Predicting the Biodegradability of Chemicals from QSAR Data

Muzhaffar Ma'ruf Ibrahim, 230118670

***Abstract*—This report has a purpose to build a model that can determine whether 1055 chemical samples are biodegradable or not, utilizing 41 distinct data features. Given that this scenario requires a binary classification approach, the Naive Bayes Classifier and Logistic Regression methods were employed to carry out the dichotomous categorization. This involved steps such as Data Preprocessing, Regularization, and Cross-Validation. The outcomes yielded an accuracy rate of 0.58 for the Naive Bayes Classifier and 0.90 for Logistic Regression. Higher Area Under Curve was achieved by Logistic Regression at 0.84 while Logistic Regression achieved a lower Area Under Curve at 0.72.**

***Index Terms*—Logistic Regression, Naïve Bayes Classifier, Biodegradable**

I. INTRODUCTION

One of the methodologies that REACH, A European union regulation that addresses the production and use of chemical substances, enhanced the usage is Quantitative Structure-Activity Relationships (QSAR), for gathering information regarding chemical properties. QSARs decipher patterns of chemical properties between molecules, enabling the prediction of properties without the need for extensive experimental testing.

The basis of this report is coming from research conducted in [1] which provide the 1,055 chemicals sample given with 41 feature data and labeled data on Biodegradable and Non-Biodegradable chemical. In this research, three classification modeling methods were applied, including K Nearest Neighbors (kNN), Partial Least Squares Discriminant Analysis, and Support Vector Machines as well as their consensus models to build the QSAR models for predicting biodegradability. The models were validated using training, test, and external validation sets. The classification performance of the QSAR models was evaluated based on specificity (Sp) and sensitivity (Sn), which represent the ability to correctly predict not ready biodegradable (NRB) and ready biodegradable (RB) molecules, respectively. The classification performances of the three QSAR models were comparable, with error rates (ER) in fitting and cross-validation equal to 0.14 for all computed models, and slightly higher error rates on the test set. The external validation set supported the results for the three QSAR and consensus models, with error rates in the range between 0.17 and 0.18, and a conservative behavior where specificity (Sp) was always higher than sensitivity (Sn). The lowest Error Rate (ER) in classification was reached by means of kNN, which gave an ER equal to 0.12 for the test set, and consensus analysis, which gave an error rate equal to 0.06 with 23% of not assigned molecules.

However, the usage of Machine Learning in QSAR has not completely satisfied most chemical practitioner to perfectly predict the biological activity by the reasons of chance correlation, rough response surfaces, incorrect functional forms, and overtraining [2]. The relationship between the structure and activity of molecules is complex and not always predictable. The inability to readily understand the meaning of many molecular descriptors used in QSAR models hinders the interpretation of those models and their predictive accuracy.

The aim of this study is to develop predictive quantitative structure-activity relationship (QSAR) models capable of accurately forecasting the biodegradability of chemical compounds. To achieve this objective, an experimental dataset comprising data for 1,055 chemicals was compiled. Using various modeling methods, the goal was to construct a thoroughly validated predictive model for biodegradability. A comprehensive dataset featuring both QSAR descriptors and biodegradability labels for 1,055 chemicals has been assembled in the process. In this paper, Naïve Bayes Classifier and Logistic Regression will be used to predict the biodegradability and non-biodegradability of a chemical using raw QSAR data.

II. DATA PROCESSING

A. Data Division

The original data in QSAR_data.mat will be categorized according to the information provided in [1]. This information details the experimental values for 1055 chemicals gathered from the National Institute of Technology and Evaluation of Japan's (NITE) website. The first 41 Columns are associated with each chemical characteristic and the final column corresponds to biodegradability and non-biodegradability status of each chemical. Thus, the feature needs to be separated from the data label using data indexing of X_QSAR for feature data and Y_QSAR for label data.

B. Missing Data or Infinite Value

Missing data check, also known as data cleaning or data scrubbing, is the process of identifying and addressing missing values in data sets. In the QSAR_data.mat, there has been checked using isnan and isinf function to address the missing

value and infinite value existence. The total missing data and the total infinite value in the data is 0.

C. Duplicated Rows Removal

By employing the unique function in MATLAB, it was discovered that there are six rows of data that contain duplicated entries. These rows, which appear more than once, were displayed in the MATLAB output. This founding is used the methodology of unique function to retrieve the index of unique rows. Afterward, the duplicated rows would be detected by a function of setdiff which finds the indices of all duplicated rows. The new set of feature data and label would be established using new variable of X_QSAR_new and Y_QSAR_new.

