PREDICITING THE BIODEGRADABILITY OF CHEMICALS FROM QSAR DATA

Ruairi McElhatton, Student Reg No:190134101, ACS6427 - December 11, 2023

Abstract - **In this report, three models made using MATLAB pre-built functions and adjusted using different techniques are analysed on their effectiveness at predicting whether a molecule is biodegradable or not. The three Machine Learning (ML) models are K-Nearest Neighbour (KNN), Support Vector Machine (SVM) and a random tree classifier. The data processing techniques used for each of these are discussed, then the actual model design. Each model is then analysed and a conclusion is reached on what is the best option for this problem.**

1 Introduction

This report outlines the fully validated, predictive model for biodegradability using [methods behind the model] on MATLAB. This is based on the paper [1]. The assignment takes place in an important subject area as the classification of molecules into biodegradable and non-biodegradable can result in much lower amounts of landfill on a global scale. Landfills are where non-biodegradable waste is tossed and they have an adverse effect on the surrounding environment and humans living nearby [2]. Being able to classify whether a molecule is biodegradable or not could become very important as European REACH regulation is trying to reduce use on non-biodegradable molecules. Many tests for this are currently done using animal testing so REACH is encouraging alternatives to this for ethical reasons. Currently, the percentage of molecules with biodegradation data is only 61% [3]. [1] Discusses at length a number of models which could be used for the task of predicting whether a material is biodegradable or not. This project is going to look at some of the models mentioned within that paper and some others, with the hope of improving the accuracy of predictors in order for this to be a plausible option in this area in the future.

In this assignment, I decided to focus on a SVM [4, 5] ML model. I then decided to make basic benchmark comparison models which included a K-Nearest Neighbour (KNN) model [6, 4] and a Random Forest Classifier (RFC) model [7, 4]. I used the referenced websites to help me with designing these models.

2 Data Processing

In this part of the assignment, I used different methods of data processing (DP) to shape the data in a way that increased the accuracy of the final models produced. I tested each method against my models at every decision point. This meant I was testing frequently to see which method was best and which way each method should be implemented.

The first DP technique I utilised was to normalise the data. I left the result column alone in this part as I firstly didn’t see the point in normalising it, and because when I did try to, I ended up with models of lower accuracy across the board. Within normalising the data, I tried a few different methods of normalising but after trying all the available options on MATLAB, it seemed that the default, which calculates z-score, was the best after all, with other methods offering very small increases in accuracy of some methods, but too small to take any notice of. A notable method was the ‘Norm’ method which led to around a 30% decrease in model accuracy for my SVM model.

The next DP method I tried to use was to find and remove outliers in the data. This led to a large drop off in the accuracy of all the models. Upon seeing this, I delved a bit deeper into the data set itself and realised that there would only be ‘true outliers’ which should be left in the data set as they aren’t miscalculated.

The final DP technique I utilised was to up-sample the dataset. I saw that biodegradable molecules were under-represented when compared with non-biodegradable molecules. I used a simple method here of just duplicating the biodegradable data points. This led to a significant increase in accuracy in all three models. This supports ML theory that increasing the size of the dataset and making sure all classes are well-represented leads to an improvement in performance.

3 Methodology

For my SVM model, I used the inbuilt function ‘fitcsvm’ in MATLAB. The first thing to note with this was that I had to use a data set which I had altered less than with the other two to gain optimal accuracy. I still could not get the accuracy to be greater than RFC model but below I demonstrate everything I tried in the aim of doing this.

Initially I just used this by itself but then after becoming acquainted with it, I started to change some of the hyperparameters within to optimise the performance of the model. These included changing the box constraint from a default of 1, which I eventually decided against as performance didn’t improve noticeably after trying a range of values. I then tried all the different prepared ‘Kernel Functions’ [5] within MATLAB, again settling on the default as the best option (linear Kernal function in this case). Despite many papers saying that RBF Kernel is the best choice for practical application. There were some other parameters of the kernel function that I could have changed also but all led to little or no increase in model accuracy.

