ReadMe for Covid Drug Discovery - ChEMBL Data ML Project

Anirudh Kalla

October 13, 2020

Project Description: Machine Learning Treatment of Covid Drug Discovery data from ChEMBL Database using graphical and numerical tools in Python.

Programming Language: Python 3

Distribution: Anaconda v2020.07

ChEMBL DataBase Versioning: $ChEMBL_25$

Database License: Creative Commons Attribution-ShareAlike 3.0 Unsupported license

DataSet Used: CHEMBL5118
($https://www.ebi.ac.uk/chembl/target_report_card/CHEMBL5118/$)

Dataset File (.csv): $bioactivitydatapreprocessed(3).csv$

Libraries and Packages used:

1. numpy: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.
2. os: This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the fileinput module. For creating temporary files and directories see the tempfile module, and for high-level file and directory handling see the shutil module.
3. matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.
4. pandas: Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

5. rdkit: RDKit is a collection of cheminformatics and machine-learning software written in C++ and Python. BSD license - a business friendly license for open source. Core data structures and algorithms in C++. Python 3.x wrapper generated using Boost.Python. Java and C# wrappers generated with SWIG. 2D and 3D molecular operations. Descriptor and Fingerprint generation for machine learning. Molecular database cartridge for PostgreSQL supporting substructure and similarity searches as well as many descriptor calculators. Cheminformatics nodes for KNIME. Contrib folder with useful community-contributed software harnessing the power of the RDKit.

6. Array: This module defines an object type which can compactly represent an array of basic values: characters, integers, floating point numbers. Arrays are sequence types and behave very much like lists, except that the type of objects stored in them is constrained. The type is specified at object creation time by using a type code, which is a single character.

7. sklearn.preprocessing: Provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators. Standardization of datasets is a common requirement for many machine learning estimators implemented in scikit-learn; they might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with zero mean and unit variance.

8. Seaborn: Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

9. $sklearn.model_selection.train_test_split$: Split arrays or matrices into random train and test subsets. Quick utility that wraps input validation and next($ShuffleSplit().split(X, y)$) and application to input data into a single call for splitting (and optionally subsampling) data in a oneliner.

10. Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language.It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

11. LinearRegression: Fits a linear model with coefficients $w = (w_1, \ldots, w_p)$ to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

12. Logistic Regression: In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the '$multi_class'$ option is set to 'ovr', and uses the cross-entropy loss if the '$multi_class'$ option is set to 'multinomial'.

13. K-Neighbors Regression: Based on k-nearest neighbors. The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set.

14. Decision Tree Regressor: Non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

15. Linear Discriminant Analysis (LinearDiscriminantAnalysis) and Quadratic Discriminant Analysis (QuadraticDiscriminantAnalysis): Classic Regressors, with, as their names suggest, a linear and a quadratic decision surface, respectively.

16. Gausssian Naive Bayes: supervised learning algorithm based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

17. Support vector machines (SVMs): Supervised learning methods used for regression.

18. Random Forest Regressor: Ensemble model that consists of many decision trees. Predictions are made by averaging the predictions of each decision tree. Or, to extend the analogy—much like a forest is a collection of trees, the random forest model is also a collection of decision tree models. This makes random forests a strong modeling technique that's much more powerful than a single decision tree.

19. Stochastic Gradient Descent: Linear model fitted by minimizing a regularized empirical loss with SGD. The gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule.

20. Gaussian Processes: Generic supervised learning method designed to solve regression and probabilistic classification problems.

21. Gradient Boosting Machine: Ensemble machine learning algorithm that uses decision trees. Boosting is a general ensemble technique that involves sequentially adding models to the ensemble where subsequent models correct the performance of prior models. AdaBoost was the first algorithm to deliver on the promise of boosting.

22. Ridge Regression: This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. Also known as Ridge Regression or Tikhonov regularization. This estimator has built-in support for multi-variate regression (i.e., when y is a 2d-array of shape $(n_samples, n_targets)$).

23. Multi-Output Regressor: Consists of fitting one regressor per target. This is a simple strategy for extending regressors that do not natively support multi-target regression.

24. Voting Regressor: Ensemble meta-estimator that fits several base regressors, each on the whole dataset. Then it averages the individual predictions to form a final prediction.