases with a higher rate than the true positive rate when compared to the ROC curve produced by the study. In general this makes sense because my classifier did not perform as well as the classifier in the study. The best performance of my SVM after hyperparameter tuning was 67% as opposed to the 91% accuracy reported in the paper. According to literature reviews such as 'Applications of Support Vector Machine (SVM) Learning in Cancer Genomics' by Huang et. al, 'Compared to the other ML methods SVM is very powerful at recognizing subtle patterns in complex datasets' .Many papers are also referenced with extremely successful (80%+ accuracy) SVM applications for drug classification and cancer detection. In my case further tuning could be a factor that could be tweaked to help the performance of the model.

As mentioned above, the support vector machines classifier relies on the separability of the two classes. In this case, the PCA that was run through the Deseq2 package did not show ideal separation between the two classes. To make sure that perhaps it was a limitation of viewing the data in two dimensions instead of 3, a 3D t-distributed stochastic neighbor embedding (**t-SNE**) was produced with the PyCaret library. The TSNE confirmed that the data did not show clear clusters between the neural constructs that were exposed to a toxic chemical and those that were exposed to a non-toxic chemical. There could be several reasons for this. First, I am unable to speak about the pre-processing of the data that occurred in order to obtain the gene count matrices that were used to perform the analysis in this project. When RNA Seq data is analyzed, the general workflow includes raw fastq file trimming and quality control and alignment to a reference genome with specific parameters. It would have been very helpful if I had access to the files that give metrics about the performance of read trimming and read alignment. Based on my past experience with RNA Seq, it is possible that the dimensionality analysis shows that some of the samples are explained by different principal components from those in the same group due to discrepancies in the number of reads that were mapped or certain qualities of the reads such as GC content or repetitive regions. After consulting with NYU Transcriptomics professor, Dr. Manpreet Katari, there is still a possibility that these reads appear to be different than the others due to "real"

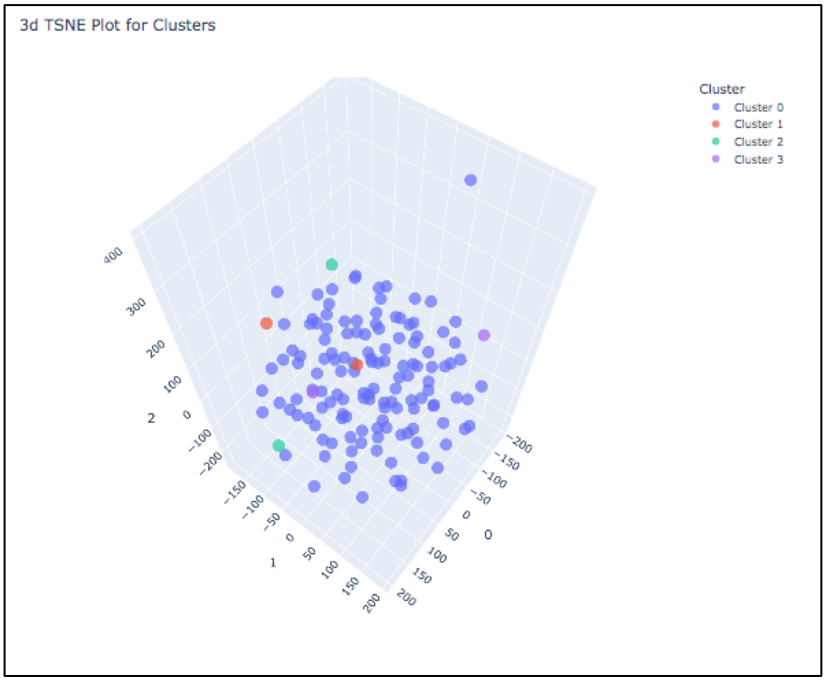


Fig 6. TSNE plot

biological variability. It is important to note the procedure that was conducted to run the PCA analysis. At first, my intention was to use the rlog() transformation. This transformation minimizes the effects of variance with features with low counts- regularizing the data. This regularization allows the PCA to not be dominated by features with low counts.

