Analysis of Quantitative Regression QSAR Models for Predicting Chemical Toxicity

New regulations requiring acute toxicity testing of chemicals towards fish and supports the use of QSAR models for compliance. Pre-curated acute toxicity data for 908 chemicals was analyzed, processed, and modeled in order to develop an accurate and scalable model. Reducing the data via Principal Component Analysis and UMAP showed potential correlations between chemical descriptors and toxicity. These correlations were more pronounced with UMAP, suggesting non-linear interaction between descriptors and toxicity. Multiple regression, regularization, gradient descent, random forest, neural networks, and k nearest neighbor were applied on a training set of 726 chemicals and validated on 182 chemicals in order to predict LC_{50} 96 hours for the fathead minnow. The k nearest neighbor model had the best overall performance on the validation set, preserving nonlinear trends while preventing overfitting. The k nearest neighbor model scales well when adding new instances, improving the accuracy of each cluster. Suggestions were made by the researcher to further explore toxicity, by converting toxicity into a categorical feature using a cutoff value for LC_{50} in order to classify whether a chemical is safe.

INTRODUCTION

With the introduction and enforcement of REACH regulations in 2007 came a surge of interest in computational studies on chemical toxicity [1]. With a requirement of proving products are safe for humans and the environment before introducing them to the market and a goal of ending unnecessary animal testing, REACH promoted alternative testing methods. Quantitative structure–activity relationships (QSAR) emerged as the go-to approach regarding chemical testing. QSAR aims to find functional relationships in chemical structure and their properties.

As part of aquatic testing, REACH requires evaluation of short-term toxicity towards fish on goods imported or manufactured more than 10 tons per year [1]. QSAR modeling provides an economic cost and animal welfare benefit. Past studies utilizing QSAR point to two methods of predicting toxicity—some classifying chemicals based on their mode of action and others by estimating a quantitative parameter [3-5]. The second method is applied at various scales. Some seek to apply the quantitative response techniques to chemicals with similar modes of action [6-8]. The success of this method depends on the ability to identify each chemical's mode of action before applying it to a model, a computationally expensive task. The other scale the method is applied to is a global scale, which takes a large set of chemicals and models them. The heterogenous nature of these chemicals can add noise to the models, decreasing accuracy.

This study focuses utilizing different modelling approaches to model a quantitative response variable (LC_{50}) at a global scale. This aims to address current problems in the literature: poor scaling, small domains of applicability. Past studies have attempted to address the issue of applicability by building complex neural networks [21-25], but due to their initial computational cost and complexity, they scale poorly with small datasets and are prone to overfitting. On the contrary, linear regression algorithms have much less cost and better scaling but tend to underfit the data as they do not consider nonlinear trends [9-20]. Models in this work will be based on six molecular descriptors in order to predict the toxicity.

MATERIALS AND METHODOLOGY

Data

Data is from the QSAR fish toxicity Data Set located within the UCI Machine Learning Repository [2]. The data and is comprised of 928 chemicals, each containing 6 molecular descriptors: MLOGP (molecular properties), CIC0 (information indices), GATS1i (2D autocorrelations), NdssC (atom-type counts), NdsCH (atom-type counts), SM1_Dz(Z) (2D matrix-based descriptors) and a quantitative response variable LC_{50} which is the concentration of a chemical that causes death in 50% of test fish over a duration of 96 hours.

Environment

The analysis and modeling of the data was performed in an anaconda environment containing Python 3.9.7 and the following dependencies: pandas, numpy, scikit-learn, statsmodels, matplotlib, seaborn, tensorflow, and umap [26-33]. Random seed of 101 was set for all stochastic operations in order to maintain consistency between runs.

Analysis

The dataset was loaded via the pandas module and inspected for potential outliers and missing values. A pairplot of the data was produced and any interesting correlations were plotted in higher resolution. Pairplots using the discreet variables NdssC and NdssCH were used to inspect potential collinearity between them and the continuous variables. Correlation matrix was produced in order to investigate potential multicollinearity between the descriptors. Data was then z-score normalized and split into a training set, with 80% of the data, and a validation set with the other 20%. The 6 descriptors were then transformed via Principle Component Analysis (PCA) and UMAP fitted against l1 target metric into 2 dimensions, allowing for visualization of trends in the data.