I was then interested to see that changing the optimisation routine from the default of Sequential Minimal Optimisation (SMO) to Iterative Single Data Algorithm (ISDA) led to a noticeable increase in accuracy of the model. This is due to the dataset and this solver is not necessarily better than SMO. Finally, I created a double for loop to test out a range of different costs for misclassification, both for misclassifying biodegradable and non-biodegradable and vice versa. There should be a higher cost for misclassifying biodegradable as non-biodegradable due to these outweighing the cost of it happening the other way around. However, the disparity between the cost of false positive and false negative is minimal and therefore I decided to go with the costs which have the best accuracy.

The first benchmark model I worked on was the KNN classifier. I cross validated using k-fold method. This meant I initially split the data using holdout validation, making a model out of this, using k-fold cross validation to detect overfitting. It is worth noting that this is also the approach I took with the RFC model as it is a computationally cheap method. I didn’t try too much else with the KNN classifier as it is only a benchmark model, but I think with more time it would be worth experimenting with the different hyperparameters as the default KNN model performs better than the default SVM model.

The second and last benchmark model I tested was the Random Forest model which in MATLAB is called the RFC model. This came out with a great initial accuracy score after cross validation and due to it being a benchmark model I didn’t alter it.

4 Model Analysis

I decided upon an SVM model as the main one for this assignment. This meant I spent more time investigating hyperparameters and adjusting them accordingly. The surface plot which can be seen in Figure 1 shows how the accuracy of the SVM model changes depending on costs. The plot takes the measure of model accuracy instead of AUC and shows us that many combinations of costs lead to a model with very high accuracy, with a trend of similar cost for misclassification of each class leading to the highest levels of accuracy. This is a common trend in most models of different data sets, but I made this plot to ensure that the optimal solution was found. The eventual result was an increase in accuracy of 1% from the default settings. If I had more time on this project, I would have optimised every hyperparameter, eventually ending up with a model which is significantly more accurate due to separate, marginal increases in accuracy. The reason this graph is done for the SVM pre-sampling is because it was initially performing very well but when I corrected the code the performance dropped significantly. This analysis of the respective costs is still beneficial however and could be applied to the SVM built with the up sampled data which would most likely produce different results to optimise that model.

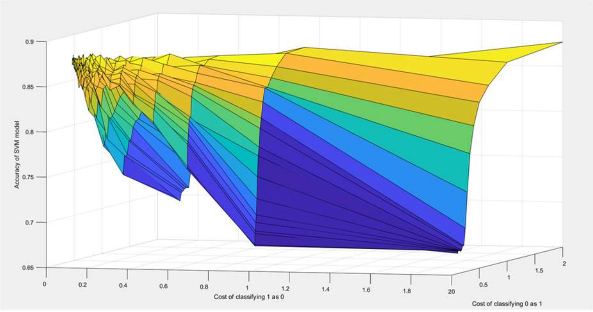
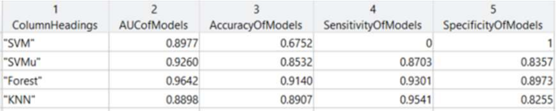


Figure 1 - Surface plot looking at varying cost of misclassification.

In Table 1 we can see some different measures for performance. We can see a notable difference in the AUC and the Accuracy for SVM model. The difference between these two measures is that AUC measures how true positive rate and false positive rates trade off whereas accuracy looks at the proportion of true positives and negatives in the data set.



1	2	3	4	5
ColumnHeadings	AUCofModels	AccuracyOfModels	SensitivityOfModels	SpecificityOfModels
"SVM"	0.8977	0.6752	0	1
"SVMu"	0.9260	0.8532	0.8703	0.8357
"Forest"	0.9642	0.9140	0.9301	0.8973
"KNN"	0.8898	0.8907	0.9541	0.8255

Table 1 - Various measures of each model

In Figure 2 generated by the script we can see how the performances of the four models vary at all classification thresholds in the ROC curve graph. With the RFC model performing best at all points on the graph. The area under this curve is the AUC measure seen in Table 1. As we can see, the RFC works best at almost every point. This RFC wasn't altered at all from the default and still outperforms the other two models. 'SVM' represents the SVM model before it has undergone up sampling. 'SVMu' is post up sample.