D. Normalization and Data Compression using PCA

Principal Component Analysis (PCA) is a statistical procedure that transforms a set of possibly correlated variables into a smaller set of uncorrelated variables called principal components. These principal components are chosen so that they capture as much of the variance in the original variables as possible. In this particular case, the code extracts the first 22 Principal Component from the normalized X_QSAR_new matrix using the pca() function. This choice is based on the observation that the first 22 PCs explain a substantial portion of the total variance in the data. The bar plot of the variances (latent) confirms this, showing that the first 22 PCs account for over 90% of the total variance in figure 1, indicating that they capture the most important variations in the data. Before the dimension reduction executed, the existing data would need to be normalized which consist of mean centering using z-score such that columns of features are centered to have mean 0 and scaled to have standard deviation 1.

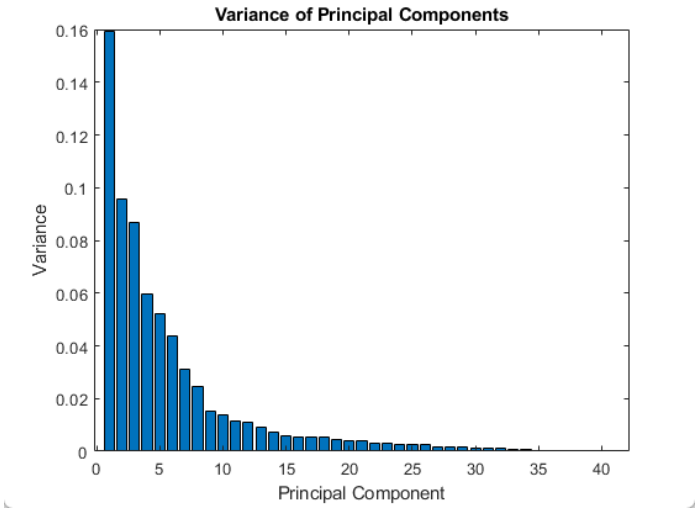

Figure 1. Variance of Principal Component

Based on the data set of each chemical Biodegradability and Non-biodegradability shown, the proportion of dataset could be shown in figure 2 which explained that 66% of chemicals in the data set is non-biodegradable and 34% is Biodegradable.

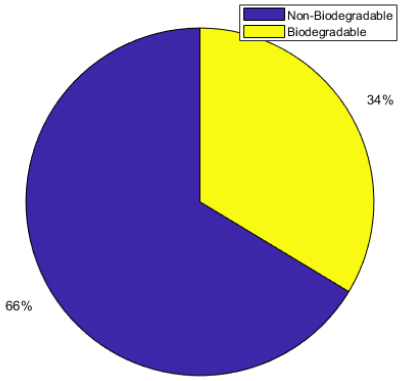

Figure 2. Proportion of Labels between Non-biodegradable and Biodegradable Chemicals

III. METHODOLOGY

A. Naïve Bayes Classifier

Naïve Bayes Classifier got its name because it is a Bayesian classifier that makes a simplifying (naive) assumption about how the features interact [3].

Naïve Bayes classifies the observations to the highest probability of a class called the maximum a posteriori decision rule. In MATLAB this function is initiated using fitcnb function, which takes some steps:

1) Estimate the likelihood of each class, which illustrates the distribution of each possible outcome or class.
2) Execute the modeling process of Posterior Probability for each class based on the predictors.

$$\hat{P} = \left(Y = k \middle| X_1, \ldots\ldots, X_p\right) = \frac{\pi(Y = k)\prod_{j=1}^{p} P(X_j|Y = k)}{\sum_{k=1}^{K} \pi(Y = k)\prod_{j=1}^{p} P(X_j|Y = k)} \quad (1)$$

Where:

- Y is the random variable related to the class index of observation
- $X_1, \ldots\ldots, X_p$ are the predictors of features or observations
- $\pi(Y = k)$ is the prior probability

3) Assign the class with the highest posterior probability or maximum posterior decision rule. The result of posterior probability of QSAR can be seen at the score variable in the matlab code.