Modelling

Univariate regression models were developed for each feature in order to provide a baseline for a multivariate model. The first multivariate model had each feature without any consideration for interaction. It was then reduced removing non-significant variables and ANOVA was performed to confirm there was no statistical difference between the models. Then

to test interaction, a model was developed with every first order interaction term. Again, insignificant terms were removed to form a reduced model and ANOVA was performed to confirm a lack of difference. All assumptions of linear regression were checked for the multivariate models. Regularization techniques were utilized on a model with interaction in order to reduce overfitting. Random Forrest was then applied in order to allow the computer to learn nonlinear trends in the data. This algorithm underwent grid search to optimize its hyperparameters.

Parameter	Search Space
n_estimators	10,25,30,50,100,200
max_depth	2,3,5,10,20
min_samples_leaf	5,10,20,50,100,200

FIGURE 1: Search space for Random Forest Optimization

Neural networks were tested to determine the efficacy of a computationally expensive model.

```
Model: "sequential_32"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_167 (Dense)           (None, 64)                448

 dense_168 (Dense)           (None, 64)                4160

 dense_169 (Dense)           (None, 128)               8320

 dense_170 (Dense)           (None, 64)                8256

 dense_171 (Dense)           (None, 32)                2080

 dense_172 (Dense)           (None, 1)                 33

=================================================================
Total params: 23,297
Trainable params: 23,297
Non-trainable params: 0
_________________________________________________________________
```

FIGURE 2: Neural Network Model Summary

Finally, the k-nearest neighbor was optimized using grid search and used to test the efficacy of distance-based modeling.

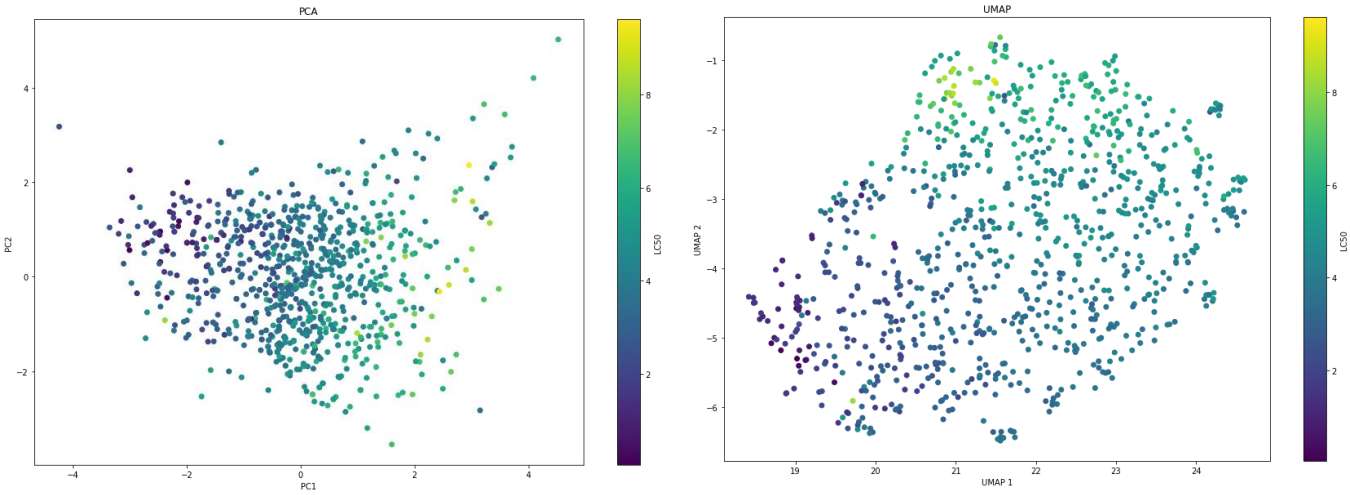
Parameter	Search Space
n_neighbors (k)	1:6
weights	Uniform, Distance
p	1 (Manhattan),2 (Euclidean)

FIGURE 3: Search space for KNN Optimization

All these models were tested against the validation set and Mean Squared Error (MSE), Mean Average Error (MAE), and R^2 scores were reported. Full code and hyperparameters are found in the appendix.

