Predicting Biodegradability of Molecules through QSAR Analysis

Group 23
Pedro Barata (54483)
Diogo Pinto (55179)
Beatriz Rosa (55313)
Marina Novaes (55942)
All students contributed an equal amount of time.

1 Introduction and Goals

This report aims to utilize a quantitative structure-activity relationship (QSAR) analysis to predict the biodegradability of molecules. Biodegradability, the ability of a substance to be broken down naturally by biological processes, holds innumerous uses across several fields, including environmental science, pharmaceuticals, and chemical engineering. Accurately predicting biodegradability is essential as it assists researchers and scientists in making informed decisions regarding its usage, disposal, and potential environmental impacts.

To accomplish the goal of classifying molecules based on their biodegradability, we will make use of machine learning algorithms. These have demonstrated remarkable accuracy in learning from data and identifying complex patterns and relationships. In conjunction with these, we will be utilizing an edited and augmented version of the QSAR biodegradation dataset. This dataset will serve as the foundation for training and evaluating our models, enabling us to develop an effective classification system for accurately predicting the biodegradability of molecules.

2 Data Processing

In this section, we describe the steps taken to process the dataset before building the classification models.

We used pipelining to encapsulate the processing steps (preprocessing, feature selection, model fitting) into one object. This approach avoided data leakage and ensured that the same preprocessing steps were applied to the training and testing data.

2.1 Preprocessing

To measure the proportion of NaN values in the dataset, we calculated how many values were null per column and then performed the mean. In the results we obtained, the feature SpMax_B has the biggest proportion:

- Identification of categorical columns and unique values in dataset:

  We implemented a heuristic **get_categorical_numerical_columns(df, max_unique_ratio=0.05)** to determine whether a column should be treated as categorical or numerical based on the number of unique values it contains. Identifying them allowed us to apply the appropriate preprocessing steps specific to categorical data, by using **OneHotEncoder**(explained below). The **max_unique_ratio** parameter specified the threshold for considering a column as categorical. If the number of unique values in a column was less than or equal to **max_unique_values**, the column was considered as categorical; otherwise, numerical.

- Normalization, handling missing values:

  To deal with both categorical and numerical variables simultaneously, we used the **ColumnTransformer** class to specify separate imputation strategies for each type of column.

  For numerical columns, we made use of the **SimpleImputer** class to impute missing values which were replaced with the median of the respective column. We chose this strategy because it is robust to outliers and maintains the overall distribution of the data. Additionally, to categorical columns we opted for the same class to impute missing values, which were replaced by the most frequent value in the column since it is suitable for categorical data.

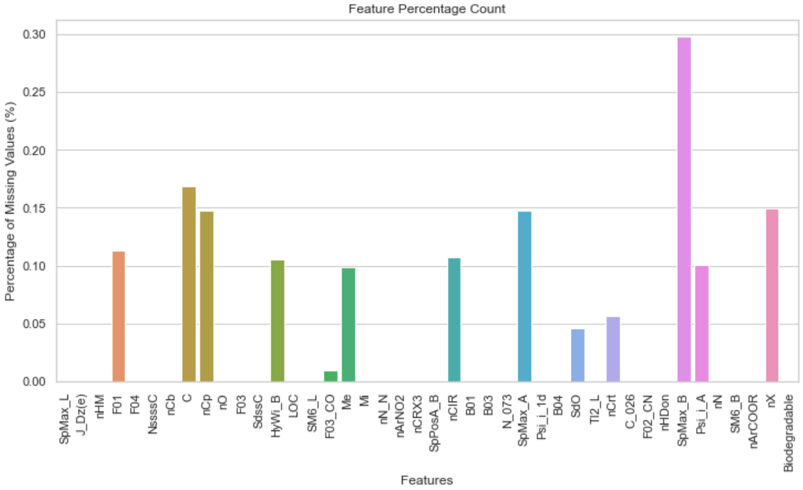

Figure 1: Barplot with Features in x-axis and Percentage of Missing Values in y-axis

As our algorithms can only work with numerical values, we decided to use **OneHotEncoder** to transform the categorical columns to numerical columns by returning a binary representation of the categorical columns. This encoding method ensured that categorical columns data was properly interpreted and used by treating each category independently as a separate feature, guaranteeing that the encoded categorical variables did not introduce any unintended relationships or orderings, thus preventing bias and incorrect interpretations in the models.