The Pseudocode of the algorithm would be presented below:

Algorithm 1 Naïve Bayes

Define **Cross_Validation**
Initialize arrays for **accuracy** and **model_each_fold**

For each fold **i** in Cross_Validation:
    Split the **training data** and **test data** based on **i**

fold
Train Naive Bayes model (mdl) using **Feature_training** and **Label_training**
Predict labels for test data to the **Label_pred**
Append true label of test data to **true_labels** and **predicted_scores**
Calculate and store **accuracy** for fold **i**
Store the trained model for fold **i** in **model_each_fold**
Display **accuracy** for fold **i**

Find the **fold** with the highest accuracy with its corresponding model
Use the model with highest accuracy to predict data label from **X_QSAR_new**
Display the highest Accuracy of Naïve Bayes method

The programming implemented 5-fold cross validation since it utilizes 80% of data set from feature data and only 20% considering as test data. This portion is followed by another research which use the same portion to evaluate the model [1]. After defining the number of folds of Cross Validation, the program initializes several arrays such as accuracy variable to gather each accuracy for i-fold and model_each_fold array to gather the model generated by each k-fold training data.

In k-fold loop, model and accuracy is gathered along the completion of each loop. Each process is separated by the training data and test data. Then, the model created by initializing Bayes Classifier function of fitcnb will be stored in model_each_fold. Each Label_test that is used in each fold will also be stored in true_label to compare the performance of the best model using score or the posterior probability of label 1.

After finishing the process of k-fold validation, the accuracy for each k will be presented. The model with the highest accuracy would be picked to generate a prediction for the whole feature data in the data set to find its performance using the confusion matrix and Receiver Operating Characteristic (ROC).

B. Logistic Regression

Generally, logistic regression is well suited for describing and testing hypotheses about relationships between a categorical outcome variable and one or more categorical or continuous predictor variables [4]. Because the two outcomes variable are difficult to be described with an ordinary least squares regression equation due to the dichotomy of outcomes, one may instead create categories for the predictor and compute the mean of the outcome variable for the respective categories, showing a sigmoidal or S-shaped curve at the end [5]. Although logistic regression can accommodate categorical outcomes that are polytomous, in QSAR study case, the outcomes only consist of dichotomy label between biodegradability and non-biodegradability.

Logistic function can be represented in the equation 2

$$f(x) = \frac{e^{a_0 + a_1^T x}}{1 + e^{a_0 + a_1^T x}} \quad (2)$$

The Pseudocode of the algorithm can be viewed below:

Algorithm 2 Logistic Regression

Set **max_iterations, epsilon, lambda, k for k-fold cross validation to 5**
Initialize array for **predicted_labels, actual_labels, probabilities_predicted,** and **probabilities_actual**
For each fold **i** in Cross_Validation:
Split the **training data** and **test data** based on **i** fold
Initialize **theta** for each fold

For each iteration from 1 to **max_iteration**
Compute **Logistic Function**
Compute **diagonal matrix W**
Compute **gradient** of the **logistic loss**
Compute **Hessian Matrix** with **regularization**
Update theta **i** using Newton-Raphson update rule
Check for convergence; if **delta_theta** is small enough, BREAK the loop

Compute probabilities using the logistic function on the test set
Convert probabilities to binary predictions
Store predicted and actual labels for the current fold
Calculate accuracy for the current fold
Display fold number, regularization, and accuracy

The programming of Logistic regression begins with the initialization of maximum iteration, epsilon, lambda for regularization, and number of folds for k-fold cross-validation. Epsilon is used to detect the parameter if the point has not changed much in each iteration or converged. To prevent overfitting and reduce the model complexity, L2 regularization is applied to the Hessian Matrix calculation by applying a penalty to each squared residual of the feature weights. Hessian matrices find the direction and magnitude of the next step by substituting the function $f(x)$ for $L(\theta)$ which fulfills the update rule of Newton-Raphson in the equation 3

After iteratively receiving delta_theta with less than epsilon, the updated theta will be used to predict the test data in order to gather the information on the predicted label and actual label for each fold, which resulted as accuracy for each fold.

IV. Model Analysis

In this paper, Logistic Regression and Naïve Bayes is chosen since they are suitable for dichotomous classifier between biodegradability and non-biodegradability. In the model analysis, the performance of the algorithm would be examined by several measurements, such as accuracy, confusion matrix and Receiver Operating Characteristic curve.

TABLE 1

Model	*Fitting*		
	Accuracy	*Sensitivity*	*Specificity*
Naïve Bayes Classifier	*0.58*	*0.98*	*0.39*
Logistic Regression	*0.90*	*0.79*	*0.89*

A. Accuracy

To compute the accuracy between Naïve Bayes and Logistic Regression, the comparison between quantity of truly predicted label and targeted label divided by the length of label data.

$$Accuracy = \frac{Quantity\ of\ Truly\ Predicted\ Label}{The\ Number\ of\ Label} \quad (3)$$

In the measurement of accuracy in Naïve Bayes, the program takes the model with the highest accuracy between the process of k-fold. Afterward, the model will be used to predict the label of QSAR data using full feature data in X_QSAR_new. For the logistic regression, the accuracy calculation executed after the model reached convergence or epsilon less than 1×10^{-6}. Based on the result, Logistic regression obtains the higher accuracy of 0.90 and Naïve Bayes Classifier obtain 0.58.