RESULTS

Dimensionality Reduction



FIGURES 4&5: PCA and UMAP on Training Features

Linear Regression

Univariate models were first tested to establish a performance baseline for the multivariate model and determine which variables might be of more importance in the multivariate model. The strongest univariate model was based on MLOGP and had a training $r^2 = 0.423$ and a testing $r^2 = 0.412$. The multivariate model without interaction had higher performance, with a training $r^2 = 0.564$ and testing $r^2 = 0.610$. Partial F-test revealed the

NdssC variable was not significant at $\alpha = 0.05$ and thus a reduced model was tested removing the insignificant variable. ANOVA was performed between the full and reduced model, and there was no indication of statistically significant difference between the full and reduced model. Training $r^2 = 0.563$ and testing $r^2 = 0.607$ for the reduced model. The model with all possible first order interactions was then fitted and produced a training $r^2 = 0.614$ and testing $r^2 = 0.607$. All insignificant terms were removed from the model and ANOVA confirmed there was no statistically significant differences between the full and reduced model at $\alpha = 0.05$. The reduced first order model produced a training $r^2 = 0.603$ and a testing $r^2 = 0.589$. All assumptions of linear regression were validated for the model utilizing diagnostic plots and VIF analysis. All the model summaries, ANOVA, diagnostics, and validation scores are found in the appendix.

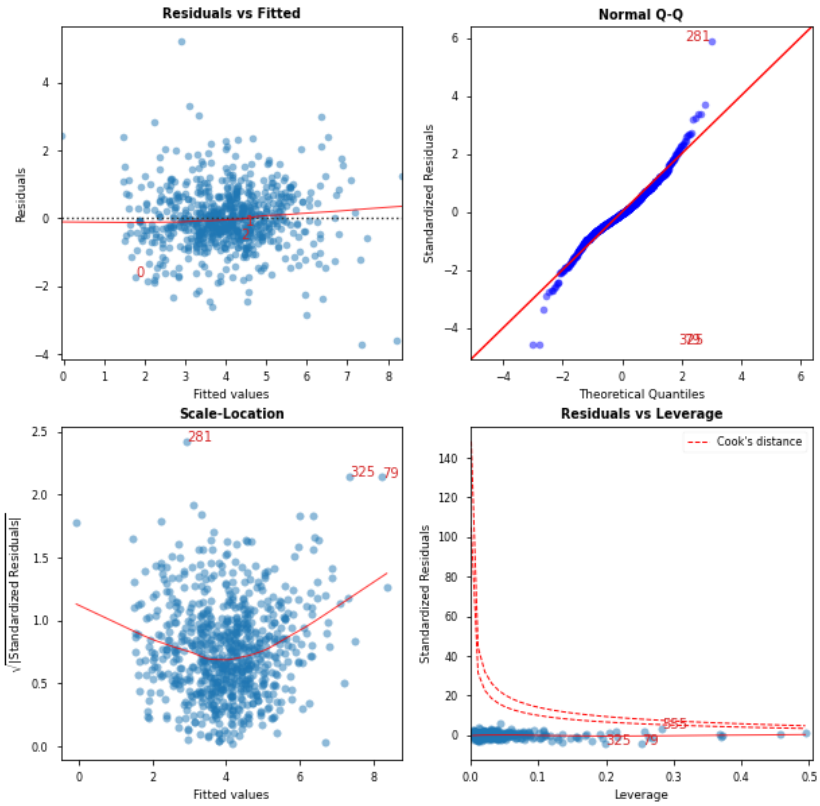


FIGURE 6: Diagnostic Plots for the Full Interaction Model

Regularization

Regularization techniques were applied on the second order regression model with all terms. While adding more features gives the model more information to train, it is prone to

overfitting leading to poor validation results. The goal of regularization is to constrain the model and prevent overfitting while providing more information than simple multivariate regression. Three regularization techniques were utilized: Ridge, Lasso, and Elasticnet. Ridge regularization produced validation $r^2 = 0.620$, Lasso regularization produced validation $r^2 = 0.608$, and Elasticnet produced validation $r^2 = 0.620$. All validation scores are in the appendix.