By splitting the data before preprocessing, we prevented data leakage from the test set to the training set. The inverse process would have led to unintentional information flow from the test set, resulting in an overly optimistic evaluation of the model's performance. By first dividing the data into training and testing sets and then applying preprocessing solely on the training set, we guaranteed that the model's evaluation was conducted on unseen data during training, this approach enabled a more accurate assessment of the model's performance and its ability to generalize to new data.

3 Variable Selection

In this section, we describe the approach followed for variable selection. This step was incorporated into the pipeline, so the process could be performed consistently for each model tested. The feature selection technique used in each model was `SelectKBest` with the `f_classif` scoring function. `SelectKBest` selects the top K features based on their individual statistical significance. The `f_classif` scoring function is specifically designed for classification tasks, and measures the statistical significance of each feature's relationship with the target variable. Both were chosen due to being well-suited for classification tasks.

The optimize the performance of the `SelectKBest` feature selection, we tested different hyperparameters for `SelectKBest`. By tuning these hyperparameters, we aimed to find the best configuration that maximized the performance of our models. This will be discussed in a later section.

4 Model Results

In this section, we present the different models that were tested for classifying biodegradability and their respective results. The models evaluated include logistic regression, support vector machine (SVM), and random forest. They were chosen for simplicity reasons. Each of these was assessed using various performance metrics such as accuracy, precision, recall, as well as the area under the ROC curve (AUC). The table below showcases the performance metrics for each model tested:

Model	Accuracy	Precision	Recall	F1-Score	AUC
Logistic Regression	0.950	0.97	0.98	0.95	0.973
SVM	0.960	0.97	0.97	0.96	0.980
Random Forest	0.970	0.98	0.98	0.97	0.991

Table 1: Performance statistics of each classification model

By observing the table, we can see that the models perform strongly, achieving high accuracy rates ranging from 95% to 97%. The highest accuracy is achieved by the random forest, followed by the SVM with an accuracy of 96%. The lowest accuracy, 95% is achieved by the logistic regression model.

Looking at the precision, recall, and F1-score, we can see that all models perform well, with values going above 0.95 for both classes. The random forest model, however, scores the highest in all these metrics. To further evaluate the models, we generated confusion matrices to help visualize the predictions of each model.

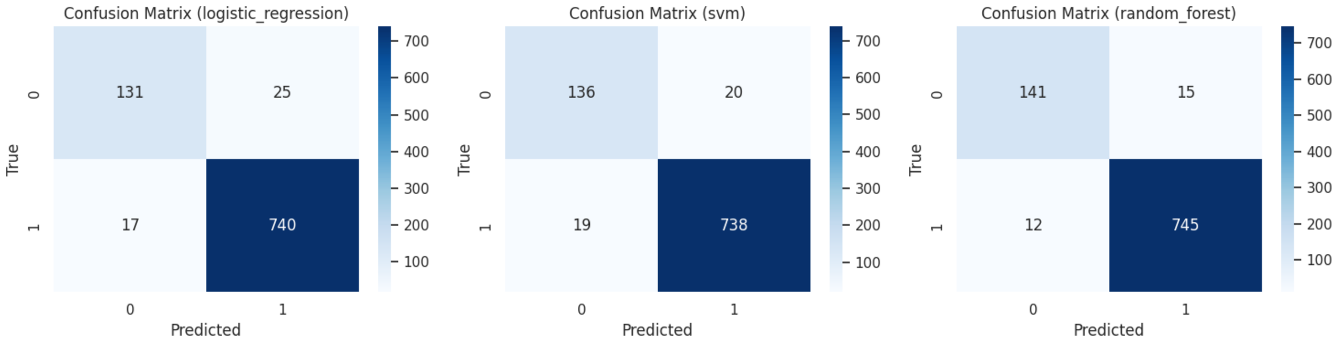

Figure 2: Confusion matrices of the classification models

Looking at the confusion matrices, we can observe the distribution of correct and incorrect predictions for each model, providing insights into the model's performance in classifying biodegradability. We can see that all models are well-balanced in terms of classification performance, seeing as they exhibit similar proportions of true positives, true negatives, false positives, and false negatives. This balance indicates that the models have found a good balance between accurately identifying biodegradable and non-biodegradable molecules. They are able to effectively distinguish between the two categories, showing a good compromise between sensitivity (recall) and specificity in their predictions.