B. Confusion Matrix and ROC

- Naïve Bayes Classifier

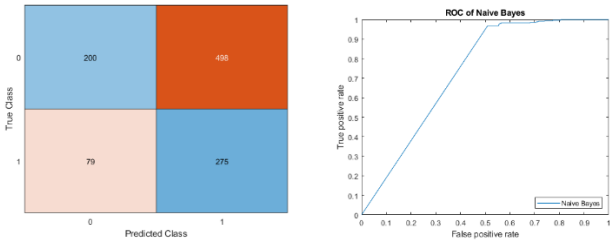

Figure 3. Confusion Matrix and ROC Curve of Naïve Bayes

Area Under Curve of Naïve Bayes Classifier: 0.72

- Logistic Regression

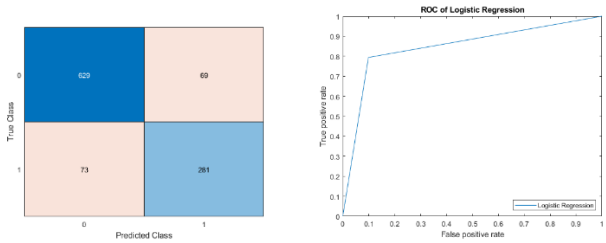

Figure 4. Confusion Matrix and ROC Curve of Logistic Regression

Area Under Curve of Logistic Regression: 0.84

To measure the effectiveness of binary classification, the AUC for each model derived from the Naive Bayes Classifier and Logistic Regression will be presented. Logistic Regression has shown a higher value of AUC compared to Naïve Bayes Classifier with the value of 0.84 and 0.72. It means the algorithm of Logistic Regression has a better performance on distinguishing two classes of label.

Based on the sensitivity and specificity which explained in the equation below,

$$Sn = \frac{TP}{TP + FN}, \qquad Sp = \frac{TN}{TN + FP} \quad (4)$$

Logistic Regression shows a good balance between specificity and sensitivity, which has a difference of 0.10. On the other hand, Naïve Bayes Classifier shows a large gap between its specificity and sensitivity around 0.59. It is caused by a high value of false positive which influences the specificity of Naïve Bayes Classifier become so small.

V. Conclusion and Recommendations

Regarding the comparison between these two models of machine learning, Logistic regression demonstrated a higher accuracy around 0.90 compared to Naïve Bayes Classifier. Conversely, while Naïve Bayes Classifier has exhibited a higher sensitivity or True Positive Rate than Logistic Regression, Logistic Regression show a better balance on detecting biodegradability and non-biodegradability of a chemical, with higher value on the specificity.

Naïve Bayes Classifier, noted for its simplicity and easy implementation, can also be easily updated with new data using online learning methods, which is useful in applications where data is continually growing, but with the assumption of conditional independence of each feature, the implementation is not practical on the real-world circumstances. Based on the complexity of the model, since the study case is categorized as high-dimensional modelling, Logistic Regression becomes more computationally expensive, when computing and inverting Hessian Matrix compared to Naïve Bayes Classifier.

In QSAR modelling, where complex relationship between each high dimensional feature is strongly considered, Neural Network might be the optimal approach to capture feature interactions.

References

[1] "Quantitative structure-activity relationship models for ready biodegradability of chemicals," J Chem Inf Model, vol. 53, no. 4, pp. 867–878, Apr. 2013, doi: 10.1021/ci4000213.

[2] S. R. Johnson, "The trouble with QSAR (or how i learned to stop worrying and embrace fallacy)," J Chem Inf Model, vol. 48, no. 1, pp. 25–26, 2008, doi: 10.1021/ci700332k.

[3] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M. Ingersoll, "Sample Data for Gender and Recommendation for Remedial Reading Instruction," 2002.

[4] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M. Ingersoll, "Sample Data for Gender and Recommendation for Remedial Reading Instruction," 2002.

[5] P. Tsangaratos and I. Ilia, "Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size," Catena (Amst), vol. 145, pp. 164–179, Oct. 2016, doi: 10.1016/j.catena.2016.06.004.