



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

---

# Machine learning for molecular property prediction and drug safety

A broad perspective on using deep learning to predict acid dissociation constants

Master's thesis in Applied Data Science

KINGA JENEI

---

Department of Computer Science and Engineering  
CHALMERS UNIVERSITY OF TECHNOLOGY  
UNIVERSITY OF GOTHENBURG  
Gothenburg, Sweden 2023



MASTER'S THESIS 2023

# Machine learning for molecular property prediction and drug safety

A broad perspective on using deep learning to predict acid  
dissociation constants

KINGA JENEI



UNIVERSITY OF  
GOTHENBURG

---



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering  
*Division of Data Science and AI*  
CHALMERS UNIVERSITY OF TECHNOLOGY  
UNIVERSITY OF GOTHENBURG  
Gothenburg, Sweden 2023

Machine learning for molecular property prediction and drug safety  
A broad perspective on using deep learning to predict acid dissociation constants  
KINGA JENEI

© KINGA JENEI, 2023.

Supervisor: Rocío Mercado, Department of Computer Science and Engineering  
Company supervisor: Vigneshwari Subramanian, AstraZeneca  
Examiner: Ola Engkvist, Department of Computer Science and Engineering

Master's Thesis 2023  
Department of Computer Science and Engineering  
Division of Data Science and AI  
Chalmers University of Technology and University of Gothenburg  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Gothenburg, Sweden 2023

Machine learning for molecular property prediction and drug safety  
A broad perspective on using deep learning to predict acid dissociation constants  
KINGA JENEI

Department of Computer Science and Engineering  
Chalmers University of Technology and University of Gothenburg

## Abstract

Utilizing machine learning methods for the prediction of acid dissociation ( $pK_a$ ) values of compounds holds great significance, as  $pK_a$  is an important parameter, optimized frequently in drug discovery. Accurate prediction of  $pK_a$  values could potentially provide valuable insights on other molecular properties and thereby support compound design. In an attempt to extend the scope of  $pK_a$  prediction, we have created several machine learning models utilizing internal AstraZeneca data. We explored both classical ML approaches with different molecular descriptors, and deep learning methods. The results showed that graph neural network based models outperform tree based methods and yielded reasonable predictions for both acidic and basic  $pK_a$  values. Through the implementation of several data splitting strategies, we have substantiated that the models hold the potential to generalize well to novel compounds and outperform state of the art methods. Besides evaluating the models on different splits of the internal data, their performance was also assessed on public datasets. This yielded comparatively lower accuracies which can be attributed to the collation of data from diverse sources and the high experimental variability of the publicly available data.

Keywords: Molecular property prediction, Acid dissociation constant, pKa, Machine learning, Graph Neural Networks, Molecular descriptors, Drug Discovery.



## Acknowledgements

I would like to express my deepest gratitude to my company supervisor, Vigneshwari Subramanian, for her guidance, invaluable feedback and encouragement throughout the project. Additionally, this endeavor would not have been possible without my university supervisor, Rocío Mercado, whose knowledge, ideas and advice helped shape this project. I would also like to express my appreciation to Emma Evertsson and Susanne Winiwarter for their support and assistance, and whose expertise has been a great contribution to this work. I have valued the time spent at AstraZeneca and would like to thank everyone who I have met and had the opportunity to work with throughout the last six months. I am also grateful for the input of my examiner, Ola Engkvist, and for the feedback and editing help of my opponents. Last but not least, I would like to thank my family and friends for supporting, motivating and believing in me.

Kinga Jenei, Gothenburg, June 2023





# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Goals and challenges . . . . .	3
1.3 Thesis outline . . . . .	3
<b>2 Theory</b>	<b>5</b>
2.1 Representing Molecules . . . . .	5
2.2 Models . . . . .	6
2.3 Model training . . . . .	8
2.4 Evaluation metrics . . . . .	9
<b>3 Methods</b>	<b>11</b>
3.1 Workflow . . . . .	11
3.2 Data . . . . .	12
3.2.1 Datasets . . . . .	12
3.2.2 Data preprocessing . . . . .	15
3.3 Model training and evaluation . . . . .	15
<b>4 Results</b>	<b>17</b>
4.1 Single task models . . . . .	17
4.1.1 Initial models . . . . .	17
4.1.2 Optimized models . . . . .	20
4.1.3 Models with temporal split . . . . .	22
4.1.4 Acid and base 2 models . . . . .	24
4.2 Multitask models . . . . .	25
4.2.1 Internal data results . . . . .	25
4.2.2 Public data results . . . . .	27
4.3 Comparison to state of the art . . . . .	29
4.4 Potential to predict new modalities . . . . .	31
<b>5 Conclusion</b>	<b>35</b>
5.1 Discussion . . . . .	35

5.2 Conclusion . . . . .	36
<b>6 Future work</b>	<b>37</b>
<b>Bibliography</b>	<b>39</b>
<b>A Appendix</b>	<b>I</b>
A.1 Hyperparameters used in the optimized models . . . . .	I
A.2 Scatter plots of single-task and multitask models on the internal test set . . . . .	II
A.3 Scatter plots of multitask models on the SAMPL7 dataset . . . . .	V

# List of Figures

2.1	Different chemical representations of aspirin [14]. . . . .	5
2.2	Random forest. . . . .	7
2.3	Architecture of D-MPNN. . . . .	8
2.4	5-fold cross validation [25]. . . . .	9
3.1	Model training and analysis workflow used in this work. . . . .	11
3.2	Distribution of ionic centers in the internal dataset. . . . .	12
3.3	Distribution of molecular weights in the internal dataset. . . . .	13
3.4	Distribution of most acidic and basic $pK_a$ in the internal and public (Opera) datasets. . . . .	14
3.5	Distribution of second most acidic and basic $pK_a$ in the internal dataset. . . . .	14
4.1	Predicted and experimental acid 1 (left) and base 1 (right) $pK_a$ values of the internal test set using the best models. . . . .	18
4.2	Predicted and experimental acid 1 (left) and base 1 (right) $pK_a$ values of the internal test set using the best optimized models. . . . .	21
4.3	Predicted and experimental acid 1 (left) and base 1 (right) $pK_a$ values of the internal test set using the best temporal models. . . . .	23
4.4	Predicted and experimental $pK_a$ values on the test set of the optimized Chemprop model built using a scaffold-based split. . . . .	27
4.5	$R^2$ score of software and the temporal multitask model on the temporal test set. . . . .	29
4.6	$RMSE$ scores for the various proprietary softwares and the temporal multitask model using the temporal test set. . . . .	30
4.7	$R^2$ and $RMSE$ scores of different models on predicting new modalities. . . . .	33
A.1	Predicted and experimental acid 2 (left) and base 2 (right) $pK_a$ values of the internal test set using the best random split models. . . . .	II
A.2	Predicted and experimental acid 2 (left) and base 2 (right) $pK_a$ values of the internal test set using the best temporal models. . . . .	II
A.3	Predicted and experimental $pK_a$ values on the test set of the optimized Chemprop model built using a random split. . . . .	III
A.4	Predicted and experimental $pK_a$ values on the test set of the optimized Chemprop model built using a temporal split. . . . .	III
A.5	Predicted and experimental $pK_a$ values on the test set of the optimized Chemprop model built on small molecules. . . . .	IV

A.6	Predicted and experimental $pK_a$ values on the SAMPL7 test set of the Chemprop models built on a random split of the data. Model created using initial parameters on the left, optimized parameters on the right. . . . .	V
A.7	Predicted and experimental $pK_a$ values on the SAMPL7 test set of the Chemprop models built on a scaffold split of the data. Model created using initial parameters on the left, optimized parameters on the right. . . . .	V
A.8	Predicted and experimental $pK_a$ values on the SAMPL7 test set of the Chemprop models built on a temporal split of the data. Model created using initial parameters on the left, optimized parameters on the right. . . . .	VI

# List of Tables

3.1	Number of compounds in the clean internal and public (Opera) dataset.	15
4.1	5-fold cross-validation scores of baseline models trained on the internal dataset.	17
4.2	Test set scores of baseline models trained on the internal dataset.	18
4.3	Validation scores on the public dataset of baseline models trained on the internal dataset.	19
4.4	Best parameters for single-task LightGBM models trained on internal dataset.	20
4.5	5-fold cross-validation scores of tuned models trained on the internal dataset.	20
4.6	Validation scores on the internal test set and public dataset of tuned models trained on the internal dataset.	20
4.7	5-fold cross-validation scores of temporal models trained on the internal dataset.	22
4.8	Test set scores of temporal models trained on the internal dataset.	22
4.9	Validation scores on the public dataset of temporal models trained on the internal dataset.	22
4.10	Best parameters for the second most acidic and basic single-task LightGBM models trained on internal dataset.	24
4.11	5-fold cross-validation scores of acid and base 2 models trained on the internal dataset.	24
4.12	Test set scores of acid and base 2 models trained on the internal dataset.	24
4.13	Number of compounds in the train and test set of different splits in the internal dataset.	25
4.14	5-fold cross-validation scores of Chemprop models trained on the internal dataset.	26
4.15	Test set scores of Chemprop models trained on the internal dataset.	26
4.16	Validation scores on the public dataset of Chemprop models trained on the internal dataset.	27
4.17	Validation scores on the SAMPL7 dataset of Chemprop models trained on the internal dataset.	28
4.18	Number of modalities in the different splits with the total numbers in the bottom row.	31
4.19	Test set metrics by modality of model trained using a temporal split.	32
4.20	Test set metrics by modality of model trained using a scaffold split.	32

4.21	Test set metrics of model trained on small molecules. . . . .	32
A.1	Optimal hyperparameters for the single-task LightGBM models with RDKit2D descriptors. . . . .	I
A.2	Optimal hyperparameters for the multitask Chemprop models. . . . .	I

# 1

## Introduction

### 1.1 Background

Drug discovery and development is vital in order to treat diseases and clinical conditions, however, it is a complex, expensive and time-consuming process [1]. In order to accelerate the process and reduce high costs, various machine learning (ML) methods have been utilised at different stages in recent years [2], [3]. Machine learning has been used, for example, to help identify potential drug targets during the target identification phase, and to generate compounds that can interact with the target in the lead discovery and optimization phase.

A primary use of ML in drug development is molecular property prediction, since molecular properties can influence how the drug is absorbed, distributed, and excreted by the body. Such properties include lipophilicity, the ability of a chemical compound to dissolve in fats and oils; hydrogen-bonding capability, the ability of a molecule to form hydrogen bonds with other molecules; and polarity, the separation of electric charge within a molecule.

Molecular properties are highly influenced by the acid dissociation equilibrium constant  $K_a$ , also called the ionization constant.  $K_a = \frac{[A^-][H^+]}{[HA]}$ , where quantities in square brackets represent the concentrations of the species at equilibrium [4]. It is a measure of the extent to which an acid dissociates in solution, therefore indicating the strength of an acid.

$K_a$  is most often represented as the negative logarithm  $pK_a$  ( $pK_a = -\log_{10}K_a$ ).  $pK_a$  influences most aspects of drug discovery, and is especially influential in the aqueous solubility of a drug as it indicates the strength of an acid.  $pK_a$  values affect for example, the stability, permeability, and ADMET (absorption, distribution, metabolism, excretion and toxicity) profiles of a compound, making it one of the most important parameters in drug discovery [5]. However, experimentally measuring the  $pK_a$  of compounds is a time-consuming and limited procedure, therefore, it is of utmost importance to use predictive methods to obtain required  $pK_a$  values.

The Hammett equation has been one of the most widely used empirical methods for  $pK_a$  prediction [6]. It consists of two parameters, a substituent constant  $\sigma$  and an equilibrium constant  $\rho$  which form the following equation:  $pK_a = A - \rho(\sum \sigma)$ , where  $A$  is the  $pK_a$  of the unsubstituted acid or base [7]. Despite the popularity there are several disadvantages to this method, such as, it's limited scope and high reliance on experimental data.

With the evolution of artificial intelligence and different machine learning methods, and more extensive data collection and digitalization,  $pK_a$  prediction became faster and more accurate over the last decades. Besides using classical machine learning approaches including Support Vector Machines and Random Forests, the focus in recent years has been shifted towards deep machine learning methods, such as, Graph Neural Networks.

Quantitative structure-activity relationship (QSAR) models have been used for decades to predict different physicochemical parameters, including  $pK_a$  [5]. These models are highly dependent on the quality and quantity of data, therefore the growing amount of publicly available datasets have helped to develop new *in silico*<sup>1</sup> methods. One such public dataset [8] was used by Mansouri *et al.* to create three  $pK_a$  prediction models with the machine learning methods of (1) support vector machines (SVM) combined with k-nearest neighbors (kNN), (2) extreme gradient boosting (XGB), and (3) deep neural networks (DNN). Even though the performance of these models were defined as reasonably good, they lack the ability to simultaneously predict acidic and basic  $pK_a$  values of a compound.

