APDSE 2023/24 – SEM 2- Project

This year we will use the CvTdb, a database of pharmacokinetic time-series for environmental chemicals to predict the concentration of a toxic chemical after a given time after exposure. The dataset is formed by more than 16.000 series of concentration values of 187 analytes across more than 550 scientific studies, totaling more than 38.000 concentration value measurements.

To start working in your project you should read the paper **Database of pharmacokinetic time-series data and parameters for 144 environmental chemicals** available at https://www.nature.com/articles/s41597-020-0455-1. The full dataset can be found and downloaded from https://github.com/USEPA/CompTox-PK-CvTdb but we have prepared a set of CSV files with the contents of the database, organized as one CSV file for each table of the dataset (you can obtain detailed explanation of each table in the accompanying paper).

The project is organized in several task that you can extend at will.

Task 1 – Preparing your environment

We suggest that you to use the seaborn library to visualize the data, so check if it is already installed in your anaconda environment; if not, please add it.

Additionally, you might require sqlite3 library to access the dataset directly through pandas (you will learn how in the next lab sessions).

Task 2 – Prepare the data

Before anything else, read each table into a different DataFrame; note that the separator of the file is a vertical bar symbol '|' so that we do not have problems with colons or semicolons in text fields. Also, guarantee that the columns in the DataFrames have the appropriate types.

The dataset has some issues, so you must clean it, transform it, and prepare it for later use. In particular, do not forget to check for missing values (NULLS), duplicates and errors/outliers in the data. There are also some text columns which labels may not be coherent (e.g. small caps and upper caps in the text), and so they need to be normalized.

Task 3 – Create a function to visualize a (set of) series

We suggest that you also prepare some functions/procedures that allow you to visualize the pharmacokinetic time-series, for better understanding the data and also to check/visualize errors in the data. You should implement a function to draw a single series (don't forget to put an appropriate title, axis labels, units, and legend). It is also interesting to implement a function to draw several time-series (in the style of Fig. 4 of the paper).

Task 4 – Perform EDA

Perform Explanatory Data Analysis, taking into account our main goal, i.e. we would like to predict concentration predict the concentration of a toxic chemical after a given time after exposure.
Do not forget to check if there are correlations among the attributes to avoid unwanted bias in the machine learning algorithms. You might also want to see if there are different behaviors depending on the species, sex, or administration route.

Task 5 – Tune and train your classifier

The paper **Machine Learning and Pharmacometrics for Prediction of Pharmacokinetic Data: Differences, Similarities and Challenges Illustrated with Rifampicin** (available at https://www.mdpi.com/1999-4923/14/8/1530) uses the features TAD (time after dose), dose, OCC (treatment week), BMI (body mass index), age, gender, race, WT (weight), HT (height), HIV co-infection and FFM (fat-free mass) to predict the rifampicin plasma concentration over time or the rifampicin AUC0–24h. By following a similar approach, you will train a classifier to predict the concentration of a substance over time (one to predict the full time series from the initial dose, and another to predict the time series from two observations).

Be careful in splitting the data, since you need to put a full series either in the train or the test set, since the analysis will be performed in the full series. You should use the same Regressors studied in the paper (Linear, GradientBoosting, and RandomForest), but you can add extra ones for a more thorough analysis. To do hyperparameter tuning, use one of the classes provided by sklearn. The easiest to use is GridSearchCV that performs exhaustive cross validation with the given parameters to try (which can take a lot of time…). So, don't start too ambitious. You can find code examples in the solutions of lab sessions, as well as in sklearn documentation.

With the obtained parameter train your classifiers and test it in your test sample data. You should try to understand if the species or the administration route can influence the results.

Task 6 – Visualize the prediction results

You can adapt the previous code to present the results of the predictions. However, you can do it in a different way. This task is optional and is just to help you to understand the results.