Fish Toxicity Prediction using Machine Learning

Shamin Riza Hussain

224107211

Abstract

This project addresses the need for a QSAR (Quantitative Structure-Activity Relationship) model to predict acute toxicity towards the fathead minnow, (Pimephales promelas). The goal is to develop a model that can efficiently prioritize compounds for testing and risk assessment. A dataset comprising 125 organic compounds with known toxicity data for fathead minnows was compiled from publicly available sources. Various molecular descriptors were calculated for each compound in the dataset. These descriptors capture important physicochemical and structural properties of the compounds.LC50 data, which is the concentration that causes death in 50% of test fish over a test duration of 96 hours, was used as model response. The model comprised 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).

Regressor Models used:

K-Nearest Neighbours

Multiple Linear Regression

Support Vector Machine Regressor

Random Forest Regressor

Main Findings:

The developed model demonstrated promising predictive capabilities for acute toxicity towards the fathead minnow. The model exhibited a good correlation between the predicted and experimental toxicity values. This suggests its potential utility in prioritizing compounds for further testing and risk assessment related to fathead minnow toxicity.

Introduction

The acute toxicity assessment of chemical compounds towards aquatic organisms is a critical component of environmental risk assessment. Evaluating the toxicity of chemicals towards fish species is important due to their ecological significance and potential for bioaccumulation in aquatic food chains. The motivation for this project is to develop a predictive model that can effectively estimate the acute toxicity of chemical compounds towards fathead minnows. Such a model would aid in prioritizing compounds for further testing and risk assessment, potentially reducing the need for extensive and costly experimental testing on live organisms. QSAR (Quantitative Structure-Activity Relationship) models provide a valuable tool in this regard, as they establish relationships between the chemical structures of compounds and their biological activities.

Artificial Intelligence (AI) and Machine Learning (ML) have gained significant importance in the field of chemical engineering, revolutionizing various aspects of research, development, and process optimization. The project showcases the potential of AI and ML in predictive toxicology and environmental impact assessment in chemical engineering. AI&ML has huge potential in the areas of

1. Development of Predictive Models
2. Reduction of Experimental Testing
3. Process Monitoring and Control
4. Data Analysis and Pattern Recognition
5. Sustainability and Green Chemistry

Methodology

Data Preprocessing Techniques used:

Checking for missing values:

isnull().sum() function

Dropping rows with missing values:

The dropna() function

Checking for duplicates:

duplicated().sum() function

Dropping duplicates:

drop_duplicates() function

Handling outliers using Z-score:

The Z-score is calculated for each feature in the DataFrame to identify outliers. The formula (X - X.mean()) / X.std() calculates the Z-score by subtracting the feature mean and dividing by the standard deviation.

The np.where() function is used to find the indices of outliers based on the Z-score exceeding the threshold.We have set the threshold to 3.

The drop() function is used to remove rows with outlier indices from the feature DataFrame (X) and the target variable Series (y). This ensures that the outliers are eliminated from the data.

The to_csv() function is used to save the cleaned DataFrame (df) to a CSV file named 'cleaned_dataset.csv' . This allows for further analysis or use of the cleaned dataset.For our further training and testing of machine learning models we will be using this new dataset.

Visualisation of the data using pairplots and pearson correlation were done. LC50 was found to have a positive correlation with MLOGP, positive

correlation with SM1_Dz(Z), high negative correlation with GATS1i, and small positive correlation with the rest

Machine Learning Models used :

1. KNN Regression Model

   Firstly the data was normalized using the minmax scaler to a value in between 0 to 1 using scikit library.A for loop iterates over a range of values from 2 to 9 of values of k. This sets the number of neighbors to consider in the KNN algorithm. The KNN model is trained using the training data X_train and y_train with the fit() function. The R-squared score of the trained model is calculated using the score() function, which compares t he predicted values with the actual target values from the training data.

   Hyperparameter tuning - Grid Search

   Hyperparameters used- n_neighbours {3,5,7]

   weights[uniform,distance]

   minkowski parameter[1,2]

* Quantities in[] were tested in the gridsearch algorithm

   Cross Validation - 5 fold

   The param_grid dictionary defines the hyperparameters to be tuned for the KNN model. It specifies different values for the number of neighbors, the type of weight used , and the distance metric.This is how the hyperparameter grid is defined. The GridSearchCV function is used to perform grid search with cross-validation.It takes the KNN estimator (knn), the hyperparameter grid (param_grid), and the number of cross-validation folds (cv) as inputs.The fit() function is called to perform the grid search and find the best hyperparameters based on cross-validation performance.