Message Passing Neural Networks (MPNNs) provide a generalised framework for the task of supervised learning with Graph Neural Networks (GNNs) [9]. By abstracting common features of existing GNNs, Gilmer *et al.* developed novel variations which feed on the topology of the molecules. They achieved state of the art results and chemical accuracy with several models, without the need of complicated feature engineering. These MPNNs, however, do not generalize well to large graphs in which further improvements are needed.

Accurately predicting the value of  $pK_a$  is highly relevant and thus has been the subject of multiple blind predictive challenges held in recent years, such as the Statistical Assessment of Modeling of Proteins and Ligands (SAMPL) challenges [10], [11]. SAMPL physical property challenges concentrate on computational modeling areas in need of improvement and thereby provide appropriate guidelines to design drugs with optimal properties. The SAMPL7 challenge focused on predicting the molecular properties  $\log P$  and  $pK_a$  for 22 molecules [11]. There were 9 submitted models, and the two best performing methods were both Quantum mechanics (QM) based. Even though, QM based approaches provide reasonable prediction accuracies, the computational costs involved with using these methods always remain a challenge.

---

<sup>1</sup>experimentation performed by computer



## 1.2 Goals and challenges

$pK_a$  plays a particularly important role in molecular design and drug discovery. Nevertheless, predicting this property remains a challenge, owing to the various ionization states a molecule can adopt at a specific pH range. Even though, multiple models currently exist and show promising results for predicting  $pK_a$  on small molecules, they often fail and can be slow for larger molecules.

The aim of this thesis is to, through exploring a wide-array of chemical features and machine learning approaches, extend the scope of  $pK_a$  prediction using both internal and publicly available datasets. We achieve this by building a predictive model using an internal dataset and evaluate it with both internal and publicly available datasets. Additionally, we investigate the potential of the model to predict the  $pK_a$  values of large molecules.

## 1.3 Thesis outline

The remainder of the thesis is organized as follows. Chapter 2 presents a theoretical overview discussing the representation of molecules, different models utilized, the process of model training including the different splits used, and the evaluation metrics employed to assess performance.

In Chapter 3, the methods utilized in the thesis are described in detail. It provides insights into the workflow followed throughout the work, including the datasets used, data preprocessing techniques applied, and the model training and evaluation process.

Chapter 4 presents the results obtained from the experiments. First, it showcases the outcomes of the single task models, discussing the initial models, optimized models, models with a temporal split, and the acid and base 2 models. Following that, the next section describes the performance of the different multitask models. Additionally, a comparison with the state of the art is presented, highlighting the strengths and potential of the developed models. Furthermore, the potential of the models to predict new modalities is discussed.

In Chapter 5, the thesis concludes with a comprehensive discussion and conclusion. This section provides a thorough analysis and interpretation of the results obtained, with a comparison of the different models. It also summarizes the main contributions of the thesis.

Lastly, Chapter 6 offers insights into future research directions and potential improvements.



# 2

## Theory

### 2.1 Representing Molecules

Molecules can be represented, for instance, as chemical or structural formulas. These are easily understandable by humans but much harder to interpret for computers. Therefore, the more widespread use of machine learning methods in drug discovery has led to the development of chemical representations that can be processed by computers [12]. On a computer, molecular data can be represented, for example, in the form of graphs, line notations, descriptors, and molecular fingerprints. While some of the representations encode the exact structure of a compound, others capture different properties of the molecules.

One of the most widely used line notations is the Simplified Molecular Input Line Entry System (SMILES) [13]. SMILES describe the two-dimensional structure of molecules in a compact way, as a linear string of characters.

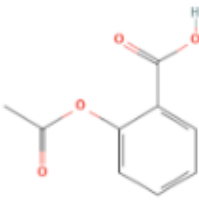
<b>Chemical formula</b>	C <sub>9</sub> H <sub>8</sub> O <sub>4</sub>
<b>Structural formula</b>	
<b>SMILES</b>	<chem>CC(=O)OC1=CC=CC=C1C(=O)O</chem>

Figure 2.1: Different chemical representations of aspirin [14].

Molecular fingerprints represent chemical structures by encoding the structure and properties of a molecule. These descriptors can be calculated from one-, two-, or three-dimensional representations of a structure incorporating different properties of the molecule [15]. Some of the most commonly used two-dimensional molecular fingerprints are the Extended-Connectivity Fingerprint (ECFP, also known as the Morgan fingerprint) [16], and the Maccs (Molecular ACCess System) fingerprint [17].

ECFP is a circular fingerprint that represents molecules as a bit vector, by capturing features of the molecular graph. These features refer to the presence or absence of certain substructures in a molecule. The ECFP generation process is based on the Morgan algorithm and includes three stages: identifier assignment to atoms, iterative updating based on each atoms neighbors, and duplicate identifier removal. Different ECFPs can be generated by specifying the diameter of the largest feature. This is reflected by the appended number which is equal to twice the number of iterations performed. For example, if 2 iterations are performed, the largest possible fragment will have a width of 4 bonds, and the fingerprint name will end in 4, e.g., ECFP4.

Maccs keys are commonly used structural fingerprints with 166 bit keys where each bit encodes a pre-defined substructure pattern, such as functional groups, ring systems, and other molecular features. The result is a list of binary values where bits can either be 1, meaning the given substructure is present or 0, absent in the molecule.

RDKit 2D descriptors [18] are calculated based on the two dimensional structure of a molecule encoding it's physicochemical properties. These include for example, physical properties, atom and bond counts, partial charge, and adjacency and distance matrix descriptors.

## 2.2 Models

Random forests [19] are ensembles of decision trees, where each tree is trained on its own sampled training set. The splitting of the nodes is also done using a random subset of the features, thus introducing more randomness into the models. Random forests can be used for both regression and classification problems. The trees individually vote for what the most probable class, and a decision is made based on the votes. For classification, the final class is determined by majority voting, and for regression by averaging.

Boosted trees [20] are ensembles which sequentially combine multiple weak trees to create a stronger model. Different approaches for boosting are, for example, adaptive boosting and gradient boosting.

Adaptive boosting aims to improve the performance of a weak classifier by focusing on previously misclassified data points. At each iteration, the algorithm assigns weights to the data points in the training set based on their classification accuracy, and then trains a new classifier. The weight of misclassified data points is increased and the weight of correctly classified data points is decreased. Finally, the output of the weak classifiers is combined using a weighted sum.

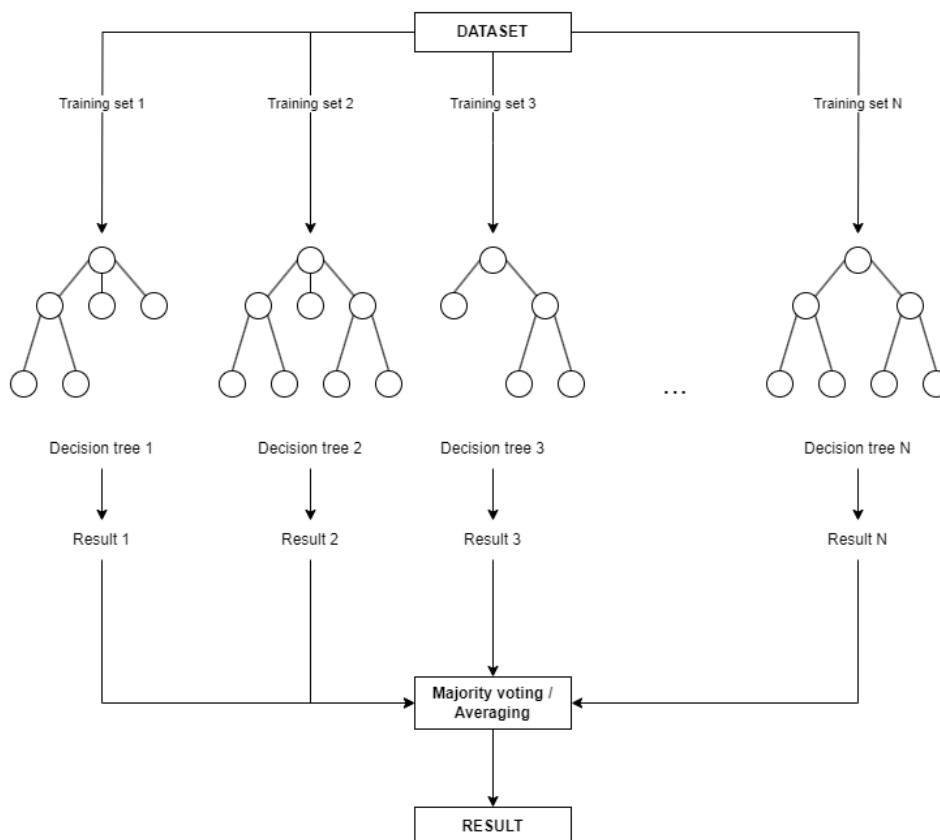


Figure 2.2: Random forest.

In gradient boosting, the objective is to minimize the loss function of the model by adding weak learners using gradient descent. The algorithm starts by training a base model and then each subsequent tree is trying to correct the errors of the previous one, thus improving the predictive power of the ensemble. A weight is assigned to each tree based on their performance, and then all trees are combined creating a boosted model.

Graph Neural Networks (GNNs) [21] are deep learning models that can handle graph structured data. They can be used for node, edge, and graph-level predictions, in both supervised and unsupervised tasks. Through a message passing process, GNNs embed information into each node about its neighbours. This can then be used to find patterns and make predictions. GNNs can be divided into different categories, such as Recurrent GNNs which aim to learn node representations with recurrent neural architectures, and Convolutional GNNs which generalize the convolution operation from grid to graph data.

Directed MPNN (D-MPNN) [22] is a Convolutional GNN specifically designed for molecular property prediction. It is a variation of the MPNN architecture that operates on undirected graphs using a message passing mechanism. The D-MPNN model consists of two phases, a message passing phase that uses an encoder, and a readout phase with feed-forward layers. In the message passing phase, a neural representation is built of the molecule by processing the molecular graph. Nodes of

the graph represent atoms, and directed edges represent bonds between two atoms. Different to MPNN, D-MPNN uses messages associated with directed bonds rather than atoms to avoid adding noise to the graph representation. In the readout phase, the final representation of the molecule is used to predict the properties of interest.

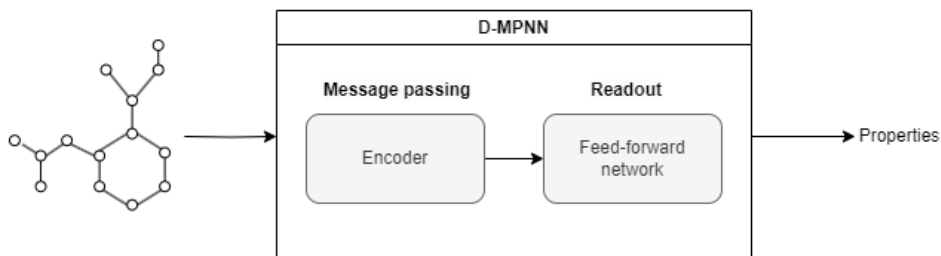


Figure 2.3: Architecture of D-MPNN.

## 2.3 Model training

In order to train the machine learning models, we will first divide the dataset into a train and test set which can be done in multiple ways. The most commonly used method is a random split, where a portion of the data is randomly selected and set aside for testing. Although, a model trained on such data might perform good on the randomly selected test set, it often fails for new data collected in a different context than the initial dataset. For example at a later point in time, or by a different company or lab. This could be due to the new dataset being from a different distribution than the one the model was trained on. Therefore it is important to also try out methods that generalize better to unseen data.

One such approach is a scaffold-based split [23], a method of splitting a molecular dataset based on molecular scaffolds of the chemical structures. A scaffold reduces the structure of a compound to its core components meaning that multiple compounds can have the same scaffold. The dataset is split in a way that compounds with different scaffolds are in the test set than in the train set making them as distinct as possible.

Another method that can better generalize to unseen data is the time-based (temporal) split, where the data is split based on a timestamp. In this case, the model is trained on older samples and tested on newer ones ensuring that no future data is used to predict previous data. It is especially useful for tasks where the data changes over time as it considers the changing distributions over time.

Besides splitting the data into a train and test set, cross-validation (CV) will be implemented [24]. In a k-fold CV the training set is further split into k sets (folds). For each fold a model is trained using k-1 folds and validated on 1 fold, see Figure 2.4. During temporal CV, the model is trained on the initial n folds and assessed on fold n+1, guaranteeing that the training data precedes the testing data in each fold. The error metrics of the k models are then averaged to determine how well the model can generalize to unseen data.