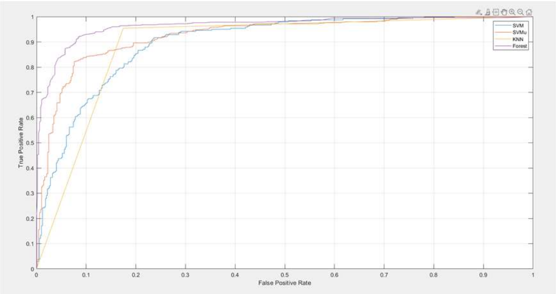


Figure 2 - ROC curve for all models

5 Conclusions and Recommendations

The most noteworthy comment that can be made regarding this project and the basis of all other machine learning projects is that the data-processing stage is absolutely pivotal in affecting performance. It plays the most noticeable part in how accurate a model is. The two stages of pre-processing utilised in this project were essential in making the accuracy of the models as high as they are.

Despite the literature generally saying that SVM models perform well when the number of dimensions is greater than the number of samples, the modified SVM worked very well. The up sampling led to a great improvement in performance. If this project was to be conducted again however, SVM is not the model that should be chosen for this dataset.

The KNN model also performed well, with minimal changes made to the model itself. One of the obvious benefits of KNN is that it is simple to grasp. With more time hyperparameters could have been adjusted to find the optimal number of neighbours and the optimal distance criteria. I'm unsure how much of a difference this would have made.

In conclusion, of the three model types, the one which is recommended for use in future for this dataset and problem is the forest of decision trees classification model. This is because it is the best performing model by all metrics compared to the other two. There is also room for improvement on it too as I didn't optimise it aside from cross-validating it. However, due to this kind of model being black-box, one would have little control over it anyway. Some hyperparameters that could be changed would be the number of 'learning cycles' which means the number of decision trees in the forest so to speak.

6 References

[1] K. Mansouri, T. Ringstead, D. Ballabio, R. Todeschini and V. Consonni, "Quantitative Structure–Activity Relationship Models for Ready Biodegradability of Chemicals," *Journal*

of Chemical Information and Modelling, vol. 53, no. 4, pp. 867-878, 2013.

[2] P. O. Njoku, J. E. Edokpayi and J. O. Odiyo, “Health and Environmental Risks of Residents Living Close to a Landfill: A Case Study of Thohoyandou Landfill, Limpopo Province, South Africa,” *International Journal of Environmental Research and Public Health,* vol. 16, no. 12, pp. 1-27, 2019.

[3] R. Allanou, B. G. Hansen and Y. v. d. Bilt, “Public Availability of Data on EU High Production Volume Chemicals,” *Institute for Health and Consumer Protection - European Chemicals Bureau,* pp. 1-23, 1999.

[4] MathWorks, “Statistics and Machine Learning Toolbox,” MathWorks, 2023. [Online]. Available: https://uk.mathworks.com/help/stats/index.html?s_tid=CRUX_lftnav. [Accessed December 2023].

[5] A. Patle and D. S. Chouhan, “SVM kernal functions for classificiation,” in *IEEE Xplore*, Mumbai, India, 2013.

[6] IBM, “What is the k-nearest neighbours algorithm?,” IBM, [Online]. Available: https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.. [Accessed December 2023].

[7] M. Schonlau and R. Y. Zou, “The random forest algorithm for statistical learning,” *The Stata Journal: Promoting communications on statistics and Stata,* vol. 20, no. 1, pp. 3-29, 2020.

[8] T. Evgeniou and M. Pontil, “Support Vector Machines: Theory and Applications,” in *Center for Biological and Computational Learning, and Artificial Intelligence Laboratory, MIT*, Cambridge, MA 02139, USA, 2001.