   Predictions are made on the test set and the training set using the predict() function of the best model. The R-squared scores are calculated using the r2_score() function from sklearn.metrics to evaluate the model's performance on the test set (y_test).The R-squared score for the training set is also calculated.

2. Multiple Linear Regression

A linear regression model object mlr is created using the LinearRegression() class from scikit-learn and The fit() function was called to train the linear regression model using the training data X_train and y_train. The predict() function is used to generate predictions for the training data (X_train) using the trained linear regression model (mlr). The predicted values are stored in the mlr_train_pred variable. Similar to the KNN model the predicted results were analysed using Rsquared score, MSE, RMSE and MAE

Hyperparameter tuning – Randomized gridsearch

hyperparameters - estimator

param_distributions

n_iter

cv

random state

The randomized search will explore different combinations of hyperparameters and select the best combination based on cross-validated performance.

Hyperparameter tuning(2) - GridSearch

Hyperparameters - fit_intercept

normalize

copy_x

3. Support Vector Regressive(SVR) Algorithm

   The data is standardised using minmax scaler.After initializing the SVR model with the specified hyperparameters(c=1 and gamma=0.1), we fit the model to the training data using the fit method by passing X_train as the input features and y_train as the target label. All the usual analysis were do ne for the training and testing data after the model development.

   Hyperparameter Tuning – Gridsearch with cross validation

   Hyperparameters - kernel [linear,rbf]

   c [0.1,1,10]

   epsilon [0.01,0.1,0.5]

   After hyperparameter tuning optimal values of rbf,1,0.5 was obtained and it was used for the best model.Even though the tuned hyperparameters didn't improve the r2 score of test data, significant improvement was observed in the r2 score of training set.

4. Random Forest Regressor

Random Forest Regressor object with maximum depth of 6 and maximum features of 0.4 were used to train the data.Data was split into 80 percent for training and rest for testing. All the techniques like MAE,MSE etc were carried out to analyse the efficiency of the model.

Hyperparameter Tuning - Randomized Search with cross validation

Hyperparameters - n_estimators: [100, 200, 300]

max_depth: [None, 5, 10]

min_samples_split: [2, 5, 10]

min_samples_leaf: [1, 2, 4]

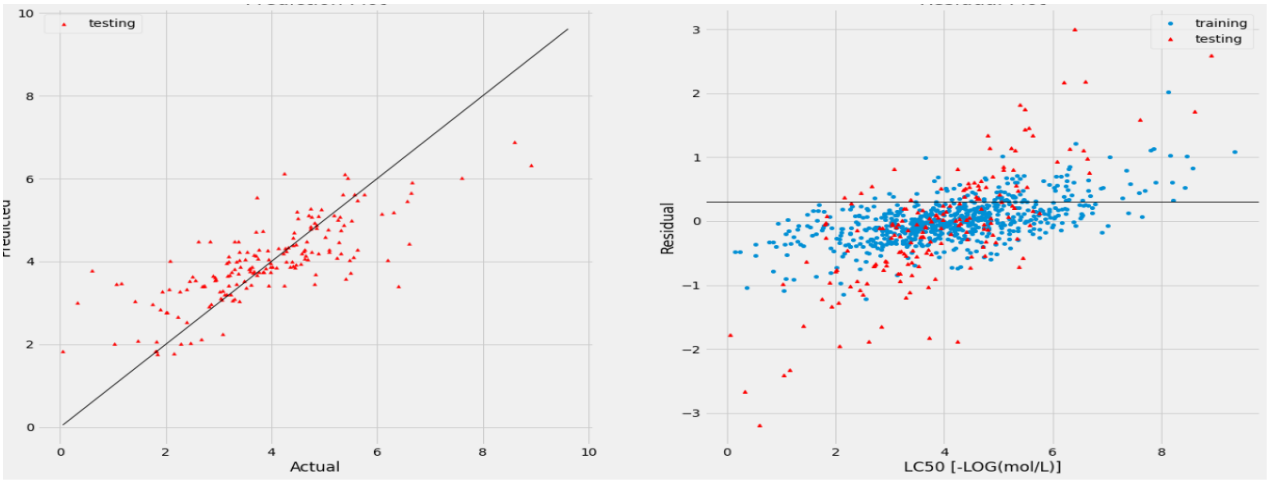
max_features: [auto, sqrt]