In order to confirm the fact that several samples had survived normalization procedures, a boxplot of the counts for each sample was produced for each of the groups of samples (toxic/non-toxic samples). These boxplots can be seen in Fig.7. and they show that certain samples do indeed look different from the others.

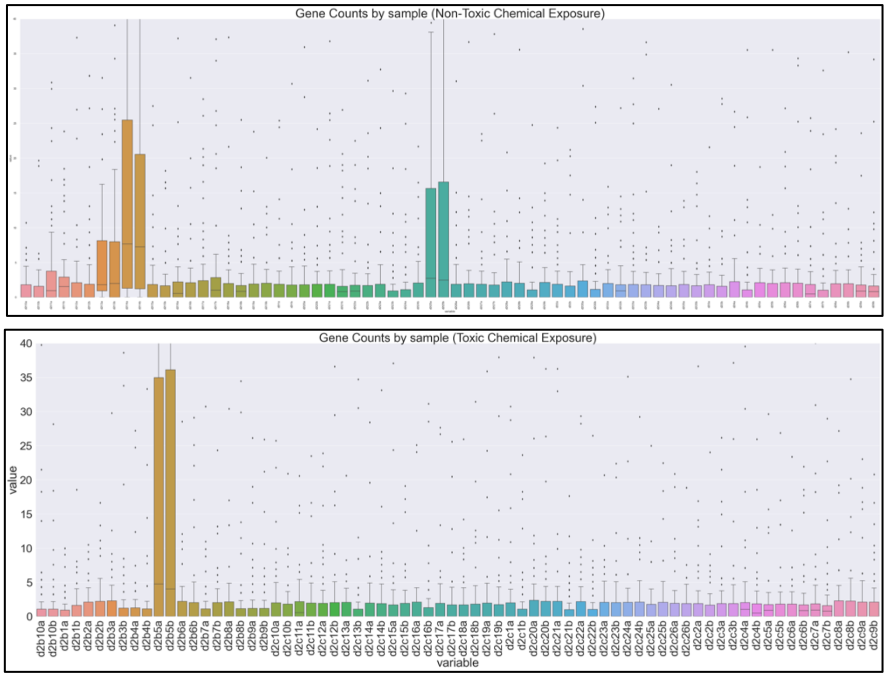


Fig 7. Boxplots showing the spread of the data of each sample. Outliers can be seen here

Due to the large size of this dataset relative to the processing power of my personal computer, I had to use the variability stabilizing transform (vst()) instead. The vst() transformation corrects changes in the variance of the data similar to the rlog() transformation. In general, the vst() transform was better suited in this case because it is less computationally intensive. A limitation of this dataset that is also addressed in the paper is that of the size of the dataset. Although this dataset is not very small at 140 samples, the machine learning analysis can still benefit from the addition of more samples.

Although the support vector machines performance was not as high as the performance reported on Schwartz et. al paper, and the PCA nor hierarchical clustering seemed to not show any separation with respect to the samples, something interesting happened when hierarchical clustering was applied on the genes (features). When plotting the genes and samples on a heatmap with hierarchical clustering, we can see two distinct categories of genes appear representing the toxic and non-toxic sample groups (Fig.8). As seen in the dendogram from the rows (samples) we can see that they do not closely cluster with one another. The fact that the genes do cluster in distinct groups explains why the machine learning models were able