We also evaluated the models' performance by plotting their ROC curves, which demonstrate the balance between accurately identifying positive cases and incorrectly classifying negative cases. The figure below shows the ROC curve for each model, as well as the corresponding AUC score.

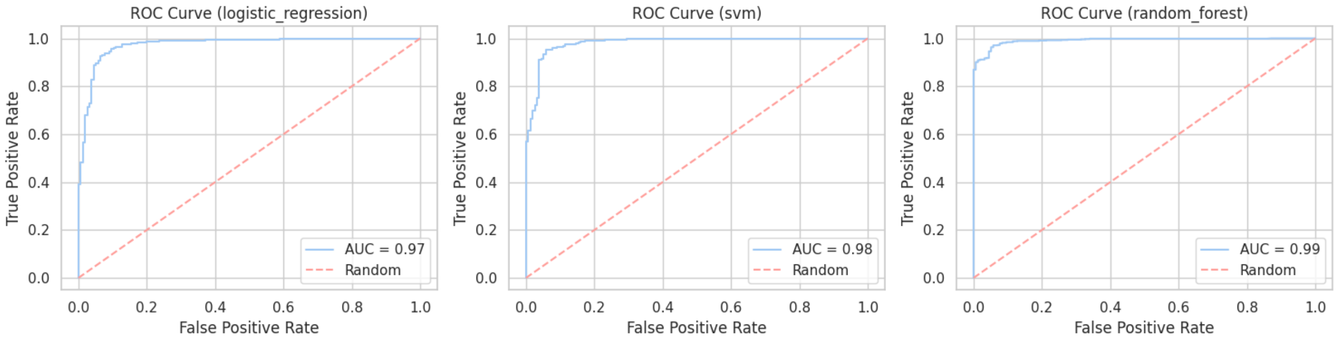

Figure 3: ROC curves of the classification models

Analyzing the models' AUC values and ROC curves allows us to measure their discriminative power. From Figure 3 and Table 1, we can see that the random forest model produces the highest AUC score of 0.991, followed closely behind by the SVM with an AUC score of 0.980 and logistic regression with an AUC score of 0.973. We can therefore conclude that the best performance was achieved by the random forest model.

5 Hyperparameter Tuning

In this section, we describe the process of tuning hyperparameters for the tested models to optimize their performance. We employed `GridSearchCV`, which explores various hyperparameter combinations to identify the best configuration. This approach allowed us to thoroughly search a wide range of possibilities and determine the optimal combination of hyperparameter values.

For each model, we defined parameter grids which contained the range of reasonable values to be used for the hyperparameters. The selection of these values was based on documentation and prior knowledge. Since we concluded in the previous section that the random forest model performed the best among the three models, we will focus on the hyperparameters used to tune this particular model. The hyperparameters considered for the random forest model and their optimal values were as follows:

- `feature_selection__k`: Number of features selected for each tree in the ensemble. This parameter was selected to ensure the model only selects the most relevant features during training. In the tuning process, we explored several values and found that the optimal one was `'all'`. This indicates that all available features were considered for each tree in the random forest.
- `model__n_estimators`: Number of decision trees in the random forest ensemble. This parameter helps regulate the complexity of the model. Amongst the values tested in the tuning process, we found that the optimal one for this parameter was `'200'`.
- `model__max_depth`: Maximum depth allowed for each decision tree in the ensemble. The adjustment of this parameter is important to regulate the complexity of the model and prevent overfitting. The optimal value was `'20'`.

6 Discussion and Conclusions

The results of our project indicate that the three models - logistic regression, support vector machines (SVM), and random forest models - can accurately perform the classification of the biodegradability of a molecule. The models performed reliably, balancing simplicity and accuracy. Among them, the random forest model outperformed the others and exhibited the highest performance. Through hyperparemeter tuning, we were able to enhance the performance of the model by tuning the number of trees in the ensemble, as well as the maximum depth of each tree.

There are, however, some limitations to our project: as discussed earlier, only one feature selection algorithm and scoring function were tested. This leaves room to test other feature selectors and scoring functions, which may potentially improve the performance of the model. Continued research and refinement are recommended to address the model's limitations and enhance its performance.