Prediction of LC50 value using QSAR Models

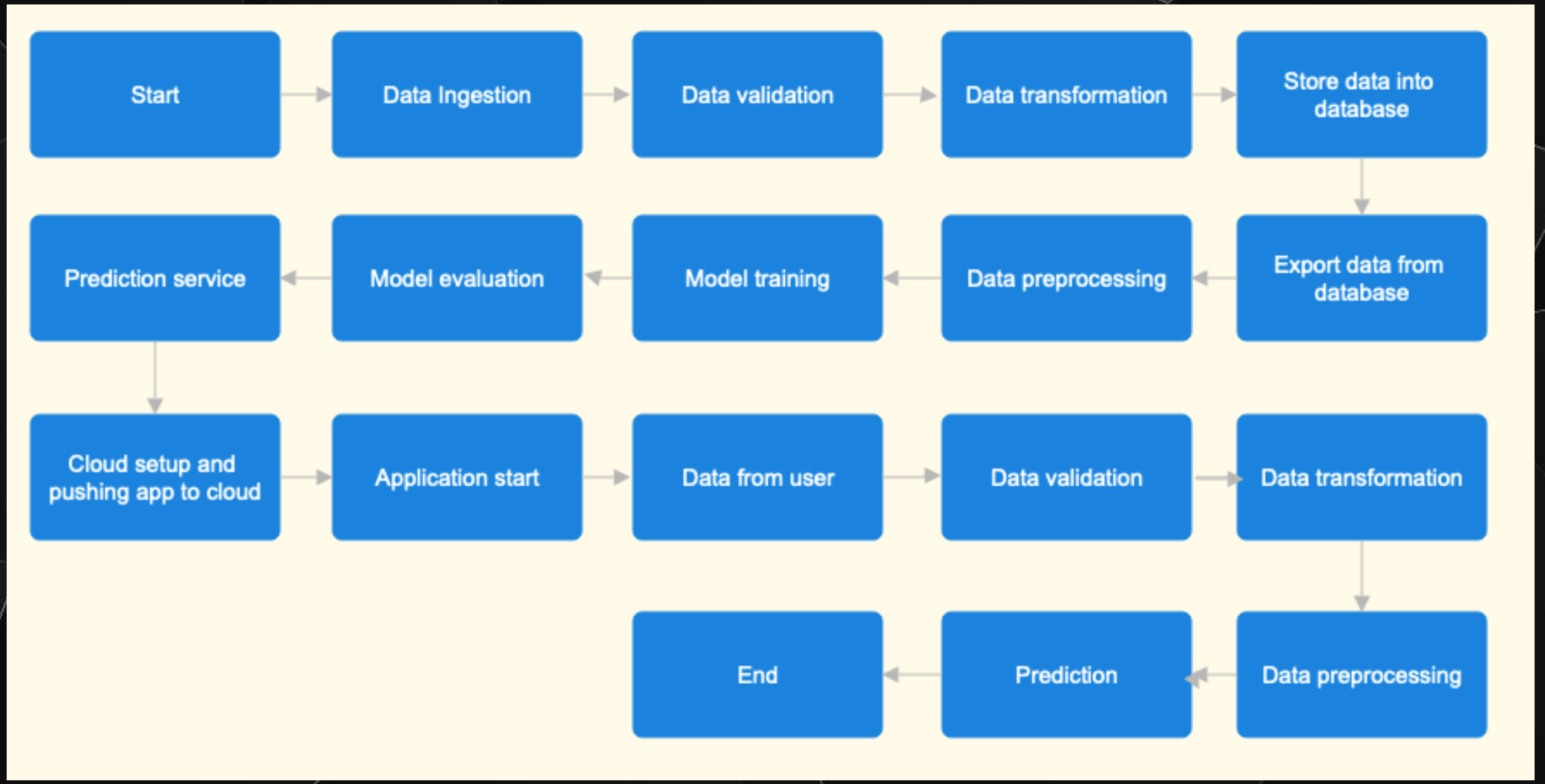
Objective

- Develop a quantitative regression QSAR(Quantitative Structure Activity Relationship) model to predict acute aquatic toxicity towards the fish Pimephales promelas (fathead minnow)
- QSAR models
  - provide a cost-effective alternative to expermental methods
  - Provide guidance for compound design
  - Are a tool for regulatory complicance

Data Sharing Agreement

- Dataset is available at UCI Machine Learning Repository. Please refer here
- Data set consists of 908 instances of 6 attributes (molecular descriptors) and 1 quantitative experimental response (LC50 value)
- Column names and datatypes
  - CIC0 (Float)
  - GATS1i (Float)
  - NdssC (Int)
  - SM1_Dz(Z) (Float)
  - NdsCH (Int)
  - MLOGP

Architecture



Data Ingestion

- Dataset is available at UCI Machine Learning Repository. Please refer here
- Create artifacts/data_ingestion/data directory. Download and extract files to this directory.
- Keep only CSV files to the directory and remove any other types of files

Data Validation and transformation

- Row file name validation with respect to filename mentioned in config/config.yaml
- Validate number of columns is 7 (6 molecular descriptors and 1 quantitative response)
- Validating the data doesn't contain missing values in whole columns
- If the validation is successful, the data is copied to artifacts/data_validation/traning_raw_files_validated
- Otherwise the data is moved to bad data directory
- Missing values (if exists) are substituted as Null

Database Operations

- Create database table
- Insert good data into table
- Export data from table into CSV file. At the end of this stage, inputfile.csv is available at artifacts/training_file_from_db directory

Data Preprocessing

- Load input data for training
- Separate features and label
- Dealing with null values / Imputing missing values
- Dropping columns with zero std deviation
- Perform standard scaling
- Save preprocessed data. At the end of this stage, the preprocessed data is available in the artifacts/preprocessed_data directory

Model Training

- Load input data for training
- Perform Hyper parameter tuning for different machine learning algorithms such linear regression, support vector machine, decision tree and choose the model with best metrics
- Save the best model in terms of performance metrics(r2_score and rmse)

Prediction Service

- Input feature values from the user
- Perform data preprocessing
- Load the best model and predict the output
- Display output to the user

Q & A

- What is the source of data?
  - Data is available at UCI Machine Learning Repository. Please refer here
- How logs are managed
  - Logging in maintained for each stage in the pipeline
- What techniques are used for data preprocessing?
  - Imputing missing values with KNNImputer
  - Removal columns with Zero standard deviation
  - Perform standard scaling

Q & A (Continued)

- How training was done or what models were used?
  - Hyper parameter tuning is performed with different machine learning algorithms such as linear regression, support vector machine, decision trees
  - Chosen the model that provide the best performance
- How prediction was done?
  - The best model in terms of performance metric evaluation is stored
  - User input is received for all the features
  - Based on input features using best model, the corresponding prediction is made and displayed to the user