Results

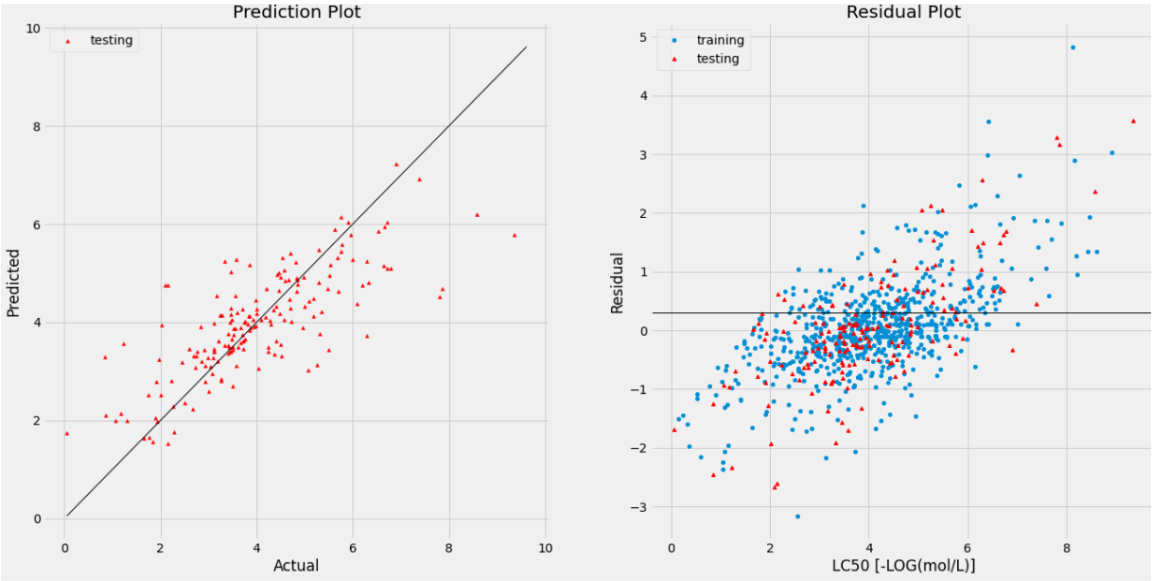
Model	R-squared	MSE	RMSE
KNN	0.6481	0.714	0.714
SVM	0.711	0.590	0.768
MLR	1	0.8495	0.921
Random Forest	0.946	0.11	0.331