Random Forest

Random Forest regression was utilized to examine the inclusion of nonlinear trends in the model. Grid search was first utilized to optimize the hyperparameters (FIGURE 1), and resulted in a Random Forest model with n_estimators = 50, max_depth = 20, and min_samples_leaf = 5. The model with the best hyperparameters was then fitted on the training data and validated on the testing set, resulting in a validation $r^2 = 0.653$. All validation scores are in the appendix.

Sequential Neural Network

Sequential neural network was utilized to provide an example of the most computationally intensive model. The neural network used in this work is defined in FIGURE 2, was trained for 100 epochs, and optimized using the rmsprop algorithm. The validation $r^2 = 0.662$. All validation scores are in the appendix.

K-Nearest Neighbor (KNN)

KNN, while not interpretable as a model provides distance based, nonlinear learning. KNN was first optimized using grid search on the search space defined in FIGURE 3. Resulting hyperparameters were k = 5, distance-based weights, and p = 1. Validation $r^2 = 0.711$. All validation scores are in the appendix.

DISCUSSION

Dimensionality Reduction

Dimensionality reduction was employed to better visualize the relations between all 6 features and the target variable. PCA exhibited strong clustering of points with lower LC50 values while points with higher LC50 values were more spread out. Supervised UMAP exhibited the same consolidation of points with low LC50 values but also exhibited a strong cluster of points with high LC50. Since PCA is a fully linear transformation, and UMAP is a nonlinear manifold transformation, it can be deduced that there are certain trends in the data not fully capturable by a linear model. While the original scope of the work was to develop a strong, multivariate linear regression model to predict chemical toxicity, it was decided that further models needed testing in order to develop a strong model for predicting toxicity.

Linear Regression

As suggested by the previous section, there were nonlinear trends in the data that the linear model would be unable to learn. Within the scope of linear regression, two main models were tested, one with interaction, and the other without considering interaction. The models performed standardly, with the model with interaction having higher training scores than the model without interaction. This training success did not carry over to validation, indicating the model with interaction was overfitting the data, as the model without interaction validated much better than the model with interaction. The model with interaction potentially included too extraneous information, such that the model fit the training dataset too well and did not learn the general trend as well as the model provided with less information. Based on validation r^2 and considering potential overfitting, the best linear model is: $E(LC50) = 4.038 + 0.307 * CIC0 + 0.541 * SM1_{Dz} - 0.273 * GATS1i + 0.243 * NdsCH + 0.5562 * MLOGP$.

Regularization

As previously mentioned, the model with interactions was prone to overfitting. To solve this issue, regularization was applied to constrain the coefficients of the model. Both Lasso and Elasticnet were able to increase performance against the validation set compared to not only the overfit interaction model, but the best linear model as well. Regularization allowed for the

consideration of interactions and higher order terms while weighing them less than individual terms, increasing the overall performance of the model without overfitting.

Random Forest

Random Forest, unlike the previous models is not limited by being linear and is able to learn nonlinear trends in the data. The increased validation score combined with the analysis of PCA and UMAP validates the existence of nonlinear trends in the data, indicating the best performing model must be able to learn these trends. While the points in the UMAP are well clustered, Random Forest does not significantly weigh distance, and acts more like a probabilistic model instead, and thus isn't able to learn from the clustering.

Sequential Neural Network

The neural network indicated the added complexity of the nonlinear model could increase performance. As in the case of Random Forest, dense layers of the network do not consider distances. While this model performed the best out of all the models which did not consider distance, its training takes much longer than the random forest and other models. The scale of the problem is also inappropriate for neural networks, as there are only a limited number of chemicals. Neural networks function better when there are millions of points to learn from.

K-Nearest Neighbor

KNN regression, while being computationally less intensive than Random Forest and neural network, factors in distance when learning the data. Because of the clustered nature of the data as evidenced by UMAP, this model is highly appropriate. The validation score of the model suggests this claim is true, as KNN is the best performing model compared to the rest. As KNN lacks interpretability compared to linear regression, there are no coefficients to report. Thus, the model will be reported in terms of the hyperparameters. The best overall model in this study is a 5-Nearest Neighbors model with Manhattan distance and weighing points by distance.

