Data Mining: Molecular property prediction (Final Project)

Tuesday 23rd April, 2024

Aim

In this project, you are going to work with molecular data and your goal is to predict the "pIC50" - a measurement used in pharmacology and drug discovery to assess the potency of a compound in inhibiting a specific biological target or enzyme - as well as the "logP" - a measure of how a molecule interacts with different solvents. It indicates the molecule's preference for nonpolar (oily) or polar (water-like) environments.

Dataset

The given dataset (data.csv) consists of a set of formatted values for 16.087 molecules/SMILES and their associated pI50 and logP values. You are also given the number of atoms in each molecule, but this number can be easily recovered from the SMILES representation.

Model

Before starting your analysis it is very important to understand the data and the objective of this project.

Note that molecules can be represented as a sequence (as SMILES), as a graph (with atoms as nodes and bonds as edges), or as a 3D structure. You are free to choose the representation of the molecules that you are going to use. Character encoding, one-hot encoding, and target rescaling are some common practices that you may consider useful.

To predict the targets you can also use extra properties and descriptors (number of bonds, number of atoms, number of C, number of O e.t.c.) that you can extract using RDKit. (The properties that we mention are random examples, there are many more properties and descriptors for the molecules. Do not spend too much time finding the best properties).

You are free to try different approaches and models, you can use ready libraries for your algorithm, but you understand all the methods that you use and be able to justify your choice (including the arguments including the argument by default). The main focus is to see the model you have come up with, but also the other approaches that you tried. You have to understand deeply all the algorithms that you have tried.

You can use a different model for each target.

After cleaning the data, split the data into train and test sets. Use cross-validation for hyper-parameter optimization and select the best model based on statistical significant tests (the statistical test is required only if you have seen it in class). You will use the test data only for the best models (of course you have to include in your report and present all the models that you have tried). **It is mandatory to use three or more different algorithms in addition to a baseline**[1].

The final model we would expect is a model that can work on universal data, which means it can give a reasonable prediction on different molecular datasets.

Report

For the project, you are going to work in groups of 2. You have to submit a formal report. Your report has to include the approaches that you followed and the main results not only for your final model but for all the models you have tried. We want a full picture of what exactly you have done and how. You should be explicit and justify your assumptions. You should also discuss the different performances you have with your methods and explain why these work or not. What is important is to show us that you have a good understanding of the problem and of how to model it, what problems you encountered, and how you address them.

Note that this is an open project, you can try many different approaches as long as they make sense to get the best performance (creative ideas are always welcome).

Final submission

For your final submission, you have to submit on Moodle a folder named using your names (ex. NAME1_NAME2_DM_project) which should include your code (all the scripts), the dataset, and your report (the report should also be saved using your names: NAME1_NAME2_DM_project.pdf). If the size of your submission is big you can upload your submission on SWITCH drive and put on Moodle the shared link.

[1] The baseline is a naive baseline algorithm, providing context on just how good a given method is.

Oral Presentation

The exam consists of an oral presentation (15 minutes, prepared with slides, followed by questions and discussion). You should deliver your report at least two days (48 hours) before your exam.

APPENDIX

One Hot Encoding

One-hot encoding is a sparse way of representing data in a binary string in which only a single bit can be 1, while all others are 0. In the case of molecular graphs represented as a string, one-hot-encoding consists of encoding each character of the string with a binary vector.

Padding

Molecules do not have the same number of atoms, and their SMILES are not the same length. To address this issue, a popular technique is padding. Padding refers to the practice of adding extra data, typically zeros or a special token, to input sequences to ensure they are all the same length. Note that depending on the representation you will use, padding may not be necessary.

RDKit

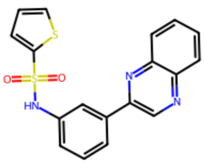
RDKit is an open-source Cheminformatics Software. It is a collection of cheminformatics and machine-learning tools written in C++ and Python. It allows you to work with many representations of chemical data and has the power to extract almost every chemical descriptor from the data you have. For more information about how to use RDKit and examples, see RDKit documentation.

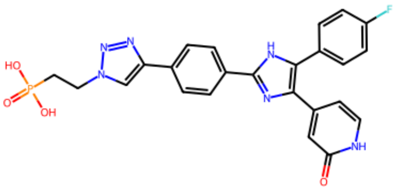
SMILES representations

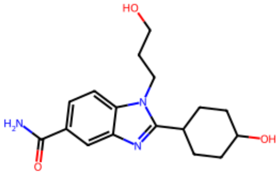
There are several ways to represent graphs for machine learning. The most popular way is the SMILES string representation. SMILES strings are a non-unique representation that encodes the molecular graph into a sequence of ASCII characters using a depth-first graph traversal. Atoms of chemical elements are represented by chemical symbols in capital letters. The hydrogen is usually ignored. Single bonds are not displayed; for double, triple, and quadruple bonds we shall use '=', '#', '$' respectively. Atoms that are bonded must stand nearby. Ring structures are written by breaking each ring at an arbitrary point (although some choices will lead to more legible SMILES than others) to make a 'straight non-ring' structure (as if it wasn't a ring) and adding numerical ring closure labels to show connectivity between non-adjacent atoms. Aromacity is commonly illustrated by writing the constituent B, C, N, O, P, and S atoms in lower-case forms b, c, n, o, p, and s, respectively. SMILES are typically first converted into a one-hot-based representation. The representation of a molecule through a SMILE string is not unique and the non-uniqueness of SMILES arises from a fundamental ambiguity about which atom to start the SMILES string construction. However, it is possible to transform a smile into canonical form (using specific tools such as RDKit).

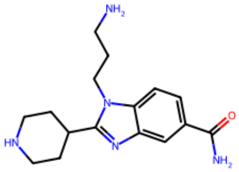
Here are examples of SMILES from the dataset and the corresponding 2D visualization:

’O=S(=O)(Nc1cccc(-c2cnc3ccccc3n2)c1)c1cccs1’
’O=c1cc(-c2nc(-c3ccc(-c4cn(CCP(=O)(O)O)nn4)cc3[nH]c2-c2ccc(F)cc2)cc[nH]1’
’NC(=O)c1ccc2c(c1)nc(C1CCC(O)CC1)n2CCCO’
’NCCCn1c(C2CCNCC2)nc2cc(C(N)=O)ccc21’




F
HO
P
O
OH
N
N
N
H
N
N
NH
O


HO
H2N
N
N
OH
O


NH2
N
HN
N
O
NH2