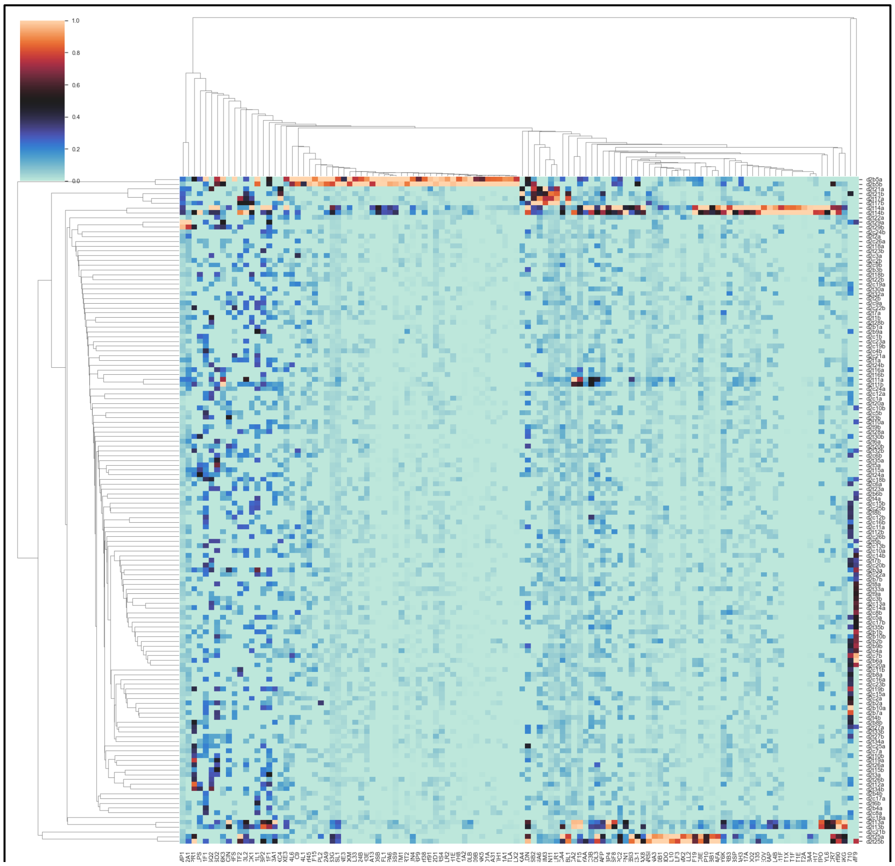


Fig 8. Heatmap showing top differentially expressed genes (columns) and samples (rows). Genes are clearly divided in two groups corresponding to toxic and non toxic chemical-exposed group

to perform well despite samples not clustering together based on labels. After all, the features (genes) are the ones that need to be distinct in order for the model to be able to classify samples correctly.

This project not only trained an SVM model, but also trained and tuned a K-nearest neighbors (KNN) model, a Random Forest (RF) classifier, and a Decision Tree (DT) classifier-all of these are supervised machine learning methods. Overall, the best performing model was the KNN classifier at 75% accuracy. In general I had expected the support vector machines classifier to perform the best given the great accuracy the model had in the Schwartz. Et al paper. Because there were some outlier samples in each group, these may have changed the sample space enough to where the 'nearest neighbors' were a better predictor than the margin created by the support vectors/hyperplanes in SVM. (5) 'The KNN algorithm assumes that similar things exist in close proximity'. In this case the assumption would be that toxic-chemical-exposed samples would exist in proximity to other samples exposed to a toxic chemical and the same holds true for samples exposed to non-toxic chemicals. KNN is able to infer which group a sample or query belongs to by seeing what kind of samples are closest to it in space, or which 'neighbors' are nearest to it. After performing a grid search cross validation, the best parameter for the number of neighbors was n=4 meaning that If I had plotted a plot of k value vs error rate, the error rate would have likely stopped decreasing as much with as n approached larger numbers. The KNN algorithm is versatile and easy to implement but it becomes slower with increasing dataset sizes. In this case, the computational time was comparable to the computational time of SVM, RF and DT. If this study is repeated with larger samples, this will likely become an issue.