Residual plots

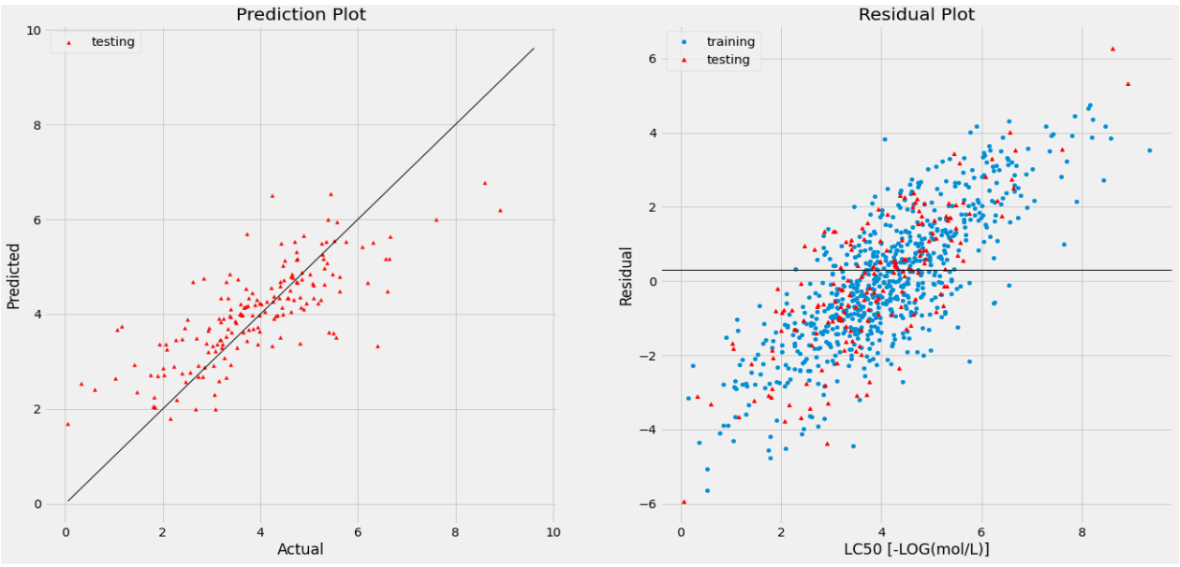
1. Random Forest



2. SVM Regressor



3. Multiple Linear Regression



4. KNN

After comparing all the 4 models we can conclude that random forest regressor turned out to be the best overall. After hyperparameter tuning it showed an MSE of 0.11 and R-squared of 0.946. Which is excellent result for the training conditions. Also all the other models were having an r squared value much lesser than the random forest model. Exception was the hypertuned multiple linear regression model which gave an r2 of 1. However this model had the least MSE values among the 4 so it should be suspected as a case of overfitting.

Conclusion

The study identified correlations between the model descriptors and toxicity, highlighting the influence of factors such as lipophilicity and the number of heteroatoms. Additionally, three descriptors were found to be related to known modes of action. It was expected that the selected descriptors would capture more general trends rather than specific modes of action due to the modeling of the dataset as a whole.

Overall, this project provides a valuable QSAR model for predicting acute toxicity in the context of chemical risk assessment. The model's satisfactory performance, along with the insights gained from the correlations between descriptors and toxicity, enhances our understanding of the factors influencing chemical toxicity towards the fathead minnow. The consideration of the applicability domain further contributes to the reliability of predictions.

Limitations:

Dataset limitations: Although the study mentioned using a large dataset, the specific details regarding the dataset's size, diversity, and representativeness were not provided. The performance and generalizability of the model could be influenced by the quality and comprehensiveness of the dataset used for modeling. Further information about the dataset's sources and potential biases would be beneficial

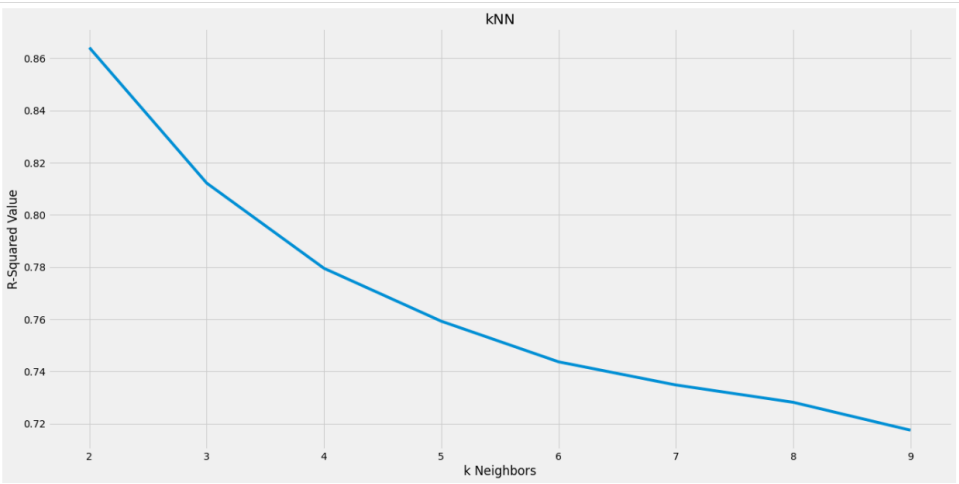
Descriptor selection: The choice of only six molecular descriptors, which do not require geometry optimization, may limit the model's ability to capture the full complexity of the chemical features related to toxicity. Including a broader range of descriptors that consider various physicochemical and structural properties could potentially enhance the model's predictive capability.