SUGGESTIONS FOR FUTURE RESEARCH

The current scope of the research presents a regression problem, where 6 features are used to predict the value of the target feature—LC50. With future regulations potentially specifying a hard cutoff value for toxicity, there would be more value in classifying a chemicals

safety rather than its specific LC50 value. This would turn the problem into one of binary classification, allowing the use of logistic regression, and increasing the accuracy scores of the model. Points near the decision boundary in logistic regression would weigh the most, thus even if a point regressed poorly, if it were far from the decision boundary, it would matter much less. The model would also return confidence in each point, meaning that only chemicals that pass the model at a set α level would be allowed to be imported. Overall, these findings serve to provide a baseline model for future QSAR toxicity analysis and suggest improvements to future studies—including the use of classification techniques—in order to prevent damage to human health and to the health of the environment caused by toxic chemicals.

REFERENCES

[1] *Regulation (EC) No 1907/2006*. 2006, pp. 1–849

[2] M. Cassotti, D. Ballabio, R. Todeschini, V. Consonni. *A similarity-based QSAR model for predicting acute toxicity towards the fathead minnow (Pimephales promelas)*, SAR and QSAR in Environmental Research (2015) vol. 26, no. 3, 217-243,

[3] C.L. Russom, S.P. Bradbury, S.J. Broderius, D.E. Hammermeister, and R.A. Drummond, *Predicting modes of toxic action from chemical structure: Acute toxicity in the fathead minnow (Pimephales promelas)*, Environ. Toxicol. Chem. 16 (1997), pp. 948–967

[4] L. Michielan, L. Pireddu, M. Floris, and S. Moro, *Support vector machine (SVM) as alternative tool to assign acute aquatic toxicity warning labels to chemicals*, Mol. Inform. 29 (2010), pp. 51–64.

[5] M. Nendza, M. Müller, and A. Wenzel, *Discriminating toxicant classes by mode of action: 4. Baseline and excess toxicity*, SAR QSAR Environ. Res. 25 (2014), pp. 393–405.

[6] A. Levet, C. Bordes, Y. Clément, P. Mignon, H. Chermette, P. Marote, C. Cren-Olivé, and P. Lantéri, *Quantitative structure–activity relationship to predict acute fish toxicity of organic solvents*, Chemosphere 93 (2013), pp. 1094–1103.

[7] W.D. Marzio and M.E. Saenz, Quantitative structure–activity relationship for aromatic hydrocarbons on freshwater fish, Ecotoxicol. Environ. Saf. 59 (2004), pp. 256–262.

[8] K. Rose and L.H. Hall, *E-state modeling of fish toxicity independent of 3D structure information*, SAR QSAR Environ. Res. 14 (2003), pp. 113–129.

[9] T.I. Netzeva, A.O. Aptula, E. Benfenati, M.T.D. Cronin, G. Gini, I. Lessigiarska, U. Maran, M. Vračko, and G. Schüürmann, *Description of the electronic structure of organic chemicals using semiempirical and ab initio methods for development of toxicological QSARs*, J. Chem. Inf. Model. 45 (2005), pp. 106–114.

[10] M. Pavan, T.I. Netzeva, and A.P. Worth, *Validation of a QSAR model for acute toxicity*, SAR QSAR Environ. Res. 17 (2006), pp. 147–171.

[11] K. Roy and R.N. Das, *QSTR with extended topochemical atom (ETA) indices. 15. Development of predictive models for toxicity of organic chemicals against fathead minnow using second-generation ETA indices*, SAR QSAR Environ. Res. 23 (2012), pp. 125–140.

[12] Y.-Y. In, S.-K. Lee, P.-J. Kim, and K.-T. No, *Prediction of acute toxicity to fathead minnow by local model based QSAR and global QSAR approaches*, Bull. Korean Chem. Soc. 33 (2012), pp. 613–619. [10] T. Martin, P. Harten, R. Venkatapathy, and D. Young, T.E.S.T. (Toxicity Estimation Software Tool). U.S. E.P.A., 2012

[13] M. Casalegno, E. Benfenati, and G. Sello, *An automated group contribution method in predicting aquatic toxicity: The diatomic fragment approach*, Chem. Res. Toxicol. 18 (2005), pp. 740–746.