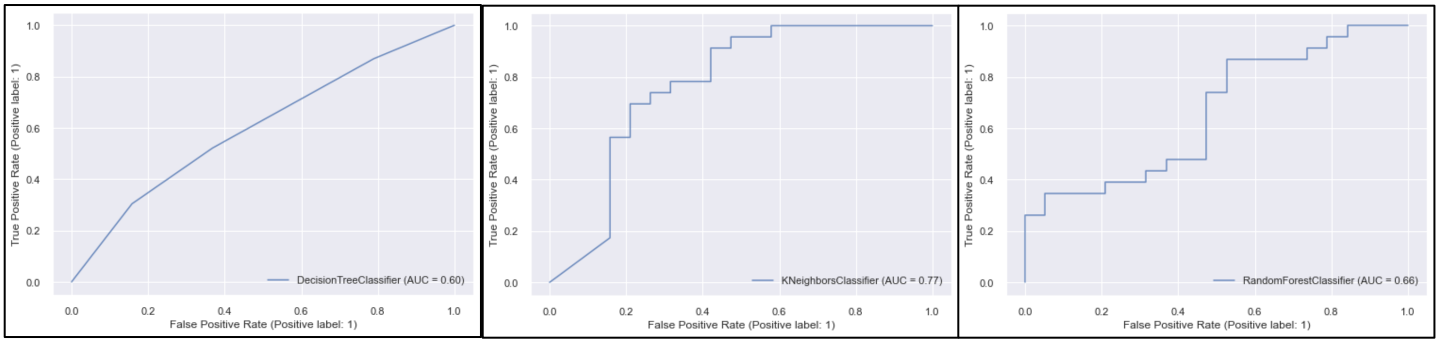


Fig 9. ROC curves showing model performance. From left to right, ROC curve decision tree classifier performance, random forest performance and K nearest Neighbors classifier performance.

The next best performing model was the Random Forest classifier. A RF model is able to work with both classification and regression which made it a suitable model to implement in the classification task in this investigation. RF models fall under ensemble-based methods which are a type of ML models that use other 'base' models to make a decision. 'Rather than splitting at similar features at each node throughout, Random Forest models implement a level of differentiation because each tree will split based on different features' (11). In order to understand random forest models, it is important to understand the concepts of bagging and boosting. 'Bagging is an ensemble algorithm that fits multiple models on different subsets of a training dataset, then combines the predictions from all models.'(1). In boosting, weak learners are used and you ' "focus" on the samples [the model] got wrong. '(8).In the case of the RF classifier used in this analysis, boosting was used. At each iteration the model attempted to correct the 'mistakes' made in the previous iteration. The tuned RF model had a prediction accuracy of 60%. When looking at the ROC curve (Fig. 9), one can see a sharp drop off in true positive rates right around the halfway point of the curve. It is not a very well performing model since there is a 50-50 chance that one can randomly assign a compound as toxic or non-toxic and 60% is not too far from 50%.

The last model that was tested was a decision tree classifier. Decision trees aim to reduce entropy in the data which refers to the randomness in the dataset. Another metric that is used in decision tree classifiers is gini impurity. 'Gini Impurity is the probability of incorrectly classifying a randomly chosen element in the dataset if it were randomly labeled according to the class distribution in the dataset.' (12) After hyperparameter tuning, the best metric was entropy and model performance was quite bad at 58%. Decision trees were likely not a suitable model for this dataset.

The last step in my machine learning implementation was cross validation. Cross allows one to explore how the splitting of data into testing and training sets affects model performance. Like Schwartz. Et. al did in the paper, I implemented Leave One Out cross validation (LOO). LOO cross validation is repeated as many times as instances in the dataset. In this case, there were 140 samples so LOO was repeated 140 times leaving one sample out at a time. I also implemented K-fold cross validation. In k-fold cross validation a 'single parameter called k that refers to the number of groups that a given data sample is to be split into.' (1). The LOO cross validation method outperformed k-fold cross validation when implemented in all of the machine learning models implemented in this analysis.

