Low Level Design

Prediction of LC50 using QSAR models

written by:Aditya Rokade

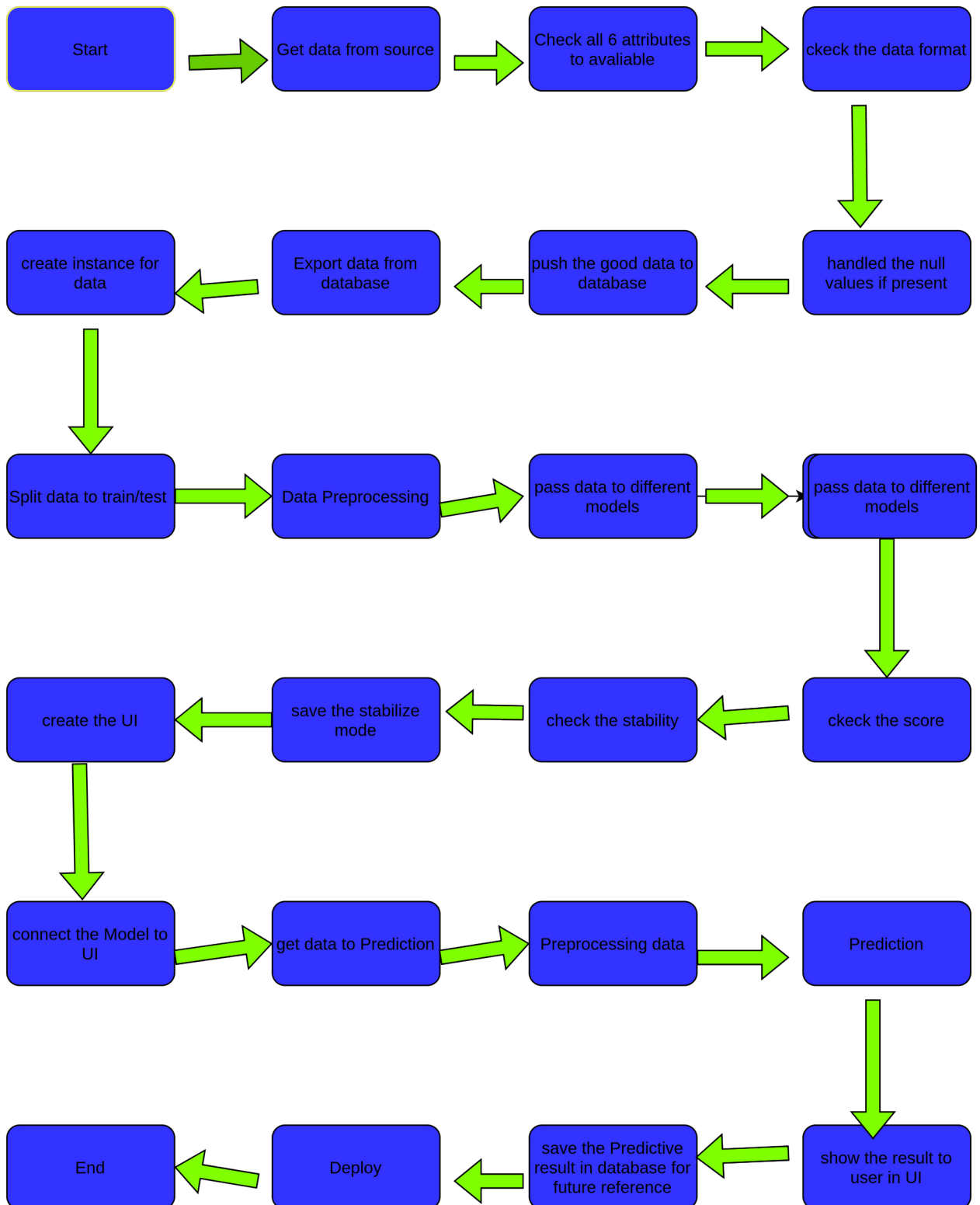# Contents

1. Introduction

1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for QSAR models. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software

architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2.Architecture

```mermaid
flowchart
    Start --> Get_data_from_source["Get data from source"]
    Get_data_from_source --> Check_attributes["Check all 6 attributes to avaliable"]
    Check_attributes --> Check_format["ckeck the data format"]
    Check_format --> Handled_null["handled the null values if present"]
    Handled_null --> Push_good["push the good data to database"]
    Push_good --> Export_data["Export data from database"]
    Export_data --> Create_instance["create instance for data"]
    Create_instance --> Split_data["Split data to train/test"]
    Split_data --> Data_Preprocessing["Data Preprocessing"]
    Data_Preprocessing --> Pass_models1["pass data to different models"]
    Pass_models1 --> Pass_models2["pass data to different models"]
    Pass_models2 --> Check_score["ckeck the score"]
    Check_score --> Check_stability["check the stability"]
    Check_stability --> Save_stabilize["save the stabilize mode"]
    Save_stabilize --> Create_UI["create the UI"]
    Create_UI --> Connect_model["connect the Model to UI"]
    Connect_model --> Get_prediction["get data to Prediction"]
    Get_prediction --> Preprocessing_data["Preprocessing data"]
    Preprocessing_data --> Prediction["Prediction"]
    Prediction --> Show_result["show the result to user in UI"]
    Show_result --> Save_predictive["save the Predictive result in database for future reference"]
    Save_predictive --> Deploy["Deploy"]
    Deploy --> End
```

- Start
- Get data from source
- Check all 6 attributes to avaliable
- ckeck the data format
- handled the null values if present
- push the good data to database
- Export data from database
- create instance for data
- Split data to train/test
- Data Preprocessing
- pass data to different models
- pass data to different models
- ckeck the score
- check the stability
- save the stabilize mode
- create the UI
- connect the Model to UI
- get data to Prediction
- Preprocessing data
- Prediction
- show the result to user in UI
- save the Predictive result in database for future reference
- Deploy
- End

3.Archicture Description

3.1 Data description
vivo test data on fish for hundreds of chemical substances using the
ECOTOX database of the US Environmental Protection Agency.
There are csv file seperated by ';'.dataset contaions 6 molecular descriptors and 1 quantitative experimental response.

3.2 Data Transformation
the given dataset is in csv format seprated by ";" and in int format.so no need to transform the data

3.3 Data validation
validate the data /check the data.data is in structed ir not.if not then first structed it

3.4 Insert data into database
to store the data we can use the cassandra database(online).so when we need the data we can retrive the data

1.create account/database
2.create connection
3.create the table for data storing
4.insert the data fine on that
5.close connection

3.5 Exporting data from database

retrive the data to train the model .

3.6 Create instance to data
store the data in variable

3.7 splitting data
split the data to train and test .so we can ckeck the our model is working good or not
so we can split the data in (75%-25%) respectively.so after we can check the model

3.8 Data proprocessing
if data is skeued then we need to convert data into normal distribution.means we transfer the whole scale of data in which mean is zero and standred deviaction is 1

3.9 pass data to different models
to check the which model is working will on this dataset we can test on different models
1. we can create multimodel system using clustering
2. directly checking the algorithm

3.10 check the score ans stability of model
to avoid the overfitting /underfitting condition we need to check the stability of model
based on that we can choose the best /stabilize model

3.11 craete the UI

to access the model by user we can craete a simple UI,so our user easily connect the model
by using the UI user can predict the value(concentration of LC50) .and after prediction the result is
also show using UI.

3.12 Prediction

when data is recive from user through UI,we need to preprocess the data and then pass the data to
model to prediction .because we train the model on different scale so we need to transfer the data on
that scale

3.13 Result show to user

the prediction is recive from model is in different scale then again we need to transfer on oiginal scale
and then show to user through UI.

3.14 save the result

we can save the result on database for future reference ,in future when we required we can get the
result

3.15 Deploy

we need to deploy the our whole system on cloud so users can access from different places
we can deploy the system on AWS/Azure.so everycan use this system