High Level Design (HLD) Document

Prediction of LC50 value using Quantitative structure–activity relationship models (QSAR models)

Document Version Control

Date Issued Version Description Author

25/08/2022	1.0	Abstract, Introduction, General Description	Darshan
25/08/2021	1.1	Design Detail	Darshan
12/09/2021	1.2	Final Revision	Darshan

Contents

Abstract

In our industrialized society, a huge amount of chemical substances are used and produced every day. This increasing number of chemicals around us raises the problem of characterization, prediction and evaluation of their consequences to the human health and to the environment.

Toxicology provides the knowledge of mechanisms, rules, and data characterized by quality levels and defines the limits of safety of chemical agents. Chemistry offers knowledge on chemical descriptors and physico-chemical properties. Biology studies the mechanisms of the action of the chemicals on the animals and the other organisms used for tests.

In this project we are predicting the fish toxicity property of chemical compounds, and show how this can be approached using a computational intelligence method.

This dataset was used to develop quantitative regression QSAR models to predict acute aquatic toxicity towards the fish Pimephales promelas (fathead minnow) on a set of 908 chemicals. LC50 data, which is the concentration that causes death in 50% of test fish over a test duration of 96 hours, was used as model response. The model comprised 6 molecular descriptors: MLOGP (molecular properties), CIC0 (information indices), GATS1i (2D autocorrelations), NdssC (atom-type counts), NdsCH ((atom-type counts), SM1_Dz(Z) (2D matrix-based descriptors). Details can be found in the quoted reference: M. Cassotti, D. Ballabio, R. Todeschini, V. Consonni. A similarity-based QSAR model for predicting acute toxicity towards the fathead minnow (Pimephales promelas), SAR and QSAR in Environmental Research (2015), 26, 217-243; doi: 10.1080/1062936X.2015.10189380

Introduction

Why High-Level Document (HLD)

The HLD uses nontechnical to mild technical terms that should be understandable to the administrators of the system.
This documentation can be used as reference also helps remove any contradictions before coding.
The purpose of this HLD is to add necessary detail to the current project description to represent a suitable model for coding.
A HLD design provides an overview of a system, product, service or process.
HLD presents all of the performance requirements, design aspects, hardware and software interfaces, user interfaces, design features and architecture of the project.

Project Description

The goal of this project is to predict quantitative structure activity relationship with toxicity level. To build the project we have used the scientific dataset containing 6 attributes (molecular descriptors) of 908 chemicals used to predict quantitative acute aquatic toxicity towards the fish Pimephales promelas (fathead minnow).

Data Set Characteristics:	Multivariate	**Number of Instances:**	908	**Area:**	Physical
Attribute Characteristics:	Real	**Number of Attributes:**	7	**Date Donated**	2019-09-23
Associated Tasks:	Regression	**Missing Values?**	N/A	**Number of Web Hits:**	46489

Source : UCI Machine Learning Repository

Tools Used:

Business Intelligence tools and libraries works such as NumPy, Pandas, Seaborn, Matplotlib, MS-Excel, MS-Power BI, Jupiter Notebook and Python Programming Language are used to build the whole framework.

Design Details:

Step 1:
Acquire data from source and load into data warehouse of analytics repository.

Step 2:
Perform data wrangling.

Step 3:
Perform analytics and queries and built visualizations, dashboards and reports.

Optimisation:

1. **Your data strategy drives performance**
   - **Minimize the number of fields**
   - **Minimize the number of records**
   - **Optimize extracts to speed up future queries by materializing calculations, removing columns and the use of accelerated views**
2. **Reduce the marks (data points) in your view**
   - **Practice guided analytics. There's no need to fit everything you plan to show in a single view. Compile related views and connect them with action filters to travel from overview to highly-granular views at the speed of thought.**
   - **Remove unneeded dimensions from the detail shelf.**
   - **Explore. Try displaying your data in different types of views.**
3. **Limit your filters by number and type**
   - **Reduce the number of filters in use. Excessive filters on a view will create a more complex query, which takes longer to return results. Double-check your filters and remove any that aren't necessary.**
   - **Use an include filter. Exclude filters load the entire domain of a dimension while including filters do not. An include filter runs much faster than an exclude filter, especially for dimensions with many members.**
   - **Use a continuous date filter. Continuous date filters (relative and range-of- date filters) can take advantage of the indexing properties in your database and are faster than discrete data filters.**
   - **Use Boolean or numeric filters. Computers process integers and Booleans (t/f) much faster than strings.**
   - **Use parameters and action filters. These reduce the query load (and work across data sources).**