Interpretability of the model: Although the study highlighted correlations between the selected descriptors and toxicity, the interpretability of the model in terms of mechanistic insights or causative relationships is limited. The emphasis on general trends rather than specific modes of action may restrict the model's ability to provide detailed mechanistic understanding.

REFERENCES

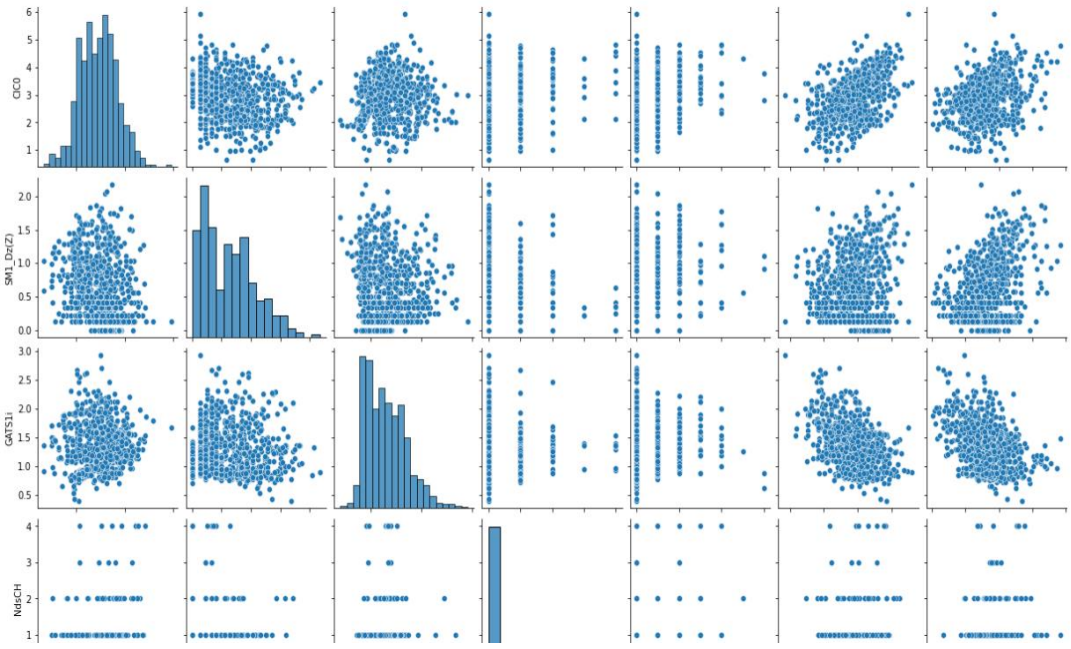
1. https://www.kaggle.com/datasets/ishandutta/qsar-fish-toxicity-data-set
2. https://www.researchgate.net/publication/273700321_A_similarity-based_QSAR_model_for_predicting_acute_toxicity_towards_the_fathead_minnow_Pimephales_promelas
3. https://archive.ics.uci.edu/ml/datasets/QSAR+fish+toxicity

APPENDIX



R2 values for different values of k in knn

```
<seaborn.axisgrid.PairGrid at 0x263a8462a90>
```



Pairplots of the data using seaborn library

Pearson Coefficient test

```
df_cleaned.corr(method='pearson')
```

	CIC0	SM1_Dz(Z)	GATS1i	NdsCH	NdssC	MLOGP	LC50
CIC0	1.000000	-0.235461	0.147199	0.121153	0.246397	0.463794	0.291799
SM1_Dz(Z)	-0.235461	1.000000	-0.146144	-0.141461	0.163138	0.200638	0.410881
GATS1i	0.147199	-0.146144	1.000000	-0.011036	0.091910	-0.451240	-0.398328
NdsCH	0.121153	-0.141461	-0.011036	1.000000	0.187976	0.048507	0.171944
NdssC	0.246397	0.163138	0.091910	0.187976	1.000000	0.028328	0.172310
MLOGP	0.463794	0.200638	-0.451240	0.048507	0.028328	1.000000	0.651648
LC50	0.291799	0.410881	-0.398328	0.171944	0.172310	0.651648	1.000000

LC50 has high positive correlation with MLOGP, positive correlation with SM1_Dz(Z), high negative correlation with GATS1i, and small positive correlation with the rest