[14] D.V. Eldred, C.L. Weikel, P.C. Jurs, and K.L.E. Kaiser, *Prediction of fathead minnow acute toxicity of organic compounds from molecular structure*, Chem. Res. Toxicol. 12 (1999), pp. 670–678.

[15] M. Hewitt, M.T.D. Cronin, J.C. Madden, P.H. Rowe, C. Johnson, A. Obi, and S.J. Enoch, *Consensus QSAR Models: Do the benefits outweigh the complexity?*, J. Chem. Inf. Model. 47 (2007), pp. 1460–1468.

[16] S. Lozano, M.-P. Halm-Lemeille, A. Lepailleur, S. Rault, and R. Bureau, *Consensus QSAR related to global or MOA models: Application to acute toxicity for fish*, Mol. Inform. 29 (2010), pp. 803–813.

[17] U. Maran, S. Sild, P. Mazzatorta, M. Casalegno, E. Benfenati, and M. Romberg, *Grid computing for the estimation of toxicity: Acute toxicity on fathead minnow (Pimephales promelas), in Distributed, High-Performance and Grid Computing in Computational Biology*, W. Dubitzky, A. Schuster, P. Sloot, M. Schroeder, and M. Romberg, eds., Springer, Berlin, 2007, pp. 60–74

[18] M. Nendza and C.L. Russom, *QSAR modelling of the ERL-D fathead minnow acute toxicity database*, Xenobiotica 21 (1991), pp. 147–170.

[19] E. Papa, F. Villa, and P. Gramatica, *Statistically validated QSARs, based on theoretical descriptors, for modeling aquatic toxicity of organic chemicals in Pimephales promelas (fathead*

minnow), J. Chem. Inf. Model. 45 (2005), pp. 1256–1266. SAR and QSAR in Environmental Research 241

[20] A.P. Toropova, A.A. Toropov, A. Lombardo, A. Roncaglioni, E. Benfenati, and G. Gini, *Coral: QSAR models for acute toxicity in fathead minnow (Pimephales promelas)*, J. Comput. Chem. pp. 1218–1223.

[21] Y. Wang, M. Zheng, J. Xiao, Y. Lu, F. Wang, J. Lu, X. Luo, W. Zhu, H. Jiang, and K. Chen, *Using support vector regression coupled with the genetic algorithm for predicting acute toxicity to the fathead minnow*, SAR QSAR Environ. Res. 21 (2010), pp. 559–570.

[22] P. Mazzatorta, E. Benfenati, C.-D. Neagu, and G. Gini, *Tuning neural and fuzzy-neural networks for toxicity modeling*, J. Chem. Inf. Model. 43 (2003), pp. 513–518.

[23] P. Mazzatorta, M. Vracko, A. Jezierska, and E. Benfenati, *Modeling toxicity by using supervised kohonen neural networks*, J. Chem. Inf. Model. 43 (2003), pp. 485–492.

[24] J. Devillers, *A new strategy for using supervised artificial neural networks in QSAR*, SAR QSAR Environ. Res. 16 (2005), pp. 433–442.

[25] S.P. Niculescu, A. Atkinson, G. Hammond, and M. Lewis, *Using fragment chemistry data mining and probabilistic neural networks in screening chemicals for acute toxicity to the fathead minnow*, SAR QSAR Environ. Res. 15 (2004), pp. 293–309.

[26] McKinney, W. & others, 2010. *Data structures for statistical computing in python*. In Proceedings of the 9th Python in Science Conference. pp. 51–56.

[27] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. *Array programming with NumPy*. Nature 585, 357–362 (2020).

[28] *Scikit-learn: Machine Learning in Python*, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[29] Seabold, Skipper, and Josef Perktold. *statsmodels: Econometric and statistical modeling with python*. Proceedings of the 9th Python in Science Conference. 2010.

[30] J. D. Hunter, *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[31] M. L. Waskom, *seaborn: statistical data visualization*, Journal of Open Source Software, vol. 6, no. 60, p. 3021, 2021

[32] M. Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015

[33] McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). *UMAP: Uniform Manifold Approximation and Projection*. Journal of Open Source Software, 3(29), 861