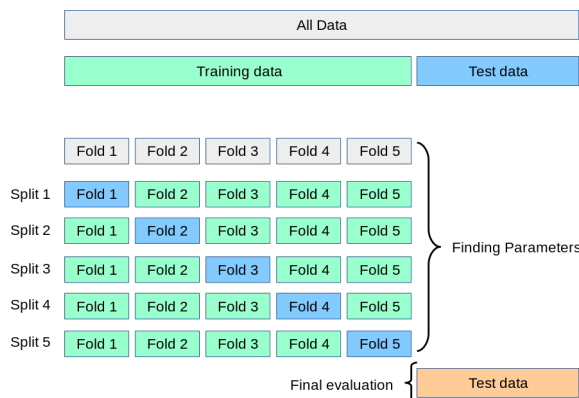


Figure 2.4: 5-fold cross validation [25].

To find the optimal model and further increase accuracy, hyperparameter optimization will be implemented. To do this we will use Optuna, an open-source optimization software [26] for the single-task models, and Chemprop’s built-in optimizer, using Bayesian optimization, for the multitask models.

## 2.4 Evaluation metrics

To evaluate the models, the following error metrics have been used in previous studies [5], [11], [27]: root mean square error ( $RMSE$ ), mean (signed) error ( $ME$ ), mean absolute error ( $MAE$ ), coefficient of determination ( $R^2$ ), linear regression slope ( $m$ ), and Kendalls Tau rank correlation coefficient ( $\tau$ ). Given  $x_i$  as the actual and  $y_i$  as the predicted value of an observation, these error metrics can be calculated as follows [28], [29].

The root mean square error is the square root of the variance of the residuals (prediction errors),  $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2}$ .

The mean (signed) error is the average of all the errors,  $ME = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)$ , while the mean absolute error is the average of the absolute errors,  $MAE = \frac{1}{n} \sum_{i=1}^n |x_i - y_i|$ .

The coefficient of determination measures how well the model predicts an outcome,  $R^2 = 1 - \frac{\sum_{i=1}^n (y_i - x_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$ , where  $\bar{x} = \frac{1}{n} \sum_{i=1}^n (x_i)$ .

The linear regression slope indicates the rate of change in  $y$  relative to  $x$ ,  $m = \frac{\Delta y}{\Delta x}$ .

Kendalls Tau rank correlation coefficient can be determined as  $\tau = \frac{C-D}{C+D}$ , where  $C$  denotes the number of concordant pairs, and  $D$  denotes the number of discordant pairs.





# 3

## Methods

### 3.1 Workflow

The workflow of creating the models consists of the following steps: data preparation, train-test set split, molecular descriptor calculation, model training, and model evaluation. These steps will be discussed in more detail in the next sections.

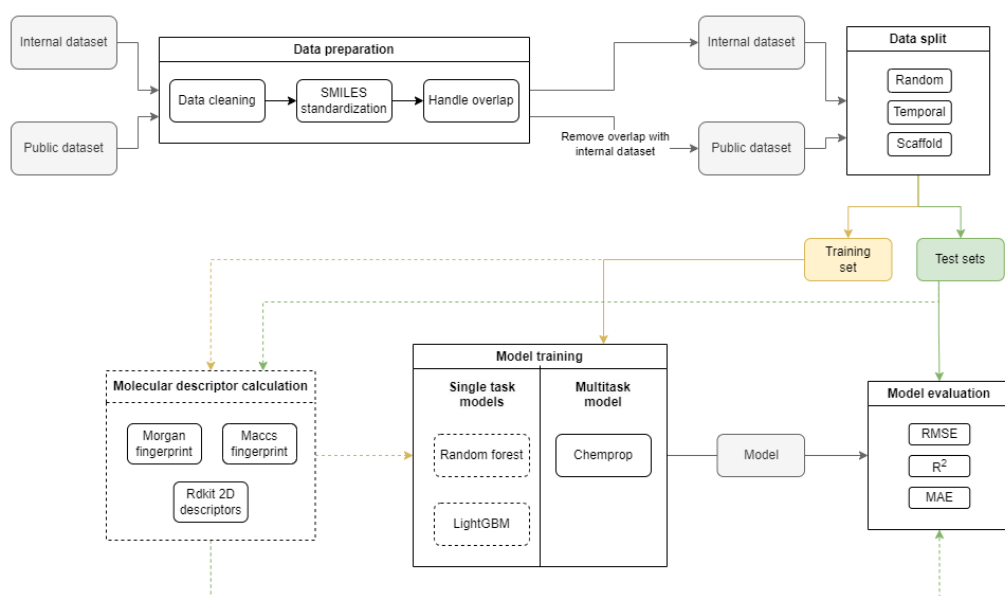


Figure 3.1: Model training and analysis workflow used in this work.

## 3.2 Data

### 3.2.1 Datasets

For benchmarking, the data provided for the SAMPL6 [10] and SAMPL7 [11] blind predictive challenges, and QSAR models [5] will be used.

The SAMPL6 and SAMPL7 datasets consist of 24 and 22 molecules, respectively. In the SAMPL6 dataset each compound has one, two or three  $pK_a$ , whereas the SAMPL7 dataset only lists one  $pK_a$  for each molecule. The molecules are represented in SMILES strings, and can be downloaded from the corresponding SAMPL challenge website [30], [31].

The dataset from the QSAR models (Opera dataset) contains  $pK_a$  data measured for 7912 chemicals and their corresponding SMILES strings [32]. Among these, 436 compounds have both an acidic and basic  $pK_a$ , for the rest only one is provided. The data was originally obtained from multiple sources from literature, however, no references are supporting the  $pK_a$  values. The authors of the paper also note a high diversity in the data and the different methods used to measure  $pK_a$ .

Based on this data, three different datasets were created and are publicly available: 1) Option 1: all chemicals with replicates removed, 2) Option 2: low variability replicates included, and 3) Option 3: all data included.

Apart from the publicly available datasets, we will also use a high quality internal dataset with known experimental acidic and basic  $pK_a$  values of approximately 20k-25k compounds. As show in Figure 3.2, most compounds have 1 or 2 ionic centers, therefore we will limit our experiments to the first and second most acidic and basic  $pK_a$  values.

Throughout the report, we will refer to the most acidic and basic  $pK_a$  as acid / base 1, and to the second most acidic and basic  $pK_a$  as acid / base 2.

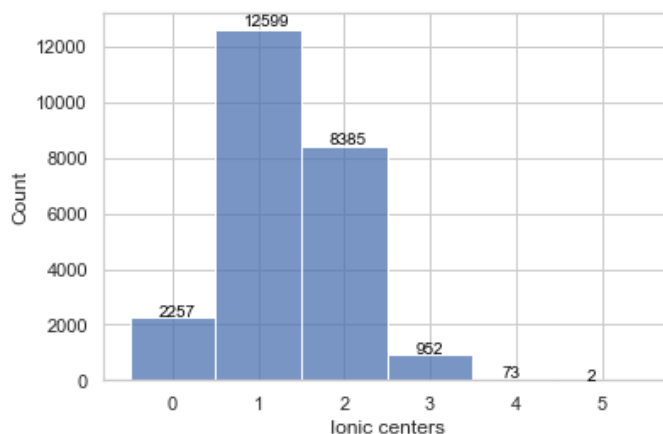


Figure 3.2: Distribution of ionic centers in the internal dataset.

In Figure 3.3 the distribution of molecular weights in the internal dataset is presented. It shows that most compounds have a low molecular weight ...

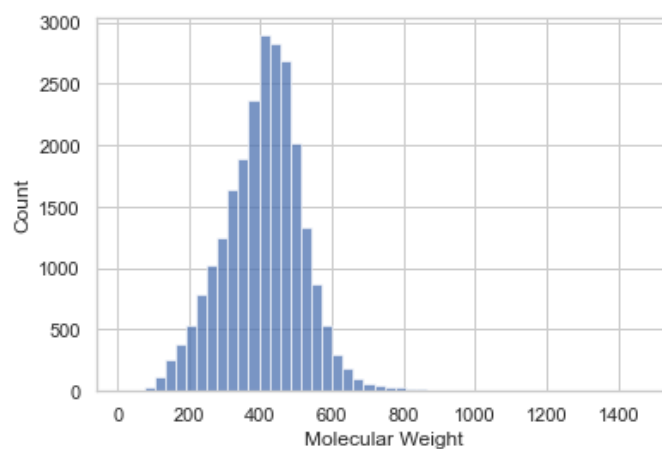


Figure 3.3: Distribution of molecular weights in the internal dataset.

Figure 3.4 presents the distribution of acid 1 and base 1  $pK_a$  in the internal and public (Opera) datasets. As shown, the internal and public datasets have similar distributions within similar ranges, however there is significantly more internal than public data available.

Figure 3.5 presents the distribution of acid 2 and base 2  $pK_a$  in the internal dataset. As we can see, there is much less data available for the second most acidic and basic  $pK_a$  than the first ones, moreover, the values are distributed in a smaller range.

### 3. Methods

---

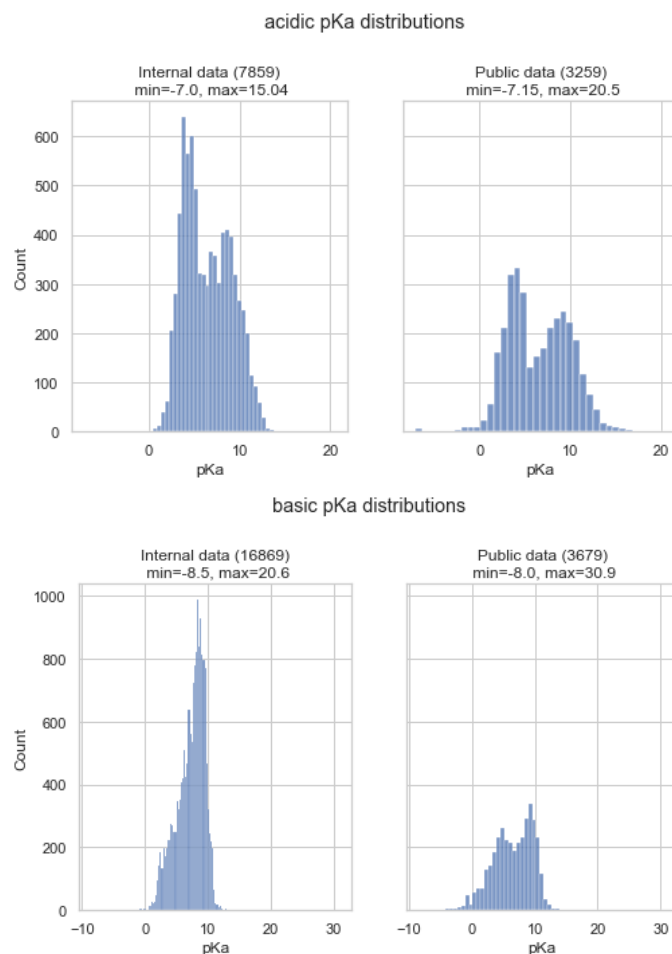


Figure 3.4: Distribution of most acidic and basic  $pK_a$  in the internal and public (Opera) datasets.

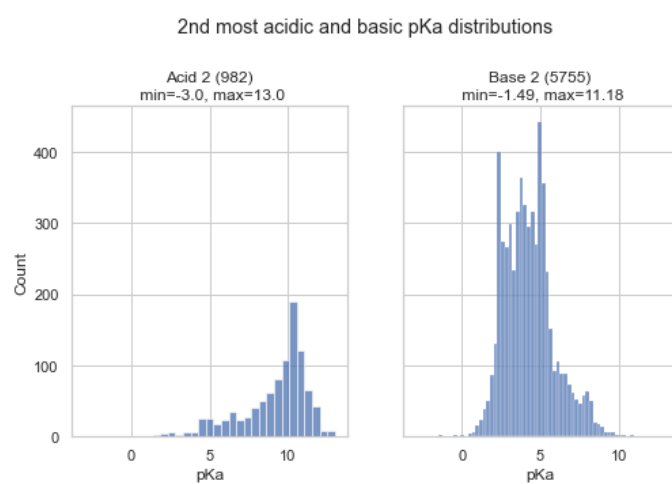


Figure 3.5: Distribution of second most acidic and basic  $pK_a$  in the internal dataset.

### 3.2.2 Data preprocessing

As part of the data preparation process, we first performed data cleaning on the datasets. From the internal dataset, we excluded all acid 2 / base 2 values without an acid 1 / base 1  $pK_a$  and then removed all compounds without experimental values. From the public (Opera) dataset, we chose the *Option 1: all chemicals with replicates removed* [5] refined dataset and only kept compounds with a  $pK_a$  between 0 and 14.