Conclusion

In general my work achieved most if not all of the objectives that I had planned. I was able to replicate certain aspects of the paper while adding some of my own analysis. SVM models will be used to classify compounds as toxic or non-toxic based on the gene expression with 67% accuracy. In addition to SVM, K-nearest neighbors, decision trees and random forest algorithms were be implemented with 75%, 58%,and 60% accuracy. Also, ROC curves were successfully produced and interpreted to explain the performance of each of the models. Hierarchical Clustering and dimensionality analysis as PCA and TSNE were performed in order to observe relationships within the data. Both LOO and K-Fold cross validation were implemented

It would be interesting to remove the outlier samples and see how this affects the differential gene expression analysis. Based on the preliminary exploration that was conducted with the Pycaret library, it might be interesting to test the performance of other models that had high accuracy scores such as logistic regression. I would like to further explore more visualizations with sklearn and Pycaret to continue to investigate model performance. Knowing that classifiers are able to classify samples based on gene expression, I would like to apply the workflow I created to different datasets and to continue to optimize model performance.

Appendix

1. RSEM_To_DESEQ2.nb.html
    a. This file contains the differential gene expression analysis that was used in order to see which genes differed between the toxic and non-toxic groups.
2. GEOQuery Data Exploration- Neural Constructs.html
    a. This file contains the data acquisition from GEOQuery.
3. Recount3_rawcounts.nb.html
    a. This file contains the procedure that was going to be used to obtain the raw data from Recount3 database.
4. Neurotixicity_Data_Prep_Machine_Learning.ipynb
    a. This notebook contains some exploratory data analysis along with ML applications

Works Cited

1. Brownlee, J. (2020, August 2). *A Gentle Introduction to k-fold Cross-Validation*. Machine Learning Mastery. Retrieved May 12, 2022, from https://machinelearningmastery.com/k-fold-cross-validation/
2. *Exploring transcriptome-wide changes using PCA*. (n.d.). Exploring Transcriptome-Wide Changes Using PCA. Retrieved May 12, 2022, from https://tavareshugo.github.io/data-carpentry-rnaseq/03_rnaseq_pca.html
3. *Exploring gene expression patterns using clustering methods*. (n.d.). Exploring Gene Expression Patterns Using Clustering Methods. Retrieved May 12, 2022, from https://tavareshugo.github.io/data-carpentry-rnaseq/04b_rnaseq_clustering.html
4. *GEOquery*. (n.d.). Bioconductor: GEOQuery. Retrieved May 12, 2022, from https://bioconductor.org/packages/release/bioc/html/GEOquery.html#:%7E:text=GEOquery%20is%20the%20bridge%20between,%2C%20Meltzer%20P%20(2007)
5. Harrison, O. (2019, July 14). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Medium. Retrieved May 12, 2022, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
6. Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2018). Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. *Cancer genomics & proteomics*, *15*(1), 41–51. https://doi.org/10.21873/cgp.20063
7. Love, M. S. I. A. (2022, April 26). *Analyzing RNA-seq data with DESeq2*. Analyzing RNA-Seq Data with DESeq2. Retrieved May 12, 2022, from http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html
8. Raschka, S. (2022b, May 12). *How does the random forest model work? How is it different from bagging and boosting in ensemble models?* Dr. Sebastian Raschka. Retrieved May 12, 2022, from https://sebastianraschka.com/faq/docs/bagging-boosting-rf.html
9. *recount3: uniformly processed RNA-seq*. (n.d.). Recount3 Website. Retrieved May 12, 2022, from http://rna.recount.bio/
10. Team, D. (2020, July 10). *Support Vector Machine Algorithm (SVM) – Understanding Kernel Trick*. DataMites Offical Blog | Resources for Data Science. Retrieved May 10, 2022, from https://datamites.com/blog/support-vector-machine-algorithm-svm-understanding-kernel-trick/#:%7E:text=A%20Kernel%20Trick%20is%20a,Lagrangian%20formula%20using%20Lagrangian%20multipliers.%20(
11. Lutins, E. (2018, June 17). *Ensemble Methods in Machine Learning: What are They and Why Use Them?* Medium. Retrieved May 12, 2022, from https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f
12. *What is difference between Gini Impurity and Entropy in Decision Tree?* (n.d.). Quora. Retrieved May 12, 2022, from https://www.quora.com/What-is-difference-between-Gini-Impurity-and-Entropy-in-Decision-Tree