DSE 601: Project Assignment I Report

ANKUR GURIA (17039)
ankur17@iiserb.ac.in

PROBLEM STATEMENT:

Prediction of the **Bioactivity of SARS coronavirus 3C-like proteinase (CHEMBL5118)**

DATA RETRIEVAL:

The bioactivity data for **coronavirus 3C-like proteinase (CHEMBL5118)** that are reported as IC_{50} values in the nM (nanomolar) unit was collected using the ChEMBL web service package from the ChEMBL database.

- ➔ Used search tag 'coronavirus'.
- ➔ Among all the variants the bioactivity data for **CHEMBL5118** was retrieved.

DATA PRE-PROCESSING and DATASET PREPARATION:

- ➔ Handled **missing data** rows.
- ➔ Divided the bioactivity (standard_value column in data) of each molecule into **three classes** ("inactive" for value <= 1000, "active" for value >= 10000 and the rest as "intermediate"). We have then excluded the "intermediate" class and considered the remaining two classes only.
- ➔ The **features** that have been **directly extracted** except for the above bioactivity feature from the original dataset are:
  - ★ canonical_smiles
  - ★ standard_value
- ➔ Converted **IC50** to the negative logarithmic scale which is **-log10(IC50)** to allow **IC50** data to be **more uniformly distributed**.
- ➔ **Normalized** standar_value column data.

➔ **Feature Generation:** The features that had been generated are:

★ **Lipinski Descriptors** using **rdkit**:
- ❏ Molecular weight < 500 Dalton
- ❏ Octanol-water partition coefficient (LogP) < 5
- ❏ Hydrogen bond donors < 5
- ❏ Hydrogen bond acceptors < 10

★ **PaDEL Descriptors**

➔ Removed **low variance features**.

EXPLORATORY DATA ANALYSIS:

➔ Performed statistical analysis using the **Mann-Whitney U test** for each of the 4 Lipinski descriptors and pIC50 values.

➔ **Interpretation of Statistical Results**

◆ **pIC50 values**

Taking a look at pIC50 values, the **actives** and **inactives** displayed ***statistically significant difference***, which is to be expected since threshold values (IC50 < 1,000 nM = actives while IC50 > 10,000 nM = inactives, corresponding to pIC50 > 6 = Actives and pIC50 < 5 = Inactives) were used to define actives and inactives.

◆ **Lipinski's descriptors**

Of the 4 Lipinski's descriptors (MW, LogP, NumHDonors, and NumHAcceptors), only LogP exhibited ***no difference*** between the **active** and **inactive** while the other 3 descriptors (MW, NumHDonors, and NumHAcceptors) shows a ***statistically significant difference*** between **active** and **inactive**.

MODEL BUILDING:

➔ Regression Evaluation Metrics

Here we have used three common evaluation metrics for regression problems:

Mean Absolute Error (MAE) is the mean of the absolute value of the errors:

$1n\sum i=1n|yi-\hat{y}\ i|$

Mean Squared Error (MSE) is the mean of the squared errors:

$1n\sum i=1n(yi-\hat{y}\ i)2$

Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

$1n\sum i=1n(yi-\hat{y}\ i)$

Comparing these metrics:

MAE is the easiest to understand because it's the average error. MSE is more popular than MAE because MSE "punishes" larger errors, which tends to be useful in the real world. RMSE is even more popular than the MSE, because RMSE is interpretable in the "y" units. All of these are loss functions because we want to minimize them.

- **Gradient Boosting Regression**

Gradient Boosting trains many models in a gradual, additive, and sequential manner. The major difference between AdaBoost and Gradient Boosting Algorithm is how the two algorithms identify the shortcomings of weak learners (eg. decision trees). While the AdaBoost model identifies the shortcomings by using high weight data points, gradient boosting performs the same by using gradients in the loss function (y=ax+b+e, e needs a special mention as it is the error term). The loss function is a measure indicating how good are model's coefficients are at fitting the underlying data. A logical understanding of loss function would depend on what we are trying to optimize.

- ★ MAE: 0.2921631518593673
- ★ MSE: 0.15533380658934887
- ★ RMSE: 0.39412410049291435

- **Decision Tree Regression**

  The decision tree is a simple machine learning model for getting started with regression tasks. A decision tree is a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. The topmost node in a tree is the root node.

  ★ MAE: 0.3943060812758865
  ★ MSE: 0.32685097200234553
  ★ RMSE: 0.5717088174957121

- **Support Vector Machine Regression**

  Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because the output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken into consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.

  ★ MAE: 0.2963077475258582
  ★ MSE: 0.1518306599282603
  ★ RMSE: 0.38965453921167187

- **Random Forest Regression**

  A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging. What is bagging you may ask? Bagging, in the Random Forest method, involves training each decision tree on a different data sample where sampling is done with replacement.

  - ★ MAE: 0.302772897988821
  - ★ MSE: 0.1416303415627379
  - ★ RMSE: 0.37633806818170534

We can say the **best working model** by looking at **the MSE rates**. The best working model is **RANDOM FOREST**.