In the next step, all internal and public SMILES were standardized, and invalid ones were removed. Then, since multiple molecules can share the same standardized SMILES, we removed the ones where the standard deviation of the overlaps was greater than 1, and averaged the others. Lastly, we removed all compounds from the public dataset that were also present in the internal dataset. This resulted the following number of compounds for the different categories, presented in Table 3.1.

Dataset	Acid 1	Acid 2	Base 1	Base 2
internal	7787	840	16504	5362
public	2823	-	2979	-

Table 3.1: Number of compounds in the clean internal and public (Opera) dataset.

For each dataset (internal and public) different molecular descriptors, namely Morgan fingerprint (ECFP4), Maccs fingerprint and RDKit 2D descriptors have been calculated, resulting in a total of six descriptor datasets. Following the computation of the descriptors, we removed descriptors with a low variance (less than 2%) and correlated features (with a correlation greater than 0.95). The resulting descriptors were then used for creating and testing the single-task models, while the standardised SMILES strings for creating and testing the multitask models.

## 3.3 Model training and evaluation

For training and testing the models, the data was split into a train and test set consisting of 80% and 20% of the data. Three different approaches were utilized for the split: random split, scaffold-based split, and temporal split. The training sets were utilized for (5-fold) cross-validation and hyperparameter optimization, while the test sets were used to assess the final performance of the model. For models with a temporal split, the cross validation was also carried out in a temporal way. For model evaluation, the main metrics used were the RMSE score and the coefficient of determination ( $R^2$ ).

During the experiments we explored both single-task and multitask models. Since single-task models can only predict one value, separate models were built for acidic and basic  $pK_a$ . Multitask models, on the other hand, can predict multiple values, therefore a single model was sufficient to predict all  $pK_a$  values of a compound.

To create the single-task models, the implementations of Scikit-learn’s RandomForestRegressor [33] and LightGBM [34] were used. For the multitask models, a message

### 3. Methods

---

passing neural network created for molecular property prediction called Chemprop [22] was utilized.

# 4

## Results

### 4.1 Single task models

#### 4.1.1 Initial models

To create baseline models, we first used a Random Forest and a LightGBM regressor model with each descriptor to predict the most acidic and basic  $pK_a$ . For all models the same random split was used in the dataset. Besides the previously mentioned data preparation process, for the Random Forest model we also had to remove all compounds containing NaN values in the descriptors as RF models cannot handle missing values.

In Table 4.1 and 4.2 the 5-fold cross-validation scores and test set scores of the baseline models trained on the internal dataset are presented. It can be seen that the performance of the two models is really similar, however, we observed that the training of LightGBM is much faster than the training of Random Forest models. Therefore, in case of equal performance, the LightGBM model is preferred. Regarding the different descriptors, the best performance could be achieved with RDKit’s 2D descriptors, and the ECFP4 fingerprint. Highlighted are the best achieved metrics for the most acidic and basic  $pK_a$ .

Descriptor	Model	Acid		Base	
		$R^2$	$RMSE$	$R^2$	$RMSE$
ECFP4	Random Forest	<b>0.72</b>	<b>1.40</b>	0.67	1.28
	LightGBM	0.70	1.44	0.65	1.32
RDKit2D	Random Forest	0.71	1.42	0.64	1.34
	LightGBM	<b>0.72</b>	<b>1.40</b>	<b>0.67</b>	<b>1.27</b>
Maccs	Random Forest	0.72	1.40	0.66	1.30
	LightGBM	0.70	1.45	0.61	1.38

Table 4.1: 5-fold cross-validation scores of baseline models trained on the internal dataset.

## 4. Results

Descriptor	Model	Acid		Base	
		$R^2$	$RMSE$	$R^2$	$RMSE$
ECFP4	Random Forest	<b>0.72</b>	<b>1.34</b>	<b>0.67</b>	<b>1.30</b>
	LightGBM	0.70	1.39	0.64	1.37
RDKit2D	Random Forest	0.72	1.36	0.64	1.36
	LightGBM	<b>0.72</b>	<b>1.34</b>	0.66	1.32
Maccs	Random Forest	0.71	1.36	0.66	1.33
	LightGBM	0.71	1.36	0.61	1.42

Table 4.2: Test set scores of baseline models trained on the internal dataset.

Figure 4.1 shows the performance of the best models, on the internal test set. The models were selected based on the CV results, and were trained with the RDKit2D descriptors using the LightGBM model. In each plot, the x axis corresponds to the experimental, and the y axis to the predicted  $pK_a$  value of the compounds. The line of best fit is also shown.

The plots show that the predictions lie in a similar range as the experimental values, however, there are a number of outliers for both acidic and basic  $pK_a$ .

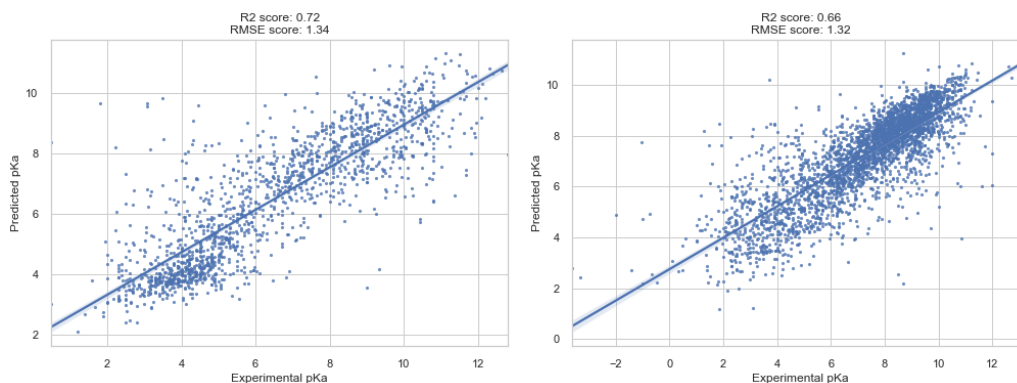


Figure 4.1: Predicted and experimental acid 1 (left) and base 1 (right)  $pK_a$  values of the internal test set using the best models.

Besides training and testing the models on the internal dataset, we also used the public (Opera) dataset as a validation set to evaluate the models. Table 4.3 presents the  $R^2$  and  $RMSE$  scores of the internal models tested on the public dataset. As shown, all models perform worse on the public test set than on the internal one. We observe a 17-27% drop in  $R^2$  for acidic and basic  $pK_a$  for the best models.



---

Descriptor	Model	Acid		Base	
		$R^2$	$RMSE$	$R^2$	$RMSE$
ECFP4	Random Forest	0.37	2.50	0.39	2.25
	LightGBM	0.40	2.43	0.41	2.22
RDKit2D	Random Forest	0.42	2.39	0.40	2.23
	LightGBM	<b>0.45</b>	<b>2.32</b>	<b>0.49</b>	<b>2.06</b>
Maccs	Random Forest	0.39	2.45	0.41	2.21
	LightGBM	0.39	2.44	0.47	2.10

Table 4.3: Validation scores on the public dataset of baseline models trained on the internal dataset.

### 4.1.2 Optimized models

To optimize the models, we implemented hyperparameter tuning using Optuna for the ECFP4 and RDKit descriptors. The hyperparameters tuned were the number of estimators in the range of 10 and 1000, and the maximum depth of the tree between 2 and 32.

The best model for each descriptor was a LightGBM model with the parameters presented in Table 4.4.

Descriptor	Acid		Base	
	n_estimators	max_depth	n_estimators	max_depth
ECFP4	423	25	1000	32
RDKit2D	843	9	983	29

Table 4.4: Best parameters for single-task LightGBM models trained on internal dataset.

Using the best hyperparameters, the performance of the models could be improved on the internal dataset, as shown in Table 4.5 and 4.6.

For acidic  $pK_a$  we observe a 3%, and for basic  $pK_a$  a 5% improvement in  $R^2$  on the test set. On the public test set, the optimized models show similar performance as the initial ones.

Descriptor	Acid		Base	
	$R^2$	$RMSE$	$R^2$	$RMSE$
ECFP4	0.72	1.39	0.72	1.18
RDKit2D	0.74	1.35	0.72	1.17

Table 4.5: 5-fold cross-validation scores of tuned models trained on the internal dataset.

Descriptor	Test set	Acid		Base	
		$R^2$	$RMSE$	$R^2$	$RMSE$
ECFP4	internal	0.74	1.31	0.71	1.23
	public	0.39	2.44	0.43	2.17
RDKit2D	internal	0.75	1.28	0.71	1.23
	public	0.43	2.37	0.48	2.07

Table 4.6: Validation scores on the internal test set and public dataset of tuned models trained on the internal dataset.

In Figure 4.2, the performance of the best models can be seen on the internal test set. Comparing to the initial best models, the optimized models show slightly better results with less outliers.

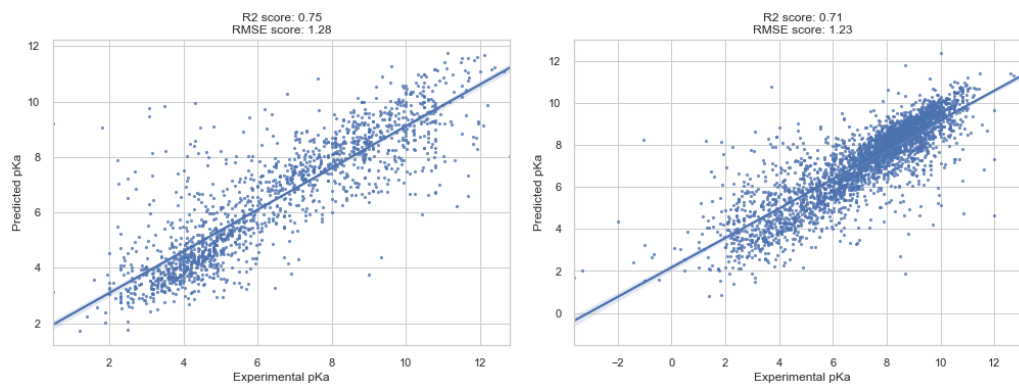


Figure 4.2: Predicted and experimental acid 1 (left) and base 1 (right)  $pK_a$  values of the internal test set using the best optimized models.

### 4.1.3 Models with temporal split

On the internal dataset, we performed a time-based split where the newest 20% of the compounds were used as a test set and the remaining compounds as a training set. For the experiments, we only used LightGBM models both with the initial model parameters and optimized parameters based on the previous hyperparameter tuning.

As shown in Table 4.7, 4.8 and 4.9, using a time-based split yielded worse results than randomly splitting the data into a train and test set. In the CV scores, we observed a drop of more than 20% in  $R^2$  and an increase of around 30% in  $RMSE$  for both acidic and basic  $pK_a$ . This is also reflected on the internal test set. However, models trained with a temporal split performed similarly to models with a random split in the data on the public test set.

Descriptor	Parameters	Acid		Base	
		$R^2$	$RMSE$	$R^2$	$RMSE$
ECFP4	initial	0.50	1.75	0.47	1.56
	optimized	0.50	1.75	0.48	1.54
RDKit2D	initial	0.51	1.73	0.50	1.51
	optimized	<b>0.51</b>	<b>1.73</b>	<b>0.53</b>	<b>1.47</b>

Table 4.7: 5-fold cross-validation scores of temporal models trained on the internal dataset.

Descriptor	Parameters	Acid		Base	
		$R^2$	$RMSE$	$R^2$	$RMSE$
ECFP4	initial	0.53	1.99	0.43	1.65
	optimized	0.50	2.04	0.47	1.59
RDKit2D	initial	0.57	1.90	0.51	1.54
	optimized	0.57	1.89	0.53	1.50

Table 4.8: Test set scores of temporal models trained on the internal dataset.

Descriptor	Parameters	Acid		Base	
		$R^2$	$RMSE$	$R^2$	$RMSE$
ECFP4	initial	0.38	2.46	0.42	2.19
	optimized	0.38	2.47	0.44	2.16
RDKit2D	initial	0.43	2.36	0.49	2.06
	optimized	0.44	2.34	0.48	2.08

Table 4.9: Validation scores on the public dataset of temporal models trained on the internal dataset.

Figure 4.3 shows the performance of the best temporal models on the internal test set. Comparing to the plots of the models with a random split, the points on these plots are more scattered and show greater deviation from the experimental values.

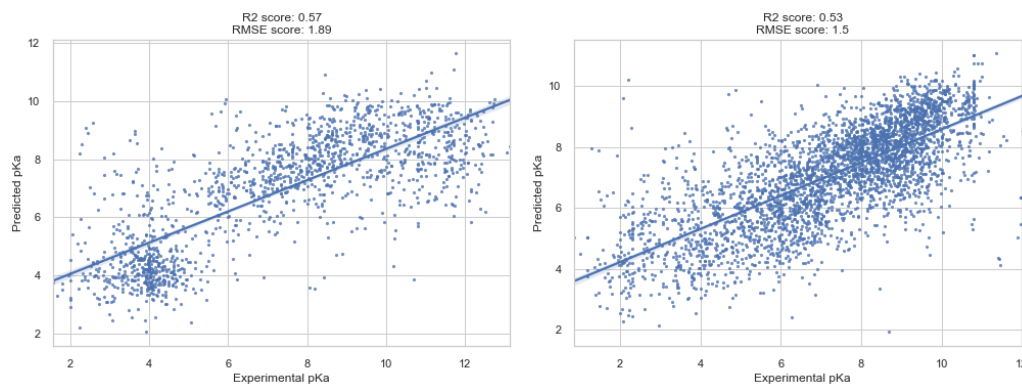


Figure 4.3: Predicted and experimental acid 1 (left) and base 1 (right)  $pK_a$  values of the internal test set using the best temporal models.

#### 4.1.4 Acid and base 2 models

The models presented earlier only predict the most acidic and basic  $pK_a$  of the compounds. To establish a basis for comparison with the multitask models, we created baseline models for the second most acidic and basic  $pK_a$  values as well. To do this, we used a LightGBM model with the RDKit descriptors as this combination yielded the best result in previous experiments. Hyperparameter optimization was also performed, with the best parameters presented in Table 4.10.

	n_estimators	max_depth
Acid 2	186	2
Base 2	231	8

Table 4.10: Best parameters for the second most acidic and basic single-task LightGBM models trained on internal dataset.

Table 4.11 and 4.12 present the 5-fold CV and test set scores of the models trained using a random and temporal split in the data, initialized with the initial (default) and optimized hyperparameters. As shown, models trained on the second most acidic and basic  $pK_a$  of compounds have significantly lower performance than models trained with the most acidic and basic  $pK_a$ .

Split	Parameters	Acid 2		Base 2	
		$R^2$	RMSE	$R^2$	RMSE
Random	initial	0.47	1.51	0.56	1.05
	optimized	0.47	1.52	0.58	1.02
Temporal	initial	-0.10	1.86	0.19	1.28
	optimized	-0.01	1.81	0.17	1.29

Table 4.11: 5-fold cross-validation scores of acid and base 2 models trained on the internal dataset.

Split	Parameters	Acid 2		Base 2	
		$R^2$	RMSE	$R^2$	RMSE
Random	initial	0.74	1.09	0.61	1.01
	optimized	0.73	1.11	0.62	1.00
Temporal	initial	0.09	1.63	0.22	1.44
	optimized	0.19	1.54	0.20	1.46

Table 4.12: Test set scores of acid and base 2 models trained on the internal dataset.

## 4.2 Multitask models

To simultaneously predict all  $pK_a$  values of the compounds, we created different multitask models utilizing the D-MPNN introduced in Section 2.2. We generated models with a random, scaffold-based and time-based split in the internal data, running each model for 50 epochs.

In Table 4.13 the number of compounds in the train and test set of different splits for the internal dataset is presented.

Split	Acid 1		Acid 2		Base 1		Base 2	
	train	test	train	test	train	test	train	test
Random	6201	1586	682	158	13230	3274	4285	1077
Scaffold	6251	1536	668	172	13259	3245	4397	965
Temporal	6251	1536	676	164	12983	3521	3630	1732

Table 4.13: Number of compounds in the train and test set of different splits in the internal dataset.

To create the models, we used both the initial parameters of the network and performed hyperparameter optimization. The hyperparameters optimized were the hidden size, depth, dropout, and number of feed-forward layers of the network.

The best parameters found during optimization were the following:

- depth: 3
- dropout: 0.3
- hidden size: 2100
- number of feed-forward layers: 3

These parameters were used for all models created with different splits in the data, referred to as *optimized*.

### 4.2.1 Internal data results

Tables 4.14 and 4.15 present the 5-fold cross-validation and test set scores of the multitask models trained on the internal dataset. The results show that the most acidic and basic  $pK_a$  (acid and base 1) can be predicted with higher accuracy than the second most acidic and basic  $pK_a$  (acid and base 2) of the compounds. It can also be seen that the optimized models generally perform better than the initial ones for all split types.

We also observe, that models created with a scaffold-based split yield similar CV results as models using a random split, even outperforming it in most categories. Models created using a temporal split have slightly lower performance in acid and base 1, and much worse performance in acid and base 2 compared to using random or scaffold-based splits.

## 4. Results

Split	Parameters	Acid 1		Acid 2		Base 1		Base 2	
		$R^2$	$RMSE$	$R^2$	$RMSE$	$R^2$	$RMSE$	$R^2$	$RMSE$
Random	initial	0.78	1.22	0.50	1.39	0.81	0.98	0.66	0.96
	optimized	0.80	1.16	0.48	1.39	0.82	0.96	0.67	0.95
Scaffold	initial	0.81	1.14	0.44	1.47	0.82	0.95	0.64	0.96
	optimized	<b>0.83</b>	<b>1.09</b>	<b>0.55</b>	<b>1.33</b>	<b>0.82</b>	<b>0.94</b>	<b>0.67</b>	<b>0.92</b>
Temporal	initial	0.62	1.53	-0.75	1.95	0.73	1.11	0.36	1.17
	optimized	0.63	1.52	-0.35	1.82	0.74	1.08	0.32	1.20

Table 4.14: 5-fold cross-validation scores of Chemprop models trained on the internal dataset.

Split	Parameters	Acid 1		Acid 2		Base 1		Base 2	
		$R^2$	$RMSE$	$R^2$	$RMSE$	$R^2$	$RMSE$	$R^2$	$RMSE$
Random	initial	0.79	1.16	0.66	1.26	0.79	1.05	0.66	0.95
	optimized	<b>0.81</b>	<b>1.12</b>	<b>0.72</b>	<b>1.13</b>	0.81	0.99	<b>0.67</b>	<b>0.94</b>
Scaffold	initial	0.76	1.28	0.39	1.62	0.83	0.92	0.64	1.00
	optimized	0.79	1.20	0.45	1.53	<b>0.84</b>	<b>0.87</b>	0.62	1.02
Temporal	initial	0.76	1.42	0.25	1.49	0.74	1.12	0.31	1.35
	optimized	0.74	1.48	0.18	1.56	0.76	1.08	0.32	1.34

Table 4.15: Test set scores of Chemprop models trained on the internal dataset.

Figure 4.4 shows the test set performance of the best performing model based on the CV results. The presented plots correspond to the optimized scaffold-split model as this model achieved the best CV scores. As shown, for the most acidic and basic  $pK_a$  the predictions are quite accurate with 77% of acidic and 86% of basic  $pK_a$  predictions being within 1  $pK_a$  unit. Compared to this, on the second most acidic and basic  $pK_a$  we observe a higher deviation in predictions and more outliers with 68% of acidic and 78% of basic  $pK_a$  predictions being within 1  $pK_a$  unit.



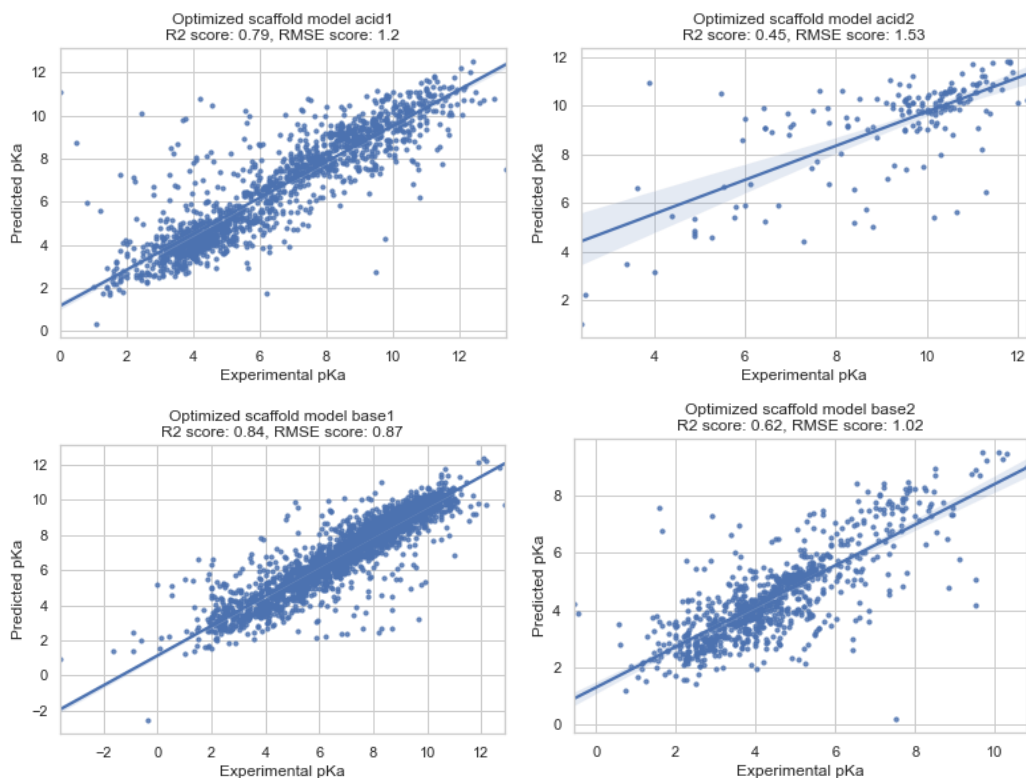


Figure 4.4: Predicted and experimental  $pK_a$  values on the test set of the optimized Chemprop model built using a scaffold-based split.

## 4.2.2 Public data results

In Table 4.16 the results of the Chemprop models tested on the public dataset are presented. As shown, models with the optimized hyperparameters perform better than models trained with the initial parameters for all splits. Compared to the internal test set results, we observe a 30% drop in  $R^2$ . We also note that different to the internal test set, on the public test set the temporal model yields the best results.

Split	Parameters	Acid 1		Base 1	
		$R^2$	$RMSE$	$R^2$	$RMSE$
Random	initial	0.34	2.55	0.47	2.10
	optimized	0.46	2.30	0.51	2.02
Scaffold	initial	0.36	2.50	0.47	2.09
	optimized	0.47	2.27	0.51	2.01
Temporal	initial	0.35	2.53	0.43	2.17
	optimized	0.49	2.23	0.52	1.99

Table 4.16: Validation scores on the public dataset of Chemprop models trained on the internal dataset.

Besides testing the multitask models on the OPERA data, referred to as *public* through the report, we also analyzed the models’ predictive performance on the SAMPL7 data. This dataset only consisted of one  $pK_a$  per compound without specifying whether its an acidic or basic value. Therefore, based on expert knowledge we assigned them into the most acidic  $pK_a$  category. The results of this experiment are presented in Table 4.17. As shown, the models can predict the Acid1  $pK_a$  of these compounds better than for the OPERA dataset. However, it has to be noted that there is significantly less data in the SAMPL7 dataset.

Split	Parameters	Acid 1	
		$R^2$	$RMSE$
Random	initial	0.75	1.24
	optimized	0.65	1.46
Scaffold	initial	0.47	1.79
	optimized	0.84	0.98
Temporal	initial	0.68	1.39
	optimized	0.78	1.15

Table 4.17: Validation scores on the SAMPL7 dataset of Chemprop models trained on the internal dataset.

### 4.3 Comparison to state of the art

In addition to evaluating the models on both internal and public datasets, we also compared the multitask model to existing software for  $pK_a$  prediction. In this comparison, we opted for the temporal model over the model trained on a random split of the data despite its lower accuracy. The decision was driven by the fact that the model’s primary purpose is to predict the  $pK_a$  of novel compounds, such that the temporal model generates more relevant results for real-life applications. For comparison, the temporal test set was used which contains the newest 20% of the compounds from the internal dataset. The multitask model used for the comparison is the temporal model with the initial parameters. We chose to use this model as it performed better on the test set than the model with the optimized hyperparameters.

Figure 4.5 and 4.6 present the  $R^2$  and  $RMSE$  values of the software and our temporal multitask model on the temporal test set. Here, *commercial method x* refers to different software, and *multitask model* refers to our temporal multitask model. The plots show that our model is better at predicting the most acidic and basic  $pK_a$  of compounds than all commercial methods, however, it has lower accuracy in predicting the second most acidic and basic  $pK_a$  than some methods.

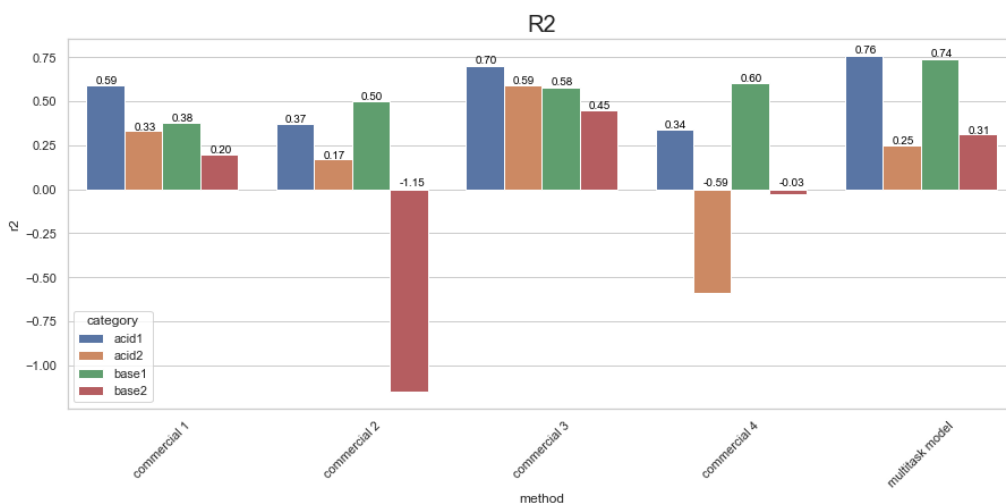


Figure 4.5:  $R^2$  score of software and the temporal multitask model on the temporal test set.

## 4. Results

---

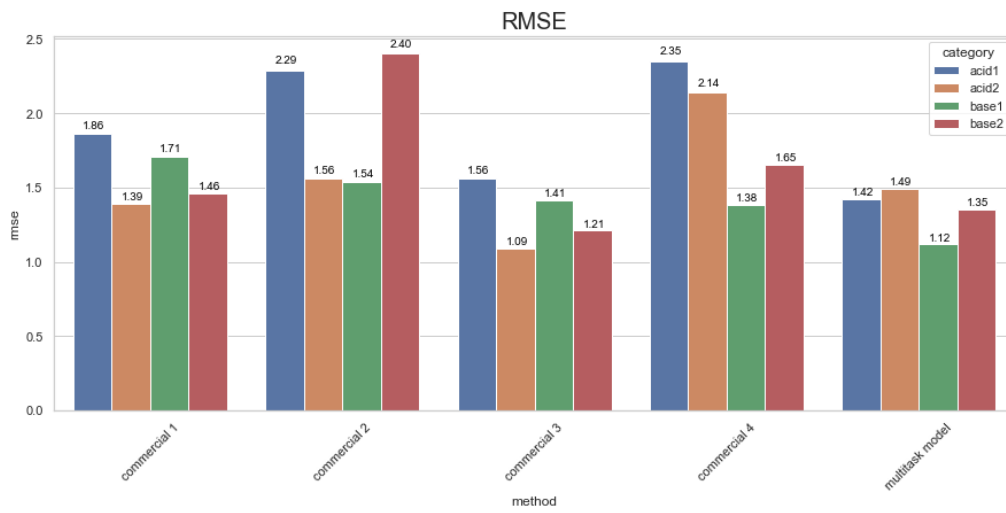


Figure 4.6:  $RMSE$  scores for the various proprietary softwares and the temporal multitask model using the temporal test set.

## 4.4 Potential to predict new modalities

Besides testing the model’s general performance, another interesting aspect to explore is its ability to predict  $pK_a$  in new modalities. This can provide an insight into the model’s generalization capabilities, and potential for application in real-world scenarios.

Modalities refer to different classes of molecules that are being used as drugs, with the most common being small molecules. New modalities that are available in our internal dataset include peptides, macrocycles and PROTACs.

First, we analyzed the temporally- and scaffold-split models’ potential to accurately predict  $pK_a$  in these new modalities. Then, we built a model using only small molecules and utilized it to make predictions on new modalities, which are generally larger than the average small molecule drug.

Table 4.18 presents the number of each modality in the training and test sets of the temporal and scaffold splits as well as the total numbers. As shown, the data consist of mostly small molecules and only 0.7% of it are new modalities. It can also be seen that the temporal split contains most of the new modalities in the test set, while in the scaffold split most macrocycles and PROTACs are in the training set.

Split	Small		Peptide		Macrocycle		PROTAC	
	Train	Test	Train	Test	Train	Test	Train	Test
Temporal	16748	4122	60	17	6	29	1	36
Scaffold	16693	4177	63	14	29	6	29	8
	20870		77		35		37	

Table 4.18: Number of modalities in the different splits with the total numbers in the bottom row.

The test set results of the multitask models trained with the optimized hyperparameters for the different splits are presented in Table 4.19 and 4.20. These show the number of compounds in each modality with the corresponding  $R^2$  and  $RMSE$  scores. As shown, the scaffold model performs better in predicting both the most acidic and basic  $pK_a$  than the temporal model. This can be attributed to the scaffold model having more new modalities in the training set. For predicting the second most acidic and basic  $pK_a$  none of the models perform well.

In Table 4.21 the results of the model built using only small molecules is presented. As shown, for acid 1 macrocycles and PROTACs can be predicted with high accuracy. For base 1 peptides and PROTACs have acceptable scores, however, the prediction of macrocycles lacks reliability.

	Acid 1			Acid 2		
	n	$R^2$	$RMSE$	n	$R^2$	$RMSE$
peptide	7	-1.34	3.05	1		
macrocycle	16	0.7	2.09	2		
PROTAC	25	0.53	1.4	5	-0.34	1.05
	Base 1			Base 2		
	n	$R^2$	$RMSE$	n	$R^2$	$RMSE$
peptide	16	-2.29	2.61	6	-1.19	2.07
macrocycle	25	-0.05	1.02	19	-0.5	0.94
PROTAC	33	0.53	1.15	20	-0.12	1.39

Table 4.19: Test set metrics by modality of model trained using a temporal split.

	Acid 1			Acid 2		
	n	$R^2$	$RMSE$	n	$R^2$	$RMSE$
peptide	2			1		
macrocycle	3	0.98	0.41	0		
PROTAC	3	0.68	0.93	0		
	Base 1			Base 2		
	n	$R^2$	$RMSE$	n	$R^2$	$RMSE$
peptide	13	0.62	0.59	2		
macrocycle	5	0.33	0.97	3	-6.09	0.31
PROTAC	8	0.55	1.18	3	-0.9	0.92

Table 4.20: Test set metrics by modality of model trained using a scaffold split.

	Acid 1			Acid 2		
	n	$R^2$	$RMSE$	n	$R^2$	$RMSE$
peptide	24	0.43	1.86	4	0.48	3.06
macrocycle	19	0.73	2.02	2		
PROTAC	25	0.8	0.9	5	-0.6	1.14
	Base 1			Base 2		
	n	$R^2$	$RMSE$	n	$R^2$	$RMSE$
peptide	70	0.48	1.35	18	-0.35	1.71
macrocycle	30	-0.04	0.97	20	-0.52	0.96
PROTAC	34	0.58	1.1	21	0.06	1.25

Table 4.21: Test set metrics of model trained on small molecules.

In Figure 4.7 we present a comparison between the different models for predicting new modalities. Here, the x-axis corresponds to the different categories (most acidic and basic, 2nd most acidic and basic  $pK_a$ ), the y-axis to the  $R^2$  /  $RMSE$  score and the color to the different models: *temporal*: model trained using the temporal split; *scaffold*: model trained using the scaffold split; and *small*: model trained on small molecules.

The plots show that overall the models can predict PROTACs the most accurately, and peptides the least accurately. The prediction of macrocycles is good in the case of acid 1 but not as reliable for other categories.

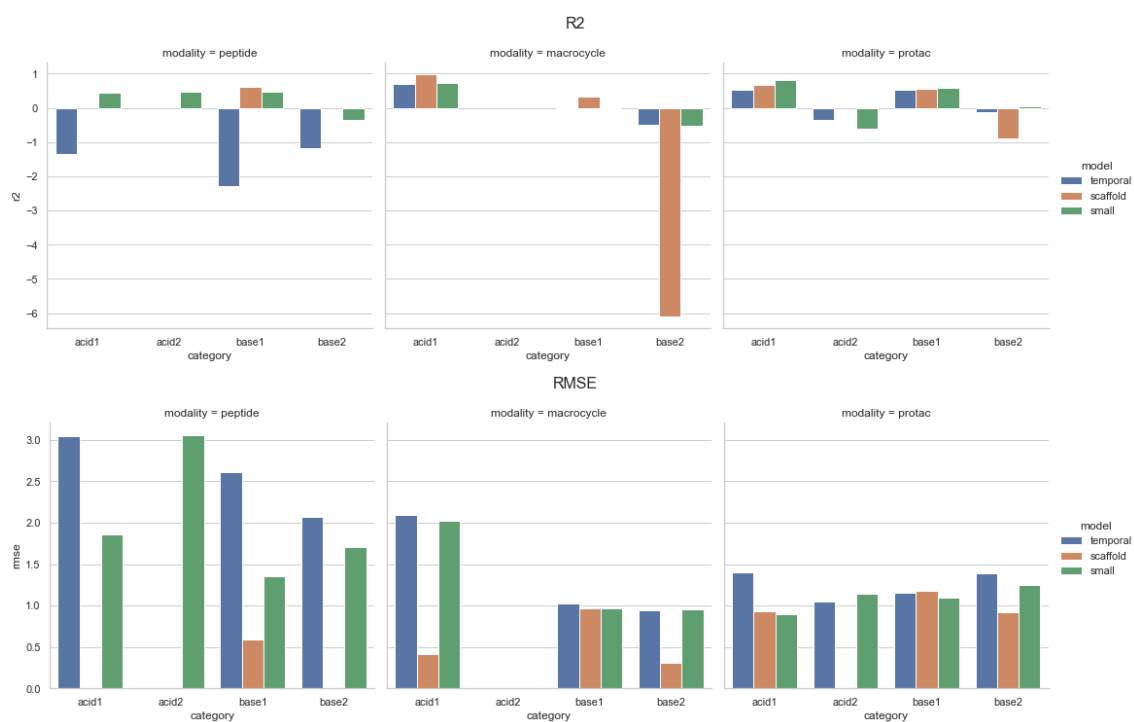


Figure 4.7:  $R^2$  and  $RMSE$  scores of different models on predicting new modalities.





# 5

## Conclusion

### 5.1 Discussion

The results of the single-task models show that using the RDKit descriptors with the LightGBM model yields the best performance for predicting both the acidic and basic  $pK_a$  of compounds. By implementing hyperparameter optimization, the model’s performance can be improved up to an  $R^2$  of around 0.7 and  $RMSE$  of 1.2 for the most acidic and basic  $pK_a$  on the test set with a random split in the data. When building the models using a temporal split, the best achievable  $R^2$  score is around 0.45 and the  $RMSE$  higher than 2 for the most acidic and basic  $pK_a$ . The metrics indicate slightly poorer results for the second most acidic and basic  $pK_a$  values in the random split, but their performance significantly deteriorates when using the temporal split.

The results of the multitask models demonstrate that the best model built with a random split in the data can predict both the most and second most acidic and basic  $pK_a$  of the compounds accurately. We observe an  $R^2$  greater than 0.8 for the most acidic and basic, and around 0.7 for the second most acidic and basic  $pK_a$  on the test set of the model built with a random split in the data. With the scaffold and temporal split models we can see a good performance on the most acidic and basic  $pK_a$ , but poorer performance when predicting the second most acidic and basic  $pK_a$ .

Comparing the multitask models to the single-task models, we observe better  $R^2$  and  $RMSE$  scores on the test sets with the multitask models. Even the initial multitask models outperform the best single-task models. We can also see a significant improvement in the prediction accuracy of the second most acidic and basic  $pK_a$  in the multitask models.

The results also show that the most acidic and basic  $pK_a$  can be predicted with higher accuracy than the second most acidic and basic  $pK_a$  for all models and splits. This can be attributed to the significantly lower number of experimental measurements for these categories in the dataset.

It can also be observed that better accuracy can be achieved with a random or scaffold-based split compared to a temporal split in the data. With a temporal split we observe a drop of around 5% in  $R^2$  for the most acidic and basic, and a drop of more than 30% for the second most acidic and basic  $pK_a$  for the best models. One

possible explanation for this could be that compounds measured more recently tend to be more complex compared to older ones. Consequently, the absence of recent compounds in the training set poses a greater challenge for the model in effectively predicting them. Moreover, in a real-life scenario a model built using a temporal split is more significant as it gives more information about the models generalization capabilities to new molecules. Despite the lower accuracy, the temporal multitask model shows a great performance and outperforms single-task models built on a temporal split.

We also note that testing the models on the public dataset yields a poor performance. This could be explained by the public dataset containing data collected from multiple sources for which various methods were used to measure  $pK_a$ .

Upon comparing the temporal multitask model with the state of the art approaches, it can be observed that our model not only competes with commercial methods but also outperforms them in predicting the most acidic and basic  $pK_a$  values.

When analyzing the models' potential to predict new modalities, we observe that reasonable accuracy can be achieved for the most acidic and basic  $pK_a$  of the compounds. However, the prediction of the second most acidic and basic  $pK_a$  does not yield good metrics. It is also notable that PROTACs can be generally predicted with higher accuracy than other novel modalities. An explanation for this could be that PROTACs are more similar in size to small molecules than for example macrocycles. This experiment showed that there is a possibility to extrapolate from small to large molecules. However, due to the limited data and high experimental variability in larger molecules we should not draw definitive conclusions.

## 5.2 Conclusion

This thesis focused on utilizing various molecular descriptors and machine learning models to predict the  $pK_a$  value of compounds. First, we explored the  $pK_a$  prediction capabilities of classical ML approaches, including random forests and boosted methods with various molecular descriptors, such as the Morgan and Maccs fingerprints, as well as RDKit's 2D descriptors. Then, we evaluated the performance of a graph neural network to predict the first and second most acidic and basic  $pK_a$  values. We also applied different data splitting strategies to investigate the models' generalization capabilities. Finally, we benchmarked our temporal model to commercial methods and explored different models' potential to predict the  $pK_a$  in novel modalities such as peptides, macrocycles and PROTACs.

Through thorough experimentation, we have demonstrated that the developed models can accurately and effectively predict the first and second most acidic and basic  $pK_a$  values. The models employed have also shown promising performance and outperformed various existing commercial method in predicting newer compounds. Finally, we have showed that the models have a great potential to predict the most acidic and basic  $pK_a$  of novel modalities, although the accuracy on new modalities is significantly reduced relative to small molecules.

# 6

## Future work

In the future, several avenues can be explored to further enhance the prediction of  $pK_a$  values using machine learning models.

A potential next step could be to further optimize the multitask models by investigating and improving the model architecture and conducting a more extensive hyperparameter optimization.

Another idea would be to incorporate additional molecular features, such as, molecular descriptors in the graph neural network. These descriptors have demonstrated promising performance when employed in classical machine learning approaches. Therefore, incorporating them into the graph neural network architecture could potentially further enhance the performance of the model.

Additionally, one could explore the possibility of using transfer learning to optimize the model further and extrapolate from small to large molecules. This could be done, for example, by using a model built on small molecules as a base model and fine-tuning it on new modalities. Alternatively, refining a model trained on internal data to more accurately predict public datasets would also be a potential next step.

An interesting area to explore is the application of active learning. Active learning has the key idea that the model can achieve greater accuracy with fewer labeled training data if it can choose which data it wants to learn from. The model may ask queries during training for an annotator to label unlabeled instances. This approach holds great potential in the context of  $pK_a$  prediction, where data availability is limited, and experimental measurements are both time-consuming and costly. By incorporating active learning, we can select the most informative and relevant data points for labeling, effectively maximizing the model’s learning capacity while minimizing the labeling effort.

Another avenue worth exploring is the integration of quantum mechanics-based (QM) methods. QM methods combine principles from quantum mechanics and machine learning to enhance the accuracy of ML models. These models have shown promising results in previous  $pK_a$  prediction challenges. However, it is important to consider the high computational cost associated with QM methods due to the complexity of quantum mechanical calculations. As an alternative, quantum chemistry-augmented graph neural networks could be employed for more accurate prediction of complex molecular properties like  $pK_a$  from the incorporation of a few quantum descriptors into the input molecular representations.



# Bibliography

- [1] J. P. Hughes, S. Rees, S. B. Kalindjian, and K. L. Philpott, "Principles of early drug discovery," *British journal of pharmacology*, vol. 162, no. 6, pp. 1239–1249, 2011.
- [2] D. Paul, G. Sanap, S. Shenoy, D. Kalyane, K. Kalia, and R. K. Tekade, "Artificial intelligence in drug discovery and development," *Drug discovery today*, vol. 26, no. 1, p. 80, 2021.
- [3] V. Patel and M. Shah, "Artificial intelligence and machine learning in drug discovery and development," *Intelligent Medicine*, vol. 2, no. 3, pp. 134–140, 2022.
- [4] Wikipedia, The Free Encyclopedia, *Acid dissociation constant*. [Online]. Available: [https://en.wikipedia.org/wiki/Acid\\_dissociation\\_constant](https://en.wikipedia.org/wiki/Acid_dissociation_constant).
- [5] K. Mansouri, N. F. Cariello, A. Korotcov, *et al.*, "Open-source qsar models for pka prediction using multiple machine learning approaches," *Journal of cheminformatics*, vol. 11, no. 1, pp. 1–20, 2019.
- [6] C. Hansch, A. Leo, and R. Taft, "A survey of hammett substituent constants and resonance and field parameters," *Chemical reviews*, vol. 91, no. 2, pp. 165–195, 1991.
- [7] D. D. Perrin, B. Dempsey, and E. P. Serjeant, *pKa prediction for organic acids and bases*. Springer, 1981, vol. 1.
- [8] T. Sander, J. Freyss, M. von Korff, and C. Rufener, "Datawarrior: An open-source program for chemistry aware data visualization and analysis," *Journal of chemical information and modeling*, vol. 55, no. 2, pp. 460–473, 2015.
- [9] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*, PMLR, 2017, pp. 1263–1272.
- [10] M. Ik, D. Levorse, A. S. Rustenburg, *et al.*, "Pka measurements for the sampl6 prediction challenge for a set of kinase inhibitor-like fragments," *Journal of computer-aided molecular design*, vol. 32, no. 10, pp. 1117–1138, 2018.
- [11] T. D. Bergazin, N. Tielker, Y. Zhang, *et al.*, "Evaluation of log p, pka, and log d predictions from the sampl7 blind challenge," *Journal of computer-aided molecular design*, vol. 35, no. 7, pp. 771–802, 2021.
- [12] L. David, A. Thakkar, R. Mercado, and O. Engkvist, "Molecular representations in ai-driven drug discovery: A review and practical guide," *Journal of Cheminformatics*, vol. 12, no. 1, pp. 1–22, 2020.

- [13] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.
- [14] National Center for Biotechnology Information, *Pubchem compound summary for cid 2244, aspirin*. [Online]. Available: <https://pubchem.ncbi.nlm.nih.gov/compound/Aspirin>.
- [15] L. Xue and J. Bajorath, "Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening," *Combinatorial chemistry & high throughput screening*, vol. 3, no. 5, pp. 363–372, 2000.
- [16] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, 2010.
- [17] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse, "Reoptimization of mdl keys for use in drug discovery," *Journal of chemical information and computer sciences*, vol. 42, no. 6, pp. 1273–1280, 2002.
- [18] G. Landrum, *Rdkit 2d descriptors*. [Online]. Available: <http://www.rdkit.org/docs/GettingStartedInPython.html#list-of-available-descriptors>.
- [19] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [20] A. Mayr, H. Binder, O. Gefeller, and M. Schmid, "The evolution of boosting algorithms," *Methods of information in medicine*, vol. 53, no. 06, pp. 419–427, 2014.
- [21] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.
- [22] K. Yang, K. Swanson, W. Jin, *et al.*, "Analyzing learned molecular representations for property prediction," *Journal of chemical information and modeling*, vol. 59, no. 8, pp. 3370–3388, 2019.
- [23] M. Dablander, "Out-of-distribution generalisation and scaffold splitting in molecular property prediction," *Oxford Protein Informatics Group*, 2021.
- [24] M. W. Browne, "Cross-validation methods," *Journal of mathematical psychology*, vol. 44, no. 1, pp. 108–132, 2000.
- [25] *Cross-validation: Evaluating estimator performance*. [Online]. Available: [https://scikit-learn.org/stable/modules/cross\\_validation.html#multimetric-cross-validation](https://scikit-learn.org/stable/modules/cross_validation.html#multimetric-cross-validation).
- [26] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [27] M. Ik, A. S. Rustenburg, A. Rizzi, M. R. Gunner, D. L. Mobley, and J. D. Chodera, "Overview of the sampl6 pka challenge: Evaluating small molecule microscopic and macroscopic pka predictions," *Journal of computer-aided molecular design*, vol. 35, no. 2, pp. 131–166, 2021.
- [28] M. Naser and A. Alavi, "Insights into performance fitness and error metrics for machine learning," *arXiv preprint arXiv:2006.00887*, 2020.
- [29] R. Nelsen, *Kendall tau metric. encyclopedia of mathematics*, 2001.

- [30] SAMPL6, “Sampl6 data,” DOI: 10.5281/ZENODO.2651393. [Online]. Available: <https://github.com/samplchallenges/SAMPL6>.
- [31] SAMPL7, “Sampl7 data,” DOI: 10.5281/ZENODO.5637494. [Online]. Available: <https://github.com/samplchallenges/SAMPL7>.
- [32] QSAR, “Qsar data,” [Online]. Available: <https://github.com/NIEHS/OPERA>.
- [33] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [34] G. Ke, Q. Meng, T. Finley, *et al.*, “Lightgbm: A highly efficient gradient boosting decision tree,” *Advances in neural information processing systems*, vol. 30, 2017.





# A

## Appendix

### A.1 Hyperparameters used in the optimized models

	n_estimators	max_depth
Acid 1	843	9
Acid 2	186	2
Base 1	983	29
Base 2	231	8

Table A.1: Optimal hyperparameters for the single-task LightGBM models with RDKit2D descriptors.

depth	dropout	hidden size	feed-forward layers
3	0.3	2100	3

Table A.2: Optimal hyperparameters for the multitask Chemprop models.

## A.2 Scatter plots of single-task and multitask models on the internal test set

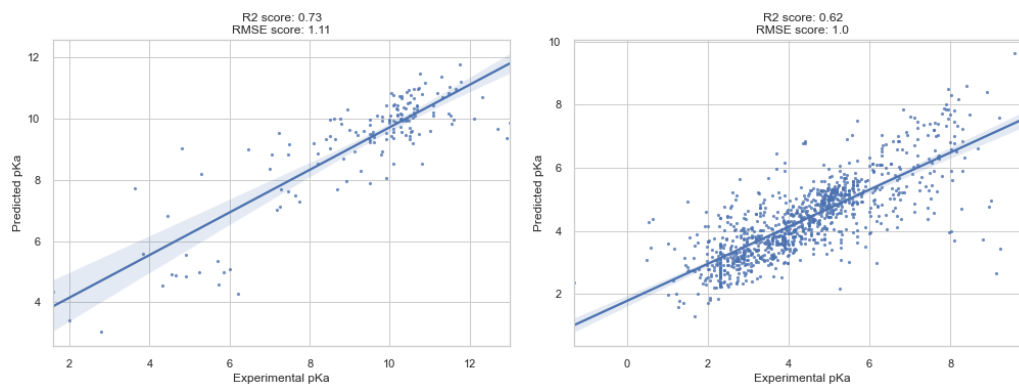


Figure A.1: Predicted and experimental acid 2 (left) and base 2 (right)  $pK_a$  values of the internal test set using the best random split models.

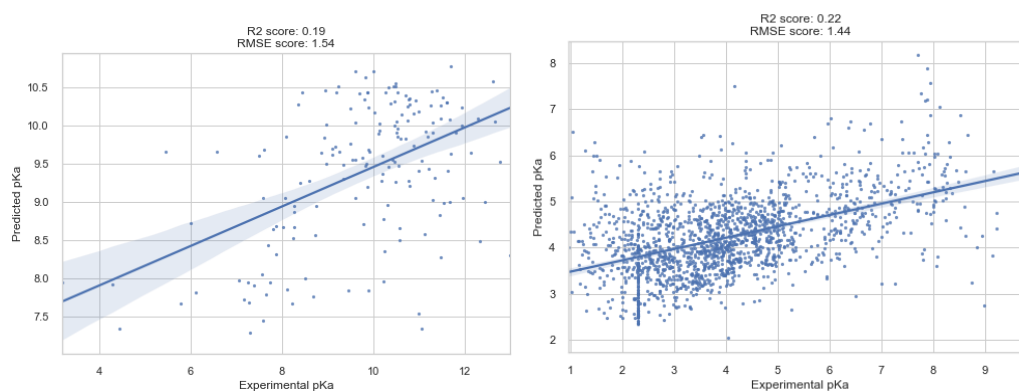


Figure A.2: Predicted and experimental acid 2 (left) and base 2 (right)  $pK_a$  values of the internal test set using the best temporal models.

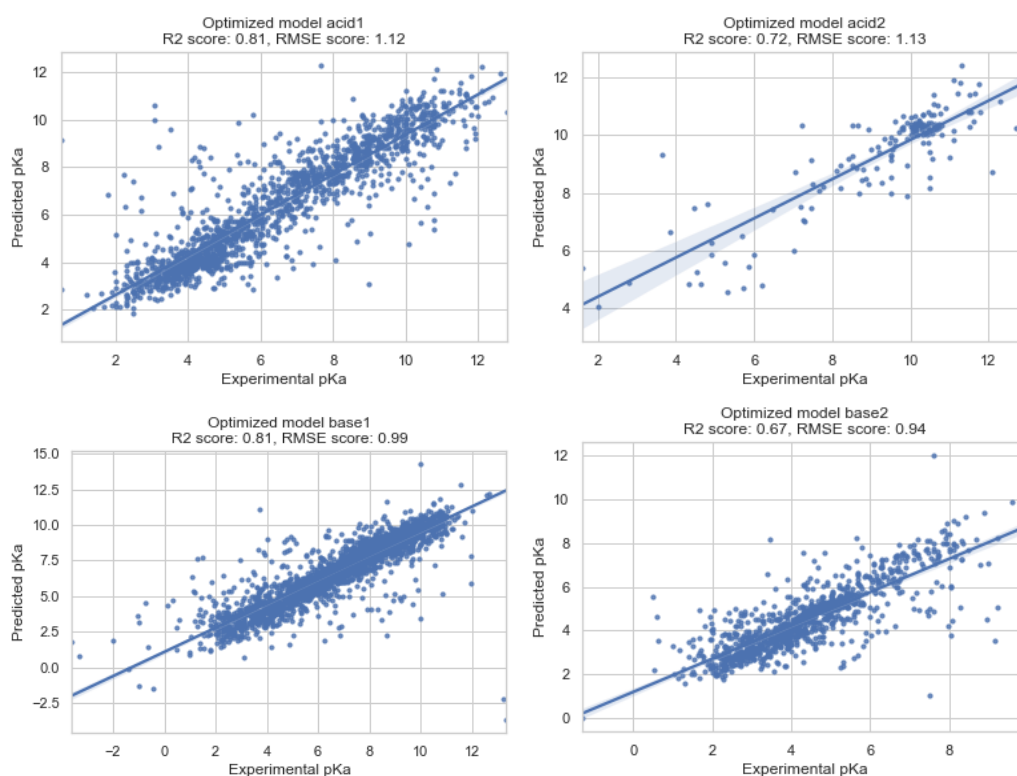


Figure A.3: Predicted and experimental  $pK_a$  values on the test set of the optimized Chemprop model built using a random split.

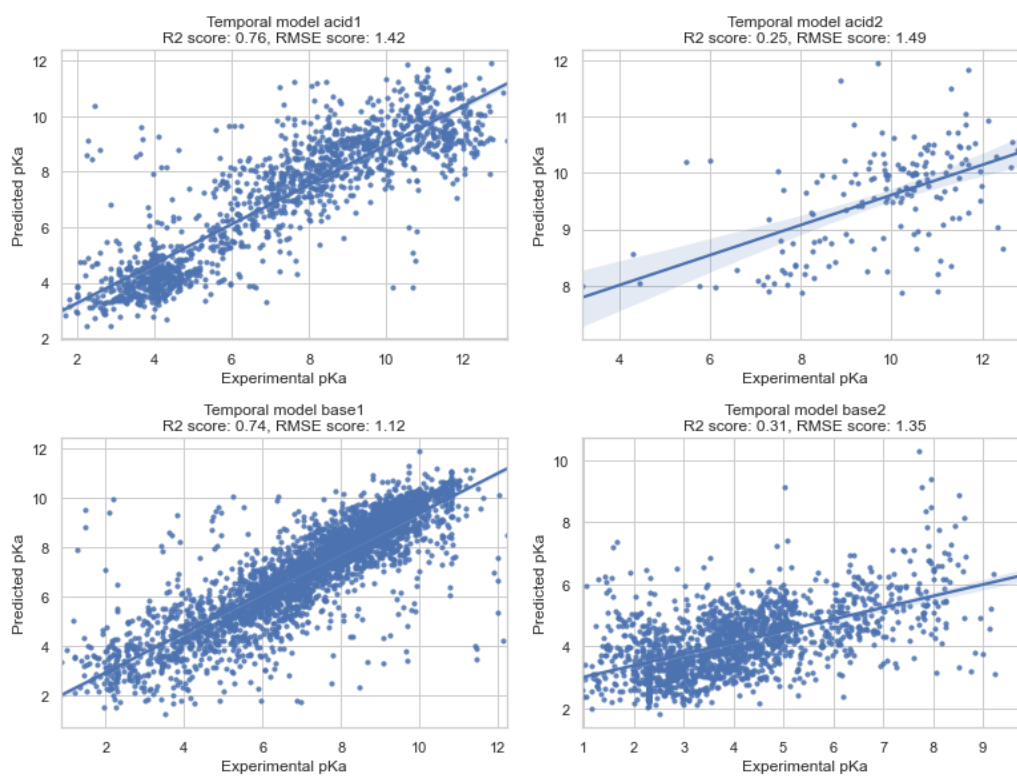


Figure A.4: Predicted and experimental  $pK_a$  values on the test set of the optimized Chemprop model built using a temporal split.

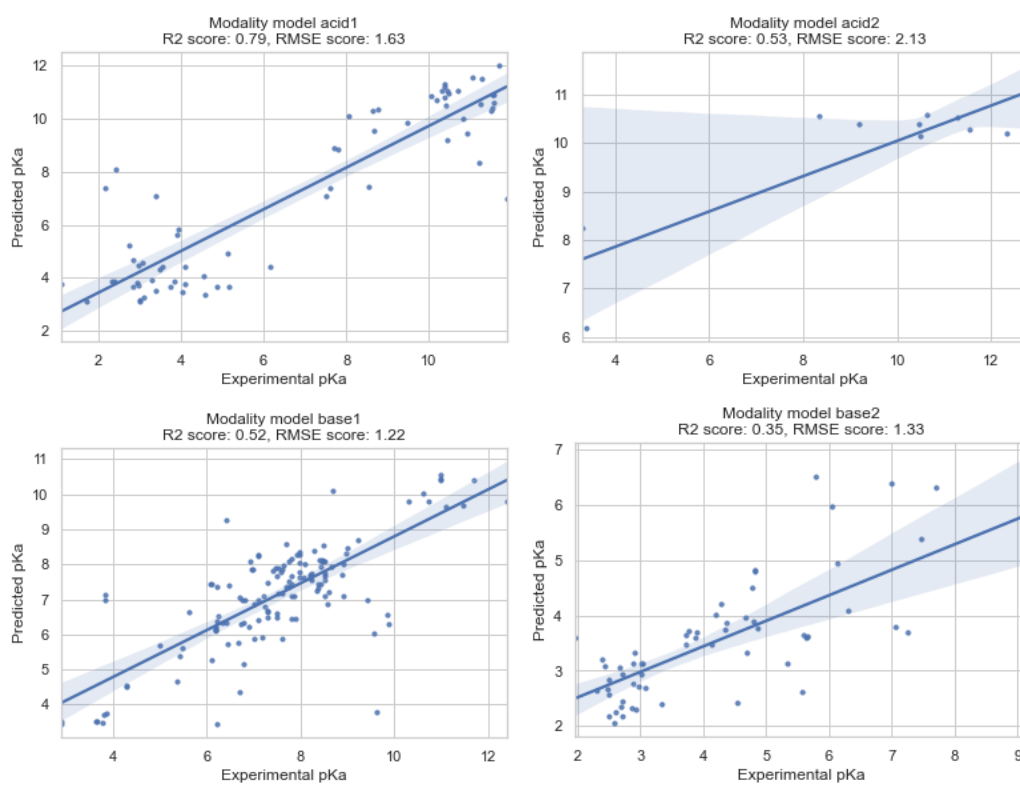


Figure A.5: Predicted and experimental  $pK_a$  values on the test set of the optimized Chemprop model built on small molecules.

### A.3 Scatter plots of multitask models on the SAMPL7 dataset

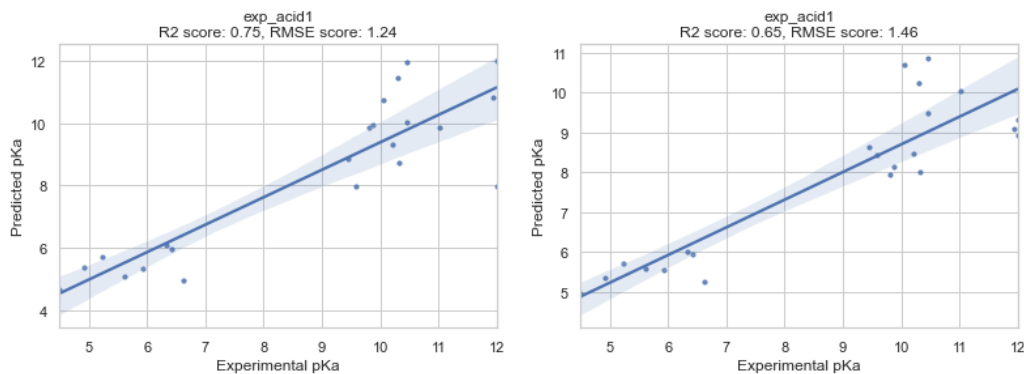


Figure A.6: Predicted and experimental  $pK_a$  values on the SAMPL7 test set of the Chemprop models built on a random split of the data. Model created using initial parameters on the left, optimized parameters on the right.

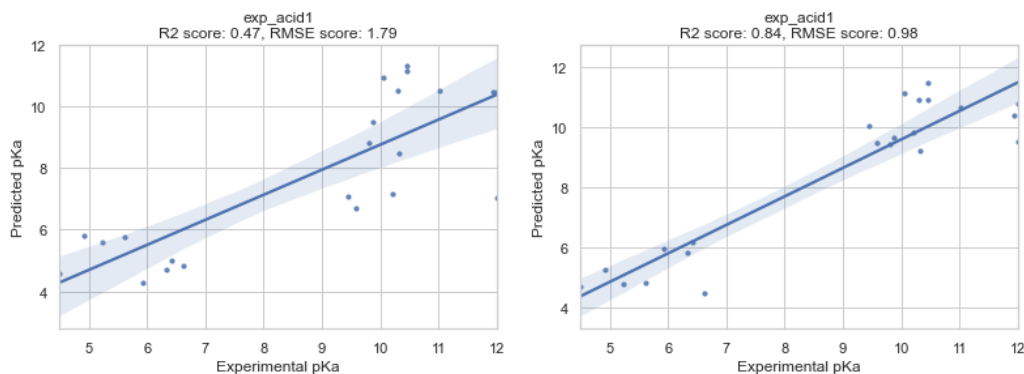


Figure A.7: Predicted and experimental  $pK_a$  values on the SAMPL7 test set of the Chemprop models built on a scaffold split of the data. Model created using initial parameters on the left, optimized parameters on the right.

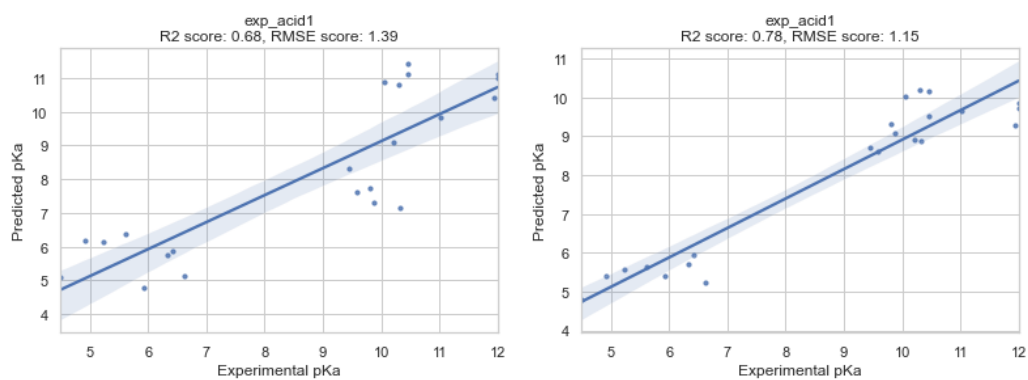


Figure A.8: Predicted and experimental  $pK_a$  values on the SAMPL7 test set of the Chemprop models built on a temporal split of the data. Model created using initial parameters on the left, optimized parameters on the right.